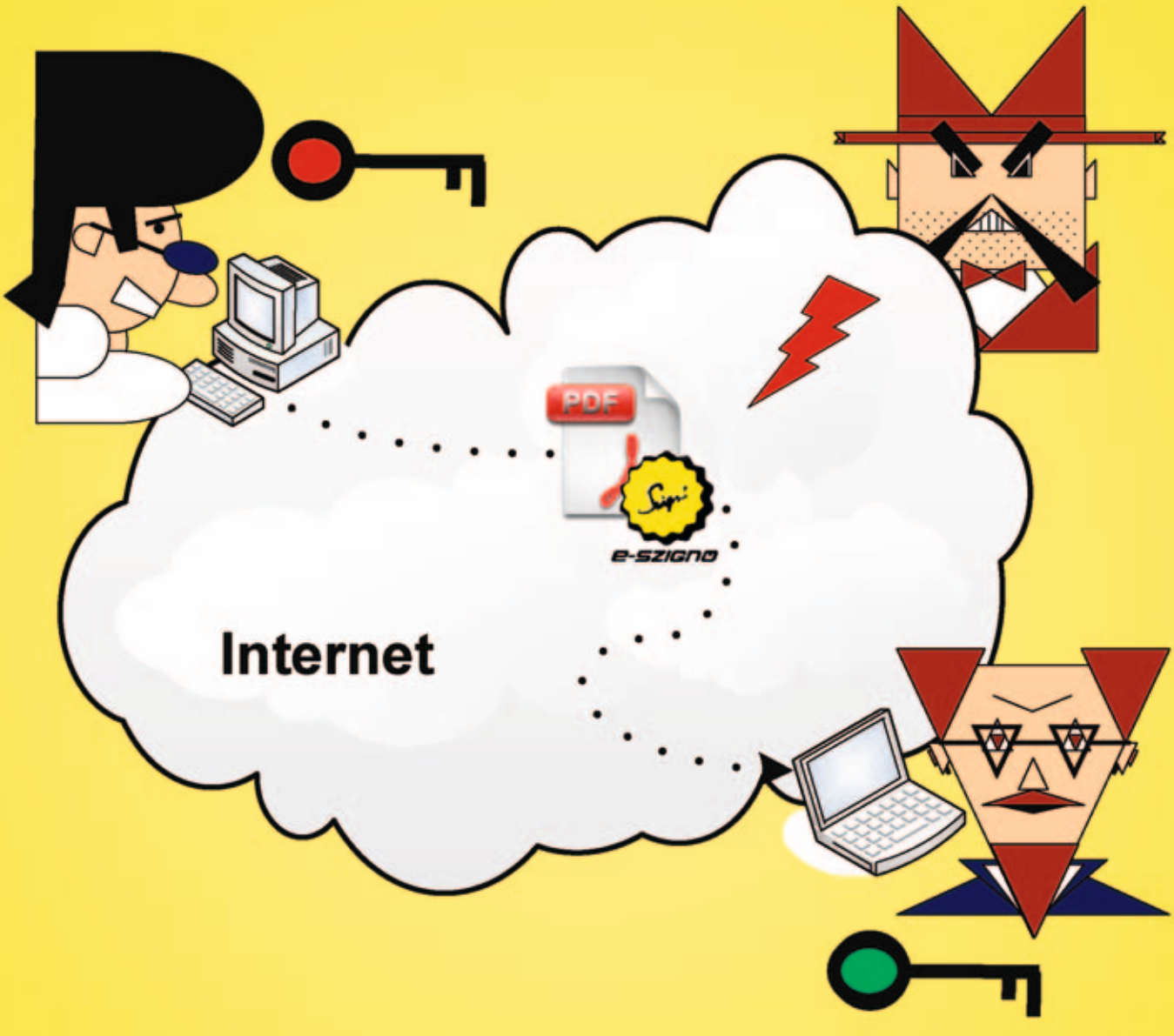
Dr. Berta István Zsolt

NAGY E-SZIGNÓ KÖNYV

amit az elektronikus aláírásról tudni akartál, csak féltél megkérdezni



Microsec

NAGY E-SZIGNÓ KÖNYV

Amit az elektronikus aláírásról tudni akartál, csak féltél megkérdezni

Dr. Berta István Zsolt

2011.

Írta:
Dr. Berta István Zsolt

Lektorálta:
Dr. Forgács Anna
Dr. Horváth András
Réti Kornél

ISBN: 978-963-08-1168-2

Kiadó:
Microsec Kft.

Készítette:
`www.konyvmuhely.hu`
Tel: (46) 532-085

Kinek ajánljuk ezt a könyvet?

Az elektronikus aláírással már kapcsolatban lévő, vagy a közeljövőben várhatóan hamarosan kapcsolatba kerülő, jellemzően informatikai, esetleg jogi vagy gazdasági területen dolgozó szakembereknek ajánljuk e könyvet.

A könyvben az elektronikus aláírással és a nyilvános kulcsú infrastruktúrával és ezek gyakorlatban történő alkalmazásával kapcsolatos ismereteket foglaljuk össze. Elsősorban műszaki szempontból közelítjük a témakört, de annak jogi és gazdasági vonatkozásaival egységesen tárgyaljuk.

A könyv bevezetésében átfogóan szólunk az elektronikus aláírásról, és röviden összefoglaljuk a könyv további fejezeteit. Az ezt követő fejezetekben kézikönyvszerűen írjuk le az elektronikus aláíráshoz kapcsolódó területeket. Minden fejezet legelején néhány szóval bemutatjuk a fejezet által érintett témakört, és az utolsó alfejezetében a fejezet legfontosabb megállapításait összegezzük.

A bevezetést követően a könyv további fejezetei akár önállóan is olvashatóak. Az egyes fejezetek megértéséhez jellemzően nem szükségesek a könyv korábbi fejezeteinek részletei. A műszaki vagy matematikai részletek iránt nem érdeklődő olvasónak javasoljuk, hogy ugorja őket át, és csak az összegző szakaszokat olvassa el.

Tartalom (áttekintés)

Tartalomjegyzék

1. fejezet

Bevezetés

„Toto, I've got a feeling we're not in Kansas anymore"
(Totó, az az érzésem, már nem Kansas-ben vagyunk)
– Az Óz, a csodák csodája c. filmből

Az elektronikus kommunikáció mára már szinte teljesen felváltotta a papír alapút. Ügyeinket weben, e-mailen, mobil- vagy vezetékes telefonon intézzük, és a hagyományos újságok és folyóiratok szerepét is egyre inkább átveszik a rádió- és tévécsatornák, illetve az internetes hírportálok.

Elektronikusan könnyebben hozható létre, és könnyebben kezelhető az információ, így általában a papírra kerülő tartalmat is elektronikusan készítjük el. Ezen túl az információ elektronikusan gyorsabban továbbítható és jobban „menedzselhető", mintha papíron lenne. Egyetlen területen használunk továbbra is papírokat: ha *hiteles* iratokat kezelünk, ha az a célunk, hogy bizonyítható legyen, hogy ki, kinek, mikor és mit küldött, vagy milyen nyilatkozatot tett. Ekkor az elektronikus dokumentumok előnyei – éppen az, hogy könnyű létrehozni, könnyű nyom nélkül módosítani – hátránnyá válnak.

Egy irat hitelessége általában azt jelenti, megállapítható róla, hogy ki készítette, és a tartalma nem változott meg a készítés óta. Nem feltétlenül a készítő kiléte érdekel bennünket, és gyakran nem vagyunk kíváncsiak sem az illető nevére, sem a személyes adataira. Sokszor csak az a fontos, hogy az illető személy vagy szervezet milyen minőségben bocsátotta ki az adott dokumentumot, és gyakran az is elegendő, ha tudjuk, szükség esetén visszakereshető, hogy ki volt az illető. Lényeges, hogy a *hitelesség megállapítható legyen*, ezért – a hitelesség ellenőrizhetőségét elősegítendő – hitelesítő elemeket szokás hozzáadni lényeges dokumentumokhoz.

Az emberiség régen felismerte, hogy a „fontos" információt célszerű hitelesíteni, *írásba foglalni*. Ha csak szóban állapodunk meg egymással, könnyen félreérthetjük egymást, vagy vita támadhat arról, hogy pontosan miben állapodtunk meg. A rováspálca volt az egyik

első hitelesítési módszer. A pálcára rovásjelekkel vésték fel a megállapodást (például, hogy egy pásztor hány birkát visz legeltetni), a pálcát hosszában kettéhasították, majd egyik fél (a pásztor) megkapta az egyik felét, a másik fél (a birkák tulajdonosa) pedig a másik felét kapta meg. [102] Más információk hitelességét a nyilvános elérhetőségük biztosította. Az ókori Rómában a „tizenkét táblás törvények" kint függtek a Fórumon, hogy mindenki szabadon elolvashassa őket, és hivatkozhasson rájuk. Sok esetben pecsét igazolta egy irat hitelességét. Pecsétet ma leginkább intézmények használnak, de például Japánban ma is általános, hogy az emberek saját, személyes pecsétnyomóval (ún. hanko) rendelkeznek, és ügyleteiket ezzel hitelesítik. Az írástudás elterjedésével a pecséteket egyre inkább felváltotta az *aláírás*.

Az aláírás hozzákapcsolódik a dokumentumhoz, és egyedileg jellemző az aláírást készítő személyre. Aki papír alapú, aláírt dokumentumot próbál hamisítani, az vagy az eredeti papíron lévő tartalmat manipulálja (pl. szavak eltüntetésével vagy új oldalak beszúrásával), vagy egy teljesen új dokumentumra próbálja „rátenni" egy másik személy kézzel írott aláírását (ez történhet pl. fénymásolással vagy a másik aláírás utánzásával). Mindkét módszer nyomokat hagy, amelyeket vitás esetekben meg lehet vizsgálni. Ezzel szemben, az egyszerű elektronikus dokumentumokról – éppen az elektronikus dokumentumok előnyei miatt – nagyon könnyű olyan „tökéletes" hamisítványt készíteni, amelyen nincs nyoma a manipulációnak, így a dokumentum vizsgálatával nem dönthető el, hogy a dokumentumot manipulálták-e. (Ha bizonyítani szeretnénk, hogy ki, kinek, mikor és mit küldött, a dokumentumot tároló és továbbító rendszerek naplóit szokás megvizsgálni.)

Ezért amikor „hivatalos" iratot küldünk, általában számítógépen szerkesztjük meg, majd kinyomtatjuk, aláírjuk, borítékba tesszük, és postán küldjük el a címzettnek. Ha a címzett nem magánszemély, hanem számítógépes rendszerrel rendelkező hivatal vagy szervezet, akkor miután kibontotta a levelet, begépeli, számítógépre viszi annak tartalmát, ugyanakkor az aláírt levelet is megőrzi, hogy szükség esetén bizonyítani tudja, hogy mi küldtük.

1.1. Hitelesség elektronikusan

Mindez ma már sokkal egyszerűbben is történhetne. A világ fejlődésével változik a médium – ma már nem pálcákra, nem agyag- vagy kőtáblákra, és nem elefántcsont-lapokra írunk –, és változnak a hitelesítő elemek is. Az alapelv nem változik: a fontos információkat leírjuk, és az írást hitelesítjük. Ha papír helyett elektronikus médiát használunk, elektronikus hitelesítő megoldásokra van szükség.

Hiteles dokumentumokat nemcsak papír alapon, hanem elektronikusan is létrehozhatunk. Míg a papír alapú dokumentumok esetén a dokumentum hitelességét a rajta szereplő kézzel írott aláírás biztosítja, *az elektronikus dokumentumokat elektronikus aláírással hitelesíthetjük.* Az elektronikus aláírás nem a beszkennelt kézzel írott aláírást jelenti, hanem a kódolás egy speciális változata. Ha egy dokumentumot elektronikusan írunk alá, akkor olyan módon

kódoljuk, hogy a létrejött kódolt dokumentum hitelességét annak szerkezete, kódolása biztosítja. *Az így kódolt, azaz elektronikusan aláírt dokumentum hitelességét jogszabály* – az elektronikus aláírásról szóló 2001. évi XXXV. törvény – *is elismeri.* [180] Az ún. fokozott biztonságú elektronikus aláírással hitelesített dokumentum írásba foglaltnak minősül, az ún. minősített elektronikus aláírással pedig teljes bizonyító erejű magánokirat vagy akár közokirat is készíthető.

Könyvünkben a nyilvános kulcsú infrastruktúra (public key infrastructure, PKI) eszköztárát és felhasználási lehetőségeit mutatjuk be. E technológia segítségével egymást korábban nem ismerő felek biztonságosan kommunikálhatnak, és hiteles elektronikus dokumentumokat hozhatnak létre. Könyvünkben a PKI eszköztárát, majd ezen eszközök felhasználási lehetőségeit tekintjük át. Jelen bevezetés hátra lévő részében a PKI alapfogalmairól nyújtunk rövid áttekintést, az itt használt fogalmakat a későbbi fejezetekben fejtjük ki részletesen.

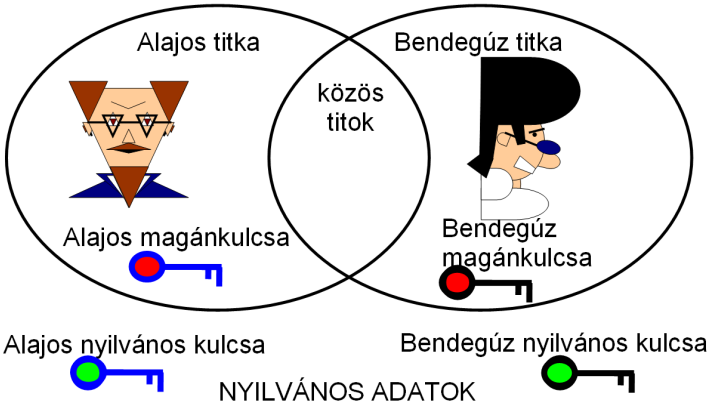
1.2. Aláírás, Titkosítás, Autentikáció

A nyilvános kulcsú infrastruktúra ún. *kriptográfiai kódolásokra* épül. A kódolás olyan matematikai művelet, amelynek során egy dokumentumot egy másik, kódolt dokumentummá alakítunk át. A kódolás során gyakran ún. *kulcsot* is felhasználunk, a kódolás e kulcs segítségével kapcsolódik a nyilvános kulcsú infrastruktúra valamely szereplőjéhez. A kulcs maga is információ, például egy nagy számként képzelhető el. A kriptográfiáról, és a kriptográfiai kulcsokról a 2. fejezetben írunk részletesen.

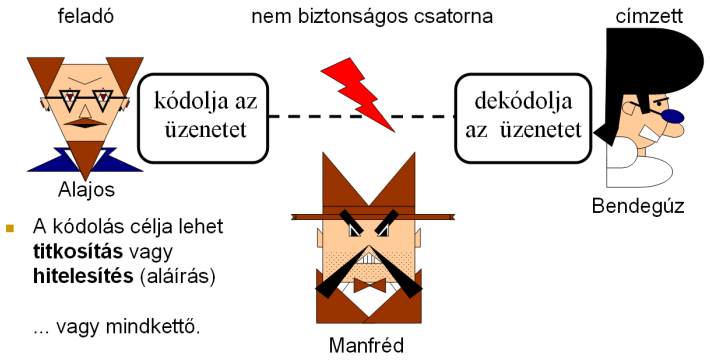
A nyilvános kulcsú infrastruktúra minden szereplőjének van nyilvános kulcsa és magánkulcsa. (Lásd: 1.1. ábra.) Minden szereplő titokban tartja a saját magánkulcsát (ezt jellemzően csak ő ismerheti), míg a nyilvános kulcsukat nem kell titokban tartaniuk, akár nyilvánosságra is hozhatják őket. Amikor nem biztonságos csatornán (pl. Interneten) keresztül kommunikálnak egymással, e kulcsok segítségével *kódolják* az üzeneteiket. (Lásd: 1.2. ábra.)

A PKI segítségével elvégezhető kódolási műveleteket három nagy csoportba sorolhatjuk, attól függően, hogy milyen célból kódolunk e kulcsokkal:

- *Aláírhatunk* dokumentumokat, így az aláírt dokumentumokról később bárki megállapíthatja, hogy az adott dokumentumhoz tartozó aláírást mi készítettük. (E lépést dokumentum-hitelesítésnek is nevezzük.) Az aláírás jellemzően az aláírt dokumentumnak (vagy egy hozzáfűzött záradéknak) az elfogadását vagy tudomásul vételét jelenti. Aláíráshoz a saját magánkulcsunkra van szükség, míg aki ellenőrizni szeretné az aláírást (meg akar győződni róla, hogy mi írtunk alá egy adott dokumentumot, és mi pontosan azt a dokumentumot írtuk alá), annak a mi nyilvános kulcsunkra van szüksége. Az elektronikus aláírást a 6. fejezetben mutatjuk be részletesen.



1.1. ábra. **Minden szereplőnek van két kulcsa. A magánkulcsát mindenki titokban tartja, a nyilvános kulcsokat bárki megismerheti.**



1.2. ábra. **Alajos és Bendegúz nem biztonságos csatornán kommunikálnak egymással, üzeneteiket kódolással védik a támadó – Manfréd – ellen**

- *Titkosíthatunk* dokumentumokat, így azok tartalmát kizárólag arra jogosult felek ismerhetik meg. Titkosításhoz a címzett (a dokumentum elolvasására jogosult fél) nyilvános kulcsára van szükségünk, és a címzett a saját magánkulcsával fejtheti vissza a titkosított dokumentumot. A titkosítást a 9. fejezetben mutatjuk be részletesen.

- *Biztonságos csatornát létesíthetünk* valakivel. A csatorna felépítése során meggyőződünk róla, hogy akivel a csatornát felépítjük, az valóban rendelkezik egy adott *nyilvános kulcs*hoz tartozó magánkulccsal (e lépést nevezik partner-hitelesítésnek[1] vagy *autentikációnak* is), majd titkosított és hitelesített csatornát építhetünk ki az illetővel. Partnerünk hitelesítéséhez az illető nyilvános kulcsára van szükségünk, míg neki a saját magánkulcsával kell kódolnia. Az így kiépült csatornát harmadik fél (aki nem ismeri a felek magánkulcsait) nem hallgathatja le, és nem tud közbeékelődni sem. Maga a csatorna titkosított, de kizárólag addig védi üzeneteinket, amíg azok a csatornán haladnak keresztül. Miután egy üzenet elhagyta a biztonságos csatornát, már lehallgatható lehet, illetve ekkor már semmi nem igazolja hitelességét. (Ha üzenetünket nemcsak a csatornán való továbbítás során szeretnénk védeni, akkor célszerű aláírni, illetve titkosítani.) Az autentikációt a 10. fejezetben mutatjuk be.

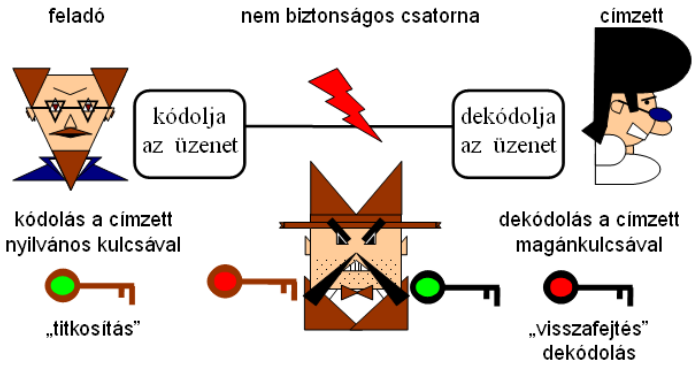
A PKI e három célra nagyon hasonló eszköztárat nyújt, de ezen eszközöket mégis markánsan különböző módon kell használni, attól függően, hogy aláírásra, titkosításra vagy autentikációra használjuk őket. Könyvünk nagyrészt e három fontos eset közti különbségeket magyarázza el.

1.3. Tanúsítványok és hitelesítés-szolgáltatók

A PKI alapján akkor kommunikálhatunk valakivel biztonságosan, ha megszerezzük az illető nyilvános kulcsát. A nyilvános kulcshoz nyilvános csatornán is hozzájuthatunk, de feltétlenül *hitelesen*, például tanúsítvány formájában kell megszereznünk, mert ha nem tudjuk, hogy kié egy adott nyilvános kulcs, lehet, hogy éppen a támadóval kommunikálunk „biztonságosan”.

> ***1.1. Példa:*** *Manfréd, a támadó, elhiteti Alajossal, hogy egy adott nyilvános kulcs Bendegúz nyilvános kulcsa. Becsapja Alajost, mert a kulcs Manfréd saját nyilvános kulcsa, és a magánkulcsot kizárólag Manfréd ismeri. Ha Alajos Bendegúznak küld titkosított üzenetet, Manfréd nyilvános kulcsával titkosít, így ezt Manfréd el tudja olvasni. Manfréd visszafejti az üzenetet, majd Bendegúz valódi nyilvános kulcsával titkosítja, és így küldi tovább Bendegúznak. Ez egy ún. man-in-the-middle támadás.*
>
> *Nagyon fontos, hogy partnerünk nyilvános kulcsát hiteles módon szerezzük meg, különben áldozatul eshetünk egy hasonló támadásnak.*

[1] A „hitelesítés” szó önmagában nem elengedő, mert ez könnyen összekeverhető lehet a dokumentum-hitelesítéssel, a partner-hitelesítés kifejezés pedig túlságosan körülményes, ezért a továbbiakban az „autentikáció” kifejezést használjuk e fogalomra.



1.3. ábra. **Man-in-the-middle támadás – Alajos azt hiszi, hogy Bendegúz nyilvános kulcsával titkosítja az üzeneteit, de Manfréd kulcsát használja. Manfréd visszafejti az üzenetet a saját magánkulcsával, majd Bendegúz nyilvános kulcsával titkosítva küldi el Bendegúznak.**

A tanúsítvány egy (elektronikusan) aláírt igazolás, amelyben egy megbízható fél, egy hitelesítés-szolgáltató igazolja, hogy egy adott nyilvános kulcs egy adott szereplőhöz tartozik. A tanúsítványokról a 3. fejezetben, a hitelesítés-szolgáltatókról a 4. fejezetben szólunk részletesen.

Egy hitelesítés-szolgáltató aláírását is a hitelesítés-szolgáltató tanúsítványa alapján ellenőrizhetjük. A szolgáltatói tanúsítvány is egy elektronikusan aláírt igazolás a szolgáltató nyilvános kulcsáról. Ebben vagy eleve megbízhatunk, mert hiteles módon (4.1.4. fejezet) jutottunk hozzá – ekkor ún. megbízható gyökértanúsítványról beszélünk –, vagy egy másik hitelesítés-szolgáltató tanúsítványa alapján ellenőrizhetjük. Egy végfelhasználói tanúsítványtól a gyökértanúsítványig általában egy vagy több „köztes” szolgáltatói tanúsítványon keresztül jutunk el, ezen tanúsítványok láncolatát tanúsítványláncnak nevezzük. A tanúsítványláncokról az 5. fejezetben írunk részletesen.

1.4. A PKI és a jogszabályok

1.4.1. Miért fontosak a PKI világában a jogszabályok?

Titkosítás és autentikáció jogszabályi háttér nélkül is elképzelhető. Ha egy üzenetet titkosítunk, azt kizárólag a dekódoló kulcsot birtokló felek képesek visszafejteni, függetlenül attól, hogy a jogszabályok mit mondanak. Hasonlóképpen, ha például egy webszerverre csak adott típusú tanúsítvánnyal (és a hozzá tartozó magánkulccsal) rendelkező felek juthatnak be, akkor a szervert más nem képes elérni.

Titkosítás és autentikáció esetén is igaz, hogy ha egy tanúsítvány alapján valaki fontos döntést hoz (például rábízza az adatait egy a tanúsítványhoz tartozó magánkulcs birtokosára), kára származhat belőle, ha a tanúsítványt kibocsátó hitelesítés-szolgáltató elront valamit a

tanúsítvánnyal kapcsolatban. Ez esetekben is célszerű tisztázni a felelősségi viszonyokat, de a titkosítási és autentikációs technikák jogi szabályozás nélkül is jól használhatóak sok területen.

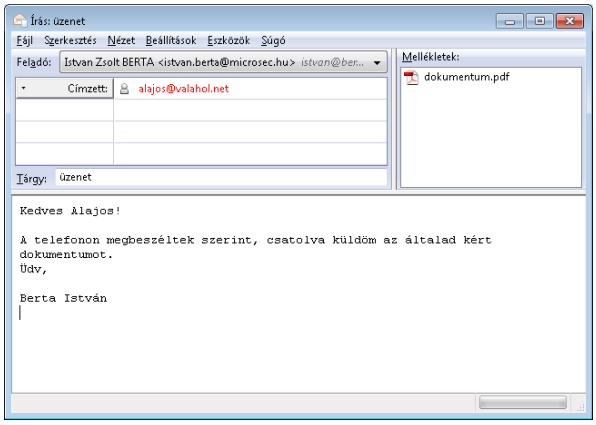
Az elektronikus aláírásnak azonban elsősorban akkor van értelme, ha hitelességét jogszabály is elismeri. Ha nem lenne jogszabály az elektronikus aláírásról, akkor is igaz volna, hogy egy aláírást a megfelelő magánkulcs nélkül nem lehet létrehozni, de ekkor minden egyes alkalommal, amikor egy aláírt dokumentumról vita támad és a dokumentum bíróság elé kerül, meg kellene győzni a bíróságot az elektronikus aláírás technológiája által biztosított hitelességről. A bíróságnak azt is meg kellene vizsgálnia, hogy az adott technológia alkalmazható-e az adott környezetben, és az alkalmazott kulcsgondozási megoldások megfelelő szintű biztonságot nyújtottak-e. Ez esetben nagyon kevesen mernének belevágni az elektronikus aláírás használatába.

1.4.2. Az elektronikus aláírásról szóló törvény

Magyarországon a PKI területén egyedül az elektronikus aláírásra vonatkozik részletes jogi szabályozás. A legfontosabb, elektronikus aláírásról szóló jogszabály az elektronikus aláírásról szóló 2001. évi XXXV. törvény (Eat). [180] Az Eat. az elektronikus aláírásról szóló 1999/93/EK EU irányelvre épül [62], és négy ún. elektronikus aláírással kapcsolatos szolgáltatást határoz meg:

- elektronikus aláírás hitelesítés-szolgáltatást, amely az aláíró tanúsítványok kibocsátását és fenntartását jelenti (röviden: *hitelesítés-szolgáltatás*);
- *időbélyegzés-szolgáltatás*t, amely szerint időbélyegek hozhatók létre, és ezen időbélyegekkel igazolható, hogy egy dokumentum (vagy aláírás) egy adott időpillanatban már létezett;
- aláírás-létrehozó eszközön aláírás-létrehozó adat elhelyezése szolgáltatást (röviden: *eszköz-szolgáltatás*);
- *elektronikus archiválás-szolgáltatást*, amely az elektronikusan aláírt dokumentumok hosszú távú hiteles megőrzésére szolgál.

Az Eat. az egyes szolgáltatások esetén megkülönböztet minősített és nem minősített szolgáltatókat. A minősített szolgáltatókra erősebb követelmények (szigorú ellenőrzés, anyagi felelősségvállalás, kötelező felelősségbiztosítás) vonatkoznak, így a minősített szolgáltatók által nyújtott szolgáltatásokhoz nagyobb bizonyító erő tartozik. Ha minősített szolgáltató nyújt egy szolgáltatást, akkor abból kell kiindulni (vélelmezni kell), hogy a szolgáltatást „jól” nyújtja, és annak kell bizonyítani az állítását, aki ezt mégis megkérdőjelezi. Például ha egy dokumentumon minősített szolgáltató által kibocsátott, érvényes időbélyeg szerepel, vélelmezni kell, hogy a dokumentum az időbélyegen szereplő időpontban már létezett.



1.4. ábra. **Egyszerű elektronikus aláírásnak minősül, ha egy e-mail végére odaírom a nevemet.**

Minősített elektronikus aláírás csak minősített hitelesítés-szolgáltató által kibocsátott minősített tanúsítvány alapján hozható létre, akkor és csak akkor, ha az aláírás-létrehozó adat biztonságos aláírás-létrehozó eszközön helyezkedik el. Ha egy dokumentumon érvényes minősített aláírás szerepel, vélelmezni kell, hogy a dokumentum az aláírás időpontja óta nem változott (minősített hitelesítés-szolgáltatás), és az aláíráshoz használt magánkulcs az aláíró birtokában volt (minősített eszköz-szolgáltatás). (Lásd: 6.1. fejezet.)

Az Eat. szerint létezik a minősített elektronikus aláírásnál alacsonyabb biztonsági szint is, ez az ún. fokozott biztonságú elektronikus aláírás. A fokozott biztonságú elektronikus aláírással ellátott dokumentum írásba foglaltnak minősül, de nem kapcsolódik hozzá a fenti vélelem.

Létezik olyan elektronikus aláírás is, amely még fokozott biztonságúnak sem minősül (ilyen például ha valaki odaírja a nevét egy dokumentum végére), erről az Eat. mindössze annyit mond, hogy nem lehet önmagában azért elutasítani, mert elektronikusan létezik. Ezt korábban „egyszerű elektronikus aláírás” néven is nevezték, de a 2004. évi Eat. módosítás óta már nincsen külön neve. (lásd: 1.4. ábra)

Az Eat. értelmében a szolgáltatókról – a minősített szolgáltatókról és a nyilvános körben szolgáltatást nyújtó nem minősített szolgáltatókról – a Nemzeti Média- és Hírközlési Hatóság vezet nyilvántartást, illetve folyamatosan vizsgálja és ellenőrzi ezen szolgáltatók működését.

Az Eat. és a hozzá kapcsolódó 3/2005. IHM rendelet követelményeket – például biztonsági és pénzügyi követelményeket – határoz meg az elektronikus aláírással kapcsolatos szolgáltatásokat nyújtó szolgáltatókra, illetve szabályozza ezen szolgáltatók felelősségét is.

1.4.3. Az elektronikus aláírásról szóló további jogszabályok

Az elektronikus aláírás területéhez más jogszabályok is kapcsolódnak. A digitális archiválás szabályairól szóló 114/2007. GKM rendelet meghatározza, hogy elektronikus adatokat milyen

módon lehet hitelesen archiválni, itt egyik lehetőségként az elektronikus aláírással történő hitelesítést jelöli meg. [69] Az ÁFA törvény [185] valamint az elektronikus számlázásról szóló 46/2007. PM rendelet [138] kimondják, hogy számlát kizárólag elektronikusan, papír felhasználása nélkül is ki lehet bocsátani, a 24/1995. PM rendelet értelmében pedig – a rendeletben meghatározott feltételek mellett – a papír alapú számlák másodpéldányai is megőrizhetőek kizárólag elektronikus módon. A 13/2005. IHM rendelet azt szabályozza, hogy egy papír alapú dokumentumról hogyan készíthető hiteles elektronikus másolat. [79]

Ezen kívül egyes speciális területekre vonatkozó jogszabályok is tartalmaznak az elektronikus aláírás használatára vonatkozó rendelkezéseket. Például a közjegyzői törvény meghatározza, hogy a közjegyzőnek hogyan, milyen módon kell minősített aláírást használnia, és hogyan kell archiválnia az elektronikusan aláírt iratait. [182] Könyvünk 12. fejezetében a PKI és az elektronikus aláírás gyakorlati alkalmazási területeit tekintjük át ezen jogszabályoknak megfelelően, illetve egy elektronikus aláírás létrehozására szolgáló professzionális alkalmazás működését mutatjuk be.

1.5. A PKI szereplői

A nyilvános kulcsú infrastruktúra (angolul: public key infrastructure, PKI) arra nyújt lehetőséget, hogy egymást korábban nem ismerő felek nyilvános hálózaton keresztül is biztonságosan kommunikálhassanak egymással. A PKI szereplői kriptográfiai módszerek segítségével cserélnek üzeneteket. A PKI az üzenetek kódolásához használt úgynevezett kriptográfiai kulcsok (nyilvános kulcsok és magánkulcsok) gondozásához (generálásához, tárolásához, továbbításához, megsemmisítéséhez) nyújt eszköztárat, amely kriptográfiai algoritmusokból, szoftverekből, hardver eszközökből, szervezetekből, intézményekből, valamint szabványokból, szabályokból és jogszabályokból áll. Az alábbiakban az ezekben közreműködő szereplőket mutatjuk be részletesen.

1.5.1. Végfelhasználók: alany/aláíró és érintett fél

A PKI arra szolgál, hogy a végfelhasználók biztonságosan kommunikálhassanak egymással, így a PKI legfontosabb szereplője a *végfelhasználó*. Azt a végfelhasználót, aki tanúsítvánnyal rendelkezik, a tanúsítvány alanyának (subject) nevezzük, míg azt a végfelhasználót, aki a tanúsítványt ellenőrzi, érintett félnek (relying party) nevezzük. Aláíró tanúsítványok esetén a tanúsítvány alanyát – aki magánkulcsával az aláírást létrehozza – aláírónak (signer, signatory) is nevezzük.

Az érintett fél az a végfelhasználó, aki a tanúsítványban lévő nyilvános kulcsról elfogadja, hogy a tanúsítvány alanyához tartozik, és ennek megfelelően használja a nyilvános kulcsot pl. titkosításra vagy aláírás ellenőrzésére.

Lényeges, hogy *az érintett fél jellemzően nem áll szerződésben a hitelesítés-szolgáltatóval*, így nem vonatkoznak rá a szolgáltatási szerződésben szereplő megszorítások, kikötések. Ha az érintett fél gondosan jár el, akkor a tanúsítvány elfogadása előtt tanulmányozza a tanúsítványra vonatkozó hitelesítési rendet és esetleg a szolgáltató szabályzatait, és követi a bennük szereplő ajánlásokat. Az érintett fél akár figyelmen kívül is hagyhatja a bennük foglaltakat, és bármilyen módon ellenőrizheti a tanúsítványt. Az érintett félre egyedül a jogszabályok vonatkoznak; aláírás ellenőrzésére szolgáló tanúsítványok esetén[2] ezek közül különösen fontos az elektronikus aláírásról szóló törvény.

1.5.2. Képviselt szervezet

Előfordulhat, hogy a tanúsítványban az alany neve mellett egy szervezet is feltüntetésre kerül, így a tanúsítvány alanya a tanúsítvánnyal e szervezetet is „képviseli". Az így képviselt szervezetnek – a tanúsítvány típusától függően – az alany lehet például a vezetője vagy csak egyszerű dolgozója, tagja (pl. egyesületnek), de az is lehet, hogy az alany nem ember, hanem egy domain vagy egy berendezés[3], amely az adott szervezethez tartozik. Az érintett fél látja, hogy a képviselt szervezet neve megjelenik a tanúsítványban, és ez alapján tekintheti úgy, hogy a tanúsítvány is az adott szervezethez tartozik. A képviselt szervezet általában jogosult kérni, hogy a tanúsítványt kibocsátó hitelesítés-szolgáltató tegye érvénytelenné, azaz „vonja vissza" a tanúsítványt.

1.5.3. Előfizető

Gyakori, hogy a tanúsítvány alanya nem az, aki fizet a tanúsítványért; például előfordulhat, hogy a tanúsítványra (vagy hitelesítés-szolgáltatásra) egy cég fizet elő, míg az alany egy természetes személy (a cég egy dolgozója). Ekkor előfordulhat, hogy az előfizető cég neve képviselt szervezetként feltüntetésre kerül a tanúsítványban, de ez nem törvényszerű. Az is sokszor fordul elő, hogy a tanúsítványban szerepel ugyan képviselt szervezet, de ez a szervezet nem ugyanaz, mint az előfizető. Előfizetőnek (subscriber) azt a felet nevezzük, aki a tanúsítvánnyal kapcsolatos díjakat megfizeti a hitelesítés-szolgáltatónak. Így az előfizetőnek vannak bizonyos jogosultságai a tanúsítvánnyal kapcsolatban (pl. jogosult kérni a tanúsítvány visszavonását). Az előfizető neve nem szerepel a tanúsítványban (kivéve, ha az előfizető egybeesik az alannyal vagy a képviselt szervezettel), így az érintett fél nem tudja, hogy ki az előfizető.

[2] A titkosításra és autentikációra szolgáló tanúsítványokra nem vonatkozik az Eat.

[3] Vegyük figyelembe, hogy jogi értelemben vett elektronikus aláírást csak természetes személy vagy jogi személy készíthet. Ha technikailag automata hozza létre az elektronikus aláírást (és például számlákat bocsát ki), akkor az általában az adott természetes vagy jogi személy nevében jár el.

1.5.4. Hitelesítés-szolgáltató és regisztrációs szervezet

Hitelesítés-szolgáltatónak (certification authority, CA, certification service provider) azt a szervezetet nevezzük, aki a tanúsítványt kibocsátotta (aláírta). A hitelesítés-szolgáltató „fenntartja" a tanúsítványt: közzéteszi a visszavonási állapotát és esetleg magát a tanúsítványt is, fogadja a felfüggesztési vagy visszavonási kérelmeket, és megőrzi a tanúsítvány alanyára vonatkozó információkat. A tanúsítvány kibocsátása előtt vagy a hitelesítés-szolgáltató győződik meg az alany kilétéről, vagy egy vele szerződésben álló regisztráló szervezet (registration authority, RA). (Magyarországon nem terjedt el, hogy a hitelesítés-szolgáltatók külön regisztráló szervezeteket használjanak.) A tanúsítványban szereplő adatok helyességéért a hitelesítés-szolgáltató vállal felelősséget (a tanúsítvány típusától függő mértékben) akkor is, ha az alany azonosítását regisztráló szervezet végezte. A hitelesítés-szolgáltatókról a 4. fejezetben szólunk részletesen.

1.5.5. Időbélyegzés-szolgáltató

Az időbélyegzés-szolgáltató olyan ún. időbélyegeket bocsát ki, amelyekkel igazolható, hogy egy adott dokumentum[4] egy adott időpillanatban már létezett. Az időbélyegek egyik legfontosabb felhasználási területe az, amikor elektronikus aláírásokra helyezik el őket, így az időbélyeggel igazolható, hogy az aláírás egy adott időpontban már létezett (és nem később készült). Később megmutatjuk, hogy az elektronikus aláírások biztonsága nagyon nagy mértékben az időbélyegek biztonságára épül. Az időbélyegzésről a 7. fejezetben írunk részletesen.

1.5.6. Az aláírás-létrehozó eszközt biztosító szolgáltató

Az eszköz-szolgáltató feladata az aláírás-létrehozó eszköz – esetleg ún. biztonságos aláírás-létrehozó eszköz – megszemélyesítése, és a magánkulcs elhelyezése az aláírás-létrehozó eszközön. Bár e szolgáltatás külön is nyújtható, Magyarországon általában – különösen minősített tanúsítványok esetén – a hitelesítés-szolgáltatók nyújtják, vagy pedig egyáltalán nem jelenik meg eszköz-szolgáltató, és a magánkulcsot az alany számítógépe tárolja. Az eszköz-szolgáltatókról is a hitelesítés-szolgáltatókról szóló fejezetben (4. fejezet) írunk részletesen.

1.5.7. Archiválás-szolgáltató

A papír alapú aláírásokhoz hasonlóan elektronikus aláírás esetén is fennáll az a probléma, hogy a „régen" készült aláírások hitelességét nem könnyű megbízhatóan ellenőrizni. Ha azt szeretnénk, hogy egy aláírás hosszú távon is ellenőrizhető maradjon, akkor – a papír alapú aláírásokhoz hasonlóan – az elektronikus aláírásokat is speciális körülmények között kell

[4]Pontosabban: egy adott kriptográfiai lenyomatú dokumentum.

archiválni. Ezen archiváláshoz professzionális archiválás-szolgáltatót is igénybe vehetünk, vagy saját magunknak kell ellátnunk az archiválás-szolgáltató feladatait. Az archiválásról a 8. fejezetben írunk részletesen.

1.5.8. Alkalmazás-fejlesztő

Kulcsainkat, tanúsítványainkat mindig valamilyen szoftver vagy alkalmazás segítségével használjuk. Alkalmazás-fejlesztőnek azt nevezzük, akitől (vagy akiktől) a PKI-t használó alkalmazásunkat vásároltuk. Nagyon sok, PKI-t támogató alkalmazás létezik, de a PKI-t nem egy, hanem igen számos szabvány írja le, és a PKI igen sokféleképpen és igen sokféle célra használható. Előfordulhat, hogy más és más módon kell használni attól függően, hogy milyen célra használjuk. Gyakori, hogy hazai (vagy európai) szabványok nincsenek összhangban az amerikai szabványokkal, ebből sok probléma vagy félreértés adódhat. Fontos, hogy az általunk használt alkalmazás a PKI-nek azon részét is támogassa, amire éppen szükségünk van. Például az elektronikus aláírás ellenőrzése (6.5. fejezet) rendkívül összetett művelet, amelyet – az alkalmazott tanúsítványok típusától és az igénybe vett hitelesítés-szolgáltatóktól függően – igen sokféle módon lehet jól végezni.

1.5.9. Szabályozó szervek

Nagyon sok fél vesz részt a PKI-ben; tevékenységüket szabályok hangolják össze. Ide soroljuk a jogalkotókat, a szabványok készítőit, és a felügyeleti szerveket. Magyarországon a Nemzeti Média- és Hírközlési Hatóság felügyeli az elektronikus aláírással kapcsolatos szolgáltatásokat nyújtó szolgáltatókat.

1.5.10. Rendszerintegrátor

A PKI nagyok sok szereplőből állhat, vannak közöttük végfelhasználók, illetve szervezeteik, hitelesítés-szolgáltatók, időbélyegzés-szolgáltatók, archiválás-szolgáltatók, illetve a különféle (hazai, külföldi, tengerentúli) alkalmazások fejlesztői. Ezek érdekei ütközhetnek egymással.

Például egy hitelesítés-szolgáltató könnyen eleget tehet kötelezettségeinek, ha kellően szűkre szabja az általa nyújtott szolgáltatások körét (és például nehézkes nála tanúsítvány visszavonását kérelmezni, és csak ritkán bocsát ki visszavonási listát), és sok esetben kizárhatja a felelősségét is. Az alkalmazás-fejlesztő könnyen fejleszthet olyan alkalmazást, amely a szabványoknak megfelelően ellenőriz aláírást, és az ellenőrzést azonnal el is tudja végezni. Előfordulhat, hogy ilyen esetben – amikor a hitelesítés-szolgáltató is szabványosan és jogszerűen működik, és az aláírás-ellenőrző alkalmazás is szabványos – a végfelhasználókon és a rendszer biztonságos működéséért felelős rendszerintegrátoron csattan az ostor; kiderülhet, hogy habár a rendszer működik, az aláírásokat nem kellő körültekintéssel fogadja el, és minden felelősség a végfelhasználókra nehezedik.

A rendszerintegrátor az, aki e sokféle termékből és szolgáltatásból hasznos, jól működő, értelmes rendszert szeretne létrehozni a felhasználók számára. A rendszer megbízható és biztonságos működéséért felelős rendszerintegrátor nehéz feladat előtt áll, neki erősen ajánljuk e könyvet.

1.5.11. A támadó

> *„Me? I'm dishonest, and a dishonest man you can always trust to be dishonest. Honestly. It's the honest ones you want to watch out for, because you can never predict when they're going to do something incredibly...stupid."*
> *(Én? Becstelen vagyok, és egy becstelen ember mindig megbízhatóan becstelen. Becsszóra. A becsületesekkel kell vigyázni, mert soha nem tudod, mikor fognak valami rendkívül nagy...butaságot csinálni.)*
>
> – *Jack Sparrow kapitány szavai A Karib-tenger kalózai című filmből*

Ha a világ tökéletes lenne, nem lennének rossz szándékú emberek, és senki nem tagadná le az állításait, akkor nem lenne szükség sem PKI-re, sem elektronikus aláírásra, és majdnem minden informatikai biztonsági megoldás[5] feleslegessé válna.

A világ nem ilyen tökéletes. Léteznek támadók, akik rossz szándékkal használják az informatikai rendszereket, egy részük külső fél, de egy részük általában a rendszer jogosult felhasználói közül kerül ki. Az ellenük hozott informatikai biztonsági megoldások általában azt célozzák meg, hogy adott erőforrásokkal rendelkező támadó ne érhessen célt, ne érhessen célt költséghatékonyan, illetve csak elhanyagolható valószínűséggel érhessen célt.

1.6. Összegzés – a PKI dióhéjban

- A nyilvános kulcsú infrastruktúra minden résztvevőjének van két kulcsa:
  - magánkulcs (ezt csak ő ismeri),
  - nyilvános kulcs (ezt bárki megismerheti).

  A nyilvános kulcs alapján (reális erőforrásokkal) nem lehet kiszámítani a magánkulcsot.
- Ha magánkulcsunkkal kódolunk valamit, a nyilvános kulcsunkkal bárki ellenőrizheti, hogy a kódolást mi végeztük el. Ezt nevezzük *aláírásnak*, *hitelesítésnek*.
- Ha egy nyilvános kulccsal kódolunk valamit, azt kizárólag a hozzá tartozó magánkulccsal lehet visszafejteni. Ezt nevezzük *titkosításnak*.

[5] Leszámítva a véletlenül bekövetkező hibák elleni lépéseket.

- Csak akkor támaszkodhatunk egy nyilvános kulcsra, ha tudjuk, hogy ki birtokolja a hozzá tartozó magánkulcsot.
- A *hitelesítés-szolgáltatók* olyan szereplők, akik aláírt igazolásokat állítanak ki arról, hogy egy adott nyilvános kulcs (és a hozzá tartozó magánkulcs) kihez tartozik. Ezen aláírt igazolásokat nevezzük *tanúsítványnak*.
- A tanúsítványokat (illetve a rajtuk lévő aláírásokat) általában más tanúsítványok alapján ellenőrizhetjük, az ellenőrzést *gyökér hitelesítés-szolgáltatók* nyilvános kulcsaira vezethetjük vissza; e kulcsokat sokan ismerik és elfogadják.
- Az *időbélyegzés-szolgáltatók* aláírt igazolásokat bocsátanak ki arról, hogy egy adott dokumentum egy adott időpontban létezett.
- Jogszabály bizonyító erőt rendel
  - a minősített és a fokozott biztonságú elektronikus aláírásokhoz és
  - a minősített időbélyegekhez.

2. fejezet

Kriptográfiai összefoglaló

„Ahol ember van, ott titok van."

– Márai Sándor

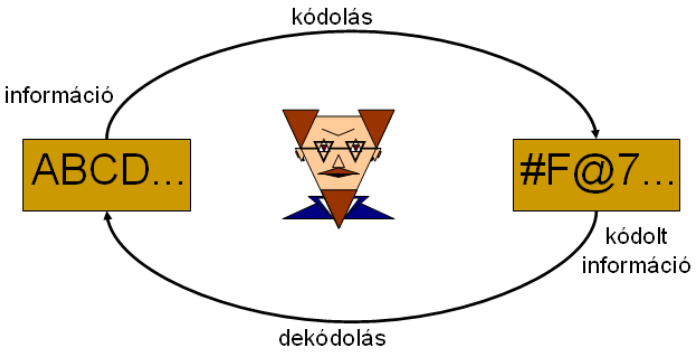
„Two can keep a secret if one is dead."
(Két ember csak akkor tud megőrizni egy titkot, ha egyikőjük már halott.)

– Ismeretlen

„Ignorance is bliss"
(A tudatlanság csodálatos)

– A Mátrix; Cipher szavai

A kriptográfia a titkosítással, rejtjelezéssel és ezen kódok megfejtésével foglalkozó tudományág. Mind a nyilvános kulcsú infrastruktúra, mind az elektronikus aláírás különféle kriptográfiai technológiák biztonságára épül, e fejezetben ezek legfontosabb alapelveit foglaljuk össze. Nem célunk, hogy teljes körű áttekintést nyújtsunk a kriptográfiáról, és szintén nem célozzuk meg az egyes kriptográfiai algoritmusok vagy protokollok részleteinek bemutatását sem. Akiket részletesebben érdekel a kriptográfia világa, azoknak Buttyán Levente és Vajda István „Kriptográfia és alkalmazásai" című könyvét vagy Bruce Schneier „Applied Cryptography" című munkáját javasoljuk. [21], [164] E fejezetben kizárólag azon pontokra összpontosítunk, amelyek a nyilvános kulcsú infrastruktúra és az elektronikus aláírás működésének megértéséhez elengedhetetlenül szükségesek.



2.1. ábra. **Kódolással átalakítjuk az információt**

2.1. Miért kódoljuk az információt?

Ha érzékeny információt nem biztonságos csatornán továbbítunk (vagy nem biztonságos környezetben tárolunk), gondoskodnunk kell az információ védelméről. Különböző módszerek léteznek az információ védelmére: védhetjük fizikailag (például falakkal zárhatjuk el illetéktelen felektől), logikailag (például tűzfalakkal vagy más hálózatbiztonsági eszközökkel), szabályzatilag (megtilthatjuk, hogy bizonyos felek hozzáférjenek) stb. A továbbiakban azzal a lehetőséggel foglalkozunk, hogy az információt *kódolással* védjük meg: olyan módon alakítjuk át, olyan módon változtatjuk meg a szerkezetét, hogy az új szerkezet valamilyen értelemben védelmet nyújtson. Ha a kódolt információt ismét fel akarjuk használni, dekódolnunk kell.

Sok különböző fajta kódolás létezik, több különböző célból dönthetünk úgy, hogy kódoljuk az információt.

2.1.1. Forráskódolás

Forráskódolási módszerek segítségével *tömöríthetjük* az információt, azaz olyan módon kódoljuk, hogy az kevesebb biten is elférjen. Ekkor nem valamilyen veszélyforrással szemben védjük az információt, hanem „kezelhetőbbé” tesszük, hogy kisebb helyen férjen el, vagy adott sávszélességű csatornán gyorsabban lehessen továbbítani. A forráskódolás célja, hogy csökkentsük vagy megszüntessük az információban a redundanciát (így például a tömörített információban ne legyenek többször előforduló, azonos blokkok).

A forráskódolási módszerek között elkülöníthetünk veszteségmentes módszereket (amelyek esetén a kódolt információból az eredeti információ pontosan visszaállítható), és adatvesztéses módszereket (amelyek esetén a kódolt információból az eredeti információ nem állítható ugyan vissza pontosan, de valamilyen értelemben „elég jól” visszaállítható). Veszteségmentes tömörítési módszer például a Huffman kódolás és az LZW, adatvesztéses tömörítési módszert használnak például a JPG képek és az MP3 fájlok.

2.1. Példa: *Ha a következő szöveg helyett:*
`AAAAAAAAAAAAAAAAAAAAAAAAAAABB`
Ezt írjuk:
`27db A és 2db B`
akkor tömörítettünk. Továbbra is ugyanazt jelenti, csak rövidebben fogalmaztuk meg azzal, hogy a szabályszerűséget próbáltuk leírni.

2.1.2. Csatornakódolás

Csatornakódolás segítségével *véletlen hibákkal szemben védhetjük* az információt. Itt megkülönböztethetünk hibadetektáló kódolást (amelynek segítségével – bizonyos keretek közt – észlelhető, ha a kódolt információ bitjei megváltoztak) és hibajavító kódolást (amely segítségével nemcsak a hiba ténye deríthető ki, hanem – bizonyos keretek közt – a hiba ki is javítható).

Míg ha forráskódolást alkalmazunk, az információ „kisebb", tömörebb lesz, a csatornakódolás általában éppenséggel megnöveli az információ méretét. A csatornakódolás általában speciális redundanciát visz az információba, amelynek segítségével ellenőrizhető, hogy bizonyos összefüggések fennállnak-e a kódolt információn belül.

Nagyon egyszerű példa erre, ha az információt „megduplázzuk", azaz minden bitet kétszer írunk le. Ekkor az információ kétszer akkora lesz, mint eredetileg, de azonnal észlelhető, ha valamely egy bitje megváltozik. (Ez egyébként pazarlás, egy hiba detektálásához elegendő egyetlen, ún. paritásbitet fűznünk az információhoz. [77])

2.2. Példa: *Ha azt mondjuk: „2010. szeptember 28-a, kedd", az egy hétköznapi hibadetektáló kód. Egy naptárral könnyen ellenőrizhető, hogy a 2010. szeptember 28-a keddre esik-e, és ha mégsem, kérhetünk pontosítást. Redundanciát fűztünk a dátumhoz, amely bizonyos tévedések felismerésére alkalmas.*

Nagyon sok protokoll fűz CRC (cycle redundancy check) kódot a továbbított adatokhoz, amelynek segítségével az adatokat befogadó fél kimutathatja, ha a kapott adatok a továbbítás során meghibásodtak. (A különböző CRC-k különböző számú hiba kimutatására alkalmasak.) A ma használt merevlemezek beépített hibajavító kódolást is tartalmaznak, olyan módon tárolják az adatokat, hogy akkor is vissza tudják állítani az adatokat, ha a lemez egy-egy bitje meghibásodik. (Bizonyos számú vagy típusú hibával viszont már ők sem boldogulnak.)

2.1.3. Kriptográfiai kódolások

A kriptográfia a titkosítással, rejtjelezéssel és ezen kódok megfejtésével foglalkozó tudományág, *kriptográfiai kódolások* segítségével *szándékos támadások ellen védhetjük* az

információt. A támadás vagy az információ megszerzésére, megismerésére, vagy az információ észrevétlen módosítására irányulhat. Az előbbi ellen véd a *titkosítás*, az utóbbi ellen a *hitelesítés*.

Ha az információt *titkosítjuk*, akkor olyan módon kódoljuk, hogy a kódolt információból illetéktelen fél ne következtethessen sem az eredeti információra, sem annak egy részére. A titkosított információban nem jelenhetnek meg az eredeti, nyílt információ statisztikai tulajdonságai (szabályosságai), mert ezek fogódzót jelenthetnének a támadó számára. Így, ha a nyílt információban voltak is redundáns elemek, a titkosított információban ezek várhatóan nem jelennek meg.

Ha az információt *hitelesítjük*, akkor a kódolás során olyan redundáns elemeket helyezünk el benne, amelyek az információ minden egyes bitjétől függenek, és amelyeket illetéktelen fél nem tud létrehozni. A hitelesítés egyik célja, hogy a hitelesített információt illetéktelen fél ne módosíthassa észrevétlenül, másik célja, hogy bizonyítható legyen, hogy a hitelesített információt nem illetéktelen fél hozta létre.

Lényeges, hogy míg a csatornakódolás célja véletlen hibák észlelése vagy javítása (azaz olyan hibák ellen nyújt védelmet, amelyek az információ egyes bitjeit véletlenszerűen érintik), addig a kriptográfiai hitelesítés a szándékos módosítás ellen véd (jellemzően a módosítás tényének megállapítására alkalmas). Ez esetben a támadó intelligens, tudatosan azokat a pontokat támadja, amelyeken a legnagyobb valószínűséggel számíthat a sikerre. Mindkét technológia alkalmas lehet hibák tényének felderítésére, de minőségileg más célt szolgálnak, és más eszköztárral dolgoznak. A kriptográfiai hitelesítés alkalmas lehet ugyan véletlen hibák tényének kimutatására, de véletlen hibák felderítésére sokkal hatékonyabbak a csatornakódolási módszerek, és azok képesek lehetnek a hiba helyének felderítésére, vagy akár a hiba javítására is. Ezzel szemben, a szándékos támadás ellen a csatornakódolás egyáltalán nem nyújt védelmet.

Gyakori, hogy a különféle kódolásokat valamilyen módon kombinálva használjuk. Az információt először mindig tömöríteni célszerű (ekkor kisebb lesz, és a további kódolásokat kevesebb biten kell elvégeznünk). Ezt szokták követni a kriptográfiai kódolások – jellemzően előbb a hitelesítés (például az elektronikus aláírás), és ezt követően a titkosítás. A csatornakódolást (hibajavító vagy hibadetektáló kódolást) célszerű utoljára alkalmazni.

E sorrendtől nem szerencsés eltérni. Például ha először titkosítunk, azzal elfedjük az eredeti információ szabályosságait (eltűnik a redundancia), és a titkosított információt már hiába próbáljuk tömöríteni, az nem lesz jelentősen kisebb. Hasonlóan rossz ötlet, ha először alkalmazunk hibajavító kódolást, és utána titkosítunk: egyrészt, a titkosítás elfedi a hibajavító kódolás által biztosított redundanciát, másrészt, a legtöbb titkosító algoritmus esetén egyetlen bit változás a titkosított információban teljesen más nyílt információhoz vezet.

E fejezet további részében a kriptográfiai kódolásokról lesz szó, a forráskódolással és a csatornakódolással kapcsolatban az „Információ és kódelmélet” című könyvet javasoljuk. [77]

2.2. A kulcs fogalma

A kriptográfiai kódolások segítségével illetéktelen személyek támadásaival szemben védhetjük információink bizalmasságát vagy hitelességét. Ezen megoldás biztonsága arra épül, hogy a kódolást (illetve titkosítás esetén a dekódolást) kizárólag jogosult felek tudják elvégezni, illetéktelen fél nem képes rá.

Elsőre az tűnne kézenfekvő megoldásnak, ha a kódolási módszert – azaz a kriptográfiai algoritmust – tartanánk titokban: kizárólag jogosult felek tudhassák, hogy hogyan van kódolva az információ; ha illetéktelen fél nem tudhatja meg, hogy hogyan kódoltuk az információt, nincs esélye.

E megoldásnak súlyos hibái vannak. Magát a kódoló algoritmust nagyon nehéz titokban tartani: minél többen használják, annál könnyebben előfordulhat, hogy a kódoló (berendezés vagy algoritmus) illetéktelen kezekbe kerül, és nagyon nehéz megszervezni annak biztonságos lecserélését. Tovább súlyosbítja a helyzetet, hogy viszonylag kevés „biztonságos" kódoló algoritmust ismerünk, nem lehet mindig újat kitalálni, ha az információt más célból, más módon vagy más számára szeretnénk kódolni.

Először Auguste Kerckhoffs[1] mondta ki a fenti gondolatokat, és azt javasolta, hogy a kódoló algoritmusoknak legyen egy olyan paramétere, amelyet könnyen és gyorsan lehet cserélni, nevezzük ezt *kulcsnak*, és a kódolás akkor is maradjon biztonságos, ha a kódoló/dekódoló algoritmus minden egyes részlete – a kulcsot kivéve – nyilvános, így a támadó által is ismert. [93] Ma általánosan elfogadott, hogy a kriptográfiai algoritmusoknak ezen szigorú feltétel – az ún. *Kerckhoffs feltétel* (Kerckhoffs' principle) – mellett is biztonságosnak kell lennie. A támadó megismerheti az algoritmus minden egyes részletét, ha a kulcsot nem ismeri, az algoritmus biztonságos kell, hogy maradjon; azaz *a rendszer biztonsága kizárólag a kulcs titkosságára kell, hogy épüljön.* Minden biztonsági rendszer titkokra épül. Minden titok kiszivárgása potenciális hibaforrás. Kerckhoffs szerint a titok legyen kicsi, rövid és könnyen cserélhető.

Kerckhoffs gondolatmenete szerint biztosítanunk kell, hogy a jogosult felek ismerjék a kulcsot, és illetéktelen felek ne ismerhessék a kulcsot. Ekkor – ha az algoritmusunk kellően biztonságos – megoldottuk, hogy kizárólag jogosult felek tudják elvégezni a megfelelő kódolást/dekódolást.

A kulcs a kriptográfiai algoritmus egy paramétere, tehát maga is *információ, adat.* Így tekinthetjük például számnak vagy bitsorozatnak. A kulcsot mindig valamilyen kulcstérből – a lehetséges kulcsok halmazából – választjuk.

A Kerckhoffs feltétel szerint a támadó a kulcsteret is ismeri, így tudja, hogy melyek a lehetséges kulcsok. Bármilyen módon kódolunk, és bármilyen módon választjuk a kulcsot, a támadó megteheti, hogy végigpróbálja az összes lehetséges kulcsot; ezt a megoldást nevezzük *kimerítő*

[1] Auguste Kerckhoffs (teljes nevén Jean-Guillaume-Hubert-Victor-François-Alexandre-Auguste Kerckhoffs von Nieuwenhof, amelyet később – érthető okokból – lerövidített) a XIX. században élt holland nyelvész, kriptográfus.

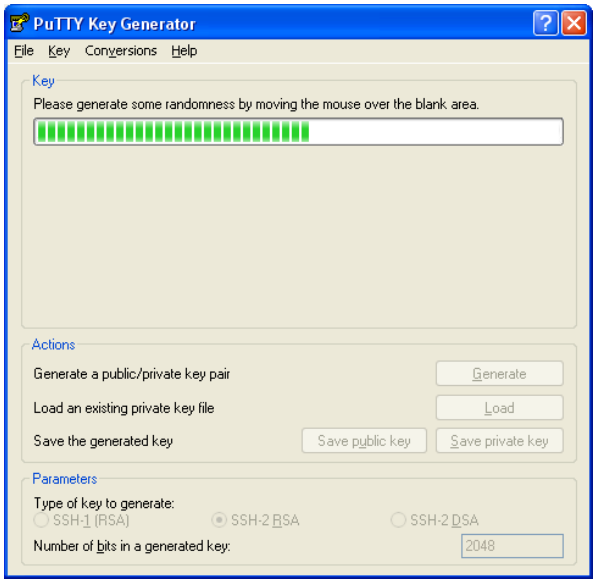
keresésnek vagy *a nyers erő módszerének* (angolul: brute force attack). Ez ellen az nyújt védelmet, ha a kulcstér kellően nagy. Nem szabad, hogy a támadó – a rendelkezésére álló erőforrások birtokában – akár a teljes kulcsteret, akár annak egy „jelentős" részét (amely elég nagy ahhoz, hogy a támadó jelentős valószínűséggel rábukkanhasson a kulcsunkra) végigkereshesse. *A kulcstér mérete – így a kulcs hossza – alapvetően meghatározza a kódolás biztonságát.*

A ma biztonságosnak ítélt algoritmusok legalább 112 bit vagy 128 bit hosszú kulcsokkal dolgoznak, így a kulcs kimerítő kereséssel történő megkereséséhez 2^{112} vagy 2^{128} kulcs végigpróbálása volna szükséges, amely a ma (vagy a közeljövőben várhatóan) rendelkezésre álló eszközökkel elképzelhetetlen. A mai eszközökkel az ilyen méretű kulcstereknek csak egy töredéke kereshető végig, így elenyészően kicsi az esély egy kulcs sikeres megtalálására.

A Kerckhoffs feltétel szerint a támadó nemcsak a kulcsteret ismeri, hanem még azt a módszert is ismerheti, hogy milyen módon választjuk ki a kulcsot a kulcstérből. Éppen ezért, a kulcsot véletlenszerűen kell kiválasztanunk, méghozzá úgy, hogy minden egyes kulcs kiválasztásának pontosan ugyanakkora esélye legyen. Ha valamely kulcsokat nagyobb eséllyel választanánk, akkor a támadónak is könnyebb lenne a dolga: kereséskor az „esélyesebb" kulcsokat próbálná ki először, és így várhatóan hamarább találná meg a kulcsunkat. [109], [64]

A kulcs kiválasztásához, azaz a *kulcsgeneráláshoz* valamilyen véletlen forrást, véletlenszám-generátort kell használnunk. *A kulcsgeneráláshoz használt véletlen forrás minősége alapvetően meghatározza a kódolás biztonságát.* Ha a véletlen forrás „rossz" minőségű, a támadó esetleg a véletlen próbálkozásnál jobb eséllyel tud következtetni a kulcsra, bizonyos kulcsokat esélyesebbnek tekinthet, így a kimerítő keresésnél ügyesebb módszert is találhat kulcsunk megkeresésére. Ha „rossz" minőségű véletlen forrásból származó kulcsot használunk, hiába választunk erős kriptográfiai algoritmusokat, azok nem nyújtanak védelmet, ha a támadó ki tudja találni a kulcsunkat.

A legtöbb számítógép csak ún. *álvéletlen* (pseudorandom) számokat tud generálni, valamilyen álvéletlenszám-generátor segítségével. Az álvéletlenszám-generátor matematikai algoritmus alapján állít elő számokat a bemenete vagy belső állapota, az ún. „random seed" alapján. Ha az álvéletlenszám-generátor jó minőségű, akkor a belőle kinyert álvéletlen számok között – a random seed ismerete nélkül – nehéz bármilyen összefüggést kimutatni, és nehéz őket megkülönböztetni a valódi véletlen számoktól. Ugyanakkor ha a támadó ismeri (vagy meg tudja becsülni) az általunk használt random seed értékét, pontosan meg tudja mondani, hogy milyen véletlen számot generáltunk. Például ha egy álvéletlenszám-generátor kizárólag a számítógép órájából képzi a random seedet, akkor ha a támadó tudja (vagy sejti), hogy mikor generáltuk a kulcsot, következtethet a random seed értékére és akár a kulcsunkra is. Léteznek jobb minőségű álvéletlenszám-generátorok is, amelyek más adatokat is felhasználnak az álvéletlen szám képzésére, de általánosságban is kimondhatjuk, hogy *tisztán algoritmikus módon nem lehet valódi véletlenszámot generálni.* (Ilyen értelemben egy pénzfeldobással vagy

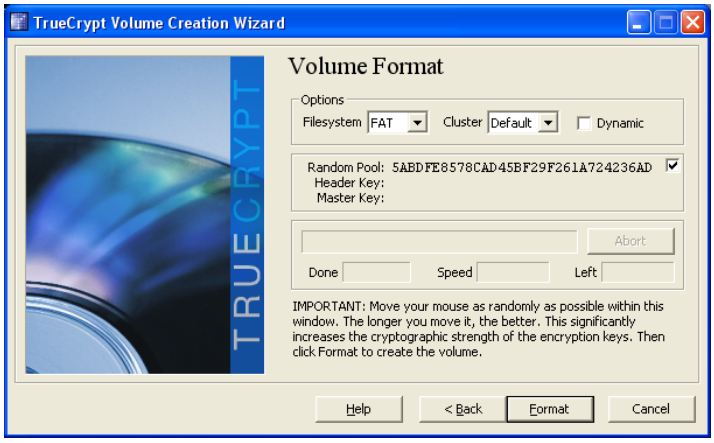


2.2. ábra. **Kulcsgenerálás a PuTTY nevű program segítségével**

dobókockával generált véletlenszám sokkal jobb minőségűnek tekinthető minden algoritmus által generált véletlenszámnál.)

A valódi véletlenszám generálására alkalmas véletlenszám-generátorok valamilyen fizikai folyamatot is figyelnek, és ezekből nyert, lehetőleg a támadó számára megbecsülhetetlen értékeket is felhasználnak a véletlenszám generálására. Legegyszerűbb megoldás, amikor a véletlenszám-generátor a felhasználótól vár inputot: megkéri a felhasználót, hogy mozgassa az egeret (lásd: 2.2. ábra), vagy gépeljen véletlenszerűen. Ekkor a véletlenszám-generátor nemcsak azt figyeli, hogy mit gépel a felhasználó (ez feltehetően nem lenne jó véletlen forrás), hanem például az egyes billentyűleütések közötti apró idő-eltéréseket is felhasználja. (Létezik olyan véletlenszám-generátor is, amelyik a számítógép hálózati kártyáján jelentkező forgalmat használja fel véletlenszámok előállítására. Habár a hálózati kártya forgalmát az ugyanazon lokális hálózaton lévő támadó is láthatja, ő már kicsit más időzítéssel látja a jeleket, és esetleg nem tudja megbecsülni, hogy egy adott gépre pontosan mikor érkeztek meg. [4])

A felhasználótól bekért véletlen inputnak több hátulütője is van. Az egyik legnagyobb, hogy a felhasználótól viszonylag sok inputot kell bekérni, és ebből viszonylag kevés megfelelő minőségű véletlen bit nyerhető ki. (Egy átlagos számítógép felhasználó is sok olyan folyamatot indít el, amelyekhez véletlenszámokra van szükség, és a felhasználótól nagyon nehéz lehet megfelelő mennyiségű véletlent begyűjteni.) A professzionális véletlenszám-generátorok fizikai folyamatokat (pl. termikus zaj) is figyelnek, ezekből nyerik ki a véletlenszámokat. Ma már léteznek olyan számítógépek, amelyek rendelkeznek beépített, valódi véletlenszám generálására alkalmas eszközökkel, de vannak nagyon drága, nagy mennyiségű és jó minőségű véletlen előállítására alkalmas speciális bevizsgált célhardverek, hardveres véletlenszám-



2.3. ábra. **Kulcsgenerálás a Truecrypt nevű program segítségével**

generátorok is. (Mint ahogy azt a későbbiekben látni fogjuk, a hitelesítés-szolgáltatóknak bizonyos esetekben ilyet kell használniuk.)

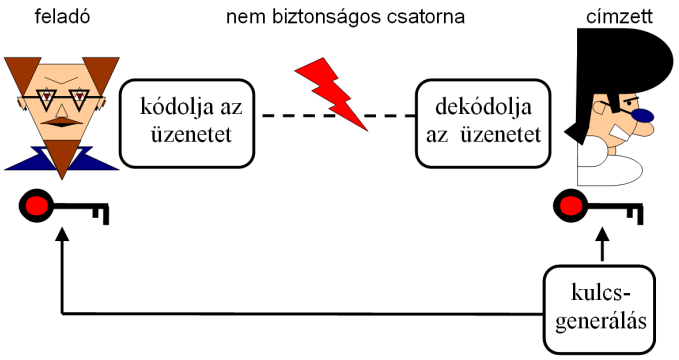
A kulcsgenerálás – a kulcs kulcstérből való kiválasztása – jelenti a kulcs életciklusának első lépését. Ezt követően a kulcsot biztonságos módon – titkosan és/vagy hitelesen – el kell juttatnunk a megfelelő helyre. Ha a kulcs célba ért, ott a kulcsot tárolnunk, majd használnunk kell. Végül, ha a kulcsra már nincsen szükség, a kulcsot meg kell semmisítenünk. A kulcsok generálásának, továbbításának (kiosztásának), tárolásának, használatának és megsemmisítésének folyamatát együttesen *kulcsgondozásnak* vagy *kulcsmenedzsmentnek* nevezzük.

Ha – például tárolás vagy továbbítás során – nem megfelelően gondoskodunk a kulcs bizalmasságáról, előfordulhat, hogy illetéktelen fél (például a támadó) is megismeri a kulcsot. Ez sok esetben a biztonság súlyos sérülését (akár elvesztését) jelenti. (Léteznek ún. nyilvános kulcsú rendszerek, ahol egyes kulcsokat nem kell titokban tartanunk, de arra ott is vigyáznunk kell, hogy a támadó ne változtathassa meg a kulcsot, ne csempészhesse be a saját kulcsát valaki más kulcsának a helyére.) Ha egy kulcsot illetéktelen fél is megismert (vagy felmerült annak a gyanúja, hogy illetéktelen fél is megismerte), azt a kulcs *kompromittálódásának* nevezzük.

2.3. Szimmetrikus kulcsú és nyilvános kulcsú kriptográfia

A rendelkezésünkre álló kriptográfiai megoldások két nagy csoportra oszthatók: szimmetrikus kulcsú kriptográfiai megoldásokra, valamint aszimmetrikus kulcsú, más néven nyilvános kulcsú kriptográfiai megoldásokra. Mindkét csoport alkalmas mind titkosításra, mind hitelesítésre.

A szimmetrikus kulcsú titkosítás során a dokumentumot ugyanazzal a kulccsal kódoljuk, amivel majd dekódolni is lehet; a kódolásra és dekódolásra használt kulcsot ekkor titokban kell tartanunk, ezért *titkos kulcsnak* is nevezzük. E módszer hátránya, hogy a titkos kulcsot



2.4. ábra. **Szimmetrikus kulcsú kriptográfia esetén kódoláshoz és dekódoláshoz ugyanazt a kulcsot használjuk. Így – függetlenül attól, hogy a kulcsot hol generáljuk – a kulcs biztonságos, titkos csatornán kell, hogy eljusson mind a kódolóhoz, mind a dekódolóhoz.**

biztonságos módon kell eljuttatni a fogadó fél számára, hogy illetéktelen fél ne ismerhesse meg. (Lásd: 2.4. ábra.) Ez bizonyos esetekben nagyon nehéz problémát jelenthet.

A szimmetrikus kulcsú kriptográfia hitelesítésre is alkalmas, ekkor ún. *kriptográfiai ellenőrző összeg* (message authentication code, MAC) számítható egy üzenetből a titkos kulcs alapján. E megoldásnak korlátja, hogy ugyanazon titkos kulcs szükséges az ellenőrző összeg ellenőrzéséhez, mint amivel ki lehet azt számítani. Így e megoldás nem alkalmas arra, hogy egy dokumentum hitelességét a közös titkos kulcsot nem ismerő harmadik fél számára igazoljuk. Ha mégis erre van szükségünk, nyilvános kulcsú kriptográfiát – például elektronikus aláírást – kell használnunk.

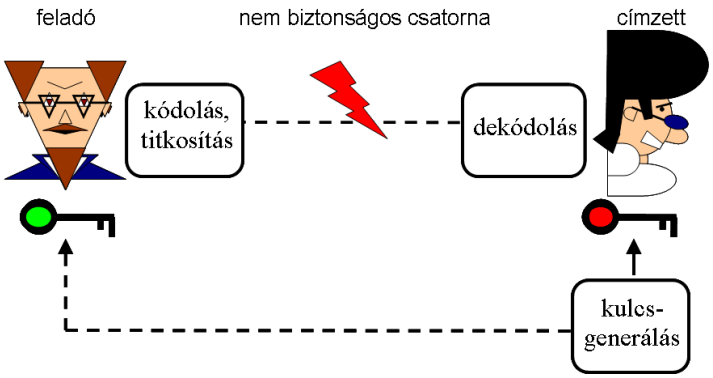
A szimmetrikus kulcsú titkosítás előnye, hogy az ilyen megoldások gyorsak, rövid kulcsokkal (pl. 128 bit vagy 256 bit) dolgoznak, és elég sok szimmetrikus kulcsú algoritmust ismerünk. Szimmetrikus kulcsú titkosító algoritmusok például a következők: DES (már nem használatos[2]), 3DES[3], AES (más néven Rijndael), Blowfish, stb.

Az aszimmetrikus kulcsú (más néven nyilvános kulcsú) kriptográfia a fenti hátrányt küszöböli ki úgy, hogy a kódolás és a dekódolás nem ugyanazzal a kulccsal történik. (Lásd: 2.5. ábra.) Minden szereplőnek van egy nyilvános kulcsa (public key) és egy magánkulcsa[4] (private key). A magánkulcs soha nem kerül ki birtokosa tulajdonából, de bárki hozzáférhet mások nyilvános

[2]A DES 56 bites kulcsokkal dolgozik. A 70-es években, amikor kifejlesztették, ez a kulcsméret még biztonságosnak számított, de a mai számítástechnikai eszközökkel az 56 bites kulcs már törhető (az összes lehetséges 56 bit hosszú kulcs végigpróbálása már nem igényel irreálisan nagy erőforrást). Ez a kulcsméret ma már rövidnek minősül, így akár kimerítő kereséssel is végigpróbálható. A DES ellen ma már létezik ennél hatékonyabb támadás is. [106]

[3]A 3DES (avagy triple-DES) esetén a DES algoritmust háromszor egymás után alkalmazzák, vagy kódoló-dekódoló-kódoló, vagy dekódoló-kódoló-dekódoló felállásban. Ha két különböző DES kulcsot használnak, akkor a kulcsméret 112 bit, ha mindhárom kulcs különbözik, akkor 168 bit. Ha mindhárom kulcsot azonosra választjuk, a sima DES-t kapjuk vissza. A 3DES ma kellően biztonságos, de lassúnak számít.

[4]A magánkulcsot titkos kulcsnak is szokás nevezni. Mi következetesen a magánkulcs elnevezést használjuk, a titkos kulcs kifejezést a szimmetrikus kulcsra tartjuk fent.



2.5. ábra. **Nyilvános kulcsú (más néven aszimmetrikus kulcsú) kriptográfia esetén a kódolás és a dekódolás különböző kulcsokkal történik. Ekkor elegendő az egyik kulcsot titokban tartanunk, a másik kulcsot akár nyilvános csatornán is továbbíthatjuk.**

kulcsához. A nyilvános kulcsot nem kell titokban tartani, azt bárki megismerheti. [32]

Ha titkosított üzenetet szeretnénk küldeni valakinek, meg kell szereznünk az ő nyilvános kulcsát, és azzal kell kódolnunk a neki szóló üzeneteket. Az így kódolt üzeneteket a címzett a saját magánkulcsával fejtheti vissza.

A nyilvános kulcsú kriptográfia más módon is használható: ha a saját magánkulcsunkkal kódolunk egy dokumentumot, az így kapott bitsorozatról – a nyilvános kulcsunk alapján – bárki megállapíthatja, hogy azt mi hoztuk létre. E műveletet aláírásnak nevezzük.

A nyilvános kulcs és a magánkulcs között matematikai összefüggés van, a nyilvános kulccsal kódolt adat a hozzá tartozó magánkulccsal dekódolható, valamint a magánkulccsal kódolt adat is dekódolható (azaz az aláírás ellenőrizhető) a nyilvános párjával. Ugyanakkor nem szabad, hogy a nyilvános kulcsból (hatékonyan, reális mennyiségű erőforrással) ki lehessen számítani a magánkulcsot.

A nyilvános kulcsú kriptográfiai algoritmusok előnye, hogy segítségükkel úgy is kommunikálhatunk valakivel biztonságosan (titkosan vagy hitelesen), hogy előtte nem állapodtunk meg közös titkos kulcsban. Ugyanakkor alapvető követelmény, hogy hitelesen jussunk hozzá a másik fél nyilvános kulcsához. Egy nyilvános kulcs használata előtt meg kell bizonyosodnunk róla, hogy a hozzá tartozó magánkulcs valóban annak a személynek (szervezetnek) a birtokában van, akinek titkos üzenetet szeretnénk küldeni, vagy akinek az aláírását ellenőrizni szeretnénk. E célra tanúsítványokat szokás használni (3. fejezet). A tanúsítvány egy megbízható szervezet, egy hitelesítés-szolgáltató (4. fejezet) által kiállított igazolás arról, hogy egy adott nyilvános kulcs egy adott személyhez vagy entitáshoz (alanyhoz) tartozik (azaz csak ő birtokolja a hozzá tartozó magánkulcsot).

A nyilvános kulcsú kriptográfiai algoritmusok hátránya, hogy jelentősen lassabbak a szimmetrikus kulcsú kriptográfiai algoritmusoknál, és lényegesen hosszabb kulcsokat

használnak. Például az RSA algoritmus esetén ma a 2048 bit hosszú kulcs hosszú távon is biztonságosnak minősül, míg pl. az AES szimmetrikus kulcsú algoritmus esetén a 256 bites kulcs hasonló szintű biztonságot jelent. (Megjegyezzük, két kriptográfiai algoritmus nyújtotta biztonság nagyon nehezen hasonlítható össze.)

További probléma, hogy viszonylag kevés nyilvános kulcsú kriptográfiai algoritmust ismerünk. A legelterjedtebb nyilvános kulcsú algoritmus az RSA, de léteznek más, kevésbé ismert nyilvános kulcsú rendszerek is. Ilyenek például a diszkrét logaritmus problémára épülő algoritmusok (ElGamal, DSA) és ezek elliptikus görbéken értelmezett változatai (EC-ElGamal, ECDSA), és ilyen például az NTRU is. [156], [95], [91], [78]

2.4. Lenyomatképző függvények

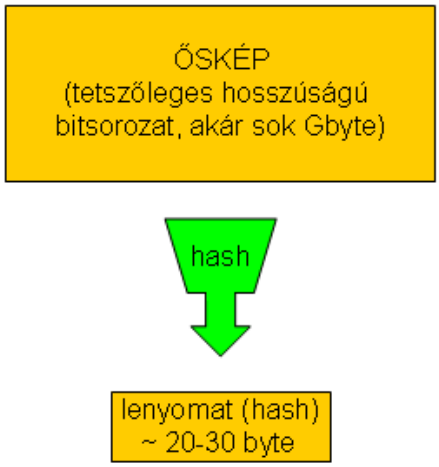
A lenyomatképző függvények (hash függvények) tetszőleges hosszúságú üzenetet fix hosszúságú bitsorozatba képeznek le. Az így kapott eredményt *hash* értéknek vagy lenyomatnak is nevezik. (Lásd: 2.6. ábra.) Mivel a bemenet hossza nagyobb, mint a kimenet, vagyis a lenyomat hossza, így elvileg nem kizárt, hogy két különböző üzenet hash értéke megegyezik. Két alapvető követelményt szokás támasztani a hash függvényekkel szemben:

- Őskép-ellenállóság (preimage resistance): Egy hash érték alapján nagyon nehéz legyen következtetni a hash függvény bemenetére.
- Ütközés-ellenállóság (collision resistance): Legyen nagyon nehéz (szinte lehetetlen) két olyan különböző üzenetet találni, amelyeknek azonos a lenyomata.

  Megjegyzés: Könnyebb két olyan ősképet találni, amelyekhez azonos lenyomat tartozik, ha mindkét ősképet magunk választhatjuk meg. Nehezítjük a feladatot, ha az egyik ősképet lerözítjük, és egy adott őskép-lenyomat párhoz keresünk olyan második ősképet, amelyhez ugyanaz a lenyomat tartozik. Ha egy hash függvény ez nehéz feladat, akkor a hash függvény teljesíti a második őskép-ellenállóság (second preimage resistance) követelményt. Ha egy hash függvény ütközés-ellenálló, akkor egyben második őskép-ellenálló is.

A gyakorlatban a legelterjedtebb hash függvények az SHA-1 és az SHA-2, bár sokat használják a már nem biztonságos MD5 függvényt is. Az MD5 128 bites, a SHA-1 160 bites hash értéket állít elő, viszont mindkettő 512 bites blokkokban dolgozza fel az üzeneteket. Az SHA-2 több függvény, az SHA-224, SHA-256, SHA-384 és SHA-512 összefoglaló neve, e függvények a nevükben is jelzett hosszúságú lenyomattal dolgoznak.

Az MD5 már nem biztonságos, mert rá már nem teljesül az ütközés-ellenállóság: Már létezik hatékony algoritmus, amellyel értelmes, azonos MD5 lenyomattal rendelkező fájlokat



2.6. ábra. **A hash függvény tetszőleges hosszúságú ősképet fix hosszúságú lenyomatba képez le**

találhatunk. Sőt már olyan, MD5-re épülő tanúsítványokat is sikerült létrehozni, amelyek azonos MD5 lenyomatra épülnek, így azonos aláírás szerepel bennük. Így, ahol ütközés-ellenállóságra van szükség, ott *már nem szabad MD5 lenyomatokat használni.* [173]

A közelmúltban ütköző SHA-1 lenyomatokat is találtak, és már van olyan – az elvi korlátnál hatékonyabb – algoritmus, amellyel további ütköző lenyomatú fájlokat lehet készíteni. Ennek egyelőre csak elvi jelentősége van, de várható, hogy a közeljövőben az SHA-1-et (egy vagy több) másik hash függvény váltja majd fel. SHA-1 helyett egyelőre a hasonló elvekre épülő, de hosszabb lenyomatokkal működő SHA-2 algoritmus-családot célszerű használni. Napjainkban folyik az SHA-3 hash függvény kiválasztása. [126]

2.5. Mennyire biztonságos?

2.5.1. Gyakorlati biztonság, avagy feltételes biztonság

Kimondhatjuk, bármilyen kódolási megoldásra is támaszkodunk, az nem lehet biztonságosabb, mint a kulcskezelési módszer, amit használunk. Például ha „gyenge” (könnyen kitalálható) kulcsot választunk, a támadó hozzájuthat a kulcshoz, és akkor is sikerrel támadja a rendszerünket, ha a kódoláshoz „erős” kriptográfiai algoritmusokat veszünk igénybe. Ha „erős” kulcsot választunk, akkor is megfelelően biztonságos módon kell tárolnunk és továbbítanunk, különben ez válik a rendszer leggyengébb láncszemévé.

A gyakorlatban használt algoritmusok esetén a kulcs kimerítő kereséssel megtalálható. Ezért a kulcsot kellően nagy kulcstérből, egyenletes eloszlás szerint kell kiválasztanunk. A kulcstér elég nagy kell legyen ahhoz, hogy a támadó – a tudomány és a technológia jelen állása szerint – csak irreális mennyiségű erőforrás birtokában nézhesse át a kulcsteret (a teljes kulcsteret,

illetve annak egy „kellően nagy” részét), hogy elhanyagolható legyen annak a valószínűsége, hogy véletlenül a megfelelő kulcsra bukkan. A nagy kulcstér sok lehetséges kulcsot jelent, ez pedig úgy valósítható meg, hogy hosszú kulcsokat kell használni. A kulcs elég hosszú kell, hogy legyen ahhoz, hogy megfelelő szintű biztonságot nyújtson, de a túl hosszú kulcsok csak feleslegesen terhelik a rendszereket. Lenyomatképző (hash) függvények esetén kulcsról nem beszélhetünk, ott a lenyomat mérete jelent hasonló felső korlátot arra, hogy az adott hash függvény sikeres megtámadásához (pl. két eltérő őskép kereséséhez, amelyekhez azonos lenyomat tartozik) mennyi erőforrás szükséges.

Tegyük fel, hogy a kulcstér elég nagy! Mennyire nehéz „feltörni” egy adott algoritmust? E kérdés megválaszolásához először azt kell tisztázni, hogy mit értünk a „feltörés” kifejezésen. A támadó célja nem feltétlenül a használt kulcs kitalálása. A támadó akkor is sikert érhet el, ha például a kulcs ismerete nélkül fejt vissza üzeneteket vagy hamisít aláírást. Az algoritmusok „feltörésére” vagy sikeres megtámadására a szakirodalomban számos definíció létezik, például Goldreich ad a témában precíz matematikai modellt. Goldreich modelljében már az is megtörésnek minősül, ha a támadónak a véletlen próbálkozásnál (vagy kimerítő keresésnél) akár egy kicsit is jobb ötlete lehet. [70]

Itt nem célunk ezen modellek részletes ismertetése. Mindössze arra kívánjuk felhívni a figyelmet, hogy egy algoritmus sikeres megtámadása vagy „feltörése” több dolgot is jelenthet. Gondolhatunk gyakorlatban is kihasználható támadásra, ennek szélsőséges esete, ha az Internetről le lehet tölteni egy olyan programot, amely 1-2 másodperc alatt fejti vissza kulcs nélkül a titkosított üzeneteinket. Kevésbé szélsőséges eset, ha a támadás a tudomány és a technológia jelen állása szerint hónapokat vagy éveket igényel egy sok erőforrással rendelkező támadótól is. (Itt vegyük figyelembe, hogy a tudomány és a technológia várhatóan fejlődni fog, így a jövőben a támadás kevesebb erőforrással, rövidebb idő alatt is végrehajtható lesz.) A kriptográfus számára már az is sikeres támadást jelenthet, ha bármilyen „fogást találnak” egy algoritmuson, például ha egy titkosított üzenet alapján akár a dekódoló kulcs, akár a nyílt üzenet egyetlen bitjét is meg lehet határozni 50 százaléknál jobb eséllyel.

Gyakran hangzik el a kérdés, hogy például egy nagyhatalom titkosszolgálata fel tudja-e törni az általunk használt kódolást. Erre egyrészt az a válasz, hogy a tudomány mai állása szerint valószínűleg nem, bár a titkosszolgálatok kriptográfiai ismereteiről nagyon-nagyon keveset tudunk. Másrészt, valószínűleg nincs jelentősége, hogy magát a kódolást fel tudja-e törni, hiszen a magánszemélyek vagy kis szervezetek által használt kulcskezelési megoldásokat szinte biztosan meg lehet támadni, és ezek megtámadása szinte biztos, hogy sokkal olcsóbb, mint magának a kódolásnak a támadása.

Ha publikált, széles körben elterjedt algoritmusokra támaszkodunk, amelyeknek nincsenek ismert gyengeségei, és algoritmusainkat kellően hosszú és megfelelő módszerrel generált kulcsokkal használjuk, akkor polgári alkalmazásra kellően nagy biztonság érhető el. Ekkor maga a kódolás biztonsági rendszerünk egyik legerősebb láncszeme lehet, és szinte biztos,

hogy a kódolást az elkövetkező években, évtizedekben reális mennyiségű erőforrás befektetése mellett nem lehet a gyakorlatban sikeresen támadni. A kódolás biztonsága ekkor is a használt kulcsok kezelésének, tárolásának, továbbításának biztonságára épül. Továbbra is alapvető, hogy a titkos kulcsokat csak arra jogosult felek ismerhessék meg, a magánkulcsokat kizárólag birtokosuk ismerhesse, a nyilvános kulcsokhoz pedig az érintett felek hiteles módon juthassanak hozzá.

2.5.2. Tökéletes titkosítás (one-time-pad)

A tökéletes titkosítás azt jelenti, hogy a támadó – akármennyi erőforrással is rendelkezik – a titkosított üzenet ismeretében semmilyen információhoz nem jut a nyílt üzenettel kapcsolatban. Ez akkor teljesül, ha a nyílt üzenet és a titkosított üzenet független valószínűségi változóknak tekinthetők.

Létezik olyan titkosító algoritmus, amely ezt teljesíti, ez one-time-pad néven ismert. One-time-pad esetén egy n bit hosszú üzenet titkosításához n bit hosszú kulcs szükséges, a kulcs minden egyes bitje friss, egyenletes eloszlású véletlen bit kell, hogy legyen; egy kulcsbitet csak egyszer használhatunk fel, egy kulcs csak egy üzenet titkosítására alkalmas. One-time-pad esetén a titkosítás egy nagyon egyszerű művelet: a titkosított üzenet minden i. bitje úgy áll elő, hogy a nyílt üzenet i. bitjét és a kulcs i. bitjét modulo2 összeadjuk (ez a számítástechnikában a XOR műveletet jelenti). Ez gyakorlatilag azt jelenti, hogy az eredeti üzenet bitjeit 50 százalék valószínűséggel megváltoztatjuk. Shannon bizonyította be, hogy a one-time-pad tökéletes titkosítást jelent[5]. Egyúttal azt is megmutatta, hogy tökéletes titkosítás kizárólag akkor érhető el, ha a kulcs legalább olyan hosszú (legalább annyi véletlen információt tartalmaz), mint az az üzenet, amit titkosítani szeretnénk, és tökéletes titkosítás esetén egy véletlen kulcsbitet csak egyszer használhatunk fel. Ez azt jelenti, hogy nem is létezhet a one-time-padnél hatékonyabb, azaz kevesebb véletlen kulcsbitet használó, tökéletes titkosságot nyújtó algoritmus. [168] , [21]

One-time-pad segítségével elvileg tökéletesen titkos módon küldhetünk el n bit információt partnerünknek, feltéve, hogy már el tudtunk küldeni megfelelően biztonságos módon n bit kulcsot. A gyakorlatban nem használnak one-time-pad-eket, és néhány, rendkívül speciális esettől eltekintve nem is jelentenek megfelelő megoldást. Helyettük inkább a nem optimális, de sokkal hatékonyabb és számos más kedvező tulajdonsággal rendelkező, gyakorlatilag biztonságos, más néven feltételesen biztonságos algoritmusokat használják. [161] Ezen algoritmusok azon feltételezés mellett nyújtanak biztonságot, hogy a támadó adott mennyiségű erőforrással rendelkezik. Ezen algoritmusok nem tökéletesek, „feltörésükhöz”

[5]Ahogy minden kriptográfiai algoritmus esetén, úgy one-time-pad esetén is megteheti a támadó, hogy az összes lehetséges kulcsot kimerítő kereséssel végigpróbálja. Amennyiben a támadó végigpróbálja az összes lehetséges kulcsot, az összes lehetséges (adott hosszúságú) nyílt üzenetet kapja eredményül, és nem jut információhoz azzal kapcsolatban, hogy melyik nyílt üzenetet küldhette a feladó. Minden kulcs azonos valószínűséggel fordulhatott elő, így a kimerítő keresés alapján minden üzenetet azonos valószínűséggel küldhetett a feladó.

a tudomány és a technológia jelen állása szerint „csupán” irreális mennyiségű erőforrás szükséges.

2.6. Példák nyilvános kulcsú kriptográfiai algoritmusokra

2.6.1. Az RSA algoritmus

Az RSA egy nyilvános kulcsú kriptográfiai algoritmus, amelyet Ronald Rivest, Adi Shamir és Len Adleman fejlesztett ki. [156] Ma az RSA messze a legelterjedtebb nyilvános kulcsú kriptográfiai algoritmus.

Az RSA kódolás – akár titkosítást, akár dekódolást, akár aláírás készítését vagy annak ellenőrzését értjük kódolás alatt – *moduláris hatványozás*sal történik. Ez azt jelenti, kódoláskor hatványozást végzünk, a kódoló bemenetét (pl. a titkosítandó vagy aláírandó üzenetet) felemeljük a kulcs[6] által meghatározott hatványra, majd a kapott eredményt elosztjuk egy ún. *modulussal*, és a kapott maradékkal számolunk tovább.

Amikor az Alajos nevű felhasználó RSA kulcspárt generál, a következő műveleteket végzi el:

1. Kiválaszt két nagy véletlen prímszámot. E két prímszám legyen p és q.
2. Összeszorozza őket, a kapott szorzat lesz a *modulus*, $m = p * q$.
3. Kiszámítja a $\Phi(m) = (p-1) * (q-1)$ értéket.
4. Kiválaszt egy olyan e egész számot, amely relatív prím $\Phi(m)$-hez (azaz a legnagyobb közös osztójuk 1).
5. Kiszámít egy olyan d értéket, amelyre

$$d * e = 1 \qquad (modulo\ \Phi(m))$$

azaz, ha e és d szorzatát elosztjuk $\Phi(m)$-mel, akkor 1-et kapunk maradékul.

Az így kiszámított értékek közül:

- Alajos nyilvános kulcsa m és e.
- Alajos magánkulcsa d. (Ezen kívül p és q értéke sem kerülhet nyilvánosságra, de e két számra a későbbiekben már jellemzően nincsen szükség.)

[6] A kódoláshoz használt nyilvános kulcs vagy magánkulcs.

Ekkor bármilyen $x < m$ egész számra igaz, hogy $(x^e)^d = x$ (modulo m). Erre az egyenlőségre épül az RSA kódolás és dekódolás.

Tegyük fel, hogy Bendegúz az x üzenetet szeretné titkosítva elküldeni Alajosnak. Ehhez megszerzi Alajos nyilvános kulcsát, azaz az m és e számokat. Bendegúz a következő módon határozza meg az y titkosított üzenetet:

$$y = x^e \qquad (modulo\ m)$$

Alajos a következő módon állítja vissza az x nyílt üzenetet:

$$x = y^d \qquad (modulo\ m)$$

Tegyük fel, hogy Alajos alá szeretné írni az x dokumentumot. Ekkor a következő módon számítja ki az s aláírást:

$$s = x^d \qquad (modulo\ m)$$

Bendegúz megkapja az x dokumentumot és az s aláírást, és Alajos (m, e) nyilvános kulcsát. A következő egyenlőség ellenőrzésével döntheti el, hogy ezek összetartoznak-e:

$$s^e = x \qquad (modulo\ m)$$

Az RSA algoritmus biztonsága a következőkre épül:

- Nem ismert hatékony algoritmus egy nagy egész szám prímtényezőinek meghatározására. (Ez az ún. IFP, azaz integer factorization problem.) Így a támadó m birtokában nem tudja kiszámítani p és q értékét.
- Nem ismert hatékony algoritmus egy szám moduláris i. gyökének meghatározására. Így a támadó e és $y = x^e$ birtokában nem tudja kiszámítani x értékét.

  Megjegyzés:

  1. RSA esetén a kulcshosszat az m modulus méretével szokás megadni. Ha 2048 bites RSA-ról beszélünk, akkor m hossza 2048 bit, azaz p és q két 1024 bites prím. (Célszerű, ha p és q hasonló nagyságrendbe esik. Ha az egyikük nagyon kicsi, akkor az RSA könnyebben támadható.) Az e nyilvános kitevőt kicsinek szokás választani, míg d valószínűleg 2048 bites lesz.
  2. A p és q prímek kiválasztása a következő módon történik: Generálunk egy kellően nagy véletlen számot, majd megnézzük, hogy prím-e. Az olyan méretű számok esetén, amelyekkel az RSA már biztonságos, hagyományos módon, pl. az Eratosztenészi szita segítségével nem reális megvizsgálni,

hogy az adott szám prím-e. A gyakorlatban valószínűségi alapon működő prímteszteket (pl. a Miller-Rabin tesztet) szokás alkalmazni, de alkalmazható pl. a determinisztikus AKS prímteszt is. [21]

3. A p és q értékekre a kulcsgenerálást követően elvileg nincsen szükség. Ugyanakkor ezen értékek nem juthatnak a támadó kezébe, mert segítségükkel könnyen meghatározható a d magánkulcs. Sok esetben rögtön a kulcsgenerálást követően megsemmisítjük őket. Ugyanakkor vannak olyan alkalmazások, amelyek megőrzik a p és q számokat, mert segítségükkel az RSA műveletek – a kínai maradéktételre alapuló, ún. RSA-CRT módszerek segítségével – gyorsabban végezhetőek el. [71] (A p és q számokkal kizárólag a magánkulccsal végzett műveletek gyorsíthatóak, mert p és q nem hozható nyilvánosságra.)

4. Az RSA szimmetrikus, igaz rá, hogy $(x^e)^d = (x^d)^e \pmod{m}$, így RSA esetén a dekódolás és az aláírás (valamint a titkosítás és az aláírás ellenőrzése) ugyanaz a művelet.

   A fenti szimmetriából az is következik, hogy ha az e értéket véletlenül választjuk, és titokban tartjuk, akkor e akár a magánkulcs szerepét is betöltheti. Ha ekkor d értékét hoznánk nyilvánosságra, akkor d lehetne a nyilvános kulcs.

   Ezek RSA-specifikus tulajdonságok, más nyilvános kulcsú kriptográfiai algoritmusok esetén jellemzően nem állnak fent.

5. Az e nyilvános kitevő kiválasztásakor egyetlen szempont, hogy az relatív prím legyen a $\Phi(m)$ számhoz. Mivel e értékét nyilvánosságra hozzuk, e általában nem véletlen szám, hanem sokszor rögzített érték. Az e-edik hatványra emelést általában az ismételt négyzetreemelés és szorzás (method of repeated squaring) algoritmussal számítjuk ki, és ennek sebességét az határozza meg, hogy e binárisan felírva hány 1-est tartalmaz. Korábban gyakori volt az $e = 3$ választás, mert 3 a legkisebb szóba jöhető e, és binárisan ábrázolva 11. Tekintve, hogy a nagyon kis e értékek ellen léteznek ismert támadások, ma az $e = 65537$ a gyakori választás. (A 65537 prím szám, és binárisan ábrázolva 1 0000000 00000001.) [16]

6. Létezett olyan megközelítés, hogy egy rendszer összes felhasználója azonos modulust használjon, és különböző e értékeket válasszanak. (Ekkor egy központi szerver osztaná a d értékeket.) E megoldás nem szerencsés, léteznek támadások a közös m értékekre. [170]

2.6.2. Az elliptikus görbékre épülő kriptográfia (ECC)

Az elliptikus görbék elméletére épülő kriptográfia (elliptic curve cryptography, ECC) az ún. ECDLP (elliptic curve discrete logarithm problem) nevű matematikai problémára épülő kriptográfiai megoldások együttes elnevezése. ECC alatt több algoritmust is értünk, köztük aláírásra (pl. ECDSA), titkosításra (pl. EC ElGamal) és autentikációra (pl. ECDH) szolgáló algoritmusokat is.

Az ECDLP-re épülő algoritmusokra általánosságban igaz, hogy kisebb kulcsmérettel nyújtanak hasonló biztonságot, mint az RSA. Például egy 160 bites ECC kulcs egy 1024 bites RSA kulccsal, egy 224 bites ECC kulcs egy 2048 bites RSA kulccsal ekvivalens biztonságot nyújt az NIST 2007-es ajánlása szerint. [128] Ugyanakkor az ECDLP-re épülő algoritmusok bonyolultabb műveleteket végeznek a kulcsokkal, így az ECC nem feltétlenül tekinthető gyorsabbnak, mint az RSA. [14]

Az elliptikus görbék fogalmának bevezetése előtt áttekintjük a csoport és a test algebrai fogalmakat, majd ezek alapján bemutatunk egy „nehéz” matematikai problémát, az ECDLP-t. Végül leírjuk, hogy az ECDLP-re hogyan lehet kriptográfiai algoritmusokat, például elektronikus aláírást építeni.

2.6.2.1. Csoportok és testek

Egy halmaz elemei $A = a_1, a_2, a_3, ...$, és egy köztük értelmezett művelet („+”) *csoportot* (G) alkotnak, ha a következők teljesülnek rájuk:

- zártság:
  ha $a_i, a_j \in A$, akkor $a_i + a_j \in A \qquad \forall i, j$;
- asszociativitás:
  $(a_i + a_j) + a_k = a_i + (a_j + a_k) \qquad \forall i, j, k$;
- létezik egységelem:
  $\exists a_e$, amelyre $a_e + a_i = a_i + a_e = a_i \qquad \forall i$;
- minden elemnek létezik egyértelmű inverze:
  $\forall a_i \exists a_j$, amelyre $a_i + a_j = a_e \qquad \forall i, j$;

Csoportot alkotnak például az egész számok az összeadás művelettel.

Egy halmaz elemei $A = a_1, a_2, a_3, ...$, és két köztük értelmezett művelet („+” és „$*$”) *testet* (F) alkotnak, ha:

- F zárt a „+” és „$*$” műveletekre;
- F csoport a „+” műveletre, és „+” e csoportban kommutatív művelet;

- $F \setminus \{0\}$ a „$*$” műveletre csoport, és „$*$” e csoportban kommutatív művelet, és a 0 elem az előző pontban szereplő csoport egységeleme;
- teljesül a disztributivitás, azaz:
  $a_i * (a_j + a_k) = a_i * a_j + a_i * a_k \qquad \forall i, j, k;$

Testet alkotnak például a valós számok az összeadás és a szorzás művelettel.

2.6.2.2. Elliptikus görbék

Az alábbi egyenlet definiál egy F test feletti elliptikus görbét:

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

A fenti egyenletben mind az a, b, c, d, e együtthatók, mind az x és y változók az F test elemei. A görbét azon $(x, y) \in F^2$ pontok alkotják, amelyek kielégítik a fenti egyenletet. További követelmény, hogy a görbe „sima” legyen, azaz ha a fenti egyenletet $f(x, y) = 0$ alakra hozzuk, akkor az f függvény akárhányszor differenciálható legyen.

Attól függően, hogy a görbét (illetve a fenti egyenletet) milyen F test felett értelmezzük, az egyenlet egyszerűbb alakban is felírható. Ha a görbét a valós számok felett értelmezzük, a fenti általános egyenlet az alábbi alakra hozható:

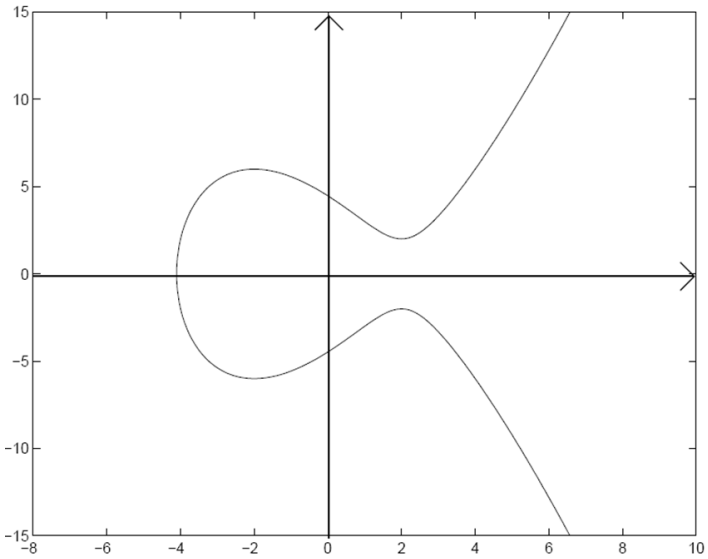
$$y^2 = x^3 + ax + b$$

A valós számok feletti görbe grafikusan is ábrázolható (lásd: 2.7. ábra).

Tekintsünk a görbe részének egy különleges O pontot. Az O egy absztrakt, végtelen távoli pont, amely rajta van minden függőleges egyenesen, és az x tengelyre vonatkozó tükörképe önmaga (azaz $O = -O$).

Az alábbi műveleteket definiáljuk a görbe pontjain:

- *A görbe két pontjának összeadása.* Legyen P és Q a görbe két pontja, ekkor $P \oplus Q$ az alábbi módon határozható meg:
  - Ha $P = O$, akkor $P \oplus Q = Q$. Ha $P = O$, akkor $-P = O$.
  - Ha $P = (x, y)$, akkor $-P = (x, -y)$, azaz $-P$ a P pont x tengelyre vett tükörképe. Ha $Q = -P$, akkor $P \oplus Q = O$.
  - Ha a P és Q pontok x koordinátája különböző, akkor a P és Q pontokon átmenő egyenes egy harmadik ponton is metszi a görbét, e pont legyen $-R$, ennek az x tengelyre való tükörképe R, amely szintén pontja a görbének (lásd: 2.8. ábra). Ekkor $P \oplus Q = R$. Ha $P = Q$, akkor a rajtuk átmenő egyenes alatt a görbe P



2.7. ábra. **Egy elliptikus görbe a valós számok teste felett**

pontbeli érintőjét kell érteni. Ha $-R$ megegyezik a P és Q pontok valamelyikével, akkor úgy kell tekinteni, hogy az egyenes a $-R$ pontban érinti a görbét.

A görbe $P(x_1, y_1)$ és $Q(x_2, y_2)$ pontjainak összeadása algebrailag is definiálható a következő módon:

$$x_3 = s^2 - x_1 - x_2$$

$$y_3 = s(x_1 - x_3) - y_1$$

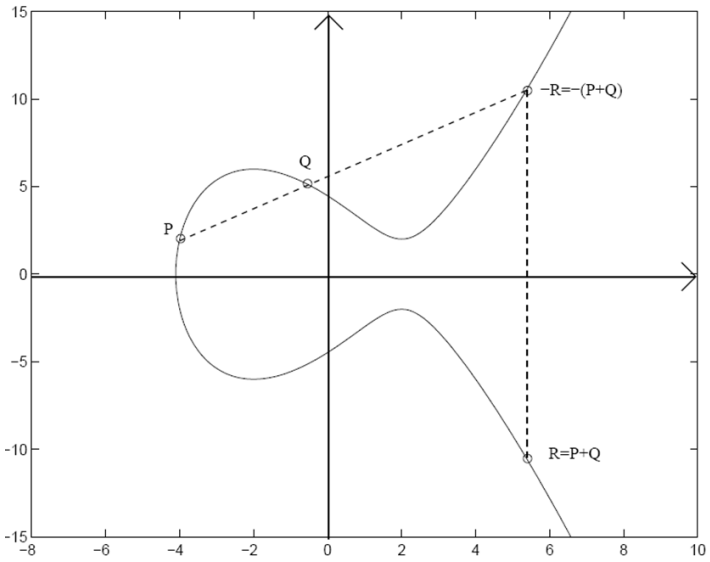
ahol s a görbe „meredeksége". Ha $P \neq Q$, akkor $s = (y_2 - y_1)/(x_2 - x_1)$; ha $P = Q$, akkor $s = (3x_1^2 + a)/2y_1$.

- *Pont szorzása egész számmal.* A görbe Q pontjának a k egész számmal való megszorzása alatt azt érjük, hogy a Q pontot k-szor összeadjuk önmagával. Például: $5 * Q = Q \oplus Q \oplus \oplus Q \oplus Q \oplus Q$.

  Megjegyezzük, nagy k esetén a k-szori összeadásnál ennél sokkal hatékonyabb módon szokás elvégezni ezt a műveletet. Például $33Q$ értékét a következő módon határozhatjuk meg: $2Q = Q \oplus Q$, majd $4Q = 2Q \oplus 2Q$, majd $8Q = 4Q \oplus 4Q$, majd $16Q = 8Q \oplus 8Q$, majd $32Q = 16Q \oplus 16$, és végül $33Q = 32Q \oplus Q$. Így $33Q$ értékét nem 32, hanem csupán 7 darab $\oplus$ művelet elvégzésével határoztuk meg. A gyakorlatban olyan nagy k értékeket szokás használni, ahol már nem lehetséges valamit k-szor végrehajtani.

A görbe pontjai (beleértve az O pontot is) csoportot alkotnak az $\oplus$ műveletre.

A valós számok teste feletti elliptikus görbék grafikusan szemléletesen ábrázolhatóak, de



2.8. ábra. **A görbe P és Q pontjainak összeadása.** $P \oplus Q = R$

kriptográfiai szempontból nem bírnak jelentőséggel. A kriptográfia területén ún. *véges testek*, azaz véges elemszámú testek feletti görbéket szokás használni.

A gyakorlatban a következő testek feletti görbék jelennek meg:

- *A $GF(p)$, azaz a modulo p egész számok (modulo p maradékosztályok) alkotta test, ahol p prímszám.* Ekkor a görbe egyenletében mind az a és b együtthatók, mind az x és y változók $GF(p)$-beli egész számok.

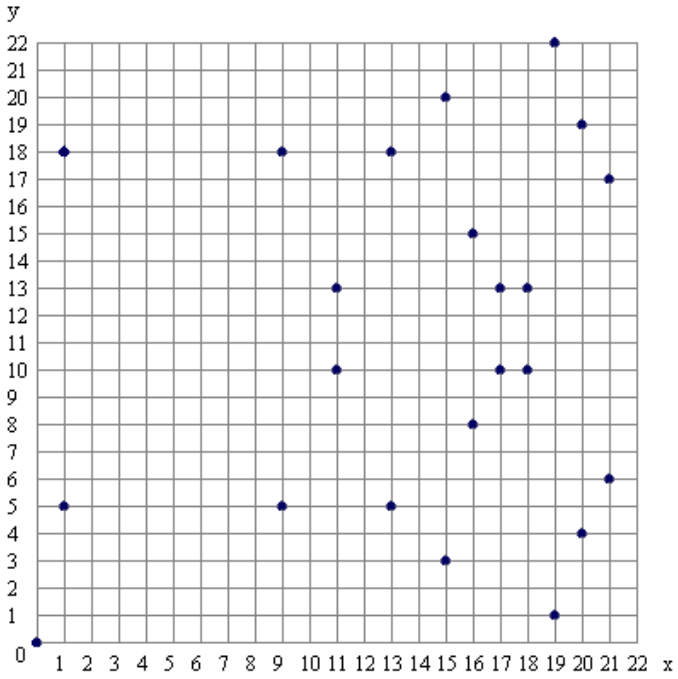
  A $GF(7)$ testben például: $2+2=4$, és $4+4=1$, és $4*4=2$, valamint $2^{-1}=4$ (mert $2*4=1$), és $3/2=3*2^{-1}=5$.

  A $GF(p)$ feletti görbék esetén az $\oplus$ művelet algebrai definíciója megegyezik a valós számok teste feletti görbék esetén használt definícióval.

- *A $GF(2^m)$ test.* E test elemei m hosszú bitsorozatokként vagy m-edfokú bináris polinomokként ábrázolhatóak.

  A $GF(2^m)$ ún. „2-karakterisztiájú" test, így a $GF(2^m)$ feletti görbék esetén a görbe általános egyenlete nem alakítható $y^2 = x^3 + ax + b$ alakra, így a valós számok teste feletti görbék esetén használt definíció egy az egyben nem alkalmazható. (A $GF(2^m)$ esetre alkalmazható képletek a görbe általános egyenletéből vezethetőek le.)

A $GF(p)$ és $GF(2^m)$ testek feletti görbék nem ábrázolhatóak olyan szemléletesen, mint a valós számok teste feletti görbék (lásd: 2.9. ábra), az $\oplus$ művelet esetén is elsősorban az algebrai definíciónak van értelme. Ugyanakkor egyes $GF(p)$ és $GF(2^m)$ testek feletti görbék pontjai alkotta csoportokban a diszkrét logaritmus probléma (DLP) különösen „nehéz", így a



2.9. ábra. **Az $y^2 = x^3 + x$ görbe a $GF(23)$ felett. A „görbe" pontjai azon (x, y) párosok, amelyek kielégítik a görbe egyenletét. (Forrás: certicom.com)**

DLP-re épülő kriptográfiai protokollok e csoportokban hatékonyan alkalmazhatók, rövidebb kulcsokkal is biztonságot nyújtanak.

2.6.2.3. ECDLP (elliptic curve discrete logarithm problem)

Az elliptikus görbék kriptográfiai jelentőségét az adja, hogy egyes véges testek felett értelmezett elliptikus görbék pontjain a diszkrét logaritmus probléma (DLP), azaz az ECDLP „nehéz" feladat, nem ismert rá hatékony algoritmus.

Ha Q egy elliptikus görbe egy pontja, és k egy egész szám, akkor a Q és $k*Q$ pontok ismeretében k meghatározása „nehéz" feladat.

> **Megjegyzés:** A görbe pontjai között értelmezett $\oplus$ műveletre az „összeadás" elnevezést használtuk, mert a szakirodalom általában így hívja e műveletet. Ezen elnevezés önkényes, az $\oplus$ műveletet akár a pontok közti „szorzás" műveletnek is nevezhettük volna. Ebben az esetben a másik művelet, a többször egymás után elvégzett szorzás a „hatványozás" nevet kapta volna. Ekkor az ECDLP alatt azt értettük volna, hogy a Q és a Q^k pontok ismeretében k meghatározása, azaz a logaritmusképzés „nehéz" feladat.

Miller és Koblitz 1985-ben egymástól függetlenül javasolták az elliptikus görbék pontjain

értelmezett diszkrét logaritmus probléma (ECDLP) kriptográfiai alkalmazását. [114], [95], [159]

A következőkben bemutatott protokollok az ECDLP nehézségét használják ki.

2.6.2.4. Kulcsgenerálás

Nem könnyű megfelelő, azaz kellően biztonságos és kellően hatékony görbét választani, ezért „nevezetes", nemzetközi szervezetek által ajánlott görbéket szokás használni. Az USA-ban az NIST (FIPS 186-3), az EU-ban a „Brainpool" munkacsoport javasolt görbéket. [65], [38] Általában egy ECC-re épülő rendszer minden felhasználója egyazon G görbét és annak egyazon Q pontját használja, tehát G és Q közös, nyilvános információk.

Amikor az Alajos nevű felhasználó ECC kulcspárt generál, a következő műveleteket végzi el:

1. Generál egy k_A véletlen számot. Ez lesz az ő magánkulcsa.
2. Megszorozza a Q pontot a k_A számmal, a $k_A * Q$ pont lesz az ő nyilvános kulcsa.

   Megjegyzés: Amikor ECC esetében bitekben kifejezett kulcshosszról beszélünk, akkor a k_A magánkulcs méretét értjük alatta.

ECC esetén a kódolási műveletek nem olyan szimmetrikusak, mint pl. RSA esetén. Itt az aláírás és a titkosítás nem inverz műveletei egymásnak, hanem egészen más módon történnek.

2.6.2.5. ECDH – elliptikus görbék feletti Diffie-Hellman protokoll

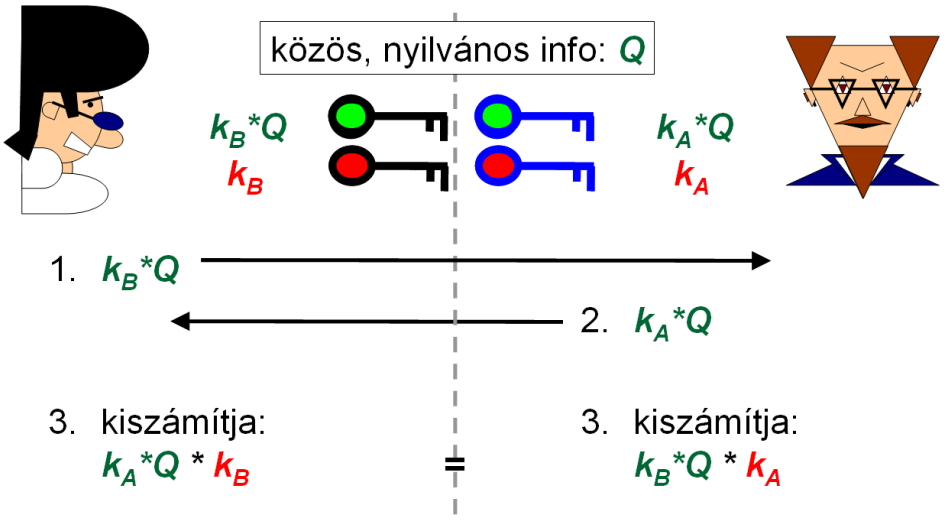
Az ECDH protokoll segítségével két fél – Alajos és Bendegúz – nyilvános csatornán keresztül állapodhatnak meg közös titokban.

1. Alajos és Bendegúz átküldik egymásnak a nyilvános kulcsaikat: Alajos elküldi Bendegúznak a $k_A * Q$ nyilvános kulcsot. Bendegúz elküldi Alajosnak a $k_B * Q$ nyilvános kulcsot.

   Ha a csatornán hallgatózó támadó nem tudja megoldani az ECDLP-t, akkor a kapott adatokból nem ismeri meg sem k_A-t, sem k_B-t.

2. Mindkét fél megszorozza a kapott nyilvános kulcsot a saját magánkulcsával:
   Alajos kiszámítja a $(k_B * Q) * k_A = k_A * k_B * Q$ értéket.
   Bendegúz kiszámítja a $(k_A * Q) * k_B = k_A * k_B * Q$ értéket.

Az eredményül kapott, $k_A * k_B * Q$ értéket csak ők ketten tudják kiszámítani, csak ők ismerik. Ez az érték lesz a közös titkuk, ebből képezhetnek közös szimmetrikus kulcsot (lásd: 2.10. ábra).



2.10. ábra. **Az elliptikus görbék feletti Diffie-Hellman protokoll**

2.6.2.6. EC ElGamal – elliptikus görbék feletti ElGamal protokoll

Az EC ElGamal protokoll segítségével titkos üzenetet lehet küldeni valakinek (lásd: 2.11. ábra). Tegyük fel, hogy Bendegúz titkos üzenetet szeretne küldeni Alajosnak. Üzenetének a görbe M pontját felelteti meg, az M pont koordinátáit szeretné átküldeni.

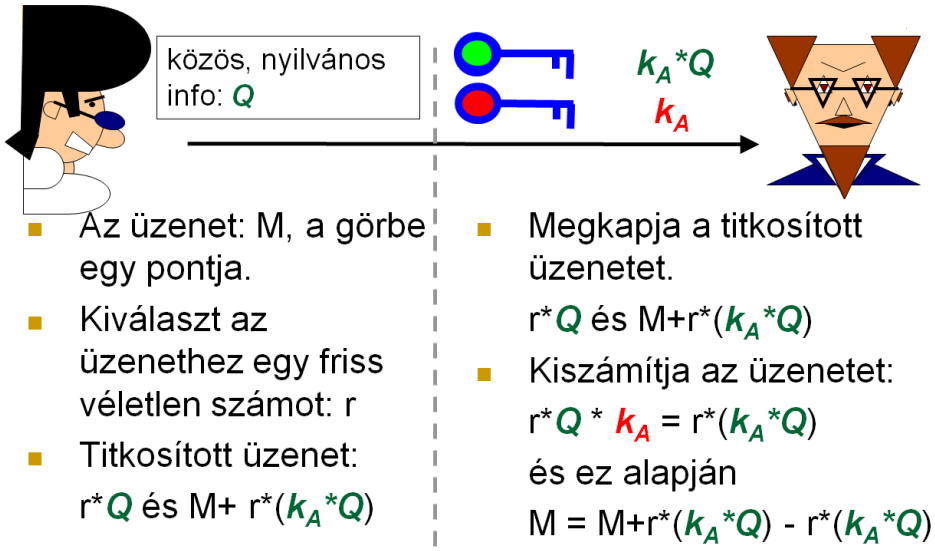
1. Bendegúz kiválaszt egy r friss véletlen számot.

2. Bendegúz kiszámítja a titkosított üzenetet, az
   $r*Q$ és az $M \oplus r*(k_A*Q)$
   értékeket, azaz a görbe két pontját. E két pont koordinátáit küldi át Alajosnak. Bendegúz csak a $(k_A * Q)$ értéket (azaz Alajos nyilvános kulcsát) ismeri, a k_A magánkulcsot nem.

3. Alajos a kapott $r*Q$ pontot megszorozza a saját k_A magánkulcsával, így megkapja az $r*k_A*Q$ pontot. E pontot kivonja a kapott $M \oplus r*(k_A*Q)$ pontból, és eredményül megkapja a görbe M pontját, vagyis azt az üzenetet, amelyet Bendegúz küldött.

   A protokoll biztonsága arra épül, hogy Alajoson és Bendegúzon kívül más nem tudja kiszámítani az $r*k_A*Q$ pontot a csatornán megjelenő és a nyilvános információk alapján.

Megjegyzés: Az EC ElGamal titkosításhoz friss véletlen szám generálására van szükség, míg az RSA titkosítás önmagában determinisztikus művelet, nem igényel véletlen számot. Általában is igaz, hogy jó minőségű titkosításhoz randomizálásra van szükség, mert ha teljesen determinisztikus titkosítást alkalmazunk, akkor a támadó észleli, ha kétszer ugyanazt az üzenetet küldjük el. RSA esetében is szokás randomizálni, véletlen padding alkalmazásával.



2.11. ábra. **Az elliptikus görbék feletti ElGamal protokoll**

2.6.2.7. ECDSA – elliptikus görbék feletti DSA (digital signature algorithm)

Tegyük fel, hogy az Alajos nevű felhasználó alá szeretné írni az m dokumentumot. Ehhez a következő lépéseket végzi el ($GF(p)$ feletti görbe esetén):

1. Kiszámítja az $e = h(m)$ $(modulo\ p)$ értéket, ahol $h()$ egy hash függvény.
2. Generál egy t véletlen egész számot, ahol $t \in [1, n-1]$.
3. Kiszámítja az $r = (t*Q)[x]$ (modulo n) értéket, ahol $(t*Q)[x]$ a görbe $t*Q$ pontjának x koordinátáját jelenti.
4. Kiszámítja az $s = t^{-1}*(e+r*k_A)$ $(modulo\ p)$ értéket. (E művelethez van szükség a k_A magánkulcsra.)

Az így kapott (r, s) páros az Alajos nevű felhasználó aláírása az m dokumentumon.

Az aláírás ellenőrzéséhez szükség van az m dokumentumra, az (r, s) aláírásra, az aláíró $k_A * *Q$ nyilvános kulcsára, valamint a közös G görbére és annak Q pontjára. Az ellenőrző fél a következő lépéseket végzi el:

1. Elvégzi a lenyomatképzést, azaz kiszámítja az $e = h(m)$ $(modulo\ n)$ értéket.
2. Kiszámítja a $w = s^{-1}$ $(modulo\ n)$ értéket.
3. Kiszámítja az $u_1 = (e*w)$ $(modulo\ n)$ értéket.
4. Kiszámítja az $u_2 = r*w$ $(modulo\ n)$ értéket.

5. Kiszámítja a görbe $(x_1, y_1) = u_1 * Q + u_2 * k_A * Q$ pontját.

   Ebből a Q pont kiemelésével:
   $(x_1, y_1) = u_1 * Q + u_2 * k_A * Q = Q * (u1 + u2 * k_A)$

   Az aláírás készítésének 4. lépése szerint $s = t^{-1} * (e + r * k_A)$ (modulo n), ezt átrendezve:
   $t = s^{-1} * (e + r * k_A) = w * e + w * r * k_A = u_1 + u_2 * k_A$

   Ezt az előző egyenletbe behelyettesítve:
   $(x_1, y_1) = Q * (u1 + u2 * k_A) = Q * t$

   Ahol x_1 az aláírás készítésének 3. lépése miatt meg kell, hogy egyezzen az aláírás r komponensével.

6. Az ellenőrző fél akkor fogadja el az aláírást, ha $x_1 = r$. [90]

2.6.2.8. Hol használnak ECC-t?

A legtöbb nyilvános kulcsú alkalmazás jelenleg RSA-ra épül, bár sok alkalmazás támogatja az ECC-t, és vannak ECC-alapú nemzetközi hitelesítés-szolgáltatók.

Az amerikai nemzetbiztonsági hivatal, az NSA 2009. végén kibocsátott, kriptográfiai algoritmusokra vonatkozó ajánlása szerint *át kell térni az ECC-re*, és csakis az áttérés lezártáig lehet még RSA-t használni, 2048 bites kulcshosszal.

A biometriai azonosítókat (ujjlenyomatot) tartalmazó *EU-s útleveleken lévő chipek*ből csak arra jogosult készülékek olvashatják ki az útlevél birtokosának biometriai azonosítóit. Az útlevélchip titkosított és hitelesített csatornát épít ki az olvasókészülékkel, e csatorna felépítése során mind az útlevélchip, mind az olvasókészülék tanúsítvánnyal azonosítja magát (egy, az SSL-hez (10.3.2. fejezet) hasonló protokollon keresztül). Mind az olvasókészülékek, mind az útlevélchipek tanúsítványai egy dedikált, ECC-re épülő PKI hierarchiából származnak, és e berendezések ECC alapon egyeztetik a kapcsolat titkosításához és hitelesítéséhez használt kriptográfiai kulcsokat. (Lásd: 13.4. fejezet.)

2.6.3. Lamport aláírások

A nyilvános kulcsú kriptográfia alapjait Diffie és Hellman fektették le 1976-ban megjelent cikkükben. [32] Ezen első cikk egy DLP-re alapuló kulcscsere-protokollt mutat be, amely azóta is Diffie és Hellman nevét viseli. (E protokoll ECDLP-re épülő változatát korábban (2.6.2.5. fejezet) be is mutattuk.) Már a kezdetekkor feltételezték, hogy nyilvános kulcsú kriptográfia segítségével a kulcscserén kívül titkosítani és aláírni is lehet, de Diffie és Hellman cikkében még csak kulcscsere-protokoll szerepelt.

Lamport publikálta az egyik első digitális aláírás sémát. [103] A Lamport aláírások sok szempontból hasonlítanak a one-time-padhez: nagyon nagy mennyiségű véletlen kulcsbitet

használnak, egy kulcsbitet egyszer szabad felhasználni, ugyanakkor igen „masszív” biztonságot nyújtanak (bár tökéletes biztonságról itt nem beszélhetünk).

Tegyük fel, hogy egy felhasználó n bit hosszú dokumentumot szeretne Lamport aláírással ellátni. A felhasználónak ekkor egy $2*n$ db tetszőleges hosszúságú véletlen bitsorozatból álló magánkulcsra van szüksége. Ezek legyenek: O_1, O_2, ..., O_n, és a Z_1, Z_2, ..., Z_n. A felhasználó nyilvános kulcsa szintén $2*n$ db bitsorozatból áll, amelyek a magánkulcsot alkotó bitsorozatok lenyomatai, azaz: $h(O_1)$, $h(O_2)$, ..., $h(O_n)$, és a $h(Z_1)$, $h(Z_2)$, ..., $h(Z_n)$, ahol $h()$ egy hash függvény. A nyilvános kulcsot a felhasználó nyilvánosságra hozza.

A felhasználó úgy hozza létre az aláírást, hogy a magánkulcs egy részét is nyilvánosságra hozza. Az aláírandó dokumentum minden i. bitjére nyilvánosságra hoz egy-egy bitsorozatot. Ha az aláírandó dokumentum i. bitje 1, akkor az O_i értéket, ha az aláírandó dokumentum i. bitje 0, akkor a Z_i értéket hozza nyilvánosságra. Az adott kulcspárt kizárólag egyetlen alkalommal, egyetlen dokumentum aláírására szabad használni.

Tegyük fel, hogy az aláírt dokumentum j. bitje 0. Ha a támadó a j. bitet 1-re szeretné változtatni, akkor fel kellene tudnia mutatni egy olyan O'_j értéket, amelyre $h(O'_j) = h(O_j)$. Mivel O_j értéke független valószínűségi változónak tekinthető minden Z_k ($\forall k$) és O_k ($\forall k \neq j$) értéktől, ezért a támadó kizárólag a hash függvény invertálásával próbálkozhat.

Bizonyítható, hogy egy csaló üzenet generálásához a támadónak invertálnia kellene a hash függvényt, azaz meg kellene sértenie az őskép-ellenállóságát. Bár a Lamport aláírás nem nyújt feltétel nélküli biztonságot, ez nagyon erős állítás; nagyon ritka, hogy egy hash függvény őskép-ellenállósága támadható lenne. (A támadások legtöbbször a hash függvény ütközés-ellenállóságát szokták kikezdeni.)

Annak ellenére, hogy a Lamport aláírások biztonságáról nagyon erős állítást tehetünk[7], a gyakorlatban nem használják őket. A Lamport aláírások leginkább elméleti jelentőséggel bírnak, érdekességként szolgálnak.

Nehezen képzelhető el nyilvános kulcsú infrastruktúra Lamport aláírás alapon. Egyrészt az aláíró kulcspárok egyszeri felhasználhatósága, másrészt a sok véletlen bit miatt.

2.7. Hogyan kombinálhatók ezen alapelemek?

2.7.1. Üzenetek titkosítása

Amikor titkos üzenetet küldünk valakinek, általában nem a teljes üzenetet titkosítjuk a címzett nyilvános kulcsával, mert ez nagyon lassú és időigényes volna. Ehelyett (minden egyes üzenet titkosításához) kisorsolunk egy véletlen szimmetrikus kulcsot, és csak ezt a

[7]Egyes nézetek szerint a Lamport aláírások akkor is biztonságosak maradnak, ha a kvantum-számítógépek fejlődésével más nyilvános kulcsú kriptográfiai algoritmusok (pl. RSA, ECC) használhatatlanná válnak.

szimmetrikus kulcsot titkosítjuk a címzett nyilvános kulcsával, magát az üzenetet pedig ezen szimmetrikus kulccsal titkosítjuk.

Az üzenet címzettje a saját magánkulcsa segítségével tudja visszafejteni a nyilvános kulcsával titkosított szimmetrikus kulcsot, és e szimmetrikus kulcs birtokában vissza tudja fejteni az üzenetet is.

Előnye e megoldásnak, hogy egy fájlt, dokumentumot úgy is titkosíthatunk, hogy azt több címzett is visszafejthesse. Ilyenkor a fájl titkosítására használt szimmetrikus kulcsot különkülön kell titkosítani minden egyes címzett nyilvános kulcsával, és az így titkosított blokkokat kell elküldenünk a fájllal együtt.

Így működik például a levelezőprogramok által használt S/MIME vagy a W3C által kifejlesztett XML encryption is.

2.7.2. Digitális aláírás

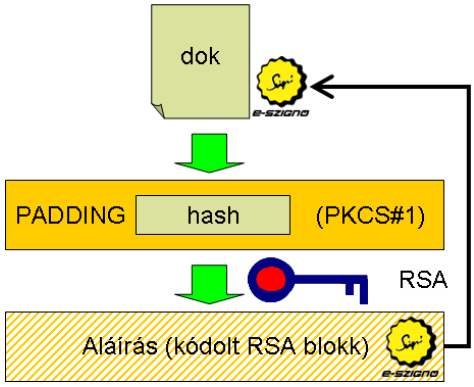
Aláíráskor nem a teljes dokumentumot szokás kódolni a magánkulccsal. Ez egyrészt nagyon lassú és időigényes volna, másrészt a magánkulcsot minél ritkábban célszerű használni, mert minden egyes kódolás egy-egy aláírást jelent. Szerencsés, ha az aláíró tisztában van vele, hogy mikor ír alá valamit. Ezért egy lenyomatképző függvénnyel kriptográfiai lenyomatot képzünk az aláírandó dokumentumból, és csak ezt a lenyomatot kódoljuk a magánkulcsunkkal. (Lásd: 2.12. ábra.) Így a magánkulccsal csak egyetlen műveletet végzünk. A magánkulcsunkkal kódolt, aláírt lenyomatot nevezzük aláírásnak.

Aki ellenőrizni szeretné az aláírást, annak az aláírt dokumentumra is szüksége van. Ellenőrzéshez neki is lenyomatot kell képeznie az aláírt dokumentumból, majd az aláíró nyilvános kulcsával kódolnia kell a dokumentumhoz tartozó aláírást (így azt a lenyomatot kapja vissza, amit az aláíró aláírt). Ha az általa képzett lenyomat megegyezik az aláírásból visszanyert lenyomattal, akkor az aláírás érvényes.

Amikor a technológia jogszabály által is elismert változatáról szólunk, akkor a digitális aláírás kifejezés helyett az elektronikus aláírást szokás használni. Ezzel szemben digitális aláírás kifejezést olyan esetekben is szokás használni, amikor nem jognyilatkozatot teszünk, hanem egy véletlen „kihívás” aláírásával igazoljuk, hogy birtokoljuk a nyilvános kulcsunkhoz tartozó magánkulcsot. (Lásd: 2.7.3. fejezet.)

A műszaki szakemberek eredetileg a digitális aláírás kifejezést használták e fogalomra. Az „elektronikus aláírás” kifejezést az EU direktíva vezette be, hogy a szabályozás technológiafüggetlen legyen. [1] A direktíva leírja, hogy milyen tulajdonságokkal kell rendelkeznie az elektronikus aláírásnak, de nem kapcsolja közvetlenül össze sem a kriptográfiával, sem a digitális aláírás technológiával. [62]

> **Megjegyzés:** A közelmúltban – elsősorban az MD5, illetve az SHA-1 függvényben felfedezett gyengeségekre válaszul – előtérbe került az ún. *randomized*



2.12. ábra. **A dokumentumból lenyomatot képzünk, a lenyomatot paddinggel egészítjük ki, az így kapott blokkot kódoljuk a magánkulcsunkkal**

hashing (más néven hash salting) nevű technika. Ez azt jelenti, hogy az aláíró az aláírásra kerülő bitsorozatban (jellemzően annak elején) egy friss véletlen számot helyez el, és ezen véletlen bitsorozattal együttesen írja alá. E véletlen bitsorozat nem titkos, sőt az aláírás ellenőrzéséhez is szükség van rá. [127]

A randomized hashing technikát akkor szokás használni, ha az aláíró egy valaki más által előállított bitsorozatot ír alá, és nem bízik meg teljesen a használt hash függvény ütközés-ellenállóságában.

Ha az aláírásra kerülő bitsorozatot összeállító fél ki tudja használni egy hash függvény gyengeségét, és elő tud állítani két olyan bitsorozatot, amelynek azonos a lenyomata, akkor csalhat: Aláírathajta az egyik dokumentumot az aláíróval, és az így kapott aláírás a másik dokumentumra is érvényes lesz. A randomized hashing az ellen véd, hogy az aláírásra kerülő bitsorozatot összeállító fél – egy gyenge, már nem teljesen ütközés-ellenálló hash függvény esetén – ne tudjon olyan dokumentumot aláíratni az aláíróval amelynek az aláírása egyúttal egy másik dokumentumhoz is tartozik. A beszúrt véletlen szám a hash függvény ütközés-ellenállóságát erősíti meg ezen támadással szemben.

Ugyanakkor nem biztosít védelmet e technika pl. azzal szemben, ha maga az aláíró próbál csalni. Mivel ő befolyásolhatja, hogy milyen véletlen számot illeszt az aláírandó információhoz, az ő számára ez nem nehezíti meg, hogy két olyan dokumentumot állítson elő, amelyekhez azonos aláírás tartozik. Szintén nem véd ez a technika, ha a hash függvény már „nagyon gyenge”, azaz például már nem második őskép-ellenálló.

Álláspontunk szerint nem a randomized hashing jelenti a helyes megoldást, helyette inkább olyan hash függvényeket kell használni, amelyek ütközés-ellenállóságával kapcsolatban nem merült fel probléma. Ha ez valami miatt nem

oldható meg, akkor lehet ilyen megoldásokhoz folyamodni.

Ma a hitelesítés-szolgáltatók számára ajánlott, hogy a tanúsítványok elején (jellemzően a tanúsítvány sorozatszámában) néhány byte véletlen információt helyezzenek el. [115] Így a tanúsítvány igénylője – aki a tanúsítványba kerülő adatokat megadja – még gyenge hash függvény esetén sem tudja más célra használni a szolgáltató aláírását. (A későbbiekben (10.4.2.4. fejezet) bemutatunk egy konkrét támadást, ahol ez ténylegesen megtörtént.)

2.7.3. Biztonságos bejelentkezés

Gyakori, hogy nem levelek küldésekor, hanem online kapcsolatok felépítésekor használnak kriptográfiai módszereket. Ekkor a két fél – a kliens és a szerver – általában nyilvános kulcsú kriptográfiai módszerek segítségével közös szimmetrikus kulcsokban állapodnak meg (ezt nevezzük kulcscserének), és a kommunikációt később ezen szimmetrikus kulcsú módszerekkel titkosítják, hitelesítik.

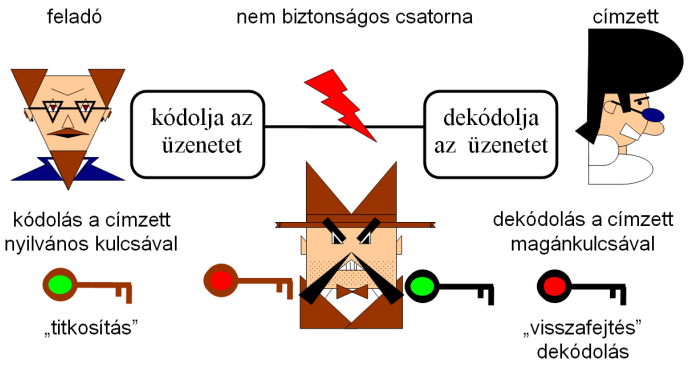
E megoldás egyik alapköve a *kihívás és válasz alapú hitelesítés* (challenge and response authentication), melynek keretében például a szerver megállapíthatja, hogy valóban a klienssel kommunikál-e. Példánkban a szerver egy véletlen számot generál, azt küldi el a kliensnek, a kliens a saját magánkulcsával kódolja azt. Az így kódolt véletlen számot visszaküldi a szervernek, aki a kliens nyilvános kulcsával dekódolja — megállapítja, hogy valóban a kliens magánkulcsával kódolták-e az általa generált véletlen számot. (Lásd: 10. fejezet.)

(Ekkor magánkulccsal való kódolás történik ugyan, de e művelet jogi értelemben nem nevezhető elektronikus aláírásnak, hiszen itt jogi értelemben nem történik aláírás, nem történik nyilatkozattétel. Az elektronikus aláírásról szóló törvény szerint az aláírásra szolgáló magánkulcsot kizárólag elektronikus aláírás létrehozására szabad felhasználni (13. § (4)). Ennek egyik oka, hogy még véletlenül se keveredhessen össze, hogy valaki épp alá akart írni egy dokumentumot, vagy csak egy szerverre akart bejelentkezni. Így az aláírásra használt kulcspárt semmilyen más célra nem szabad használni.)

A fent leírt kihívás és válasz alapú hitelesítésre, majd azt követően szimmetrikus kulcsok alapján történő titkosított és hitelesített kommunikációra épül például az SSH és az SSL protokoll (de sok VPN megoldás is így működik). SSH és SSL esetén általában csak a kliens győződik meg kriptográfiai módszerek segítségével a szerver kilétéről (és a szerver ilyenkor csak a kliens jelszavát ellenőrzi), de mindkét megoldás támogat kétoldalú kihívás és válasz alapú hitelesítést is.

2.8. Hogyan jutunk hozzá valakinek a nyilvános kulcsához?

Egy nyilvános kulcsú rendszer minden résztvevőjének van két kulcsa (lásd: 1.1. ábra):

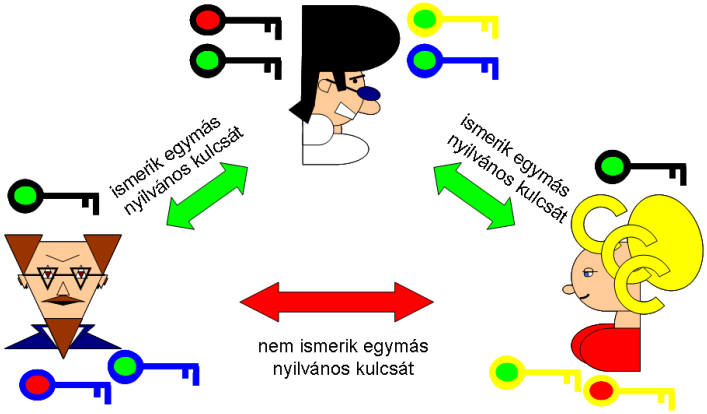


2.13. ábra. **A nyilvános kulcsot hitelesen kell megszerezni. Ha Manfréd, a támadó, becsempészi a saját nyilvános kulcsát Bendegúz nyilvános kulcsának helyére, akkor Alajos Bendegúznak küldött titkos üzeneteit Manfréd fejtheti meg.**

- magánkulcsa, amelyet titokban tart, és
- nyilvános kulcsa, amelyet akár nyilvánosságra is hozhat.

E rendszerek arra épülnek, hogy mindenkinek külön magánkulcsa van, és a saját magánkulcsát mindenki csak saját maga ismeri, míg bárki megismerheti bárkinek a nyilvános kulcsát, de a nyilvános kulcsokhoz *hitelesen* kell hozzájutni. Ha nem győződünk meg róla, hogy kié a nyilvános kulcs, amit használunk, lehet, hogy éppen a támadó az, akivel „biztonságosan” kommunikálunk. Ha Alajos a támadó nyilvános kulcsáról elhiszi, hogy az Bendegúz nyilvános kulcsa, akkor a Bendegúznak küldött titkosított üzeneteit a támadó el fogja tudni olvasni (2.13. ábra), illetve a támadó által aláírt üzeneteket ekkor Alajos Bendegúztól származó hiteles üzeneteknek fogja elfogadni. (Az ehhez hasonló támadásokat, amikor a támadó két egymással kommunikáló fél közé kerül, akik ezen túl a támadón keresztül kommunikálnak, közbeékelődéses támadásnak vagy man-in-the-middle támadásnak nevezzük.) Ezért minden nyilvános kulcsú kriptográfiára épülő rendszerben biztosítani kell, hogy a nyilvános kulcs hitelesen jusson el az érintett felekhez.

Egyik lehetőség az ún. *out-of-band* megoldás, azaz ha a kulcs valamilyen biztonságos csatornán, és nem a nyílt hálózaton jut el a címzetthez. Ilyen megoldás például ha Alajos és Bendegúz találkoznak, bemutatkoznak, megmutatják egymásnak az igazolványaikat, és Bendegúz Alajos kezébe adja a saját nyilvános kulcsát (pl. CD lemezen). E megoldással az a probléma, hogy éppen azért szeretnénk nyilvános kulcsú kriptográfiát használni, mert nincsen megbízható csatorna a feladó és a címzett között. Megbízható csatornából kevés van, és ezek fenntartása költséges vagy problémás. Ráadásul, ha n szereplő szeretne egymással páronként megbízható csatornán kommunikálni, akkor $n*(n-1)/2$ megbízható csatornára lenne szükségünk. Ezért az out-of-band megoldások nem skálázhatóak, nagy tömegben más megoldásokat célszerű használni helyettük, de minden rendszerben van néhány olyan nyilvános kulcs, amely out-of-band módon jut el a szereplőkhöz. Ilyenek például az ún. megbízható



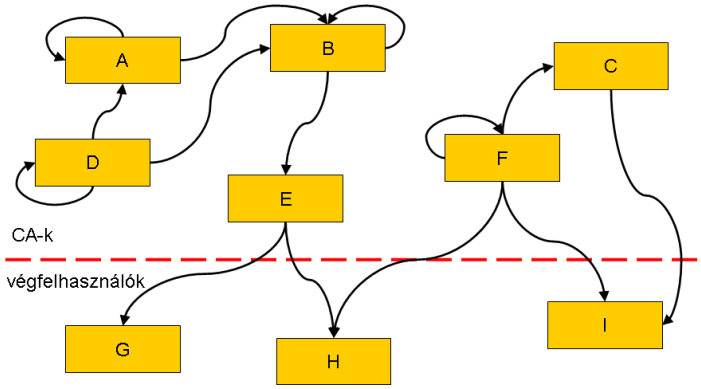
2.14. ábra. **Alajos és Cili ismerik Bendegúz nyilvános kulcsát, de egymás nyilvános kulcsát még nem ismerik**

gyökértanúsítványok nyilvános kulcsai (4.1.4. fejezet). Vannak rendszerek, ahol csak out-of-band megoldást használnak, pl. jellemzően így szokott történni az SSH bejelentkezéshez szükséges RSA kulcsok kiosztása.

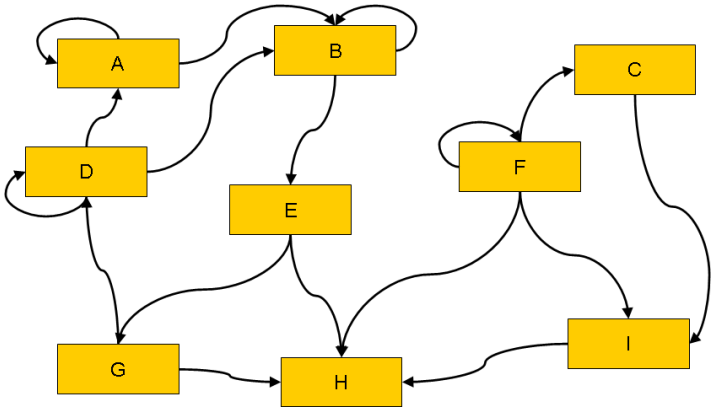
Sok megoldás arra épül, hogy vannak olyan megbízható felek, akiknek a nyilvános kulcsát már ismerjük (így aláírásaikat már ellenőrizni tudjuk), és ők aláírva juttatják el hozzánk mások nyilvános kulcsait. Tegyük fel, hogy Alajos és Cili ismerik Bendegúz nyilvános kulcsát, de egymás nyilvános kulcsát még nem ismerik! (2.14. ábra) Ekkor Bendegúz küldhet Alajosnak egy olyan aláírt üzenetet, amelybe beleteszi Cili nyilvános kulcsát, és azt is mellé írja, hogy „ez Cili nyilvános kulcsa”. Alajos – Bendegúz nyilvános kulcsa birtokában – ellenőrizheti az aláírást (meggyőződhet róla, hogy az üzenet Bendegúztól származik, és Bendegúz pontosan ezt az üzenetet küldte és – mivel megbízik Bendegúz állításában – elfogadja, hogy valóban Cili nyilvános kulcsát kapta meg. Ezt követően Alajos már ismeri Cili nyilvános kulcsát, így ha Cili küld egy aláírt üzenetet, amelyben szerepel egy nyilvános kulcs, és az szerepel benne, hogy „ez Dezső nyilvános kulcsa”, akkor – megbízva Cili állításában – elfogadhatja, hogy valóban Dezső nyilvános kulcsát kapta meg. Több rendszer is ilyen elvek mentén épül fel.

A *nyilvános kulcsú infrastruktúra (PKI)* szerint egyes kitüntetett szereplők ún. hitelesítés-szolgáltatók, és csak ők jogosultak igazolni, hogy a nyilvános kulcsok bizonyos entitásokhoz tartoznak; ezen igazolásokat tanúsítványnak nevezzük. Vannak ún. gyökér hitelesítés-szolgáltatók, akiknek a nyilvános kulcsát – jellemzően önhitelesített gyökértanúsítvány formájában – out-of-band módszerekkel kapják meg a résztvevők, minden más tanúsítványt (mind a végfelhasználói tanúsítványokat, mind a hitelesítés-szolgáltatók tanúsítványait) ezen gyökerekre próbálják visszavezetni. (Lásd: 2.15. ábra.) A PKI legfontosabb alapelveit az ITU-T X.509 specifikációja fekteti le. [191]

A *PGP – pretty good privacy* elosztott rendszerben gondolkozik. Nincsenek kitüntetett szereplők, bárki jogosult állítani, hogy egy nyilvános kulcs és egy szereplő összetartoznak;



2.15. ábra. **A PKI modellje – Egyes kitüntetett szereplők hitelesítés-szolgáltatók (CA-k), ők jogosultak igazolni, hogy egy nyilvános kulcs egy adott entitáshoz tartozik.**



2.16. ábra. **A PGP modellje – Nincsenek kitüntetett szereplők, bárki jogosult állítani, hogy egy nyilvános kulcs egy adott entitáshoz tartozik.**

akik ismerik egymást, azok aláírják egymás nyilvános kulcsát. Amikor azt vizsgáljuk, hogy egy adott kulcs valóban egy adott személyhez tartozik-e, a saját PGP kulcsunkra próbáljuk visszavezetni. A PGP felhasználó megadhatja, hogy kinek az állítását fogadja el, illetve hány megbízható szereplő állítása szükséges ahhoz, hogy elfogadja, hogy egy szereplő és egy kulcs összetartoznak. (Lásd: 2.16. ábra.) A PGP és variánsai (pl. GPG – GNU privacy guard) bizonyos körökben (ide tartozik az egyetemi szféra, illetve a szabad szoftverek fejlesztői) elterjedtek, de használatukhoz a technológiát értő és biztonságilag tudatosan gondolkozó felhasználókra van szükség, így az elektronikus ügyintézés területén a központosított alapokon nyugvó PKI elterjedése a várható.

Más megoldások is léteznek a fent leírt problémára. Az *identity-based cryptography* olyan speciális technikákat használ, amelynek segítségével a nyilvános kulcsok a felhasználók azonosítójából képezhetőek. [167], [17] A ZPhone nevű VOIP megoldás pedig arra épít, hogy az első kapcsolatfelvételt követően a partnerek alkalmazásai megjegyzik egymás nyilvános

kulcsait, így a man-in-the-middle támadásra csak az első kapcsolatfelvételkor van lehetőség (és a támadó még ekkor is lebukik, ha a partnerek gondosan járnak el, és egyeztetik a kliensük által megjelenített ellenőrző összeget). [194] E különféle megoldásokról remek összefoglaló olvasható Carlisle Adams és Mike Just „PKI: Ten Years Later" című, 2004-es cikkében. [**?**]

A következő fejezetekben a nyilvános kulcsú infrastruktúra (PKI) tanúsítványokra épülő módszereit mutatjuk be részletesen.

2.9. Titkosság és hitelesség

Az eddigiekben többször is használtuk a *titkosság* és a *hitelesség* kifejezéseket.

A titkosság és a hitelesség két teljesen különböző követelmény, amelyeket hasonló kriptográfiai eszközökkel valósíthatunk meg, de ezen eszközöket eltérő módon célszerű alkalmazni, ha titkosságot, illetve ha hitelességet szeretnénk elérni.

Ha titkosítás a célunk, úgy kódoljuk az információt, hogy azt kizárólag a jogosult fél fejthesse vissza (dekódolhassa), illetéktelen fél ne legyen rá képes. Titkosítás esetén nem cél, hogy illetéktelen fél ne legyen képes észrevétlenül manipulálni a kódolt információt. Igaz, hogy a támadó nem tudja megfejteni a titkosított információt, de rendelkezhet bizonyos információkkal az üzenet felépítéséről, esetleg ismerheti egyes részeit is. Például a támadó tudhatja, hogy a kódolt üzenet egy átutalási megbízás, akár azt is tudhatja, hogy az hogyan épül fel, milyen mezőkből áll, sőt még egyes mezők tartalmáról is lehetnek feltételezései.

Léteznek olyan titkosítási megoldások, amelyek hitelességet is biztosítanak, de ha a támadó nem tudja elolvasni az üzenetet, az még önmagában nem jelenti azt, hogy manipulálni sem tudja. Például ha úgy járunk el, hogy az üzenetet blokkokra bontjuk, majd ezen blokkokat egymástól függetlenül titkosítjuk (ezt nevezik ECB – electronic code book módszernek), akkor az üzenet titkos lesz ugyan, de a támadónak számos lehetősége marad a manipulációra: kivághat, megcserélhet blokkokat, korábbi üzenetekből beszúrhat, stb. Másik példa: A one-time-pad tökéletes titkosítást jelent, megfejthetetlenné teszi az üzenetet, de önmagában semmilyen védelmet nem biztosít annak (akár célzott) manipulációja ellen (a támadó az üzenet bármely bitjét észrevétlenül invertálhatja).

Ha hitelesítés a célunk, úgy kódoljuk az információt, hogy bizonyítható legyen, hogy a hitelesített információt nem illetéktelen fél hozta létre, és a kódolt, hitelesített információt illetéktelen fél ne módosíthassa észrevétlenül. Hitelesítés esetén nem cél, hogy a támadó a kódolt információból ne nyerhesse vissza az eredeti információt. Sőt, hitelesítés esetén a kódolt információ gyakran közvetlenül tartalmazza az eredeti információt, a kódolás csupán „kiegészíti" az információt olyan redundáns elemekkel, amelyek az eredeti információ minden egyes bitjétől függenek, és amelyeket a támadó nem tud létrehozni.

Léteznek olyan hitelesítési módszerek, amelyek titkosságot is biztosítanak, de ez nem

feltétlenül előny. Ilyenkor ugyanis körülményes az üzenet hitelességének bizonyítása harmadik fél felé.

Megjegyzés: A titkosítás célja, hogy a kódolt, titkosított üzenetet elfogó támadó ne tudja megismerni annak tartalmát. Az üzenet bizalmasságát más módon is biztosíthatjuk, például elbújtathatjuk a bizalmasnak szánt üzenetet más, „ártatlan" üzenetek között. A szteganográfia az üzenetek elrejtésének tudománya, azt célozza meg, hogy a támadó ne is vegye észre, hogy érzékeny üzenettel áll szemben. [188], [155]

Szteganográfiára épülő megoldás például, ha egy aranyos kismacskát ábrázoló képfájlban a pixelek legalacsonyabb bitjeiben helyezzük el bizalmas üzenetünket, vagy ha egy salátatermesztésről szóló filmben egyes mozdulatok vagy szavak titkos jelentéssel is bírnak. Az ilyen képről vagy filmről csak a beavatottak veszik észre, hogy rejtett tartalommal is bír, így akár a támadó orra előtt is továbbítható.

A kriptográfiát és a szteganográfiát gyakran együttesen használják: az információt először titkosítják, majd a titkosított információt rejtik el, így még nehezebb kimutatni az elrejtett információt, illetve nehezebb visszafejteni, ha mégis megtalálják azt. Könyvünkben nem foglalkozunk szteganográfiával, hanem feltételezzük, hogy a nyilvános csatornán küldött üzeneteinket a támadó felismerheti, elfoghatja. Az érzékeny üzenetek védelmét könyvünkben kizárólag a kriptográfiára bízzuk, de a gyakorlaban az üzenetek elrejtésével további védelem is biztosítható lehet.

2.10. A kriptográfia története, vázlatosan

A kriptográfia története az ókorig, szinte az első írásos emlékekig vezethető vissza. [171], [158]

Az első titkosítási módszerek egyike a spártaiak által használt szkütalé. A titkosított levél írása előtt egy vékony bőrszalagot feltekertek egy henger alakú botra, az úgynevezett szkütaléra, és a levelet a szkütaléra tekert bőrszalagra írták. A szalagot ezt követően letekerték, így a botra írt betűk a szalagon összekeveredtek, és a szalagot a bot nélkül küldték el. A címzettnek is ugyanolyan botja (szkütaléja) volt, amelyre feltekerte az üzenetet, és így el tudta olvasni. E megoldás egy permutációs rejtjelező, ahol a szkütalé, illetve annak átmérője volt a „kulcs", az határozta meg a permutációt.

Julius Caesar nevéhez is fűződik egy titkosírás. Caesar a tábornokaival való levelezését olyan módon titkosította, hogy az ABC-t 3 karakterrel eltolta; A betű helyett D-t írt, B betű helyett E-t stb. Később Augustus megreformálta e monoalfabetikus titkosírást, ő csak egy betűvel tolta el az ABC-t (A helyett B, B helyett C stb); a mai szóhasználattal élve „kulcsot cserélt".

Az ókori és középkori titkosírások javarészt egyedi trükkökre alapultak. A titkosírások megfejtése pedig fokozatosan alakult át az ad hoc próbálkozásokból a szöveg karakterei előfordulási gyakoriságának analízisébe, majd matematikai statisztikai módszerekbe.

Az első világháború során használtak először one-time-pad-et, amelyről Shannon később, 1949-ben bebizonyította, hogy tökéletes titkosítást jelent. [168] Szintén Shannon nevéhez fűződik az a sejtés, hogy bitek felcserélésének (permutáció) és egyszerű helyettesítések sorozatával „erős" rejtjelező konstruálható. (Ilyen elemekből épül fel a DES és az AES is.)

A második világháború során különös hangsúlyt kapott a titkosírások matematikai analízise, illetve a titkosítások gépesítése. A számítógépek megjelenésével és elterjedésével e tendencia tovább folytatódott. Az Internet megjelenése óta már bárki könnyen hozzájuthat nagyon erős, jó minőségű titkosító eszközökhöz (szoftverekhez). Egy időben export-korlátozásokkal próbáltak gátat vetni a kriptográfiai módszerek kiszivárgásának, de ez nem sikerült.

A szimmetrikus kulcsú DES (data encryption standard) algoritmust az 1970-es években fejlesztették ki, és 1976-ban publikálták. A DES volt az egyik első nyitás a katonai alkalmazások irányából a polgári felhasználás felé. A DES szabványosítása során az USA nemzetbiztonsági hivatala megosztotta, nyilvánosságra hozta tudásának egy részét, ezzel egy erős, nyilvános algoritmus jelent meg a polgári felhasználásban. Bár a DES működése nyilvános, azt nem publikálták, hogy miért pont úgy működik. Az akadémiai szféra kriptográfiai kutatásainak egy jelentős része sokáig arra irányult, hogy vajon milyen elvek szerint konstruálták a DES-t, milyen további, nem nyilvános információkkal rendelkeztek a DES tervezői.

A nyilvános kulcsú kriptográfia alapgondolatát Diffie és Hellman publikálta 1976-ban. [32] Az RSA, az első, gyakorlatban is használható, aláírásra, titkosításra és autentikációra is alkalmas nyilvános kulcsú kriptográfiai algoritmus 1978-ban jelent meg. [156] A nyilvános kulcsú kriptográfia megjelenésével óriási fejlődésnek indult a kriptográfia polgári alkalmazása. A korábban használt, szimmetrikus kulcsú rendszerek leginkább katonai környezetben működtek, ahol egy zárt szervezeten belül biztonságosan meg lehetett valósítani a szimmetrikus kulcsok generálását, kiosztását és kezelését. Az egymástól független, egymással üzletelő vállalatok közötti biztonságos kommunikációt nem lehetett nagy tömegben szimmetrikus kulcsú alapon szervezni.

A nyilvános kulcsú kriptográfia tette lehetővé a bárki által ellenőrizhető elektronikus aláírás megjelenését is.

Az elektronikus aláírásról szóló EU irányelv 1999-es megjelenésével e technológiát a jog is elismeri, mint hitelesítésre szolgáló, a kézzel írott aláírással egyenértékű megoldást. [62] Korábban is lehetett szerződést kötni kriptográfiai módszerekkel (sőt, akár azok nélkül is), de akkor ezek egyedi megállapodások alapján történhettek.

2.11. Összegzés

- A Kerckhoffs feltétel értelmében egy „erős” kriptográfiai algoritmus akkor is biztonságos, ha a támadó minden részletét ismeri, kivéve egy bizonyos paramétert, az úgynevezett kulcsot, amely könnyen és gyorsan cserélhető. Általánosan elfogadott, hogy a kriptográfiai algoritmusokat nyilvánosságra hozzák.
- Ha a kódolás és a dekódolás egyazon kulccsal történik, szimmetrikus kulcsú kriptográfiáról beszélünk. Ha kódolásra és dekódolásra eltérő kulcsot használunk, nyilvános kulcsú kriptográfiáról beszélünk (mert ekkor az egyik kulcs nyilvánosságra is hozható).
- A szimmetrikus kulcsú módszerek gyorsabbak, de az egymással kommunikáló feleknek előzetesen titkos csatornán közös szimmetrikus kulcsban kell megállapodniuk. A nyilvános kulcsú módszerek bár lassabbak, de hozzájuk nincsen szükség előzetesen megbeszélt közös titkokra.
- A két módszert kombinálni szokás, például úgy, hogy a kulcscsere nyilvános kulcsú alapon, a kommunikáció szimmetrikus kulcsú alapon történik.
- Kizárólag nyilvános kulcsú alapon lehet „aláírást” készíteni, amely harmadik fél számára is bizonyítékot jelent. Ilyenkor az aláírandó dokumentumnak ún. lenyomatát (hash-ét) kódoljuk a magánkulccsal.
- A gyakorlatban használt kriptográfiai megoldások feltételes biztonságot jelentenek, azaz feltételezzük, hogy a támadó valamilyen korlátos erőforráskészlettel rendelkezik. Az erős megoldásokat reális erőforrásokkal csak elhanyagolható valószínűséggel lehet sikeresen támadni.

A továbbiakban a kriptográfiai algoritmusokat „fekete dobozként” kezeljük, azaz az algoritmusok működésének részleteivel nem foglalkozunk, és – leszámítva a hosszú távú archiválásról szóló fejezetet (8. fejezet) – azzal a feltételezéssel élünk, hogy a támadó nem tudja feltörni az „erős” kriptográfiai algoritmusokat.

3. fejezet

Tanúsítvány

„To whomsoever gives me the other half of any of these coins, I will grant one favor."
(Bárki is adja át nekem e félbetört pénzérmék másik felét, teljesítem egy kívánságát.)

– *Dirk Struan hagyatéka James Clavell „A Nemesi Ház" című művében*

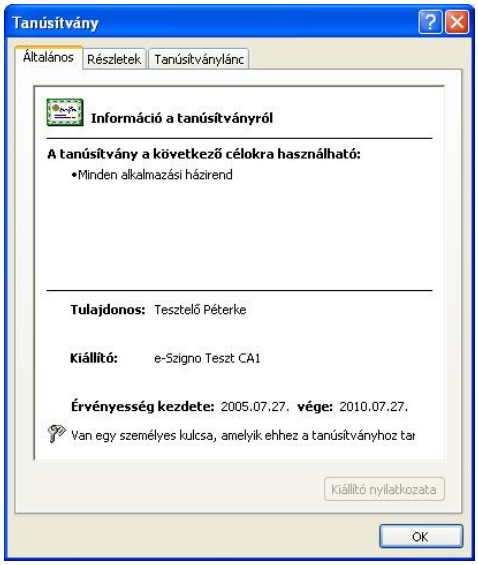
3.1. Mit nevezünk tanúsítványnak?

Tanúsítványt megbízható, bevizsgált szervezet, úgynevezett hitelesítés-szolgáltató állít ki emberek vagy számítógépek részére. A tanúsítvány tartalmazza annak a megnevezését, akinek a számára a tanúsítványt kiállították, és emellett olyan információkat tartalmaz, amely segítségével mások biztonságosan, azaz titkosan vagy hitelesen kommunikálhatnak a tanúsítvány birtokosával. A hitelesítés-szolgáltató aláírja a tanúsítványt, így igazolja a benne foglaltak hitelességét.

Műszaki szempontból *a tanúsítvány egy megbízható fél által aláírt igazolás, miszerint egy nyilvános kulcs egy adott entitáshoz tartozik.* (Lásd: 3.1. ábra.)

Az aláírásra szolgáló tanúsítványok alapján az elektronikus aláírásról szóló törvény által is elismert, fokozott biztonságú vagy minősített elektronikus aláírás hozható létre. Az ilyen tanúsítvány azt írja le, hogy egy adott személy aláírásait hogyan lehet ellenőrizni, így az aláírási címpéldány elektronikus megfelelőjének tekinthető.

Minden tanúsítványhoz kapcsolódik valamilyen titkos információ, amelyet kizárólag a tanúsítvány alanya, birtokosa (vagyis aki számára a tanúsítványt kiállították) ismer. Ezen információt nevezik *magánkulcs*nak, elektronikus aláírásra használható tanúsítvány esetén pedig aláírás-létrehozó adatnak. Ezen információ lehet egy fájl egy számítógépen (ekkor beszélünk szoftveres tanúsítványról), de lehet intelligens kártyán vagy más hardver eszközön is.



3.1. ábra. **A Windows XP így jelenít meg egy tanúsítványt**

A tanúsítványhoz tartozó magánkulcsot mindig a tanúsítvány alanya használja, míg a tanúsítványt bárki más, bármely *érintett fél* ellenőrizheti, aki a tanúsítvány alapján győződik meg róla, hogy a magánkulcs az alany birtokában van.

3.2. Tanúsítványok csoportosítása

A tanúsítványok nagyon sokféle módon csoportosíthatóak: funkciójuk szerint, az alany kiléte szerint, az alany megnevezése alapján, műszaki biztonsági szempontból vagy akár jogi szempontból is. Vizsgáljuk meg részletesebben e kérdéskört!

3.2.1. Funkció szerint

Megkülönböztetünk aláíró, titkosító és autentikációs (partner hitelesítésre szolgáló) tanúsítványokat.

- Az *aláíró tanúsítvány* alapján a tanúsítvány alanya – az aláíró – elektronikus aláírást hozhat létre. Az aláírást egészen pontosan a magánkulccsal hozza létre, a tanúsítvány az aláírás *ellenőrzéséhez* szükséges. Ugyanakkor az aláírás általában szorosan kötődik egy tanúsítványhoz, aláíráskor általában magát a tanúsítványt is aláírjuk, így igazoljuk, hogy pontosan milyen minőségünkben – milyen tanúsítvány szerint – készült az aláírás (6.4. fejezet).

  Ha az aláíró a magánkulcsával kódol egy dokumentumot, olyan bitsorozat jön létre, amelyről – az aláíró nyilvános kulcsa segítésével – bárki megállapíthatja, hogy azt a

tanúsítványban szereplő aláíró hozta létre, és az nem változott meg az aláírás óta[1].

Az aláírásra szolgáló tanúsítványt az elektronikus aláírásról szóló törvény így definiálja:

> *Eat. 2 § „ 21. Tanúsítvány: a hitelesítés-szolgáltató által kibocsátott igazolás, amely az aláírás-ellenőrző adatot a 9. § (3), illetőleg (4) bekezdése szerint egy meghatározott személyhez kapcsolja, és igazolja e személy személyazonosságát vagy valamely más tény fennállását, ideértve a hatósági (hivatali) jelleget. ”*

Az elektronikus aláírásnak és az aláíró tanúsítványoknak a későbbiekben egy külön fejezetet (6. fejezet) is szentelünk.

- A *titkosító tanúsítvány* segítségével titkosított üzeneteket küldhetünk a tanúsítvány alanyának: a tanúsítványban lévő nyilvános kulccsal *kódolunk egy üzenetet, amelyet a tanúsítvány alanya a magánkulcsával tud dekódolni.* Mivel a tanúsítvány alanya a magánkulcsot titokban tartja, a titkosított üzenetet rajta kívül más nem tudja elolvasni.

  A titkosításról és a titkosító tanúsítványokról a későbbiekben részletesen is szólunk (9. fejezet).

- Az *autentikációs tanúsítványok* segítségével hitelesíthetjük a tanúsítvány alanyát, azaz biztonságosan *meggyőződhetünk róla, hogy akivel kommunikálunk, az valóban birtokolja a tanúsítványban lévő nyilvános kulcshoz tartozó magánkulcsot.* Ez úgy történik, hogy egy véletlen számot küldünk a kommunikációs partnerünknek, aki a magánkulcsa segítségével kódolja azt. A tanúsítványban lévő nyilvános kulcs segítségével bizonyosodhatunk meg arról, hogy a kódolást valóban a magánkulcs segítségével végezték el. Ilyenkor általában biztonságos – titkosított és hitelesített – kapcsolatot is kiépíthetünk a tanúsítvány alanyával, és ha ezen keresztül adatokat küldünk neki, azt más nem tudja sem visszafejteni, sem észrevétlenül módosítani. Sokféle autentikációs tanúsítvány létezik, vannak közöttük webszerver (SSL szerver) tanúsítványok (10.4.1. fejezet), SSL kliens autentikációs tanúsítványok, VPN tanúsítványok, Windows bejelentkezésre használható tanúsítványok stb.

  Az autentikációs tanúsítványok használatát később külön fejezetben mutatjuk be (10. fejezet).

E három típusú tanúsítványt célszerű – sőt bizonyos esetekben kötelező – szigorúan szétválasztani egymástól. A szabványok elsősorban aláírás esetén hangsúlyozzák, hogy az aláíró kulcsot csak aláírásra használjuk, sőt az elektronikus aláírásról szóló törvény (13. § (4)) is kimondja, hogy „*az aláíró az aláírás létrehozó adatot kizárólag az aláírás létrehozására használhatja*”. Az autentikációt azért célszerű minden mástól elválasztani, mert ott a

[1] Az aláírás készítésének időpontját nem lehet önmagában az aláírás alapján meghatározni, ehhez pl. időbélyegre van szükség.

magánkulcs segítségével egy kívülről érkező „véletlen” blokkot kódolunk, így előfordulhatna, hogy egy cseles támadó autentikációkor aláírat vagy dekódoltat velünk egy üzenetet. A titkosítást szintén célszerű minden mástól elválasztani, ugyanis a dekódolásra szolgáló kulcsot egészen más módon, más életciklus szerint használjuk: Egyrészt a dekódoló kulcsot kulcsletét szolgáltatás keretében letétbe szokás helyezni egy megbízható szolgáltatónál (3.3.3. fejezet). Az aláírásra és autentikációra szolgáló kulcsokat nem szabad, nincs értelme letétbe helyezni (nem jelent nagy kárt, ha az aláíró vagy autentikáló magánkulcs megsemmisül, viszont letétbe helyezésük súlyos visszaélésre adhatna lehetőséget). Másrészt a dekódoló kulcsot a tanúsítvány lejártát követően is használhatjuk, míg aláíró és autentikációs tanúsítványok esetén ez tilos és értelmetlen.

Kimondhatjuk, aláírásra, titkosításra és autentikációra külön-külön kulcspárt kell használni, így e célra mindenkinek három külön kulcspárra és három külön tanúsítványra van szüksége.

A tanúsítvány funkciója a tanúsítványban feltüntetett kulcshasználat (Key Usage) alapján, illetve a tanúsítványra vonatkozó hitelesítési rend alapján állapítható meg.

3.2.2. Ki a tanúsítvány alanya?

A tanúsítványok aszerint is csoportosíthatóak, hogy ki a tanúsítvány alanya. A tanúsítvány alanya lehet:

- Természetes személy, azaz ember.
- Nem természetes személy, például szervezet, automata, számítógép, honlap stb.

3.2.3. Az elektronikus aláírásról szóló törvény szerint

Attól függően, hogy a tanúsítványhoz tartozó aláírásokhoz milyen joghatás kapcsolódik, illetve hogy a tanúsítványt kibocsátó hitelesítés-szolgáltató milyen garanciát vállal a tanúsítvánnyal kapcsolatban, léteznek minősített és nem minősített tanúsítványok.

Az elektronikus aláírásról szóló törvény alapján a következő tanúsítványokat különböztethetjük meg:

- *Minősített tanúsítvány*: A minősített tanúsítvány megfelel a legszigorúbb biztonsági követelményeknek, minősített tanúsítványt csak minősített hitelesítés-szolgáltató bocsáthat ki. Kizárólag természetes személy számára bocsátható ki minősített tanúsítvány, és kizárólag aláírásra használhatjuk őket, más célra nem szabad és nem is lehet őket felhasználni (kizárólag a letagadhatatlanságot jelentő kulcshasználati bit szerepel az ilyen tanúsítványban). Minősített elektronikus aláírás létrehozásához minősített tanúsítvány szükséges, valamint az aláírás-létrehozó adatot biztonságos aláírás-létrehozó eszközön (például egy megfelelő minősítéssel rendelkező intelligens

kártyán) kell tárolni, és az aláírást ezen eszköz segítségével kell létrehozni. A minősített elektronikus aláírással hitelesített dokumentum teljes bizonyító erejű magánokirat. Bizonyos szolgáltatások igénybevételéhez minősített elektronikus aláírásra van szükség. (Lásd: 6.1. fejezet.)

- *Nem minősített tanúsítvány*: Nem minősített tanúsítvány az elektronikus aláírásról szóló törvény szerint minden olyan tanúsítvány, amely aláírásra szolgál, és nem teljesülnek rá a minősített tanúsítványra vonatkozó előírások. Így a nem minősített tanúsítványokra a minősített tanúsítványoknál enyhébb követelmények vonatkoznak. A kibocsátásukra, használatukra kevésbé szigorú jogszabályi előírások vonatkoznak (pl. a nem minősített tanúsítványokat nem kötelező intelligens kártyán kibocsátani), és a beszerzésük, használatuk kevésbé körülményes. A nem minősített tanúsítvány alapján fokozott biztonságú elektronikus aláírás hozható létre, amelyhez nem tartozik a minősített aláíráshoz kapcsolódó joghatás. A fokozott biztonságú aláírásról az elektronikus aláírásról szóló törvény annyit állít, hogy megfelel az írásba foglalás követelményeinek.

  Megjegyzés: A jogszabályban nem szerepel olyan fogalom, hogy „fokozott biztonságú tanúsítvány", e kifejezést mégis sokan használják, és nem minősített tanúsítványt értenek rajta.

- *Nem aláírásra szolgáló tanúsítványok*: Az elektronikus aláírásról szóló törvény kizárólag aláírásra szolgáló tanúsítványokról szól, más célra használható – pl. titkosító, autentikációs (pl. webszerver) – tanúsítványok nem szerepelnek a törvényben. Ennek ellenére – tekintve, hogy ezek sem minősített tanúsítványok, őket is nevezik „nem minősített" tanúsítványoknak, és általában hasonló szabályokat alkalmaznak rájuk, mint a nem minősített tanúsítványokra.

3.2.4. Álnév vagy valódi név szerepel a tanúsítványban?

Attól függően, hogy egy tanúsítványban a hitelesítés-szolgáltató hogyan nevezi meg a tanúsítvány alanyát, léteznek álneves tanúsítványok és nem álneves tanúsítványok. Akkor nevezünk egy tanúsítványt álneves tanúsítványnak, ha a tanúsítványban nem a tanúsítványhoz tartozó felhasználó (alany/aláíró) valódi neve szerepel, hanem valamilyen más szöveg. Többféle nézet létezik azzal kapcsolatban, hogy ez pontosan mit jelent:

- Álneves tanúsítványban elvileg bármilyen szöveg szerepelhet a tanúsítvány alanya nevének a helyén. Szerepelhet ott az, hogy „Micimackó", de akár más személy neve is szerepelhet ott. Ekkor csak a hitelesítés-szolgáltató tudja, hogy valójában ki rejtőzik a tanúsítvány mögött.

- Az lehetne az álneves tanúsítványok funkciója, hogy a tanúsítványban nem a személy neve, hanem a szerepköre jelenik meg. Ez lehetne pl. „XY Kft, Ügyfélszolgálat”. Akkor lehetne ilyen tanúsítványokat használni, ha a funkció, szerepkör mögötti személy nevét nem szeretnénk nyilvánosságra hozni.
- Szigorú értelemben véve az is álneves tanúsítványnak minősül, amikor az alany/aláíró neve nem pontosan az igazolványában lévővel megegyező módon szerepel a tanúsítványban. Például az alany igazolványában az szerepel, hogy „Dr. Kovács János”, de a tanúsítványban az szerepel, hogy „Kovács János” („Dr”. nélkül) vagy az, hogy „Dr. Kovacs Janos” (ékezet nélkül).

Az elektronikus aláírásról szóló törvény kötelezővé teszi, hogy a hitelesítés-szolgáltatók álneves tanúsítványokat is bocsássanak ki, és a tanúsítványban fel kell tüntetni, hogy álneves tanúsítványról van szó. Több probléma is van az álneves tanúsítványokkal kapcsolatban. Egyrészt többféle gyakorlat létezik annak a megjelölésére, hogy egy tanúsítvány álneves. Másrészt a fenti esetek nehezen különíthetőek el egymástól, így nehéz megállapítani, hogy az álnév félrevezető-e.

> ***3.1. Példa:*** *Az e-Szignó Hitelesítés Szolgáltató az alany megnevezésének* `pseudonym` *mezejébe írja az álnevet. E megoldás egyértelmű, külföldiek számára és automatizmusok számára is értelmezhető.*
>
> *Van olyan hazai hitelesítés-szolgáltató, amely a név előtt feltüntetett „ALNEV” felirattal jelzi, hogy álneves tanúsítványról van szó. E megjelölés egyértelmű, de külföldi érintett fél nehezen tudja értelmezni.*
>
> *Van olyan hazai hitelesítés-szolgáltató, amely a név előtt és után feltüntetett „~” jellel jelöli, hogy a név álnév. E megoldás a hitelesítési rend alapos tanulmányozása nélkül nehezen ismerhető fel, és könnyen félrevezetheti az érintett felet.*
>
> *Volt olyan megoldás, hogy az alany megnevezésének* `title` *mezejében szerepelt az az utalás, hogy álneves tanúsítványról van szó. Sajnos sok alkalmazás a* `title` *mezőt nem jeleníti meg, valamint külföldiek nehezen tudják értelmezni az ott szereplő magyar szöveget.*

Habár a tanúsítvány/aláírás joghatásának nincsen köze hozzá, hogy a tanúsítvány álneves-e vagy sem (minősített tanúsítvány is lehet álneves, és álneves minősített tanúsítvány alapján is teljes bizonyító erejű magánokirat hozható létre), az álneves tanúsítványok a gyakorlatban nagyon nehezen használhatóak, ezért nagyon kevés álneves tanúsítvány van Magyarországon. (Lásd: 3.2.4. fejezet.)

Sajnos az álneves tanúsítványok ugyanakkor jelentősen megnehezítik a többi, nem álneves tanúsítvány használatát is, mert egy tanúsítványról először el kell dönteni, hogy álneves-e (és ez nem egyszerű feladat), az álneves tanúsítványok pedig akár teljesen félrevezető

információkat is tartalmazhatnak. Álneves tanúsítvány esetén sajnos – pusztán a tanúsítvány alapján – semmilyen támpontunk nem lehet arról, hogy a tanúsítvány valójában kihez is tartozik. Ezért a hitelesítés-szolgáltatók (akik a törvény szerint kötelesek álneves tanúsítványt is kiadni) markánsan elkülönítik az álneves tanúsítványokat a nem álnevesektől.

> ***3.2. Példa:*** *Az e-Szignó Hitelesítés Szolgáltató a nem álneves tanúsítványok esetén mindig az alany/aláíró igazolványában szereplő írásmóddal, ékezethelyesen szerepelteti a nevet a tanúsítványban. Az álneves tanúsítványokat külön hitelesítő egységgel, külön kulcspárral bocsátja ki, és ezzel a kulcspárral nem bocsát ki nem álneves tanúsítványt. Az álneves tanúsítványokban a* `pseudonym` *mezőben szerepelteti az álnevet, a* `common name` *mezőben pedig az „álneves tanúsítvány" szöveget tünteti fel.*

3.3. Tanúsítványok életciklusa

3.3.1. Tanúsítványigénylés

Ha tanúsítványt szeretnénk, fel kell keresni egy hitelesítés-szolgáltatót. Első lépésként tudatnunk kell a hitelesítés-szolgáltatóval, hogy pontosan milyen tanúsítványt szeretnénk. Ezután megadjuk adatainkat a hitelesítés-szolgáltatónak, és felhatalmazzuk a hitelesítés szolgáltatót, hogy adatainkat a tanúsítvány kibocsátása céljából kezelje. Ez a folyamat általában a hitelesítés-szolgáltató honlapján keresztül történik.

Az elektronikus aláírásról szóló törvény szerint a hitelesítés-szolgáltatónak tájékoztatnia kell az igénybe vevőt a szolgáltatással kapcsolatban:

> *Eat. „ 9. § (1) A szolgáltató a szerződéskötést megelőzően köteles tájékoztatni az igénybe vevőt a szolgáltatás felhasználásának módjáról, biztonsági fokáról, amennyiben a szolgáltató rendelkezik önkéntes akkreditációs rendszer keretében szerzett – különösen a szervezetének, rendszerének, valamint a szolgáltatás során alkalmazott termékeknek és hálózatnak a külön jogszabály szerinti informatikai biztonsági követelményeknek, vagy más önként vállalt követelményeknek való megfelelését igazoló – tanúsítással, úgy erről a tényről, továbbá szolgáltatási szabályzatáról, valamint a szerződés feltételeiről, különösen a (2) bekezdés szerinti korlátozásokról. Amennyiben a szolgáltatás működésének megkezdését követően a 20. § (3) bekezdésében foglalt vizsgálat még nem zárult le, a szolgáltató köteles erről az igénybe vevőt tájékoztatni. "*

E tájékoztatást általában a szolgáltató honlapja, illetve szabályzatai tartalmazzák, és az igénylés során csak úgy léphetünk tovább, ha beikszeljük, hogy az adott tájékoztatást

megkaptuk. Az önkéntes akkreditációs rendszer fogalmát az EU irányelv vezette be, Magyarországon nem működik ilyen, de erről tájékoztatni kell az igénylőt.

3.3.2. Kulcspár generálása

A tanúsítvány – többek között – azt igazolja, hogy a benne foglalt nyilvános kulcs egy adott végfelhasználóhoz (személyhez, szervezethez, eszközhöz) tartozik, és a hozzá tartozó magánkulcs kizárólag ezen entitás birtokában van.

Ki hozza létre e kulcsokat? Milyen módon jönnek létre? Miért és mennyire biztos abban a hitelesítés-szolgáltató, hogy a magánkulcs valóban a tanúsítvány alanyánál van? E kérdésekre többféle válasz adható, de a válasz markánsan meghatározza a tanúsítvány nyújtotta biztonságot.

A tanúsítványhoz kapcsolódó kulcspárt vagy a hitelesítés-szolgáltató, vagy az ügyfél generálja. Ez utóbbi esetben generálhatja a kulcsot maga a leendő tanúsítvány alanya, esetleg annak szervezete (pl. munkahelye) vagy ezen szervezet rendszergazdája. Ha a kulcspárt a szolgáltató generálja, a magánkulcsot biztonságos módon – titkosan és hitelesen – kell eljuttatnia a tanúsítvány alanyához, illetve biztosítania kell, hogy a magánkulcs minden példányát átadja az alanynak, nem őrzi meg, és nem is továbbítja jogosulatlanul harmadik félnek (még rendőrségnek, titkosszolgálatnak sem) a magánkulcsot. (Ez alól titkosító tanúsítványok esetén az ún. kulcsletét szolgáltatás jelenthet kivételt (3.3.3. fejezet).) A nyilvános kulcs eljuttatása ekkor kevésbé problémás, hiszen a szolgáltató a tanúsítvány kibocsátását követően jellemzően úgyis nyilvánosságra hozza. Ha a kulcspárt az ügyfél generálja, akkor viszont a nyilvános kulcsot kell biztonságos módon – hitelesen – eljuttatnia a hitelesítés-szolgáltatóhoz. Ekkor különösen fontos a nyilvános kulcs hitelessége, mert ha egy támadó kicseréli az igénylő nyilvános kulcsát a saját nyilvános kulcsára (azaz egy olyan nyilvános kulcsra, amelynek magánkulcsa a támadó birtokában van), a támadó olyan tanúsítványhoz juthat, amely alapján később aláírhat az eredeti igénylő nevében, elolvashat neki szóló titkosított üzeneteket, vagy sikeresen adhatja ki magát az igénylőnek a hálózaton keresztül.

Bárki generálja a kulcspárt, annak megfelelően biztonságos módon kell készülnie. Szabványok és jogszabályok határozzák meg, hogy hogyan kell biztonságos módon jó minőségű kulcsokat generálni. [57] A kulcsot véletlenszerűen kell kiválasztani, de előfordulhat, hogy egyes így kapott kulcsok „gyengébbek” a többinél, ami azt jelenti, hogy olyan támadások léteznek a szakirodalomban, amelyek e kulcsok esetén jelentősen hatékonyabbak. A kulcsgenerálást leíró specifikációk elsősorban jó minőségű véletlen-forrást, bizonyos gyenge kulcsok kerülését írják elő, illetve kimondják, hogy a kulcsot a generálást követően biztonságosan kell kezelni. Kriptográfiai értelemben a kulcs jellemzően jobb minőségű, ha a hitelesítés-szolgáltató állítja elő, mert a szolgáltató e szabványok szerint kell eljárjon, és birtokában kell lennie a biztonságos kulcsok generálásához szükséges szaktudásnak. A szolgáltató általában bevizsgált, tanúsított

kriptográfiai hardver eszközzel állítja elő az érzékeny kulcsokat, így azok garantáltan jó minőségűek lesznek. Ha a kulcspárt az ügyfél generálja, a hitelesítés-szolgáltatónak nincs ráhatása a kulcsgenerálás folyamatára. Ekkor problémát jelenthet, ha az ügyfél nem elég jó minőségű kulcsot generál. Lehet, hogy az ügyfél nem megfelelő minőségű véletlenszám-forrást használt a kulcsgeneráláshoz. Lehet, hogy az ügyfél nem jó algoritmussal generálja a kulcsot, és így gyenge kulcsot választ. Az ügyfél általában valamilyen szoftver segítségével hozza létre a kulcsot. Előfordulhat, hogy a szoftver fejlesztői nem voltak elég körültekintőek, és nem zártak ki ismert gyenge kulcsokat. Az is lehet, hogy az adott szoftver nem olyan céllal készült, hogy az ügyfél által elvárt szintű biztonsággal generáljon kulcsot. Sőt az is lehet, hogy az ügyfél által használt szoftver hibás.

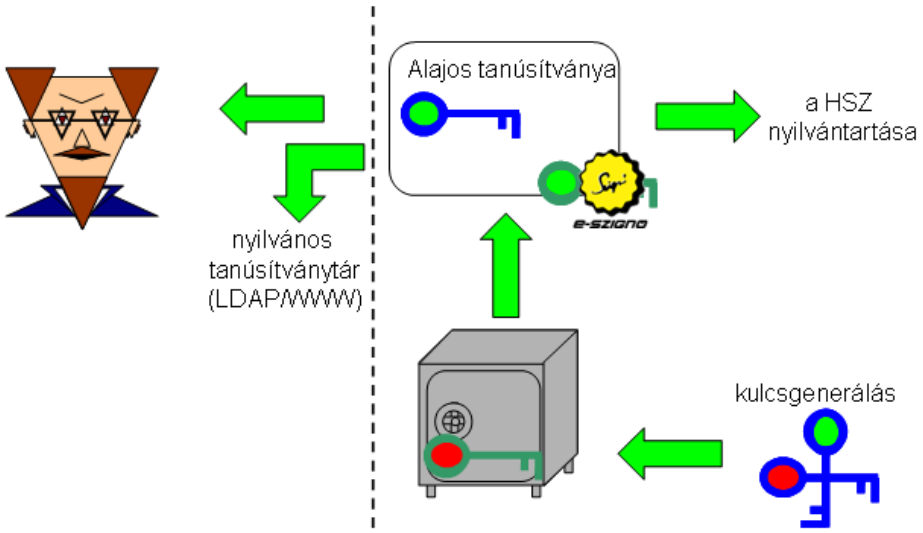
> ***3.3. Példa:*** *A közelmúltban történt az az eset, hogy a Debian Linuxban használt OpenSSL véletlenszám-generátora hibás volt, és mindössze néhány ezer különböző véletlenszámot tudott generálni. Ebből adódóan, ha valaki tudta, hogy egy adott kulcsot egy bizonyos időszakban a Debian Linux alatti OpenSSL-lel generáltak, nagyon könnyen meg tudta fejteni a magánkulcsot: könnyűszerrel le tudta generálni az összes lehetséges kulcspárt. [34]*

Ha a kulcspárt a hitelesítés-szolgáltató generálja, akkor a kulcspár kriptográfiai értelemben jobb minőségű lehet, de a szolgáltatónak meg kell győznie az ügyfelet, hogy nem őrzi meg jogosulatlanul a magánkulcsot. Nagyobb bizalmat kelt, ha kulcspárt az ügyfél generálja, mert a magánkulcs ekkor el sem jut a szolgáltatóhoz, de ekkor az ügyfél felelőssége, hogy a kulcspár megfelelően jó minőségű legyen.

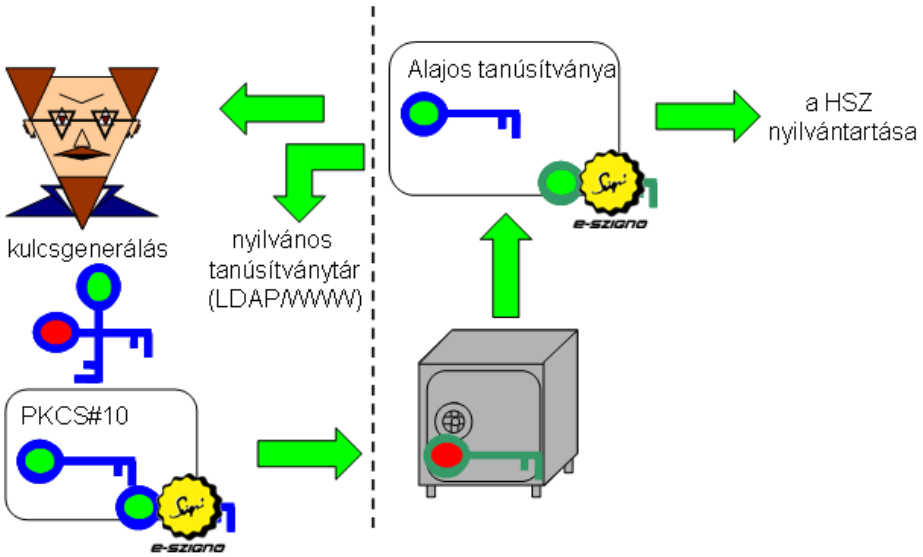
A következő esetek szoktak előfordulni a gyakorlatban:

- *A hitelesítés-szolgáltató generálja a kulcspárt* biztonságos aláírás-létrehozó eszköz (BALE) segítségével; ezen eszköz bevizsgált, tanúsított termék, arra szolgál, hogy a magánkulcs a teljes életciklusát BALE-n, védett környezetben élje le. (lásd: 3.2. ábra)

  Megkülönböztethetünk 1-es, 2-es és 3-as típusú BALE-t. Az 1-es típusú BALE csak kulcsgenerálásra szolgál, és a rajta generált kulcs – egy megfelelően biztonságos csatornán – áttölthető egy 2-es típusú eszközre. Ha a magánkulcs átkerült a 2-es típusú eszközre, akkor az 1-es típusú BALE megsemmisíti a magánkulcsot. Az 1-es típusú BALE nem személyhez kötött eszköz, jellemzően egyazon 1-es típusú BALE sok felhasználónak generál kulcsot (az 1-es típusú BALE lehet pl. egy HSM egy hitelesítés-szolgáltatónál). A 2-es típusú BALE már személyhez kötött eszköz, aláírások készítésére szolgál (a 2-es típusú BALE lehet pl. egy intelligens kártya). Ha a felhasználó megad egy PIN kódot, a 2-es típusú BALE aláírást készít a magánkulccsal, de a BALE védelmi mechanizmusai biztosítják, hogy a magánkulcs soha nem nyerhető ki belőle. A 3-as típusú BALE egyaránt szolgál kulcsgenerálásra és aláírásra; a 3-as típusú BALE-ból



3.2. ábra. **A kulcspárt a hitelesítés-szolgáltató generálja, így biztos lehet benne, hogy rajta kívül más nem birtokolja a magánkulcsot. A szolgáltató aláírja és eltárolja a kibocsátott tanúsítványt, valamint – a magánkulccsal együtt – átadja az alanynak, és esetleg nyilvánosságra hozza.**



3.3. ábra. **A kulcspárt az igénylő generálja, a szolgáltató csak a nyilvános kulcsot tartalmazó, aláírt, PKCS#10 formátumú kérelmet kapja meg. A rajta lévő aláírásból bizonyosodik meg róla, hogy az igénylő valóban birtokolja a magánkulcsot. A szolgáltató aláírja és eltárolja a kibocsátott tanúsítványt, valamint átadja az alanynak, és esetleg nyilvánosságra hozza.**

sem lehet kinyerni a magánkulcsot. A kulcspárt ő maga generálja, és a magánkulcs a teljes életciklusát egyazon eszközön éli le. [29]

Tekintve, hogy egy aláírókulcs kizárólag egyetlen BALE-n létezhet, azt követően, hogy a hitelesítés-szolgáltató átadja a (2-es vagy 3-as típusú) BALE-t az aláírónak, a szolgáltatónál garantáltan nem marad példány a kulcsból.

Ehhez hasonló megoldást gyakran használnak más, nem BALE eszközökkel is. A kulcsot ekkor is pl. egy intelligens kártyán generálják (vagy biztonságos környezetben, biztonságos módon töltik fel egy intelligens kártyára), és a kártyával együtt adják át a kulcsot a végfelhasználónak. Szintén előfordulhat, hogy az eszköz BALE ugyan, de nem BALE-ként alkalmazzák. A BALE fogalom csak az elektronikus aláírás kontextusában értelmezhető, így csak aláíró tanúsítványok esetén van értelme. Titkosító vagy autentikációs tanúsítvány magánkulcsával kapcsolatban a BALE fogalom nemigen értelmezhető, egy eszköz BALE tanúsítása egyáltalán nem biztos, hogy jelentőssséggel bír a titkosító vagy autentikációs tanúsítványok esetére.

Megjegyezzük, hogy az elektronikus aláírásról szóló törvény szerint az aláírás-létrehozó eszköz kibocsátása (és így esetünkben a kulcsgenerálás) ún. eszköz-szolgáltatás, és eszköz-szolgáltatók végzik, de a gyakorlatban ezek egybeesnek a hitelesítés-szolgáltatókkal.

- *A kulcspárt az ügyfél generálja*, és a tanúsítványigényléssel együtt – pl. egyazon SSL csatornán – juttatja el a hitelesítés-szolgáltatónak a nyilvános kulcsot. Az ügyfél által generált magánkulcs általában szoftveres; vagy egy fájl az ügyfél számítógépén, vagy az ügyfél böngészőprogramja tárolja. (lásd: 3.3. ábra)

  A nyilvános kulcs ekkor ún. PKCS#10 formátumban jut el a szolgáltatóhoz. A PKCS#10 a nyilvános kulcs mellett tartalmazza, hogy a tanúsítványban milyen adatok (DN) feltüntetését kéri az igénylő. Az igénylő a PKCS#10-es tanúsítványkérelemben szereplő nyilvános kulcshoz tartozó magánkulccsal írja alá a PKCS#10 tanúsítványkérelmet, így a PKCS#10-es kérelmen lévő aláírás az ugyanazon kérelemben szereplő nyilvános kulccsal ellenőrizhető. A PKCS#10-es kérelem mindössze azt igazolja, hogy valaki, akinek a birtokában volt az adott nyilvános kulcshoz tartozó magánkulcs, valamikor, valamely szolgáltatónál valóban igényelt tanúsítványt az adott DN-re. [133]

Szerencsétlen megoldás, ha a kulcspárt a szolgáltató kriptográfiai hardver eszköz nélkül generálja és juttatja el az ügyfélnek. Ekkor ugyanis csak a szolgáltató belső eljárásai biztosíthatják, hogy nem őrzi meg a magánkulcsot, más egyéb védelmi mechanizmus (pl. BALE) nem támogatja ezt.

3.3.3. Magánkulcs letétbe helyezése

Titkosító tanúsítványok – és csakis titkosító tanúsítványok – esetén előfordulhat, hogy a hitelesítés-szolgáltató mégis megőrzi a magánkulcsot; ez az ún. kulcsletét szolgáltatás esete.

A kulcsletét szolgáltatás azt jelenti, hogy a titkosító tanúsítvány magánkulcsát egy megbízható félre bízzuk, aki biztonságosan megőrzi a magánkulcsot, és csak arra jogosult feleknek adja át. Azért lehet szükség rá, mert a titkosításra szolgáló tanúsítványban szereplő nyilvános kulccsal értékes információt titkosíthatnak, és ha a magánkulcs megsemmisül – pl. mert a felhasználó elveszíti az intelligens kártyáját, vagy mert összeomlik a számítógépe – ezen érzékeny információk végérvényesen elveszhetnek. Tekintve, hogy a titkosító tanúsítvánnyal (illetve annak magánkulcsával) bármennyi, és bármilyen értékes információt titkosíthatunk, a magánkulcs elvesztése akár óriási károkat is okozhat. Ezt elkerülendő, a titkosító tanúsítvány magánkulcsát letétbe szokás helyezni egy megbízható félnél, hogy a magánkulcs mindenképpen elérhető legyen. E megbízható fél a kulcsletét szolgáltató.

A kulcsletét szolgáltató nem élhet vissza a magánkulccsal, és csak arra jogosult félnek adhatja ki. A kulcsletét szolgáltatóval kötött szerződés szabályozhatja, hogy a kulcsletét szolgáltató kinek adhatja ki a magánkulcsot. A tanúsítvány alanya jellemzően jogosult hozzáférni a magánkulcshoz, és gyakori, hogy a tanúsítványban feltüntetett szervezet, illetve a tanúsítványra előfizető fél is.

> ***3.4. Példa:*** *Bendegúz a Kókler Bt. alkalmazottja, munkája során titkosító tanúsítvánnyal titkosított levelezést folytat a Kókler Bt. üzleti partnereivel. Amikor elbocsátják a munkahelyéről, Bendegúz magával viszi a magánkulcsát (a munkahelyén lévő másolatokat letörli). Így a Kókler Bt. nem férhet hozzá, hogy alkalmazottja kikkel és milyen levelezést folytatott. Bendegúz zsarolni kezdi a munkahelyét...*

Kizárólag titkosító tanúsítványok esetén lehet szükség kulcsletét szolgáltatásra, mert kizárólag ekkor igaz, hogy a magánkulcs megsemmisülése korlátlan kárt is okozhat. Ha egy aláíró vagy egy autentikációs tanúsítvány magánkulcsa semmisül meg, az alany új kulcspárhoz új tanúsítványt kaphat, nem vesznek el az adatai, és a kár legfeljebb az új tanúsítvány kibocsátásának a költsége.

A magánkulcs letétbe helyezése egyfajta rendezett, szabályzott „kibúvót” jelent a PKI azon alapelve alól, hogy a magánkulcsát mindenki csak saját maga ismerheti. Ha a magánkulcs harmadik félnél – még akkor is, ha az egy megbízható harmadik fél – is jelen van, az mindenképpen kockázatot jelent. Aláíró és autentikációs tanúsítványok esetén a magánkulcs letétbe helyezése felesleges kockázatot jelentene, ezért értelmetlen, és általában tilos is.

Titkosító tanúsítványok esetén viszont kimondhatjuk, hogy kulcsletét szolgáltatás nélkül nemigen képzelhető el olyan felhasználás, ahol érdemi, „fontos” dokumentumokat védenek

titkosító tanúsítványokkal. A legtöbb végfelhasználó képtelen egyszerre titokban is tartani a magánkulcsát, és a titokban tartott magánkulcsot megvédeni a megsemmisüléstől. Titkosító tanúsítványok esetén mindenképpen szükség van valamilyen fajta kulcsletétre. Ezt végezheti a hitelesítés-szolgáltató, végezheti dedikált kulcsletét szolgáltató, végezheti a tanúsítványban feltüntetett szervezet vagy akár maga a tanúsítvány alanya is.

Magyarországon az elektronikus aláírásról szóló törvény kizárólag aláíró tanúsítványokról és kulcsokról szól, titkosító (és autentikációs) tanúsítványokra nem vonatkozik. Ebből kifolyólag hazánkban nincs szabályozás a PKI-hoz kapcsolódó kulcsletét szolgáltatásra sem. A hitelesítés-szolgáltatók saját maguk nyújthatnak kulcsletét szolgáltatást a titkosító tanúsítványokhoz.

A magyar közigazgatásban használható hitelesítési rendek titkosító tanúsítvány esetén előírják a kulcsletét szolgáltatás használatát. [83]

3.3.4. Regisztráció

Regisztrációnak azt nevezzük, amikor a hitelesítés-szolgáltató a tanúsítvány kibocsátása előtt megállapítja az igénylő (személy vagy szervezet) kilétét, személyazonosságát.

Tekintve, hogy a tanúsítvány azt igazolja, hogy egy adott nyilvános kulcs egy adott entitáshoz tartozik, a regisztráció biztonsága alapvetően meghatározza ezen összetartozás erősségét, vagyis azt, hogy mennyire bízhatunk egy adott tanúsítványban.

„Erős” tanúsítványok esetén *személyes regisztráció*t szokás alkalmazni. Ekkor a hitelesítés-szolgáltató képviselője regisztrációkor személyesen találkozik az igénylővel, és személyes azonosítás során, pl. a személyi igazolványa alapján győződik meg az illető kilétéről. (Magyarországon személyi igazolvány, új típusú jogosítvány vagy útlevél alapján szokás személyes regisztrációt végezni.) Minősített tanúsítványok esetén a személyes regisztráció követelmény.

Másik lehetőség az ún. *távoli regisztráció*, amikor a hitelesítés-szolgáltató képviselője nem találkozik az igénylővel, hanem valamilyen más módon győződik meg az illető kilétéről. Ennek többféle módja is lehet:

- az igénylő postán juttatja el okmányait (vagy azok másolatát);
- a szolgáltató csak az igénylő e-mail címét ellenőrzi (pl. küld neki egy e-mailt, amit az adott címről vissza kell küldenie);
- webszerver tanúsítványok esetén alkalmazott megoldás, hogy a hitelesítés-szolgáltató csak a tanúsítványba kerülő domain birtoklását ellenőrzi; ezek az ún. domain validated tanúsítványok (10.4.3.1. fejezet);
- a hitelesítés-szolgáltató harmadik fél (pl. az alany munkahelyének) állítása alapján bocsátja ki a tanúsítványt.

A személyes azonosítás gyakran a tanúsítványhoz kapcsolódó egyik legsúlyosabb költségtényezőt jelenti. Olyan kényelmetlenséget jelent az alanynak (és szervezetének), amelyet a hitelesítés-szolgáltató nemigen tud csökkenteni vagy átvállalni. Vagy az alanynak kell elmennie a hitelesítés-szolgáltató regisztrációs irodájába, vagy a hitelesítés-szolgáltató mobil regisztrációs egységének kell kiszállnia az alanyhoz. A hitelesítés-szolgáltató megteheti, hogy nem saját maga végzi a regisztrációt, hanem ún. regisztráló szervezetet vesz igénybe. Ekkor a hitelesítés-szolgáltató regisztrációs egységét egy külső szervezet helyettesítheti. Így szervezetileg is elválhatna egymástól a hitelesítés-szolgáltató, amely csak a számítástechnikai infrastruktúrát működteti, és az ügyfelekkel kapcsolatot tartó, a személyes találkozásokat lebonyolító, akár országos lefedettségű regisztrációs szervezet. A tanúsítványt ellenőrző érintett fél a hitelesítés-szolgáltatóval kerül kapcsolatba, holott valójában a regisztráló szervezet határozza meg, hogy a hitelesítés-szolgáltató kinek a számára bocsátja ki a tanúsítványt. Így logikus, hogy a *hitelesítés-szolgáltató felel a regisztrációs szervezet tevékenységéért.* Tekintve, hogy egy aláíró tanúsítvánnyal óriási kárt lehet okozni, és a magyar jogszabályok szerint a hitelesítés-szolgáltatóra akár egyetlen regisztrációs hiba miatt is óriási kártérítés hárítható, a hitelesítés-szolgáltatót és a regisztrációs szervezetet szétválasztó modell nem terjedt el Magyarországon. Nálunk az az általános megoldás, hogy a regisztrációt a hitelesítés-szolgáltatók saját maguk végzik.

Korábban volt olyan kezdeményezés, hogy a regisztrációt közjegyzők végezzék, de erről kiderült, hogy nem jogszerű. A közjegyző független fél, így regisztráció során nem képviselheti a hitelesítés-szolgáltatót, és a hitelesítés-szolgáltató nem ellenőrizheti a közjegyző tevékenységét. [119] Így annak ellenére, hogy a közjegyzői regisztráció minden fél számára nagyobb biztonságot jelentene, Magyarországon közjegyző nem végezhet regisztrációt. (A hitelesítés-szolgáltató támaszkodhat közjegyzőkre a regisztráció során, de a közjegyző előtti megjelenés nem helyettesítheti a személyes találkozást a hitelesítés-szolgáltató munkatársával.)

Bár hivatalosan nem kapcsolódik a regisztrációhoz, de a hitelesítés-szolgáltató ekkor határozza meg, hogy pontosan milyen adatok kerülnek a tanúsítványba. Ökölszabály, hogy a hitelesítés-szolgáltatónak minden adatot ellenőriznie kell, amit a tanúsítványban feltüntet. Ellenőriznie kell az igénylő nevét, e-mail címét, ha a tanúsítványban szervezet is szerepel, a szervezet nevét, illetve, hogy a szervezet felhatalmazta-e az igénylőt, hogy a tanúsítványában szerepeljen. Ha az igénylő bármilyen más adat feltüntetését kéri a tanúsítványban (beosztás, végzettség, igazolvány száma stb.), a hitelesítés-szolgáltatónak erről meg kell győződnie. Az elektronikus aláírásról szóló törvény előírja (12. § (2)), hogy a hitelesítés-szolgáltatónak adategyeztetést kell végeznie a személyi adat- és lakcímnyilvántartással, az úti okmány nyilvántartással, gépjárművezetői nyilvántartással, illetve az aláírási jogosultság ellenőrzése céljából a cégnyilvántartással. [180]

3.3.5. Tanúsítvány kibocsátás

A 3/2005. IHM rendelet szerint (2. § k) a tanúsítvány kibocsátása az a pillanat, amikor a hitelesítés-szolgáltató átadja a tanúsítványt az aláírónak, vagy amikor közzéteszi a tanúsítványt a nyilvános tanúsítványtárban. Más szóval a jogszabály értelmében a tanúsítvány kibocsátása az a pillanat, amikor a tanúsítvány elhagyja a hitelesítés-szolgáltató rendszerét, és amikortól kezdve harmadik felek is kapcsolatba kerülhetnek a tanúsítvánnyal.

Műszaki, kriptográfiai szempontból a tanúsítvány kibocsátása kifejezés azt a pillanatot is jelentheti, amikor a hitelesítés-szolgáltató magánkulcsával aláírja a tanúsítványt, azaz azt a pillanatot, amikor a tanúsítvány létrejön.

A tanúsítvány kibocsátásakor a hitelesítés-szolgáltató nyilvánosságra hozza a tanúsítványt – ha az alany hozzájárul. Bár a nyilvános kulcsú infrastruktúra szerint a nyilvános kulcs nyilvános, az adatvédelmi jogszabályok értelmében nem nyilvánvaló, hogy a tanúsítvány automatikusan nyilvánosságra hozható volna. [179] Megjegyezzük, a tanúsítvány nyilvánosságra hozatalának egyedül titkosító tanúsítványok esetén van értelme, aláíró és autentikációs tanúsítványok esetén nincs. (12.4.6. fejezet)

3.3.6. A tanúsítvány és a magánkulcs használata

Az aláíró, a titkosító és az autentikációs tanúsítványok különböző módon használhatóak. Mindössze annyi a közös bennük, hogy a tanúsítvány alanya mindig a magánkulcsot használja, magára a tanúsítványra mindig egy harmadik félnek, az ún. érintett félnek van szüksége.

Aláíró tanúsítvány esetén a tanúsítvány meghatározza, hogy milyen bizonyító erő kapcsolódik az aláíráshoz. A titkosító és az autentikációs tanúsítványokhoz nem tartozik joghatás, esetükben a kriptográfiai eszközök csak technikai célokra szolgálnak.

3.3.6.1. Aláíró tanúsítványok

Aláírói tanúsítvány esetén a tanúsítvány alanya – az aláíró – a magánkulcs segítségével hozhat létre elektronikus aláírást. Ilyenkor a saját magánkulcsával kódolja egy dokumentum lenyomatát. Magára a tanúsítványra nincs szükség az aláírás készítésekor, ennek ellenére szokás a csatolni a tanúsítványt az aláíráshoz, így az érintett félnek könnyebb megkeresnie, hogy melyik tanúsítvánnyal kell ellenőrizni az aláírást. Így az egyes aláírás-formátumok (pl. PKCS#7, XMLDSIG) lehetőséget biztosítanak a tanúsítvány csatolására, más esetekben (pl. XAdES, CAdES) kötelező is csatolni a tanúsítványt (6.4. fejezet).

> **Megjegyzés:** Egy aláírás több tanúsítvánnyal is ellenőrizhető lehet, ha az egyes tanúsítványokban azonos nyilvános kulcs szerepel. Ez akkor jelenthet problémát, ha az azonos nyilvános kulcsot tartalmazó tanúsítványokban más adatok

szerepelnek az aláíróról, és ez befolyásolja az aláírás értelmét. Például egészen mást jelenthet az aláírás, ha az aláíró közjegyző vagy ha csak magánszemély. Ezt elkerülendő egyes aláírás formátumok előírják, hogy az aláíró egyúttal a saját tanúsítványát (illetve annak lenyomatát) is aláírja.

Az érintett fél az aláíró tanúsítványával (illetve a benne lévő nyilvános kulcs segítségével) tudja ellenőrizni az így kapott aláírást. A tanúsítvány nemcsak technikailag igazolja, hogy a nyilvános kulcs az aláíróhoz tartozik, de ez biztosítja, hogy az aláíráshoz az elektronikus aláírásról szóló törvényben meghatározott jogkövetkezmény kapcsolódjon. Az aláírás érvényességének feltétele, hogy az aláíró tanúsítványa az aláírás pillanatában érvényes legyen, így érvénytelen tanúsítvány esetén nincs értelme elektronikus aláírást készíteni.

Az aláírói tanúsítványnak nem feladata, hogy az aláíró kilétét megállapíthassuk. Az ún. álneves tanúsítványok éppen arra szolgálnak, hogy magából az aláírásból ne derülhessen ki, hogy ki készítette az aláírást. (Lásd: 12.4.8. fejezet.)

3.3.6.2. Titkosító tanúsítványok

Egy titkosító tanúsítvány (illetve a benne lévő nyilvános kulcs) segítségével egy érintett fél titkosított üzenetet küldhet a titkosító tanúsítvány alanyának, és az üzenetet az alany a saját magánkulcsa segítségével tudja visszafejteni.

Ekkor az érintett fél használja először tanúsítványt, és az alany csak ezt követően használja a magánkulcsot. Kérdés, hogy az érintett fél hogyan jut hozzá a tanúsítványhoz. Egyik lehetőség, hogy az alany elküldi neki. Másik lehetőség, hogy megkeresi a hitelesítés-szolgáltató nyilvános tanúsítványtárában. Ekkor honnan tudja, hogy melyik szolgáltatónál kell keresni? Ha több érvényes tanúsítványt talál, honnan tudja, hogy melyiket kell használni? A titkosító tanúsítványnak sem célja, hogy az alany kiléte megállapítható legyen. Ha több „Kovács János" névre szóló titkosító tanúsítványt talál, honnan tudja, hogy ő melyiknek szeretne titkosított levelet küldeni? E problémák nagy része ahhoz hasonlóan kezelhető, mint amikor e-mailt szeretnénk küldeni egy partnerünknek. Sok levelezőprogram szoftveres támogatást nyújthat a titkosító tanúsítvány kiválasztásában, például e-mail cím alapján a szolgáltatók LDAP címtárából elő tudják keresni a címzett titkosító tanúsítványát.

A titkosító tanúsítvány szerepe pusztán annyi, hogy az érintett fél megbizonyosodjon róla, hogy az adott nyilvános kulcs az adott alanyhoz, végfelhasználóhoz tartozik. Ezt követően a titkosító tanúsítványnak már nincsen szerepe.

Míg aláíró tanúsítvány esetén a biztonság szempontjából nem mindegy, hogy mely hitelesítés-szolgáltató, milyen feltételek mellett bocsátotta ki a tanúsítványt, a titkosító tanúsítványnak csak a nyilvános kulcs hiteles eljuttatásában van szerepe. Az aláíró tanúsítványának típusa határozza meg, hogy milyen jogkövetkezmény kapcsolódik az aláíráshoz, míg titkosító tanúsítvány esetén nincsen ilyen szerepe a tanúsítványnak. Amint hitelesen hozzájutottunk

partnerünk nyilvános kulcsához, üzeneteink titkosságát már csak a használt kriptográfiai algoritmusok befolyásolják, teljesen mindegy, hogy partnerünknek milyen tanúsítványa van, illetve van-e tanúsítványa.

3.3.6.3. Autentikációs tanúsítványok

A titkosító tanúsítványokhoz hasonlóan az autentikációs tanúsítványok sem tartoznak az elektronikus aláírásról szóló törvény hatálya alá. Így nem tartozik hozzájuk joghatás, és csupán műszaki szerepük van: segítségükkel az alany hitelesen juttathatja el nyilvános kulcsát partnereinek.

Az autentikációs tanúsítvány magánkulcsával az alany véletlen kihívást kódol, e kódolás technikailag nagyon hasonlít az aláírásra, de nem jelent beleegyezést egy dokumentum tartalmába.

Az autentikációs tanúsítvány egyúttal arra is szolgál, hogy a tanúsítvány alanyának partnerei meggyőződhessenek az alany kilétéről (és így pl. eldönthessék, hogy nyújtanak-e valamilyen szolgáltatást az alany számára). Ennek ellenére, az autentikációs tanúsítvány tartalma gyakran megegyezik az alany aláíró tanúsítványának tartalmával, és csak egy zárt közösség állapíthatja meg az alany kilétét a tanúsítványa alapján.

3.3.7. Tanúsítvány ellenőrzése

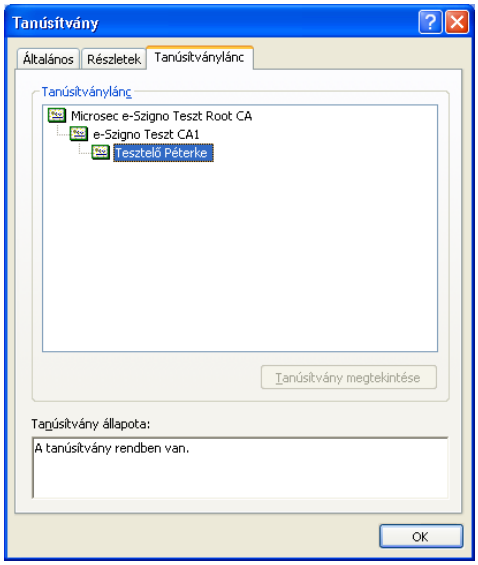
Ha hozzájutunk valakinek a tanúsítványához, biztonságos (titkosított, hitelesített) kapcsolatot létesíthetünk vele; így biztosak lehetünk benne, hogy amit küldünk, azt csak ő tudja elolvasni, és amit kapunk, azt valóban ő küldte, és az üzeneteit senki sem módosította útközben.

Mindez csak akkor teljesül, ha a kapcsolatot *érvényes* tanúsítvány alapján építjük ki. Egy tanúsítvány akkor érvényes, ha megbízható hitelesítés-szolgáltató állította ki, nem járt le, és nem vonták vissza. Egy tanúsítvány ellenőrzéséhez a következő lépéseket kell elvégeznünk (valamilyen számítógépes program, például az e-Szignó segítségével):

1. Meg kell bizonyosodnunk róla, hogy a tanúsítványt megbízható hitelesítés-szolgáltató bocsátotta ki. Ehhez kell találnunk egy olyan megbízható gyökértanúsítványt, amely alapján a kérdéses tanúsítványt ellenőrizhetjük, azaz fel tudunk építeni a kérdéses tanúsítványtól egy úgynevezett tanúsítványláncot ezen gyökértanúsítványhoz. (Lásd: 3.4. ábra.)

2. Ellenőriznünk kell, hogy a tanúsítványláncban egyik tanúsítvány sem járt-e még le.

3. Ellenőriznünk kell, hogy a tanúsítványláncban egyik tanúsítványt sem vonták-e vissza. A megbízható gyökértanúsítványt ilyenkor nem kell ellenőriznünk.



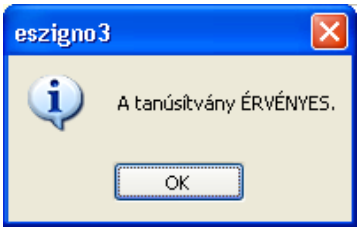
3.4. ábra. **A Windows XP így jeleníti meg a tanúsítványláncot. Az ábrán „Tesztelő Péterke" végfelhasználói tanúsítványához tartozó lánc látható. „Tesztelő Péterke" tanúsítványa az őt kibocsátó „e-Szigno Teszt CA1" tanúsítványa alapján ellenőrizhető, ezen utóbbi tanúsítvány pedig az őt kibocsátó „Microsec e-Szigno Teszt Root CA" tanúsítványa alapján ellenőrizhető. Esetünkben a „Microsec e-Szigno Teszt Root CA" tanúsítványa a gyökértanúsítvány.**

Ha egy tanúsítványhoz tartozó magánkulcs elvész, vagy illetéktelen kezekbe kerül, arról tájékoztatni kell a tanúsítványt kibocsátó hitelesítés-szolgáltatót, aki visszavonja a magánkulcshoz tartozó tanúsítványt. Ilyenkor a hitelesítés-szolgáltató közzéteszi, hogy az adott tanúsítvány érvénytelen. Ezt szabványos, számítógépek által is értelmezhető módon – például CRL, azaz visszavonási lista (4.1.5.1. fejezet) vagy OCSP, azaz online tanúsítvány-állapot protokoll (4.1.5.2. fejezet) segítségével – teszi közzé, így ha valaki mégis használni szeretné a tanúsítványt, az általa használt számítógépes program jelzi[2] neki, ha a tanúsítvány érvénytelen. Így a visszavont tanúsítványt visszavonást követően már nem lehet használni, a visszavont tanúsítvánnyal nem lehet visszaélni.

A fenti lépéseket nem nekünk kell elvégeznünk, hanem valamely aláírás-ellenőrző alkalmazás – például az e-Szignó program – végzi el helyettünk. Így e bonyolult folyamatból az ember általában csak annyit lát, hogy a program kiírja, hogy a tanúsítvány érvényes. (Lásd: 3.5. ábra.)

A fenti módszer akkor igaz, ha a tanúsítvány érvényességét az ellenőrzés időpontjára nézve vizsgáljuk. Ez titkosító tanúsítványok és autentikációs tanúsítványok esetén jellemző, aláíró

[2]feltéve, hogy a program megfelelően működik, és helyesen van beállítva



3.5. ábra. **Az e-Szignó program jelzi, hogy a tanúsítvány érvényes**

tanúsítványok esetén az ellenőrzést általában egy múltbéli időpontra, az aláírás időpontjára nézve végezzük. Ebből következően az aláírások ellenőrzése sokkal összetettebb folyamat, e kérdéskörrel később részletesen is foglalkozunk (6.5. fejezet).

3.3.8. Tanúsítványcsere

Tanúsítványcseréről akkor beszélünk, ha egy a regisztráción átesett alany meglévő tanúsítványa helyett új tanúsítványt igényel.

Attól függően, hogy a tanúsítványcserére miért kerül sor, a tanúsítványcseréhez különböző folyamatok tartozhatnak:

- Ha az alany meglévő tanúsítványa még érvényes, de hamarosan le fog járni, és az ügyfél olyan tanúsítványt igényel, amelybe az alany korábbi tanúsítványában lévőkkel megegyező adatok kerülnek, és a két tanúsítvány ugyanazon nyilvános kulcshoz kerül kibocsátásra, akkor *megújításról* beszélünk.
- Ha a Szolgáltató az új tanúsítványt új nyilvános kulcshoz bocsátja ki, a folyamatot *kulcscserének* nevezzük. Kulcscserére jellemzően akkor kerül sor, ha az alany tanúsítványa már nem érvényes (például ha visszavonásra került), de még érvényes tanúsítvány esetén is történhet kulcscsere (például ha a régi kulcsok mérete már nem megfelelő).
- Ha az alany régi tanúsítványa még érvényes, de az alany a tanúsítványban szereplő adatok megváltoztatását kéri, *adatváltozásról* beszélünk. Adatváltozás esetén a Szolgáltató ellenőrzi az új adatok helyességét.

Tanúsítványcsere esetén az alanynak, illetve a tanúsítványban szereplő szervezetnek általában ismételten nyilatkoznia kell, hogy a tanúsítványba kerülő adatok továbbra is helyesek, érvényesek.

3.3.9. A tanúsítvány visszavonása vagy felfüggesztése

Kezdetben, mikor a hitelesítés-szolgáltató kibocsátja a tanúsítványt, az *érvényes* állapotban van, míg az érvényességi ideje le nem jár, mert akkor érvényét veszti. Az érvényességi ideje alatt

a tanúsítványhoz tartozó magánkulcs illetéktelen kezekben kerülhet (kompromittálódhat), ekkor a tanúsítvány visszavont vagy felfüggesztett állapotba kerülhet. Kicsit részletesebben:

- *Érvényes tanúsítvány*: A tanúsítvány akkor érvényes, ha még nem járt le, nem vonták vissza, és éppen nincs felfüggesztve. Titkosító tanúsítvány esetén kizárólag érvényes tanúsítványban lévő nyilvános kulccsal szabad titkosítani. Autentikációs tanúsítvány esetén kizárólag az érvényes tanúsítványt szabad elfogadni. Aláíró tanúsítvány esetén kizárólag az érvényes tanúsítvány alapján készült aláírás az érvényes. (Aláíró tanúsítvány esetén az aláíró tanúsítványának az aláírás pillanatában kell érvényesnek lennie. Ha az aláíró tanúsítványa lejár vagy visszavonásra kerül, a korábban létrehozott aláírások akkor is érvényesek, de ekkor — például időbélyeg segítségével — bizonyítani kell, hogy az aláírás akkor készült, amikor a tanúsítvány még érvényes volt.)

- *Visszavont tanúsítvány*: Ha a tanúsítványt kibocsátó hitelesítés-szolgáltató közzéteszi, hogy a tanúsítványt visszavonja, akkor a tanúsítvány már nem érvényes, nem szabad elfogadni. A tanúsítványokat általában akkor vonják vissza, ha a hozzájuk tartozó magánkulcs illetéktelen kezekbe került, vagy ha a tanúsítványban szereplő valamely adat megváltozott, vagy ha ezek valamelyike alappal feltételezhető. A visszavont tanúsítvány soha többet nem tehető érvényessé. A visszavont tanúsítványhoz tartozó magánkulccsal korábban aláírt dokumentumok érvényességét a visszavonás nem érinti, feltéve, hogy bizonyítható, hogy az aláírás akkor készült, amikor a tanúsítvány még érvényes volt.

- *Felfüggesztett tanúsítvány*: A felfüggesztett tanúsítvány éppúgy érvénytelen, mintha visszavonták volna. A különbség az, hogy a felfüggesztett tanúsítványok újra érvényessé tehetőek, ha visszaállítják őket. Akkor szoktak egy tanúsítványt felfüggeszteni, ha felmerül a gyanúja, hogy a hozzá tartozó magánkulcs (intelligens kártya) illetéktelen kezekbe került. Akkor szokás a felfüggesztett tanúsítványokat visszaállítani, ha kiderül, hogy a kulcs mégsem volt illetéktelen kezekben. A felfüggesztés egy fontos felhasználói igényt elégít ki: könnyen és gyorsan letiltható a tanúsítvány, viszont – mivel a felfüggesztés nem végleges – egy téves felfüggesztés sokkal kisebb problémát okoz, mint egy téves visszavonás. Ugyanakkor a felfüggesztés sok bonyodalmat is okoz: például az előbb felfüggesztett, majd visszaállított tanúsítványok esetén a különböző időpontban begyűjtött visszavonási információk akár eltérő eredményt is adhatnak.

Azt, hogy egy tanúsítvány fel van-e függesztve vagy vissza van-e vonva, a tanúsítvány *visszavonási állapot*ának is nevezik. Tanúsítvány ellenőrzésekor nemcsak azt ellenőrizzük, hogy a tanúsítvány nem járt-e le, hanem a visszavonási állapotot is megvizsgáljuk. (Lásd: 3.3.7. fejezet.)

A hitelesítés-szolgáltató közzéteszi a tanúsítványok visszavonási állapotát, hogy bárki ellenőrizhesse, hogy egy adott tanúsítvány érvényes-e. A legtöbb hitelesítés-szolgáltató

periodikusan kibocsát egy olyan listát, amely a visszavont tanúsítványok sorozatszámát tartalmazza (ez az ún. tanúsítvány-visszavonási lista, más néven CRL), de sok szolgáltatónál online módon is rákérdezhetünk az egyes tanúsítványok visszavonási állapotára (ez online tanúsítvány-állapot protokoll, más néven OCSP). A visszavonási állapot közzétételének módjáról a hitelesítés-szolgáltatókról szóló fejezetben részletesen is írunk (4.1.5. fejezet).

A hitelesítés-szolgáltatók általában csak azon tanúsítványok visszavonási állapotát teszik közzé, amelyek még nem jártak le, mert a lejárt tanúsítványok eleve érvénytelenek. Ez akkor jelent problémát, ha elektronikus aláírást ellenőrzünk, mert akkor általában nem az aktuális visszavonási állapotra, hanem egy múltbéli visszavonási állapotra vagyunk kíváncsiak. Aláírás ellenőrzésekor érdektelen, hogy a tanúsítvány érvényes-e még, ekkor az a kérdés, hogy érvényes volt-e akkor, amikor az aláírás készült. (Lásd: 6.5. fejezet.)

3.3.10. A tanúsítvány és a kulcspár életciklusának vége

Előfordulhat, hogy a szolgáltató nem bocsát ki több tanúsítványt egy adott nyilvános kulcshoz. Lehet, hogy az új tanúsítványt egy másik kulcshoz bocsátja ki (mert például a régi kulcspár magánkulcsa kompromittálódott), de az is lehet, hogy a tanúsítvány alanya nem szeretne többet tanúsítványt használni.

Ha egy aláíró vagy autentikációs magánkulcshoz nem tartozik érvényes tanúsítvány, a kérdéses magánkulcsot nincs értelme többet használni. Az érvénytelen autentikációs tanúsítványt várhatóan senki sem fogja elfogadni, és az érvénytelen aláíró tanúsítvány alapján készült aláírások érvénytelenek. A tanúsítványra vonatkozó hitelesítési rend ilyenkor általában tiltja is a magánkulcs további használatát, és előírja, hogy a magánkulcsot meg kell semmisíteni.

Titkosító tanúsítvány esetén a helyzet nem ilyen egyszerű, ekkor lehet értelme tovább őrizni és használni a magánkulcsot. Előfordulhat, hogy még vannak olyan titkosított dokumentumok, amelyeket csak az adott magánkulccsal lehet visszafejteni. Az is előfordulhat, hogy ezek nincsenek az alany birtokában, és az alany esetleg nem is tud a létezésükről.

> ***3.5. Példa:*** *Alajos titkosított e-mailt küld Bendegúznak. Bendegúz egy ideig nem kapja meg a levelet, akár azért, mert nincs Internet-közelben, akár azért, mert az e-mail néhány napig bolyong a levelező szerverek útvesztőjében. Lehet, hogy közben Bendegúz titkosító tanúsítványa lejár, és az e-mail csak akkor jut el Bendegúzhoz, amikor az ő tanúsítványa már érvénytelen. Ez nem baj, Bendegúz természetesen ekkor is visszafejtheti a neki szóló titkosított e-mailt.*

Ha egy titkosító tanúsítványunk lejár, és az új titkosító tanúsítványunk másik kulcspárhoz tartozik, két lehetőségünk van. Vagy át kell titkosítani a régi titkosított dokumentumainkat az új tanúsítványunk szerint, vagy a régi tanúsítvány magánkulcsát is meg kell őriznünk és rendszeresen használnunk kell. (Lásd: 12.4.18. fejezet.)

Míg egy érvénytelen aláíró vagy autentikációs tanúsítvány magánkulcsával már nem lehet visszaélni, titkosító tanúsítványok esetén ez sem igaz. Ha egy támadó megszerzi egy érvénytelen titkosító tanúsítványunk magánkulcsát, segítségével visszafejtheti az általa korábban elfogott, nekünk szóló, az adott tanúsítvánnyal titkosított üzeneteket.

A tanúsítvány lejártát követően a hitelesítés-szolgáltató meghatározott ideig megőrzi a tanúsítvánnyal kapcsolatos információkat, például, hogy a tanúsítványt pontosan mely alanynak bocsátotta ki, illetve a tanúsítvány visszavonási állapotával kapcsolatos változásokat. (Lásd: 4.3.8. fejezet.)

3.4. A tanúsítvány felépítése

A tanúsítványok felépítését az X.509 specifikáció, illetve az RFC 5280 specifikáció határozza meg. A minősített tanúsítványokra vonatkozóan az ETSI TS 101 862, a nem minősített, de természetes személy számára kibocsátott tanúsítványokkal kapcsolatban az ETSI TS 102 280 fogalmaz meg további követelményeket. [191], [152], [50], [59]

Minden tanúsítvány tartalmazza a kibocsátójának (a hitelesítés-szolgáltatónak) és az *alany*ának az adatait, az alany *nyilvános kulcs*át, és más, a tanúsítvány felhasználásához vagy ellenőrzéséhez szükséges információkat. A tanúsítvány emellett tartalmazza a hitelesítés-szolgáltató aláírását.

Napjainkban az X.509-es specifikáció 3-as verziójának megfelelő tanúsítványokat használunk, ezek főbb mezőit az alábbiakban mutatjuk be. Nagyon sokféle X.509 v3 tanúsítvány hozható létre, az egyes szolgáltatók, alkalmazások, országok markánsan különböző módon értelmezik a tanúsítványok egyes mezőit, és különböző mezőket tüntetnek fel, követelnek meg vagy hagynak el; az így létrejövő tanúsítvány-sablonokat *tanúsítvány-profil*nak nevezzük. Itt nem célunk teljeskörű képet adni az X.509 v3 tanúsítványok szerkezetéről, csak a fontosabb mezőket és azok gyakori értelmezését írjuk le. Az X.509 tanúsítvány-mezők értelmezésében elmélyedni kívánó, vagy az újabb tanúsítvány-profilok létrehozását tervező olvasóknak elrettentésül Peter Gutman „X.509 Style Guide” című, rendkívül szórakoztató munkáját ajánljuk. [75]

A mező angol nyelvű megnevezését zárójelben tüntettük fel. Előfordul, hogy egy mezőnek többféle magyar neve, fordítása is használatos, akkor ezeket „/” jellel választottuk el egymástól.

- Verzió (`Version`): A tanúsítvány formátumának verziója. Az X.509 (v3) formátumú tanúsítványok esetén itt mindig „v3” szerepel.
- Sorozatszám (`Serial Number`): A tanúsítványt kibocsátó hitelesítés-szolgáltató minden tanúsítványnak egyedi sorozatszámot ad. A sorozatszám egyediségét a szolgáltató általában a tanúsítvány aláírásához használt szolgáltatói kulcspár (szolgáltatói

tanúsítvány) vonatkozásában biztosítja, tehát ha egy szolgáltató több hitelesítő egységet működtet, előfordulhat, hogy a különböző hitelesítő egységek (pl. a minősítettnél és a nem minősítettnél) azonos sorozatszámú tanúsítványokat bocsátanak ki.

- Algoritmus azonosító (`Algorithm ID`): Annak az algoritmusnak vagy algoritmuskészletnek a neve, amellyel a kibocsátó hitelesítés-szolgáltató aláírta a tanúsítványt. Ma a leggyakrabban az „sha1RSA" érték szerepel itt, amely azt jelenti, hogy a kibocsátó az sha1 lenyomatképző (hash) algoritmust használta, az aláírást pedig az RSA algoritmussal készítette el.

- Kibocsátó megnevezése (`Issuer`): A tanúsítványt kibocsátó hitelesítés-szolgáltató megnevezése. A megnevezés egy X.500 szerinti megkülönböztetett név (distinguished name, DN), amely például a következőképpen nézhet ki:

  ```
  CN = Qualified e-Szigno CA
  OU = e-Szigno CA
  O = Microsec Ltd.
  L = Budapest
  C = HU
  ```

  A fenti név egy általános elnevezéssel (`common name`) kezdődik, és a „kis" egységektől (`organization unit`) a „nagyok" (`organization`, `locality`, `country`) felé haladva írja le, hogy miről van szó. A megnevezés mindig tartalmazza az ország kétbetűs kódját (Magyarország esetén „HU").

  A tanúsítványban szereplő megnevezésekről egy külön alfejezetben (3.5. fejezet) részletesen írunk.

- A tanúsítvány érvényessége kezdetének (`notBefore`) és végének (`notAfter`) időpontja: Végfelhasználói tanúsítványok esetén a `notBefore` időpont meg szokott egyezni a tanúsítvány kibocsátásának[3] időpontjával.

- Az Alany/Aláíró/Tulajdonos megnevezése (`Subject`): A tanúsítvány alanyának (aláírásra szolgáló tanúsítvány esetén az aláírónak) az X.500 szerinti megnevezése. Ez a megnevezés ugyanúgy épül fel, mint a kibocsátó hitelesítés-szolgáltató megnevezése, például így nézhet ki:

  ```
  Sorozatszám = 1.3.6.1.4.1.21528.2.2.3.2
  E = istvan.berta@microsec.hu
  CN = Dr. Berta István Zsolt
  OU = e-Szignó HSZ
  ```

[3] Azzal az időponttal, amikor a hitelesítés-szolgáltató létrehozta a tanúsítványt, ami a tanúsítvány aláírását közvetlenül megelőző időpont. A tanúsítvány jogi értelemben vett kibocsátása egészen más időpont is lehet. (Lásd: 3.3.5. fejezet.)

`O = Microsec Kft.`
`L = Budapest`
`C = HU`

A fenti példában nem álneves tanúsítványról van szó, és a common name az alany személyi igazolványa szerinti nevét tartalmazza. Ez a megnevezés tartalmazza az alany e-mail címét, és tartalmaz egy sorozatszámot is.

A tanúsítványban szereplő megnevezésekről egy külön alfejezetben (3.5. fejezet) részletesen írunk.

- Nyilvános kulcs (`Subject Public Key Info`): Az alany nyilvános kulcsa, és annak az algoritmusnak (pl. RSA) az azonosítója, amellyel a kulcs használható.

- Tanúsítvány irányelv / Hitelesítési rendek (`Certificate Policies`): E mezőben egy vagy több hitelesítési rendre való hivatkozás szerepel, de e hivatkozások egészen mást jelentenek attól függően, hogy milyen típusú tanúsítványról van szó:

  – *Végfelhasználói tanúsítványok esetén* a `certificate policies` mező *a tanúsítványra vonatkozó* hitelesítési rend azonosítóját (OID-jét[4]) tartalmazza. (Ezen túl itt szerepelhet például a hitelesítési rend, illetve a szolgáltatási szabályzat elérhetősége és egy szöveges megjegyzés is.)

  A hitelesítési rend határozza meg, hogy a hitelesítés-szolgáltató milyen szabályrendszer szerint bocsátotta ki a tanúsítványt, hogyan kell azt ellenőrizni, és mennyi felelősséget vállal érte a hitelesítés-szolgáltató. Egy tanúsítvány ellenőrzése, elfogadása előtt célszerű megnézni a rá vonatkozó hitelesítési rendet, abból állapítható meg, hogy a tanúsítványt hogyan kell értelmezni, és használható-e egyáltalán arra a célra, amire fel szeretnénk használni. (Lásd: 4.1.3. fejezet.)

  – *Hitelesítés-szolgáltatók tanúsítványai esetén* a `certificate policies` mezőben szereplő OID-k *a tanúsítványláncra tartalmaznak megkötések*et, korlátozzák, hogy az adott szolgáltatói tanúsítványt milyen tanúsítványláncokban lehet felhasználni. (Lásd: 5. fejezet.)

  Csak olyan tanúsítványláncban szerepelhet az adott szolgáltatói tanúsítvány, amelynek minden eleme tartalmaz legalább egyet az adott szolgáltatói tanúsítványban feltüntetett hitelesítési rend OID-k közül[5]. (Legalább egy OID-nek végig kell vonulnia az egész láncon.)

  3.6. Példa: *Ha egy szolgáltatói tanúsítványban az X, Y és Z hitelesítési rendek OID-i szerepelnek, az adott szolgáltatói tanúsítvány csak olyan*

[4] Az OID egy egyedi azonosító, amely egymástól pontokkal elválasztott számokat tartalmaz. OID bármihez rendelhető, OID segítségével egyértelműen hivatkozhatunk objektumokra, dokumentumokra, szervezetekre, személyekre, algoritmusokra, és az X.509 tanúsítványok egyes mezőire is OID segítségével szokás hivatkozni.

[5] Az adott OID-vel ekvivalens OID is elfogadható (5.5.3.3. fejezet)

láncokban használható, amelyek vagy minden elemében szerepel az X OID, vagy minden elemében szerepel az Y OID, vagy minden elemében szerepel a Z OID.

A szolgáltatói tanúsítványban feltüntetett OID-vel így korlátozható, hogy az adott szolgáltató tanúsítvány alatt (és felett) milyen OID-k lehetnek a láncban, így az adott szolgáltatói tanúsítvány alatti végfelhasználói tanúsítványokra vonatkozó hitelesítési rendek köre is korlátozható.

Megjegyzés: Ez egyúttal azt is jelenti, hogy a láncban szereplő minden hitelesítés-szolgáltatónak támogatnia kell az adott hitelesítési rendet – a szükséges mértékben.

A hitelesítés-szolgáltatók tanúsítványaiban gyakran az `anyPolicy` nevű speciális OID szokott szerepelni, így az adott szolgáltatói tanúsítvány bármilyen láncban előfordulhat.

A tanúsítványláncban szereplő hitelesítési rend OID-kre további megszorítások is megfogalmazhatóak a láncban szereplő tanúsítványokban. Ezeket a tanúsítványláncokról szóló fejezetben tárgyaljuk részletesen (5.5.3.3. fejezet).

- Az Alany/Aláíró/Tulajdonos alternatív neve (`Subject Alternative Names`): Ez a mező az alany valamely egyéb megnevezését tartalmazhatja. Gyakori, hogy valaki nem pontosan úgy használja a nevét, ahogy az a személyi igazolványában szerepel. E nevet alternatív névként lehet feltüntetni. Az alany e-mail címét is ebben a mezőben szokás feltüntetni, de sok alkalmazás elvárja, hogy az megnevezésében (DN) is szerepeljen.

- CRL elérési helyei (`CRL Distribution Point`): Azt mondja meg, hogy hol érhető el az adott tanúsítványra vonatkozó visszavonási lista. E mező néha alapvetően fontos a tanúsítvány ellenőrzéséhez, de sajnos ez a mező nem minden tanúsítványban szerepel.

- Hozzáférés a kiállítói információkhoz (`Authority Information Access`): Ebben a mezőben például a tanúsítványt kibocsátó hitelesítés-szolgáltató tanúsítványának elérési helye és a tanúsítványhoz tartozó OCSP szolgáltatás elérési helye szerepelhet. E mező nem minden tanúsítványban szerepel.

- Kulcshasználat (`Key Usage`) és Kibővített kulcshasználat (`Extended Key Usage`): E mezők azt mondják meg, hogy a tanúsítványt milyen célra lehet felhasználni, tehát például aláíró, titkosító vagy autentikációs tanúsítványról van-e szó.

- Alapvető típusmegkötések (`Basic Constraints`): Itt az szerepel, hogy hitelesítési szolgáltatói vagy végfelhasználói tanúsítványról van-e szó, valamint az, hogy van-e hossz-korlátozás a tanúsítványláncra. E mező segítségével egy hitelesítés-szolgáltató arra lehet befolyással, hogy egy általa felülhitelesített kulcspár működhet-e hitelesítés-szolgáltatóként, és felülhitelesíthet-e más hitelesítés-szolgáltatókat.

A PKI logikája szerint a hitelesítés-szolgáltatók megbízható felek, és elhisszük, ha egy hitelesítés-szolgáltató azt állítja, hogy egy adott kulcspár egy adott entitáshoz tartozik. A végfelhasználók ilyen állításait a PKI szerint nem kell elhinni. A `basic constraints` mező alapján dönthető el, hogy szolgáltatói vagy végfelhasználói tanúsítványról van szó, így e mező korrekt kezelése különösen fontos biztonsági kérdés.

> ***3.7. Példa:*** *Manfréd vesz egy végfelhasználói tanúsítványt az X hitelesítés-szolgáltatótól. Ezt követően Manfréd a saját magánkulcsával aláír magának egy Alajos névre kibocsátott tanúsítványt, majd ezen új tanúsítvány szerint Alajos nevében készít aláírásokat.*
>
> *Ha Bendegúz ilyen aláírással találkozik, el kell utasítania őket, mert Manfréd végfelhasználó, és nem jogosult tanúsítványt kibocsátani. Ha Bendegúz alkalmazása nem kezeli a* `basic constraints` *mezőt, könnyen becsaphatja őt Manfréd.*

- `QCStatements`: Minősített tanúsítványokban szerepel az az állítás, hogy a tanúsítvány minősített, és emellett egyéb állítások, korlátozások is szerepelhetnek. Például a hitelesítés-szolgáltató korlátozhatja az adott tanúsítvánnyal egy alkalommal vállalható kötelezettség mértékét, azaz megadhatja a tranzakciós limitet (4.8.3.3. fejezet), és azt itt is feltüntetheti. Sajnos ezt a mezőt a Windows nem tudja rendesen megjeleníteni, így a hitelesítés-szolgáltatók a fontosabb információkat máshol (pl. a tanúsítvány-irányelv mező szöveges figyelmeztetéseiben) is fel szokták tüntetni. [50]
- Aláírás (`Certificate Signature`): A kibocsátó aláírása a tanúsítványon. (A Windows nem jeleníti meg)

Egy tanúsítvány a fentieken túl még további mezőket is tartalmazhat.

3.5. A szereplők megnevezése (DN)

A tanúsítvány azt igazolja, hogy egy adott nyilvános kulcs egy adott alanyhoz tartozik. A tanúsítványból az alany megnevezése alapján következtethetünk az alany kilétére.

Az X.509 tanúsítványok mind az alanyra, mind a tanúsítvány kibocsátójára az X.500 szerinti megkülönböztetett nevével (distinguished name, DN) hivatkoznak. Az X.500 specifikáció egy *globális címtár*at határoz meg, amelyben minden szereplőnek egyedi megnevezése van. A hitelesítés-szolgáltatók e globális címtárban adnak egyedi nevet az alanynak. A hitelesítés-szolgáltatónak kötelessége biztosítania a DN egyediségét.

A DN mezőnév-érték párokból áll. A mező neve helyén OID szerepel, amelyet a DN-t értelmező programok szövegesen jelenítenek meg, az érték pedig általában UTF-8 kódolású.

(A legtöbb mező esetén ma már az UTF-8 kódolás használata kötelező, de más mezők, például a `serialNumber` esetében más kódolást kell használni. Bizonyos esetekben visszafelé kompatibilitás miatt az UTF-8 mellett más kódolások is megengedettek. Ilyen eset például, ha egy új tanúsítvány DN-jének valami miatt meg kell egyeznie egy régebbi tanúsítványban már szereplő DN-nel.) A név-érték párok sorozata egy hierarchikus struktúrába, az X.500 által meghatározott globális címtárba sorolja be a tanúsítvány alanyát.

Például a következő egy lehetséges DN, amely Magyarországon, a Kókler Bt. alkalmazásában álló Kovács Józsefet jelenti:

```
CN = Kovács József
O = Kókler Bt.
C = HU
```

A DN-ben elvileg tetszőleges mezők előfordulhatnak, és egyazon típusú mező (pl. a szervezet nevét tartalmazó `O`, azaz organization) akár többször is szerepelhet. Célszerű a DN-ek elemeit az X.500 logikája szerinti sorrendben szerepeltetni. Az X.500 egy hierarchikus címtárat határoz meg, amelyben például megcímezhetőek az országokon belüli szervezetek, illetve azok munkatársai. Ha a DN az X.500 logikáját követi, akkor egyúttal az X.500 egy entitására is hivatkozik. Az RFC 5280 meghatározza [152], hogy az alkalmazásoknak mely mezőket kell kötelezően támogatniuk. Nem célszerű saját, egzotikus mezőket használni, mert akkor sok alkalmazás sem megjeleníteni, sem értelmezni nem tudja majd őket.

A következő DN-mezők különös figyelmet érdemelnek:

- `Common name (CN)`: Az alany „neve”. Ez lehet valódi név, lehet álnév, lehet a személyi igazolvány szerinti írásmóddal, de lehet másképp is. Webszerver tanúsítványok esetén a szerver domainje, illetve subdomainje szerepel itt. Alapvetően bármi szerepelhet itt, általában így hívjuk az alanyt. E mező nagyon sokféleképpen használható, és ez sok problémához vezet. A legtöbb program (pl. levelezőprogram) ezt a mezőt jeleníti meg, így általában e mező tartalma a legfontosabb.

  Természetes személy számára kibocsátott tanúsítvány esetén legtöbbször a személy valódi neve szokott itt szerepelni a személyazonosításra használt okirat szerinti írásmóddal. Szervezet számára kibocsátott tanúsítvány esetén a szervezet neve is szerepelhet itt, automatizmus számára kibocsátott tanúsítvány esetén az automatizmus neve.

  > **Megjegyzés:** Strukturáltan is tárolhatjuk az alany nevét a `surname` és `given name` mezőkben, de ennek igen kevés az értelme. Egyrészt mert a programok általában a `common name` mezőt használják, másrészt mert a magyar nevek nem illeszkednek erre a struktúrára.

  3.8. Példa: *Kovács János vezetékneve Kovács, a keresztneve pedig János.*

Dr. Kis-Tóth Béla Eduárdné Dr. S. Nagy-Szabó Olga Emília esetén nem egyszerű kérdés, hogy melyik a keresztneve, és melyik a vezetékneve. Ha létezik is rá szabály, a különféle hitelesítés-szolgáltatók, alkalmazások, hivatalnokok egészen biztosan másképp alkalmazzák. Álláspontunk szerint az ilyen nevek `surname` *és* `given name` *mezőkbe kényszerítéséből semmi jó nem származhat.*

`Email address (EA)`: Az alany e-mail címe. Az X.500 szerint ez nem való a DN-be, nem itt kell feltüntetni. Az RFC 5280 szerint az alany alternatív neve mező `rfc822name` elemébe kell írni az e-mail címet. Ennek ellenére sok alkalmazás mégis az alany DN-jében keresi az e-mail címet, és nem működik, ha ott nem találja. Ezért sok szolgáltató mégis feltünteti az e-mail címet az alany megnevezésében.

`Organization (O)`: Az alany szervezetének, azaz a képviselt szervezetnek a neve. E mező jelentheti, hogy az alany a szervezet alkalmazottja, jelenthet egy kamarát vagy egyesületet, amelynek az alany a tagja, esetleg egy szervezetet, amelynek az alany az ügyfele, vagy jelentheti azt, hogy az alany a szervezetet képviseli (pl. könyvelője az adott szervezetnek), de van olyan szolgáltató is, amely azon regisztrációs egységet írja ide, amely az alanyt regisztrálta. (Lásd: 3.3.4. fejezet.) Ha az alany egy számítógép, akkor e mezőben vagy a számítógépet működtető szervezet szokott megjelenni, vagy az a szervezet, amelyiknek a nevében a számítógép működik.

Ha egy tanúsítványban több `organization` mező szerepel, egyáltalán nem biztos, hogy az alany ugyanolyan viszonyban van a két szervezettel. Például van olyan hazai hitelesítés-szolgáltató, amely egy ügyvédi tanúsítvány egyik `organization` mezejébe az ügyvéd irodájának a nevét, a másik `organization` mezejébe a területileg illetékes ügyvédi kamara nevét írja bele.

`Country (C)`: Talán ezen, országra hivatkozó mező értelmezhető a legtöbb módon. Jelentheti az alany születése szerinti országot, az alany állandó lakcíme szerinti országot, az alany tartózkodási helye szerinti országot, az alany szervezetének székhelye szerinti országot, az alany szervezetének telephelye szerinti országot. Azt a helyet is jelentheti, ahol az alany dolgozik, de jelentheti a szervezet központi címe szerinti országot is, ezen kívül, jelentheti azt az országot is, amelynek jogrendje szerint a tanúsítványt értelmezni kell. Ezen kívül olyan hitelesítés-szolgáltató is van, amely roppant következetesen konstans értéket ír ide, teljesen függetlenül az alany kilététől. E mezőben általában az ország kétbetűs kódja szerepel, de némely hitelesítés-szolgáltató a DNS szerinti top-level-domain kódját írja ide (ez utóbbi a szabványok szerint helytelen, de mégis előfordul).

`Locality (L)`: Általában egy város vagy település neve szokott szerepelni itt, e mezőt a `country` mezőhöz hasonlóan sokszínű módon lehet használni.

`Pseudonym`: Álnevet szokás itt feltüntetni.

`Title`: Az alany szerepköre, beosztása, fokozata, hivatása vagy valamely jogosultsága szerepelhet itt. E tulajdonság – attribútum – lehet, hogy a szervezetétől, valamely szervezeti egységtől származik (pl. „osztályvezető"), de lehet, hogy önállóan rendelkezik vele (pl. „ügyvéd"). E mező tartalmát célszerű inkább attribútum-tanúsítványban (11. fejezet) feltüntetni.

`Serialnumber (SN)`: A hitelesítés-szolgáltatónak biztosítania kell, hogy a DN egyedi legyen. Ha nincs más megoldás, ezt e mezővel teheti meg. Például egy Kovács József kaphatja a következő DN-t, ha már van egy Kovács József az adott szolgáltatónál: `CN = Kovács József, SN = 2, C = HU`.

E megoldás helyett a hitelesítés-szolgáltatók gyakran írnak értelmes azonosító számokat a DN-be. Például személyi igazolvány szám szokott itt szerepelni, bár ez személyes adat, és csak az alany beleegyezésével tüntethető fel itt.

> ***3.9. Példa:*** *Az e-Szignó Hitelesítés Szolgáltató a* `serialnumber` *mezőben egy OID-t tüntet fel, amelyet saját rendszerében az alanyhoz rendel. Ezen azonosító egyértelműen meghivatkozza az alanyt, így ha valaki két tanúsítvánnyal találkozik, amelyben ugyanazon azonosító szerepel, akkor biztos lehet benne, hogy a két tanúsítvány alanya egy és ugyanaz.*

A tanúsítványra vonatkozó hitelesítési rendből derül ki, hogy a tanúsítványt kibocsátó hitelesítés-szolgáltató milyen módon töltötte ki e mezőket.

Ha egyazon szolgáltató egyazon hitelesítő egysége kibocsát két tanúsítványt, amelynek azonos a DN-je, akkor biztosak lehetünk benne, hogy a két tanúsítvány ugyanahhoz az alanyhoz tartozik. Ha a két DN egy bitben is eltér, akkor már nem.

Az X.500 DN-eket igen sok kritika éri: [75]

- Szinte bármi lehet a DN-ben, nemigen van olyan mező, amelynek kötelezően szerepelnie kell.
- A DN nagyon sok módon értelmezhető.
- A DN nem alkalmas a valós viszonyok leírására. A világ nem szigorúan hierarchikus, még akkor sem, ha egyes specifikációk készítői ezt szeretnék. A magyar viszonyok leírására szintén nem alkalmas.
- A DN-t nem lehet begépelni, vagy megjegyezni, túl hosszú, túl körülményes.
- A DN-ek gépi összehasonlítása problémás. Különbözik-e két DN, ha ugyanazok az adatok különböző sorrendben szerepelnek benne? Mi a helyzet, ha két DN-ben eltérő kódolással szerepelnek az adatok? Ekkor az RFC 5280 mindössze annyit mond, hogy helyesen jár el, aki különbözőnek tekinti a két DN-t.

- Sok szolgáltató szabálytalanul, össze-vissza tölti ki a DN-eket.
- Az alany kilétét a DN alapján nem lehet egyértelműen megállapítani, mert személyes adat csak az alany hozzájárulásával tüntethető fel a tanúsítványban. Ez nem tehető kötelezővé az alanyok számára, lesz, aki nem járul hozzá, és a hitelesítés-szolgáltatónak akkor is ki kell bocsátania a tanúsítványt.
- A DN nem egyedi. A hitelesítés-szolgáltató csak saját rendszerén belül biztosíthatja a DN egyediségét, így semmi akadálya nincs, hogy pl. két különböző szolgáltató két különböző személynek bocsásson ki tanúsítványt a `CN = Kovács József, C = HU` DN-re.
- Sokszor nemcsak az alany kilétét akarják feltüntetni a DN-ben, hanem az alany szerepkörét, tulajdonságait, attribútumait is, és ebből jókora kalamajka származhat. E problémával és ennek kezelésével később egy külön fejezetben (11. fejezet) foglalkozunk.
- Nincs globális X.500 címtár. Nem használják, nem tartják karban, és e technológia túl bonyolult.

3.6. Miért van olyan sokfajta tanúsítvány?

A nyilvános kulcsú infrastruktúra (PKI, public key infrastructure) minden szereplőjének van egy nyilvános kulcsa és egy magánkulcsa. A nyilvános kulcsát mindenki nyilvánosságra hozza, míg a magánkulcsát mindenki titokban tartja. Ha ismerjük valakinek a nyilvános kulcsát, akkor biztonságosan (például titkosan, hitelesen) kommunikálhatunk vele. A nyilvános kulcsot hiteles módon kell megszereznünk, csak így lehetünk biztosak benne, hogy az valóban annak a nyilvános kulcsa, akivel kommunikálni szeretnénk. A nyilvános kulcsú infrastruktúrában ez általában úgy történik, hogy a nyilvános kulcsot tanúsítványba foglalva szerezzük meg, a tanúsítványban egy megbízható fél — egy hitelesítés-szolgáltató -– aláírásával igazolja, hogy az adott nyilvános kulcs kihez tartozik.

A gyakorlatban a helyzet nem ilyen egyszerű. A PKI szereplői általában rákényszerülnek, hogy nem egy, hanem több tanúsítványuk és kulcspáruk legyen. Aki kapcsolatba kerül a PKI-vel, gyakran szembesül azzal a problémával, hogy minden célra külön-külön tanúsítványt kell vagy kellene vásárolnia, mert meglévő tanúsítványát (esetleg tanúsítványait) valami miatt nem tudja máshol felhasználni.

Ezen okok egy része alapvető biztonsági kérdésekre vezethető vissza, és szabványokban, esetleg jogszabályokban is megjelenik. Másik részük szerencsétlen, hibás megoldások, szabványok közötti ellentmondások miatt fordul elő. Ezek vagy ismert, elterjedt alkalmazások hibái, korlátai vagy furcsaságai miatt kötött kompromisszumokban gyökereznek, vagy amiatt

jelentkeznek, hogy a különálló PKI rendszerek tervezése során nem gondolták végig kellőképpen, hogy hogyan lehet majd őket összekapcsolni.

A következőkben ezen jelenség okait tekintjük át.

3.6.1. Lejárt és visszavont tanúsítványok

A tanúsítványok nem örökké érvényesek. Ennek egyik oka, hogy a PKI szereplői — akár emberek, akár számítógépek — általában nem tudják tökéletesen biztonságosan őrizni a tanúsítványukhoz tartozó magánkulcsukat. Másik oka, hogy a tanúsítványban szereplő adataik, illetve a tanúsítvány által igazolt szerepköreik gyakran megváltoznak.

Egy tanúsítvány kétféle módon válhat érvénytelenné: lejár vagy visszavonják. Mindkettő azt eredményezi, hogy a tanúsítvány birtokosa nem használhatja többé régi tanúsítványát, új tanúsítványra van szüksége.

Ha megnézzük egy hitelesítés-szolgáltató tanúsítványtárát, nem kell csodálkoznunk rajta, ha valakinek több tanúsítványa található benne. Szét kell választanunk a lejárt, visszavont tanúsítványokat az aktuális, érvényes tanúsítványtól (vagy tanúsítványoktól).

3.6.2. Aláíró, titkosító és autentikációs tanúsítványok

Megmutattuk (3.2.1. fejezet), hogy az aláíró, titkosító és autentikációs tanúsítványokat el kell különíteni egymástól. Más az életciklusuk, és Magyarországon még törvény is tiltja, hogy az aláíró tanúsítvány magánkulcsát bármilyen más célra használjuk. Az aláírásra, titkosításra és autentikációra külön-külön kulcspárt kell használni, így e célra mindenkinek három külön kulcspárra és három külön tanúsítványra van szüksége.

3.6.3. A tanúsítvány használatának célja

Egy tanúsítványban feltüntethető, hogy milyen célra használható. Így kerülhető el például az, hogy valaki egy aláíró tanúsítványban lévő nyilvános kulccsal titkosítson egy üzenetet. A kulcshasználat célját a tanúsítvány `Key Usage` mezejében lévő bitekkel szokás jelölni, és a tanúsítvány `ExtendedKeyUsage` mezejével lehet tovább szűkíteni.

A kulcshasználati bitek nagyon sok kombinációt megengednek, és e téren gyakran ellentmondanak egymásnak az európai és az amerikai szabványok. Előfordulhat, hogy egyes alkalmazások elvárják, hogy bizonyos kulcshasználati bitek szerepeljenek a tanúsítványban, illetve elvárják, hogy bizonyos bitek ne szerepeljenek (mert akkor a tanúsítvány biztosan más célra való). A különféle szabályozások, szabványok és alkalmazások miatt nem könnyű feladat, hogy egy tanúsítvány mindenhol használható legyen. (Például ha a Windowsban intelligens kártyás beléptetéshez szeretnénk használni egy tanúsítványt, olyan egzotikus `KeyUsage` értéket kell beállítanunk, amely arra utal, hogy a tanúsítványt egyszerre használnánk autentikációra

és titkosításra.) További nehézség, hogy a tanúsítványt kommunikáció védelmére szeretnénk használni, így nemcsak saját szoftverünk igényeinek kell megfelelnünk, hanem kommunikációs partnerünk (vagy partnereink) szoftverének (szoftvereinek) is.

Bizonyos célokra a `KeyUsage` biteken túl `ExtendedKeyUsage` értékekre is szükség van a tanúsítványban. Például ha azt szeretnénk, hogy az általunk fejlesztett alkalmazást ügyfeleink szoftverei megbízható helyről származó alkalmazásnak tekintsék, aláírásához `CodeSigning ExtendedKeyUsage` értéket tartalmazó tanúsítványt kell használni. Hasonló módon, egy-egy kitüntetett `ExtendedKeyUsage` értékkel kell megjelölni a webszerverek tanúsítványait, az SSL kliensek tanúsítványait, a VPN tanúsítványokat stb.

Előfordulhat, hogy az egyes alkalmazások igényei miatt, illetve a speciális felhasználási célokra külön-külön tanúsítványokkal kell rendelkeznünk.

3.6.4. A tanúsítvány biztonsági szintje

A végfelhasználói tanúsítványok között az ún. minősített tanúsítványok képviselik a legmagasabb biztonsági szintet, ezek kibocsátását, kezelését az elektronikus aláírásról szóló törvény szabályozza. A minősített tanúsítvány alapján (biztonságos aláírás-létrehozó eszköz segítségével) minősített elektronikus aláírás hozható létre. A minősített tanúsítvány kizárólag aláírásra szolgálhat, „minősített titkosító” tanúsítvány nem létezik. Minősített tanúsítványt csak minősített hitelesítés-szolgáltató bocsáthat ki, kizárólag személyes regisztráció során. A szolgáltatónak felelősséget kell vállalnia a tanúsítvánnyal okozott károkért, meghatározott pénzügyi követelményekkel, felelősségbiztosítással kell rendelkeznie stb. [180]

A minősített elektronikus aláíráshoz nemcsak szigorú szabályok és biztonsági mechanizmusok kapcsolódnak, hanem erős jogi vélelmek (6.1. fejezet) is, így sok eljárásban kizárólag minősített elektronikus aláírással hitelesített dokumentum használható.

A minősített tanúsítványokon belül is léteznek különböző fokozatok. A hitelesítés-szolgáltató a tanúsítvány kibocsátásakor meghatározhatja, hogy az adott tanúsítvánnyal legfeljebb mekkora kötelezettség vállalható, ez az ún. tranzakciós limit (4.8.3.3. fejezet). Így előfordulhat, hogy egy nagy értékű ügyletben valaki nem fogad el egy tanúsítványt, mert az adott tanúsítvány ekkora értékű ügyletben már nem használható, és baj esetén a tanúsítványt kibocsátó hitelesítés-szolgáltató nem térítené meg a kárát. Még nem gyakori, hogy bizonyos célokra csak meghatározott tranzakciós limitű tanúsítványok lennének használhatóak, de a technológia terjedésével a tranzakciós limit várhatóan nagyobb jelentőséget kap majd.

A minősített tanúsítványok mellett léteznek nem minősített tanúsítványok is. Nem minősített tanúsítvány alapján legfeljebb fokozott biztonságú elektronikus aláírás hozható létre. A nem minősített tanúsítványokra (és a fokozott biztonságú elektronikus aláírásokra) nagyon kevés szabály vonatkozik, ezért különféle biztonsági szintek fordulnak elő, és gyakran nagyon nehéz közöttük különbséget tenni. Léteznek olyan nem minősített tanúsítványok, amelyeket már-már

a minősített tanúsítványokéihoz hasonló szabályok szerint bocsátanak ki, más nem minősített tanúsítványok távolról, személyes találkozás nélkül is kibocsáthatóak.

Mivel a minősített aláírásokra nagyon merev szabályok vonatkoznak, sok esetben nem minősített tanúsítványt és fokozott biztonságú aláírást szokás használni. Ilyen például az elektronikus számlázás esete, amikor nagy mennyiségű aláírást kell gyorsan elkészíteni.

Mind a minősített, mind a nem minősített biztonsági szint értelemmel bír, és ezeken belül is különféle fokozatok képzelhetőek el. Előfordulhat, hogy valakinek különböző biztonsági szintű tanúsítványai is vannak; a magasabb biztonsági szint a nagyobb bizonyító erő, az alacsonyabb a rugalmas felhasználás miatt. A tanúsítványok kibocsátására és felhasználására vonatkozó szabályokat, így a tanúsítvány biztonsági szintjét az a hitelesítési rend határozza meg, amelynek megfelelően a tanúsítványt kibocsátották. Minden tanúsítványban szerepel azon hitelesítési rend azonosítója, amelynek a tanúsítvány megfelel. Tanúsítvány biztonsági szintjére talán a hitelesítési rend alapján a legcélszerűbb hivatkozni.

3.6.5. Magánkulcs tárolása

Ha egy tanúsítványt minősített elektronikus aláírás létrehozásához szeretnénk használni, a hozzá tartozó magánkulcsnak mindenképpen biztonságos aláírás-létrehozó eszközön (pl. intelligens kártyán) kell lennie. A kulcs csak az intelligens kártyán létezik (gyakori, hogy a kártyán is keletkezett), soha nem hagyja el a kártyát, nem is lehet kinyerni a kártyából (azaz nem lehet a kártyát „lemásolni”). Így a kártyabirtokos biztos lehet benne, hogy amíg a kártya nála van, addig illetéktelen személy nem élhet vissza a kulcsával. (Lásd: 6.3.3. fejezet.)

Ha a magánkulcs nem (vagy nem csak) intelligens kártyán van, akkor szoftveres tanúsítványról, illetve „szoftveres kulcsról” beszélünk. A szoftveres kulcs egy fájl egy számítógépen, így le lehet másolni. Ezért sokkal nehezebb kézben tartani, hogy a szoftveres kulcsból hány másolat készül. Megjegyezzük, a szoftveres kulcsot jelszóval – a jelszóval képzett szimmetrikus kulccsal – titkosítva szokás tárolni, és ez bizonyos esetekben jelenthet némi védelmet az illetéktelen felhasználással szemben.

Ugyanakkor a szoftveres kulcsoknak számos előnye is van. A kulcs birtokosa egyszerre több helyen, több gépen is használhatja, vagy biztonsági másolatot készíthet belőle. Általában minden alkalmazás támogatja a szoftveres kulcsokat, és a szoftveres kulcsok használatához nincs szükség az adott típusú intelligens kártya meghajtó programjára. (Előfordulhat, hogy egy adott típusú intelligens kártyát nem támogat egy alkalmazás, vagy a kártyához nincs olyan típusú meghajtóprogram, amelyet az adott alkalmazás támogatna stb.)

Így könnyen előfordulhat, hogy valakinek azonos célú, azonos adatokat tartalmazó érvényes tanúsítványai vannak, csak az egyik szoftveres, a másiknak a magánkulcsa pedig valamilyen intelligens kártyán van.

3.6.6. Hol van letétben a magánkulcs?

Titkosító tanúsítványok esetén a magánkulcsot letétbe szokás helyezni például arra az esetre, ha a dekódoló kulcsot tartalmazó kártyánk megsemmisülne, így a letétbe helyezett kulccsal a korábban kódolt adatok visszafejthetőek. (Lásd: 3.3.3. fejezet.) Másik gyakori alkalmazás, hogy a titkosító tanúsítványát valaki egy szervezet munkatársaként használja, és a szervezet ragaszkodik ahhoz, hogy ő is hozzáférhessen a letétbe helyezett magánkulcshoz, így a dolgozó elbocsátása esetén annak közreműködése nélkül is hozzáférhet a hivatalos levelezéséhez. A szervezet ahhoz is szokott ragaszkodni, hogy más szervezet ne férjen hozzá a magánkulcshoz.

Ha valaki több szervezet nevében is folytat titkosított kommunikációt, előfordulhat, hogy az egyes szervezetekhez külön-külön titkosító tanúsítványra (és magánkulcsra) van szükség.

3.6.7. Tanúsítványban feltüntetett személyes adatok, szerepkörök

Minden tanúsítványban szerepel a tanúsítvány alanyának megnevezése, azaz DN-je (3.5. fejezet), amely az elektronikus aláírásról szóló törvény szerint álnév is lehet. Ha aláíráskor az aláíró álneves tanúsítványt használt, az aláírásból még a valódi nevét sem lehet megállapítani, de később – a hitelesítés-szolgáltató nyilvántartása alapján – mégis be lehet bizonyítani, hogy az aláírást ő készítette. Annak ellenére, hogy az álneves tanúsítványok alapján ellenőrizhető aláírások jogilag egyenértékűek a valódi nevet tartalmazó tanúsítványok alapján ellenőrizhető aláírással, álneves tanúsítványt a gyakorlatban szinte nem használnak. [5] Bizonyos felhasználási területeken (például a magyar közigazgatásban) egyáltalán nem fogadnak el álneves tanúsítványt.

Gyakori, hogy valakinek nem a nevéről, hanem a szerepköréről kell meggyőződnünk. Abban szeretnénk biztosak lenni, hogy valóban ott dolgozik-e, valóban van-e olyan jogosultsága, képzettsége, valóban az-e a hivatása stb. Ha egy tanúsítványban a tanúsítvány alanyának a nevén túl más információk is szerepelnek, a tanúsítvány ezen információkat is igazolhatja. Ez akkor jelent előnyt, ha a tanúsítványt vagy aláírást befogadó fél ezen információkat felismeri, és elfogadja. Igazolhatja például egy tanúsítvány, hogy az alanya közjegyző, vagy például egy adott cég nevében cégjegyzésre jogosult stb.

E megoldás korlátokkal is rendelkezik:

- Nem biztos, hogy a befogadó (vagy az általa használt automatizmus) megérti a tanúsítványban szereplő információkat. Ezen információk feltüntetésének módja szabványos ugyan, de a nemzetközi szabványban meghatározott mezőkben közel nem egyértelmű, hogy hogyan kell feltüntetni a speciális magyar adatokat (pl: TAJ szám). Emellett a szabványban szereplő hierarchikus struktúra nem mindig alkalmazható a gyakorlati helyzetre. A különféle alkalmazások elvárásai itt is komoly problémát jelentenek. Például sok levelezőprogram nem fogad el olyan tanúsítványt, amelyben

nem szerepel az alany e-mail címe. Ugyanakkor sokan nem szívesen tüntetnék fel e-mail címüket minden egyes aláírásukban.

- A tanúsítványt vagy aláírást befogadó fél nem mindig fogadja el a tanúsítványban szereplő adatokat, esetleg ezen adatokról nem a hitelesítés-szolgáltató igazolása alapján szeretne meggyőződni.
- Akinek több szerepköre van, annak nem szerencsés, ha minden szerepköre ugyanabban a tanúsítványban jelenik meg. Egyrészt így akivel csak kapcsolatba kerül, az minden szerepkörét megismeri, másrészt bármelyik szerepköre megváltozik, vissza kell vonni a tanúsítványát, és az új tanúsítványt esetleg csak az első kibocsátáshoz hasonló eljárás során kaphatja meg.
- Annak eldöntése is problémát jelent, hogy valaki éppen melyik szerepkörében használja a tanúsítványt, és használhatja-e a szerepköréhez tartozó (pl. közjegyzői) tanúsítványát magánszemélyként.

Előfordulhat, hogy azért van szükségünk több tanúsítványra, mert az egyes tanúsítványokban különböző adatoknak kell szerepelnie. Vagy mi nem szeretnénk, hogy minden adatunk ott legyen a tanúsítványunkban, vagy a különböző helyeken (például a szakmai kamarában, az egészségügyben vagy a közigazgatásnál) támasztanak egymásnak ellentmondó követelményeket a felhasználható tanúsítványok adattartalmával kapcsolatban.

Nem tartjuk szerencsésnek, hogy a tanúsítványban az alany nevén kívül más információ is szerepeljen. Úgy gondoljuk, ezen információkat más módon, máshol kell nyilvántartani, és e nyilvántartásokat nem a PKI szolgáltatóknak kellene vezetnie, mert ez túlbonyolítja a PKI-t, és szigetmegoldásokat hoz létre. Ugyanakkor sok PKI megoldás mégis ezt az utat választja, mert más módon nem tud könnyen építeni a tanúsítványok nyújtotta infrastruktúrára.

E problémakörrel később külön fejezetben (11. fejezet) is foglalkozunk.

3.6.8. Melyik gyökértanúsítványt használjuk?

A tanúsítványt elfogadó érintett félnek meg kell győződnie a tanúsítvány érvényességéről. Ehhez – többek között – vissza kell vezetnie a tanúsítványt egy általa elfogadott gyökértanúsítványra. Mely gyökértanúsítványokat fogadhatja el?

Előfordulhat, hogy valakinek a kommunikációs partnerei különböző gyökértanúsítványokat fogadnak el, így több – különböző gyökérre visszavezethető – tanúsítványra van szüksége.

3.6.8.1. Jogilag elfogadott gyökerek

Dönthet úgy, hogy az Eat. szerint elfogadott gyökereket fogadja el. Ilyenkor leggyakrabban a magyar hitelesítés-szolgáltatók gyökereire gondolunk, de sok külföldi gyökértanúsítvány is

ide tartozik. Mivel azonban nem létezik hiteles nyilvántartás ezen gyökerekről, ezek körét elég nehéz meghatározni.

3.6.8.2. Az alkalmazások által elfogadott gyökerek

Dönthet úgy, hogy az alkalmazása (vagy operációs rendszere) által támogatott gyökereket fogadja el. A legtöbb alkalmazásnak van tanúsítványtára, amelyben alapértelmezetten szerepelnek bizonyos gyökerek. Az alkalmazásfejlesztők általában valamilyen saját követelményrendszer szerint veszik fel ide a hitelesítés-szolgáltatókat. A nemzetközi alkalmazások által elfogadott gyökértanúsítványok köre általában nem esik egybe a Magyarországon jogilag is elfogadott gyökerek körével. Így előfordulhat, hogy a magyar hitelesítés-szolgáltató tanúsítványa alapján létrehozott aláírást a külföldi partner nem fogadja el, mert nem ismeri a magyar gyökeret; vagy egy külföldi aláírást az alkalmazás elfogad, pedig a magyar jogszabályok szerint nem szabadna elfogadnia.

3.6.8.3. PKI közösség saját gyökere

Sok PKI közösségnek saját gyökere van, és csak az ide visszavezethető tanúsítványokat fogadja el. Ekkor a közösség minden számítógépén kizárólagosan ezt az egy gyökeret kell beállítani, így a közösség elfogadja az ez által felülhitelesített szolgáltatókat, illetve elutasítja, ha a gyökér hitelesítés-szolgáltató visszavonja ezek tanúsítványait. Korlátja ennek a megoldásnak, hogy nehéz tovább szűrni ezen szolgáltatók tanúsítványai között, illetve a megoldás költséges, mert kell hozzá egy saját hitelesítés-szolgáltató, amelyet biztonságosan kell üzemeltetni. Az is végiggondolandó, hogy a saját gyökér üzemeltetése növeli-e vagy éppen csökkenti a rendszer biztonságát.

A saját gyökér egy másik buktatót is rejt. Így könnyen beleesünk abba a csapdába, hogy összemossuk a tanúsítvány érvényességét (azt, hogy a magánkulcs a tanúsítvány alanyának birtokában van) a közösséghez való tartozással (vagy más jogosultságokkal), és nehéz lesz más közösségből származó tanúsítványt befogadni a rendszerünkbe.

3.6.9. Visszavonási információk elérhetősége

Vannak tanúsítványok, amelyek visszavonási állapota könnyebben, gyorsabban ellenőrizhető, és bizonyos esetekben ilyen tanúsítványokat lehet jól használni. Aláírás ellenőrzésekor arról kell meggyőződnünk, hogy az aláíró tanúsítványa – illetve az aláíró tanúsítványához tartozó tanúsítványlánc minden egyes eleme (kivéve a gyökértanúsítványt) – érvényes volt-e az aláírás pillanatában. Ehhez olyan visszavonási információkra — visszavonási listákra (CRL), online tanúsítvány-állapot válaszokra (OCSP) — van szükség, amelyeket az aláírás pillanatát követően bocsátottak ki. Egyes hitelesítés-szolgáltatók esetén ezen információkat esetleg csak

hosszú idő után lehet beszerezni. Például sok gyökér hitelesítés-szolgáltató csak havonta, vagy még annál is ritkábban bocsát ki visszavonási listát, így esetleg csak hónapok múlva lesz a birtokunkban olyan visszavonási információ, amely igazolná az aláírás érvényességét.

Itt két külön problémával álluk szemben: egyrészt meg kell győződnünk róla, hogy a szolgáltató nem vonta vissza a tanúsítványt, és ez alapján döntést kell hoznunk. Másrészt igazolnunk kell, hogy a döntést milyen információk alapján hoztuk. Van olyan szolgáltató, amely esetén a döntést viszonylag gyorsan meghozhatjuk (látjuk, hogy nem jelent meg új visszavonási lista), de nem tudjuk ezt bizonyítani (a nem létező visszavonási listát nem tudjuk felmutatni).

Könnyen előfordulhat, hogy az aláírásunk jogilag érvényes ugyan, de az aláírást befogadó fél nem tudja, vagy csak nagyon hosszú idő után tudja ellenőrizni (és igazolni) az érvényességét. Így bizonyos célokra (például ha archiválás-szolgáltatónál akarjuk archiválni az aláírást, akkor a szolgáltatót jogszabály kötelezi arra, hogy 3 napon belül minden bizonyítékot összegyűjtsön az aláírás érvényességéről) olyan tanúsítványt kell választani, amelyhez rugalmasan érhetőek el visszavonási információk.

3.6.10. Mire ügyeljünk PKI-re épülő rendszerek tervezésénél?

Látható, hogy könnyen előállhat olyan helyzet, amikor egy új környezetben, új alkalmazásban már nem tudjuk használni a meglévő tanúsítványunkat, hanem új tanúsítványra van szükség. E jelenség bizonyos okai a PKI alapvető elveiből következnek; ilyenek például a lejáró és visszavont tanúsítványok, az aláíró, titkosító és autentikációs tanúsítványok, és a tanúsítványokhoz tartozó biztonsági szintek. Más okok viszont tervezési hibák vagy a PKI technológia gyermekbetegségeinek tekinthetők, ezek egy része megfelelő tervezéssel, szabályozással orvosolható. Úgy látjuk, hogy a PKI gyakorlati elterjedése szempontjából létfontosságú, hogy egy tanúsítvány sok helyen, sok PKI közösségben használható. Így nem kell minden közösséghez külön-külön tanúsítványt vásárolnunk, így a PKI összességében olcsóbbá válhat.

PKI-re épülő rendszerek tervezésekor legyünk tekintettel a következőkre:

- Tudomásul kell vennünk, hogy a tanúsítványok nem érvényesek örökké, így számolnunk kell azzal a jelenséggel, hogy a PKI szereplőinek a tanúsítványai időben változnak.
- El kell különíteni egymástól az aláíró, titkosító és autentikációs tanúsítványokat és az ezeket felhasználó funkciókat. Figyelembe kell vennünk, hogy az aláíró, titkosító és autentikációs tanúsítványok gyökeresen eltérő kulcsmenedzsmentet igényelnek.
- Hangsúlyt kell fektetnünk a szabványos tanúsítványok használatára, hogy a nálunk használható tanúsítványt más rendszer is befogadhassa.

- Meg kell határoznunk, hogy a rendszer milyen biztonsági szintű tanúsítványokat fogad be, és hogyan győződik meg a tanúsítványok biztonsági szintjéről.
- Végig kell gondolnunk, hogy rendszerünk később hogyan fog kapcsolatba kerülni más PKI-re épülő rendszerekkel: hogyan fogjuk elfogadni a máshonnan származó tanúsítványokat, és hogyan állapítjuk meg a hozzájuk tartozó felhasználók jogosultságait.
- A tanúsítvány érvényessége mindössze annyit kellene hogy jelentsen, hogy a kulcs az adott személy birtokában van. Ne mossuk ezt össze a felhasználók jogosultságaival.
- Kerülnünk kell, hogy a tanúsítványban az alanyok nevén kívül bármilyen más adat is szerepeljen.

3.7. Összegzés

- A tanúsítvány egy megbízható fél által aláírt igazolás, miszerint egy nyilvános kulcs egy adott entitáshoz (a tanúsítvány alanyához) tartozik, azaz kizárólag az ő birtokában van a megfelelő magánkulcs.
- Léteznek aláíró, titkosító és autentikációs tanúsítványok, ezeket célszerű szétválasztani egymástól. Az elektronikus aláírásról szóló törvény tiltja, hogy az aláíró tanúsítványt (illetve magánkulcsot) aláíráson kívül bármi más célra használjuk.
- Regisztrációnak nevezzük, amikor egy hitelesítés-szolgáltató megállapítja a tanúsítványt igénylő kilétét. A hitelesítés-szolgáltató általában ekkor határozza meg az alany tanúsítványba kerülő megnevezését is.
- A tanúsítványban az alany megnevezése (DN) szerepel, ez alapján jellemzően nem állapíthatóak meg az alany személyes adatai.
- A hitelesítés-szolgáltató visszavonja vagy felfüggeszti a tanúsítványt, ha tudomására jut, hogy a tanúsítványhoz tartozó magánkulcs illetéktelen kezekbe került, vagy ha a tanúsítványban szereplő valamely adat megváltozott. A visszavont tanúsítvány soha többet nem lehet érvényes, a felfüggesztett tanúsítvány még visszaállítható.
- Tanúsítvány ellenőrzésekor általában felépítünk egy tanúsítványláncot a kérdéses tanúsítványtól egy megbízható gyökértanúsítványig, majd megvizsgáljuk a lánc minden tanúsítványának érvényességét.
- A tanúsítványt ellenőrző érintett félnek nagyon kevés kötelezettsége van, lényegében bármilyen módon eljárhat a tanúsítvány elfogadása során. Ennek ellenére célszerű

a szabványok szerint, a hitelesítés-szolgáltató ajánlásait követve eljárnia, mert a hitelesítés-szolgáltató csak ekkor felel a tanúsítványért.

- Az aláíró, titkosító és autentikációs tanúsítványok használata gyökeresen eltér egymástól.

4. fejezet

Hitelesítés-szolgáltató

„Hozzánk legfeljebb tankkal tudnának behatolni, arra pedig még nem volt példa."

– Macskafogó

A nyilvános kulcsú infrastruktúra világának azon szereplőit nevezzük hitelesítés-szolgáltatóknak, akik igazolhatják, hogy kihez tartozik egy adott nyilvános kulcs, azaz ki birtokolja a hozzá tartozó magánkulcsot. A tanúsítvány a hitelesítés-szolgáltató által kibocsátott, aláírt igazolás, amely egy nyilvános kulcs mellett annak a megnevezését is tartalmazza, aki a hozzá tartozó magánkulcsot birtokolja.

A nyilvános kulcsú infrastruktúra minden egyes művelete arra épül, hogy az érvényes tanúsítvány alapján biztosak lehetünk benne, hogy egy adott nyilvános kulcs valóban a tanúsítvány alanyához tartozik, így a hitelesítés-szolgáltatók biztonsága alapvetően meghatározza a nyilvános kulcsú infrastruktúra biztonságát.

A hitelesítés-szolgáltató a következő főbb feladatokat látja el:

- azonosítja a tanúsítványt igénylő felet (például úgy, hogy elkéri a személyi igazolványát), illetve közhiteles adatbázisban ellenőrzi a tanúsítványt igénylő fél adatait;
- meghatározott időre bocsátja ki a tanúsítványokat;
- még a lejárta előtt visszavonja a tanúsítványt, ha tudomására jut, hogy valamely, a tanúsítvány által igazolt tény már nem áll fent. (Például ha tudomására jut, hogy illetéktelen fél is megismerte a tanúsítványhoz tartozó magánkulcsot, vagy hogy megváltozott a tanúsítvány alanyának a megnevezése.)

Ha a hitelesítés-szolgáltató hibát követ el (például nem a megfelelő személynek bocsát ki egy tanúsítványt, vagy nem dolgoz fel elég gyorsan egy visszavonási kérelmet), azzal kárt okozhat, és ekkor meg kell térítenie az okozott kárt. Az elektronikus aláírásra szolgáló

tanúsítványok esetén e kérdéskört részletesen szabályozza az elektronikus aláírásról szóló törvény. A hitelesítés-szolgáltatók felelősségbiztosítással[1] szoktak rendelkezni, és hatóságok vagy más auditorok rendszeresen felülvizsgálják a tevékenységüket.

Az elektronikus aláírásról szóló törvény a hitelesítés-szolgáltató fogalmat kizárólag az elektronikus aláírásra szolgáló tanúsítványokat kibocsátó szolgáltatókra használja, a nem aláírásra szolgáló tanúsítványokról nem szól a törvény, így a benne szereplő követelmények nem vonatkoznak például a webszerver tanúsítványokra és az őket kibocsátó szolgáltatókra.

A továbbiakban a „hitelesítés-szolgáltató” kifejezést nem kizárólag az Eat. szerinti szolgáltatókra használjuk, hanem – az X.509 specifikációnak megfelelően – minden, tanúsítványt kibocsátó entitást hitelesítés-szolgáltatónak nevezünk.

Elvileg bárki bármikor kijelentheti, hogy egy adott nyilvános kulcs egy adott személyhez tartozik. Egyáltalán nem biztos, hogy más is hisz e kijelentésnek. Attól válik valaki professzionális hitelesítés-szolgáltatóvá, hogy mások is hisznek neki, azaz elfogadják a tanúsítványait különböző helyzetekben, alkalmazásokban vagy jogi helyzetekben.

Nem minden hitelesítés-szolgáltató minden tanúsítványa alkalmas minden célra. Például egy Eat. szerinti, minősített hitelesítés-szolgáltató által kibocsátott tanúsítvány egyáltalán nem biztos, hogy használható webes környezetben (sőt egy minősített tanúsítvány garantáltan nem használható webszerver tanúsítványként). Ezen állítás fordítottja is igaz: ha egy hitelesítés-szolgáltató tanúsítványait elfogadják a nagy böngészőprogramok, abból nem következik, hogy a tanúsítvány alapján Eat. szerinti elektronikus aláírást lehetne létrehozni (sőt például webszerver tanúsítvány alapján nem lehet aláírni).

A hitelesítés-szolgáltató mindig valamilyen szabályrendszer szerint bocsátja ki a tanúsítványt, és fel is tünteti a tanúsítványban, hogy milyen szabályrendszer szerint bocsátotta ki. Így a tanúsítványt ellenőrző fél tisztában lehet vele, hogy mire számíthat a tanúsítvánnyal kapcsolatban. *Hitelesítési rend*nek nevezzük azon szabálygyűjteményt, amely egy tanúsítvány felhasználásának feltételeit írja elő valamely közös biztonsági követelményekkel rendelkező csoport vagy valamely alkalmazások számára.

A hitelesítési rend (illetve a szolgáltató szolgáltatási szabályzata) leírja vagy meghivatkozza mindazon információkat, amelyeket egy tanúsítványt igénylő vagy elfogadó félnek tudnia érdemes a szolgáltatóról és a tanúsítványról.

E fejezet hátralévő részében *hitelesítési rend RFC 3647 specifikációban leírt struktúrája szerint tekintjük át a hitelesítés-szolgáltatókkal kapcsolatos főbb kérdéseket.*

[1] Csak az Eat. szerinti, azaz aláíró tanúsítványokat kibocsátó hitelesítés-szolgáltatók esetén van ilyen előírás.

4.1. Közzététel

4.1.1. Nyilvánosan és zárt körben működő hitelesítés-szolgáltató

A hitelesítés-szolgáltatók tanúsítvány kiállításával igazolják, hogy melyik nyilvános kulcs kihez tartozik. E tanúsítványok alapján más felek döntéseket hoznak: elfogadnak vagy elutasítnak egy elektronikus aláírást, titkosítanak egy dokumentumot, vagy beengednek rendszerükbe egy felhasználót. A tanúsítvány alapján hiszik el, hogy a tanúsítványban szereplő nyilvános kulcshoz tartozó magánkulcs a megfelelő fél birtokában van.

Lényeges, hogy egy hitelesítés-szolgáltató által kiállított tanúsítványokat nemcsak az adott szolgáltató ügyfelei használják fel. A tanúsítványokat jellemzően *a nyilvánosság számára* állítják ki, hogy bárki – bármely érintett fél – felhasználhassa őket. Az jelenti a nyilvános kulcsú infrastruktúra egyik fő erősségét, hogy elegendő egyetlen kulcspárral rendelkeznünk, és e kulcspárt – elvileg – bárhol, bárki felé használhatjuk.

Megjegyezzük, léteznek zárt körben működő hitelesítés-szolgáltatók, akik csak egy zárt közösség számára bocsátanak ki tanúsítványokat. Jó példa erre egy vállalaton belüli szolgáltató, amely csak a vállalat munkatársai számára szolgáltat tanúsítványokat. Ekkor lehet, hogy nem szempont, hogy bárki felhasználhassa az általa kibocsátott tanúsítványokat, de ekkor nem is lehet cél, hogy tanúsítványunkat, kulcspárunkat máshol is használhassuk. A zárt körű felhasználásra kibocsátott tanúsítványok alapján akár fokozott biztonságú elektronikus aláírás is készíthető, feltéve, hogy teljesülnek az elektronikus aláírásról szóló törvény által meghatározott, fokozott biztonságú elektronikus aláírásra vonatkozó követelmények[2]. [180] Minősített elektronikus aláírás csak minősített tanúsítvány alapján hozható létre, amelyet csak a Nemzeti Média- és Hírközlési Hatóság nyilvántartásában – vagy hasonló külföldi nyilvántartásban – szereplő hitelesítés-szolgáltató bocsáthat ki.

> ***4.1. Példa:*** *Alajos és Bendegúz a Kókler Bt. alkalmazottai, mindketten megbíznak a Kókler Bt. belső hitelesítés-szolgáltatójában, és hitelesen hozzájutottak annak nyilvános kulcsához. Így műszaki szempontból számukra e megoldás megfelelő biztonságot jelent, és ha (pl. szerződésben) elfogadják, hogy a Kókler Bt. hitelesítés-szolgáltatója által kibocsátott tanúsítványok alapján fokozott biztonságú aláírás készíthető, a tanúsítványaik alapján létrehozott elektronikus aláírás fokozott biztonságú aláírásnak tekinthető. Ehhez az is szükséges, hogy a Kókler Bt. szolgáltatója az Eat. szerint működjön, és Alajos és Bendegúz is az Eat-nek megfelelően használják tanúsítványaikat, magánkulcsaikat. (Pl. az aláírásra használt magánkulcsot nem használhatják SSL bejelentkezésre.)*

[2]Az Eat. 2 § 15. értelmében a fokozott biztonságú aláírás alkalmas az aláíró azonosítására, egyedülállóan az aláíróhoz köthető, olyan eszközökkel hozták létre, amelyek kizárólag az aláíró befolyása alatt állnak, és az aláírás elhelyezését követően az aláírt dokumentumon minden módosítás érzékelhető.

Cili nem a Kókler Bt. dolgozója, nem ismeri a Kókler Bt. belső szolgáltatóját. Ha Alajos tanúsítványával találkozik, nem tudja megállapítani, hogy a rajta lévő aláírás hiteles, így azt sem tudja eldönteni, hogy e tanúsítványt nem a gonosz Manfréd készítette-e. Ha megszerzi hitelesen a szolgáltató gyökértanúsítványát, akkor technikailag már képes ellenőrizni Alajos tanúsítványát. Ekkor továbbra is kérdés, hogy megbízik-e a Kókler Bt. szolgáltatójában. Ha meg is bízik benne, még nem feltétlenül tudja eldönteni, hogy az adott tanúsítvány alapján hozható-e létre fokozott biztonságú aláírás. Így míg az Alajos által létrehozott aláírás Bendegúz számára fokozott biztonságú aláírásnak tekinthető, Cili ezt nem feltétlenül tekintheti fokozott biztonságú aláírásnak. Feloldható a probléma, ha Cili és Alajos megállapodnak, hogy az ilyen és ilyen típusú aláírásokat elfogadják fokozott biztonságú aláírásnak, de e megoldás nehezen skálázható; minden szereplőnek külön-külön el kell fogadnia (pl. szerződésben) az adott típusú aláírásokat. Nyilvánosan működő, a hatósági nyilvántartásban szereplő szolgáltatók esetén e probléma nem áll fent.

Dezső üzemelteti a Kókler Bt. hitelesítés-szolgáltatóját. Így lehetősége van rá, hogy bármilyen tanúsítványt kibocsásson. Lehet, hogy az Alajos és Bendegúz közötti vitákban Dezső pártatlan félnek tekinthető, de amint valaki Dezsővel vagy magával a Kókler Bt-vel kerül vitába, már kérdés, hogy a Dezső által üzemeltetett szolgáltató valóban hiteles-e, azaz nem csalt-e Dezső. Így annak ellenére, hogy Dezső is és Alajos is a Kókler Bt. dolgozója, a Dezső és Alajos közötti vitában a Kókler Bt. belső szolgáltatója alapján ellenőrizhető aláírás nem feltétlenül és nem minden esetben jelent erős bizonyítékot.

A továbbiakban a nyilvánosan működő hitelesítés-szolgáltatókkal foglalkozunk, a csak zárt körben működő szolgáltatókkal nem. Feltételezzük, hogy a felek kulcspárjaikat, tanúsítványaikat minél több helyen, minél több célra szeretnék használni. Ha egy szolgáltató azt szeretné, hogy tanúsítványait bárki felhasználhassa, *közzé kell tennie* minden olyan információt, amely vagy a tanúsítványok ellenőrzéséhez szükséges, vagy ahhoz kell, hogy a tanúsítványt felhasználó érintett fél megállapíthassa, hogy milyen tanúsítványról van szó és az milyen szintű biztonságot jelent.

4.1.2. Tanúsítványok közzététele

A hitelesítés-szolgáltató – mint a PKI egyik alappillére – általában közzéteszi az általa kibocsátott tanúsítványokat. Így, ha egy érintett fél meg szeretné ismerni valakinek a nyilvános kulcsát, megkeresheti az illető tanúsítványát a hitelesítés-szolgáltató nyilvános tanúsítványtárából, és a tanúsítványból nyerheti ki a nyilvános kulcsot. (A tanúsítvány alapján

egyúttal meg is győződhet róla, hogy a nyilvános kulcs valóban a kérdéses személy nyilvános kulcsa.)

A hazai szabályozás értelmében a hitelesítés-szolgáltató nem hozhat automatikusan nyilvánosságra minden általa kibocsátott tanúsítványt; egy tanúsítvány akkor hozható nyilvánosságra, ha a tanúsítvány alanya ebbe beleegyezik. (A tanúsítvány az alanyhoz egyértelműen hozzárendelhető, így személyes adatnak minősül, és az adatvédelmi törvény értelmében személyes adat csak az adat tulajdonosának hozzájárulásával kezelhető. [179])

Bár ez ellentmond a nyilvános kulcsú infrastruktúra azon alapelvének, hogy a nyilvános kulcsok nyilvánosak (lehetnek), a gyakorlatban ez nem szokott problémát okozni. Aláírás és autentikáció esetén az érintett félnek nincs szüksége rá, hogy a hitelesítés-szolgáltató tanúsítványtárában keressen. A legtöbb aláírás-formátum – így pl. a XAdES vagy a PKCS#7 – esetén az aláíró csatolja tanúsítványát az aláíráshoz, ezért ha egy érintett fél aláírást ellenőriz, már eleve adott, hogy milyen tanúsítványt kell használnia az ellenőrzéshez. (Lásd: 6.4. fejezet.) A legtöbb autentikációs protokoll szerint – pl. SSL esetén – a felek elküldik egymásnak tanúsítványaikat, ezért nem kell a hitelesítés-szolgáltatóhoz fordulniuk, ha meg szeretnék tudni partnerük nyilvános kulcsát (10.3.2. fejezet).

E probléma egyedül titkosítás esetén nem oldható meg ilyen egyszerűen. Titkosítás esetén ugyanis nem a tanúsítvány alanya, hanem az érintett fél kezdeményez, amikor titkosított üzenetet küld egy címzettnek (aki egy tanúsítvány alanya). Így titkosítás esetén valóban problémát jelent, hogy a kezdeményező érintett fél hogyan jut hozzá a címzett nyilvános kulcsához. Ha ekkor a hitelesítés-szolgáltató tanúsítványtárában keresi a kérdéses tanúsítványt, a következő problémákkal találja szembe magát:

- Meg kell tudnia, hogy a címzettnek melyik hitelesítés-szolgáltatótól (vagy szolgáltatóktól) van tanúsítványa, illetve hol és hogyan érhető el a szolgáltató tanúsítványtára.
- Nem mindig könnyű eldönteni, hogy egy adott tanúsítvány pontosan kihez tartozik.

  4.2. Példa: *Kovács János titkos üzenetet szeretne küldeni Nagy Lajosnak. A hitelesítés-szolgáltató tanúsítványtárában három Nagy Lajos névre kiállított tanúsítványt talál, egyik* `CN=Nagy Lajos, L=Budapest, C=HU`*, a másik* `CN=Nagy Lajos, L=Debrecen, C=HU`*, a harmadik* `CN=Nagy Lajos, L=Kecskemét, C=HU`*. Kovács János nem tudja, hogy a keresett Nagy Lajos hol lakik, csak az e-mail címét ismeri, de az nem szerepel egyik tanúsítványban sem.*

- Láttuk, hogy egy személynek is sok tanúsítványa lehet (3.6. fejezet). A feladónak a megfelelő célra használt, megfelelő biztonsági szintű, érvényes tanúsítványt kellene kiválasztania.

Előfordul, hogy a felek egyeztetik, hogy ki mely titkosító tanúsítványát fogja használni a kommunikációhoz. Ennél az jelenthet hatékonyabb megoldást, ha a feladó LDAP szolgáltatás segítségével kéri le a címzett tanúsítványát. Amennyiben ez az ő saját vállalati LDAP-ja, amelyet a vállalat megfelelően töltött fel, a fenti problémák kezelhetőek. Ha ez egy hitelesítés-szolgáltató LDAP-ja, az csak a legelső problémát oldja meg, a másik kettő továbbra is fennáll.

A hitelesítés-szolgáltatók általában működtetnek nyilvános tanúsítványtárat, de ez inkább tradíció (illetve ebből származó előírás), a gyakorlatban sokszor igen-igen kicsi a jelentősége. A tanúsítványtár legtöbbször a szolgáltató honlapján, egy webes keresőfelületen keresztül érhető el, de sok szolgáltató LDAP-on is közzéteszi a tanúsítványtárat. Általában nem szerencsés, hogy a teljes adatbázis a szolgáltató minden felhasználójának minden tanúsítványával letölthető legyen. Így az egy kérdéssel lekérhető adatok mennyiségét korlátozni szokták.

4.1.3. Hitelesítési rend és szolgáltatási szabályzat közzététele

Láttuk, hogy sokféle tanúsítvány létezik, és az egyes tanúsítványfajtákra eltérő szabályok vonatkoznak. Például:

- bizonyos tanúsítványokat csak aláíráshoz szabad felhasználni, míg más tanúsítványok tetszőleges célra használhatóak;
- a minősített tanúsítványok alapján minősített aláírás (teljes bizonyító erejű magánokirat) is létrehozható, míg nem minősített tanúsítvány alapján csak fokozott biztonságú aláírás készíthető;
- bizonyos tanúsítványokhoz több milliós szolgáltatói felelősségvállalás tartozik, míg más tanúsítványok esetén a szolgáltató esetleg semmilyen felelősséget sem vállal;
- bizonyos tanúsítványok esetén bármikor hozzájuthatunk friss visszavonási információkhoz (például CRL-ekhez), míg más tanúsítványok esetén esetleg csak napok vagy hetek múlva szerezhetjük be az aktuális időpontra vonatkozó visszavonási információt;
- bizonyos tanúsítványok az alany valódi nevét tartalmazzák, míg más tanúsítványok álnevet is tartalmazhatnak;
- bizonyos tanúsítvány esetén az aláíró csak magánszemélyként hozhat létre aláírást, míg más tanúsítványok esetén az aláíró valamely szervezet képviseletében vagy valamilyen hivatalos minőségben (pl. közjegyző) ír alá;

Mielőtt felhasználunk egy tanúsítványt (illetve a benne lévő nyilvános kulcsot), fontos tisztában lennünk azzal, hogy milyen szabályok vonatkoznak a tanúsítványra, milyen szabályok szerint használhatjuk fel. (Például nem szerencsés több millió forintos döntést olyan

tanúsítvány alapján hozni meg, amely az őt kibocsátó hitelesítés-szolgáltató szerint legfeljebb tízezer forintos tranzakciókban használható.)

> *Eat. 2. § „ 23. Hitelesítési rend: olyan szabálygyűjtemény, amelyben egy szolgáltató, igénybe vevő vagy más személy (szervezet) valamely tanúsítvány felhasználásának feltételeit írja elő igénybe vevők valamely közös biztonsági követelményekkel rendelkező csoportja, illetőleg meghatározott alkalmazások számára. ”*

A *hitelesítési rend* (CP, certificate policy) egy olyan dokumentum, amely tanúsítványok felhasználására vonatkozó szabályokat, kikötéseket stb. tartalmaz. Minden tanúsítványban (a tanúsítvány `certificate policies` mezejében) szerepel, hogy az adott tanúsítvány milyen hitelesítési rendeknek felel meg, tehát a hitelesítés-szolgáltató milyen követelményrendszer szerint bocsátotta ki, milyen szabályok szerint vonja vissza, mekkora felelősséget vállal érte stb. Egy tanúsítvány egyszerre akár több hitelesítési rendnek is megfelelhet, több szabályhalmaz is vonatkozhat rá. A tanúsítványok `certificate policies` mezejében szerepel azon hitelesítési rendek OID-je, amelyeknek az adott tanúsítvány megfelel. (A Windows tanúsítvány irányelv néven jeleníti meg e mezőt.)

> **Megjegyzés:** A hitelesítési rendek általában nyilvános dokumentumok, amelyek megtalálhatóak a szolgáltatók honlapján. Ha egy tanúsítványban ismeretlen hitelesítési rend OID-t találunk, az Interneten rákereshetünk az OID-re, és így megtalálhatjuk a kérdéses dokumentumot.

A minősített hitelesítés-szolgáltatókra vonatkozó követelményeket a hazai jogszabályok (az elektronikus aláírásról szóló törvény és a 3/2005. IHM rendelet) valamint mértékadó nemzetközi specifikációk – a minősített hitelesítés-szolgáltatók tekintetében az ETSI TS 101 456, nem minősített hitelesítés-szolgáltatók tekintetében az ETSI TS 102 042 – határozzák meg. [180], [80], [47], [53]

Mielőtt elfogadunk egy aláírást, célszerű elolvasnunk az aláíró tanúsítványában meghivatkozott hitelesítési rendet, ez alapján tudhatjuk meg, hogy az adott tanúsítványt hogyan kell ellenőriznünk, mennyire bízhatunk meg benne, és mit jelentenek a benne szereplő információk.

A hitelesítési rendet nem feltétlenül hitelesítés-szolgáltatók készítik, alapvetően bárki, akár egy felhasználói közösség is meghatározhat hitelesítési rendeket. Például a magyar közigazgatás is meghatározott hitelesítési rendeket, és a közigazgatásban használható tanúsítványokat e hitelesítési rendeknek megfelelően kell kibocsátani, illetve csak az ezen hitelesítési rendek szerint kibocsátott tanúsítványok fogadhatóak el közigazgatási alkalmazásokban. [83]

Ennek ellenére a hazai gyakorlat jelenleg az, hogy minden hitelesítés-szolgáltató saját hitelesítési renddel is rendelkezik. A hitelesítési rendek jellemzően általános követelményeket fogalmaznak meg, és egy hitelesítési rendet több hitelesítés-szolgáltató is támogathat. Az azonos hitelesítési rendnek megfelelő tanúsítványokra – elvileg – azonos alapvető biztonsági követelmények vonatkoznak, így például áruk könnyen összehasonlítható. Nagyon sokfajta tanúsítvány létezik, a különböző követelményeknek megfelelő tanúsítványokhoz általában egészen más árak tartoznak. (Például egy minősített tanúsítvány – amelyet csak személyes találkozás során, csak biztonságos aláírás-létrehozó eszközön lévő magánkulcshoz bocsát ki a szolgáltató – jellemzően drágább, mint egy távolról is kibocsátható, nem minősített tanúsítvány.)

A hitelesítés-szolgáltatók ún. szolgáltatási szabályzatot készítenek, amelyben leírják, hogy milyen hitelesítési rendeket vállalnak fel, és hogyan, milyen módon felelnek meg a bennük szereplő követelményeknek. Optimális esetben az általános követelményeket a hitelesítési rend fogalmazza meg, a szolgáltatási szabályzatban pedig csak ezek részletes megvalósítása szerepel egyes szolgáltató-specifikus elemekkel (pl. a CRL-ek elérésére vonatkozó URL-ekkel) együtt. A gyakorlatban a hitelesítési rend sokszor csak annyit tartalmaz, hogy az adott típusú követelmények a szolgáltatási szabályzatban szerepelnek, így néha a szolgáltatási szabályzat is tartalmaz bizonyos alapvető követelményeket.

A hitelesítési rendek és szolgáltatási szabályzatok általában egy rögzített struktúrát követnek, amelyet az RFC 3647 határoz meg. Így a különböző szolgáltatók dokumentumai – és a bennük vállalt biztonsági követelmények – könnyen összehasonlíthatóak egymással. [145]

4.1.4. Szolgáltatói tanúsítványok közzététele

A hitelesítés-szolgáltató aláírja az általa kibocsátott tanúsítványokat, így bárki meggyőződhet azok hitelességéről. A szolgáltató aláírását a szolgáltató tanúsítványa (pontosabban: nyilvános kulcsa) alapján lehet ellenőrizni. A szolgáltató tanúsítványán lévő aláírás általában más szolgáltató tanúsítványa alapján ellenőrizhető, és végül e lánc – az ún. tanúsítványlánc (5. fejezet) – egy megbízható gyökértanúsítványra vezethető vissza.

A hitelesítés-szolgáltatás csak akkor használható, ha az érintett felek *hitelesen* hozzájuthatnak a szolgáltató nyilvános kulcsához. Ezért a legtöbb hitelesítés-szolgáltató közzéteszi a szolgáltatói tanúsítványait, ezek közül is legalább a gyökértanúsítványokat.

A legtöbb szolgáltatónak saját gyökere van, vagy legalábbis működtet saját gyökeret is. Nagyon sok szolgáltató nem közvetlenül a gyökérrel bocsátja ki a végfelhasználói tanúsítványokat, hanem a gyökérrel csak alegységei számára bocsát ki tanúsítványt, és a végfelhasználók tanúsítványait alegységeivel írja alá. E problémakört a későbbiekben részletesebben is körüljárjuk (5. fejezet). Szintén gyakori, hogy egy szolgáltató több gyökeret működtet, és a különböző gyökerek szerint különböző célra szolgáló vagy különböző biztonsági

szintű tanúsítványokat bocsát ki. Lényeg, hogy egy szolgáltatónak több, esetleg akár sok szolgáltatói tanúsítványa és gyökértanúsítványa lehet.

Mit jelent egy gyökértanúsítvány? A szolgáltató generált egy kulcspárt, választott magának valamilyen megnevezést, és a saját kulcspárával saját maga számára bocsátott ki egy tanúsítványt. Fontos, hogy *a gyökértanúsítvány önmagában nem hiteles.* Bárki, bármikor készíthet magának gyökértanúsítványt, és bármilyen adatokat elhelyezhet benne. Csak azt a gyökértanúsítványt szabad megbízható gyökérként használni, amelyről tudjuk, hogy ki birtokolja a hozzá tartozó magánkulcsot, és valóban megbízunk az illetőben.

A PKI-t leíró szabványok az ún. *trust anchor* (bizalmi horgony) fogalmát használják a megbízható kulcspárra és hozzá kapcsolódó névre, akikre a tanúsítványokat visszavezetjük. E trust anchor nem feltétlenül önhitelesített tanúsítvány, de a gyakorlatban így szokták terjeszteni. A gyökértanúsítványban csekély a jelentősége a rajta lévő aláírásnak, a hangsúly a nyilvános kulcson és a hozzá kapcsolódó megnevezésen (distinguished name) van, illetve azon, hogy a magánkulcsot birtokló szervezetben megbízunk.

A PKI nyújtotta biztonság mindig megbízható gyökértanúsítványokra (illetve bennük szereplő nyilvános kulcsokra) vezethető vissza, ezért különösen fontos, mely gyökereket tekintünk megbízhatónak, és valóban hitelesen szerezzük-e be a megbízható gyökereket. A gyökerek hitelessége ugyanis nem ellenőrizhető PKI alapon.

A szolgáltatói (gyökér)tanúsítványok többféle módon tehetőek közzé:

- A hitelesítés-szolgáltató a honlapján teszi közzé a gyökértanúsítványt. E gyakran alkalmazott megoldás azt a kérdést veti fel, hogy ekkor mitől hiteles az az információ, amit egy érintett fél letölt a szolgáltató honlapjáról. A PKI lehetővé teszi, hogy titkosított és hitelesített kapcsolatot létesítsünk egy webszerverrel, de e titkos és hiteles kapcsolat egy tanúsítványra, tanúsítványláncra és végül egy gyökértanúsítványra vezethető vissza. Rossz megoldás, ha egy gyökértanúsítványt olyan csatornán keresztül szerzünk be, amelynek hitelessége ugyanezen gyökértanúsítványra épül.

  4.3. Példa: *Az XYZ HSZ kizárólag a honlapján teszi közzé gyökértanúsítványát. Alajos úgy jut hozzá a gyökértanúsítványhoz, hogy beírja böngészőjébe az XYZ HSZ weboldalának címét (`http://...`) , és letölti az oldalról a gyökértanúsítványt. Innentől kezdve az így letöltött gyökértanúsítvány alapján épít ki hiteles (SSL) kapcsolatot webbankjával.*

  Alajost átverték: Manfréd, a támadó eltérítette a HTTP kapcsolatot, és a saját weboldalára irányította Alajost. E weboldal tökéletes mása volt az XYZ HSZ weboldalának, de egy másik gyökértanúsítványt küldött el Alajosnak. E csaló gyökér az XYZ HSZ adatait tartalmazta, de más nyilvános kulcs szerepelt benne. A hozzá tartozó magánkulcsot Manfréd birtokolta. Amikor Alajos legközelebb SSL kapcsolatot létesít a bankjával, a böngésző ellenőrizni

fogja a bank tanúsítványát. Nem fogja észrevenni, hogy Manfréd eltérítette a bank felé indított (már HTTPS) kapcsolatát, mert a csaló bank (amit Manfréd működtet) tanúsítványát (amit Manfréd bocsátott ki) Alajos érvényesnek fogja tekinteni a csaló gyökér alapján. A csaló bank weboldala tökéletes mása lesz az igazi weboldalnak, sőt még továbbítja is az adatokat az igazi bank felé. Amint Alajos belép a webbankba a jelszavával (és esetleg beírja a banktól érkező, SMS-ben küldött kódot), Manfréd átveszi az irányítást, és átutalja Alajos megtakarításait a saját számlájára.

Nem attól rossz a fenti megoldás, hogy a felhasználó weben tölti le a gyökértanúsítványt. Az vele a probléma, hogy nem hiteles forrásból származik a gyökértanúsítvány, és Alajos nem ellenőrzi a hitelességét.

4.4. Példa: *A Csigaidomító Egyetem saját belső hitelesítés-szolgáltatót működtet, tanszéki laborjai számára ez bocsátja ki a tanúsítványokat. E belső hitelesítés-szolgáltató weblapja számára vásárolt egy darab tanúsítványt egy olyan szolgáltatótól, akinek a gyökere eleve szerepel Alajos böngészőprogramjában.*

Alajos HTTPS kapcsolatot épít ki a `hsz.csigaidomitoegyetem.hu` *weboldallal, e kapcsolat biztonsága e másik szolgáltató gyökerére épül, így ez biztonságos. Alajos így hitelesen tölti le a Csigaidomító Egyetem hitelesítés-szolgáltatójának gyökértanúsítványát, és e gyökér alapján már valóban biztonságos kapcsolatot építhet ki az éti csigák idomításával foglalkozó tanszék weblapjával. Bár e megoldás kényelmetlen Alajos számára, biztonságilag nincs benne kivetnivaló.*

Szintén jó lehet e megoldás, ha Alajos – bár HTTP-n keresztül tölti le a gyökeret – egy független csatornán ellenőrzi a gyökér hitelességét.

4.5. Példa: *Alajos HTTP kapcsolaton keresztül tölti le az XYZ HSZ honlapjáról a gyökértanúsítványt, majd megnézi az XYZ HSZ telefonszámát a telefonkönyvben, és felhívja a szolgáltatót. A szolgáltató munkatársával telefonon keresztül egyezteti a gyökér lenyomatát. (Hiba lett volna, ha Alajos a szolgáltató honlapjáról veszi a telefonszámot. Ha Manfréd eltéríti a szolgáltató honlapját, lecserélheti rajta a telefonszámokat, így ekkor Alajos lehet, hogy nem is a szolgáltató munkatársával, hanem éppen Manfréddal beszél.)*

- A hitelesítés-szolgáltató a médiában – pl. újságban, folyóiratban – teszi közzé a gyökértanúsítványa lenyomatát. Bár a gyakorlatban nem reális, hogy valaki egy sok évvel ezelőtti folyóiratot ás elő, hogy megállapítsa egy gyökér hitelességét, e megoldást

gyakran alkalmazzák, részben hagyománytiszteletből. Mindenesetre ekkor megvan rá a lehetőség, hogy valaki utólag, a szolgáltatótól független forrásból állapítsa meg, hogy egy adott gyökértanúsítvány hiteles-e.

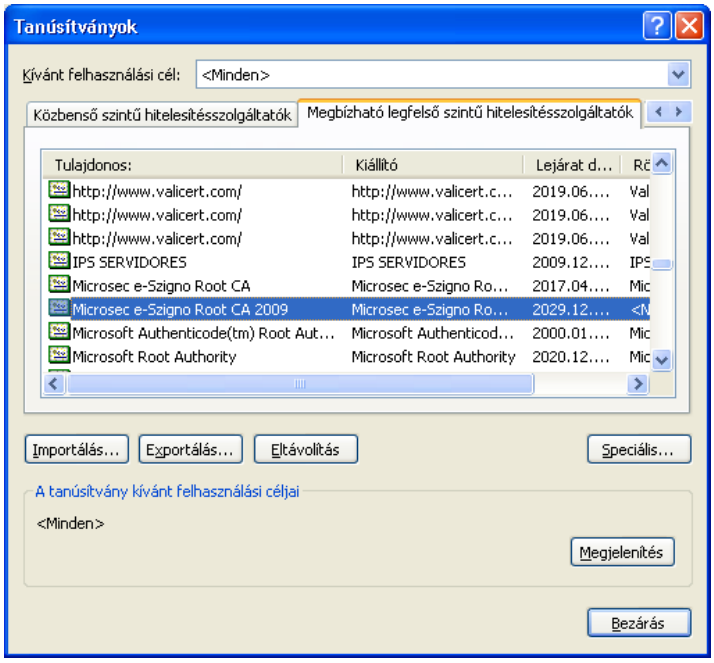
- A hitelesítés szolgáltató személyes találkozás során adja át ügyfeleinek a gyökértanúsítványát. E remek megoldás szerint a szolgáltató ügyfelei valóban biztonságosan kommunikálhatnak egymással. Ugyanakkor ettől a szolgáltató még nem tud nyilvánosan működni, ez nem oldja meg azt a problémát, hogy a harmadik felek, akik nem állnak kapcsolatban a szolgáltatóval, hogyan jussanak hozzá hitelesen a szolgáltató gyökeréhez.

- A hitelesítés-szolgáltató gyökértanúsítványát eleve tartalmazzák az elterjedt szoftverek (böngészők, levelezőprogramok). Ez a megoldás használható legjobban a gyakorlatban. Ez azt igényli, hogy a szolgáltató külön megegyezzen az egyes alkalmazások fejlesztőivel, hogy az alkalmazás eleve tartalmazza a szolgáltató gyökértanúsítványát. Ekkor az alkalmazás fejlesztője általában megköveteli, hogy a hitelesítés-szolgáltató meghatározott auditot teljesítsen, ezen audit alapján fogadja el, hogy a szolgáltató valóban megbízható és biztonságos, és nem veszélyezteti az alkalmazás felhasználóit (4.7. fejezet). Ezen együttműködés előnyös mind a hitelesítés-szolgáltatónak (mert több alkalmazás, és így több felhasználó fogadja el), mind az alkalmazás fejlesztőnek (mert több felhasználó tudja könnyen használni az alkalmazást, vagy több weboldallal működik az alkalmazás stb).

  A Windows operációs rendszereknek van egy központi tanúsítványtára (lásd: 4.1. ábra), sok szolgáltató eleve szerepel e tanúsítványtárban. Sok windowsos alkalmazás – pl. Internet Explorer, Outlook stb. – ezt a tárat használja, így alapértelmezetten elfogadja az ott szereplő gyökereket. A Microsoft Windows frissítésekkel tartja karban e tanúsítványtárat, így az új szolgáltatók automatikusan megjelennek benne, a régiek pedig automatikusan eltűnnek.

  Egy másik elterjedt tanúsítványtár az NSS (Network Security Services) szoftverkönyvtár, amelyet például a Mozilla alkalmazások is használnak. Az NSS tanúsítványtárát a Mozilla közösség tartja karban, az NSS frissítéseivel jelenhetnek meg, illetve tűnhetnek el alapértelmezett szolgáltatók, és ezek pl. Mozilla frissítésekkel kerülhetnek fel a felhasználók számítógépeire. Ezen kívül más tanúsítványtárak is léteznek, saját tanúsítványtárral rendelkezik például az OpenSSL, a Java stb.

  Itt arra kell felhívni a figyelmet, hogy egyáltalán nem biztos, hogy egy adott szoftvergyártó által elfogadott tanúsítványok valóban elfogadhatóak a végfelhasználó számára is egy adott célra. Például a magyar elektronikus aláírás törvény szerint működő szolgáltatók közül nem mindegyik szerepel a fenti tanúsítványtárakban. Ugyanakkor e szoftverek gyártói főként amerikaiak, és az EU-s, minősített aláírásokra vonatkozó



4.1. ábra. **Megbízható gyökértanúsítványok a Windows tanúsítványtárában**

követelményeknél lényegesen gyengébb követelményeknek megfelelő szolgáltatók is megjelenhetnek a tanúsítványtárakban az alapértelmezetten elfogadott gyökerek között.

Előfordult már e tanúsítványtárakban nyilvánvalóan hibás[3] tanúsítvány is, és derült már ki olyan is, hogy egy – a legtöbb szoftver által elfogadott – szolgáltató minden ellenőrzés nélkül bocsátott ki tanúsítványokat, [124] vagy már nyilvánvalóan elavult algoritmust használt, és ezzel veszélyeztette az őt elfogadó érintett feleket. [76]

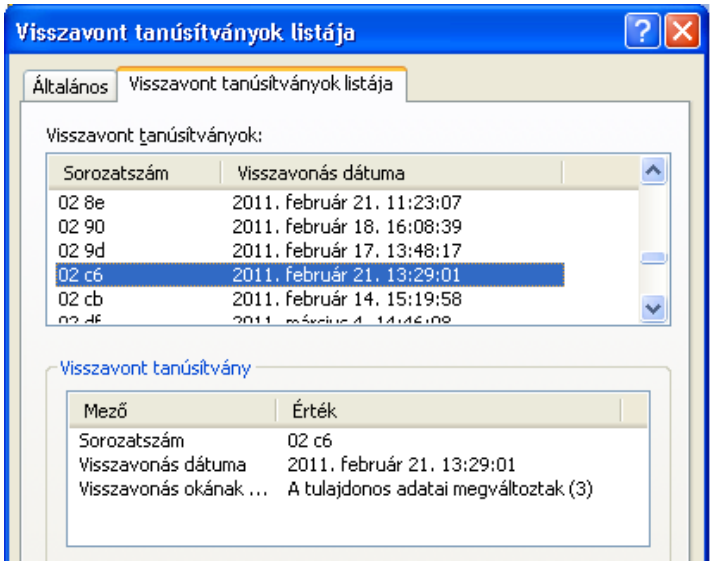
Egy adott célra kifejlesztett rendszerben – például egy ügyviteli rendszerben – általában saját tanúsítványtárat célszerű használni, és ott célszerű mérlegelni, hogy az adott célra pontosan mely gyökértanúsítványok (és mely tanúsítványláncok, mely hitelesítési rendek) fogadhatóak el.

4.1.5. Visszavonási információk közzététele

Az *érvényes* tanúsítvány igazolja, hogy egy nyilvános kulcs egy végfelhasználóhoz tartozik, és csak az adott végfelhasználó birtokolja a tanúsítványhoz tartozó magánkulcsot. Ha ez megváltozik, vissza kell vonni a tanúsítványt.

A hitelesítés-szolgáltató közzéteszi, hogy mely tanúsítványok nem érvényesek, így egy érintett fél meg tudja állapítani, hogy érvényes-e egy tanúsítvány. A hitelesítés-szolgáltató nem magukat a tanúsítványokat teszi közzé – hiszen maga a tanúsítvány nem is feltétlenül nyilvános

[3]Például olyan tanúsítvány, amelyben az RSA nyilvános kulcsban szereplő publikus exponens értéke „1” (amiből az következik, hogy a magánkulcs értéke is „1”).



4.2. ábra. **A visszavonási listán a hitelesítés-szolgáltató az érvénytelen tanúsítványok sorozatszámát és a visszavonás időpontját teszi közzé.**

(4.1.2. fejezet) –, hanem csak olyan információt publikál, amelyből megállapítható, hogy egy adott tanúsítvány érvényes-e. A visszavonási állapot közzétételének egyik módja a visszavonási lista (certificate revocation list – CRL), a másik az online tanúsítvány-állapot protokoll (online certificate status protocol – OCSP).

4.1.5.1. Visszavonási lista (CRL)

A visszavonási listán a hitelesítés-szolgáltató az érvénytelen tanúsítványok sorozatszámát és a visszavonás időpontját teszi közzé (lásd: 4.2. ábra). A CRL azt is tartalmazza, hogy milyen időpontra vonatkozik (`thisUpdate`), és (legkésőbb) mikor fog megjelenni a következő visszavonási lista (`nextUpdate`). A visszavonási listát általában a hitelesítés-szolgáltató a saját magánkulcsával írja alá. Ha egy hitelesítés-szolgáltató több hitelesítő egységet üzemeltet, általában minden egység saját visszavonási listát bocsát ki, e listát a saját magánkulcsával írja alá, és minden visszavonási lista az adott hitelesítő egység által kibocsátott tanúsítványokra vonatkozik.

A következő problémák merülhetnek fel CRL-ekkel kapcsolatban:

- Ha sok tanúsítványt kell visszavonni, a CRL nagyon nagyra nőhet. Ha minden érintett fél mindig letölti a legfrissebb CRL-t, az könnyen jókora hálózati forgalmat generálhat.

> ***4.6. Példa:*** *A CRL mérete 1 Megabyte, és percenként 1000 felhasználó tölti le a CRL-t. (Például mert percenként 1000 találatot kap egy HTTPS weboldal.) Ez percenként 1 Gigabyte forgalmat jelent a hitelesítés-szolgáltatónál.*

Ezt elkerülendő, különféle trükkökkel csökkenteni szokták a CRL-ek méretét, illetve arra szokták ösztönözni az érintett feleket, hogy ne töltsék le mindig a legfrissebb CRL-t, hanem a korábban letöltött (de még „kellően friss") CRL-t használják.

- Ha az érintett fél nem mindig tölti le a legfrissebb CRL-t, előfordulhat, hogy már létezne frissebb CRL, de ő azt nem veszi észre.

- A visszavonási listák csak bizonyos időközönként jelennek meg, így az érintett fél egy múltbéli időpontra vonatkozó igazolással rendelkezik a tanúsítvány visszavonási állapotáról. Ez aláíró tanúsítványok ellenőrzése esetén jelenthet problémát; aláírás ellenőrzésekor azt kell bizonyítanunk, hogy az aláíró tanúsítványa az aláírás pillanatában volt érvényes, és ezt nem bizonyítja kétségtelenül egy aláírás időpontja előtt készült CRL (6.5. fejezet). (Titkosító és autentikációs tanúsítványok ellenőrzésekor ez nem probléma, ott mindig a tanúsítvány aktuális visszavonási állapotára vagyunk kíváncsiak, így be kell érnünk[4] a legfrissebb elérhető visszavonási információval, ennél jobbat nem tehetünk.)

Részben a fenti problémák miatt nagyon sok különböző fajta CRL létezik. Bár ezek orvosolják a fenti problémák némelyikét, egyúttal új problémákat vezetnek be:

- A hitelesítés-szolgáltató kiveheti a lejárt tanúsítványokat a CRL-ből. Az RFC 5280 szerint, ha egy tanúsítvány lejárt, nem kell feltüntetni a CRL-ben, mert már úgyis érvénytelen. E gondolatmenet titkosító és autentikációs tanúsítványok esetén teljesen helyénvaló, de aláírás – különösen az EU-s gondolkodásmód szerinti minősített aláírás – ellenőrzése esetén a tanúsítvány múltbéli visszavonási állapotára vagyunk kíváncsiak. Így valóban előfordulhat, hogy arra volnánk kíváncsiak, hogy vajon visszavontak-e és mikor vontak vissza egy tanúsítványt, mielőtt az lejárt.

  4.7. Példa: *Alajos tanúsítványa csütörtökön jár le, de ő kedden még készít egy elektronikus aláírást. Az aláírt dokumentumot elküldi Bendegúznak. Bendegúz szabadságon van, csak pénteken kapja meg az aláírt dokumentumot. A hitelesítés-szolgáltató pénteki CRL-jéből nem tudja megállapítani, hogy kedden érvényes volt-e Alajos tanúsítványa. Ha Manfréd hétfőn ellopta volna Alajos kártyáját, és így hétfőn visszavonták volna Alajos tanúsítványát, az már nem derülne ki a pénteki CRL-ből. A pénteki CRL már nem tartalmaz releváns információt.*

  Általában nem állapítható meg a CRL-ből, hogy a lejárt tanúsítványokat bennehagyja-e a hitelesítés-szolgáltató. Az X.509 specifikációban szerepel ugyan erre egy `ExpiredCertsOnCRL` nevű CRL mező, de ezt a legtöbb szolgáltató nem használja, és

[4] Szélsőséges esetben előfordulhat, hogy a legfrissebb elérhető visszavonási információ sem a helyes eredményt mutatja.

a legtöbb alkalmazás nem kezeli. Megjegyezzük, hogy a hitelesítés-szolgáltató általában csak az aktuális CRL-t teszi közzé, a korábbi CRL-eket jellemzően nem. Hiába is tenne közzé a korábbi CRL-eket, nincs olyan szabványos mód, ahogy az érintett fél megtalálhatná őket.

- A CRL általában megadott időközönként jelenik meg, de soron kívüli CRL is kibocsátható. Előfordulhat, hogy egy hitelesítés-szolgáltató vállalja, hogy visszavonás esetén mindenképpen új CRL-t bocsát ki, de nem minden hitelesítés-szolgáltató vállal ilyen garanciát.

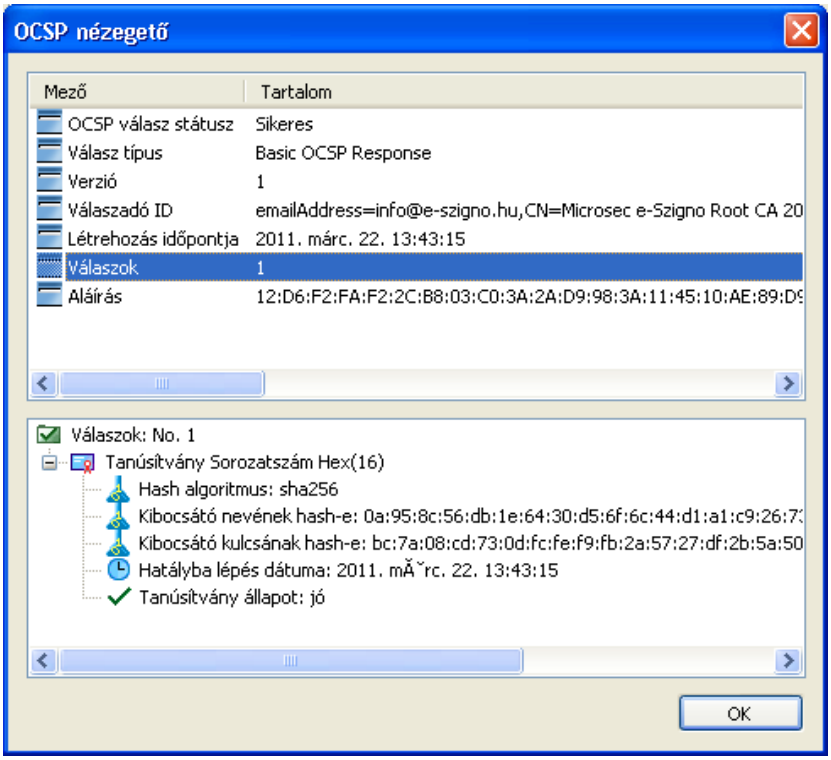
  Magából a CRL-ből nem derül ki, hogy a CRL csak periodikusan jelenik-e meg, vagy a hitelesítés-szolgáltató eseményvezérelten is bocsát ki CRL-t, és szintén nem derül ki, hogy a hitelesítés-szolgáltató visszavonás esetén vállalja-e és mennyi időn belül vállalja új CRL kibocsátását. Ezt csak az adott tanúsítványra vonatkozó hitelesítési rend alapján lehet megállapítani.

- Előfordulhat, hogy a visszavonási listát nem azzal a kulccsal írják alá, mint amelyikkel a hitelesítés-szolgáltató kibocsátja a tanúsítványokat, ekkor ún. *indirekt CRL*-ről beszélünk. (Például a hitelesítő egység egy alegységének is delegálhatja visszavonási listák kibocsátását, de az is előfordulhat, hogy egészen más tanúsítványláncban szereplő egység bocsátja ki a visszavonási listát.) [152]

- Az is előfordulhat, hogy egy visszavonási lista nem vonatkozik az adott egység által kibocsátott összes tanúsítványra. (Ez az Issuing Distribution Point kiterjesztéssel jelölhető a CRL-ben.) [152]

- A CRL általában az adott hitelesítő egység által kibocsátott összes visszavont tanúsítványt tartalmazza, de létezik ún. *delta CRL*, amely csak a legutóbbi teljes (azaz nem delta) CRL óta történt visszavonásokat tartalmazza. Egy hitelesítő egység naponta bocsát ki teljes CRL-t, de pl. fél óránként delta CRL-t is kibocsát, amely a legutolsó teljes CRL óta kibocsátott tanúsítványok sorozatszámát tartalmazza.

A CRL technológia előnye, hogy a legtöbb alkalmazás támogatja a CRL-ek legegyszerűbb változatait. Másrészt CRL segítségével egyetlen lekérdezéssel sok tanúsítvány visszavonási állapota lekérdezhető. A CRL-ek használata nagyon egyszerű a hitelesítés-szolgáltató részéről, mert a CRL-t közzétevő egységnek nem kell rendelkeznie a szolgáltatói magánkulccsal. Előfordulhat, hogy a hitelesítés-szolgáltató magánkulcsával rendelkező gép nincs hálózatra kötve, és offline módon generál CRL-t, és ezt a szolgáltató munkatársai adathordozón (pl. floppy-n vagy CD lemezen) viszik át a hitelesítés-szolgáltató webszerverét biztosító gépre. Ekkor a magánkulcsot tartalmazó, kritikus gép nincs a hálózaton, viszont több idő telik el a visszavonás és az új CRL megjelenése között.



4.3. ábra. **OCSP válasz megtekintése e-Szignó programmal**

Elektronikus aláírás használata esetén a CRL erős korlátokkal rendelkezik. Előfordulhat, hogy sokat kell várni egy releváns (az aláírás időpontjára vonatkozó) CRL megjelenésére, míg OCSP segítségével sokkal gyorsabban beszerezhető a szükséges bizonyíték (6.5. fejezet). Webszerver tanúsítványok esetén a CRL-ek nagy mérete okoz problémát, ott is az OCSP irányába terelik a felhasználókat, de ott nem a friss visszavonási információ szolgáltatása, hanem a hálózati forgalom csökkentése az OCSP célja.

4.1.5.2. Online tanúsítvány-állapot protokoll (OCSP)

Az online tanúsítvány-állapot protokoll (OCSP) során a lekérdező online kérdést küld a hitelesítés-szolgáltatónak (pontosabban egy OCSP válaszadónak), amire azonnali, online választ kap vissza. [140] Az OCSP kérés tartalmazza a kédéses tanúsítvány kibocsátója megnevezésének (DN) lenyomatát, a kibocsátó nyilvános kulcsának lenyomatát, valamint a kérdéses tanúsítvány sorozatszámát. (Lásd: 4.3. ábra.)

A hitelesítés-szolgáltató és az OCSP válaszadó nem feltétlenül ugyanaz, előfordulhat, hogy az A szervezet a B szervezet hitelesítés-szolgáltatójának tanúsítványaira válaszol. Gyakori megoldás, hogy egy szervezet saját OCSP válaszadót működtet, amely több hitelesítés-szolgáltató tanúsítványaira is válaszol.

4.8. Példa: *A Dinoszaurusz ZRt. kézben szeretné tartani, hogy munkatársai hogyan ellenőrzik a tanúsítványok visszavonási állapotát. Csak meghatározott hitelesítés-szolgáltatókat fogad el, megvizsgálta őket, és rendszeresen (pl. 5 percenként) összegyűjti a CRL-jeiket. Működtet egy olyan egységet, amely ezen szolgáltatók tanúsítványaira ad OCSP választ. Ha a Dinoszaurusz ZRt. egy dolgozója ellenőriz egy tanúsítványt, a céges OCSP válaszadó egységhez fordul. A céges OCSP válaszadó válasza alapján dönt a tanúsítvány érvényességéről.*

E megoldás webszerver tanúsítványok esetén jó lehet, de aláíró tanúsítványok esetén célszerű, ha a tanúsítványt kibocsátó hitelesítés-szolgáltató igazolja a tanúsítvány érvényességét, mert ekkor könnyebben felelősségre vonható egy adott válasszal kapcsolatban.

Az OCSP válaszadó háromféle választ adhat:

- „Good” – a tanúsítvány a válaszadó szerint nincs visszavonva.
- „Revoked” – a tanúsítványt ekkor és ekkor visszavonták.
- „Unknown” – a válaszadó nem tud a tanúsítványról.

Az OCSP-vel kapcsolatban az jelenti az egyik legnagyobb problémát, hogy a fenti három választ többféle módon is lehet értelmezni. Ennek az az oka, hogy sokféle módon lehet OCSP válaszadót működtetni, és az RFC 2560 egyik értelmes megoldást sem akarta kizárni.

Például a „good” válasz nem jelenti azt, hogy a kérdéses tanúsítvány valóban létezik, mindössze azt jelenti, hogy a válaszadó tudomása szerint nincs visszavonva. Ezáltal a hitelesítés-szolgáltató CRL-je alapján is adható OCSP válasz egy tanúsítványra: ha rajta van a tanúsítvány a CRL-en, akkor „revoked”, egyébként „good”. Ennek az a következménye, hogy helyesen működik az az OCSP válaszadó, amely egy nem létező – vagy egy általa ismeretlen, más szolgáltató által kibocsátott – tanúsítványra „good” választ ad.

4.9. Példa: *Alajos a digitális kamerájával videófelvételt készít a macskájáról, a felvételből lenyomatot készít, és elküldi a lenyomatot az OCSP válaszadónak, rákérdez az ilyen sorozatszámú tanúsítvány visszavonási állapotára. Lehet, hogy „good” választ kap rá, és az OCSP válaszadó szabványosan működik. [73]*

Szintén problémát jelenthetnek a lejárt és visszavont tanúsítványok. Tegyük fel, hogy egy tanúsítványt hétfőn visszavonnak, kedden lejár, és szerdán lekérdezzük az állapotát. Attól függ, hogy milyen választ kapunk, hogy a szabvány mely variációja szerint működik az OCSP válaszadó. A válasz lehet „good”, mert már nincsen rajta a CRL-en (ekkor a válaszadó a válaszban szereplő `archiveCutoff` kiterjesztéssel jelezheti, hogy a régi tanúsítványokra már

nem releváns a válasz). A válasz lehet „revoked”, mert visszavonták a tanúsítványt. A válasz lehet „unknown” is, mert a válaszadó nem tud a tanúsítványról (de ekkor is jelzi a válaszadó az `archiveCutoff`-ban, hogy a régi tanúsítványokról már nem tud).

Ha az érintett fél figyelembe veszi, hogy az adott OCSP válaszadó hogyan működik, akkor a fenti problémák nem jelentenek biztonsági kockázatot.

Az RFC 2560 szerint OCSP válaszadó többféle módon kapcsolódhat a hitelesítés-szolgáltató tanúsítványához:

- Egyik megoldás, hogy az OCSP válaszokat közvetlenül a végfelhasználói tanúsítványokat aláíró hitelesítő egység írja alá. Ennek előnye, hogy így egy érintett fél könnyen el tudja dönteni, hogy egy OCSP válasz egy tanúsítványhoz tartozik-e. Hátránya, hogy a hitelesítő egység magánkulcsát így nagyon intenzíven használjuk (ha nagyon intenzíven kell gyártani az OCSP válaszokat, akkor esetleg több berendezésben is ott kell tartani a kérdéses magánkulcsot), és a külvilágból érkező információt írnánk alá, esetleg automatikusan. Így egy támadó több információhoz juthat a hitelesítő egység kulcsáról, és esetleg könnyebben hozzáférhet a kulcshoz.

- Másik megoldás, hogy minden hitelesítő egység külön felhatalmaz egy (vagy több) OCSP válaszadót, és az így felhatalmazott OCSP válaszadók jogosultak válaszolni a hitelesítő egység által kibocsátott tanúsítványok visszavonási állapotával kapcsolatban. E felhatalmazás úgy történik, hogy a hitelesítő egység tanúsítványt bocsát ki ezen OCSP válaszadó részére, és a tanúsítványban feltüntetni az `ocspSigning` kiterjesztett kulcshasználatot. Az RFC 2560 e megoldást nevezi „authorized responder”-nek.

- A harmadik megoldás szerint bármilyen tanúsítvánnyal rendelkező OCSP válaszadó elfogadható, amiben az érintett fél megbízik. Az RFC 2560 „trusted responder”-nek nevezi az így működő OCSP válaszadót. Ennek leggyakoribb esete a fent leírt, belső, vállalati OCSP válaszadó példája.

  Súlyos hátránya e megoldásnak, hogy egy érintett fél, aki semmilyen kapcsolatban nincsen a hitelesítés-szolgáltatóval, nehezen tudja megállapítani, hogy egy adott OCSP válasz valóban vonatkozik-e egy adott tanúsítványra (vagy esetleg a válaszadó „good” válasza csak annyit jelent, hogy nem tud mit mondani a tanúsítványról).

Magyarországon korábban az volt az elterjedt megoldás, hogy a hitelesítés-szolgáltatók külön OCSP válaszadói tanúsítvány-hierarchiát működtettek, és így a „trusted responder” megoldás szerint válaszoltak a tanúsítványokra. E megoldással azt akarták elkerülni, hogy a minősített tanúsítványok kibocsátására használt magánkulccsal nem minősített tanúsítványt, OCSP válaszadói tanúsítványt bocsássanak ki. A szolgáltatók ekkor hitelesítési rendjükben (vagy szolgáltatási szabályzatukban) definiálhatták, hogy válaszadójuk hogyan működik, és mely tanúsítványokra ad releváns választ. Azóta az „authorized responder” megoldás látszik

elterjedni, talán ez látszik a legtisztább megoldásnak, ahol egy érintett fél könnyen meg tudja állapítani, hogy egy válasz valóban vonatkozik-e egy adott tanúsítványra.

Amennyiben az OCSP válaszadó külön tanúsítvánnyal rendelkezik (azaz nem a hitelesítő egység magánkulcsa írja alá az OCSP választ), felmerül a kérdés, hogy hogyan vizsgáljuk az OCSP válaszadó visszavonási állapotát. Egyik lehetőség CRL alapon vizsgálni, esetleg egy gyakran frissített, kizárólag OCSP válaszadói tanúsítványokra vonatkozó CRL alapján; de ekkor a CRL alapú ellenőrzés problémáival találjuk magunkat szemben. Másik lehetőség OCSP alapon vizsgálni; de ekkor ismét felmerül a probléma, hogy hogyan vizsgáljuk az OCSP válaszadó visszavonási állapotát. A harmadik, talán legelterjedtebb megoldás, hogy egyáltalán nem vizsgáljuk az OCSP válaszadó visszavonási állapotát. A hitelesítés-szolgáltató az `ocspNoCheck` kiterjesztéssel jelölheti a válaszadó tanúsítványában, hogy olyan OCSP válaszadói tanúsítványról van szó, amelyre nem vonatkozik visszavonási információ. Vegyük figyelembe, hogy ekkor PKI alapon *egyáltalán nem lehet visszavonni* az OCSP válaszadó tanúsítványát, így majdnem olyan helyzetben van, mintha gyökértanúsítványról lenne szó. Ezért írja az RFC 2560, hogy ekkor *rövid lejáratú OCSP válaszadói tanúsítványokat célszerű használni.*

4.10. Példa: *Az e-Szignó Hitelesítés Szolgáltató az „authorized responder" koncepció szerint működteti OCSP válaszadóit. Minden hitelesítő egysége külön OCSP válaszadói tanúsítványt bocsát ki. Az OCSP válaszadói tanúsítványok rövid lejáratúak, mindössze 10 percig érvényesek. Ha valamelyik OCSP válaszadó magánkulcsa kompromittálódna, úgy teszi közzé a megváltozott visszavonási állapotot, hogy nem bocsát ki több tanúsítványt a kompromittálódott kulcshoz.*

Amint a legutolsó érvényes OCSP válaszadói tanúsítvány lejár, a támadó már nem tud érvényes XAdES-A aláírást létrehozni a kompromittálódott kulccsal hamisított OCSP válaszok alapján:

- *A kompromittálódott kulcshoz az adott időpillanattól nem tartozik érvényes tanúsítvány, így a támadó nem képes előredátumozott, vagy az adott időpillanatra vonatkozó OCSP válaszokat hamisítani.*
- *A kompromittálódott kulccsal a támadó létre tud hozni érvényesnek látszó visszadátumozott OCSP választ, de nem képes rájuk olyan időbélyeget szerezni, amelyhez az adott kulcshoz tartozó, érvényes OCSP válaszadó tanúsítvány is tartozik, mert a támadónak visszadátumozott időbélyegre volna szüksége.*

Probléma esetén a hitelesítés-szolgáltató új kulcspárt generálhat az OCSP válaszadójának, az új kulcspárhoz új tanúsítványt bocsáthat ki. Az új tanúsítvány alapján aláírt OCSP válaszokat a felhasználók el fogják fogadni, lehet, hogy a felhasználónak egyáltalán nincsen teendője a kompromittálódott OCSP kulccsal.

OCSP szolgáltatás többféle megközelítés szerint nyújtható:

1. *Mindig friss OCSP szolgáltatás*, amelynek segítségével a lekérdező *friss bizonyítékhoz jut a tanúsítvány visszavonási állapotára vonatkozóan.* E bizonyítékkal később igazolhatja, hogy az adott tanúsítvány a lekérdezés időpontjában valóban érvényes volt. Ezen OCSP-nek elsősorban elektronikus aláírás ellenőrzésekor van szerepe, ha egy aláírást rövid idővel a létrehozása után szeretnénk ellenőrizni, és az ellenőrzés során olyan bizonyítékot szeretnénk begyűjteni, amely segítségével később *harmadik fél is meggyőződhet az aláírás érvényességéről.* (Lásd: 6.5. fejezet.)

   Az ilyen módon nyújtott OCSP szolgáltatásnak csak akkor van értelme, ha az alábbi feltételek mindegyike teljesül:

   a. *Az OCSP válasz „friss", azaz a lekérdezést követően, a lekérdezésre válaszul készül.*

      (Ha a hitelesítés-szolgáltató előre generált OCSP választ ad vissza, az lehet, hogy korábbi, mint a lekérdezés időpontja, és ekkor a válasz nem igazolja a tanúsítvány érvényességét a lekérdezés időpontjára vonatkozóan.)

      E követelmény az OCSP válaszra nézve azt jelenti, hogy a válaszban szereplő `producedAt`[5] és `thisUpdate`[6] értékek mindenképpen későbbiek, mint a lekérdezés időpontja. Ez jelentős terhelést jelent a hitelesítés-szolgáltató rendszerére, mert így lényegében minden egyes lekérdezésre külön-külön OCSP választ kell készítenie, és ezeket külön-külön alá kell írnia.

      Az OCSP lehetővé teszi, hogy a kérdésben a lekérdező egy ún. „nonce" értéket (pl. friss véletlen számot) tüntessen fel, és a válaszadó ugyanezen értéket szerepeltesse a válaszban. Ez – feltéve, hogy a válaszadó támogatja e megoldást – kikényszeríti, hogy a válasz friss legyen.

   b. *A hitelesítés-szolgáltató gyors visszavonás-kezelés szolgáltatást nyújt.* Ehhez egyrészt az szükséges, hogy a hitelesítés-szolgáltató gyorsan el tudja bírálni a felfüggesztési és visszavonási kérelmeket. (Ha a hitelesítés-szolgáltató csak órák alatt bírál el egy felfüggesztési kérelmet, lehet, hogy az aláíró már régen bejelentette magánkulcsa kompromittálódását, de a hitelesítés-szolgáltató OCSP szolgáltatása még mindig érvényesnek mondja a tanúsítványát.) Másrészt ehhez az is szükséges, hogy a hitelesítés-szolgáltató gyorsan át tudja vezetni a megváltozott visszavonási állapotot a visszavonási nyilvántartásában.

      A visszavonás-kezelés szolgáltatás sebessége önmagában nem derül ki az OCSP válaszból, csak a hitelesítés-szolgáltató szolgáltatási szabályzata alapján lehet megtudni. (A visszavonás-kezelés és a visszavonási állapot közzétételének együttes

[5] A válasz készítésének időpontja.
[6] A válasz a hitelesítés-szolgáltató visszavonási nyilvántartásának ezen időpontban vett állapotát tükrözi.

ideje tudható meg az OCSP válaszban szereplő `producedAt` és `thisUpdate` különbségéből.)

c. *A hitelesítés-szolgáltató gyorsan közzéteszi a visszavonási állapotot, és így az OCSP válasz az aktuális időpontra vonatkozik*, azaz a hitelesítés-szolgáltató visszavonási nyilvántartásának aktuális időpontra vett állapotát tükrözi. (Ha a hitelesítés-szolgáltató csak ritkán frissíti OCSP válaszadója adatbázisát, előfordulhat, hogy habár saját belső nyilvántartása szerint a tanúsítvány már visszavont állapotban van, az OCSP válaszadó még mindig érvényesnek mondja.)

E követelmény az OCSP válaszra nézve azt jelenti, hogy a válasz az aktuális időpontra vonatkozik, azaz a benne szereplő `producedAt` és `thisUpdate` értékek megegyeznek.

A mindig friss OCSP esetén a válaszban szereplő `nextUpdate` érték vagy nagyon közel áll a `producedAt` és `thisUpdate` értékekhez, vagy egyáltalán nem is szerepel a válaszban `nextUpdate`. Ezzel jelezhető, hogy a korábban lekérdezett választ nincs értelme cache-elni, minden időpontban frissebb és frissebb információ várható.

A mindig friss OCSP esetén bármely t időpillanatban lekérhető olyan OCSP válasz, amely igazolja a tanúsítvány érvényességét e t időpontra vonatkozóan. Ekkor kötelezően t<`producedAt` és t<`thisUpdate`. Az így lekért OCSP válaszból nemcsak a lekérdező, hanem harmadik fél is megállapíthatja a tanúsítvány érvényességét a t időpontra nézve. (Lásd: 6.5. fejezet.)

2. *A CRL kiváltására alkalmas OCSP szolgáltatás.* Ha az előző pontban szereplő követelmények nem teljesülnek, akkor az OCSP válasz nem feltétlenül igazolja harmadik fél számára a tanúsítvány érvényességét a lekérdezés időpontjára vonatkozóan. Az OCSP ekkor a CRL-hez hasonló tulajdonságokkal rendelkezik.

> ***4.11. Példa:*** *Tegyük fel, hogy egy hitelesítés-szolgáltató gyors visszavonás-kezelés szolgáltatást működtet, gyorsan teszi közzé a megváltozott visszavonási állapotot, de előre generált OCSP válaszokat bocsát ki.*
>
> *Alajos reggel 9 órakor OCSP alapján ellenőriz egy aláírást. Az aláírás a rajta lévő időbélyeg szerint aznap 8:59-kor készült. Az OCSP válasz szerint a tanúsítvány érvényes, de a válasz aznap reggel 8 órakor készült (`producedAt`=`thisUpdate` = aznap 8 óra). Alajos ebből meg tudja állapítani, hogy az aláírás 8:59-kor érvényes tanúsítvány szerint készült. (Ha érvénytelen lenne, a szolgáltató nem ezt az előre generált választ adta volna.)*
>
> *Alajos csatolja az OCSP választ az aláíráshoz, majd időbélyeggel látja el, és az így kapott – XAdES-X-L – aláírást 10 órakor küldi el Cilinek. Cili ebből nem tudja megállapítani, hogy Alajos valóban reggel 9 órakor kapta az*

adott OCSP választ. Lehet, hogy Alajos reggel 8 órakor kérte le a választ, 8:30-kor visszavonták a tanúsítványt, és az adott aláírás reggel 9 órakor már érvénytelen tanúsítvány szerint készült. Így az OCSP harmadik fél felé nem bizonyítja a tanúsítvány érvényességét a 8:59 időpontra nézve.

4.12. Példa: *Tegyük fel, hogy egy hitelesítés-szolgáltató lassú visszavonás-kezelés szolgáltatást működtet (4 óra alatt dolgoz fel egy visszavonási kérelmet), de gyorsan közzéteszi a megváltozott visszavonási állapotot, és friss OCSP válaszokat bocsát ki.*

Alajos reggel 9 órakor OCSP alapján ellenőriz egy aláírást. Az aláírás a rajta lévő időbélyeg szerint aznap 8:59-kor készült. Az OCSP válasz szerint a tanúsítvány érvényes, a válasz 9 órakor jön létre, de a szolgáltató nyilvántartásának hajnali 5 órakor vett állapotát tükrözi (azaz `producedAt` *= 9 óra,* `thisUpdate` *= 5 óra).*

Ez nem bizonyítja Alajos számára, hogy az aláírás 8:59-kor érvényes tanúsítvány szerint készült. (Lehet, hogy a tanúsítványt 7 órakor már visszavonták.) Alajos annyit tud, hogy hajnali 5 órakor még érvényes volt a tanúsítvány. Ez alapján vagy elfogadja az aláírást, vagy elutasítja, de úgy is tehet, hogy vár 12:59-ig, és ezen időpontot követően kér új OCSP választ.

4.13. Példa: *Tegyük fel, hogy egy hitelesítés-szolgáltató gyors visszavonás-kezelés szolgáltatást működtet, de lassan teszi közzé a megváltozott visszavonási állapotot (4 óra alatt frissíti az OCSP adatbázisát), és friss OCSP válaszokat bocsát ki.*

Alajos reggel 9 órakor OCSP alapján ellenőriz egy aláírást. Az aláírás a rajta lévő időbélyeg szerint aznap 8:59-kor készült. A lekérdező szempontjából ez az előző példában leírt esettel ekvivalens. A lekérdező számára teljesen lényegtelen, hogy a szolgáltató gyorsan feldolgozza a visszavonási kérelmet, és lassan teszi közzé az eredményét, vagy lassan dolgozza fel a kérelmet, és gyorsan teszi közzé az eredményét.

Az eset ebben a formában ritkán fordul elő. Akkor fordulhat elő, hogy egy szolgáltató OCSP-vel lassan teszi közzé a megváltozott visszavonási állapotot, ha pl. az OCSP válaszadója a CRL-jei alapján bocsát ki OCSP válaszokat, és a CRL csak ritkán frissül. Ekkor viszont akár előre generált OCSP válaszokat is kibocsáthatna, az sokkal hatékonyabb volna a számára.

Gyakori megoldások a CRL kiváltására használt OCSP esetén:

- Gyors visszavonás-kezelés, gyors visszavonási állapot közzététel, de előre generált OCSP válaszok. Ez akkor praktikus, ha csak a lekérdező használja fel az OCSP

válaszokat, és nem továbbítja őket harmadik félnek. Nem aláíró tanúsítványok esetében ez megfelelő megoldás. Webszerver tanúsítványok esetén különösen praktikus az így nyújtott OCSP szolgáltatás, mert ott könnyen előfordulhat, hogy egy tanúsítványra egyszerre sok felhasználó kérdez rá.

- Lassú visszavonás-kezelés, gyors visszavonási állapot közzététel, friss OCSP válaszok. Ez akkor lehet elfogadható, ha nem szükséges nagyon gyorsan ellenőrizni az aláírást. E megoldás segítségével egy aláírás érvényessége még a következő CRL megjelenése előtt is igazolható.

  4.14. Példa: *Az X hitelesítés-szolgáltató minden nap éjfélkor bocsát ki CRL-t. A szolgáltató 3 óra alatt feldolgoz egy visszavonási kérelmet, és 1 óra alatt közzéteszi a megváltozott eredményt mind CRL-lel, mind OCSP-vel.*

  Alajos aláírást ellenőriz, amelyen reggel 8 órai időbélyeg szerepel. A 8:10-kor lekért OCSP mindössze annyit mond, hogy 4:10-kor még érvényes volt a tanúsítvány. Ha Alajos 12 órakor ellenőrzi a tanúsítvány visszavonási állapotát, akkor tudja megállapítani, hogy a tanúsítvány 8 órakor is érvényes volt.

  Ha 12 órakor lekér egy OCSP választ, azt csatolhatja az aláíráshoz, és az így kapott XAdES-C vagy XAdES-X-L aláírásból Cili is megállapíthatja a tanúsítvány érvényességét a 8 órai időpontra vonatkozóan.

  Ha a szolgáltató 12 órakor kibocsát egy CRL-t, arra ugyanez lenne igaz. Csakhogy a szolgáltató valószínűleg nem pont 12 órakor bocsát ki CRL-t. Ha nem történik visszavonás, akkor a következő CRL lehet, hogy csak éjfélkor jelenik meg. Ha 12 órakor is csak az előző napi CRL érhető el, abból csak Alajos tudja megállapítani a tanúsítvány 8 órai érvényességét, ezt hiába csatolná az aláíráshoz, ez így Cilinek nem sokat mondana.

- Lassú visszavonás-kezelés, lassú visszavonási állapot közzététel, előre generált OCSP válaszok. E megoldás nem ad több információt, mint egy CRL. Mégis sokszor használják e megoldást, önmagában azért, mert ez kisebb forgalmat jelent a szolgáltatónak. Ez az OCSP legelterjedtebb változata.

  4.15. Példa: *Az X hitelesítés-szolgáltató CRL-je kb. 1 megabyte. Naponta 10 000 felhasználó tölti le a CRL-t, mindegyik csak egy-egy tanúsítvány ellenőrzése miatt, ez önmagában napi 10 gigabyte forgalmat jelent. Tegyük fel, hogy egy OCSP válasz mérete 3 kilobyte. Ha OCSP-vel kérdeznék le a visszavonási állapotot, az csak 30 megabyte forgalmat jelentene.*

Vegyük figyelembe, hogy extrém esetekben az OCSP nagyobb forgalmat is jelenthet.

4.16. Példa: *Az X hitelesítés-szolgáltató CRL-je kb. 1 megabyte. Naponta 10 000 felhasználó tölti le a CRL-t, de mindegyik 1 000 tanúsítványt ellenőriz a CRL-en. Ez napi 10 gigabyte forgalmat jelent. Tegyük fel, hogy egy OCSP válasz mérete 3 kilobyte. Ha OCSP-vel kérdeznék le a visszavonási állapotot, ez már 30 gigabyte forgalmat jelentene.*

4.2. Azonosítás és hitelesítés

A hitelesítés-szolgáltató által kibocsátott tanúsítvány azt igazolja, hogy egy adott nyilvános kulcs egy adott végfelhasználóhoz (a tanúsítvány alanyához) tartozik. Markánsan meghatározza a tanúsítvány biztonságát, hogy a hitelesítés-szolgáltató milyen módon azonosítja a végfelhasználót, és mennyire biztos benne, hogy ki az illető. A hitelesítési rend általában meghatározza, hogy az adott típusú tanúsítvány esetén mikor milyen azonosítás történik.

Több ponton azonosítja a szolgáltató a végfelhasználót:

- Regisztrációkor, azaz amikor a hitelesítés-szolgáltató a tanúsítvány kibocsátása előtt megállapítja az igénylő (személy, illetve szervezet) kilétét, személyazonosságát. (Lásd: 3.3.4. fejezet.) Az „erős” biztonságot nyújtó tanúsítványokhoz – így pl. a minősített tanúsítványokhoz – általában személyes regisztráció tartozik, és a regisztráció valamilyen arcképes igazolvány alapján történik.

  Az elektronikus aláírásról szóló törvény szerint a hitelesítés-szolgáltatónak közhiteles adatbázissal is egyeztetnie kell az igénylő adatait. [180] Így egyúttal azt is ellenőrzi, hogy valóban létezik-e az igénylő által bemutatott igazolvány (és a benne szereplő személy).

  A hitelesítés-szolgáltató ekkor határozza meg az alany tanúsítványba kerülő egyedi megnevezését (distinguished name) is. Kérdés, hogy a tanúsítványba kerülő adatokat milyen módon határozza meg, és milyen módon ellenőrzi a hitelesítés-szolgáltató. Magyarországon elterjedt megoldás, hogy az alany neve a személyazonosításra használt okmányában (személyi igazolványában, útlevelében vagy jogosítványában) szereplő írásmóddal szerepel a tanúsítványban. A legtöbb szolgáltató ékezethelyesen tünteti fel a nevet, de előfordulhat, hogy a név ékezetmentesen jelenik meg.

- Ha a tanúsítványhoz tartozó magánkulcsot nem az alany, hanem a hitelesítés-szolgáltató generálta, akkor a magánkulcsot át kell adnia az alanynak. Ez leggyakrabban úgy történik, hogy a hitelesítés-szolgáltató intelligens kártyán generálja a magánkulcsot, és a kártyával együtt adja át a végfelhasználónak. A kulcsot a kártyán generálták, és jellemzően nem is nyerhető ki a kártyából, így a kártya átadásával a hitelesítés-szolgáltató azt is biztosítja, hogy nem őriz meg másolatot a kulcsból. (Ez alól kivétel

a titkosító tanúsítványok esete, amikor kulcsletét szolgáltatás is használható (3.3.3. fejezet).)

A magánkulcs átadásának biztonsága a regisztráció biztonságához hasonlóan markánsan befolyásolja a tanúsítvány nyújtotta biztonságot. Hiába alkalmazna egy szolgáltató erős regisztrációt, ha a magánkulcsot nyílt e-mailben küldené el az ügyfélnek. A szolgáltató által alkalmazott megoldás akkor értelmes, ha e két műveletet azonos biztonsági szinten hatja végre. Gyakori, hogy a magánkulcsot tartalmazó intelligens kártyát a regisztrációval egy időben adja át a szolgáltató az ügyfélnek.

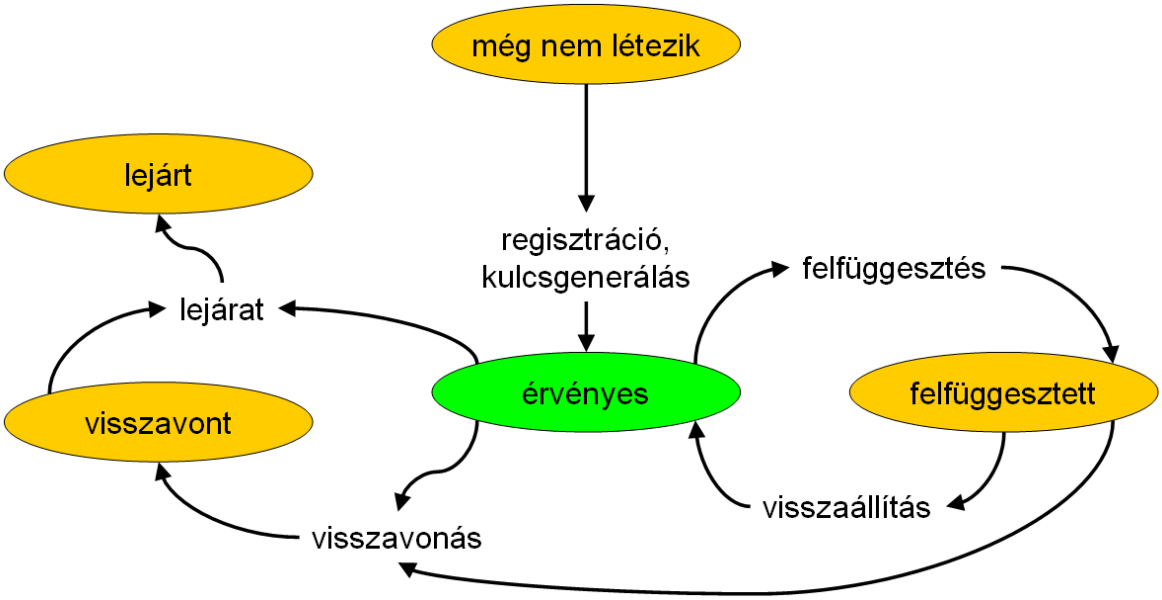
Ha a magánkulcsot az ügyfél generálja, akkor már eleve nála van, így nem kell eljuttatni hozzá. Ekkor az ügyfél nyilvános kulcsát kell hitelesen eljuttatni a hitelesítés-szolgáltatónak, és ez egy hasonlóan kritikus művelet, a regisztrációval azonos biztonsági szinten célszerű végezni. A különbséget az jelenti, hogy ekkor nem egy magánkulcsot kell titkosan, hanem egy nyilvános kulcsot kell hitelesen célba juttatni. (Lásd: 4.5.2. fejezet.)

- A tanúsítvány cseréjekor szintén azonosítás történik. Ha a tanúsítványcserére kulcskompromittálódás miatt van szükség, akkor új magánkulcsot kell eljuttatni az ügyfélnek, vagy az ügyfél új nyilvános kulcsát kell hitelesen eljuttatni a hitelesítés-szolgáltatónak, így az előző pontban szereplő megfontolások az érvényesek.

  Ha a tanúsítványcsere nem kulcskompromittálódás miatt történik, akkor egy adott magánkulcs már ott van az ügyfélnél. Például ha egy tanúsítvány le fog járni, és ezért meg kellene újítani, gyakori megoldás, hogy az ügyfél egy elektronikusan aláírt e-mailben kéri a tanúsítvány megújítását. A megújítás és a tanúsítványban szereplő adatok megváltoztatása bizonyos esetekben a regisztrációnál kevésbé kritikus művelet, illetve ekkor az ügyfél és a hitelesítés-szolgáltató elektronikusan is végezheti a tanúsítványcserét.

- A hitelesítés-szolgáltató felfüggesztés vagy visszavonás esetén is azonosítja az ügyfelet. A felfüggesztés egy ideiglenes művelet, felfüggesztéskor arra szokás törekedni, hogy minél gyorsabban elvégezhető legyen. Igaz, ekkor sem szerencsés, ha a hitelesítés-szolgáltató megbízható azonosítás nélkül függeszt fel egy tanúsítványt, mert ekkor egy támadó könnyen olyan helyzetbe hozhat egy ügyfelet, hogy az ne tudjon aláírni.

  4.17. Példa: *Alajos a határidő előtt utolsó pillanatban akar elektronikus aláírással beadni egy tendert. Manfréd meg akarja akadályozni, hogy Alajos érvényes ajánlatot tegyen, ezért megpróbálja rávenni a hitelesítés-szolgáltatót, hogy függessze fel Alajos tanúsítványát. Telefonon felhívja a szolgáltatót, Alajosnak adja ki magát, és bejelenti, hogy a magánkulcsa kompromittálódott.*



4.4. ábra. **A tanúsítvány állapotai**

Ha hisznek neki, Manfréd sikerrel jár, és Alajos egy ideig nem tud érvényes aláírást készíteni.

Míg a felfüggesztett tanúsítvány visszaállítható, a visszavonás végleges művelet. Ezért a felfüggesztéshez kapcsolódó azonosítási és hitelesítési módszerek általában egyszerűbbek, mint a visszavonáshoz kapcsolódóak. (4.4. ábra)

Általában igaz, hogy sokkal kisebb kár képződhet akkor, ha egy aláíró nem tud aláírni, mintha egy támadó alá tud írni az aláíró nevében. Ezért a felfüggesztés és visszavonás műveletek, amelyek az érvényes tanúsítványt érvénytelenné teszik, általában egyszerűbbek és kevésbé bürokratikusak. Azok a műveletek a különösen kritikusak, amelyeknek érvényes tanúsítvány a kimenete; ilyen például a regisztráció és a felfüggesztett tanúsítvány visszaállítása.

4.3. Tanúsítványok életciklusa

A hitelesítési rend (és a hitelesítés-szolgáltató szolgáltatási szabályzata) meghatározza, hogy az érintett tanúsítványok fenntartása során a hitelesítés-szolgáltató milyen műveleteket milyen szabályok szerint hajt végre. A tanúsítványok életciklusát a tanúsítványokról szóló fejezetben (3. fejezet) írtuk le, itt a hitelesítési rendben megjelenő, a hitelesítés-szolgáltató szempontjából izgalmas kérdéseket ismertetjük.

4.3.1. Tanúsítványigénylés

Tanúsítványigénylés alatt azt értjük, amikor az igénylő jelzi a hitelesítés-szolgáltatónak, hogy tanúsítványt szeretne, és előzetesen megadja az adatait a szolgáltatónak (3.3.1. fejezet).

A hitelesítési rend meghatározza, hogy ki és milyen módon nyújthat be tanúsítványigénylést. Az Eat. előírja, hogy a hitelesítés-szolgáltató csak közvetlenül az aláírótól vagy az aláíró egyértelmű hozzájárulásával gyűjthet adatot.

> *Eat. „ 11. § (1) A hitelesítés-szolgáltatók csak az aláírótól közvetlenül, vagy annak egyértelmű előzetes hozzájárulásával gyűjthetnek személyes adatokat és csak olyan mértékben, ami a tanúsítvány kiadásához szükséges. ”*

A tanúsítványigénylés általában a hitelesítés-szolgáltató weboldalán nyújtható be egy erre a célra szolgáló tanúsítványigénylő lapon.

Ha egy nagyobb szervezet igényel tanúsítványt a munkatársainak, körülményes lehet, ha a szervezet munkatársainak egyenként ki kell tölteni a szolgáltató igénylőlapját. Ez esetben a szolgáltatók lehetőséget szoktak biztosítani az adatok csoportosan történő eljuttatására. Ez is csak a tanúsítvány majdani alanyának egyértelmű hozzájárulásával végezhető; nem szabad tanúsítványt igényelni valaki számára az illető tudta nélkül.

4.3.2. Regisztráció

Regisztrációnak azt nevezzük, amikor a hitelesítés-szolgáltató a tanúsítvány kibocsátása előtt megállapítja az igénylő (személy, illetve szervezet) kilétét, személyazonosságát (3.3.4. fejezet).

A hitelesítési rend meghatározza, hogy a regisztráció milyen biztonsági szinten történjen, a regisztráció részletes lépéseit általában a hitelesítés-szolgáltató szolgáltatási szabályzatai határozzák meg.

4.3.3. Kulcspár generálása és kezelése

A hitelesítési rend meghatározza, hogy a tanúsítványhoz tartozó magánkulcsot ki és hogyan generálja, és a nyilvános kulcs hogyan jut el a hitelesítés-szolgáltatóhoz (4.5.2. fejezet). Szintén meghatározza, hogy a magánkulcsot letétbe szabad-e helyezni kulcsletét szolgáltatás keretében.

A hitelesítési rend előírhatja, hogy a tanúsítványhoz tartozó magánkulcsnak intelligens kártyán vagy akár biztonságos aláírás-létrehozó eszközön kell lennie.

4.3.4. A tanúsítvány és a hozzá tartozó magánkulcs használata

A hitelesítési rend meghatározza, hogy a tanúsítvány alanya milyen módon használhatja a tanúsítványt és a hozzá tartozó magánkulcsot. A magánkulcsot általában a tanúsítványban

feltüntetett kulcshasználatnak megfelelően szabad használni. Ez azt jelenti, hogy aláíró tanúsítvány magánkulcsát csak aláírásra, titkosító tanúsítvány magánkulcsát csak dekódolásra, és autentikációs tanúsítvány magánkulcsát csak autentikációra szabad használni. (Lásd: 3.2.1. fejezet.)

Speciális tanúsítványok esetén további megkötések is előfordulhatnak. Például webszerver tanúsítványok és programok aláírására használható (ún. code signing) tanúsítványok esetén a tanúsítványt nem szabad olyan „rossz célra” használni, amely veszélyezteti az érintett feleket. A hitelesítés-szolgáltatók gyakran megtiltják, hogy a webszerver tanúsítvánnyal adathalász oldalakat hitelesítsenek, illetve code signing tanúsítvány szerint vírust írjanak alá. Míg az aláíró, titkosító és autentikációs tanúsítványok csak annyit igazolnak, hogy egy adott magánkulcs egy adott entitás birtokában van, a webszerver és a code signing tanúsítványok azt is igazolják, hogy a magánkulcsot birtokló entitás megbízható. A webszerver és code signing tanúsítványokat elfogadó alkalmazások (böngészők, operációs rendszerek stb.) általában csak olyan hitelesítés-szolgáltatók tanúsítványait fogadják el, akik előírják, hogy e tanúsítványokkal nem szabad veszélyeztetni az érintett feleket, és akik vállalják, hogy visszavonják e tanúsítványokat, ha a tanúsítvány alanya megszegi ezen előírást.

4.18. Példa: *Alajos vírust ír, amelyet a saját code signing tanúsítványához tartozó magánkulccsal ír alá. Amikor Bendegúz gépe a vírussal találkozik, a code signing tanúsítvány szerint ellenőrizhető aláírás miatt felismeri, hogy a szoftver megbízható gyártótól származik, és lefuttatja. Alajos mások becsapására használja a tanúsítványt, így valószínűleg megsérti a hitelesítés-szolgáltatóval kötött megállapodását és a hitelesítési rendet.*

4.19. Példa: *Alajos kábítószerrel kereskedik, más kábítószer-kereskedőkkel kötött megállapodásait saját minősített aláírásra szolgáló tanúsítványának magánkulcsával írja alá, és az üzenetei titkosításához saját, illetve partnerei titkosító tanúsítványát használja. Illegális célra, de rendeltetésszerűen használja a tanúsítványait, így nem sérti sem a hitelesítés-szolgáltatóval kötött megállapodását, sem a hitelesítési rendet. Más jogszabályokat viszont megsért, és emiatt börtönbüntetésre számíthat*[7]*.*

4.3.5. Felfüggesztés és visszavonás

A hitelesítési rend meghatározza, hogy milyen módon változhat meg a tanúsítvány visszavonási állapota (3.3.9. fejezet), és hogyan kerül közzétételre a megváltozott visszavonási állapot. Általában a következő felek kezdeményezhetik a tanúsítvány visszavonási állapotának megváltozását:

[7] Ha megegyezéskor szempont, hogy a megegyezés könnyen letagadható legyen, nem célszerű írást készíteni róla, és végképp nem célszerű elektronikus aláírást használni.

- a tanúsítvány alanya (pl. ha kompromittálódik a magánkulcsa, vagy ha többet nem szeretné használni a tanúsítványát);
- az előfizető, aki fizet a tanúsítványért (pl. ha már nem szeretné tovább fizetni);
- a tanúsítványban szereplő szervezet (pl. ha megszűnik az alany munkaviszonya az adott szervezettel);
- a hitelesítés-szolgáltató (pl. ha az előfizető nem fizet a tanúsítványért, vagy ha alappal feltételezhető, hogy a magánkulcs kompromittálódott, vagy a tanúsítványt a hitelesítési rend értelmében vissza kell vonni);

Akárki kezdeményezi a visszavonási állapot megváltozását, a hitelesítés-szolgáltatónak azonosítania és hitelesítenie kell a kérelmezőt (4.2. fejezet). A visszavonás művelet végleges, így egy téves visszavonás új tanúsítvány kibocsátásával, és az ehhez szükséges bürokratikus lépésekkel jár. Így visszavonáshoz „erős” azonosítást, „nagy” biztonságot jelentő lépéseket – egyes értelmezések szerint a tanúsítvány kibocsátásával egyenszilárdságú biztonságot – szokás alkalmazni. A visszavonás gyakran írásbeli bizonyíték alapján történik, amelyet vagy személyesen írnak alá a szolgáltató regisztrációs munkatársa előtt, vagy postán juttatnak el a szolgáltatónak, vagy elektronikusan aláírt üzenetben küldik el.

A téves felfüggesztés kevésbé költséges, így felfüggesztés esetén arra szokás törekedni, hogy gyors legyen, és a hitelesítés-szolgáltató minél hamarabb tehesse közzé a megváltozott visszavonási állapotot. A felfüggesztés gyakran telefonon történik – a hatóság állásfoglalása szerint a minősített tanúsítványokhoz kötelező 24 órás telefonos felfüggesztés szolgáltatást működtetni. [123] Telefonon általában jelszót vagy az alany személyes adatait kéri el a szolgáltató, így győződik meg a kérelem hitelességéről. Felfüggesztésnél általában az a cél, hogy a hitelesítés-szolgáltató a jogosult felek kérését minél hamarabb teljesíthesse.

A felfüggesztett tanúsítvány visszaállítása kritikus művelet, nem szabad, hogy a kártyát ellopó tolvaj vissza tudja állítani a felfüggesztett tanúsítványt. A visszaállítást általában a felfüggesztésnél körülményesebb, de az új tanúsítvány kibocsátásánál egyszerűbb eljárással szokás lebonyolítani.

A következő lépések szerint zajlik le egy tanúsítvány felfüggesztése.

1. A felfüggesztés vagy visszavonás bejelentése.
2. A hitelesítés-szolgáltató megállapítja a kérelem jogosságát.
3. A hitelesítés-szolgáltató átvezeti nyilvántartásában a kérelmet.
4. A hitelesítés-szolgáltató közzéteszi a megváltozott visszavonási állapotot.
5. Az érintett felek értesülnek az új visszavonási állapotról.

Onnantól kezdve, hogy a felfüggesztési kérelmet bejelentették, és a szolgáltató azt elfogadta, a hitelesítés-szolgáltatót terheli a felelősség a tanúsítvánnyal okozott károkért, egészen addig, amíg közzé nem tette a megváltozott visszavonási állapotot (vagyis azt, hogy a tanúsítványt felfüggesztették vagy visszavonták).

Előfordulhat, hogy a tanúsítványt ellenőrző érintett félhez ekkor még nem jut el megváltozott visszavonási állapot; például azért, mert ő még egy korábbi, nem lejárt visszavonási információ alapján dolgozik. (Lásd: 6.5.4.4. fejezet.)

4.3.6. A tanúsítvány ellenőrzése – ajánlások érintett felek részére

A hitelesítés-szolgáltató a nyilvánosság számára bocsátja ki a tanúsítványt. Bárki találkozhat a tanúsítvánnyal, így bármely érintett fél ellenőrizheti. Az érintett félnek gyakran semmilyen kapcsolata nincsen sem hitelesítés-szolgáltatóval, sem a hitelesítési rend kibocsátójával, így a hitelesítési rend nem mondhatja meg, hogy az érintett félnek hogyan *kell* ellenőriznie a tanúsítványt. A hitelesítési rend általában csak ajánlásokat fogalmaz meg az érintett fél részére, és leírja, hogy az érintett félnek hogyan *célszerű* eljárnia a tanúsítvány ellenőrzésekor.

A hitelesítési rend (illetve a hitelesítés-szolgáltató szolgáltatási szabályzata) általában csak annyit mond, hogy az érintett félnek ajánlott kellő gondossággal ellenőrizni a tanúsítványt, és célszerű minden szóba jöhető körülményt figyelembe vennie.

Ezen túlmenően a hitelesítési rend különféle nemzetközi specifikációkat szokott meghivatkozni, elsősorban az X.509 specifikációt, RFC 5280-at, és aláíró tanúsítványok esetén a CWA 14171-et. [152], [31] E specifikációk a következő lépéseket határozzák meg:

1. Aláírás ellenőrzésekor az aláírás, az aláírt dokumentum és az aláírói tanúsítvány összetartozásának vizsgálata. (Lásd: 6.5.3. fejezet.)

2. Tanúsítványlánc keresése egy megbízható gyökértanúsítványig. (Lásd: 5. fejezet.) Aláírás ellenőrzése esetén e műveletet az aláírás időpontjára nézve célszerű elvégezni.

3. A tanúsítványláncban szereplő tanúsítványok visszavonási állapotának vizsgálata. Aláírás ellenőrzése esetén e műveletet az aláírás időpontjára nézve célszerű elvégezni.

A tanúsítványok ellenőrzéséről a tanúsítványokról szóló fejezetben (3.3.7. fejezet) szólunk, az elektronikus aláírás és az aláíró tanúsítvány ellenőrzéséről az elektronikus aláírásról szóló fejezetben írunk részletesen (6.5. fejezet).

4.3.7. Tanúsítványcsere

A hitelesítési rend meghatározza, hogy hogyan és milyen körülmények között történhet a tanúsítvány cseréje (3.3.8. fejezet). Lényeges, hogy ki kezdeményezheti a cserét, a hitelesítés-

szolgáltató milyen módon azonosítja és hitelesíti az illetőt, és milyen ellenőrzési eljárást végez az új tanúsítvány kibocsátása előtt.

Gyakran felmerülő kérdések:

- Az újonnan kibocsátott tanúsítványban lehet-e ugyanaz a nyilvános kulcs, mint a lecserélt tanúsítványban? Ha igen, mely esetekben lehet ismét felhasználni a korábbi kulcspárt?
- Ha az eredeti tanúsítványt személyes regisztráció során bocsátották ki, és az új tanúsítvány magánkulcsa már az aláíró birtokában van (mert ugyanazt a kulcsot használjuk, vagy másik kulcsot, de ugyanazon a kártyán), meg kell-e ismételni a személyes találkozást?
- Meg kell-e ismételni a közhiteles adatbázissal történő adategyeztetést, és ha igen, mely esetekben (minden esetben vagy csak adat változása esetén)?

4.3.8. A tanúsítvánnyal kapcsolatos adatok megőrzése

A hitelesítési rend meghatározza, hogy a hitelesítés-szolgáltató mennyi ideig őrzi meg a tanúsítvánnyal kapcsolatos adatokat. Ezen adatok megőrzése elsősorban az elektronikus aláírások érvényességének megállapításához szükséges. A szolgáltatónak meg kell őriznie a *regisztrációs adatokat*, azaz, hogy kinek, mely személynek bocsátotta ki a tanúsítványt. (Így ha egy aláíró tanúsítványban annyi szerepel, hogy az aláíró neve „Kovács János", ki lehessen deríteni, hogy melyik „Kovács János" készítette az aláírást.) A szolgáltatónak szintén meg kell őriznie az arról szóló nyilvántartását, hogy melyik tanúsítvány mikor vált érvénytelenné, mert a lejárt tanúsítványok PKI alapon történő ellenőrzése problémás. (Lásd: 4.1.5. fejezet.) A hitelesítés-szolgáltatóknak egyéb információkat, például naplófájljaikat is meg kell őriznie, hogy utólag is fényt lehessen deríteni az esetleges visszaélésekre.

Az elektronikus aláírásról szóló törvény szerint a hitelesítés-szolgáltatónak a tanúsítvány lejártát követően 10 évig meg kell őriznie a tanúsítvánnyal kapcsolatos információkat. (Eat. 9. § (7)) A nem aláírásra szolgáló tanúsítványokra nincsen ilyen szabály.

4.3.9. Szolgáltatások leállítása

Ha a hitelesítés-szolgáltató megszűnik, gondoskodnia kell arról, hogy a szükséges információkat más őrizze tovább helyette. Így kerülhető el, hogy egy hitelesítés-szolgáltató megszűnése miatt aláírásokról ne lehessen megállapítani, hogy ki készítette őket. Hitelesítés-szolgáltató csak szabályozott körülmények között szűnhet meg. Az aláíró tanúsítványokat kibocsátó szolgáltatók megszűnését az Eat. 16. § szabályozza. A minősített hitelesítés-szolgáltatóknak jelentős bankgaranciával is rendelkezniük kell, hogy ha a megszűnésükkel kapcsolatos lépéseket

nem végzik el, a Hatóság ezen összeg terhére gondoskodhat a szabályzott megszűnésről és a szükséges nyilvántartások átvételéről.

4.4. Fizikai, eljárásbeli és személyzeti óvintézkedések

A hitelesítés-szolgáltatónak biztonságos informatikai rendszert kell működtetnie, hogy az érintett felek elfogadják az általa kiállított tanúsítványokat. Ehhez fizikailag is biztonságos környezetet kell kialakítania. A továbbiakban elsősorban a minősített hitelesítés-szolgáltatókra vonatkozó követelményeket és legjobb gyakorlatokat ismertetjük. A követelmények többségét az elektronikus aláírásról szóló törvény és a hozzá kapcsolódó 3/2005. IHM r. határozza meg. Mértékadónak tekinthető még a már nem hatályos 2/2002. MeHVM irányelv, amely a CWA 14167 specifikációra épül. [28]

A hitelesítés-szolgáltatónak kockázatelemzést kell készítenie, eszközeit, helyiségeit biztonsági osztályokba kell sorolnia, és biztosítania kell, hogy a kritikus biztonsági zónáiba csak arra jogosult személyek léphessenek be, és kritikus erőforrásaihoz csak az arra jogosult személyek férhessenek hozzá. Védekeznie kell mind illetéktelen behatolás, mind tűz, árvíz ellen.

Általánosságban is kimondható, hogy egy szolgáltatás csak akkor működhet igazán megbízhatóan, ha a szolgáltatója több fizikai helyszínről nyújtja. A hitelesítés-szolgáltatónak tartalék helyszínnel, és ott háttérrendszerrel kell rendelkeznie. Ha az elsődleges helyszínen működő rendszere kiesik, e háttérrendszernek meghatározott időn belül át kell tudnia venni az elsődleges rendszer szerepét. Ennek biztosítására a hitelesítés-szolgáltatónak üzletmenet-folytonossági tervvel és katasztrófatervvel kell rendelkeznie, és e tervek végrehajtását rendszeresen tesztelnie kell.

A minősített hitelesítés-szolgáltatókra az a követelmény vonatkozik, hogy kritikus szolgáltatásaikat – a tanúsítványtár elérhetőségét, a visszavonás-kezelést és a visszavonási állapot közzététele szolgáltatásokat – éves szinten 99,9%-os rendelkezésre állással kell biztosítaniuk, és a leghosszabb kiesés nem haladhatja meg a 3 órát. A nem minősített hitelesítés-szolgáltatókra nincsen általános rendelkezésre állási követelmény, de a közigazgatási követelményrendszer a közigazgatási területen felhasználható tanúsítványokat kibocsátó, nem minősített hitelesítés-szolgáltatókra éves szinten 99%-os rendelkezésre állást ír elő, és a leghosszabb kiesés nem haladhatja meg a 24 órát.

A 3/2005. IHM r. a következő *bizalmi munkaköröket* határozza meg a hitelesítés-szolgáltatókra: az informatikai rendszerért általánosan felelős vezető, biztonsági tisztviselő (aki a rendszer biztonságáért felel), rendszeradminisztrátor, rendszerüzemeltető, regisztrációs felelős (aki a tanúsítványok kibocsátásának, felfüggesztésének, visszavonásának jóváhagyásáért felel), és független rendszervizsgáló (aki napi rendszerességgel ellenőrzi a hitelesítés-szolgáltató rendszerének biztonságát). A rendelet kimondja, hogy a bizalmi szerepkört betöltő dolgozóknak a hitelesítés-szolgáltató alkalmazásában kell állniuk, erkölcsi

bizonyítvánnyal és megfelelő szakértelemmel kell rendelkezniük. Kimondja továbbá, hogy bizonyos szerepkörök kizárják egymást; például a regisztrációs felelős nem lehet egyúttal független rendszervizsgáló is, hiszen a rendszervizsgáló feladata lenne többek között a regisztrációs felelős munkájának ellenőrzése is. (A rendelet ezáltal egy minimális személyzeti létszámot is meghatároz a hitelesítés-szolgáltató számára.) A hitelesítés-szolgáltatónak gondoskodnia kell munkatársainak biztonságos módon történő kiválasztásáról és folyamatos képzéséről.

E követelmények előírják, hogy a hitelesítés-szolgáltatóknak minden lényeges (a tanúsítványok érvényességével kapcsolatos) eseményt naplózniuk kell, és a rendszerükben képződött adatokat archiválniuk kell. Alapelv, hogy minden, a tanúsítványok érvényességével kapcsolatos adatot a tanúsítvány lejártától számított 10 évig (vagy a tanúsítvánnyal kapcsolatos jogviták lezártáig) meg kell őrizni.

4.5. Műszaki biztonsági óvintézkedések

> *„The chances of anything coming from Mars are a million to one, he said.*
> *The chances of anything coming from Mars are a million to one – but still they come!"*
> *(Millió az egyhez az esélye, hogy bármi is jön hozzánk a Marsról – mondta [a csillagász].*
> *Millió az egyhez az esélye, de attól még... jönnek!)*
>
> *– H. G. Wells, Világok harca, Jeff Wayne musical-feldolgozásából*

A hitelesítés-szolgáltatók egyik központi kérdése a kulcsmenedzsment, azaz hogy a szolgáltató milyen kulcsokat kezel, és ezeket hogyan generálja, továbbítja, tárolja, és hogyan semmisíti meg. A hitelesítési rend e kérdéskört részletesen körüljárja.

4.5.1. Szolgáltatói kulcsok kezelése

A hitelesítés-szolgáltató saját magánkulcsait általában[8]kriptográfiai hardver modulok (HSM, hardware security module) védik (6.3.3.4. fejezet). A hitelesítés-szolgáltató rendszerében a HSM hasonló szerepet tölt be, mint az aláírónál az intelligens kártya: a szolgáltatói magánkulcsokat HSM generálja, és e kulcsok – nyílt formában – soha nem hagyják el a HSM-et. Fontos különbség, hogy az aláírói magánkulcsokról nincs értelme mentést készíteni (3.3.3. fejezet), míg a hitelesítés-szolgáltató magánkulcsát feltétlenül menteni kell. Ha a szolgáltatói magánkulcs megsemmisül, az hatalmas kárt okozhat mind a szolgáltatónak, mind az ügyfeleinek, mind az érintett feleknek. Ekkor a szolgáltató nem tud új CRL-eket (vagy OCSP válaszadói tanúsítványokat) kibocsátani, így nem lehet ellenőrizni a tanúsítványok

[8]Technikailag nincsen akadálya, hogy egy hitelesítés-szolgáltató szoftveres magánkulccsal működjön, de lényegében minden mértékadó szabályozás megköveteli, hogy a szolgáltatói magánkulcsot kriptográfiai hardver eszköz védje.

visszavonási állapotát. Ezért a HSM-ből általában – titkosított formában – menteni lehet a magánkulcsot, és e titkosított magánkulcsot be lehet tölteni egy másik – jellemzően azonos típusú – HSM-be. A HSM jó minőségű, hardveres véletlenszám-generátorral rendelkezik, így a vele generált kulcsok várhatóan jó minőségűek lesznek.

A szolgáltatói magánkulcsokkal kapcsolatos főbb műveleteket (a kulcs generálását, a kulcs betöltését a HSM-be, a kulcs mentését vagy megsemmisítését) általában több, bizalmi munkakört betöltő tisztviselő együttesen végezheti el. Gyakori megoldás, hogy e tisztviselők egy-egy kártyával rendelkeznek, és a kritikus műveletek előtt be kell dugniuk kártyájukat a HSM-be, és be kell írni PIN kódjukat, így igazolják, hogy valóban hozzájárulnak az adott művelet elvégzéséhez.

A magyar hitelesítés-szolgáltatók csak minősített, bevizsgált HSM-et használhatnak. Az elektronikus aláírásról szóló törvény kimondja, hogy tanúsítvány előállításához kizárólag olyan eszköz (ún. elektronikus aláírás termék) használható, amely szerepel a Nemzeti Média- és Hírközlési Hatóság nyilvántartásában, vagy más EU tagállam hasonló nyilvántartásában. Magyarországon meghatározott tanúsító szervezetek tanúsíthatnak HSM-eket, a Hatóság az ő tanúsításuk alapján veszi őket nyilvántartásba.

A hitelesítés-szolgáltatók a nyilvános kulcsaikat általában nyilvánosságra hozzák (4.1.4. fejezet).

4.5.2. Ügyfelek kulcsainak kezelése

A nyilvános kulcsú infrastruktúra egyik fő alapelve, hogy a végfelhasználó magánkulcsa csak magánál a végfelhasználónál lehet. Ez alól a kulcsletét szolgáltatás jelenti az egyedüli kivételt (3.3.3. fejezet). Így ha a hitelesítés-szolgáltató generálja az ügyfél magánkulcsát, a kulcsból nem őrizhet meg másolatot. Ha a kulcsot intelligens kártyán adja át, a kártya PIN kódjából sem őrizhet meg másolatot.

Az elektronikus aláírásról szóló törvény szerint minősített aláírás csak biztonságos aláírás-létrehozó eszközzel (és minősített tanúsítvány szerint) készíthető. A biztonságos aláírás-létrehozó eszköznek, avagy BALE-nak (angolul: SSCD – secure signature creation device) rendelkeznie kell vagy a Nemzeti Média- és Hírközlési Hatóság nyilvántartásába vett tanúsító szervezet által kibocsátott igazolással, miszerint az eszköz BALE, vagy más EU tagállamban nyilvántartott tanúsító szervezet által kibocsátott hasonló igazolással. Magyarországon a BALE-kat ugyanazon tanúsító szervezetek tanúsítják, akik a HSM-ek tanúsítását is végzik. (Lásd: 6.3.3.3. fejezet.)

A következő megoldások a leginkább jellemzőek:

- A hitelesítés-szolgáltató az ügyfél kulcspárját intelligens kártyán generálja[9], és a

[9] Az is előfordulhat, hogy a kulcsot a hitelesítés-szolgáltató HSM segítségével állítja elő, és olyan módon tölti át a kártyába, hogy a HSM-ben garantáltan nem marad meg másolat.

magánkulcsot az intelligens kártyával együtt átadja a végfelhasználónak. Minősített tanúsítványok esetén ez a megoldás terjedt el, egy végfelhasználó jellemzően nem tud BALE-t megfelelően[10] beállítani, kulcsot generálni.

Az elektronikus aláírásról szóló törvény külön szolgáltatásként kezeli az ún. *eszköz-szolgáltatást*, teljes nevén „aláírás létrehozó eszközön aláírás-létrehozó adat elhelyezése" szolgáltatást. Így elképzelhető lenne, hogy valaki egy minősített eszköz-szolgáltatótól veszi meg a magánkulcsát tartalmazó BALE-t (intelligens kártyát), elviszi egy másik szervezethez, egy minősített hitelesítés-szolgáltatóhoz, és az bocsát rá ki minősített tanúsítványt. A gyakorlatban e modell nem terjedt el. Az eszközök sokféleképpen működnek, és nem tud minden szolgáltató minden eszközt támogatni. Ráadásul egy szolgáltatónak azt sem lenne könnyű megállapítania, hogy egy adott eszköz valóban BALE-e (és valóban BALE üzemmódban működik-e), így nem lehetne biztos benne, hogy valóban kiadhat egy minősített tanúsítványt a benne lévő magánkulcshoz tartozó nyilvános kulcshoz.

- A kulcspárt a végfelhasználó generálja, és csak a nyilvános kulcsot juttatja el a hitelesítés-szolgáltatónak. Ekkor a magánkulcs nem jut el a hitelesítés-szolgáltatóhoz, de ekkor a végfelhasználó nyilvános kulcsának kell hitelesen eljutnia a hitelesítés-szolgáltatóhoz.

  A végfelhasználó ekkor ún. PKCS#10 formátumú tanúsítványkérelmet szokott küldeni a hitelesítés-szolgáltatónak. A PKCS#10 a végfelhasználó tanúsítványba kerülő nyilvános kulcsát és a tanúsítványba kerülő adatokat – a végfelhasználó megnevezését (DN) – tartalmazza, és a végfelhasználó a saját magánkulcsával írja alá. A PKCS#10 kérelmen lévő aláírás a benne foglalt nyilvános kulccsal ellenőrizhető, így a PKCS#10 kérelem azt igazolja, hogy valaki, valamikor, az adott DN-hez valamely hitelesítés-szolgáltatótól valóban igényelt tanúsítványt. A PKCS#10 kérelmet általában a hitelesítés-szolgáltató honlapján keresztül szokás eljuttatni a tanúsítványra vonatkozó megrendeléssel együtt, és a szolgáltatónak valamilyen módon ellenőriznie kell, hogy az adott PKCS#10 kérelmet valóban az a személy küldte, akinek az adatai a kérelemben szerepelnek. Személyes regisztráció során kibocsátott tanúsítványok esetén e két lépés jellemzően nem egyszerre történik, mert az egyik elektronikusan, a másik viszont személyesen zajlik le, így e kettő összekötése problémás lehet.

  E módszert „szoftveres" tanúsítványok esetén szokás használni, amikor a tanúsítványhoz tartozó magánkulcsot nem védi intelligens kártya.

- Elvileg előfordulhat, hogy a hitelesítés-szolgáltató „szoftveresen", intelligens kártya

[10] Egy BALE tanúsítással rendelkező eszköz kizárólag akkor tekinthető BALE-nak, ha valóban olyan módon állították be, ahogy a tanúsítás rá vonatkozik. Ha egy BALE-t nem a tanúsítási dokumentációnak megfelelően állítanak be, semmilyen biztosíték nincs rá, hogy az valóban védi a magánkulcsot.

nélkül generálja a kulcspárt, de ez problémás. Ekkor nincs biztosítva, hogy megfelelő véletlen forrást használna a kulcsgeneráláshoz, és nincs rá műszaki garancia, hogy nem őriz meg másolatot a magánkulcsból; ezt csak a szolgáltató szabályzott eljárásai biztosíthatják.

4.5.3. Algoritmusok és paraméterek

A hitelesítési rend (vagy a hitelesítés-szolgáltató szolgáltatási szabályzata) rögzíteni szokta, hogy a hitelesítés-szolgáltató milyen kriptográfiai algoritmusokat és milyen kulcsméreteket használ a szolgáltatások nyújtásához. Magyarországon a szolgáltatók az RSA nyilvános kulcsú kriptográfiai algoritmust használják. A használt kulcsméret legalább 1024 bit, de folyamatban van az áttérés a 2048 bites RSA kulcsokra. A lenyomatképző algoritmus általában az SHA-1, de folyamatban van az áttérés az SHA-256 algoritmusra.

4.6. Tanúsítvány, CRL és OCSP profilok

Egy hitelesítési rend általában meghatározza a rendnek megfelelő tanúsítványok, illetve a rájuk vonatkozó visszavonási listák és OCSP válaszok felépítését. A szolgáltató általában azt írja le, hogy a vonatkozó nemzetközi specifikációk – elsősorban az RFC 5280 és az RFC 2560 – mely mezőit és mely opcionális megoldásait használják.

4.7. A megfelelőség vizsgálata

A hitelesítés-szolgáltatóknak biztonságosan és megbízhatóan kell működniük, hogy az érintett felek elfogadják az általuk kibocsátott tanúsítványokat. Ezért megbízható informatikai rendszert kell működtetniük, és megbízható belső folyamatok szerint kell dolgozniuk. Nem elég, hogy ők maguk biztonságosan működjenek, erről másokat is meg kell győzniük, csak így tölthetik be a megbízható harmadik fél szerepét az elektronikus világban. A hitelesítés-szolgáltatók saját belső ellenőrzésük mellett külső, független vizsgálatokon, ellenőrzéseken, auditokon is átesnek. E független ellenőrzések eredménye általában nyilvános, hogy az érintett felek meg tudjanak győződni egy szolgáltató biztonságáról.

Magyarországon az elektronikus aláírásról szóló törvény szerint a Nemzeti Média- és Hírközlési Hatóság folyamatosan vizsgálja és ellenőrzi a hitelesítés-szolgáltatók működését. [180] E vizsgálat a minősített hitelesítés-szolgáltatók esetén különösen szigorú. A minősített szolgáltatóknak a jogszabályok értelmében minőség-irányítási (pl. ISO 9001) és információbiztonság-irányítási (pl. ISO 27001) rendszert kell működtetniük, és évente egy, a Hatóság nyilvántartásában szereplő, független elektronikus aláírás szakértővel is meg kell vizsgáltatniuk rendszerüket. Ezen túlmenően a Nemzeti Média- és Hírközlési

Hatóság is évente helyszíni ellenőrzést végez a szolgáltatóknál. A minősített szolgáltatók meg kell, hogy feleljenek az elektronikus aláírásról szóló törvénynek, a hozzá kapcsolódó jogszabályoknak, illetve a szolgáltatási szabályzatukban vagy hitelesítési rendjükben vállalt egyéb követelményeknek (pl. szabványoknak). A hazai szolgáltatók általában az ETSI TS 101 456 és az ETSI TS 102 042, és a hozzájuk kapcsolódó egyéb szabványoknak és ajánlásoknak megfelelően működnek. [47], [53]

Lényeges, hogy az elektronikus aláírásról szóló törvény csak az aláírásra szolgáló tanúsítványok kibocsátását szabályozza, így a hatósági ellenőrzés elvileg nem terjed ki a titkosító és az autentikációs tanúsítványok kibocsátására. Ugyanakkor a hitelesítés-szolgáltatók általában a titkosító és autentikációs tanúsítványokat ugyanazon rendszerükkel bocsátják ki, mint amelyet a fokozott biztonságú aláírás létrehozására alkalmas tanúsítványok kibocsátására használnak[11].

A 4.1.4. fejezetben elmondtuk, hogy a szolgáltatói gyökértanúsítványok terjesztésének egyik legjobb módja, ha a gyökerek szoftverekkel együtt terjednek. Ehhez a szoftverek gyártói (pl. a Microsoft és a Mozilla) meg kell, hogy egyezzenek azon hitelesítés-szolgáltatókkal, akik tanúsítványait terjesztik. A szoftvergyártók általában szolgáltató-semlegesek szeretnének maradni, és minden szolgáltató gyökerét hajlandóak terjeszteni, feltéve, hogy a szolgáltató kellően biztonságos, és működésével nem veszélyezteti a szoftver felhasználóit.

> ***4.20. Példa:*** *Manfréd, a támadó feltörte a Kókler CA rendszerét, és bármilyen tanúsítványt alá tud írni a Kókler CA magánkulcsával. Ha az X böngészőprogram óvatlanul beveszi a Kókler CA gyökértanúsítványát az alapértelmezetten terjesztett gyökerek közé, Manfréd egy csaló weboldallal az X böngésző összes felhasználóját megtévesztheti.*

A szoftvergyártók meghatározott auditokat követelnek meg a hitelesítés-szolgáltatóktól. Legelterjedtebb az ún. WebTrust audit, de egyre többen elfogadják az ETSI TS 101 456 és az ETSI TS 102 042 alapú auditokat is. [189], [47], [53] Az auditokat általában független[12], kompetens félnek kell lebonyolítania. (Ez az egyes országokban mást és mást jelenthet.) [113], [115] Megjegyezzük, hogy az alkalmazásfejlesztők nem vállalnak felelősséget az általuk terjesztett gyökerű szolgáltatók tevékenységéért.

[11] A 3/2005. IHM r. megköveteli, hogy a hitelesítés-szolgáltató a minősített szolgáltatásokhoz használt elektronikus aláírás termékeket elkülönítetten kezelje. Így a minősített tanúsítványok kibocsátásához használható rendszer nem használható fokozott biztonságú aláírás létrehozására alkalmas tanúsítványok kibocsátására, sem titkosító és autentikációs tanúsítványok kibocsátására.

[12] A kormányzati hitelesítés-szolgáltatók esetén nem mindig követelmény, hogy független fél végezze az auditot. Egy kormányzati szerv nem mindig teheti meg, hogy független auditorokat enged saját érzékeny rendszereihez. Kormányzatok által működtetett hitelesítés-szolgáltatók esetén audit jelentés helyett néhol a szolgáltató által kiállított nyilatkozatot (audit equivalency statement) is elfogadnak.

4.8. Üzleti és jogi tudnivalók

A hitelesítés-szolgáltatók sok esetben vállalatok, akik a tanúsítványt pénzért adják. Általános szerződési feltételeket határoznak meg, ezeket közzéteszik, és ezen ÁSZF-ek az ügyfelekkel kötött szerződések mellékletét képezik.

Létezhetnek nem vállalatként működő hitelesítés-szolgáltatók is, amelyek más konstrukcióban működnek. Például előfordulhat, hogy egy államilag kibocsátott kártyára – például személyi igazolványra – kerül tanúsítvány.

4.8.1. Díjak, árak

A hitelesítés-szolgáltatók általában honlapjukon teszik közzé, hogy szolgáltatásaikat milyen árakon nyújtják. Ezen árak arra vonatkoznak, ha valaki egy vagy néhány tanúsítványt szeretne vásárolni tőlük. Nagy mennyiségű tanúsítvány esetén, illetve ha az ügyfél az adatok összegyűjtésének vagy a regisztráció megszervezésének, lebonyolításának egyes lépéseit magára vállalja, a honlapon közzétettnél akár jelentősen kedvezőbb árak is elérhetőek.

A tanúsítványokért általában éves vagy havi díjat kérnek a szolgáltatók. Előfordulhat, hogy e díjban szerepel a tanúsítványhoz kapcsolódó egyéb eszközök (intelligens kártyák, kártyaolvasó készülékek) ára, és a regisztrációhoz kapcsolódó költségek, de van, hogy ezekhez külön, eseti díjak kapcsolódnak.

Az is előfordulhat, hogy a szolgáltató minden tanúsítvány kibocsátásakor kér díjat az ügyféltől, de nem mindig szerencsés, ha minden tanúsítványcsere esetén ki kell fizetni az ügyfélnek a teljes tanúsítvány díját. Névváltoztatás, e-mail cím változás, munkahely nevének megváltozása, esetleg költözés miatt nagyon könnyen előállhat, hogy valakinek sokszor soron kívül le kell cserélnie a tanúsítványát.

Az aláíró, titkosító és autentikációs tanúsítványokat el kell választani egymástól, de az ügyfélnek sokszor mindegyikre szüksége van, így a hitelesítés-szolgáltatók gyakran csomagokat alakítanak ki tanúsítványaikból, és ezeket együtt, havi vagy éves díjért kínálják. E díj gyakran független attól, hogy az adott időszakban – a tanúsítványcserék és megújítások következtében a tanúsítványok érvényességében bekövetkező átlapolódások miatt – pontosan hány tanúsítvány kerül kibocsátásra.

4.8.2. Jogok és kötelezettségek

A hitelesítési rend meghatározza a szolgáltató és az ügyfél jogait és kötelezettségeit. Jellemzően a szolgáltatónak kötelessége a hitelesítési rend és a szolgáltatási szabályzat szerint eljárnia, az ügyfélnek pedig kötelessége bejelentenie, ha a magánkulcsa kompromittálódik, vagy ha megváltozik valamely adata, különösen akkor, ha az adat a tanúsítványban is szerepel.

4.8.3. A hitelesítés-szolgáltató felelőssége

4.8.3.1. Hogyan okozhat kárt egy hitelesítés-szolgáltató?

Ha egy hitelesítés-szolgáltató a tanúsítvánnyal kárt okoz, a kárt meg kell térítenie. Hogyan okozhat kárt egy hitelesítés-szolgáltató?

- Egyik lehetőség, ha egy tanúsítványban nem annak a megnevezését tünteti fel, aki a tanúsítványhoz tartozó magánkulcsot birtokolja. Ez például akkor fordulhat elő, ha egy bűnöző megtéveszti a hitelesítés-szolgáltatót, és lopott vagy hamisított igazolvánnyal más nevében igényel tanúsítványt.

- Másik lehetőség, hogy a hitelesítés-szolgáltató nem elég gyorsan vonja vissza, vagy függeszti fel a tanúsítványt, esetleg nem tartja a hitelesítési rendben vagy a szolgáltatási szabályzatban előírt felfüggesztési vagy visszavonási határidőket.

Ekkor a tanúsítványban feltüntetett félnek kára származhat belőle, ha a bűnöző aláírásokat készít a nevében, hozzáfér a neki szóló titkos levelekhez, vagy az autentikációs tanúsítvánnyal különféle szolgáltatásokat vesz igénybe a tanúsítvány alanyának nevében. Az is lehet, hogy az érintett fél szenved kárt, aki elfogadja a tanúsítványt: érvénytelen aláírást fogad el, kiszivárognak a titkai, és egy bűnöző fér hozzá a szolgáltatásaihoz.

4.8.3.2. A szolgáltatói felelősség korlátozása

Ha a hitelesítés-szolgáltató hibázik, abból bármekkora kár származhat. Elegendő egyetlen hibát elkövetnie a regisztráció vagy a visszavonás-kezelés során, egyetlen hibás aláírás is óriási károkat okozhat.

Ezért a szolgáltatók korlátozni szokták, hogy mekkora kártérítést hajlandóak fizetni egy tanúsítvánnyal kapcsolatban. Ha e korlátozás a szolgáltatási szerződésben szerepel, az – a hatályos jogszabályok értelmében – csak a szolgáltatási szerződést aláíró felekre, azaz az alanyra, az előfizetőre, a képviselt szervezetre vonatkozik, az érintett félre nem.

4.8.3.3. Tranzakciós limit

Az elektronikus aláírásról szóló törvény értelmében a hitelesítés-szolgáltató meghatározhatja, hogy egy tanúsítvány mire használható:

> *Eat. 9. § (2) „ A hitelesítés-szolgáltató a minősített tanúsítványban meghatározhatja a tanúsítvány felhasználásának tárgybeli, földrajzi vagy egyéb korlátait, illetve az egy alkalommal vállalható kötelezettség legmagasabb értékét. ”*

Ennek keretében meghatározhatja, hogy a tanúsítvánnyal legfeljebb mekkora kötelezettség vállalható. Ekkor a hitelesítés-szolgáltató kijelenti, hogy az adott tanúsítvány X forintig terjedő ügyletekben használható, mert ennek megfelelő eljárásokat alkalmaz vele kapcsolatban, és ezt meghaladó tranzakciókra már nem alkalmas. Ha valaki mégis ezt meghaladó tranzakciókra használja, akkor megszegi a szolgáltatási szerződést, és ez esetben a szolgáltató nem vállal felelősséget a tanúsítványért.

> *Eat. 15. § „ (2) A 9. § (2) bekezdése szerinti korlátokat meghaladó ügyletekben kibocsátott és aláírt elektronikusan aláírt elektronikus dokumentumból származó követelésekért, illetve az így okozott kárért a hitelesítés-szolgáltató nem felel. ”*

Az ilyen korlátozást fel kell tüntetni a minősített tanúsítványban:

> *Eat. 2. melléklet „ A minősített tanúsítványoknak tartalmazniuk kell az alábbiakat:*
> *[...]*
> *i) a tanúsítvány használhatósági körére vonatkozó esetleges korlátozásokat, [...] ”*

A tanúsítvánnyal egy alkalommal vállalható kötelezettség legmagasabb mértékét *tranzakciós limit*nek vagy ügyleti értéknek szokás nevezni. *A tranzakciós limit lehetővé teszi, hogy a hitelesítés-szolgáltató korlátozza a kockázatát akár a vele szerződésben nem álló érintett felekkel szemben is.* A tranzakciós limit mértékét a minősített tanúsítvány *QCStatements* (3.4. fejezet) mezejében szokás feltüntetni, de – tekintve, hogy e mezőt a legtöbb szoftver nem jeleníti meg – szövegesen is szerepelni szokott a tanúsítványban.

> **Megjegyzés:** Érdekes módon, az Eat. csak minősített tanúsítvány esetén ír tranzakciós limitről. Lehet, hogy ez azért van, mert nem minősített tanúsítványt úgysem használna valaki „fontos” esetben. Ugyanakkor az is okozhatja ezt, hogy a `QCStatements` mező csak minősített tanúsítványban szerepelhet, és a nem minősített tanúsítványokról megfeledkezett a jogalkotó. Nem minősített tanúsítványok esetén sokkal valószínűbb, hogy a szolgáltató korlátozni szeretné a kockázatát.
>
> Nem minősített tanúsítványok esetén a szolgáltatási szabályzatban szokás feltüntetni ezen értéket.

Míg egy banki átutalás vagy egy megrendelés esetén a tranzakciós limit értelmezése egyszerű feladat, mindez sokkal összetettebbé válhat egy átvételi elismervény, egy cég társasági szerződése vagy egy szerelmes levél esetén. Ha nem közvetlen anyagi kötelezettségvállalás esetén a szolgáltató nem tudná korlátozni kártérítési kötelezettségét, akkor a szolgáltatókra korlátlan kockázat nehezedne, és így a tranzakciós limit értelme veszne el. Ez alapján úgy lehet

megközelíteni a tranzakciós limit fogalmát, hogy mekkora kár keletkezhet egy szolgáltatói hibából. Ha egy tanúsítvány 1 millió forintos tranzakciós limitet tartalmaz, irreális, hogy valaki 500 millió forintos kártérítési igénnyel álljon elő. Ha ez történik, akkor az illető olyan célra használta a tanúsítványt, amely nem felelt meg az 1 millió forintos tranzakciós limitnek.

> **Megjegyzés:** Ha szolgáltatói hibából kár keletkezik, a bíróság várhatóan megosztja majd a kárt a károsult és a szolgáltató között, és ennek során várhatóan a tranzakciós limitet is figyelembe fogja venni. Ha lesznek ilyen ügyek, a bírói gyakorlat alapján tisztul majd le, hogy pontosan mit jelent a tranzakciós limit.

A tanúsítványt befogadó érintett félnek kell mérlegelnie, hogy milyen kockázatok származnak egy tanúsítvány elfogadásából, és ennek megfelelő tranzakciós limitet – vagy szolgáltatói felelősségvállalást – kell megkövetelnie.

4.8.3.4. Meddig terjed a hitelesítés-szolgáltató felelőssége?

- A hitelesítés-szolgáltató kártérítési kötelezettsége a vele szerződésben álló ügyfelek – alany, előfizető, képviselt szervezet – irányában a szolgáltatási szerződésben szereplő szolgáltatói felelősségvállalás mértékéig terjed. Ezt a szolgáltatók vagy magában a szerződésben vagy az általános szerződési feltételeikben szokták korlátozni.
- A tanúsítványt befogadó érintett fél jellemzően nem áll szerződésben a hitelesítés-szolgáltatóval, így rá a szerződésben szereplő szolgáltatói felelősségvállalás nem vonatkozhat. Ő csak a tanúsíványban szereplő tranzakciós limitet látja.
- A szolgáltatóval szerződésben nem álló érintett fél rosszul teszi, ha a tranzakciós limitet meghaladó ügyletekben is elfogadja a tanúsítványt, mert ekkor a szolgáltató általában kizárja a felelősségét.
- A szolgáltatói felelősségvállalás és a tranzakciós limit gyakran különböző összeg, általában a tranzakciós limit a nagyobb.
- Van olyan szolgáltató, amely nulla tranzakciós limittel bocsát ki tanúsítványt, ami bizarr módon azt jelenti, hogy – elvileg – semmilyen kártérítést nem hajlandó fizetni vele kapcsolatban.
- A 3/2005-ös IHM rendelet a hitelesítés-szolgáltatók felelősségbiztosítását a *szolgáltatói felelősségvállalás*hoz és nem a tranzakciós limithez kapcsolja, ez valószínűleg hiba a jogszabályban.

> *3/2005. IHM r., 11. § „ (3) A felelősségbiztosítási szerződésnek egy biztosítási esemény vonatkozásában káreseményenként nem minősített*

> *szolgáltató esetében a tanúsítványban, illetve a szolgáltatási szabályzatban vállalt felelősségvállalási érték legalább háromszorosáig, minősített szolgáltató esetében ötszöröséig kell fedezetet biztosítania az összes károsultnak okozott károkra. Több azonos okból bekövetkezett, időben összefüggő káresemény egy biztosítási eseménynek minősül. ”*

Így könnyen lehet, hogy létezik olyan szolgáltató, amely nagy tranzakciós limitű – azaz nagy ügyletekben használható – tanúsítványokat bocsát ki, de az ügyfél-szerződésekben szereplő nulla szolgáltatói felelősségvállalás miatt $5 * 0 = 0$, azaz nulla felelősségbiztosítással rendelkezik. Így az érintett fél nagy kockázatnak van kitéve, mert nem tud a nulla (vagy alacsony) szolgáltatói felelősségvállalásról és az ennek megfelelő felelősségbiztosításról.

- Egy aláírt dokumentummal okozott kár sokszor csak nehezen forintosítható; egy megrendelés esetén ez viszonylag egyszerűbb feladat, egy szerelmes levél esetén már jóval nehezebb. Nem mindig könnyű eldönteni, hogy átlépte-e valaki a tranzakciós limitet.

  Ugyanakkor ha valaki a tranzakciós limitet meghaladó kártérítési igénnyel lép fel egy hitelesítés-szolgáltatóval szemben, akkor a saját kalkulációja szerint átlépte, és a szolgáltatót nem terheli felelősség.

- Ma Magyarországon nemigen használják a tranzakciós limitet, de ha majd lesznek ezzel kapcsolatos ügyek, a bírói gyakorlat alapján tisztázódni fog, hogy pontosan mit jelent.

4.8.4. Bizalmasság

A hitelesítési rend meghatározza, hogy a hitelesítés-szolgáltató milyen információkat kezel bizalmasan, és miket hozhat nyilvánosságra.

Például a regisztrációs adatokat és az ügyfél személyes adatait nem hozzák nyilvánosságra a szolgáltatók, kivéve azon adatokat, amelyeket a tanúsítványban is feltüntetnek, és a tanúsítvánnyal együtt nyilvánosságra hoznak. A szolgáltató csak akkor hozhatja nyilvánosságra a tanúsítványt, ha az ügyfél ehhez előzetesen hozzájárult.

4.8.5. Irányadó jog

A hitelesítési rend meghatározza, hogy milyen jogszabályok vonatkoznak a tanúsítványra. Magyarországon az elektronikus aláírásról szóló 2001. évi XXXV. törvény vonatkozik az elektronikus aláírás létrehozására alkalmas tanúsítványokra.

4.8.6. Szabályzatok változtatása

A hitelesítési rend is egy dokumentum, amely változhat, ezért azt is meghatározza, hogy milyen eljárásrend szerint változhat meg. A hitelesítés-szolgáltatók a saját hitelesítési rendjeiket maguk változtatják, és a megváltozott rendet a honlapjukon teszik közzé. Az aláíró tanúsítványok hitelesítési rendjeit a Nemzeti Média- és Hírközlési Hatóság is nyilvántartásba veszi.

4.9. Összegzés

A hitelesítés-szolgáltató:

- Tanúsítványt bocsát ki, ezzel igazolja, hogy egy adott nyilvános kulcs egy adott szereplőhöz (a tanúsítvány alanyához) tartozik, azaz ő birtokolja a hozzá tartozó magánkulcsot.
- A tanúsítvány kibocsátása előtt azonosítja az igénylőt, közzéteszi a tanúsítványokat, valamint azt, hogy mely tanúsítványokat vont vissza.
- A tanúsítványt mindig valamilyen szabályrendszer – hitelesítési rend – szerint bocsátották ki. A tanúsítványban feltüntetésre kerül, hogy mely hitelesítési rend vonatkozik rá.
- Ha a hitelesítés-szolgáltató hibázik, meg kell térítenie az okozott kárt.
- Ezért megbízható, biztonságos rendszerrel és pénzügyi garanciákkal kell rendelkeznie. Az elektronikus aláírás törvény szerint nyilvánosan működő szolgáltatókat az Nemzeti Média- és Hírközlési Hatóság felügyeli.

5. fejezet

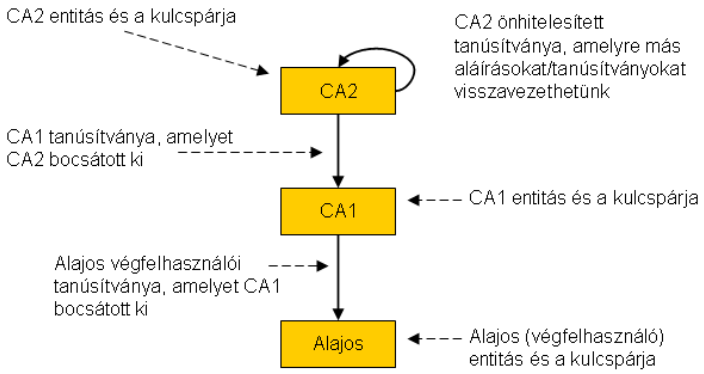
Tanúsítványlánc

> *„Nagy örömömre szolgál, hogy itt, vendégeim jelenlétében, noha ők, mármint vendégeim, egészen más elmélet igazolására szolgálnak, közölhetem önnel: elmélete szellemes és egyúttal helytálló. Egyébként egyik elmélet sem különb a másiknál, és nem is rosszabb. Van olyan felfogás is, miszerint mindenkinek az ő hite adatik. Hadd legyen így! Ön a nemlétbe távozik, én pedig örömmel iszom a létre, az életre, abból a serlegből, amellyé az ön koponyája változik!"*
>
> – *Mihail Bulgakov, A Mester és Margarita; Voland szavai Berlioz fejéhez*

Egy tanúsítvány ellenőrzése során az az egyik alapvető kérdés, hogy a tanúsítványt megbízható hitelesítés-szolgáltató bocsátotta-e ki, azaz visszavezethető-e a kérdéses tanúsítvány valamely megbízható gyökértanúsítványra. Ilyenkor egy úgy nevezett tanúsítványláncot keresünk az ellenőrizni kívánt tanúsítvány és egy megbízható gyökértanúsítvány között.

A tanúsítványlánc végén az ellenőrizni kívánt tanúsítvány található, és a lánc minden tanúsítványát a láncban őt megelőző hitelesítés-szolgáltatói tanúsítvány szerint bocsátották ki, a lánc legelső elemének kibocsátója pedig egy megbízható gyökértanúsítvány. A tanúsítványok ellenőrzésének módját is leíró X.509 és RFC 5280 specifikációkban szereplő definíciók szerint a megbízható gyökértanúsítvány nem része a tanúsítványláncnak, így ha az ellenőrizni kívánt tanúsítvány kibocsátója egy megbízható gyökértanúsítvány, akkor a lánc egy elemű.

A tanúsítványt egy érintett fél ellenőrzi, aki el szeretné dönteni, hogy valóban a tanúsítványban megnevezett fél birtokolja-e a tanúsítványba foglalt nyilvános kulcshoz tartozó magánkulcsot. Előfordulhat, hogy a tanúsítvány alanya is felépít egy tanúsítványláncot a tanúsítvány használata – aláírás vagy autentikáció – során, és lehet, hogy e tanúsítványláncot eljuttatja a tanúsítványt ellenőrző érintett félhez. Ha ez meg is történik, az érintett fél nem feltétlenül a kapott tanúsítványlánc szerint végzi az ellenőrzést. Az érintett fél úgy jár el, ahogy kedve tartja, bár célszerű követnie a szabványokban, mértékadó dokumentumokban és a tanúsítványra vonatkozó hitelesítési rendben szereplő ajánlásokat, mert a tanúsítvány ellenőrzésének kockázatát az érintett fél viseli.



5.1. ábra. **Egy egyszerű tanúsítványlánc. Az egyes téglalapok a bennük feltüntetett szereplő kulcspárját jelentik, a tanúsítványokat a téglalapokat összekötő nyilak jelentik.**

Az érintett fél felhasználhatja a kapott láncot, de kereshet másikat is. Előfordulhat, hogy más gyökereket fogad el megbízható gyökértanúsítványnak, más köztes tanúsítványokat talál meg, és akár teljesen új láncot is építhet.

Az RFC 5280 és az X.509 specifikációk írják le a tanúsítványok ellenőrzésének módját, és részletes algoritmust adnak egy tanúsítvány, illetve tanúsítványlánc ellenőrzésére. Ezen algoritmusnak a bemenete a megbízható gyökér és a tanúsítványlánc, és e specifikációk szólnak arról a kérdésről, hogy hogyan találhatjuk meg a szükséges láncot valamely, általunk elfogadott gyökérhez. [152], [191] A tanúsítványlánc felépítéséről az RFC 4158 ad felvilágosítást, de az is főként a felmerülő problémákat és a gyakran előforduló eseteket írja le, és ezek kezelésére szolgáló ökölszabályokat javasol. [147]

E fejezetben a tanúsítványlánc keresésének, felépítésének és ellenőrzésének kérdéseit járjuk körül. Ábráinkon a következő jelölésrendszert használjuk: Az egyes téglalapok a bennük feltüntetett szereplő kulcspárját jelentik, a tanúsítványokat a téglalapokat összekötő nyilak jelentik. A tanúsítványt az bocsátotta ki (azaz annak a magánkulcsával írták alá), akitől a nyíl kiindul. A tanúsítványt annak a számára bocsátották ki (azaz annak a nyilvános kulcsát tartalmazza), akihez a nyíl mutat. (Lásd: 5.1. ábra.)

5.1. Megbízható gyökér és megbízható gyökértanúsítvány

A megbízható gyökér (más néven *trust anchor*, magyarul bizalmi horgony) egy nyilvános kulcsból és a hozzá kapcsolódó megnevezésből (DN) áll, amelyeket tanúsítványok ellenőrzéséhez használunk. A megbízható gyökér magánkulcsát megbízható hitelesítés-szolgáltató birtokolja. A gyökér hitelesítés-szolgáltató megbízható és biztonságos működésében olyan mértékben megbízunk, hogy a PKI alapú ellenőrzéseinket az ő nyilvános kulcsára alapozzuk.

Különösen fontos, hogy a gyökér nyilvános kulcsát hitelesen szerezzük meg, és e hitelesség nem alapulhat ugyanezen gyökér szerint történő PKI alapú ellenőrzésre. *Általában out-of-band (azaz nem PKI alapú) hiteles módon jutunk hozzá a gyökérhez* (4.1.4. fejezet). Tekintve, hogy PKI alapú ellenőrzéseinket a megbízható gyökérre alapozzuk, a *gyökér visszavonási állapotát nem ellenőrizhetjük PKI alapon* (legalábbis semmiképpen sem ugyanezen gyökér alapján).

Habár a megbízható gyökér csak nyilvános kulcsból és megnevezésből áll, a megbízható gyökérrel általában önhitelesített tanúsítvány formájában találkozunk. Az önhitelesített tanúsítvány olyan tanúsítvány, amelynek kibocsátója és alanya megegyezik, azaz a kibocsátó és az alany megnevezése (DN) megegyezik, és a tanúsítványon lévő aláírás a tanúsítványban szereplő nyilvános kulccsal ellenőrizhető. Pusztán kényelmi okok miatt szokott a megbízható gyökér önhitelesített tanúsítvány formájában megjelenni, az önhitelesített tanúsítványon lévő aláírásnak nem sok szerepe van. Önhitelesített tanúsítványt bárki bármikor létrehozhat.

> ***5.1. Példa:*** *Manfréd, a támadó, generál magának egy kulcspárt, a nyilvános kulcsot belefoglalja egy tanúsítványba (amelyben a „Biztonságos CA") nevet tünteti fel, és a tanúsítványt a hozzá tartozó magánkulccsal írja alá. Ez önmagában nem jelent problémát, hiszen bárki bármikor létrehozhat önhitelesített tanúsítványt. Ha Manfrédnek sikerülne meggyőzi Alajost, hogy ezen önhitelesített tanúsítványt megbízható gyökérként használja, onnantól kezdve szinte bármit elhitethetne Alajossal: bármilyen webszervernek kiadhatja magát Alajos felé, és bárkinek az aláírását hamisíthatja úgy, hogy Alajos ne vehesse észre. Különösen fontos, hogy Alajos csak valóban megbízható tanúsítványokat fogadjon el megbízható gyökérnek.*

Azokat az önhitelesített tanúsítványokat nevezzük megbízható gyökértanúsítványnak, amelyeket megbízható gyökérként más tanúsítványok ellenőrzésére használunk. Technikailag bármely önhitelesített tanúsítvány használható gyökértanúsítványként, bármely (szolgáltatói) tanúsítványra próbálhatunk más tanúsítványokat visszavezetni.

Egy gyökértanúsítvány használhatóságát, „értékét" számos tényező befolyásolhatja, ezek határozzák meg, hogy az egyes felhasználói közösségek mely önhitelesített tanúsítványt vagy tanúsítványokat fogadnak el megbízható gyökértanúsítványnak. Ilyen tényezők például a következők:

- A gyökértanúsítványhoz tartozó magánkulcs biztonsága. Például mennyire biztonságos környezetben tárolják, védi-e kriptográfiai hardver eszköz, és milyen hardver eszköz védi, és milyen tanúsításokkal, auditokkal tudja igazolni a hitelesítés-szolgáltató, hogy valóban biztonságosan működik.

- Az adott gyökértanúsítványra visszavezethető végfelhasználói tanúsítvánnyal rendelkező közösség mérete.

- Az adott gyökértanúsítványra visszavezethető tanúsítványok biztonsága, beleértve a regisztrációs eljárások biztonságát is.
- A gyökértanúsítványhoz tartozó szolgáltatói felelősségvállalás mértéke.
- A gyökértanúsítvány elterjedtsége. A gyökértanúsítványra jellemzően nem az adott szolgáltató ügyfeleinek van szüksége, hanem harmadik feleknek, akikkel a szolgáltató ügyfelei kommunikálni szeretnének. Ha egy hitelesítés-szolgáltató nem fordít kellő erőforrásokat a gyökértanúsítvány elterjesztésére, akkor ez a gyökértanúsítvány nemigen lesz használható a gyakorlatban: a rá visszavezethető tanúsítványokat, aláírásokat senki nem fogja elfogadni. Az egyes felhasználói közösségek többek között a fenti szempontok alapján ítélik meg, hogy mely önhitelesített tanúsítványokat fogadják el, azaz mely gyökértanúsítványokban bíznak meg. Ez néha nem alapos mérlegelés eredménye, sokan az alapján bíznak meg bizonyos gyökértanúsítványokban, hogy az általuk használt böngésző- vagy levelezőprogram, esetleg operációs rendszer mely gyökértanúsítványokat fogadja el alapértelmezettként.
- Előfordulhat, hogy jogszabály előírja, hogy bizonyos célra bizonyos gyökértanúsítványokat kell használni. (Például a magyar közigazgatásban olyan tanúsítványokat lehet használni, amelyek a Közigazgatási Gyökér Hitelesítés Szolgáltató /KGYHSZ/ gyökértanúsítványára vezethetőek vissza.)
- Az adott gyökértanúsítványhoz kapcsolódó szolgáltatások, ezen belül például az, hogy mennyire könnyű ellenőrizni a gyökértanúsítványra visszavezethető tanúsítványok visszavonási állapotát. Például nyújt-e OCSP szolgáltatást a gyökértanúsítványban szereplő hitelesítés-szolgáltató, vagy csak ritkán kibocsátott CRL-ek alapján lehet az általa kibocsátott tanúsítványok visszavonási állapotát ellenőrizni.

A tapasztalat azt mutatja, hogy mind hazánkban, mind külföldön több gyökértanúsítványt használnak. Az is tapasztalható, hogy egyes felhasználói közösségek szívesen határoznak meg maguknak saját gyökértanúsítványt. Ekkor az egyes kliensgépeikre kizárólag egy tanúsítványt kell telepíteniük gyökértanúsítványként, és minden tanúsítványt, minden aláírást erre a tanúsítványra vezethetnek vissza.

Például az alábbi közösségek mind-mind saját gyökérrel vagy gyökerekkel rendelkeznek:

- Az elektronikus cégeljárás;
- A magyar közigazgatás;
- Az egyes kereskedelmi hitelesítés-szolgáltatók felhasználói is külön-külön PKI közösségeket alkotnak, mert jellemzően csak a saját szolgáltatójuk gyökerét fogadják el;

- Az egyes EU tagállamok útleveleit kibocsátó rendszerek (többek között a magyar útleveleket kibocsátó rendszer) [130];
- Az európai tachográf-rendszer [176];
- Nemzetközi banki rendszerek résztvevői (bankok, kereskedők, kártyabirtokosok);
- Az egyes multinacionális vállalatok eszközei, munkatársai;
- Egyes nemzetközi szervezetek (pl. NATO) saját gyökértanúsítványt hozhatnak létre, hogy az alájuk tartozó, tagországokon belüli szervezetek ezen gyökértanúsítványra visszavezethető tanúsítványokat használhassanak.
- A világ Windows felhasználói; (Ide tartoznak például az Outlookot és más, a Windows tanúsítványtárát használó levelezőprogramot használó felhasználók, és ide tartozik az összes Internet Explorer felhasználó is.)
- Hasonlóan külön-külön gyökerekkel rendelkeznek a Mozilla, az Opera és a MacOS felhasználók is. A világon minden más, saját tanúsítványtárral rendelkező levelezőprogram, böngésző és egyéb, kommunikációra (is) használható szoftver felhasználói mind külön-külön gyökérhalmazt elfogadó PKI közösségeket alkotnak;

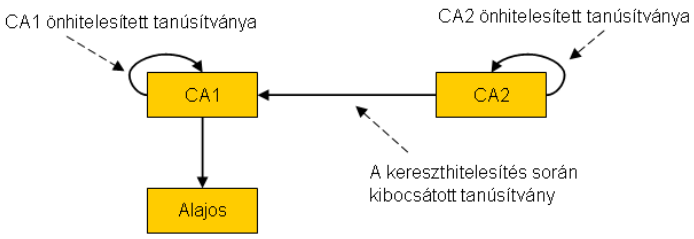
A fentiek egy része földrajzi régiókhoz vagy jogszabály által meghatározott szervekhez kapcsolódik, míg egy másik része alkalmazásokhoz, illetve aláírásokat befogadó szervezetekhez. A fenti lista közel sem teljes körű, nem soroltunk fel benne olyan PKI közösségeket, amelyek Magyarországon közvetlenül nincsenek jelen, így például nem soroltuk fel más országok (pl. EU tagállamok) PKI közösségeit, amelyekkel magyar felhasználók könnyen kapcsolatba kerülhetnek.

Egyes rendszerek (pl. az útlevél, és a tachográf rendszerek) olyan speciális feladatokat látnak el, amelyek esetén fel sem merül az összekapcsolásuk más rendszerekkel. Más tanúsítványok (pl. a bankok által a bankkártyák birtokosai számára kibocsátott tanúsítványok, a Windows operációs rendszer által alapértelmezetten elfogadott tanúsítványok, a közigazgatásban vagy az e-cégeljárás során használható tanúsítványok) esetén viszont természetesen adódik az összekapcsolás lehetősége.

5.2. Köztes hitelesítés-szolgáltatók a tanúsítványláncban

5.2.1. Kereszthitelesítés és felülhitelesítés

Kereszthitelesítésnek nevezzük, amikor egy hitelesítés-szolgáltató egy másik hitelesítés-szolgáltató számára bocsát ki tanúsítványt. (Lásd: 5.2. ábra.)



5.2. ábra. **Kereszthitelesítés: Egy hitelesítés-szolgáltató egy másik hitelesítés-szolgáltató számára bocsát ki (szolgáltatói) tanúsítványt.**

A felülhitelesítés hasonló fogalom, szintén arra vonatkozik, hogy egy hitelesítés-szolgáltató egy másik hitelesítés-szolgáltatónak bocsát ki tanúsítványt. Akkor szokás inkább a felülhitelesítés kifejezést használni, ha kifejezetten hangsúlyozni szeretnénk, hogy az egyik hitelesítés-szolgáltató valamilyen értelemben „fontosabb" vagy „magasabb szintű", mint a másik. A kibocsátott tanúsítványból nem derül ki, hogy melyik hitelesítés-szolgáltató mennyire fontos, így a tanúsítványt felhasználó vagy ellenőrző alkalmazások sem tudnak erre támaszkodni. A két fogalom közötti határvonal nem éles, és műszakilag mindkét esetben pontosan ugyanaz történik.

> **Megjegyzés:** A kereszhitelesítés fogalma másképp is definiálható. Egyes szerzőknél a kereszthitelesítés kölcsönösséget (5.3.1. fejezet) is jelent, amikor mindkét szolgáltató tanúsítványt bocsát ki a másik részére. Más források a kereszthitelesítés szót arra használják, amikor egy szigorúan hierarchikus PKI struktúrában (5.5.1. fejezet) a kereszthitelesítés átlépi e struktúra határait. Mi a fenti, általános definíciót használunk, amely szerint a kereszthitelesítés mindössze annyit jelent, hogy egy szolgáltató egy másik szolgáltatónak vagy egy másik hitelesítő egységnek bocsát ki tanúsítványt.

Azért ezt az általános definíciót használjuk a kereszthitelesítésre, mert feltételezzük, hogy minden hitelesítés-szolgáltatói kulcspárhoz tartozik önhitelesített tanúsítvány is. Gyakori, hogy egy hitelesítés-szolgáltató nem publikálja minden kulcspárjához az önhitelesített tanúsítványt, hanem csak néhány gyökértanúsítványt nevez meg, amelyekre a többi tanúsítványa visszavezethető.

A tanúsítványláncban szereplő minden köztes szolgáltatói tanúsítvány kereszthitelesítés eredményeképp jön létre.

5.2.2. A kereszthitelesítéshez kapcsolódó felelősség

Az érintett fél olyan megbízható gyökértanúsítványokat fogad el, amelyekről tudja, hogy mögöttük megbízható hitelesítés-szolgáltató áll. E szolgáltatókat egyenként megvizsgálja,

ellenőrzi az érintett fél (vagy a rendszergazdája, esetleg az alkalmazásának a fejlesztője).

A tanúsítványláncban is szerepelhetnek szolgáltatói tanúsítványok. Ezekről az érintett fél csak annyit tud, hogy a láncban őket megelőző hitelesítés-szolgáltató *tanúsítványt bocsátott ki a részére*, a legelső szolgáltatói tanúsítványt pedig a gyökér bocsátotta ki. Az érintett fél a láncban hitelesítés-szolgáltatókat lát, amelyek valamilyen típusú kereszthitelesítés során jutottak tanúsítványhoz. Nem tudja, hogy e szolgáltatók milyen viszonyban vannak egymással, illetve a gyökér hitelesítés-szolgáltatóval.

Lehetséges, hogy a láncban egymást követő szolgáltatói tanúsítványok:

- Egyazon hitelesítés-szolgáltató szervezethez tartoznak, csak e szervezet külön hitelesítő egységet működtet. Például egy hitelesítés-szolgáltató külön hitelesítő egységgel bocsáthatja ki a különböző biztonsági szintű tanúsítványokat, így az érintett felek könnyebben ki tudják választani az általuk elfogadott biztonsági szintű tanúsítványokat.
- Egy főbb hitelesítés-szolgáltatóhoz tartoznak, aki kikényszeríti, hogy az alatta lévők megadott szabályzatok szerint működjenek, és például auditálja a működésüket.
- Hasonló biztonsági szintű hitelesítés-szolgáltatókhoz tartoznak, akik egymást biztonságosnak tekintik, és egy kölcsönös megállapodás keretében kereszthitelesítették egymást.
- Teljesen független hitelesítés-szolgáltatókhoz tartoznak, és a szolgáltatói tanúsítvány csupán annyit igazol, hogy az alsóbb szolgáltatóhoz egy adott nyilvános kulcs tartozik, és a felsőbb szolgáltató esetleg semmilyen felelősséget nem vállal az alsóbb szolgáltató működésével kapcsolatban.

Lényeges, hogy az érintett fél a lánc alapján nem tud egyszerűen különbséget tenni a fenti esetek között, ő csak a láncot, a kereszthitelesítések során kibocsátott szolgáltatói tanúsítványokat látja, nem tudja, milyen viszony húzódik meg mögöttük. Aki megbízik a kereszthitelesítő szolgáltatóban, az várhatóan a kereszthitelesített szolgáltató által kibocsátott tanúsítványokat is automatikusan el fogja fogadni.

A kereszthitelesítés során *a kereszthitelesítő szolgáltató kiterjeszti a rá ruházott bizalmat a kereszthitelesített szolgáltatóra.* Ennek következtében a kereszthitelesítés nem pusztán műszaki lépés, jelentős jogi következményei is lehetnek, ezért általában részletes szabályozás és szerződés szokott hozzá tartozni.

A kereszthitelesítés kapcsán igen érdekes felelősségi viszonyok jelenhetnek meg. Tegyük fel, hogy az érintett fél megbízik az X gyökér hitelesítés-szolgáltatóban, és elfogadja a rá visszavezethető tanúsítványokat. Az X hitelesítés-szolgáltató tanúsítványt bocsát ki az Y hitelesítés-szolgáltatónak, e tanúsítvánnyal azt igazolja, hogy egy adott kulcspár az Y szolgáltatóhoz tartozik. Az Y hitelesítés-szolgáltató már nem az X szabályzatai szerint

működik, és minden ellenőrzés nélkül bocsát ki tanúsítványokat. Ha az Y hitelesítés-szolgáltató kibocsát egy tanúsítványt „Alajos” névre, de a tanúsítvány magánkulcsát Manfréd, a támadó birtokolja, az X gyökértanúsítványt elfogadó érintett fél könnyen lehet, hogy elfogadja a tanúsítványt Alajos tanúsítványaként. Felel-e ezért az X szolgáltató?

Kérdés, hogy a felelősségi viszony tranzitív-e, azaz egy gyökér hitelesítés-szolgáltató felelős-e minden, őrá visszavezethető tanúsítványért. Tegyük fel, hogy az X gyökér hitelesítés-szolgáltató bevizsgálja az Y szolgáltatót, majd tanúsítványt bocsát ki a számára. Az Y szolgáltató is bevizsgálja a Z szolgáltatót, és szintén tanúsítványt bocsát ki a számára. A Z szolgáltató nem vizsgálja be a Q szolgáltatót, hanem felelőtlenül kibocsát a számára egy szolgáltatói tanúsítványt, így a Q által kibocsátott, kérdéses megbízható tanúsítványokat mindenki elfogadja, aki az X gyökérben megbízik. Felelős-e X Q működéséért?

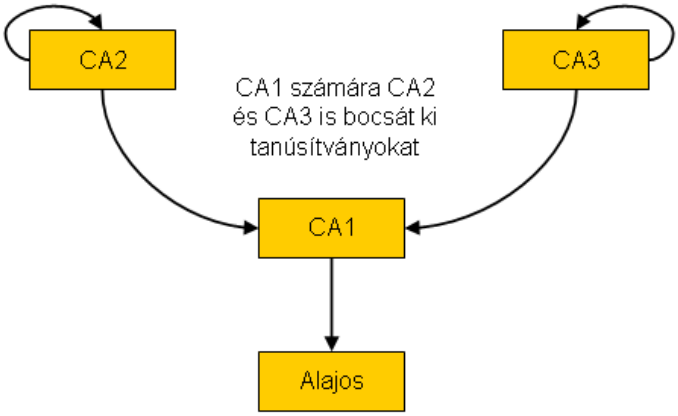
Álláspontunk szerint, ha az érintett félnek a láncban szereplő összes hitelesítés-szolgáltatót meg kell vizsgálnia, akkor irreálisan bonyolult feladatot hárítottunk rá, amit nem fog tudni elvégezni. Úgy véljük, a kereszthitelesítésnek csak akkor van értelme, ha a kereszthitelesítő szolgáltató valamilyen módon felelősséget vállal mind a kereszthitelesített szolgáltató működéséért, mind a kereszthitelesített szolgáltató által kereszthitelesített szolgáltatókért. *Egy hitelesítés-szolgáltatónak* valamilyen módon, valamilyen mértékben *minden, rá visszavezethető tanúsítványért felelősséget kell vállalnia.* Ha ez nem teljesül, a kereszthitelesítésnek és a tanúsítványláncnak nem sok értelme van.

E problémakört minden nagyobb PKI rendszerben igyekeznek kezelni valahogy, nem feltétlenül a fent leírt módon. Több nézőpont, több lehetséges megoldás is van a kérdéssel kapcsolatban, például a Hírközlési Hatóság tájékoztatója szerint a kereszthitelesített (avagy láncolt) szolgáltatónak is a kereszthitelesítő szolgáltatóval azonos szintű követelményeknek kell megfelelnie, így biztosítható, hogy egy felügyelt, „erős” biztonságú szolgáltató ne láncolhasson maga alá egy felügyelet alá nem tartozót. [118] A CA/Browser Forum által kibocsátott EV Guidelines másképpen közelíti meg e kérdést, ott ha egy EV (10.4.3.3. fejezet) tanúsítványokat kibocsátó szolgáltató kereszthitelesít egy másik szolgáltatót, a kibocsátott szolgáltatói tanúsítványban meg kell kötnie, hogy a másik szolgáltató milyen hitelesítési rend szerint bocsáthat ki tanúsítványokat (5.5.3.3. fejezet). [22]

5.3. PKI közösségek összekapcsolása kereszthitelesítéssel

Az egyes PKI közösségek gyakran határoznak meg saját gyökértanúsítványokat, és megkövetelik, hogy az általuk elfogadott tanúsítványokat, aláírásokat e gyökértanúsítványra lehessen visszavezetni. Később az is felmerül, hogy e közösségeket össze lehessen kapcsolni, és az egyik közösségben kibocsátott tanúsítványokat más közösségekben is fel lehessen használni, el lehessen fogadni.

A kereszthitelesítés jelenti az egyik legkézenfekvőbb megoldást különböző PKI közösségek



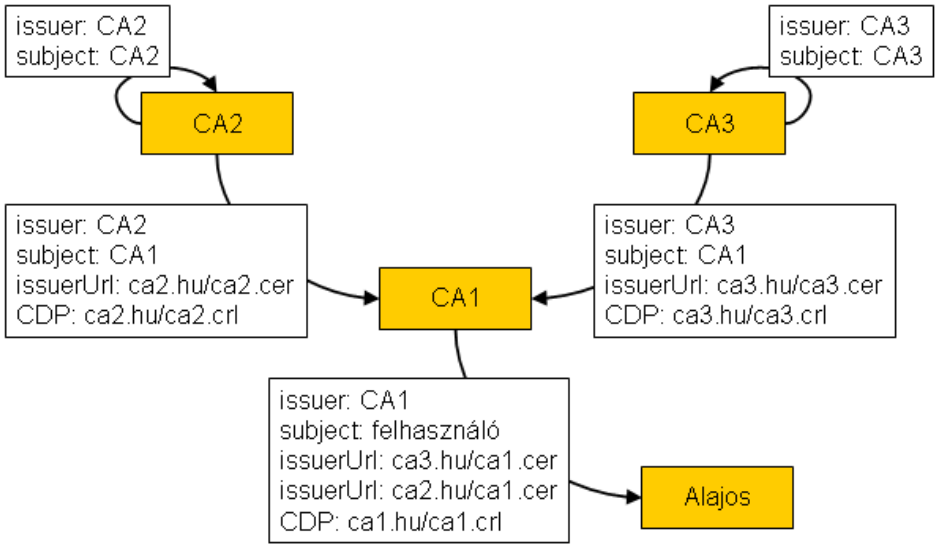
5.3. ábra. **Elágazó tanúsítványlánc, Alajos tanúsítványa a $CA2$ és a $CA3$ gyökér szerint is ellenőrizhető.**

összekapcsolására.

> ***5.2. Példa:*** *Tegyük fel, hogy van egy PKI közösség, amely kizárólag $CA3$ tanúsítványát fogadja el gyökértanúsítványnak. Tegyük fel, hogy egy másik PKI közösség kizárólag $CA2$ tanúsítványát fogadja el gyökértanúsítványnak. Ha egy felhasználó mindkét PKI közösségben részt szeretne venni (például kommunikálni szeretne a magyar közigazgatással is, de egyúttal Outlookban is alá szeretné írni a leveleit), akkor nem lenne jó, ha $CA3$-tól is és $CA2$-től is kellene vásárolnia egy-egy tanúsítványt, hanem azt szeretné, ha egyetlen tanúsítványt használhatna mindkét közösségben.*
>
> *Tegyük fel, hogy Alajosnak már van egy $CA2$-re visszavezethető tanúsítványa, amelyet a $CA1$ köztes egység bocsátott ki. Ha $CA1$ számára $CA3$ is kibocsát egy tanúsítványt, akkor a $CA1$ által kibocsátott végfelhasználói tanúsítványokhoz több tanúsítványlánc is felépíthető. (Lásd: 5.3. ábra.) Az egyik ilyen lánc a $CA2 \rightarrow CA1 \rightarrow$ Alajos, a másik a $CA3 \rightarrow CA1 \rightarrow$ Alajos. (Sőt, ha az ábrán $CA1$ publikálná a saját önhitelesített tanúsítványát, akkor olyan tanúsítványlánc is előfordulhat, amelyben kizárólag Alajos tanúsítványa szerepel, amely $CA1$ gyökértanúsítványa szerint ellenőrizhető.)*
>
> *A tanúsítványok kereszthitelesítés szempontjából releváns mezőit az 5.4. ábra szemlélteti.*

A kereszthitelesítés rugalmasabbá teszi a PKI-t, és átjárhatóságot biztosíthat különböző PKI közösségek között. Ugyanakkor a kereszthitelesítés során a hitelesítés-szolgáltatók között összetettebb viszony jöhet létre, így bonyolultabb lehet egy tanúsítvány ellenőrzése.

Több különböző kereszthitelesítés-megoldás kínálkozik PKI közösségek összekapcsolására. A szakirodalomban részletes elemzések, összehasonlítások olvashatóak róluk. [178], [66]



5.4. ábra. **Elágazó tanúsítványlánc, a tanúsítványban szerepelő kibocsátói megnevezés, alany megnevezés, kibocsátó tanúsítvány URL és CRL terjesztési pont (CDP) URL feltüntetésével**

5.3.1. Kölcsönös kereszthitelesítés

Egyik lehetőség, hogy a két közösségben alkalmazott gyökér hitelesítés-szolgáltató kölcsönösen[1] kereszthitelesíti egymást. Mindkét közösség megőrzi saját gyökerét, de ezentúl a másik gyökérhez tartozó tanúsítványokat is elfogadják. Például az 5.5. ábrán szereplő rendszerben Alajos továbbra is kizárólag a saját hitelesítés-szolgáltatója, azaz $CA1$ gyökértanúsítványát fogadja el megbízhatónak, de a kölcsönös kereszthitelesítést követően Bendegúz tanúsítványát is ellenőrizni tudja. Ilyenkor Alajos a $CA1 \rightarrow CA2 \rightarrow Bendeguz$ láncot fogadja el. Mivel a kereszthitelesítés kölcsönös, ezért Bendegúz hasonló módon fogadja el Alajos tanúsítványát $CA2$ gyökértanúsítványa alapján, de ő az $CA2 \rightarrow CA1 \rightarrow Alajos$ láncot építi fel. Így az ábrán mindkét végfelhasználó tanúsítványához több értelmes és helyes lánc tartozik.

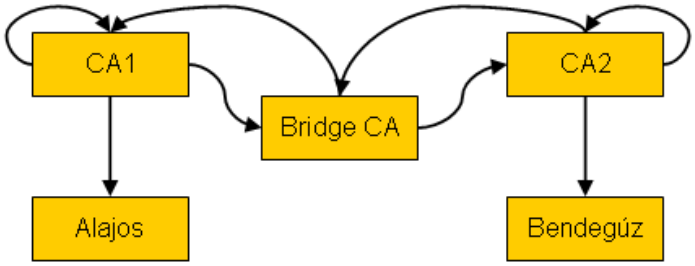
A kölcsönös kereszthitelesítés két hitelesítés-szolgáltató összekapcsolására jó megoldás, több szolgáltató esetén kevésbé hatékony.

5.3.2. Bridge CA

A kölcsönös kereszthitelesítés hátránya, hogy ha sok hitelesítés-szolgáltatót szeretnénk így összekötni, akkor nagyon sok kereszthitelesítésre van szükség. Erre a problémára az ún. *bridge CA* megoldás jelent orvosságot (lásd: 5.6. ábra). Az olyan szituációban lévő hitelesítés-

[1] Létezik olyan definíció is, amely a kereszthitelesítést automatikusan kölcsönös kereszthitelesítésnek tekinti. Itt a korábban kimondott, generikusabb definíciót használjuk.

5.5. ábra. **Kölcsönös kereszthitelesítés – mindkét végfelhasználó tanúsítványa elfogadható mindkét gyökér alapján**



5.6. ábra. **„Bridge CA" megoldás**

szolgáltatót szokás bridge CA-nak nevezni, amelynek fő szerepe, hogy más hitelesítés-szolgáltatókkal kölcsönösen kereszthitelesíti egymást, és így „központi" szerepet tölt be. [67] Így kevesebb kereszthitelesítésre, kevesebb tanúsítványra van szükség. (Például n db hitelesítés-szolgáltató esetén nem $n*n$, hanem csak $2*n$ kereszthitelesítésre van szükség.)

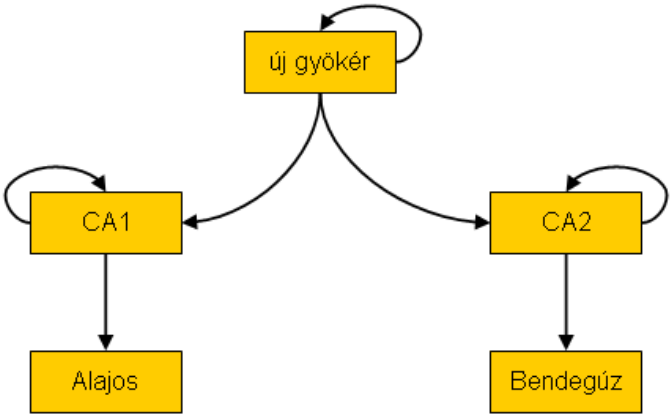
Ekkor Alajos a $CA1 \rightarrow Bridge \rightarrow CA2 \rightarrow Bendeguz$ láncot építi fel Bendegúz tanúsítványával kapcsolatban.

5.3.3. Összekapcsolás új gyökérrel

Az is megoldást jelenthet, ha a meglévő hitelesítés-szolgáltatók felé *új gyökeret hozunk létre* (lásd: 5.7. ábra). Az eddigiek közül messze ez jelenti a legnagyobb felfordulást, mert ekkor minden érintett fél minden kliensszámítógépére telepíteni kell az új gyökértanúsítványt.

Figyeljük meg, hogy a „bridge CA" és az „új gyökér" megoldást bemutató ábrákon majdnem ugyanaz szerepel. Az a különbség, hogy „bridge CA" esetén az új hitelesítés-szolgáltató kölcsönös kereszthitelesítést hoz létre, és nem publikálja saját önhitelesített tanúsítványát. Ezzel szemben az „új gyökér" esetén a kereszthitelesítés nem kölcsönös, és az új gyökér publikálja a saját önhitelesített tanúsítványát.

A két esetben nagyon más a viszony a szolgáltatók között. E második szituációban az várható, hogy a két régi szolgáltató gyökértanúsítványának jelentősége csökken a későbbiekben, ekkor egyértelmű alá-felé rendeltség jelenik meg a szolgáltatók között, ezért ekkor az új gyökér által



5.7. ábra. **Új gyökér bevezetése**

végzett kereszthitelesítést felülhitelesítésnek is szokás nevezni. A két megoldás nem zárja ki egymást, semmi akadálya nincs annak, hogy e két szituáció egyszerre valósuljon meg. A tanúsítványokat ellenőrző féltől függ, hogy milyen (önhitelesített és köztes) tanúsítványokat vesz figyelembe, azaz melyik ábra szerint dolgozik.

Új gyökér esetén felmerül a kérdés, hogy ki működteti majd az új gyökeret. E főgyökér működtetése egyrészt felelősséggel jár: ha a támadó szerez egy tanúsítványt e gyökértől, bármely érintett PKI közösség bármely résztvevőjét megszemélyesítheti; ha a gyökér leáll (pl. nem bocsát ki CRL-t), minden érintett PKI közösség megbénul. Másrészt a főgyökér hatalmat jelent: az alárendelt szolgáltatóknak hozzá kell alkalmazkodni, és a gyökér bármikor bármelyik alrendszert ki tudja tiltani. A felelősség és a hatalom együtt jár, ha csak az egyik jelenik meg, abból ritkán születik értelmes megoldás. Sok esetben azért nem ezt a megoldást választják, mert a résztvevők nem tudnak megegyezni az új gyökérrel kapcsolatban. Egyik résztvevő sem szeretné alárendeli magát egy másiknak, de senki sem szeretné viselni a felelősséget a teljes rendszerért.

Így például nincsen EU-s szintű gyökér, és nincs is terítéken, hogy létrejönne a közeljövőben. A közelmúltban az EU a bizalmi listákra (5.4.2. fejezet) épülő megoldást választotta.

5.3.4. Alegység kereszthitelesítése

Ha egy hitelesítés-szolgáltatónak több hitelesítő egysége (magánkulcsa, és hozzá tartozó megnevezése, azaz DN-je) van, akkor előfordulhat, hogy egy másik hitelesítés-szolgáltató csak bizonyos hitelesítő egységét szeretné kereszthitelesíteni. Az 5.3. ábra egy ilyen esetként is felfogható: $CA2$ egy hitelesítés-szolgáltató gyökértanúsítványa, $CA1$ pedig egy alárendelt hitelesítő egység tanúsítványa. $CA3$ nem szeretné, hogy minden, $CA2$-re visszavezethető tanúsítvány őrá ($CA3$-ra) is visszavezethető legyen, ezért nem $CA2$, hanem csak $CA1$ számára bocsát ki tanúsítványt.

5.3.5. Ideiglenes kereszthitelesítés

A kereszthitelesítés nem feltétlenül végleges lépés. Tegyük fel, hogy az X és Y és Z vállalatok saját belső hitelesítés-szolgáltatót működtetnek. Amíg az X és az Y vállalatok szorosan együtt dolgoznak, X gyökér hitelesítés-szolgáltatója kereszthitelesíti Y hitelesítés-szolgáltatóját, így X szerverei és felhasználói elfogadják Y tanúsítványait, titkosan tudnak levelezni egymással, és Y felhasználói hozzáférnek X szervereihez. Amint megszűnik az együttműködés, X hitelesítés-szolgáltatója visszavonja az Y számára kibocsátott szolgáltatói tanúsítványt, és innentől kezdve X minden jogosultságot elvett Y felhasználóitól. Ha másnap X és Z dolgozik együtt szorosan, X hitelesítés-szolgáltatója kereszthitelesíti Z gyökér hitelesítés-szolgáltatóját vagy annak valamelyik alegységét, és amint megszűnik az együttműködés, visszavonja a szolgáltatói tanúsítványt. [19]

5.4. PKI közösségek összekapcsolása egyéb módon

Előfordulhat, hogy nem PKI alapon, nem kereszthitelesítéssel szeretnénk összekapcsolni meglévő PKI közösségeket. Vizsgáljuk meg, milyen lehetőségeink vannak!

5.4.1. Független gyökerek

Egyik lehetőség, hogy nem kapcsoljuk össze a PKI közösségeket, hanem minden érintett közösség gyökerét eljuttatjuk minden másik PKI közösségbe, és telepítjük a közösség összes számítógépére. E megoldás nehézkessé válik, ha változnak a gyökerek, vagy változnak a PKI közösségek. Minden változás esetén hozzá kell nyúlni minden közösség minden egyes számítógépéhez, emiatt ez nagyon nehezen menedzselhető.

Nehézkessége ellenére sok helyen használják ezt a megoldást, méghozzá igen nagy közösségekben is, általában valamilyen gyökértanúsítvány-menedzsment megoldással együtt. Például a Windows operációs rendszer felhasználói nagyon sok – több száz – gyökér hitelesítés-szolgáltató tanúsítványait fogadják el megbízható gyökértanúsítványként. A gyökértanúsítványok körét a Windows frissítő mechanizmusai segítségével tartja karban a Microsoft. (Lásd: 4.1.4. fejezet.) Hasonló megoldást használ a Mozilla, a MacOS, az Opera és majdnem minden, saját tanúsítványtárral rendelkező alkalmazás. A gyökértanúsítvány-menedzsment megoldás akkor alkalmazható, ha van egy frissítő mechanizmussal rendelkező alkalmazás, amely már ott van a felhasználók számítógépein. Ekkor az alkalmazás fejlesztője központilag karbantarthatja a felhasználók által elfogadott gyökértanúsítványok körét. Gyökértanúsítvány-menedzsment alkalmazás nélkül ez nem képzelhető el. A legtöbb végfelhasználó még kevés gyökeret is alig-alig tud megfelelően menedzselni, több száz esetén ez teljesen esélytelen.

E megoldás azért népszerű, mert az alkalmazás fejlesztője kézben tudja tartani, hogy az alkalmazás milyen gyökereket fogad el alapértelmezetten, ugyanakkor nem kell saját hitelesítés-szolgáltatót üzemeltetnie, és azzal felülhitelesítenie az elfogadott szolgáltatókat. Azzal egyrészt beleszólna a hitelesítés-szolgáltatók működésébe, másrészt a kereszthitelesítéssel felelőssé válna a hitelesítés-szolgáltatók működéséért. Így egy tanúsítványtárat üzemeltet, és a szoftver felhasználási feltételeiben kizárja a felelősséget a hitelesítés-szolgáltatókért. Kérdés, hogy ez mennyire helyénvaló, hiszen a legtöbb végfelhasználó csak a szoftvert ismeri, és nem is tud a benne lévő gyökértanúsítványokról. Szintén kérdés, hogy mennyire felel meg a felhasználók igényeinek az a módszer, ahogy az alkalmazás fejlesztője karban tartja a tanúsítványtárat. Például ha egy perui hitelesítés-szolgáltató gyökere szerepel a Windows tanúsítványtárában, az Outlook magyar felhasználói automatikusan elfogadják az e-maileken szereplő, rá visszavezethető elektronikus aláírásokat. Közben egyáltalán nem biztos, hogy ez az aláírás egyáltalán fokozott biztonságúnak minősül, és szintén kérdés, hogy a perui szolgáltató mennyire biztonságos.

5.4.2. Bizalmi lista (trust services list)

Az Európai Unióban működő szolgáltatókról bizalmi lista (trust services list) jelenik meg, amely az elektronikus aláírás direktíva szerinti minősített szolgáltatókat és a nem minősített, de felügyelt szolgáltatókat tartalmazza.

A bizalmi lista az egyes hitelesítés-szolgáltatók, időbélyegzés-szolgáltatók és archiválás-szolgáltatók szolgáltatásait, szolgáltatói tanúsítványait, a rájuk vonatkozó visszavonási információk helyét és a szolgáltatók elérhetőségét írja le. Nemcsak az aktuális szolgáltatói tanúsítványokat tartalmazza, de visszamenőleg, történetileg is tartalmazhatja a tanúsítványokat. Így ha egy hitelesítés-szolgáltató egy hitelesítő egysége időközben megszűnik, a bizalmi listán később is visszakereshető, hogy ezen egység mettől meddig nyújtott éles szolgáltatást.

Ahogy nincsen közös EU-s gyökér, közös EU-s bizalmi lista sincsen, hanem *minden tagállam saját bizalmi listát bocsát ki a nyilvántartásában szereplő szolgáltatókról.* Létezik egy „listák listája”, amely az egyes tagállamok által közzétett bizalmi listák helyét foglalja össze. A „listák listája” a tagállamok listáinak elérhetőségét, illetve a tagállamok listáit aláíró tanúsítványokat tartalmazza, így a „listák listája” alapján az egyes tagállamok listáinak hitelessége is ellenőrizhető.

A bizalmi listán nem mindig a hitelesítés-szolgáltatók gyökerei szerepelnek, hanem egyes országok esetén a végfelhasználói tanúsítványokat kibocsátó, legalsó szintű hitelesítő egységek jelennek meg a listán. Ha a listán gyökér szerepel, akkor a lista alapján nem feltétlenül lehet megtalálni a tanúsítványláncot a végfelhasználói tanúsítványtól a gyökérig. Ha a listán köztes szolgáltatói tanúsítvány szerepel, akkor a lista alapján nem lehet megtalálni a hiteles gyökeret,

így önmagában a lista alapján biztosan nem építhető fel a tanúsítványlánc. Igaz, elvileg a listán szereplő tanúsítványok is használhatóak megbízhatóható pontként, attól függetlenül, hogy önaláírt tanúsítványok-e.

Ugyanakkor más nézetek szerint az EU-s bizalmi lista nem az EU-s gyökerek (trust anchorok) terjesztésének hiteles módját jelenti, hanem – ha már rendelkezünk a megfelelő gyökérrel – lehetőséget nyújt rá, hogy utánanézzünk, hogy egy egyébként érvényes aláírás minősített vagy felügyelt szolgáltatóra vezethető-e vissza. Így a bizalmi lista nem a gyökerek hiteles terjesztésének módját jelentené, hanem egy további ellenőrzéshez nyújtana eszközt.

A bizalmi lista formátumát az ETSI TS 102 231 specifikáció írja le. [58] Lehet géppel értelmezhető XML fájl, de lehet ember által könnyen olvasható PDF is. Az XML listán XAdES (6.4.1.2. fejezet) aláírás, míg a PDF listán PDF aláírás (6.4.2.2. fejezet) szerepel.

A bizalmi lista még egy nagyon fiatal technológia, még nem minden országban valósították meg maradéktalanul. Van olyan ország, amely nem még nem írja alá a listáit, illetve van olyan ország, ahol a listát aláíró tanúsítvány nem szerepel a „listák listáján". Ez a technológia várhatóan sokat fog fejlődni a jövőben, és ki fog alakulni, hogy pontosan mire használható.

A „listák listája" itt érhető el:
`https://ec.europa.eu/information_society/policy/esignature/trusted-list/tl-mp.xml`
`https://ec.europa.eu/information_society/policy/esignature/trusted-list/tl-hr.pdf`

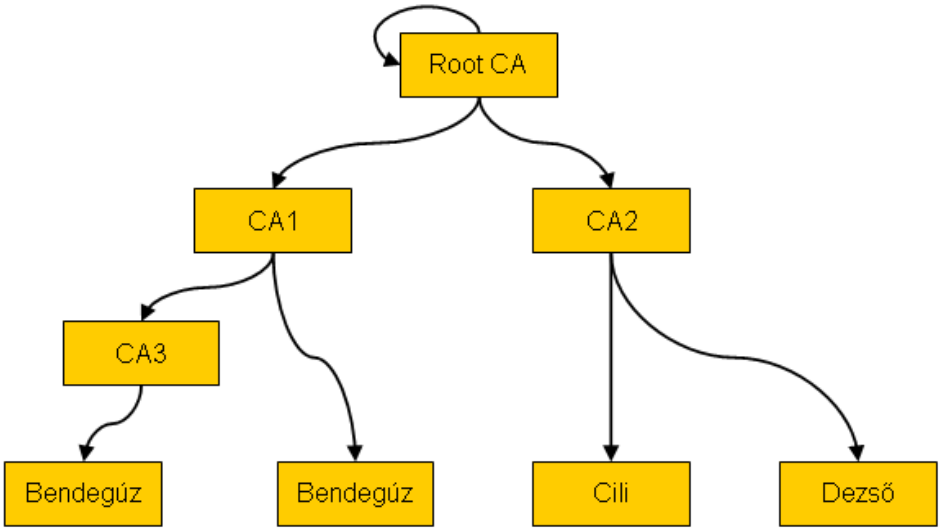
A magyar listát a Nemzeti Média- és Hírközlési Hatóság teszi közzé:
`http://www.nhh.hu/tl/pub/HU_TL.xml`
`http://www.nhh.hu/tl/pub/HU_TL.pdf`

5.4.3. Új rendszer kiépítése, régi rendszerek kivezetése

Megemlítjük még a talán legköltségesebb megoldást meglévő PKI közösségek összekapcsolására: Új gyökeret, új hierarchiát hozunk létre, amely sehogy sem kapcsolódik a korábbi közösségek hierarchiáihoz, a felhasználókat újra regisztráljuk, a régi közösségek elsorvadnak, megszűnnek, és — várhatóan — új rendszer épül ki, immár egyetlen központtal.

Meglévő, már működő rendszerek esetén mindent elvetni és a kályhától újraindulni általában fájdalmas és pazarló megoldás; különösen akkor, ha több jól ismert, és elterjedt megoldás is létezik — például kereszthitelesítés alapján — PKI közösségek összekapcsolására.

Szintén fájdalmas mindezt elölről kezdeni, ha később kiderül, hogy mégsem a megfelelő szempontok szerint vontuk össze a PKI közösségeket.



5.8. ábra. **Hierarchikus PKI**

5.5. Tanúsítványlánc hierarchikus és hálós PKI struktúrákban

A szolgáltatók közötti kereszthitelesítések során kialakuló PKI struktúrákat két nagy csoportba sorolhatjuk, léteznek szigorúan hierarchikus PKI struktúrák, és léteznek hálós PKI struktúrák, ahol nincsen egyértelmű hierarchia.

5.5.1. Hierarchikus PKI

A *hierarchikus PKI* (lásd: 5.8. ábra) esetén minden tanúsítvány egyetlen megbízható gyökérre vezethető vissza, és a rendszerben e megbízható gyökértanúsítvány az egyetlen önhitelesített tanúsítvány. Minden végfelhasználói tanúsítványtól csak egyetlen út vezet a gyökérhez, és a tanúsítványok által meghatározott irányított gráf egy fa (több, egymás mellett élő hierarchia esetén erdő).

Hierarchikus PKI környezetben egy alkalmazásnak könnyű dolga van a tanúsítványlánc felépítésekor. Csak egyetlen gyökér van, csak egyetlen útvonalat találhat, és ha megtalálta, az a „jó” útvonal. Így két alkalmazás garantáltan ugyanazt a láncot építi fel. Egyetlen érdemi probléma állhat elő a tanúsítványlánc felépítése során, ha az alkalmazás nem ismeri az egyik köztes szolgáltatói tanúsítványt.

> ***5.3. Példa:*** *Tegyük fel, hogy egy webszervernek $CA1$ adta ki a tanúsítványát, míg $CA1$ tanúsítványát a Root adta ki. A böngészőprogram megbízható gyökérnek tekinti a Root tanúsítványát. A webszervert hibásan állították be, ezért csak a saját tanúsítványát adja át a böngészőprogramnak, köztes $CA1$ tanúsítványát nem küldi el. Ha a böngészőprogram még nem ismeri $CA1$ köztes tanúsítványát, és a*

webszerver tanúsítványa nem tartalmazza a rá mutató URL-t (vagy tartalmazza ugyan, de a böngészőprogram nem tölti le), a böngésző nem fogadja el érvényesnek a webszerver tanúsítványát.

Ez például Mozilla alatt szokott problémát okozni, az Explorer letölti a hivatkozott köztes tanúsítványokat. A probléma elhárításához be kell állítani a webszervert, hogy a teljes tanúsítványláncot küldje el a böngészőnek.

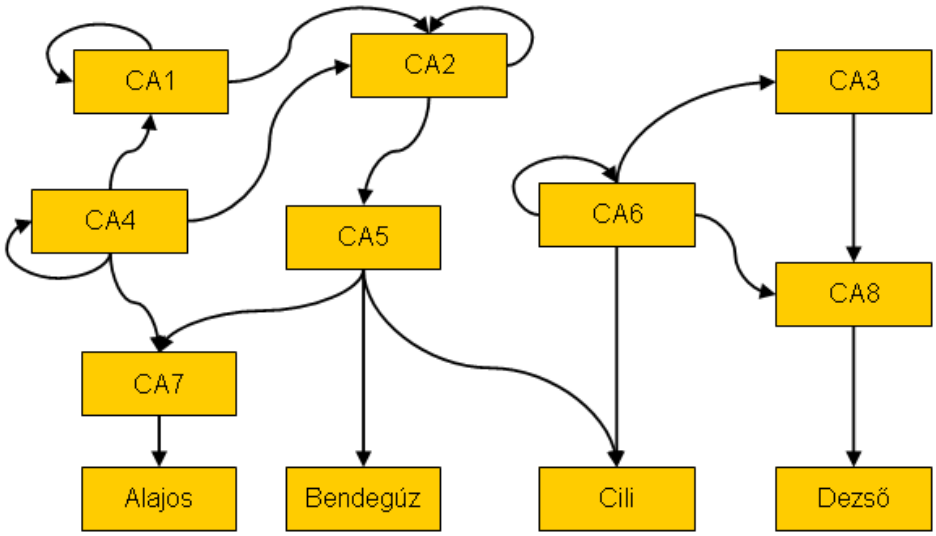
A hierarchikus PKI kényelmes ugyan, de erős korlátai vannak; ez az architektúra nagyon rugalmatlan, nehezen tudja kezelni a változásokat:

- A szigorú hierarchiába nem mindig illeszthetőek be a világ összetett viszonyai, a világ nem szigorúan hierarchikus.
- Ha szervezeti változás történik, a hierarchia nem tudja követni.
- A világon nincs egy főgyökér. Különböző PKI közösségek vannak, amelyek többé-kevésbé átfedik egymást.
- Nem lehet PKI alapon összekapcsolni szigorúan a hierarchikus PKI közösségeket (illetve ekkor megszűnik a szigorú hierarchia). Ha az egyik közösség gyökere A, a másik közösség gyökere B, és A kereszthitelesíti B-t, akkor lesznek felhasználók/alkalmazások, akik A gyökerére vezetik vissza a tanúsítványokat, és lesznek, akik B gyökerét használják. B gyökerét módszeresen ki kell irtani valahogy a rendszerből, különben a kapott rendszer már nem szigorúan hierarchikus, és az ebből adódó előnyök elvesznek.
- Idővel a hosszú időre kibocsátott szolgáltatói tanúsítványok is lejárnak, és esetleg megújításra kerülnek. Előfordul, hogy például egy algoritmusváltás miatt heterogén technológiák jelennek meg, vagy valamely alkalmazás különös igénye miatt egy hitelesítő egységet új tanúsítvánnyal kell ellátni. Amint egy egységnek több tanúsítványa van (akár úgy is, hogy az egyik érvényes, a másik lejárt), elveszítettük a szigorú hierarchia előnyeit.
- Az alkalmazások nagyon kényelmes helyzetben vannak egy szigorúan hierarchikus PKI környezetben. Ha e környezet hierarchiája megszűnik vagy változik, a szaványokat nem teljesen korrektül megvalósító alkalmazások hirtelen nem tudnak tovább működni.

Álláspontunk szerint a hierarchikus PKI-nek számos előnye van ugyan, de hosszú távon, változó környezetben nem tud ennyire sarkosan működni egy rendszer.

5.5.2. Hálós PKI (mesh PKI)

A *hálós PKI* azt jelenti, hogy nincsen szigorú hierarchia. (Lásd: 5.9. ábra.) Ekkor több gyökér is lehet, több köztes hitelesítő egység, és ezek keresztbe-kasul kereszthitelesíthetik egymást.



5.9. ábra. **Hálós (mesh) PKI**

Egy végfelhasználói tanúsítványtól több lehetséges útvonal vezethet egy gyökérhez, de az is lehet, hogy több gyökérhez is vezet útvonal. Előfordulhat, hogy több különböző érintett fél különböző gyökereket használ, akár ugyanarra a célra. A tanúsítványok tetszőleges irányított gráfot meghatározhatnak.

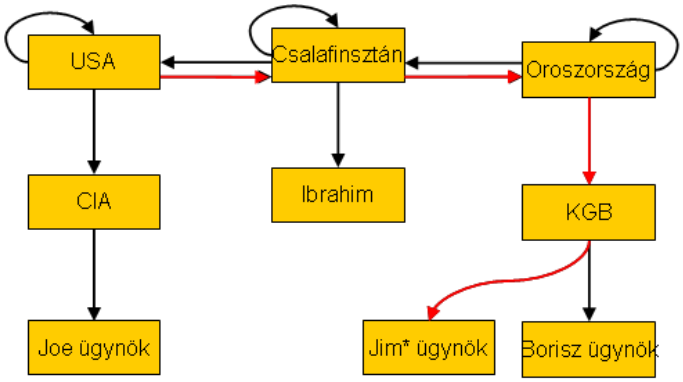
A hálós PKI elsőre ijesztőnek tűnhet ugyan, de e struktúra sokkal közelebb áll a valósághoz. Másrészt egy hálós PKI egy alkalmazás megfelelő konfigurációjával hierarchikussá tehető. Mindössze úgy kell beállítani az alkalmazást, hogy csak a megfelelő megbízható gyökértanúsítványról, és csak a megfelelő köztes tanúsítványokról legyen tudomása, és ekkor az alkalmazás számára akár fastruktúra is biztosítható. A hierarchikus PKI-nak mondott rendszerek általában hálós PKI-k ilyen módon kialakított nézeteit jelentik.

A hálós PKI világában a nehezebb feladat a tanúsítványlánc felépítése. E területen az RFC 4158 javasol tanúsítványlánc-építési stratégiákat. [147] Két probléma merülhet fel:

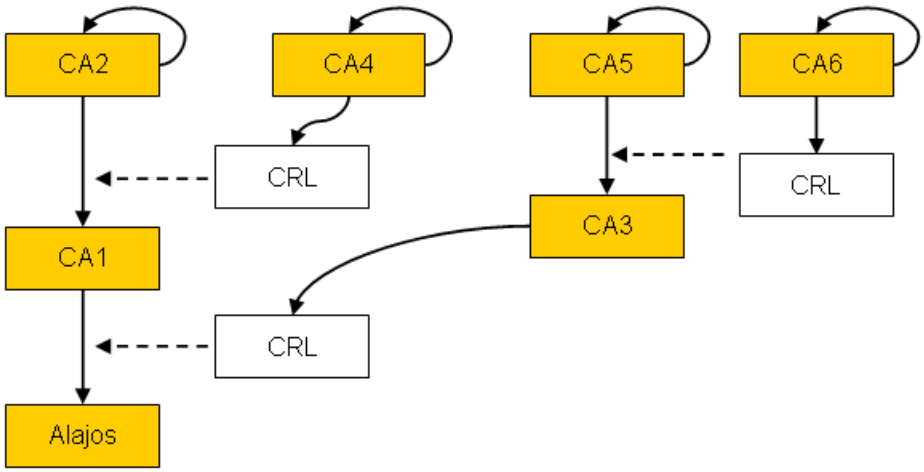
1. Létezik megfelelő tanúsítványlánc, de az alkalmazás nem találja meg. Hálós esetben ez sokkal könnyebben előfordulhat.

2. Nem létezne megfelelő tanúsítványlánc, de az alkalmazás mégis talál egyet.

> ***5.4. Példa:*** *Tekintsük a következő – képzeletbeli – esetet. Az USA és Csalafinsztán megbíznak egymásban, ezért gyökereik kölcsönösen kereszthitelesítik egymást. Csalafinsztán és Oroszország is megbíznak egymásban, az ő gyökereik is kölcsönösen kereszthitelesítik egymást. Ezáltal létrejött egy bizalmi útvonal az USA és Oroszország között is, amelyet senkinek sem állt szándékában kiépíteni. Tegyük fel, hogy erre rájön a KGB, és amerikai*



5.10. ábra. **Nem kívánt tanúsítványlánc**



5.11. ábra. **Állatorvosi ló PKI rendszer, amelyben az Alajos tanúsítványára vonatkozó CRL nem ugyanarra a gyökérre vezethető vissza, mint Alajos tanúsítványa. Sőt, itt a CRL kibocsátójára vonatkozó CRL sem ugyanarra a gyökérre vezethető vissza, mint maga a CRL stb.**

> *DN-nel bocsát ki tanúsítványt a képzeletbeli (így az ábrán csillaggal jelölt) Jim ügynöknek. Ezt sok amerikai alkalmazás – amely csak a DN-t nézi, és nem vizsgálja a teljes láncot – gond nélkül elfogadja amerikainak. (Lásd: 5.10. ábra.)*

Az eddigiekben azt tételeztük fel, hogy egy tanúsítvány egy gyökér alapján ellenőrizhető, ez sem mindig ilyen egyszerű. Előfordulhat, hogy egy tanúsítványra vonatkozó CRL vagy OCSP válasz másik gyökérre vezethető vissza, mint maga a tanúsítvány. (Lásd: 5.11. ábra.)

5.5.3. A tanúsítványlánc korlátozása

Elsősorban hálós esetben fordulhat elő, hogy egy hitelesítés-szolgáltató korlátozni szeretné, hogy mennyiben ruházza át a belé vetett bizalmat az általa kereszt- vagy felülhitelesített szolgáltatókra. Ez hierarchikus esetben is előfordulhat, itt például a gyökér korlátozhatja, hogy az általa felülhitelesített szolgáltatók kiket hitelesíthetnek tovább, és kiknek bocsáthatnak ki tanúsítványt. Tekintsük át az erre szolgáló eszközöket!

5.5.3.1. Alapvető megkötések (Basic Constraints)

A hitelesítés-szolgáltató a tanúsítványban feltüntetett `Basic Constraints` kiterjesztéssel jelölheti, hogy hitelesítés-szolgáltatói vagy végfelhasználói tanúsítványról van-e szó. Ha egy tanúsítvány a benne szereplő `Basic Constraints` szerint végfelhasználói tanúsítvány, akkor nem szerepelhet egy tanúsítványláncban köztes szolgáltatói tanúsítványként.

Csak akkor beszélünk kereszthitelesítésről, ha a kibocsátott tanúsítvány a benne feltüntetett `Basic Constraints` szerint egy hitelesítés-szolgáltató tanúsítványa.

> ***5.5. Példa:*** *Az X hitelesítés-szolgáltató tanúsítványt bocsát ki Manfréd, egy bűnöző részére. Ebben nincsen semmi rossz, Manfréd is jogosult a saját nevében tanúsítványt igényelni. A hitelesítés-szolgáltató feltünteti a tanúsítványban, hogy Manfréd tanúsítványa végfelhasználói tanúsítvány, így senki sem hiheti azt, hogy Manfréd e tanúsítvány szerint jogosult volna további tanúsítványokat kibocsátani.*

Ezen túl, egy hitelesítés-szolgáltató a `Basic Constraints` kiterjesztésben szereplő `PathLenConstraint` mezőben jelölheti, hogy az adott tanúsítvány alatt hány további köztes szolgáltatói tanúsítvány szerepelhet a tanúsítványláncban. Ezzel meggátolhatja, hogy alatta nagyon hosszú láncok alakuljanak ki, amelyek további növekedésére már nincsen befolyással.

> ***5.6. Példa:*** *Az X hitelesítés-szolgáltató tanúsítványt bocsát ki az Y hitelesítés-szolgáltató részére, mert meggyőződött róla, hogy az Y hitelesítés-szolgáltató gondosan jár el, és biztonságosan bocsát ki tanúsítványokat. Ugyanakkor nem szeretné, hogy az Y hitelesítés-szolgáltató további szolgáltatói tanúsítványokat bocsásson ki, mert az Y szolgáltatóval szerződésben lévő szolgáltatókra már nincs ráhatással. Ezért az Y szolgáltató számára feltüntetett szolgáltatói tanúsítványban a* `PathLenConstraint` $= 0$ *értéket tünteti fel, így az Y tanúsítványa után már csak végfelhasználói tanúsítványok következhetnek a láncban.*
>
> *Tegyük fel, hogy Y mégis felülhitelesíti a Z szolgáltatót, és a Z szolgáltató kibocsát egy tanúsítványt Alajos részére! Ezen tanúsítványt az X szolgáltató gyökere alapján várhatóan nem fogják elfogadni az érintett felek, mert* 0*-nál több szolgáltatói tanúsítvány szerepel a láncban Y tanúsítványa alatt.*

> ***Megjegyzés:*** *Azzal, hogy Y felülhitelesíti a Z szolgáltatót, nem feltétlenül szegi meg az X szolgáltatóval kötött megállapodását. Sőt, előfordulhat, hogy a Y ugyanazon hitelesítő egysége másik szolgáltatói tanúsítvánnyal is rendelkezik, amelyben* `PathLenConstraint` > 0*, és e lánc szerint Alajos ezen tanúsítványát akár el is lehetne fogadni.*

5.5.3.2. Megszorítás a megnevezésekre (Name Constraints)

Egy szolgáltatói tanúsítványt kibocsátó hitelesítés-szolgáltató megszoríthatja, hogy a másik szolgáltató milyen megnevezésű (DN) alanyoknak bocsáthat ki tanúsítványokat. Ekkor a lánc innentől lefelé csak a megszorításoknak megfelelő megnevezéseket tartalmazhat.

Például tehető olyan megszorítás, hogy a lánc az adott tanúsítványtól lefelé csak magyarországi (`Country = HU`) tanúsítványokat tartalmazhat. (A korábban bemutatott, 5.10. ábrán szereplő példában e megoldással könnyen orvosolni lehetett volna a problémát.) Szintén megszorítható, hogy a további tanúsítványok csak megadott cégen (`Organization` mező) belül lehetnek, vagy csak megadott e-mail címeket tartalmazhatnak stb. E korlátozás egyaránt érvényes a tanúsítványok alanyának nevére (`Subject DN`) és alternatív neveire (`Subject Alternative Names`).

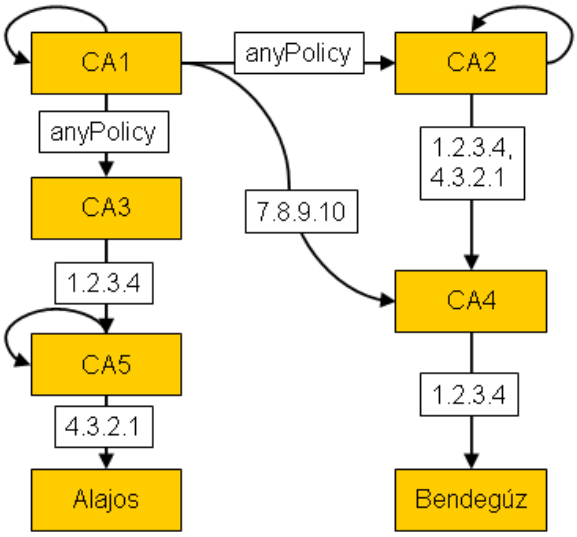
E megoldás nagyon praktikus lehet, de nem minden alkalmazás támogatja ennek minden változatát.

5.5.3.3. Hitelesítési rend OID ellenőrzése

A hitelesítési rend határozza meg, hogy milyen szabályok vonatkoznak egy tanúsítványra: milyen eljárás során bocsátották ki, milyen feltételekkel érhető el a rá vonatkozó visszavonási információ, mekkora felelősséget vállal érte a hitelesítés-szolgáltató stb. A végfelhasználói tanúsítványt kibocsátó hitelesítés-szolgáltató a végfelhasználói tanúsítvány `Certificate Policies` mezejében feltüntetheti azon hitelesítési rendek OID-jét, amelyeknek az adott tanúsítvány megfelel.

A hitelesítés szolgáltatói tanúsítványban szerepelő `Certificate Policies` kiterjesztés mást jelent: nem a szolgáltatói tanúsítványra vonatkozó szabályokat hivatkozza meg, hanem megköti, hogy az adott szolgáltatói tanúsítványt tartalmazó tanúsítványlánc minden elemének tartalmaznia kell a feltüntetett hitelesítési rend OID-k egyikét (vagy egy vele ekvivalens OID-t). Ezáltal a szolgáltatói tanúsítványban megkötheti a szolgáltatói tanúsítványt kibocsátó (azaz a másik szolgáltatót kereszthitelesítő) hitelesítés-szolgáltató, hogy milyen típusú végfelhasználói tanúsítványok szerepelhetnek alatta. (Lásd: 3.4. fejezet.)

Mindemellett, a tanúsítványokban feltüntethető olyan kiterjesztés (`policyMapping`), amellyel egy szolgáltató kijelentheti, hogy valamely hitelesítési rendet egy másik renddel ekvivalensnek ismeri el. Ezen megfeleltetés a láncban lefelé érvényesül. Ezen kívül létezik egy speciális,



5.12. ábra. **Példa hálós PKI struktúra, a tanúsítványokat jelképező nyilakon a `Certificate Policies` mezőben szereplő OID-ket tüntettük fel.**

`anyPolicy` nevű OID, amely azt jelenti, hogy az adott tanúsítvány kibocsátója nem tesz rá megkötést, hogy milyen láncokban használható a tanúsítvány, azaz az `anyPolicy` minden hitelesítési renddel ekvivalens. [152]

Ha egy hitelesítés-szolgáltató `PolicyConstraints` kiterjesztést helyez el egy szolgáltatói tanúsítványban, azzal további megszorításokat tehet a tanúsítványláncokra. Egyrészt, megtilthatja (`inhibitPolicyMapping`), hogy a lánc további részében egy hitelesítés-szolgáltató `policyMapping`-et alkalmazzon, másrészt előírhatja (`requireExplicitPolicy`), hogy a lánc ellenőrzése során kötelező elvégezni a hitelesítési rend OID-k ellenőrzését. Továbbá, az `inhibitAnyPolicy` kiterjesztést használó hitelesítés-szolgáltató megtilthatja, hogy a láncban valaki `anyPolicy` OID-t írjon egy tanúsítványba, azaz innentől kezdve kötelező felsorolni a konkrét hitelesítési rendek OID-jeit.

Előfordulhat, hogy egy köztes szolgáltatói tanúsítványban egyáltalán nem szerepel `Certificate Policies` kiterjesztés. Ekkor az adott szolgáltatói tanúsítványt tartalmazó láncok semmilyen hitelesítési rend OID szerint nem ellenőrizhetőek. Gyökértanúsítványban is feltüntethető hitelesítési rend OID, de az X.509 specifikáció szerint a gyökér nem része a tanúsítványláncnak, így egy alkalmazás ezt figyelmen kívül hagyhatja, de akár használhatja is. Gyökértanúsítványokban tipikusan nem szerepel hitelesítési rend OID.

5.7. Példa: *Tekintsük az 5.12. ábrán szereplő PKI struktúrát! Feltéve, hogy az érintett fél megköveteli, hogy a hitelesítési rend OID-k szabályosan szerepeljenek a tanúsítványláncban, a következő megállapításokat tehetjük:*

- *Alajos tanúsítványa elfogadható a CA5 gyökér szerint, ekkor a*

tanúsítványlánc csak Alajos tanúsítványából áll, e lánc a 4.3.2.1 *rend szerint ellenőrizhető.*

- *Alajos tanúsítványa nem fogadható el a* $CA1$ *gyökér szerint, mert az így felépíthető* $CA1 \rightarrow CA3 \rightarrow CA5 \rightarrow$ *Alajos lánc egyik hitelesítési rend szerint sem ellenőrizhető.* $CA5$ *tanúsítványában ugyanis az* 1.2.3.4 *rend szerepel, míg Alajos tanúsítványa csak a* 4.3.2.1 *rend szerint fogadható el.*
- *Bendegúz tanúsítványa elfogadható a* $CA2$ *gyökér szerint, a tanúsítványlánc ekkor* $CA2 \rightarrow CA4 \rightarrow$ *Bendeguz, amely ellenőrizhető az* 1.2.3.4 *rend szerint.*
- *Bendegúz tanúsítványa elfogadható a* $CA1$ *gyökér szerint, a tanúsítványlánc ekkor* $CA1 \rightarrow CA2 \rightarrow CA4 \rightarrow$ *Bendeguz, amely ellenőrizhető az* 1.2.3.4 *rend szerint.*
- *Bendegúz tanúsítványához létezik egy másik lánc is a* $CA1$ *gyökér szerint, de ez nem fogadható el. E lánc a* $CA1 \rightarrow CA4 \rightarrow$ *Bendeguz, de ez egyik rend szerint sem ellenőrizhető, mert a* $CA4$ *számára* $CA1$ *által kibocsátott tanúsítványában szereplő* 7.8.9.10 *rend nem felel meg a Bendegúz* $CA4$ *által kibocsátott tanúsítványában szereplő* 1.2.3.4 *rendnek. (Ha Bendegúz tanúsítványában szerepelne egy* `policyMapping`*, amely szerint az* 1.2.3.4 *és a* 7.8.9.10 *rendek egyenértékűek, akkor ez a lánc is elfogadható lenne.)*

Az X.509 és RFC 5280 specifikációkban szereplő, tanúsítványlánc ellenőrzésére szolgáló algoritmus többféle módon használható:

- A lánc ellenőrizhető egy (vagy több) meghatározott hitelesítési rend OID szerint. Ekkor az algoritmus csak olyan láncot fogad el, amelynek minden elemében szerepel a megadott OID (vagy a megadott OID-k valamelyike) vagy azzal (illetve azokkal) ekvivalens OID.
- Megkövetelhető, hogy legyen olyan OID, amely szerint a lánc ellenőrizhető.
- Az algoritmus futtatható a hitelesítési rend OID-k ellenőrzése nélkül is, ekkor teljesen figyelmen kívül hagyja a láncban szereplő tanúsítványokban feltüntetett OID-ket.

A legtöbb alkalmazás nem használja a hitelesítési rend OID-k szerinti ellenőrzést, és nagyon sok hibás hitelesítés-szolgáltató létezik a világon, ahol az OID-k nem megfelelő relációban állnak a tanúsítványláncban, illetve ahol nincsenek hitelesítési rend OID-k a köztes szolgáltatói tanúsítványokban.

Ugyanakkor vannak területek, ahol igenis használják a hitelesítési rend OID-k szerinti ellenőrzést, ilyen például az Extended Validation (10.4.3.3. fejezet) webszerver tanúsítványok területe (10.4.3.3. fejezet). [22]

5.5.3.4. Milyen megszorításokat használjunk?

Az eddig leírtak segítségével megszorításokat fogalmazhatunk meg a tanúsítványláncra, de mielőtt rájuk támaszkodunk, célszerű figyelembe venni, hogy az érintett fél milyen alkalmazásokat használhat, és azok támogatják-e az általunk megkívánt megszorításokat. Legtöbb esetben csak arra számíthatunk, hogy az alkalmazás a `Basic Constraints` kiterjesztést támogatja, de sajnos előfordul, hogy még azt sem.

> ***5.8. Példa:*** *Egy Moxie Marlinspike által 2002-ben leírt sebezhetőség szerint egy elterjedt böngésző nem kezeli helyesen a `Basic Constraints` kiterjesztést, és elfogadja, ha egy támadó a saját végfelhasználói tanúsítványával további tanúsítványokat bocsát ki, és így megszemélyesíthet weboldalakat. [104] 2009. decemberében még fennállt a hiba. [105]*

Ha hitelesítési rend OID-alapú ellenőrzést vagy Name Constraints alapú ellenőrzést szeretnénk végeztetni, gondosan nézzünk utána, hogy a kérdéses alkalmazások támogatják-e a nekünk szükséges funkciókat, és nekünk megfelelő módon kezelik-e őket. A szabványok nagyon sok lehetőséget megengednek, e funkciók jelentős része opcionális, és előfordulhat, hogy egy teljesen szabványos alkalmazás figyelmen kívül hagyja őket. A hitelesítés-szolgáltató ekkor azt fogja mondani, hogy ő a tanúsítványban elhelyezte a szükséges figyelmeztetéseket, megszorításokat, és az nem járt el kellően gondosan, aki nem vette figyelembe őket.

5.5.4. Felmerülő kérdések

Láttuk, hogy hálós PKI struktúrák esetén több lehetséges tanúsítványlánc is előfordulhat. Hierarchikus PKI struktúrák esetén ez elvileg nem fordulhatna elő, de kevés PKI struktúra tud hosszú távon tisztán hierarchikus maradni. A következő kérdések merülnek fel a több lánccal kapcsolatban:

- *Melyik tanúsítványlánc az „igazi”?* A szabványok szerint az ellenőrizni kívánt tanúsítványt egy megbízható gyökértanúsítványig (trust anchorig) kell visszavezetni. A tanúsítványt elfogadó érintett félen (valamint a rá vonatkozó jogszabályokon, szabályzatokon) múlik, hogy ő mely önhitelesített tanúsítványokat tekinti megbízható gyökértanúsítványnak. Amennyiben az elfogadó fél több gyökértanúsítványban is megbízik, bármelyikre sikerül visszavezetnie egy tanúsítványt, elfogadhatja azt. A kérdésre adott válasz az elfogadótól függ, aki akár mindkét tanúsítványláncot tekintheti „igazinak”.
- *Mi történik, ha az egyik láncban visszavonnak egy szolgáltatói tanúsítványt?* (Lásd: 5.3. ábra.) Ekkor a visszavonást követő időpontra vonatkozóan már nem lehet visszavezetni

a visszavont szolgáltató által kibocsátott végfelhasználói tanúsítványokat. Ugyanezen végfelhasználói tanúsítványokat a másik lánc szerint továbbra is vissza lehet vezetni egy megbízható gyökértanúsítványig. Ez egyáltalán nem baj: lehet, hogy a szolgáltatói tanúsítványt azért vonták vissza, mert megszűnt a két hitelesítés-szolgáltató között egy szerződés, ez ilyenkor nem feltétlenül érinti a végfelhasználói tanúsítványok érvényességét. Ha azért vonták vissza a kérdéses szolgáltató tanúsítványát, mert a kulcsa kompromittálódott, akkor várhatóan a másik láncban is vissza fogják vonni a tanúsítványát, és így az alatta lévő végfelhasználói tanúsítványokat sehova nem lehet majd visszavezetni.

Ha létezik olyan, számunkra elfogadható tanúsítványlánc, amely szerint a tanúsítvány érvényes, akkor a tanúsítványt érvényesnek tekinthetjük. Nem feltétlenül elég ok a tanúsítvány elutasítására, ha van olyan lánc, amely szerint érvénytelen. Lesz olyan érintett fél, aki csak azt a láncot találja meg, amely szerint érvényes, és ő érvényesnek fogja tekinteni. Ha azt szeretnénk, hogy senki ne fogadja el a tanúsítványt, minden láncot szakítsunk meg.

- *Mi történik, ha az egyik láncban visszavonnak egy végfelhasználói tanúsítványt?* Ugyanaz a végfelhasználói tanúsítvány szerepel mindkét láncban, így ha az egyikben visszavonták, akkor a másikban is visszavonásra került.

- *Ezek szerint egy tanúsítvány (és a tanúsítvány alapján készült aláírás) egyszerre lehet érvényes és érvénytelen?* Ez a kérdés a kereszthitelesítéstől és az elágazó tanúsítványlánctól függetlenül is fennáll. Tanúsítvány (és aláírás) érvényességéről abszolút értelemben nem, hanem csak valamilyen követelményrendszer – például aláírási szabályzat – alapján beszélhetünk. (Lásd: 6.8. fejezet.)

- *Mi történik, ha valaki olyan eszközzel ellenőriz egy tanúsítványt, amelyben nemcsak az ő számára előírt gyökértanúsítványok szerepelnek az elfogadható gyökértanúsítványok között?* A PKI teljes biztonsági modellje arra épül, hogy a felhasználó megbízik bizonyos hitelesítés-szolgáltatókban, és megbízhatóan hozzájutott az ő nyilvános kulcsukhoz. E kulcsokban feltétel nélkül megbízik (a gyökértanúsítványokat még visszavonni sem lehet PKI alapokon), és a felhasználó ezen tanúsítványokat használja más tanúsítványok ellenőrzésére. Ha a felhasználó attól fél, hogy a tanúsítványok ellenőrzésére használt eszköz nem megbízható tanúsítványokat is gyökértanúsítványként fogad el, akkor semmilyen PKI alapú műveletet nem tud biztonsággal elvégezni, egy támadó esetleg bármilyen hamis aláíró, titkosító vagy autentikációs tanúsítványt elfogadtathat vele. Ebben az esetben – kereszthitelesítéstől és elágazó tanúsítványlánctól függetlenül – nem lehet értelmes PKI megoldásról beszélni. A PKI használatához megbízható és megbízható módon konfigurált eszközök szükségesek.

- *Mi történik, ha valaki olyan szoftvert használ, amely nem boldogul az elágazó tanúsítványlánccal?* A szabványok egyértelműen megadják, hogy hogyan, milyen algoritmus szerint kell egy tanúsítványt ellenőrizni, ezen algoritmust az RFC 5280 6.1. (Basic path validation) írja le. Előfordulhat, hogy egy alkalmazás – hibás konfiguráció vagy hibás programozás miatt -– érvényes aláírást érvénytelennek tekint, vagy akár fordítva, de ez a problémakör is teljesen független a kereszthitelesítéstől és az elágazó tanúsítványláncoktól. Hibás (vagy hibásan konfigurált) alkalmazás esetén teljesen kiszolgáltatott helyzetben vagyunk, bármikor kerülhetünk olyan helyzetbe, hogy érvénytelen vagy hamis tanúsítványt fogadunk el tudtunkon kívül. Ebben az esetben – kereszthitelesítéstől és elágazó tanúsítványlánctól függetlenül – nem lehet értelmes PKI megoldásról beszélni. A PKI használatához megbízható és megbízható módon konfigurált eszközök szükségesek.

- *Hogyan oldható meg, hogy bizonyos tanúsítványokat bizonyos megbízható gyökértanúsítványokra, más tanúsítványokat más megbízható gyökértanúsítványokra lehessen visszavezetni?* Ekkor először meg kell állapítanunk, hogy az adott tanúsítványt (vagy aláírást) mely gyökérre kell visszavezetnünk. Ezt követően úgy kell használunk aláírás-ellenőrző alkalmazásunkat, hogy csak a megadott gyökértanúsítványt fogadja el. Ennek az a legegyszerűbb módja – amely alkalmazástól, kereszthitelesítéstől és elágazó tanúsítványlánctól függetlenül alkalmazható – hogy a nem elfogadott gyökértanúsítványt (és esetleg a hozzá tartozó köztes tanúsítványokat) kivesszük az adott alkalmazás tanúsítványtárából. (Így tulajdonképpen azt a helyzetet szimuláljuk, mintha a tanúsítványlánc nem ágazna el, és a PKI struktúra szigorúan hierarchikus volna. Ha az ellenőrző alkalmazás nem tud egyes ágakról, az pont olyan, mintha azok ott sem lennének. Az ilyen problémák nagy része megoldható az ellenőrző alkalmazás megfelelő módon elvégzett konfigurálásával.)

  A szabványok erre finomabb, elegánsabb megoldásokat is kínálnak, ilyen például a hitelesítési rend OID-alapú ellenőrzés.

5.6. Összegzés

- A tanúsítványlánc egy megbízható gyökértől az ellenőrizni kívánt tanúsítványig tart. A lánc alapján győződhetünk meg arról, hogy a tanúsítványt egy megbízható hitelesítés-szolgáltató bocsátotta ki.

- Bárkinek lehet önhitelesített tanúsítványa. Ezen önhitelesített tanúsítvány attól lesz megbízható gyökértanúsítvány, hogy mások megbíznak benne, és használják valamilyen célra.

- A tanúsítványláncban szereplő köztes szolgáltatók tanúsítványai kereszthitelesítés során jönnek létre. A láncot ellenőrző érintett fél nem tudja (automatizmus segítségével) megállapítani, hogy e szolgáltatói tanúsítványok egyazon hitelesítés-szolgáltatóhoz tartoznak, vagy a kereszthitelesítés különböző hitelesítés-szolgáltatók között történt.
- A hierarchikus PKI struktúrákban könnyű megtalálni a tanúsítványláncot, de e struktúrák erős korlátokkal rendelkeznek: rugalmatlanok, és nehezen kapcsolhatók más PKI rendszerekhez (illetve akkor elvész a struktúra hierarchikussága).
- A hálós PKI struktúrák szabadabbak, de ezekben nehezebb feladat megtalálni a szükséges tanúsítványláncot. E problémák megoldhatóak megfelelő alkalmazásokkal, és számos megoldás létezik a nem kívánt tanúsítványláncok kizárására.
- A „fent” és a „lent” (hálós) PKI esetén értelmetlen fogalmak, egy tanúsítványlánccal kapcsolatban már beszélhetünk ilyen fogalmakról.
- Ha létezik olyan, számunkra elfogadható tanúsítványlánc, amely szerint a tanúsítvány érvényes, akkor a tanúsítványt érvényesnek tekinthetjük.
- Nem objektív, hogy egy tanúsítvány/aláírás érvényes-e; mindez csak egy adott ellenőrzési módszertan (aláírási szabályzat) kontextusában lehet objektív.
- Jó minőségű alkalmazásokat kell használni, és azt jól kell beállítani, különösen ha elektronikus aláírást szeretnénk ellenőrizni. Hibásan működő, vagy rosszul beállított alkalmazással bármilyen eredményre juthatunk.

6. fejezet

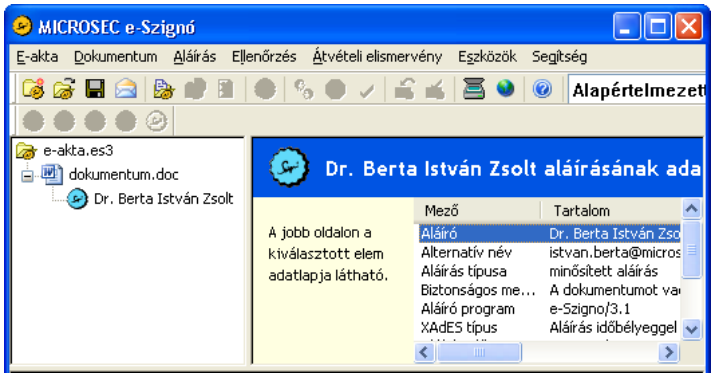
Elektronikus aláírás

„"Szolgáltass ki, kedves unokahúgom, ezen első váltómnál fogva fegyverhordozómnak, Sancho Panzának három szamarat abból az ötből, amelyet gondjaidra bíztam. A három szamár ellenértékét már megkaptam fegyverhordozómtól, aki e sorok átadójával azonos. Kelt a Feketehegyen, folyó év augusztus 22-én."
– Ez teljesen rendben van, – mondta Sancho – csak alá kell írnia.
– Nem kell aláírnom, – válaszolt Don Quijote – csak ideírom a kézjegyemet. Ez pedig nem csak három szamárra, hanem még háromszázra is elegendő lenne."

– Cervantes, Don Quijote – Randóti Miklós átdolgozása

Hiteles dokumentumokat nemcsak papír alapon, hanem elektronikusan is létrehozhatunk. Míg a papír alapú dokumentumok esetén a dokumentum hitelességét a rajta szereplő kézzel írott aláírás biztosítja, az elektronikus dokumentumokat elektronikus aláírással hitelesíthetjük. *Az elektronikus aláírás nem a beszkennelt kézzel írott aláírást jelenti, hanem a kódolás egy speciális változata.* Ha egy dokumentumot elektronikusan írunk alá, akkor olyan módon kódoljuk, hogy a létrejött kódolt dokumentum hitelességét annak szerkezete biztosítja. *Az elektronikus aláírás hitelességét jogszabály is elismeri.* [180]

Amikor aláírunk egy dokumentumot, mindig valamilyen állítást, nyilatkozatot teszünk: például elfogadjuk a dokumentum tartalmát, egyetértünk a benne foglaltakkal, esetleg azt igazoljuk, hogy átvettük, elolvastuk az adott dokumentumot. Aláírásunk ezen állításunk bizonyítékaként szolgál. Egyrészt azt bizonyítja, hogy a dokumentumot valóban mi írtuk alá, másrészt azt bizonyítja, hogy valóban ezt a dokumentumot írtuk alá. Kézzel írott aláírás esetén az aláírás az egyedileg jellemző grafika (amelyet más személy csak viszonylag nehezen tud létrehozni), és az aláíráskor használt tinta úgy kapcsolódik hozzá a papírhoz, hogy azt nem (vagy csak nehezen) lehet onnan eltávolítani. Az elektronikus aláírás szintén az aláíróra jellemző, és szintén elválaszthatatlanul hozzákapcsolódik az aláírt dokumentumhoz, de egészen más eszközökkel teljesíti ugyanezen követelményeket.



6.1. ábra. **Minősített elektronikus aláírás megjelenítése az e-Szignó program segítségével**

Amikor elektronikusan írunk alá egy dokumentumot, egy összetett kódolási műveletet végzünk el rajta. Az így létrejött aláírt, kódolt dokumentum olyan speciális szerkezettel rendelkezik, amelynek alapján bizonyítható, hogy ki volt az, aki a kódolást elvégezte, és az is bizonyítható, hogy az illető pontosan mely dokumentumot kódolta. Így ha ugyanaz a személy több dokumentumot ír alá, az egyes dokumentumokhoz tartozó aláírásainak különböznie kell egymástól. Ha egy bűnöző átmásolja valakinek az elektronikus aláírását egy másik dokumentumra (amit az illető nem írt alá), kimutatható lesz, hogy az aláírás és ezen másik dokumentum nem tartoznak össze.

Ahogy a papír alapú aláírás bíróság előtt felhasználható bizonyíték, az elektronikus aláírás is az. Az elektronikus aláírásról szóló 2001. évi XXXV. törvény szerint a legalább fokozott biztonságú elektronikus aláírással ellátott dokumentum *megfelel az írásba foglalás követelményeinek*, a minősített aláírással ellátott dokumentum pedig – a polgári perrendtartásról szóló törvény értelmében – *teljes bizonyító erejű magánokirat* (akárcsak a két tanú előtt, vagy a közjegyző előtt aláírt dokumentum). (Lásd: 6.1. ábra.)

Az elektronikus aláírás technológia jellegzetessége, hogy az aláírt dokumentum hitelessége a dokumentum kódolásából, szerkezetéből adódik. Így, ha az elektronikusan aláírt dokumentumról másolatot készítünk, a másolat is hiteles lesz, pontosan úgy, mint az eredeti. Az elektronikusan aláírt dokumentum hitelességét nem befolyásolja, hogy az hol, kinél van, vagy hogyan továbbítjuk, hiszen a dokumentum hitelességét a dokumentum szerkezete hordozza. Éppen ezért, azonnal kimutatható, ha az aláírást követően a dokumentum akár a legkisebb mértékben is megváltozik; a megváltozott dokumentumhoz már egészen más aláírás kellene, hogy tartozzon.

Egy aláírás elkészítéséhez az aláírandó dokumentum mellett saját ún. aláírás-létrehozó adatra (azaz magánkulcsra) van szükségünk. Az aláírás-létrehozó adat úgy képzelhető el, mint egy nagyon hosszú szám. Amikor aláírunk egy dokumentumot, akkor a saját aláírás-létrehozó adatunk alapján kódoljuk azt. Nagyon fontos, hogy mindenkinek különböző aláírás-létrehozó

6.2. ábra. **Az elektronikus aláírás létrehozásához szükséges magánkulcsunkat intelligens kártyával védhetjük**

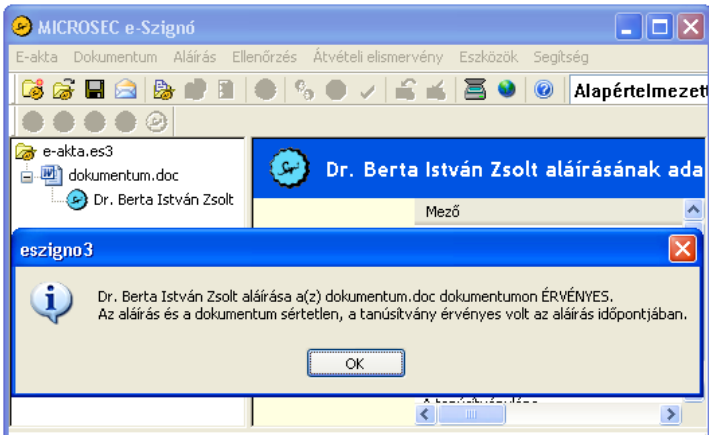
adattal kell rendelkeznie (különben az emberek „egyforma” aláírásokat hoznának létre), és mindenkinek titokban kell tartania a saját aláírás-létrehozó adatát. Ha egy bűnöző megszerzi az aláírás-létrehozó adatunkat, onnantól pontosan olyan aláírásokat hozhat létre, mint mi. (Ez olyan, mintha sok-sok aláírt üres papírlapot adtunk volna neki.) Azért, hogy ez ne fordulhasson elő, az aláírás-létrehozó adatot különösen gondosan szokás tárolni, kezelni. Gyakori megoldás, hogy az aláírás-létrehozó adatot intelligens kártyán (vagy más eszközön) tartják. (Lásd: 6.2. ábra.) Ekkor, amíg a kártya a zsebünkben van, biztosak lehetünk benne, hogy senki nem szerezte meg az aláírás-létrehozó adatunkat.

Ha valaki aláírt dokumentumot kap, *ellenőrzi az aláírást*. Kézzel írott aláírás esetén megpróbálhatja kiolvasni az aláírást (megnézheti, hogy valóban az aláíró neve szerepel-e ott), vagy megpróbálhatja összehasonlítani egy aláírási címpéldányban vagy az aláíró személyi igazolványában (vagy más, megbízható, hiteles féltől származó igazolásban) lévő hitelesnek tekinthető aláírással.

Az aláírás-létrehozó adatra azért van szükségünk, hogy aláírhassunk valamit. Ha azt szeretnénk, hogy az aláírásainkat más is felismerje, elfogadja, *tanúsítvány*t kell beszereznünk az aláírás-létrehozó adatunkhoz. Tehát érvényes aláírás létrehozásához tanúsítvány és aláírás-létrehozó adat szükséges. Ezek mellett már csak elektronikus aláírás létrehozására (vagy ellenőrzésére) szolgáló szoftverre – például az e-Szignó programra, Acrobat Readerre vagy levelezőprogramra – van szükség, amely elvégzi az aláírás létrehozásához vagy ellenőrzéséhez szükséges bonyolult kódolási lépéseket. (Lásd: 6.3. ábra.)

Az elektronikus aláírás ellenőrzéséhez szükséges tanúsítványt megbízható szervezet, ún. hitelesítés-szolgáltató bocsátja ki.

Az elektronikus aláírás használatához (létrehozásához és ellenőrzéséhez) szükséges bonyolult matematikai műveleteket tipikusan számítógépes program segítségével hajtjuk végre. (Lásd:



6.3. ábra. **Az elektronikus aláírást számítógépes programok segítségével tudjuk ellenőrizni. A program ilyenkor elvégzi helyettünk a bonyolult műveleteket, és a felhasználónak csak az ellenőrzés eredményével kell foglalkoznia.**

2. fejezet.) Ezen programok felhasználói felülete elrejti a technológia bonyolultságát, és csak azt jelzi, hogy mely dokumentumot kik írták alá. Így az elektronikus aláírás használatához nem szükséges megértenünk a mögötte lévő bonyolult kódolási műveletek részleteit.

6.1. Minősített és fokozott biztonságú elektronikus aláírás

Az elektronikus aláírásról szóló jogszabályok bizonyos műszaki technológiákhoz, matematikai műveletekhez bizonyító erőt kapcsolnak, ezáltal egyes bitsorozatokhoz akár a kézzel írott aláírással egyenértékű bizonyító erő tartozhat. Az elektronikus aláírás olyan kriptográfiai kódolás, amely joghatás kiváltására is alkalmas, így az elektronikus aláírás a jog és a kriptográfia egy érdekes határterületét jelenti.

Jogi szempontból a 2001. évi XXXV. törvény (Eat.) határozza meg elektronikus aláírás fogalmát (1.4.2. fejezet). [180]

> *Eat. 2. § „6. Elektronikus aláírás: elektronikusan aláírt elektronikus dokumentumhoz azonosítás céljából logikailag hozzárendelt vagy azzal elválaszthatatlanul összekapcsolt elektronikus adat. ”*

Az elektronikus aláírás ezen alapvető definíciójának egészen egyszerű, kriptográfiai technológiákat nem használó megoldások is megfelelhetnek. A jogszabály több különböző biztonsági szintű elektronikus aláírást különít el egymástól. A fokozott biztonságúnak nem minősülő elektronikus aláírás (amelyet a törvény 2004 előtt hatályos változata „egyszerű elektronikus aláírás” néven is nevezett) nem feltétlenül épül kriptográfiai megoldásokra. Ide tartozik, ha valaki egy dokumentum alján feltünteti a nevét, vagy odamásolja a beszkennelt, kézzel írott aláírását.

6.1. Példa: *Alajos e-mailt küld Bendegúznak, az e-mailben 5kg krumplit rendel Bendegúztól. Az e-mailt így fejezi be: „Üdvözlettel, Alajos". Ez tekinthető úgy, mintha az e-mailt fokozott biztonságú aláírásnak nem minősülő elektronikus aláírással (más néven „egyszerű" elektronikus aláírással) látta volna el.*

Az Eat. alapvetően a kriptográfiai megoldásokra épülő, fokozott biztonságú elektronikus aláírásokról (és a minősített aláírásokról) szól, a fokozott biztonságúnak nem minősülő elektronikus aláírással ellátott dokumentumról mindössze annyit mond, hogy nem szabad egy elektronikus dokumentumot pusztán azért elutasítani, mert az elektronikusan létezik.

Eat. 3 § (1) „Elektronikus aláírás, illetve dokumentum elfogadását – beleértve a bizonyítási eszközként történő alkalmazást – megtagadni, jognyilatkozat tételére, illetve joghatás kiváltására való alkalmasságát kétségbe vonni – a (2) bekezdés szerinti korlátozással – nem lehet kizárólag amiatt, hogy az aláírás, illetve dokumentum elektronikus formában létezik. "

A törvény többi része, az erősebb, kriptográfiát is használó megoldásokról szól, és a továbbiakban mi is csak a kriptográfiai megoldásokra épülő, legalább fokozott biztonságúnak minősülő elektronikus aláírást értjük az elektronikus aláírás kifejezés alatt.

Megjegyezzük, hogy néhány területen a törvény kizárja az elektronikus aláírás alkalmazását. Ezek jellemzően olyan területek, ahol a felhasználók egymással nagyon bizalmas viszonyban vannak, és könnyen elképzelhető, hogy hozzáférnek egymás kártyájához, vagy ismerik egymás PIN kódját.

Eat. 3 § (2) „ A Magyar Köztársaság Polgári Törvénykönyvének 598-684. §-aiban szereplő, illetve a házasságról, a családról és a gyámságról szóló 1952. évi IV. törvény szerinti jogviszonyokban nem lehet az elektronikus formán kívüli dokumentumokat mellőzve, csak elektronikus aláírást felhasználni, illetve elektronikusan aláírt elektronikus dokumentumot készíteni. "

6.1.1. Fokozott biztonságú elektronikus aláírás

Az Eat. 2 § 15. a következő módon definiálja a fokozott biztonságú elektronikus aláírás fogalmát:

„Fokozott biztonságú elektronikus aláírás: elektronikus aláírás, amely
a) alkalmas az aláíró azonosítására,
b) egyedülállóan az aláíróhoz köthető,
c) olyan eszközökkel hozták létre, amelyek kizárólag az aláíró befolyása alatt állnak, és

d) a dokumentum tartalmához olyan módon kapcsolódik, hogy minden – az aláírás elhelyezését követően a dokumentumon tett – módosítás érzékelhető. ”

E követelmények már csak kriptográfiai megoldások segítségével teljesíthetőek, és e biztonsági szinthez már jogkövetkezményt is kapcsol az Eat.: A törvény értelmében a fokozott biztonságú (vagy minősített) elektronikus aláírással ellátott dokumentum írásba foglaltnak minősül.

Eat. 4. § (1) „ Ha jogszabály a 3. § (2)-(4) bekezdésében foglaltakon kívüli jogviszonyban írásba foglalást ír elő, e követelménynek eleget tesz az elektronikusan aláírt elektronikus dokumentumba foglalás is, ha az elektronikusan aláírt elektronikus dokumentumot fokozott biztonságú elektronikus aláírással írják alá. ”

A fokozott biztonságú elektronikus aláíráson belül is élesen elkülöníthető két nagy kategória: a nyilvános körben használt aláírások, melyeket bárki aláírásnak tekinthet, és a csak zárt körben használható aláírások.

6.1.1.1. Zárt körben használható fokozott biztonságú aláírás

Fokozott biztonságú elektronikus aláírás minden olyan műszaki módszerrel készíthető, amely teljesíti a fenti – Eat. 2 § 15. szerinti – követelményeket, feltéve, hogy az érintettek megállapodnak benne, hogy pontosan milyen módszert használnak.

Eat 1 § (2) „Egymással szerződéses viszonyban álló felek az elektronikusan aláírt elektronikus dokumentumok szervezetek (személyek) korlátozott és zárt körében való elfogadásának feltételeit – a 3. § (2) bekezdése szerinti korlátok között – e törvény szabályaitól eltérően is megállapíthatják. ”

A zárt körben használt elektronikus aláírás olyan közösségben képzelhető el, amelynek tagjai egymással szerződéses viszonyban állnak, és megállapodnak benne, hogy pontosan mit tekintenek elektronikus aláírásnak. Ilyen közösséget jelenthetnek például egy vállalat munkatársai, esetleg a vállalat partnereivel, ügyfeleivel együtt, vagy egy egyetem hallgatói. A lényeg, hogy minden fél aláírja, elfogadja, hogy az adott technológiával hitelesített dokumentumokat fokozott biztonságú aláírással ellátott, írásba foglalt dokumentumnak tekinti. Gyakori megoldás, hogy egy vállalat saját zárt körben működő hitelesítés-szolgáltatót (4.1.1. fejezet) állít fel, és az e szolgáltató tanúsítványai szerint ellenőrizhető aláírásokat is fokozott biztonságú aláírásnak tekintik. Ugyanakkor a zárt körben használt elektronikus aláírás nem kell, hogy PKI alapon működjön. Szintén jó megoldás lehet, ha a felek arról kötnek szerződést, hogy a PGP alapon (2.8. fejezet) készített aláírásaikat fogadják el fokozott biztonságú aláírásnak. A törvény technológia-független, elvileg bármilyen megoldás

használható, az sem feltétlenül szükséges, hogy nyilvános kulcsú kriptográfia segítségével készüljenek az aláírások.

> ***6.2. Példa:*** *Alajos és Bendegúz feltalálnak egy saját nyilvános kulcsú kriptográfiai algoritmust, és megállapodnak benne, hogy az ezen algoritmussal hitelesített dokumentumokat fokozott biztonságú aláírásnak tekintik. Ilyen módon érvényes, írásba foglaltnak minősülő szerződéseket köthetnek egymással, így pl. köthetnek egy szerződést arról, hogy Alajos eladja Bendegúznak a számítógépét.*
>
> *Tegyük fel, hogy Bendegúz elküld egy ilyen szerződést Cilinek (mert igazolni szeretné, hogy valóban megvásárolta a számítógépet). Cili valószínűleg nem tud mit kezdeni az aláírással, könnyen lehet, hogy nincs olyan programja, amely értelmezni tudná. Ha történetesen lenne megfelelő programja, ő akkor sem egy írásba foglalt szerződést kapott, mert ő nem egyezett bele, hogy Alajos és Bendegúz megoldásával fokozott biztonságú aláírás hozható létre. Sőt, arról sem tud meggyőződni, hogy a kérdéses megoldás valóban megfelel-e az Eat. követelményeinek.*
>
> *Ha Bendegúz nem fizeti ki a számítógépet, Alajos bíróság elé viheti az ügyet. Ott a szerződés írásba foglalt szerződésként jelenik meg, hiszen mind Alajos, mind Bendegúz elfogadták, hogy az adott technológiával fokozott biztonságú aláírást hoznak létre. Ugyanakkor nagyon érdekes jogi eset kerekedhet, ha kiderül, hogy az alkalmazott megoldás (beleértve a kriptográfiai algoritmust, és a kulcsmenedzsmentet is) mégsem felel meg az Eat. fokozott biztonságú aláírásra vonatkozó követelményeinek (pl. Alajos utólag észrevétlenül megváltoztathatta a szerződésben szereplő összeget). Megjegyezzük, erre nagy az esély, mert a saját, „házi" készítésű, kriptográfiai algoritmusokról szinte minden esetben súlyos hibák mutathatók ki.*

Lényeg, hogy ha a felek szerződésben lerögzítik, bármilyen megoldással hozhatnak létre fokozott biztonságú elektronikus aláírást. Igaz, ekkor az ő feladatuk biztosítani, hogy a megoldás megfeleljen az Eat. követelményeinek.

6.1.1.2. Nyilvánosan használt fokozott biztonságú aláírás

Léteznek olyan megoldások, amelyek használatáról nem kell előzetesen megállapodni, mert a velük készült aláírást – az Eat. alapján – bárki fokozott biztonságú aláírásnak tekinti. Ha egy aláírás olyan tanúsítványra épül, amelyet nyilvánosan működő hitelesítés-szolgáltató bocsátott ki, és az aláírás műszakilag érvényes, akkor az aláírt dokumentum – külön szerződés nélkül is – írásba foglaltnak minősül. Magyarországon a Nemzeti Média- és Hírközlési Hatóság vezet nyilvántartást a nyilvánosan működő hitelesítés-szolgáltatókról, és folyamatosan vizsgálja és ellenőrzi a működésüket. A nyilvánosan működő hitelesítés-szolgáltatók olyan

technológiákat használnak, amelyek teljesítik az Eat. által támasztott követelményeket. A nyilvánosan használt fokozott biztonságú aláírásokhoz hitelesítés-szolgáltató szükséges, így ezek mindenképpen PKI alapúak.

Megjegyezzük, hogy a nyilvánosan használt fokozott biztonságú aláírás jogilag nem jelent erősebb bizonyítékot, mint a zárt körben használt. Mindössze arról van szó, hogy jól ismert, nyilvánosan elérhető technológiával készült, így mind az érintett fél, mind a bíróság sokkal könnyebben meg tudja állapítani, hogy mivel áll szemben. Sok esetben nem oldható meg, hogy minden érintett fél előzetesen beleegyezzen, hogy egy speciális technológia szerint hoznak létre fokozott biztonságú elektronikus aláírást, így ez esetekben eleve csak nyilvánosan használt megoldás jöhet szóba.

6.1.2. Minősített elektronikus aláírás

A különösen szigorú követelményeknek megfelelő fokozott biztonságú elektronikus aláírást minősített elektronikus aláírásnak nevezi az Eat.

> *Eat. 2 § 17. „Minősített elektronikus aláírás: olyan – fokozott biztonságú – elektronikus aláírás, amelyet az aláíró biztonságos aláírás-létrehozó eszközzel hozott létre, és amelynek hitelesítése céljából minősített tanúsítványt bocsátottak ki. ”*

Más szavakkal, egy fokozott biztonságú elektronikus aláírás akkor tekinthető minősített elektronikus aláírásnak, ha:

- *Az aláírás biztonságos aláírás-létrehozó eszközzel készült.* Ekkor a magánkulcsot biztonságos aláírás létrehozó eszköz védi, tehát a magánkulcsot nem lehet észrevétlenül lemásolni, és a magánkulcs használata előtt az eszköz (pl. PIN kód alapján) meggyőződik az aláíró kilétéről.
- *Minősített tanúsítvány igazolja, hogy a magánkulcshoz tartozó nyilvános kulcs kinek a birtokában van.* Minősített tanúsítványt kizárólag nyilvánosan működő, minősített hitelesítés-szolgáltató bocsáthat ki, és a minősített szolgáltatókat a Nemzeti Média- és Hírközlési Hatóság különösen szigorúan ellenőrzi. Minősített tanúsítványt kizárólag személyes regisztráció alapján szabad kibocsátani, és ha a magánkulcs kompromittálódott, haladéktalanul vissza kell vonni a tanúsítványt. A minősített hitelesítés-szolgáltató anyagi felelősséget vállal a tanúsítvánnyal okozott károkért (4.8.2. fejezet), és felelősségbiztosítással is rendelkezik.

A minősített elektronikus aláírásra szigorú szabályok vonatkoznak, így *ha érvényes minősített elektronikus aláírással ellátott dokumentummal találkozunk*, nyugodtan feltételezhetjük, hogy amikor az aláírást készítették, az aláírás-létrehozó adat kizárólag a tanúsítványban szereplő

személy birtokában volt. Így feltételezhetjük, hogy *a tanúsítványban szereplő személy írta alá dokumentumot.* Ennek megfelelően, az Eat. bizonyító erőt rendel a minősített elektronikus aláíráshoz.

> *Eat. 4. § „(2) Ha az elektronikusan aláírt elektronikus dokumentumon minősített elektronikus aláírás szerepel és az aláírás ellenőrzésének eredményéből más nem következik, vélelmezni kell, hogy a dokumentum tartalma az aláírás óta nem változott.*
> *[...]*
> *(5) Amennyiben az aláírás-létrehozó adatot olyan szolgáltató helyezte el az aláírás-létrehozó eszközön, amely az adat elhelyezésekor e szolgáltatás tekintetében a szolgáltatók nyilvántartásában minősítettként szerepelt, az ellenkező bizonyításig vélelmezni kell, hogy az aláírás-létrehozó adat kizárólag a szolgáltatást igénybe vevő birtokában van. ”*

A polgári perrendtartásról szóló törvény értelmében a minősített elektronikus aláírással ellátott dokumentum *teljes bizonyító erejű magánokirat*, ugyanúgy, mint a közjegyző előtt vagy a két tanú előtt aláírt okirat, vagy mint az az okirat, amelyet valaki teljes egészében saját kézzel írt. [184]

Így ha egy dokumentumon érvényes minősített elektronikus aláírás szerepel, a bíróságnak azt kell feltételeznie, hogy az aláíráshoz kapcsolódó tanúsítványhoz tartozó[1] személy készítette az aláírást, és ő pontosan ezt a dokumentumot írta alá. Ha valaki ezzel ellentéteset állít, neki kell bizonyítania az állítását (ahogy ez például a közjegyző előtt vagy a két tanú előtt aláírt teljes bizonyító erejű magánokiratokra is igaz).

> *PP. „196. § (1) A magánokirat az ellenkező bebizonyításáig teljes bizonyítékul szolgál arra, hogy kiállítója az abban foglalt nyilatkozatot megtette, illetőleg elfogadta, vagy magára kötelezőnek ismerte el, feltéve, hogy az alábbi feltételek valamelyike fennáll:*
> *a) a kiállító az okiratot sajátkezűleg írta és aláírta;*
> *b) két tanú az okiraton aláírásával igazolja, hogy a kiállító a nem általa írt okiratot előttük írta alá, vagy aláírását előttük sajátkezű aláírásának ismerte el; az okiraton a tanúk lakóhelyét (címét) is fel kell tüntetni;*
> *c) a kiállító aláírása vagy kézjegye az okiraton bíróilag vagy közjegyzőileg hitelesítve van;*
> *d) a gazdálkodó szervezet által üzleti körében kiállított okiratot szabályszerűen*

[1] Ha a tanúsítvány nem álneves, akkor az adott személy neve ott szerepel a tanúsítványban. Álneves tanúsítvány esetén az illető adatai a hitelesítés-szolgáltatótól szerezhetőek be az Eat. által meghatározott szabályok szerint.

aláírták;
e) ügyvéd (jogtanácsos) az általa készített okirat szabályszerű ellenjegyzésével bizonyítja, hogy a kiállító a nem általa írt okiratot előtte írta alá, vagy aláírását előtte saját kezű aláírásának ismerte el, illetőleg a kiállító minősített elektronikus aláírásával aláírt elektronikus okirat tartalma az ügyvéd által készített elektronikus okiratéval megegyezik;
f) az elektronikus okiraton kiállítója minősített elektronikus aláírást helyezett el. ”

A fokozott biztonságú elektronikus aláíráshoz nem kapcsolódik a fenti joghatás. Ha egy dokumentumon csak fokozott biztonságú (azaz nem minősített) aláírás szerepel, akkor – bár a dokumentum írásba foglaltnak minősül – nagyon keveset tudunk arról, hogy mennyire bízhatunk abban, hogy az aláírást valóban a tanúsítványban szereplő személy készítette. Ekkor egyáltalán nem biztos, hogy a hitelesítés-szolgáltató személyesen is találkozott a tanúsítvány alanyával, lehet, hogy a tanúsítványt postán vagy Interneten keresztül igényelték. Az sem biztos, hogy a magánkulcsot BALE védi, lehet, hogy a magánkulcs csak egy fájl az aláíró számítógépén, és lehet, hogy valaki az aláíró tudta nélkül lemásolta vagy használta azt. Az is lehet, hogy az aláíró már jelentette a kulcs kompromittálódását a hitelesítés-szolgáltatónak, de a hitelesítés-szolgáltató még nem tette közzé a visszavonási állapotot. De az is lehet, hogy a hitelesítés-szolgáltató hibázott, és valaki másnak bocsátotta ki a tanúsítványt, de kikötötte, hogy nem vállal anyagi felelősséget a tanúsítvánnyal kapcsolatban.

Léteznek „erős” fokozott biztonságú aláírások, ahol a tanúsítványt a szolgáltató személyes találkozás során bocsátotta ki, a magánkulcsot BALE (de legalábbis egy intelligens kártya) védi, amelyekhez gyors visszavonás kezelés és szolgáltatói felelősségvállalás tartozik, de ezek egyike sem következik önmagában abból, hogy egy aláírás fokozott biztonságú. Ha tudjuk, hogy minősített elektronikus aláírással állunk szemben, biztosak lehetünk benne, hogy az aláírás rendelkezik a felsorolt biztonsági tulajdonságokkal.

Általánosan elfogadott, hogy *minősített elektronikus aláírást kizárólag* ember, *természetes személy hozhat létre.* Ez nem derül ki egyértelműen az elektronikus aláírásról szóló törvényből, de a minősített aláírással ellátott dokumentum teljes bizonyító erejű magánokirat (illetve egyenértékű a kézzel írott aláírással), és ilyet csak természetes személy készíthet. Ugyanakkor az a „szokás”, hogy nem természetes személyek fokozott biztonságú aláírást hoznak létre. Ekkor az aláíró vagy egy automata[2] vagy például egy jogi személy. Szintén nem egyértelmű az Eat. alapján, hogy nem természetes személy készíthet-e elektronikus aláírást, de az elterjedt gyakorlat szerint igen. Ilyenkor az aláírás egy szervezet nevében készül, és például egy bitsorozat átvételét, vagy egy számla hitelességét igazolja.

A minősített elektronikus aláírás nem feltétlenül biztonságosabb, mint a fokozott biztonságú. Előfordulhat, hogy a fokozott biztonságú aláírás készítéséhez használt magánkulcsot egy

[2]Ez nem keverendő azzal az esettel, amikor az aláíró természetes személy, de egy automata kezeli a magánkulcsát.

HSM (kriptográfiai célhardver) védi, lehet, hogy e magánkulcs hosszabb és erősebb, mint a BALE kártyák által támogatott kulcsok, és e HSM-et lehet, hogy egy biztonságos szerverteremben működtetik, ahova csak megbízható rendszergazdák léphetnek be, alapos ellenőrzés után. Ugyanakkor előfordulhat, hogy a minősített aláírás készítéséhez használt BALE kártyát az aláíró rendszeresen egyedül hagyja az asztalán, és a hozzá tartozó PIN kódot (ami „123456") filctollal ráírta a kártyára, hogy el ne felejtse. Előfordulhat, hogy egy fokozott biztonságú aláíráshoz erősebb kulcs, és jobb kulcsmenedzsment kapcsolódik, mint egy minősített aláíráshoz. A minősített és a fokozott biztonságú aláírás között az jelenti a különbséget, hogy:

- a minősített aláírásra több szabály vonatkozik, így aki minősített aláírással találkozik, az jobban tudhatja, hogy milyen biztonsági szintre számíthat (természetes személy aláíró, személyes regisztráció, és BALE-n tárolt magánkulcs); és
- a minősített aláíráshoz erősebb bizonyító erő tartozik.

Az elektronikus aláírásról szóló EU direktíva szerint a minősített elektronikus aláírást[3] a kézzel írott aláírással egyenértékűnek ismerik el a tagállamok, és ami valamely EU tagállam szabályai értelmében minősített aláírás, azt a többi EU tagállamban is minősített aláírásnak kell tekinteni. [62] Így a direktíva értelmében a minősített elektronikus aláírásokat az egész Európai Unióban a kézzel írott aláírással egyenértékűnek kell tekintetni, függetlenül attól, hogy azok mely tagállamban készültek. Ugyanakkor az egyes tagállamokban markánsan eltérő szabályok vonatkoznak a minősített aláírásokra, így a gyakorlatban ez ennél jelentősen összetettebb.

Az egyes tagállamokban különböző követelmények vonatkoznak a minősített hitelesítés-szolgáltatókra és a BALE eszközökre, illetve egyes tagállamokban további követelmények is vonatkoznak a minősített elektronikus aláírásra. (Például van olyan tagállam, ahol minősített aláírást minősített aláírás-létrehozó alkalmazással kell készíteni.)

6.1.3. Mit jelent az, hogy „letagadhatatlan"?

Az elektronikus aláírásról szóló 2001. évi XXXV. törvény szerint a fokozott biztonságú elektronikus aláírással ellátott dokumentum írásba foglaltnak minősül, a minősített elektronikus aláírással hitelesített dokumentum pedig a polgári perrendtartásról szóló törvény szerint teljes bizonyító erejű magánokiratnak minősül. Így az elektronikus aláírásról szóló jogszabályok a letagadhatatlanság néven ismert műszaki fogalmat a bizonyító erő nevű jogi fogalomnak feleltetik meg.

[3] Az EU direktívában nem szerepel a „minősített elektronikus aláírás" kifejezés, a direktíva „olyan fokozott biztonságú elektronikus aláírásról beszél, amely minősített tanúsítványra épül, és amely biztonságos aláírás-létrehozó eszközzel készült". E fogalmat nevezzük röviden minősített elektronikus aláírásnak.

Bármekkora bizonyító erővel is rendelkezik egy elektronikus aláírás, egy bíróság – indokolt esetben – megkérdőjelezheti annak érvényességét. (Például ha bizonyítható, hogy az aláírást nem az aláíró személy, hanem például az ő számítógépét irányító vírus hozta létre.) A letagadhatatlanság műszaki fogalom, azt jelenti, hogy gyakorlatilag kizárható, hogy egy adott tanúsítvány alapján elfogadható aláírást nem a tanúsítványhoz tartozó magánkulccsal hoztak létre. Jogi értelemben letagadhatatlanságról nem, hanem csak bizonyító erőről beszélhetünk.

Egy elektronikus aláírás kizárólag akkor rendelkezhet – jogi szempontból – bizonyító erővel, ha „érvényes", tehát letagadhatatlansága műszaki szempontból bizonyítható. Ha az aláírás műszakilag érvénytelen, abból semmi sem következik.

Ha az aláírás műszakilag érvényes, kapcsolódhat hozzá jogkövetkezmény. A törvény szerint, ha egy dokumentumon érvényes minősített elektronikus aláírás szerepel, vélelmezni kell, hogy az aláíráshoz tartozó tanúsítványban szerepelő személy írta alá a dokumentumot, és a dokumentum nem változott meg az aláírást követően. E vélelem azt jelenti, hogy annak kell bizonyítania az igazát, aki ezzel ellentéteset állít. Ugyanakkor lehet helye ellenkező bizonyításnak, előfordulhat, hogy valaki bebizonyítja, hogy mégsem az aláíró készítette az aláírást, vagy valaki mégis megváltoztathatta azt. Például ellophatták az aláíró kártyáját, vagy egy vírus is megfertőzhette a számítógépét, és ekkor lehet, hogy tényleg nem ő készítette az aláírást. Ha érvényes minősített aláírás esetén hivatkozunk ilyenre, nekünk kell bizonyítanunk, hogy valóban ez történt. (Fokozott biztonságú aláírás esetén nincsen ilyen szabály, de ennek fordítottjára sincs. Egy fokozott biztonságú elektronikus aláírás esetén a bíróság szabadon mérlegelhet, hogy az aláíró készítette-e az aláírt dokumentumot, és az nem változott-e meg az aláírást követően.)

Jogi szempontból nem letagadhatatlan az elektronikus aláírás, még a minősített elektronikus aláírás sem az. Bizonyító erő, illetve a fenti vélelem kapcsolódik hozzá.

6.2. Elektronikus aláírás jogszabályok külföldön

Az „elektronikus aláírás" kifejezést az 1999/93/EC direktíva készítői teremtették. [62] A műszaki szakemberek korábban a „digitális aláírás" kifejezést használták, ez jellemzően a magánkulccsal kódolt dokumentumot, illetve kriptográfiai lenyomatot jelenti. [156] Ezzel szemben, az elektronikus aláíráshoz jogkövetkezmény kapcsolódik, de az elektronikus aláírás nem feltétlenül épül kriptográfiai alapokra. [1]

Ez EU tagállamok – Magyarországhoz hasonlóan – az elektronikus aláírásról szóló EU direktívát emelték át saját jogrendjükbe. A direktíva alapvető célja, hogy az egyik tagállamban készített elektronikus aláírásokat más tagállamban is el lehessen fogadni érvényes aláírásként. A direktíva két fontos állítást tesz az elektronikus aláírások joghatásával kapcsolatban:

1. A minősített elektronikus aláírások[4] a kézzel írott aláírásokkal egyenértékűnek tekinthetőek, attól függetlenül, hogy mely tagállam szabályai szerint készültek.

2. Az elektronikus aláírás bizonyítékként való felhasználását nem lehet önmagában azért megtagadni, mert az aláírás elektronikus formában létezik, illtve mert az aláírás nem minősített[5].

Ennek megfelelően más EU tagállamokban is hasonló szabályozás van érvényben, mint Magyarországon, bár vannak különbségek. A magyar Eat. például szól időbélyegzés-szolgáltatásról és archiválás-szolgáltatásról is, míg ezek a fogalmak nem szerepelnek a direktívában. Ugyanakkor a német és az olasz elektronikus aláírás törvényben szintén szerepel időbélyegzés-szolgáltatás (7. fejezet), és létezik minősített időbélyegzés-szolgáltatás is, míg sok más tagállam szabályozása nem különböztet meg minősített időbélyegzés-szolgáltatást. A német elektronikus aláírás törvényben esik szó minősített attribútum-tanúsítványokról, míg a magyar szabályozás nem ír attribútum-tanúsítványokról, nálunk az attribútum-tanúsítvány csupán egy műszak fogalom. A német elektronikus aláírás törvény egy másik érdekessége, hogy következetesen végigviszi azt az elvet, hogy az aláírás érvényességét egy múltbéli időpontra nézve kell vizsgálni. [169] Ennek megfelelően Németországban az időbélyegzés és a „mindig friss OCSP" szolgáltatás (4.1.5.2. fejezet) alapján történő aláírás ellenőrzés bevett gyakorlatnak számít.

Olaszországban jelentősen egyszerűbb módon használják a PKI-t, mint sok más EU tagállamban. Például nem terjedtek el a XAdES aláírások, helyette csak időbélyeggel ellátott PKCS#7 aláírásokat használnak, de úgy alakították ki a szolgáltatók szabályozását, hogy ez ne jelentsen problémát. Így az olasz szolgáltatóknak tilos eltávolítaniuk a lejárt tanúsítványokat a CRL-ről, és a szolgáltató megszűnése esetén az állam gondoskodik a CRL-ek elérhetőségéről. Így az aláíró tanúsítványának lejárta után is megbízhatóan ellenőrizhető az aláírás, nem szükséges csatolni hozzá a visszavonási információkat.

A szlovák szabályozás is nagyon hasonlít a hazaihoz, bár ott központi gyökér hitelesítés-szolgáltató működik, illetve részletes állami szabályozás van az elektronikus aláírások használatára.

Olyan országok is vannak, ahol gyakorlatilag nem használnak minősített aláírást, ilyen például Franciaország, illetve az Egyesült Királyság.

Országonként jelentősen eltérhet a tanúsítványok kibocsátásának módja. Magyarországon például nem terjedtek el a minősített tanúsítványra épülő, de nem biztonságos aláírás-létrehozó eszközzel készült (így nem minősített) aláírások, míg más országban ilyen aláírásokat

[4] A direktíva szavaival: minősített tanúsítványra épülő, biztonságos aláírás-létrehozó eszközzel készült fokozott biztonságú aláírások.

[5] Azaz azért, mert az nem minősített tanúsítványra alapul, illetve nem biztonságos aláírás-létrehozó eszközzel hozták létre.

is használnak. Az is nagyon eltérő, hogy az egyes országokban mi minősül biztonságos aláírás-létrehozó eszköznek. Vannak országok, ahol egyértelmű nyilvántartás van ezen eszközökről (Magyarországon mellett ilyen például Németországban és Szlovákia), míg más országok esetén sokszor nehéz kideríteni, hogy ott mely eszköz BALE.

A nem EU-s országok közül az Amerikai Egyesült Államokat emelnénk ki. [42] Az USA szabályozása természetesen nem az EU direktívát követi, egészen más módon kezeli az elektronikus aláírások kérdését. Amerikában 1869-ben a New Hampshire Supreme Court kimondta[6], hogy nincs különbség aközött, hogy a távírász hagyományos tollal írja le az üzenetét vagy egy több ezer mérföld hosszú rézdrót segítségével, e rézdrótot használva tollként, mert az üzenet mindkét esetben ugyanazokat a gondolatokat fejezi ki. [18]

Míg a magyar Eat-ben mindössze egyetlen mondat szól arról, hogy „elektronikus aláírás, illetve dokumentum elfogadását [...] megtagadni [..] nem lehet kizárólag amiatt, hogy az aláírás, illetve dokumentum elektronikus formában létezik", addig az amerikai elektronikus aláírás törvény ezen egyetlen egy gondolat körül forog. Az amerikai jogszabály nem ír kriptográfiáról, csupán arról szól, hogy elektronikusan is lehet jognyilatkozatot tenni, szerződést kötni stb., és ennek (nem műszaki) feltételeit írja körül. Természetesen az USA is használ kriptográfiai alapon készült aláírásokat, mert ezek esetén nagyobb a bizonyossága, hogy az aláírt dokumentum nem változott meg időközben. Különbség, hogy az amerikai törvény jelentősen nagyobb mérlegelési jogot ad a bírónak, ott nem szerepel olyan, hogy bizonyos esetben vélelmezni kellene, hogy az aláírást a megadott személy készítette, míg az EU sokkal mélyebben szabályozza e területet.

6.3. Aláírás készítése

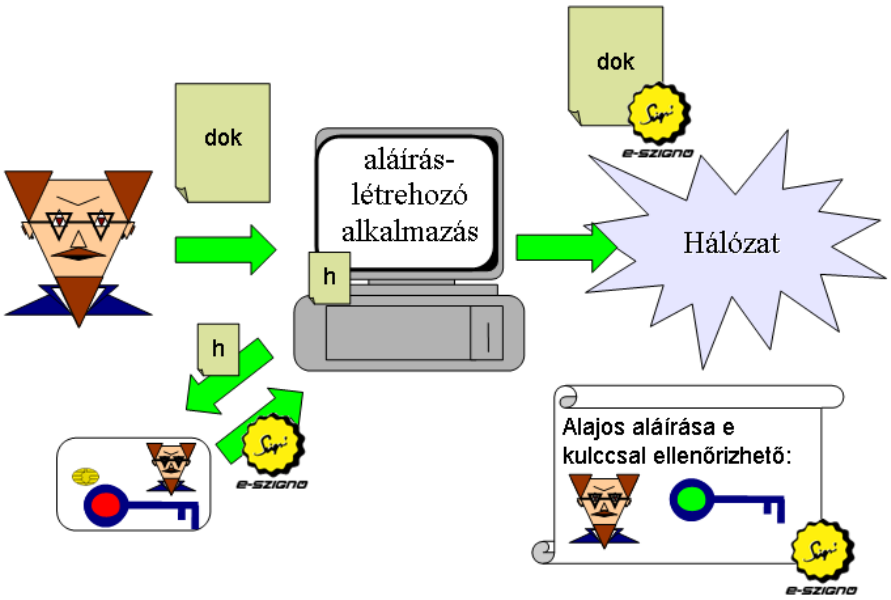
Elektronikus aláírás készítésekor a következő lépések zajlanak le (lásd: 6.4. ábra):

1. Az aláíró megtekint egy dokumentumot, majd úgy dönt, aláírja. Átadja vagy elküldi a dokumentumot a számítógépén futó aláírás-létrehozó alkalmazásának.

2. Az aláírás-létrehozó alkalmazás kriptográfiai lenyomatot, más néven hash-t (2.4. fejezet) képez a dokumentumból, és a lenyomatot átadja az aláírás-létrehozó eszköznek.

3. Az aláírás-létrehozó eszköz – amely lehet intelligens kártya vagy HSM, de lehet maga az aláírás-létrehozó alkalmazás is – az aláíró magánkulcsa segítségével kódolja (2.3. fejezet) a kapott lenyomatot, e kódolás eredménye a kriptográfiai értelemben vett aláírás. Az aláírás-létrehozó eszköz visszaküldi az aláírást az aláírás-létrehozó alkalmazásnak.

[6] [Howley v. Whipple, 48 N.H. 487]



6.4. ábra. **Aláírás készítése**

4. Az aláírás-létrehozó alkalmazás további információkkal egészítheti ki az aláírást, majd csatolja az aláírt dokumentumhoz.

Az aláírás és az aláírt dokumentum együttesen használható fel (pl. el lehet küldeni e-mailben).

6.3.1. Lenyomat aláírása (hash&sign)

Aláíráskor általában nem a teljes dokumentumot kódoljuk a magánkulcsunkkal, hanem annak egy lenyomatát. Ennek egyik oka, hogy a nyilvános kulcsú kriptográfiai műveletek lassúak, és időigényes volna a teljes dokumentumot (amely esetleg többször tíz megabyte is lehet) elküldeni az intelligens kártyának, és kódolni. Másik oka, hogy a magánkulccsal általában egyszerre csak meghatározott méretű blokkokat – pl. 1024 bites RSA esetén 1024 bit hosszú blokkokat – lehet kódolni. Rossz megoldás lenne a dokumentumot egyszerűen blokkokra bontani, és a blokkokat külön-külön aláírni, mert ha a blokkokon lévő aláírások egymástól függetlenek, akkor a támadó észrevétlenül felcserélhetné az aláírt dokumentum blokkjait, vagy észrevétlenül kitörölhetne egyes blokkokat. Így mindenképpen össze kellene fűzni a blokkokat valamilyen módon.

Ha a dokumentumból biztonságos, azaz ütközés-ellenálló és őskép-ellenálló (2.4. fejezet) hash függvénnyel képzünk lenyomatot, és az így kapott lenyomatot írjuk alá, az olyan, mintha a dokumentumot egyetlen blokként kódoltuk volna. A lenyomat a dokumentum minden egyes

bitjétől függ, és reális erőforrásokkal nem lehet másik olyan dokumentumot konstruálni, amelyhez ugyanez a lenyomat tartozik.

Ha az alkalmazott hash algoritmus nem felel meg a fenti kritériumoknak, az aláírás megkérdőjelezhető. Ha a lenyomatképző algoritmusra nézve nem teljesül a második őskép-ellenállóság (2.4. fejezet), akkor előfordulhat, hogy valaki talál egy másik bitsorozatot, amelyhez ugyanez az aláírás tartozik. Ha a lenyomatképző algoritmusra nézve nem teljesül az ütközés-ellenállóság, előfordulhat, hogy egy csaló aláíró két olyan dokumentumot készít, amelyekhez ugyanaz az aláírás tartozik.

6.3. Példa: *Alajos kölcsön kér Bendegúztól 1000 forintot, biztosítékul aláír egy váltót az összegről. Bendegúz rájön, hogy az aláíráskor használt lenyomatképző algoritmus nem őskép-ellenálló, és konstruál egy másik váltót, amely 10 millió forintról szól, de azonos a lenyomata, így erre is illik Alajos aláírása. Bendegúz ráteszi Alajos aláírását a 10 millió forintos váltóra, és e nagyobb összeget követeli Alajostól. Ha Alajosnál nincs meg az igazi váltó, amit aláírt, nehéz lesz bizonyítania, hogy nem a 10 milliós váltót írta alá.*

6.4. Példa: *Alajos kölcsön kér Bendegúztól 10 millió forintot, biztosítékul aláír egy váltót az összegről. Alajos nem adja meg a tartozását, ezért Bendegúz bírósághoz fordul. Manfréd, Alajos ügyvédje rájön, hogy az aláíráshoz használt algoritmus nem őskép-ellenálló. Konstruál egy másik dokumentumot, és azzal védekezik a bíróság előtt, hogy Alajos soha nem írta alá a 10 millió forintról szóló váltót, hanem e másik dokumentum írta alá. Előfordulhat, hogy e másik dokumentum értelmetlen, pl. így néz ki: „Z+3FG65!0X-/=TG...". Ekkor a bíró nem biztos, hogy elhiszi, hogy Alajos valóban ezt a butaságot*[7] *írta alá. Ha másik, de azonos lenyomatú bitsorozatot lehet is találni, másik értelmes és hihető ősképet találni ennél sokkalta nehezebb feladat. Ha Manfréd sikerrel jár, nem lehet majd eldönteni, hogy Alajos pontosan mit írt alá.*

Nagyon nehéz feladat a hash függvény második őskép-ellenállóságát támadni, azaz egy ismert ősképhez és lenyomathoz egy másik ősképet találni, amelyhez ugyanaz a lenyomat tartozik (különösen, ha értelmes és hihető ősképet keresünk). Ennél jellemzően sokkal-sokkal könnyebb dolga van egy csaló aláírónak, aki a hash függvény ütközés-ellenállóságát támadja, azaz két olyan ősképet keres, amelyekhez azonos lenyomat tartozik. [21] Igaz, ha a hash függvény erős, a gyakorlatban egyik támadást sem lehet végrehajtani.

6.5. Példa: *Manfréd kölcsön kér Bendegúztól 10 millió forintot, biztosítékul aláír egy váltót az összegről. Manfréd soha nem akarja visszaadni Bendegúznak a*

[7] Itt egészen más a helyzet, ha az eredeti dokumentum is strukturálatlan, és pl. egy véletlen számot tartalmaz.

pénzt, ezért különleges váltót készít. Keres két olyan dokumentumot, amelyekhez azonos lenyomat tartozik; az egyik dokumentum egy 10 millió forintról szóló váltó (sok ilyen bitsorozat létezik, Manfréd a szövegezés vagy a formázás változtatásával nagyon sok variációt előállíthat), a másik pedig egy ártatlan dokumentum, például egy karácsonyi üdvözlőlap (itt is sok szóba jöhető bitsorozat létezik). Manfréd először megkeresi e két dokumentumot, és csak ezt követően készíti el az aláírást. Így az aláírása mindét dokumentumhoz jó. Ha Bendegúz bírósághoz fordul, hogy visszakapja a pénzét, Manfréd azt állítja, hogy ő soha nem írt alá váltót Bendegúznak, ő csak ezt a karácsonyi üdvözlőlapot küldte neki, a váltót pedig Bendegúz konstruálta.

A fenti példákban azt mutattuk meg, hogy különösen fontos biztonságos – ütközés-ellenálló és őskép-ellenálló – hash függvényt használni. Ha biztonságos hash függvényt használunk, akkor a fenti támadások nem valósíthatóak meg reális erőforrások mellett.

6.3.2. Padding

A nyilvános kulcsú kriptográfiai algoritmusokkal általában csak meghatározott hosszúságú blokkokat lehet kódolni. Például RSA esetén ha 1024 bites kulcsot használunk, akkor csak 1024 bites blokkokat kódolhatunk, ha 2048 bites kulcsot használunk, akkor csak 2048 bites blokkokat kódolhatunk stb. A lenyomat – amelynek mérete a használt lenyomatképző algoritmustól függ – általában rövidebb, például: SHA-1 esetén 160 bit, SHA-256 esetén 256 bit, SHA-512 esetén 512 bit.

Ezért aláírás előtt a lenyomatot kiegészítjük a szükséges blokkméretre. E kiegészítés (padding) általában a PKCS#1 specifikáció 1.5-ös változata szerint történik. [157] A padding egy jelzést tartalmaz, miszerint paddingről van szó, tartalmazza a lenyomatképző algoritmus megnevezését, majd konstans (`FF`) byte-okkal tölti fel a maradék helyet[8]. Így a magánkulccsal nem közvetlenül az aláírandó dokumentum lenyomatát, hanem a paddinggel kiegészített lenyomatot kódoljuk.

6.3.3. Aláírás-létrehozó eszköz

Az aláírást a magánkulcs és az aláírandó dokumentum lenyomata alapján, nyilvános kulcsú kriptográfiai algoritmus (2.3. fejezet) segítségével lehet kiszámítani. Ezen algoritmusok bonyolultak, a számítást csak valamilyen számítástechnikai eszköz segítségével lehet reálisan elvégezni, így az aláírás készítéséhez szükség van valamilyen számítástechnikai eszközre. Ezen eszközt nevezzük aláírás-létrehozó eszköznek.

[8] A PKCS#1 újabb, 2.1-es változata már randomizált, de speciális struktúrával rendelkező paddinget ír le, amely bizonyos támadásokkal szemben jobb tulajdonságokat mutat, mint a PKCS#1 1.5-ös változata szerinti konstans padding. A randomizált megoldás (PSS, probabilistic signature scheme) még nem terjedt el, az XMLDSIG specifikáció csak a PKCS#1 1.5-ös változata szerinti konstans paddinget használja. [157]

6.3.3.1. Szoftveres kulcs esete

Egyik lehetőség az ún. szoftveres kulcs esete, amikor ezen eszköz egy általános célú számítógép, és például ugyanaz a számítógép készíti el az aláírást, mint amelyiken az aláírandó dokumentumot megírjuk. Ekkor nincs szükség speciális célhardverre, a magánkulcsunk ott van a számítógépen, vagy egy fájlban, vagy például a Windows registry-jében. E megoldás sok előnnyel rendelkezik:

- A magánkulcsot könnyen, korlátozások nélkül tudjuk használni. A szoftveres kulcsokat lényegében minden alkalmazás támogatja. Beállíthatjuk, hogy ne kelljen megadnunk a PIN kódunkat minden aláíráskor, és nagy mennyiségű dokumentumot (pl. sok számlát) egyetlen kattintással vagy egyetlen program elindításával aláírhatunk.
- A magánkulcsról biztonsági másolatot készíthetünk, így ha a számítógép tönkremegy, egy másik gépre visszatölthetjük a magánkulcsot, és tovább használhatjuk.
- A magánkulcsot akár több számítógépre is felmásolhatjuk, és több gépen dolgozhatunk vele. (Így nem muszáj mindig odavinni az adott gépre a magánkulcsunkat, elérhetjük, hogy az minden gépen ott van.)

E megoldásnak hátrányai is vannak. A hátrányok éppen ugyanazok, mint az előnyök, ugyanis előfordulhat, hogy a felsorolt lehetőségeket nem mi, hanem egy ellenünk áskálódó támadó használja ki:

- A magánkulcsot könnyen, korlátozások nélkül lehet használni. Esetleg nemcsak mi férünk hozzá a magánkulcshoz, hanem a támadó is, így akár a mi tudtunk nélkül is készíthet aláírást.
- A magánkulcsot a támadó is lemásolhatja, és elviheti a másolatot a számítógépünkről (és ezt esetleg észre sem vesszük).
- Nagyon nehezen tudjuk kontrollálni, hogy mikor, hány gépen és hol van a magánkulcsunk. Minél több gépen van a kulcsunk, annál több helyről szerezheti meg a támadó.

A szoftveres kulcsokat is védhetjük. Egyik megoldás, hogy titkosítva tároljuk a számítógépen, például úgy, hogy csak egy jelszó segítségével lehet visszafejteni. (Például ha PFX – más néven PKCS#12 – fájlban tároljuk a kulcsot, megadhatjuk, hogy egy jelszóval[9] titkosított módon szerepeljen a kulcs a PFX fájlban.) Ha elég erős jelszót választunk, ezzel jelentős védelmet

[9]Ilyenkor a jelszóból egy szimmetrikus kulcsot generálunk, és e szimmetrikus kulccsal titkosítjuk a magánkulcsunkat. Amikor legközelebb megadjuk a jelszót, a számítógép ugyanazon determinisztikus algoritmussal generálja belőle a szimmetrikus kulcsot, és e szimmetrikus kulccsal állítja vissza a magánkulcsot.

biztosíthatunk a kulcsnak. Igaz, minden egyes alkalommal, amikor a kulcsot használjuk, visszafejtjük a kulcsot, és az nyíltan jelen van a számítógépen.

Ha a Windows tanúsítványtárában tároljuk a magánkulcsunkat, ott is megadhatjuk, hogy csak jelszóval lehessen hozzáférni, de ezen túl ott olyat is megadhatunk, hogy magát a kulcsot ne lehessen kimenteni a Windows tanúsítványtárából. Nehéz megítélni, hogy ez pontosan mekkora védelmet jelent.

6.3.3.2. Intelligens kártya

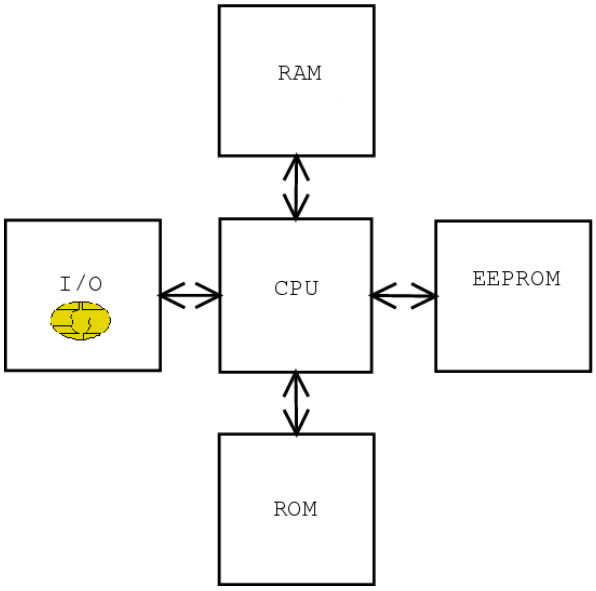
Intelligens kártya (más néven chipkártya vagy smart card) alatt olyan műanyag kártyát értünk, amelyen mikrochipet helyeztek el. A kártyán információkat elsősorban e chip segítségével tárolhatunk, de a chip mellett akár a kártya felületén is feltüntethetünk információkat. (Lásd: 6.2. ábra.)

A legegyszerűbbek az ún. *memóriakártyák*, esetükben a kártyán lévő IC lényegében egy memóriachip. A kártyát meghajtó terminál közvetlenül hozzáfér a chip tartalmához: olvashatja, írhatja. Ezek a kártyák a logikai biztonság szempontjából nem jelentenek komoly áttörést a mágneskártyákhoz képest, az általuk nyújtott logikai biztonság egy floppy diskével ekvivalens.

A továbbiakban olyan kártyákról lesz szó, amelyek már valamilyen szintű védelmet is nyújtanak a rajtuk lévő adatoknak. Előfordulhat, hogy bizonyos adatokat csak PIN kód megadásával, vagy kriptográfiai autentikációt követően lehet írni vagy olvasni, illetve biztonságos, kriptográfiailag védett csatornát lehet kiépíteni a kártyával. Ennek megfelelően a továbbiakban az intelligens kártya elnevezést használjuk.

6.3.3.2.1. „Generikus” kártyák

A *generikus kártyák* olyan intelligens kártyák, amelyeken az adatok fájlokba, azok pedig fájlrendszerbe szerveződnek. A kártyán definiálhatunk felhasználókat és azok jogosultságait. Megadhatjuk, mely felhasználóknak milyen módszerrel szükséges azonosítani magukat – PIN kód, challenge and response autentikáció, – illetve mely fájlokhoz, könyvtárakhoz milyen módon (írás, olvasás, inkrementálás, stb.) férhetnek hozzá. A legújabb generikus kártyák szimmetrikus és nyilvános kulcsú kriptográfiai módszerekkel is védhetik az adataikat. Ezen kártyák nevében a „generikus” szó arra utal, hogy a kártyagyártó általános célú eszközt gyárt, amelyet testre lehet szabni az egyes alkalmazásokhoz (pl. bankkártya, hűség- avagy loyalty kártya, aláíró kártya stb). Nem szükséges tehát minden célra külön kártyát gyártani, a termelés legutolsó fázisa, a perszonalizálás (a felhasználók, a fájlrendszer, a kulcsok és jogosultságok beállítása) a kártyakibocsátóhoz kerülhet, míg a gyártó általános célú kártyákat hoz nagy mennyiségben létre, így sokkal olcsóbban állítja őket elő. [195], [139]



6.5. ábra. **Egy intelligens kártya belsejének blokkvázlata**

6.3.3.2.2. A kártya belseje

Az összetettebb generikus kártyák valójában *biztonságos mikroszámítógépek*. Rendelkeznek processzorral, memóriával, háttértárral és I/O perifériával is. (Lásd: 6.5. ábra.)

E kártyák fő erőssége a biztonság. A fő különbség a Neumann-féle architektúrájú számítógépek és a chipkártyák között, hogy az utóbbi esetben az I/O perifériák nem kapcsolódnak közvetlenül a belső buszokra. [33] Chipkártyák esetében a külvilág (I/O periféria) és az adattároló egység (EEPROM) között egy döntő logika helyezkedik el, amely a bemenetet megszűrheti, felülbírálhatja. Az ábrán látható egységek egyetlen chipben helyezkednek el, nincsenek közöttük buszok, amelyeket egy külső támadó esetleg lehallgathat. A chip pedig olyan mikroelektronikai technológiák [139] segítségével készült, hogy minél nehezebb legyen belőle információkat a kártya felnyitása esetén kinyerni.

A kártyát kívülről egynek és oszthatatlannak szeretnénk tekinteni, amelyből információkat csakis a kontaktusokon keresztül nyerhetünk ki. A kontaktusok jelentik a kártya egyetlen kapcsolatát a külvilággal. Ez fontos, hiszen a chipkártyák biztonságának egyik legjelentősebb pillére az, hogy csakis egy jól definiált és jól ellenőrzött interfésszel rendelkeznek a külvilág felé.

A fenti technológia ugyanakkor korlátokat is támaszt. Azzal, hogy minden alkatrésznek (lásd: 6.5. ábra) egyetlen chipbe kell kerülnie, a chip bonyolulttá válik, valamint igen komoly hőelvezetési problémák jelennek meg. Ez erősen bekorlátozhatja a kártya órajelét, számítási és tárolókapacitását. [13]

A kártyák a számítógépekhez képest szerény képességekkel rendelkeznek. Processzoruk az

asztali számítógépekéhez képest igen lassú, kriptográfiai műveleteiket viszont speciális kriptokoprocesszorok gyorsítják fel. Mindössze néhány kilobyte RAM-mal és néhány tíz kilobyte EEPROM-mal rendelkeznek.

6.3.3.2.3. Programozható kártyák

A generikus kártyákat kibocsátást követően általában lezárják, és e pillanattól már nem tölthető le rájuk program. Van, ahol ez nem feltétlenül így történik, az ilyen kártyákat *programozható kártyáknak* is nevezik. Esetükben – akár a kártya egész élete során – letölthetünk rá újabb alkalmazásokat, s ezeket rajtuk futtathatjuk. Egyes kártya-típusok esetén kifejezetten ösztönzik, hogy a kártyára harmadik felek is fejlesszenek alkalmazásokat, így sok esetben nyilvános, esetleg konkrét kártyatípusoktól független nyelvek, specifikációk szerint lehet kártyákon futó alkalmazást készíteni. Például a Java Card specifikációnak megfelelő kártyákra Java nyelven – illetve a Java nyelv egy részhalmazán – írt alkalmazások tölthetőek fel. [175]

6.3.3.2.4. Kártyaolvasó

A chipkártya input-output műveleteket csakis a mikrochip kontaktusain végezhet. A kevés input periféria miatt viszonylag könnyű a lehetséges bemeneteket számbavenni, így a kártyákat (egy asztali számítógéphez képest) viszonylag könnyű biztonsági szempontból bevizsgálni. Ugyanakkor a kártya így egy olyan „fogyatékos” számítógéppé válik, amely nem rendelkezik saját felhasználói felülettel, így nem képes önállóan kommunikálni még a saját felhasználójával sem. A kártya egy kártyaolvasó készülék segítségével tartja a kapcsolatot a külvilággal, így a kártya bemenetét a kártyaolvasót kezelő számítógép vagy egyéb terminál határozza meg.

Akármilyen alakú és tudású is a terminál, roppant fontos szerep jut neki mind funkcionális, mind biztonsági szempontból. Csakis ő biztosíthatja ugyanis a kártya számára:

- a felhasználói felületet,
- a hálózati kapcsolatot,
- a tápfeszültséget (!) és
- az órát.

A kártya egy úgynevezett kártyaolvasón keresztül kapcsolódik a terminálhoz. Ezen roppant megtévesztő elnevezésnek történelmi okai vannak, hiszen a mágneskártyák, illetve

memóriakártyák esetében a terminálhoz illesztő egység valóban olvasta a kártyát. Intelligens chipkártya esetében erről szó sincs.

A chipkártya-olvasó ugyanis a kártyákat sem olvasni, sem írni nem tudja. A terminál az olvasón keresztül mindössze „kérést" küldhet a kártyának, amely azt „megfontolja", s vagy elfogadja/megválaszolja, vagy pedig visszautasítja. Az írás-olvasás modell pusztán memóriakártyák esetében igaz, de generikus kártyák esetében is alkalmazható. Igaz, ez esetben a kártya ezen műveleteket meg is tagadhatja, és az íráson és olvasáson kívül számos más lehetőség is szóba jöhet (inkrementálás, következő rekord kiválasztása, stb). A terminál és a kártya közti kommunikációt az ISO7816-4 szabvány [88] írja le, a kártyával való kommunikáció alapegysége az APDU (Application Protocol Data Unit).

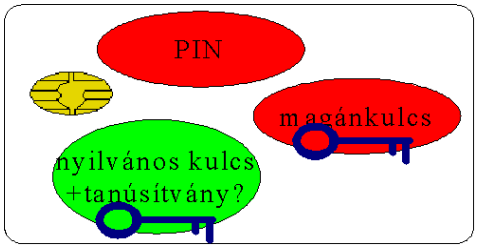
6.3.3.2.5. Adat- és kulcs-kártyák

Intelligens kártyás rendszerek esetén többféle megközelítést szokás alkalmazni. Az egyik megközelítés szerint a kártyák tartalmazzák a felhasználó lényeges adatait (pl. egy digitális pénztárca esetén a felhasználó egyenlegét), és a kártyák hozzáférés-védelmi mechanizmusai védik azokat. Másik lehetőség, hogy a kártyán csak kulcsok vannak, és a felhasználó a kártyán lévő kulcsok segítségével fér hozzá a hálózaton lévő adataihoz. Az első megoldásnak előnye, hogy a felhasználó hálózati kapcsolat nélkül is eléri az adatait, viszont hátránya, hogy rugalmatlan, nagyon nehéz elérni és pl. átstrukturálni a felhasználók kártyáin kint lévő, jellemzően offline adatokat. Az első kártya kibocsátása előtt fel kell tudni mérni, hogy a későbbiekben milyen adatok lesznek a kártyán, kik és milyen jogosultságokkal fognak hozzáférni. A második megoldás rugalmas, bármikor megváltoztatható, bármikor kiterjeszthető új alkalmazással, és könnyen megváltoztathatóak az egyes szereplők jogosultságai, ehhez semmit nem kell változtatni a felhasználók kártyáin.

6.3.3.2.6. PKI kártyák

PKI alkalmazásokban általában olyan generikus kártyákról beszélünk, amelyek kriptográfiai kulcsok, pontosabban a magánkulcsok védelmét látják le. A kulcspárok magánkulcsai csak a kártyán vannak jelen – jellemzően maga a kártya generálja őket, – és sehogyan sem nyerhetőek ki a kártyából. A nyilvános kulcsok (és esetleg a hozzájuk tartozó tanúsítványok) bármikor kiolvashatóak. A magánkulcsokat PIN kódok védik. Lehet, hogy minden magánkulcsot egyazon kód véd (ezt szokás globál PIN-nek nevezni), de lehet, hogy egyes magánkulcsokat külön, dedikált PIN kódok védenek. A megfelelő PIN megadása esetén a kártya a bemenetet kódolja a magánkulccsal, és visszaadja az eredményt, amely lehet vagy aláírás, dekódolt szimmetrikus kulcs vagy válasz egy kihívásra. (Lásd: 6.6. ábra.)

Mivel a magánkulcs soha nem hagyja el a kártyát, és még a kártya szétbontásával sem nyerhető



6.6. ábra. **A kártya egy vagy több kulcspárt véd, a kulcsokat ő maga generálja. A kártyából csak a nyilvános kulcs és a tanúsítvány olvasható ki, a magánkulcs és a PIN nem, de helyes PIN megadása esetén a kártya kódolja a bemenetet a magánkulccsal.**

ki belőle, a kártya (és annak jelszava, PIN kódja) nélkül nem lehetséges elektronikus aláírást készíteni a felhasználó nevében. [12], [13]

6.3.3.3. Biztonságos aláírás-létrehozó eszköz

Az elektronikus aláírásról szóló törvény két követelményt fogalmaz meg a minősített elektronikus aláírásra: biztonságos aláírás-létrehozó eszközzel (BALE) készüljön, és alapuljon minősített tanúsítványra. A törvény szerint csak olyan eszközzel készíthető minősített elektronikus aláírás, „amely rendelkezik a Hatóság által nyilvántartásba vett, tanúsításra jogosult szervezetek által erre a célra kiadott igazolással", vagy más EU tagállam tanúsító szervezetének igazolásával. Eat. 7. § (5) és (6)

Az Eat. 1. melléklete követelményeket is meghatároz a biztonságos aláírás-létrehozó eszközökre:

> *Eat., 1. melléklet „1. A biztonságos aláírás-létrehozó eszközöknek megfelelő technikai és eljárási eszközökkel biztosítaniuk kell legalább a következőket:*
> *a) az aláírás készítéséhez használt aláírás-létrehozó adat aláírónként biztosan mindig különbözik, s titkossága kellően biztosított,*
> *b) az aktuálisan elérhető technológiával kellő bizonyossággal garantálható, hogy az aláírás készítéséhez használt aláírás-létrehozó adat nem rekonstruálható, megvalósítható annak a jogosulatlan felhasználókkal szembeni védelme, illetve az aláírás nem hamisítható.*
>
> *2. A biztonságos aláírás-létrehozó eszközöknek nem szabad az aláírandó elektronikus dokumentumot az aláírás elhelyezéséhez szükséges mértéken felül módosítaniuk, illetőleg nem akadályozhatják meg azt, hogy az aláíró a dokumentumot az aláírási eljárás előtt megjelenítse. "*

Az 1/a követelményt általában úgy szokás teljesíteni, hogy a magánkulcsot a BALE jó minőségű véletlenszám generátora generálja (ettől egyedi), és magát a kulcsot nem lehet

kinyerni a BALE belsejéből. Az 1/b követelmény teljesül, ha a BALE csak megfelelő kriptográfiai algoritmusokkal hajlandó használni a magánkulcsot. A 2. követelmény szerint a BALE csak az aláíráshoz szükséges mértékben módosíthatja az aláírandó adatot, így például kiegészítheti paddinggel, de nem cserélheti le egy másik dokumentum lenyomatára.

A legerősebb követelmény, hogy egy EU tagállamban nyilvántartásba vett tanúsító szervezetnek igazolnia kell, hogy az adott eszköz BALE, és alkalmas minősített elektronikus aláírás készítésére. E tanúsító szervezetek általában csak akkor állítanak ki ilyen igazolást, ha az adott eszköz rendelkezik valamely mértékadó nemzetközi tanúsítással.

Leggyakoribb a Common Criteria tanúsítás az SSCD (secure signature creation device) védelmi profil (protection profile, PP) szerint, legalább EAL4 szinten. [29], [11] Az SSCD PP BALE termékekre fogalmaz meg biztonsági követelményeket, és három különböző típusú BALE szintet különböztet meg: Az ún. 1. típusú BALE nem tud aláírni, csak kulcsgenerálást végez, és biztonságosan át tudja tölteni a kulcsot egy 2. típusú BALE-ba. A 2. típusú BALE csak aláírni tud a beletöltött kulccsal, de maga nem tud kulcsot generálni. A 3. típusú BALE aláírás-készítésre és kulcsgenerálásra egyaránt alkalmas. (Lásd: 3.3.2. fejezet.)

Más tanúsítás szerint is dönthet úgy egy tanúsító szervezet, hogy egy adott eszköz BALE, azaz megfelel az Eat. és az EU direktíva követelményeinek. Elfogadhat más követelmények szerinti Common Criteria tanúsítást is, de lehet, hogy a BALE ITSEC vagy FIPS 140-x szerinti tanúsítással rendelkezik.

A BALE-k általában intelligens kártyák (kivéve az SSCD PP szerinti 1-es típusú BALE-kat, amelyek jellemzően HSM-ek), de ez nem törvényszerű. USB kulcs is lehet BALE, sőt, elvileg akár HSM is lehetne BALE. (Megjegyezzük, a BALE mögött meghúzódó elgondolás szerint a BALE személyre szabott eszköz, amelyet az aláíró saját maga kontrollál. Elég nehezen képzelhető el, hogy egy HSM az aláíró kizárólagos kontrollja alatt legyen.).

6.3.3.4. Kriptográfiai hardver modul (HSM)

Egy kriptográfiai hardver modul (HSM, hardware security module) hasonló funkciókkal rendelkezik, mint egy intelligens kártya: magánkulcsot vagy magánkulcsokat véd, a kulcs nem nyerhető ki belőle, de ha megfelelően aktiválják, aláír a kulccsal. A különbség abban rejlik, hogy míg az intelligens kártya jellemzően személyre szabott eszköz, amelyet csak az aláíró használ, HSM-et nagy informatikai rendszerek használnak, nagy tömegű aláírás készítésére, vagy különösen biztonságos kulcsok védelmére. [64]

A HSM általában egy szerverteremben működik, általában a szervezet nevében használja a magánkulcsot, és a szervert, illetve a rajta futó aláírás-létrehozó alkalmazást felügyelő rendszergazdák kontrollja alatt van. Általában nem csak egyetlen rendszergazda kezelheti a kulcsot, mert ha ő pl. szabadságra megy, a szervezet esetleg nem tudna aláírni (és pl. leállna a számlázó rendszere).

A HSM-et általában nem aláírásonként kell aktiválni, hanem ha egyszer aktiváltuk, utána nagy mennyiségű aláírás készíthető vele. A HSM esetén az aktiválás gyakran nem PIN kód gépelését jelenti, hanem a rendszergazdák (vagy biztonsági tisztviselők) intelligens kártya segítségével azonosíthatják magukat a HSM felé. Előfordulhat, hogy egy kulcs csak akkor érhető el, ha több (pl. 7-ből 3) jogosult személy azonosította magát.

Az intelligens kártyáktól eltérően a HSM-ben tárolt kulcs általában kimenthető, hogy ne vesszen el a kulcs, ha a HSM tönkre megy. E mentéssel a magánkulcs csak titkosított formában nyerhető ki a HSM-ből, és e titkosított magánkulcs betölthető lehet egy másik (ugyanolyan típusú) HSM-be. Ilyenkor a magánkulcs egy szimmetrikus kulccsal titkosítva nyerhető ki a HSM-ből, és e szimmetrikus kulcsot egy titokmegosztási protokoll segítségével olyan módon osztjuk szét a HSM biztonsági tisztviselői között, hogy n tisztviselőből m vissza tudja állítani. Ha m tisztviselő beteszi a kártyáját az új HSM-be, a HSM visszaállítja a szimmetrikus kulcsot, és e szimmetrikus kulcs segítségével már vissza tudja fejteni a magánkulcsot is. Ilyen titokmegosztási megoldás például az ún. Shamir's secret sharing scheme. [166]

Miért jó HSM-et használni?

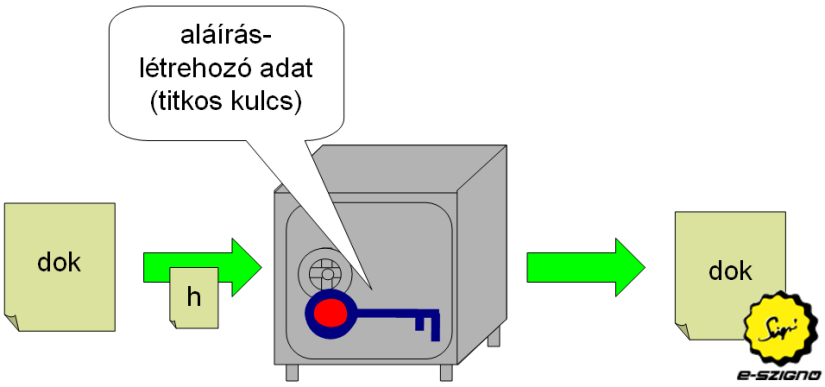
- Egy jó minőségű, bevizsgált, és megfelelően üzemeltetett HSM-ből rendkívül nehéz kinyerni a magánkulcsot, feltehetően sokkal nehezebb, mint egy intelligens kártyából.

- A HSM-mel védett magánkulcsot sem a szerver rendszergazdája, sem a szerverre esetleg bejutó támadó nem viheti észrevétlenül.

- A HSM-mel – bizonyos esetekben – nagyobb teljesítmény érhető el, mint szoftveres kulccsal. (Megjegyezzük, a szoftveres kulcs viszont könnyen duplikálható, így ár/teljesítmény viszonyuk sokkal jobb, hiszen egy HSM árából sok, szoftveres kulcsot használó számítógépet állíthatunk fel.)

Ugyanakkor ha valaki megfelelően azonosítja magát a HSM felé, akkor a HSM aláírja a tőle származó információt a tárolt magánkulccsal. Így hiába védjük a kulcsot HSM-mel, más védelmi intézkedések hiányában mind a szerver rendszergazdája, mind a szerverre bejutó támadó bármilyen információt aláírathat a HSM-mel. (Lásd: 6.7. ábra.) Általában kiegészítő védelmi intézkedésekkel szokás biztosítani, hogy egy rendszergazda önállóan ne írathasson alá tetszőleges információt a HSM-mel, illetve a hálózatról ne juthassanak be támadók a HSM-et működtető szerverre.

Bizonyos estekben jogszabály (vagy egyéb szabályozás) írja elő, hogy HSM-et kell használni. Például a hitelesítés-szolgáltatók szolgáltatói kulcsait kötelező HSM-mel védeni (4.5.1. fejezet)



6.7. ábra. **A HSM nem tudja kontrollálni, hogy mit ír alá, nem véd az ellen, hogy egy rosszindulatú aláírás-létrehozó alkalmazás bármit aláírathasson vele.**

6.3.4. Aláírás-létrehozó alkalmazás

6.3.4.1. Mit nevezünk aláírás-létrehozó alkalmazásnak?

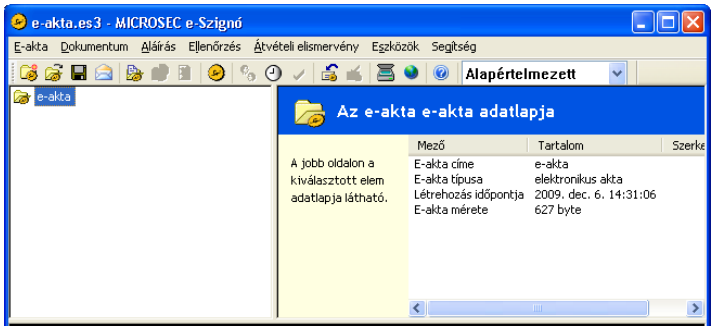
Azt a szoftvert nevezzük aláírás-létrehozó alkalmazásnak, amelynek segítségével az aláíró megadja, hogy mely dokumentumot szeretné aláírni, amely kiszámítja az aláírandó hash-t[10], és elküldi az aláírás-létrehozó eszköznek, amely majd létrehozza a szabványos formátumú aláírást.

A kriptográfiai értelemben vett aláírást nem az aláírás-létrehozó alkalmazás, hanem a magánkulcsot kezelő aláírás-létrehozó eszköz számítja ki. Szoftveres kulcs esetén (6.3.3. fejezet) előfordulhat olyan eset, amikor az aláírás-létrehozó alkalmazás egyúttal az aláírás-létrehozó eszköz szerepét is betölti, ekkor közvetlenül ő kezeli a magánkulcsot. Szintén előfordulhat, hogy az aláírás-létrehozó alkalmazás egyúttal az aláírás-ellenőrző alkalmazás szerepét is betölti. (Lásd: 6.5. fejezet.)

A CWA 14170 fogalmaz meg követelményeket az aláírás-létrehozó alkalmazásra vonatkozóan. [30] A másik mértékadó követelményrendszert a US Government Family of Protection Profiles Public Key-Enabled Applications For Basic Robustness Environments elnevezésű Common Criteria védelmi profil család jelenti. [187]

Tekintve, hogy az aláírás-létrehozó eszköz nem tudja kontrollálni, hogy az aláíró mely dokumentumot írja alá (lásd: 6.7. ábra), így igen nagy felelősség hárul az aláírás-létrehozó alkalmazásra. Ha egy támadó befészkeli magát az aláírás-létrehozó alkalmazásba, bármit aláírathat az aláíróval. Egészen egyszerűen lecseréli az aláírandó dokumentum lenyomatát

[10]Előfordulhat, hogy nem az aláírás-létrehozó alkalmazás, hanem az aláírás-létrehozó eszköz (intelligens kártya) számítja ki a hash-t. Az is lehet, hogy a hash-t közösen számítják ki, ilyenkor a műveletek többségét az alkalmazás végzi el, az aláírás-létrehozó eszközre csak a hash számítás legutolsó, befejező lépései hárulnak. Ez szűkíti a szóba jöhető alkalmazások körét, viszont nem látjuk, milyen biztonsági előnyt jelent az aláíró számára. Van olyan ország, ahol minősített aláírás csak így hozható létre. A továbbiakban azt feltételezzük, hogy az aláírás-létrehozó alkalmazás számítja a hash-t.



6.8. ábra. **Az e-Szignó program**

egy tetszőleges dokumentum lenyomatára, és azt küldi el az aláírás-létrehozó eszköznek aláírásra. [10], [7] Az aláírás-létrehozó alkalmazások figyelni szokták, hogy nem módosította-e őket idegen szoftver, de ki kell hangsúlyoznunk, hogy az ilyen jellegű támadásokat pusztán szoftveresen elvileg nem lehet kivédeni. [177]

Az aláírás-létrehozó alkalmazás többféleképpen épülhet fel:

- Lehet önálló alkalmazás, amely lehetőséget biztosít az aláírandó dokumentum elkészítésére vagy kiválasztására, aláírására és elmentésére. Ilyen például az e-Szignó program (lásd: 6.8. ábra), de ilyen lehet például a Microsoft Word is.

- Gyakori, hogy az aláírás-létrehozó alkalmazás csak egy programkönyvtár, amely valamely általános célú ügyviteli alkalmazás hátttereként működik. Például előfordulhat, hogy a felhasználó a saját, vállalati SAP rendszerét használja, amely aláíráshoz az e-Szignó program könyvtárait használja. Az SAP rendszerében állítja össze az aláírandó dokumentumot, de aláíráskor az e-Szignó program vezeti végig az aláírás folyamatán.

- Ha az aláírást szerver készíti, akkor az aláírás-létrehozó alkalmazás szintén csak egy programkönyvtár, és ekkor egyáltalán nem tartozik hozzá felhasználói felület. Számlázó-rendszerek szoktak ilyen módon működni; a számlaadatokat például XML vagy PDF sablonba írják bele, és e kitöltött sablonokat – akár több százezret is – adják át aláírásra az aláírás-létrehozó alkalmazásnak.

- „Webes” aláírás esetén a felhasználó egy weboldalon adja meg, hogy mit szeretne aláírni. Ekkor a webszerveren futó pl. CGI programok állítják össze az aláírandó dokumentumot, ők számítják ki a hash-t, és ezt küldik vissza a felhasználó böngészőjének. A felhasználó számítógépére ekkor csak egy kis program – egy ActiveX control vagy egy Java applet – kerül, ez csak aláíratja a felhasználóval a hash-t, majd visszaküldi a kapott aláírást a szerverre, ahol előáll az aláírt dokumentum (pl. e-akta).

6.3.4.2. Olvasd el, mielőtt aláírod!

Néhány nagyon fontos szabályra szeretnénk felhívni a figyelmet az aláírás-létrehozó alkalmazásokkal kapcsolatban:

1. Az aláíróval tudatni kell, hogy ő éppen aláírást készít.
2. Az aláírónak kell, hogy legyen lehetősége megtekinteni az aláírandó dokumentumot.
3. Az aláírás előtt az aláíró egyértelműen jóvá kell, hogy hagyja, hogy mit ír alá. (Egyes értelmezések szerint minősített aláírás esetén az aláíró minden egyes aláíráskor különkülön meg kell, hogy adja a PIN kódját is.)
4. Az aláírónak kell, hogy legyen lehetősége elmenteni az aláírt dokumentumot.

Ezek elsősorban a minősített aláírásokra vonatkoznak, azaz teljes bizonyító erejű magánokirat készítésére. Bár a jogszabály szerint a minősített elektronikus aláírás követelménye csak a minősített tanúsítvány és a biztonságos aláírás-létrehozó eszköz, a fenti követelmények nélkül nagyon nehezen képzelhető el, hogy valamiből teljes bizonyító erejű magánokirat lesz. Így annak ellenére, hogy minősített tanúsítvány és BALE esetén minősített aláírás jön létre, az aláírás jogi szempontból nagyon könnyen támadhatóvá válik, ha igazolható[11], hogy az aláírónak nem volt lehetősége megtekinteni, hogy mit ír alá. (6.1. fejezet) Így a fentiek nem pusztán illemszabályok, hanem nagyon alapvető követelményei az elektronikus aláírás korrekt használatának.

Fokozott biztonságú aláírás esetén a fenti követelmények néhol lazábban kezelhetőek, de ekkor nem is beszélhetünk teljes bizonyító erejű magánokiratról. Tipikusan akkor szoktak fokozott biztonságú elektronikus aláírást használni, ha nagy mennyiségű dokumentumot kell aláírni. Ha szerver vagy más automatizmus készíti az aláírást, az általában csak fokozott biztonságú lehet.

6.3.4.3. Hogyan érjük el a magánkulcsot?

Az aláírás-létrehozó alkalmazás jelenti a kapcsolatot a felhasználó és az aláírás-létrehozó eszköz között. A következő módszerek valamelyikével szokás kapcsolatot létesíteni az aláírás-létrehozó eszközzel:

- Az aláírás-létrehozó alkalmazás *fájlként hivatkozik a magánkulcsra*, és így az aláírás létrehozó alkalmazás egyben az aláírás-létrehozó eszköz szerepét is betölti. Leggyakrabban PKCS#12, avagy PFX formátumú fájlra szokás ilyenkor hivatkozni. (E

[11] Ezt vagy az aláírás-létrehozó alkalmazás vizsgálatával, vagy – ha az nem áll rendelkezésre – akár tanúkkal is lehet igazolni.

fájlok a magánkulcson kívül az aláíró tanúsítványát és a hozzá tartozó tanúsítványláncot is tartalmazhatják.) A PFX fájlban a magánkulcsot egy jelszóval titkosítva is elhelyezhetjük, így a PFX mellett egy jelszót is meg kell adni a programnak.

E megoldásnak nagyon súlyos hátránya, hogy csak szoftveres kulcsok esetén használható, és a szoftveres kulcsnak fájlban kell lennie. Így eleve kizártak például az intelligens kártyára telepített tanúsítványok, és a minősített elektronikus aláírás használata is eleve ki van zárva, mert ott a magánkulcsnak BALE-n kellene lennie.

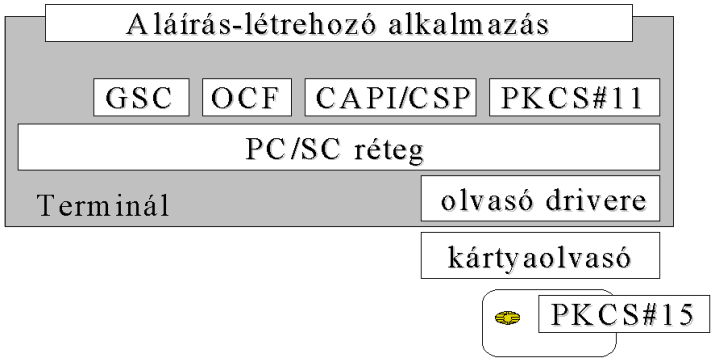
Sok alkalmazás – például a Mozilla – saját tanúsítványtárat működtet, amelybe PFX fájlok is betölthetőek. A Windows tanúsítványtárába is tölthető be PFX fájl.

- *Kriptográfiai hardver natív támogatása.* Lehet, hogy egy program meg tudja hajtani az X típusú kártyát, de az Y típusú kártyával vagy a Z típusú USB kulccsal, illetve a Q típusú HSM-mel már nem fog működni. Ez általában rossz megoldás, helyette szabványos interfészen célszerű meghajtani a kriptográfiai hardvereket, hogy a program más hardverekkel is együttműködhessen.

- Az aláírás-létrehozó alkalmazás *PKCS#11 könyvtár segítségével hivatkozik a magánkulcsra.* A PKCS#11 egy kriptográfiai token (kulcsot tároló hardver eszköz) interfészét írja le, és nagyon sok különböző kriptográfiai tokenhez – intelligens kártyához, HSM-hez stb. – készül PKCS#11 könyvtár. Elég megadnunk a szoftvernek a szükséges PKCS#11 könyvtárat, és a szoftver innentől le tudja kérdezni, hogy milyen kulcsok vannak a tokenen, milyen PIN kódok tartoznak hozzá, és tudja kezelni az adott típusú tokenen lévő kulcsokat. Sokféle platformon, sokféle hardverhez elérhetőek PKCS#11 könyvtárak.

- A CryptoAPI a PKCS#11-hez hasonló, de Microsoft-technológiákra épülő megoldás. Az egyes kriptográfiai tokenekhez a gyártók ún. CSP-t (cryptographic service provider) készíthetnek, amelyek képesek meghajtani a tokent. Ha egy windowsos gépre telepítettük egy adott token windowsos driverét, akkor a driver általában figyeli a géphez kapcsolt tokeneket. Amint behelyezzük a tokent, a driver felismeri a tokenen lévő tanúsítványokat, és „regisztrálja” őket a Windows központi tanúsítványtárába. Ez azt jelenti, megadja a Windowsnak, hogy az adott tanúsítványok magánkulcsát milyen CSP-vel lehet meghajtani. Ha egy alkalmazás a Windows tanúsítványtárát használja, úgy tudja kezelni az egyes tanúsítványokat, hogy nem kell tudnia, hogy azok milyen tokenen vannak, illetve tokenen vannak-e egyáltalán.

  A CryptoAPI jellemzően csak Windows platformon érhető el, de a Windows központi tanúsítványtára nagy mértékben megkönnyíti a tanúsítványok kezelését. (A PKCS#11 platformfüggetlen, így nem épít központi tanúsítványtárra. Az egyes alkalmazásoknak általában külön-külön meg kell mutatni az egyes tokenek PKCS#11 könyvtárait.)



6.9. ábra. **Kommunikáció a kártyával**

A legtöbb alkalmazás CryptoAPI-n vagy PKCS#11-en keresztül használja a kriptográfiai eszközöket (kártyákat, HSM-eket), de léteznek más, hasonló megközelítések is. Ilyen például az amerikai GSC (Government Smart Card) specifikáció vagy a Java alapú kripto-token interfész, az OCF (Open Card Framework) megoldása. [72]

A különféle kártyaolvasók általában a PC/SC rétegen keresztül érhetőek el, e réteghez kapcsolódhatnak az olvasók PC/SC kompatibilis driverei. Az olvasó driverén, illetve a PC/SC rétegen keresztül az adott eszköz CSP-je vagy PKCS#11 könyvtára alacsony szintű parancsokkal (APDU-kkal) fér hozzá az eszközhöz (pl. kártyához). A kártyán PKCS#15 adatstruktúra helyezhető el, amelyből a driver kiolvashatja, hogy pontosan milyen kulcsok hogyan és milyen PIN kódokkal érhetőek el a kártyán. [135]

(Lásd: 6.9. ábra.)

6.3.4.4. Biztonságos kapcsolat az aláírás-létrehozó eszközzel

Ha a támadó sikeresen beékelődik a felhasználó és az aláírás-létrehozó eszköz közé, alapvetően bármit aláíratha az aláíróval. [10], [8] Megteheti, hogy lehallgatja az aláírás-létrehozó eszköznek küldött PIN kódot, és ezt követően bármikor kezdeményezhet aláírás műveleteket. Ez bizonyos esetekben kivédhető. Például, ha a felhasználó PIN pades kártyaolvasót használ, és az olvasó PIN padén írja be a kódot, akkor a PIN kód soha nem jut be a számítógépébe, így a PIN kód védve van a gépet trójai programokkal irányítása alá vonó támadóval szemben.

Ugyanakkor a támadó azt is megteheti, hogy lecseréli az aláírandó dokumentum lenyomatát mielőtt az bekerül az aláírás-létrehozó eszközbe, és ez ellen a PIN pades kártyaolvasók sem nyújtanak védelmet. Ez ellen egyedül az védene, ha az aláírás-létrehozó eszköz saját, biztonságos felhasználói felülettel rendelkezne, amelyen a felhasználó megtekinthetné az aláírásra kerülő dokumentumot. [3], [25] Az aláírás-létrehozó célhardverek – például intelligens kártyák – azért nyújthatnak a személyi számítógépeknél nagyobb biztonságot, mert jelentősen egyszerűbbek, és kevés input-output perifériával rendelkeznek. Minél több „felhasználóbarát” funkciót helyezünk el rajtuk, annál több ponton lehet támadni őket, annál kevésbé valószínű,

hogy tanúsított, megbízható eszközökké válhatnak.

Hogyan védekezhetünk a közbeékelődéses támadással szemben? Egyik lehetőség, hogy csak biztonságos környezetben készítünk aláírást, ahol a támadó nem tud közbeékelődni. Hogy mi tekinthető biztonságos környezetnek, az nagyon függ attól, hogy pontosan milyen képességeket tételezünk fel a támadóról. A későbbiekben bemutatunk néhány elemi óvintézkedést egy számítógép védelmére (12.4.4. fejezet).

Másik lehetőség, hogy biztonságos kapcsolatot építünk ki az aláírás-létrehozó eszközzel, hogy a támadó ne tudjon közbeékelődni. Ezért a legtöbb aláírás-létrehozó eszköz kriptográfiailag védett biztonságos kapcsolatot (ún. secure channel) tud kialakítani, és akár az is beállítható, hogy kizárólag ilyen kapcsolaton keresztül fogadjon aláírandó lenyomatot. Akár egy SSL-hez hasonló tulajdonságokkal bíró, titkos és hiteles csatorna is kiépíthető, akár tanúsítvány alapon is. Az aláírás-létrehozó eszközökre, kártyákra vonatkozó specifikációk általában előírják valamilyen kriptográfiailag védett csatorna kialakítását [29], és szintén előírják, hogy a PIN kódnak is védett (de nem feltétlenül kriptográfiailag védett) csatornán kell eljutnia a kártyába.

A kriptográfiailag védett „secure channel” megoldások legnagyobb problémája, hogy nem tisztázott, pontosan mik között épül fel a biztonságos csatorna. A cél az volna, hogy a felhasználótól származó információ kerüljön aláírásra, és azt a közbeékelődő támadó ne módosíthassa észrevétlenül – sem az alkalmazásban, sem az operációs rendszerben, sem a meghajtóprogramokban, sem az olvasó USB kábelén, sem az olvasóban, sem a kártya kontaktusain. Ez azt jelentené, hogy a csatornának az emberi felhasználó és az aláírás-létrehozó eszköz között kellene kiépülnie, de ez nem lehetséges, mert az emberi felhasználó fejben nem képes kriptográfiai kódolásokat végezni.

> **Megjegyzés:** A szakirodalomban léteznek különféle gondolatkísérletek olyan kriptográfiai algoritmusok kialakítására, amelyek segítségével ember számítógép nélkül is képes lenne erős kriptográfiai védelmet biztosítani. [117], [107], [174], [165] Ezek többnyire vagy túl bonyolultak, vagy nem nyújtanak elég védelmet, vagy nem kriptográfiai, hanem biometriai alapokra épülnek. [9]

A támadó nagyon sok ponton próbálhat az aláíró és az aláírás-létrehozó eszköz közé ékelődni. Nem biztonságos környezetben többnyire fel kell tételeznünk, hogy az aláíró számítógépét is irányítása alá vonta. Ekkor nem sokat ér, ha a támadó által irányított gépen futó alkalmazás és a kártya között biztonságos csatorna létesül. Ha biztonságos környezetben dolgozunk, ahol a támadó nem tud közbeékelődni, akkor a secure channel megoldás felesleges.

Ugyanakkor problémákat is okozhat a secure channel. Ha a PIN kódot is ugyanazon a kriptográfiailag védett csatornán kell bevinni a kártyába, mint az aláírandó lenyomatot, akkor eleve kizártuk a PIN pades olvasók használatát. Ha a secure channelt a kártya drivere építi ki, akkor a számítógépről bármilyen alkalmazás elérheti a kártyát. Ha a secure

channelt az aláírás-létrehozó alkalmazás építi ki, amelynek megfelelő tanúsítvánnyal kell rendelkeznie, akkor kizártuk az általános célú alkalmazások többségét. Álláspontunk szerint a secure channel technológia elektronikus aláírás készítése esetén többnyire nem jelent érdemi védelmet, ugyanakkor a túl szigorú kikényszerítése bonyodalmakat okoz.

6.3.5. Azt látom, amit aláírok?

Mindig el kell olvasni, amit aláírunk. Míg a papír alapú dokumentumok világában viszonylag egyszerű betartani ezt az elvet, elektronikus dokumentumok esetén összetettebb problémával állunk szemben. Az elektronikus dokumentum egy bitsorozat a számítógépen. Amikor egy dokumentumot aláírunk, e bitsorozatból képzünk kriptográfiai lenyomatot, majd az így kapott lenyomatot kódoljuk a magánkulcsunkkal. A dokumentumon egy ember, egy természetes személy helyez el elektronikus aláírást, és szeretne róla meggyőződni, hogy valóban azt írja-e alá, amit a képernyőn lát.

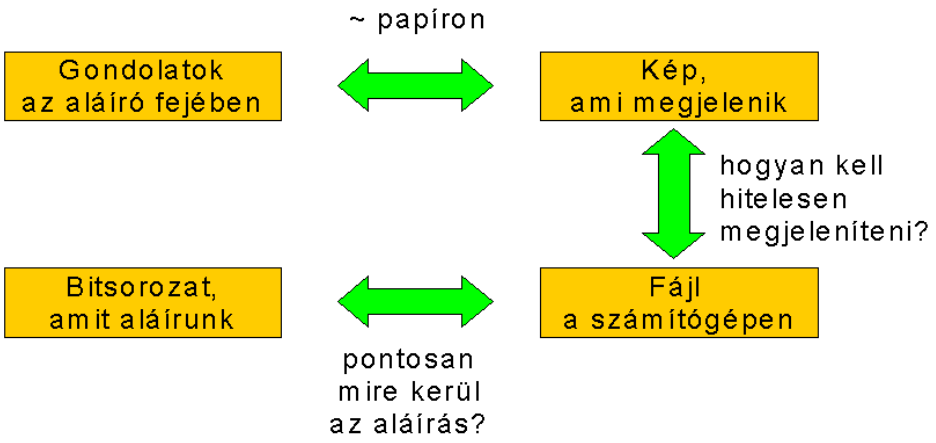
Gondoljuk végig, mi megy ekkor végbe! (Lásd: 6.10. ábra.)

1. Az aláíró kigondolja, mit szeretne aláírni, e gondolatait megpróbálja leképezni szavakkal a képernyőre. Ez már önmagában nem könnyű feladat, de papír alapon ugyanez a probléma áll elő.

2. Az aláíró egy dokumentum képét látja maga előtt. E dokumentum valamilyen bitsorozatnak felel meg a számítógépén. Az aláírás-létrehozó alkalmazások az értelmes dokumentumot mutatják meg az aláírónak aláírás előtt. Ha összetett dokumentumokat (pl. Word, Excel, PDF, HTML stb.) használunk, nem mindig egyértelmű, hogy ezeket pontosan hogyan kell megjeleníteni. Azt sem egyszerű eldönteni, hogy az aláírást befogadó, ellenőrző fél pontosan ugyanazt látja-e majd, amikor megjeleníti a dokumentumot. Problémát jelenthetnek, ha különböző felek különböző típusú, verziójú, esetleg csak különböző beállításokkal rendelkező vagy különböző környezetben működő alkalmazással tekintik meg a dokumentumot. További kérdéseket vet fel, hogy egyes dokumentum-formátumok megengednek ún. aktív tartalmakat is, így szándékosan is elő lehet állítani olyan dokumentumot, amely másképp jelenik meg különböző időpontokban vagy különböző környezetben.

3. Aláírás előtt a dokumentumot jelentő bitsorozat számos transzformáción megy keresztül, különféle információkkal egészül ki (pl. mellé kerül az aláíró algoritmus megnevezése, esetleg az aláíró X.509 tanúsítványa és az aláírási szabályzat megnevezése stb.), és az aláírásra kerülő lenyomatot általában nem az aláírandó dokumentumból, hanem egy szabványos, pl. ETSI TS 101 903 vagy PKCS#7 szerinti struktúrából képezzük.

Ha aláírás előtt azt szeretnénk megnézni, hogy mi az, amit valóban aláírunk, egyáltalán nem nyilvánvaló, hogy a folyamat mely pontját kellene vizsgálnunk. Amit látnánk, az minden



6.10. ábra. **Kapcsolat az aláírni kívánt tartalom és az aláírandó bitek között**

bizonnyal valamilyen bitsorozat lenne, és azt hiába néznénk, az emberi felhasználó nem tudná eldönteni, hogy az a bitsorozat valóban azt a dokumentumot jelenti, amit ő alá kíván írni.

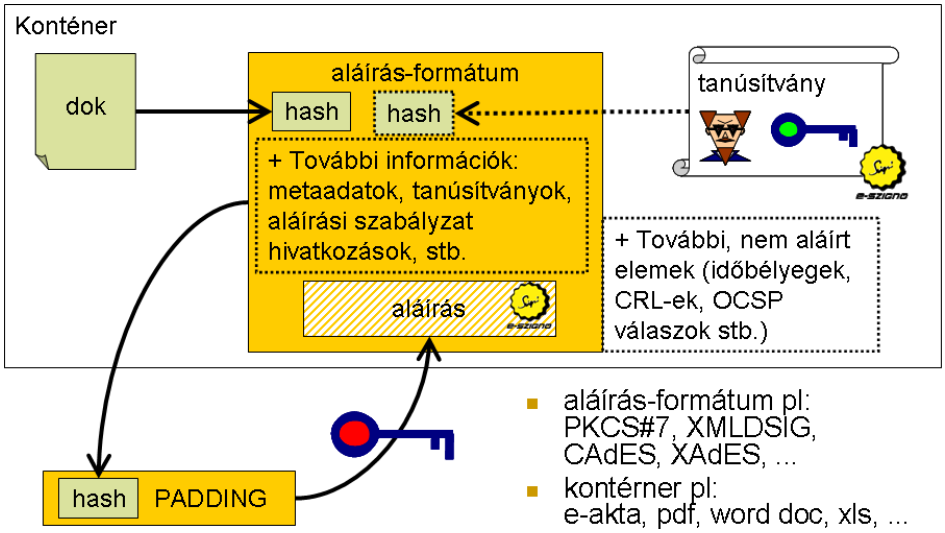
Továbbra is „mindig el kell olvasnunk, amit aláírunk”, de ilyenkor nem feltétlenül pontosan azt akarjuk látni, amit majd aláírunk, hanem az értelmes dokumentumot szeretnénk megtekinteni.

6.4. Az aláírás formátuma

Aláíráskor nem magát az aláírandó dokumentumot, hanem annak lenyomatát kódoljuk a magánkulcsunkkal. (Lásd: 2.12. ábra.) A lenyomat aláírásával biztosítható, hogy az aláírás az aláírt dokumentum minden egyes bitjétől függ, ha akár egyetlen bitet is megváltoztatunk a dokumentumban, már egészen más aláírás kellene, hogy tartozzon hozzá. (Lásd: 6.3.1. fejezet.)

A gyakorlatban mindez némiképp összetettebb. Bár a fent leírt módon is készülhetne elektronikus aláírás, általában nem közvetlenül az aláírandó dokumentum lenyomatát kódoljuk a magánkulcsunkkal. A gyakorlatban a következő történik (lásd: 6.11. ábra):

1. Az aláírandó dokumentumból lenyomatot képzünk.
2. A lenyomatot beillesztjük egy *aláírás-blokkba*. Ezen aláírás-blokk a következőket tartalmazhatja:
   - Azon dokumentumok lenyomatait, amelyeket aláírunk. Előfordulhat, hogy egyszerre több dokumentumot látunk el aláírásunkkal.
   - A használt lenyomatképző függvények típusát.



6.11. ábra. **A dokumentumból lenyomatot képzünk, ezt egy aláírás-blokkban helyezzük el. Az aláírás-blokkból képzett lenyomatot kódoljuk a magánkulcsunkkal**

Lényeges rögzíteni, hogy milyen lenyomatképző függvényt használunk. Tegyük fel, hogy az m dokumentum f lenyomatképző függvénnyel képzett lenyomata X. Tegyük fel, hogy az m' dokumentum f' függvénnyel képzett lenyomata szintén[12] X. Ekkor az X lenyomatra készített aláírás egyaránt tartozhatna az m és az m' dokumentumokhoz.

- Az aláíráshoz használt magánkulcshoz tartozó tanúsítvány lenyomatát.

  Ha az aláírás nem hivatkozza meg az aláíró tanúsítványát, akkor csak a tanúsítványban szereplő nyilvános kulcs kapcsolja össze az aláírással. Ekkor az aláíró tanúsítványa lecserélhető lehet egy másik, azonos nyilvános kulcsot tartalmazó tanúsítványra, és egy támadó úgy tehet, mintha az aláírást egy másik tanúsítvány szerint készítették volna.

  6.6. Példa: *Alajos aláír egy dokumentumot, elküldi Manfrédnak. Manfréd lecseréli Alajos tanúsítványát Alajos egy korábbi, már lejárt tanúsítványára (amelyet Alajos azóta megújított), és később arra hivatkozik, hogy azért nem fogadta be a dokumentumot, mert a tanúsítvány érvénytelen volt. Lehet, hogy van, akit sikeresen meg tud így téveszteni.*

  6.7. Példa: *Alajos közjegyző, közjegyzői tevékenysége során közjegyzői tanúsítványt használ, magánszemélyként egy személyes tanúsítványt használ. A két tanúsítványban azonos nyilvános kulcs szerepel. Ekkor*

[12] Tekintve, hogy különböző lenyomatképző függvényekről van szó, a lenyomatképző függvények ütközésellenállósága nem véd ez ellen. Ugyanakkor ha mind az f, mind az f' lenyomatképző függvények őskép-ellenállóak, valószínűleg nagyon nehéz ilyen (m,m')-párt találni.

Manfréd át tudja alakítani Alajos közjegyzőként aláírt dokumentumait magánszemélyként aláírt dokumentumokká, magánszemélyként aláírt dokumentumaiból pedig közjegyzőként aláírt dokumentumokat tud készíteni.

6.8. Példa: *Alajos aláír egy dokumentumot. A magánkulcsa később kompromittálódik (Manfréd kezébe kerül), és visszavonják a tanúsítványát. Manfréd Alajos kompromittálódott magánkulcsával készít egy PKCS#10-es tanúsítványkérelmet, és tanúsítványt igényel egy hitelesítés-szolgáltatótól a saját nevére. Ezt követően megpróbálhat úgy tenni, mintha egy Alajos által korábban aláírt dokumentumot ő írt volna alá.*

A esetek egyes nézetek szerint súlyos problémák, és – e nézetek szerint – ha egy aláírás nem függ az aláíró tanúsítványának lenyomatától, akkor már nem felel meg a fokozott biztonságú elektronikus aláírás követelményeinek. Más nézetek szerint e problémák nem súlyosak, és az az aláírás is tekinthető fokozott biztonságúnak, amely nem függ az aláíró tanúsítványának a lenyomatától. Az biztos, hogy „jobb", ha az aláírás meghivatkozza az aláíró tanúsítványát, mert akkor a fenti támadások nem végezhetőek el, ezért ilyen aláírásokat célszerű készíteni. (A vita a már létező, e hivatkozás nélkül készített aláírásokkal kapcsolatban merül fel.)

A fenti támadások mindegyike orvosolható, ha a hitelesítés-szolgáltató például olyan gyakorlatot követ, hogy nem fogad el PKCS#10-es kérelmeket, és mindig ő generálja a magánkulcsot, valamint minden tanúsítványhoz új magánkulcsot generál.

- Az egyes aláírandó dokumentumok formátumának meghivatkozását. A formátumot például MIME típussal lehet egyértelműen hivatkozni.

  Az aláírás mindig egy bitsorozathoz kapcsolódik. Előfordulhat, hogy az a bitsorozat több dolgot is jelenthet. Előfordulhat, hogy valaki arra hivatkozik, hogy habár érvényes az aláírása egy adott HTML dokumentumon, ő soha nem fogadta el a HTML-ben szereplő szöveget; ő ugyan aláírta ugyanezen bitsorozatot, de ő egy BMP képet írt alá, és a képen szereplő szöveg mást jelent.

  E támadás nem pusztán elméleti. Kutatók a közelmúltban mutattak olyan bitsorozatot, amely egyaránt értelmezhető HTML-ként és BMP-ként, és az egyes formátumokban értelmezve eltérő jelentéssel bír. [20]

- Az aláírásra vonatkozó különféle metaadatokat, amelyeket az aláíró állít az aláírással kapcsolatban. Például az aláírás idejét, helyét, az aláírás funkcióját[13],

[13] Az aláírás jelentheti a dokumentum tartalmának elfogadását, jelentheti azt, hogy az aláíró látta a dokumentumot, az a birtokában volt, de azt is jelentheti, hogy ő készítette a dokumentumot, és az tőle származik. Papír alapú aláírások esetén ezek gyakran abból derülnek ki, hogy az aláírás a papíron hova kerül. Elektronikus aláírás esetén ezt az aláírás-blokkban szabványos módon is jelezhetjük, de mindez a dokumentum tartalmából is kiderülhet.

az aláíró szerepét[14].

- Hivatkozást tartalmazhat aláírási szabályzatra.

Léteznek ASN.1 DER és XML formátumú aláírás-blokkok. [192], [92] Az előbbi csoportba tartozik például a PKCS#7/CMS és a CAdES, az utóbbiba az XMLDSIG és a XAdES. A későbbiekben egyes formátumokat részletesebben is megvizsgálunk.

3. Az aláírás-létrehozó alkalmazás az ezen aláírás-blokkból képzett lenyomatot küldi el az aláírás-létrehozó eszköznek.

   Ez azt jelenti, hogy nem közvetlenül az aláírandó dokumentum lenyomatát kódoljuk a magánkulcsunkkal, hanem az aláírandó dokumentum lenyomatát egy rögzített struktúrában helyezzük el, és e struktúra lenyomatát kódoljuk a magánkulcsunkkal. Ha „erős” lenyomatképző függvényeket használunk, akkor e többszörös indirekció nem jelent biztonsági kockázatot.

4. A kapott aláírást (azaz a magánkulccsal kódolt lenyomatot) az aláírás-létrehozó alkalmazás elhelyezi az aláírás-blokkban.

5. Az aláírás-létrehozó alkalmazás további, nem aláírt információkat csatolhat az aláíráshoz. Ide tartozhat például:

   - Az aláíráson elhelyezett időbélyeg.
   - Az aláíró tanúsítványához tartozó tanúsítványlánc.
   - Az aláíró tanúsítványára, illetve a láncban szereplő tanúsítványokra vonatkozó visszavonási információ (CRL-ek, OCSP válaszok). Az aláíráshoz pozitív visszavonási információkat szokás csatolni, amelyek alátámasztják, hogy az aláíró tanúsítványa valóban érvényes volt az aláírás pillanatában.
   - A fenti adatokra vonatkozó további időbélyegek, illetve archív időbélyegek.

6. Az így kapott, aláíratlan elemekkel kiegészített aláírás-blokkot az aláírás létrehozó alkalmazás egy ún. *aláírás-konténerbe* szokta csomagolni. A felhasználók és az alkalmazások általában konténerben lévő aláírásokkal szoktak találkozni.

   Aláírás-konténer formátum minden olyan fájlformátum, amely aláírást tartalmazhat. Egyes konténerek egyúttal dokumentum-formátumok is, ilyen pl. a PDF, a Microsoft Word, az Excel stb. Más konténerek elsősorban arra szolgálnak, hogy különféle – tetszőleges típusú – dokumentum elhelyezhető bennük, ilyen például az e-akta és az S/MIME csatolmánnyal rendelkező e-mail.

[14]Például ha az aláíró orvosként vagy közjegyzőként írt alá egy dokumentmot. Ez feltüntethető stringként is, de akár attribútum-tanúsítvány formátumú igazolás is csatolható. (Lásd: 11. fejezet.)

6.4.1. Az aláírás-blokk

A fentiekből látszik, hogy mind az aláírás-blokk, mind az aláírás-konténer fontos szerepet játszik az aláírás formátumában.

Az *aláírás-blokk* tartalmazza magát a kriptográfiai értelemben vett aláírást, emellett meghivatkozza, hogy az aláírás mire vonatkozik, leírja, hogy az aláírás milyen algoritmusok szerint készült, meghivatkozhatja (egyes formátumok, pl. a XAdES esetén kötelezően meg is hivatkozza) az aláírói tanúsítványt, különféle leíró információkat és metaadatokat tartalmazhat az aláírásról, meghivatkozhatja az aláírási szabályzatot, tartalmazhat segítséget az aláírás ellenőrzéséhez (pl. tanúsítványláncot), és tartalmazhat időbélyeget és az archiváláshoz szükséges információkat. Ezen adatok egy része elengedhetetlenül szükséges az aláírás kriptográfiai ellenőrzéséhez, egy másik része csak segítséget nyújt hozzá. Például ha az aláírás-blokkban ott szerepel egy tanúsítványlánc, akkor az aláírás-ellenőrző alkalmazásnak a nem feltétlenül kell láncot keresnie, hanem használhatja a megadott láncot (5. fejezet). Az aláírás-blokkban szereplő információk harmadik része a visszavonási állapot ellenőrzésében nyújt segítséget, negyedik része (ilyen pl. az aláíró szerepe) nem a műszaki, hanem a jogi, szabályzati szempontból végzett aláírás-ellenőrzéshez szükséges. Az aláírás-blokk jellemzően nem tartalmazza a kriptográfiailag aláírt lenyomatot, mert az előállítható belőle: ha a kriptográfiai értelemben vett aláírást kódoljuk az aláíró nyilvános kulcsával, az aláírt dokumentum paddinggel kiegészített lenyomatát kell megkapnunk.

Az aláírás-blokk egy jelentős részét aláírja az aláíró. Az aláírt elemek általában szükségesek az aláírás ellenőrzéséhez; ilyen például a lenyomatképző algoritmus megnevezése. A nem aláírt elemek általában csak segítséget nyújtanak az ellenőrzéshez, őket nem védi az aláírás, elvileg bármikor lecserélhetőek más adatokkal. Például ha az aláíráshoz csatolunk egy nem aláírt tanúsítványláncot, az egyfajta „súgást” jelent az aláírás-ellenőrző alkalmazásnak, aki vagy elfogadja a csatolt láncot, vagy keres magának másikat. Hasonlóképpen, ha időbélyeg védi az aláírást, lehet, hogy azt valaki lecseréli egy másik időbélyegre, és lehet, hogy az aláírás e másik időbélyeg szerint is érvényes. (Lásd: 6.5. fejezet.) A nem aláírt elemek általában PKI objektumok, amelyeken valamely más fél aláírása szerepel.

Nem védheti az aláírás azon csatolt információkat, amelyek függenek magától az aláírástól. Ilyen például az aláíráson elhelyezett időbélyeg: ezen időbélyeg létrehozásához szükség van az aláírás értékére, így az aláírás értékének kiszámításához nem használhatjuk az időbélyeget.

Sokféle aláírás-blokk létezik, leginkább ASN.1 és XML alapú blokkok terjedtek el. ASN.1 formátumú blokk például az RSA Laboratories által kidolgozott PKCS#7 és az erre épülő, és tőle csak minimálisan eltérő CMS (cryptographic message syntax) blokkja. XML formátumú blokk például a World Wide Web Consortium (W3C) által kidolgozott XMLDSIG. Az XMLDSIG és a PKCS#7 nagyjából hasonló információkat tartalmaz az aláírással kapcsolatban.

Az ETSI kiterjesztette a fenti ASN.1 és XML alapú aláírás-formátumokat a csak az Európai Unióban létező minősített elektronikus aláírás koncepciónak megfelelően. A CMS kiterjesztésével jött létre a CAdES (CMS Advanced Electronic Signature), az XMLDSIG kiterjesztésével jött létre a XAdES (XML Advanced Electronic Signature). E kiterjesztésekkel olyan fokozott biztonságú aláírások (advanced electronic signature) jöhetnek létre, amelyek megfelelhetnek a minősített aláírásokra vonatkozó követelményeknek is. Az ETSI megszorításokat tett a CMS, illetve XMLDSIG aláírásra, illetve definiált további, beléjük illeszthető elemeket. Ezek egy része azt a célt szolgálja, hogy egyértelmű legyen, ki készítette az aláírást (például a CAdES és a XAdES formátumok szerint az aláírásnak kötelezően függenie kell az aláíró tanúsítványának lenyomatától, míg a CMS és az XMLDSIG formátum esetén ez csak opcionális). Másik részük azt a célt szolgálja, hogy az aláírás érvényessége hosszú távon is igazolható maradjon. Így a CAdES és XAdES formátumok leírják, hogy hogyan helyezhető el az aláíráson időbélyeg, illetve hogyan csatolhatóak hozzá visszavonási információk, és hogyan lehet ezeket archív időbélyeggel ellátni (6.6. fejezet).

A CAdES és XAdES formátumok olyan értelemben kompatibilisek a CMS és az XMLDSIG formátumokkal, hogy egy CAdES és XAdES aláírás egyúttal CMS, illetve XMLDSIG aláírás is. Így ha egy alkalmazás érti a CMS, illetve XMLDSIG formátumokat, de soha nem hallott a CAdES-ről vagy a XAdES-ről, akkor a CAdES és XAdES aláírásokat CMS, illetve XMLDSIG aláírásoknak fogja tekinteni.

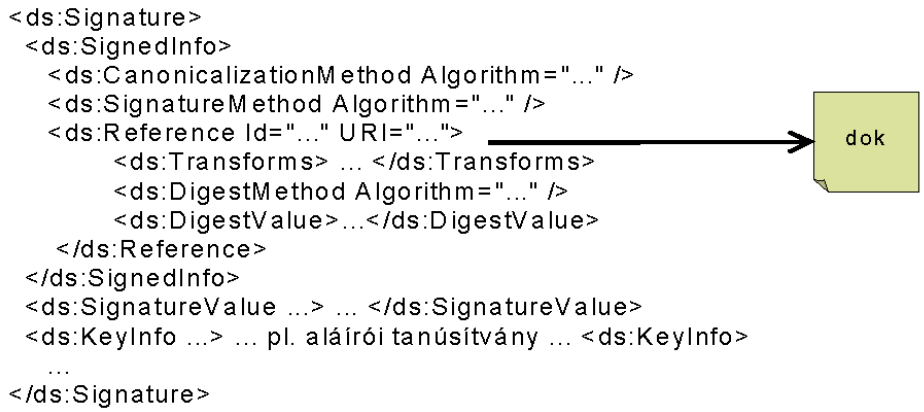
Ugyanakkor – az ETSI által tett megszorítások miatt – nem minden CMS vagy XMLDSIG aláírásból készíthető CAdES vagy XAdES aláírás. (Itt elsősorban az aláíró tanúsítványának hivatkozása okozhat problémát.) Csak akkor készíthető CAdES, illetve XAdES aláírás egy CMS vagy XMLDSIG aláírásból, ha azok teljesítik a [CX]AdES elemi formátumaira ([CX]AdES-BES vagy EPES) vonatkozó alapkövetelményeket. (Az aláírt információkon már nem lehet változtatni, de ha azok teljesítik a szükséges követelményeket, akkor nincs akadálya, hogy a nem aláírt információk közé további információkat csatoljunk.)

A következőkben részletesen is megvizsgálunk néhány aláírás-blokk típust.

6.4.1.1. XMLDSIG aláírás

Az XMLDSIG egy XML alapú aláírás-formátum, amelyet a World Wide Web Consortium (W3C) dolgozott ki. Az XMLDSIG aláírás egy `Signature` nevű elemben helyezkedik el. A `Signature` mindenképpen tartalmaz egy `SignedInfo`, egy `SignatureValue` és egy `KeyInfo` elemet, de ezen túl további elemeket is tartalmazhat. (Lásd: 6.12. ábra.)

A `SignedInfo` blokkban szerepel az aláírt tartalom. Aláíráskor e blokkból képzünk lenyomatot, és e lenyomatot írjuk alá. A `SignatureValue` tartalmazza a kriptográfiai értelemben vett aláírást, base64 kódolással. A `KeyInfo` az aláírói kulcsra vonatkozó



6.12. ábra. **XMLDSIG aláírás**

információkat tartalmazhat, PKI alapú aláírás esetén[15] az aláíró tanúsítványa szerepel itt.

Vizsgáljuk meg részletesebben a `SignedInfo` elemet! A benne szereplő `Reference` elemek hivatkozzák meg, hogy milyen tartalomra (dokumentumokra) vonatkozik az aláírás. A `Reference` URI-val hivatkozik rá, hogy hol található az aláírt tartalom. Az URI hivatkozhat az adott XML fájl valamely elemére, egy másik fájlra, egy másik XML fájl valamely elemére, de akár egy weben lévő tartalomra is. Az URI csak „súgás” az ellenőrző alkalmazásnak arra vonatkozóan, hogy hol kell keresni az aláírt tartalmat, kriptográfiailag a `Reference` elemben lévő lenyomat (`DigestValue`) kapcsolja össze az aláírást az aláírt tartalommal. (Előfordulhat, hogy az URI-n lévő tartalom megváltozik, mert egy fájlt elmozgattak, vagy mert a hivatkozott webhely nem érhető el. Az aláírás ettől függetlenül elfogadható, ha az eredeti tartalom megvan, az összekapcsolható az aláírással.)

Ezen kívül minden egyes `Reference` tartalmazza, hogy milyen algoritmussal készült az adott lenyomat, illetve milyen egyéb transzformációkat kellett elvégezni az aláírt tartalmon, hogy az adott URI-n lévő bitsorozatot kapjuk. (A szöveges tartalmakat és az XML tartalmakat általában egy az egyben, további transzformációk nélkül szokás elhelyezni az XML-ben, míg a bináris tartalmak általában base64 kódolással szerepelnek.)

A kriptográfiai értelemben vett aláírást a `SignedInfo` XML elemből képzett lenyomat kódolásával kapjuk. A `SignedInfo` elemben szereplő `Reference` elemek is hivatkozhatnak más XML elemre, ekkor is egy XML elemből kell lenyomatot képezni.

Ha XML elemből képzünk lenyomatot, problémát jelenthet, hogy egyazon jelentésű XML többféle módon is leírható, illetve egy XML-t feldolgozó alkalmazás átalakíthatja az XML-t, úgy, hogy az más bitekből áll, de továbbra is ugyanazt jelenti. Nem változik például az XML jelentése, ha bizonyos helyekre whitespace karaktereket teszünk, megváltoztatjuk a névterek jelölését, vagy ha másfajta kódolással ábrázoljuk az XML-t. E probléma feloldása

[15] Az XMLDSIG nem kizárólag PKI alapú aláírásokat enged meg.

végett minden egyes alkalommal, ha XML-ből lenyomatot képzünk, azt a lenyomatképzés előtt *kanonizálni* kell, azaz meghatározott kanonikus alakra kell hozni. A kanonizáció rögzíti például hogy az XML-ben hol legyenek whitespace-ek, az XML UTF-8 kódolású legyen, és hogy hogyan kell egységesen jelölni a névtereket. A `SignedInfo` tartalmazza, hogy lenyomatképzés előtt pontosan milyen kanonizációs algoritmust kell használni, és hogy milyen kriptográfiai algoritmusokkal készül az aláírás. Ez biztosítja, hogy a lenyomatképzés később is reprodukálható.

Az XMLDSIG specifikáció szerint olyan XML elemet helyezhetünk el egy XML fájlban, amely meghivatkozik egy vagy több URI-t, és tartalmazza az ott található tartalmak lenyomatait, és az így kapott XML elemre aláírást számíthatunk, amelyet meghatározott módon akár ugyanazon XML fájlban is elhelyezhetünk. Az XMLDSIG azt célozta meg, hogy az aláírás-ellenőrző alkalmazás rendelkezésére álljon, hogy pontosan milyen tartalomra és milyen algoritmusok szerint került az aláírás, és így az aláírás kriptográfiai ellenőrzését el lehessen végezni.

6.4.1.2. XAdES (XML Advanced Electronic Signature) aláírás

Az XMLDSIG elsősorban az aláírás kriptográfiai ellenőrizhetőségére fókuszál, de ez gyakran nem elengedő. Az ETSI XAdES formátuma az XMLDSIG megszorításával, illetve kiterjesztésével jött létre, és azt a célt szolgálja, hogy az aláírást valós, jogi környezetben is fel lehessen használni. A XAdES segítségével jobban rögzíthető, hogy mit jelent az aláírás, és hogyan kell azt értelmezni (ilyen például az aláíró szerepének megjelölése, illetve az aláírási szabályzat meghivatkozása), illetve csatolhatóak az aláíráshoz olyan információk – időbélyegek és visszavonási információk – amelyek alapján később könnyen (vagy könnyebben) igazolható, hogy az aláíró tanúsítványa érvényes volt az aláírás pillanatában.

Egy XAdES aláírás egyútal XMLDSIG aláírás is. (Lásd: 6.13. ábra.) A XAdES alapvetően annyiban nyújt többet az XMLDSIG-nél, hogy:

1. Az `xmldsig:Signature` elemében elhelyez egy további objektumot, amelyben a XAdES-kiterjesztéseket tartalmazó `xades:QualifyingProperties` elem szerepel. Ebben a `xades:SignedProperties` alatt szerepelnek az aláírt XAdES-kiterjesztések, például az aláírási szabályzat hivatkozása, az aláírás (aláíró által állított) helye és ideje, és az aláíró szerepe stb. A `xades:UnsignedProperties` elemében szerepelnek a nem aláírt XAdES-kiterjesztések, például az aláíráson elhelyezett időbélyeg, az aláíráshoz csatolt visszavonási információk és az archiváláshoz szükséges információk stb.

2. Az aláírónak aláíráskor kötelezően alá kell írnia a saját tanúsítványát is. Így az `xmldsig:SignedInfo` elemében legalább három `xmldsig:Reference` szerepel. Az egyik az aláírt dokumentumra hivatkozik, a másik pedig az aláíró tanúsítványára (amely

```
<ds:Signature>
 <ds:SignedInfo>
   <ds:CanonicalizationMethod Algorithm="..." />
   <ds:SignatureMethod Algorithm="..." />
   <ds:Reference Id="..." URI="..."> ... </ds:Reference>
 </ds:SignedInfo>
 <ds:SignatureValue ...> ... </ds:SignatureValue>
 <ds:KeyInfo ...> ... <ds:KeyInfo>
 <ds:Object><xades:QualifyingProperties>
        <xades:SignedProperties> aláírási szabályzat ref., aláírás  helye,
        ideje, aláíró szerepe stb. </xades:SignedProperties>
        <xades:UnsignedProperties> időbélyeg, visszavonási
        információk,archiváláshoz szükséges információk
   </xades:UnsignedProperties>
   </xades:QualifyingProperties></ds:Object>
</ds:Signature>
```

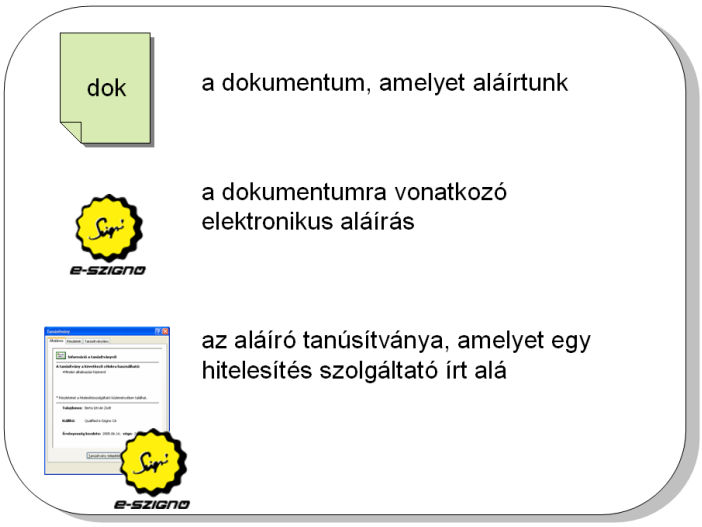
6.13. ábra. **XAdES aláírás**

általában az `xmldsig:KeyInfo` elem alatt szerepel), a harmadik pedig az aláírt XAdES kiterjesztésekre, azaz az aláírás `xades:SignedProperties` elemére.

Attól függően, hogy pontosan milyen XAdES-kiterjesztéseket csatolunk az aláíráshoz, különböző XAdES típusú aláírások jöhetnek létre. A továbbiakban ezen XAdES típusokat tekintjük át, amelyek elsősorban abban térnek el egymástól, hogy milyen további bizonyítékokat csatoltak az aláírást jelentő bitfolyamhoz a későbbi ellenőrizhetőség érdekében, illetve hogyan állapítható meg az aláírás készítésének időpontja, és mi bizonyítja, hogy az aláíró tanúsítványa akkor még érvényes volt. Hangsúlyozzuk, hogy az itt tárgyalt *különböző XAdES típusú aláírások jogi szempontból egyenértékűek*, de *a gyakorlatban előfordulhatnak olyan esetek, amikor* – hosszabb-rövidebb idő elteltével – *bizonyos fajta aláírások letagadhatatlansága műszaki szempontból megkérdőjelezhető*, és így akár a hozzá kapcsolódó bizonyító erő is elveszhet.

Az alábbiakban azt mutatjuk be, hogy az egyes XAdES aláírások közül melyik milyen problémát old meg, és mikor melyiket célszerű alkalmazni. A következő XAdES aláírás-típusokat mutatjuk be:

- Alap aláírás (XAdES-BES avagy EPES)
- Időbélyeggel ellátott aláírás (XAdES-T)
- Ellenőrzési hivatkozásokkal kiterjesztett aláírás (XAdES-C)
- Ellenőrzési adatokkal kiterjesztett aláírás (XAdES-X-L)
- Archív aláírás (XAdES-A)



6.14. ábra. **Alap aláírás (XAdES-BES)**

6.4.1.2.1. Alap aláírás (XAdES-BES)

Az *alap aláírás* (XAdES-BES – basic electronic signature) a legegyszerűbb XAdES aláírás. Olyan XMLDSIG aláírás, amely meghivatkozza az aláíró tanúsítványát, illetve tartalmazza a XAdES-kiterjesztés minimálisan kötelező elemeit. (Lásd: 6.14. ábra.)

Aláírás ellenőrzésekor többek között *fel kell építeni a tanúsítványláncot*. Ez azt jelenti, hogy meg kell vizsgálni, hogy az aláíró tanúsítványát melyik hitelesítés-szolgáltató adta ki (illetve ezen hitelesítés-szolgáltató tanúsítványát melyik felsőbb hitelesítés-szolgáltató adta ki stb.), és vissza kell vezetni az aláíró tanúsítványát egy megbízható hitelesítés-szolgáltató megbízható gyökértanúsítványára (5. fejezet). A tanúsítványlánc csatolható az alap aláíráshoz, de nem feltétlenül aláírt elem. Így az aláírás nem köti meg, hogy őt milyen tanúsítványlánc szerint kell ellenőrizni. Egy adott tanúsítványlánc szerint létrehozott aláírás más tanúsítványlánc szerint is elfogadható, illetve az aláíráshoz csatolt tanúsítványlánc akár később észrevétlenül, az aláírás megsértése nélkül ki is cserélhető.

Aláírás ellenőrzésekor (6.5. fejezet) *a tanúsítványlánc minden elemére meg kell vizsgálni, hogy az adott tanúsítvány* az aláírás pillanatában *érvényes volt-e*. Máris eljutottunk egy nagyon izgalmas problémához: Az aláírást ellenőrző fél hogyan győződhet meg róla, hogy az aláírás mikor készült? A XAdES-BES aláírás tartalmazza ugyan az aláírás időpontját, de ezen időpont nem megbízható forrásból származik – tipikusan az aláíró számítógépe szerinti időpont kerül ide. Mivel az aláíró bármikor átállíthatja a számítógépében lévő órát, *a XAdES-BES aláírásban szereplő időpont csupán az aláíró állítását jelenti, nem bizonyítja, hogy mikor készült az aláírás.*

Miért fontos, hogy pontosan mikor készült az aláírás? Előfordulhat például, hogy egy tolvaj

ellopja az aláíró intelligens kártyáját, és valahogy hozzájut a kártya PIN kódjához is. (Ekkor mondjuk, hogy a kártyán lévő aláírás-létrehozó adat kompromittálódott.) Ekkor a tolvaj pontosan ugyanolyan aláírásokat hozhatna létre, mint az aláíró. Ha az aláíró időben észreveszi a lopást, felhívja a tanúsítványát kibocsátó hitelesítés-szolgáltatót, és letiltja a kártyát, ekkor a szolgáltató visszavonja a kártyájához tartozó tanúsítványt. Így a visszavonás után készült aláírások érvénytelenek, ezeket várhatóan senki sem fogadja el. *Ha olyan aláírást ellenőrzünk, amelyhez a tanúsítvány már nem érvényes, akkor meg kell győződni róla, hogy az aláírás akkor készült, amikor a tanúsítvány még érvényes volt.*

A XAdES-BES aláírás alapján nem lehet megállapítani, hogy az aláírás mikor készült. Ebből kifolyólag, ha az aláíró tanúsítványa lejár, vagy visszavonják, akkor a XAdES-BES aláírások érvényessége nem bizonyítható. Ezért azt javasoljuk, hogy az aláíráson a lehető leggyorsabban (amíg a XAdES-BES aláírás még érvényes, azaz amíg az aláíráshoz használt tanúsítvány érvényes) helyezzünk el egy időbélyeget is, hogy a fenti probléma ne fordulhasson elő. Egyik lehetőség, hogy ezen időbélyeget rögtön az aláíró hozza létre (azaz, ne XAdES-BES, hanem XAdES-T aláírást készítsen), másik lehetőség, hogy az időbélyeget az aláírást befogadó fél helyezi el. Lehetőleg ne csak a XAdES-BES aláírást őrizzük meg, mert az aláíró bármikor letagadhatja az ilyen aláírást, ha a tanúsítványát visszavonatja.

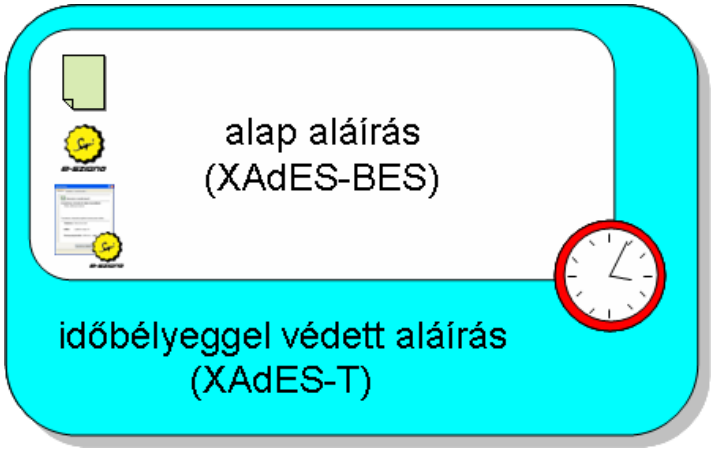
6.4.1.2.2. Alap aláírás, aláírási szabályzattal (XAdES-EPES)

Az -EPES aláírás annyiban nyújt többet a -BES aláírásnál, hogy az aláírásban – az aláírt elemek között – szerepelhet azon aláírási szabályzat hivatkozása, amely szerint az aláírás készült. Ez azt jelenti, egyértelműen rögzítve van, hogy az aláírás mit jelent, hogyan jött létre, és hogyan kell ellenőrizni. Az aláírási szabályzatokkal a későbbiekben részletesen is foglalkozunk. (Lásd: 6.8. fejezet.) Az -EPES aláírásban szerepelhet az aláírási szabályzat azonosítója, a lenyomata, és egy URL, ahol a szabályzat elérhető.

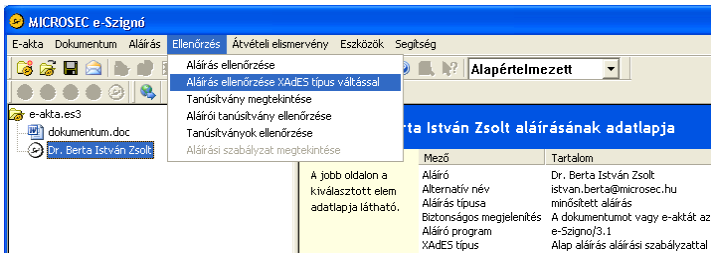
Az -EPES aláírásra is elmondható, hogy kizárólag addig igazolható az érvényessége, amíg az aláíró tanúsítványa érvényes. Az -EPES aláírás a -BES aláíráshoz hasonlóan kiterjeszthető magasabb szintű aláírásokká. Ugyanakkor az aláírási szabályzat hivatkozása aláírt elem. Ez azt jelenti, hogy -BES aláírásból utólag nem készíthető -EPES, és -EPES sem alakítható vissza egyszerű -BES aláírássá. A továbbiakban nem különböztetjük meg, hogy -BES vagy -EPES alapú -T, -C, -A stb. aláírásokról beszélünk, de ha egy aláírásban aláíráskor nem tüntették fel az aláírási szabályzatot, azt már később nem lehet elhelyezni benne.

6.4.1.2.3. Időbélyeggel ellátott aláírás (XAdES-T)

Az időbélyeggel ellátott elektronikus aláírás (XAdES-T, electronic signature with time) magában foglal egy XAdES-BES (vagy -EPES) aláírást, amelyet időbélyeg véd. (Lásd: 6.15.



6.15. ábra. **Időbélyeggel védett aláírás (XAdES-T)**



6.16. ábra. **XAdES-típus váltása az e-Szignó programmal**

ábra.) *Az időbélyeg igazolja, hogy az aláírás az időbélyegzés pillanatában már létezett*, vagyis nem később készült, és így például az aláíró tanúsítványának lejárta vagy visszavonása előtt jött létre. *Az időbélyeggel ellátott aláírás a legegyszerűbb olyan aláírás, amely biztosítja a műszaki értelemben vett letagadhatatlanságot.*

Aláírás létrehozásakor célszerű legalább XAdES-T aláírást készíteni. Ezt követően – amennyiben az adott dokumentum esetén ez szükséges – az aláírás később kiterjeszthető magasabb szintű XAdES aláírásokká is. (Lásd: 6.16. ábra.) Ezen kiterjesztést – a szabvány adta lehetőségek között – nem feltétlenül szükséges az aláírás pillanatában elvégezni, hanem később is történhet.

6.4.1.2.4. Visszavonási információkkal kiterjesztett aláírás (XAdES-C)

A XAdES-T aláíráshoz további információkat csatolhatunk, köztük olyan adatokat, amelyek a befogadó számára alátámasztják az aláírásunk érvényességét. A XAdES-C aláírásokban elhelyezhetjük az aláírásban mindazon visszavonási információkat (pl. CRL-eket), amelyek igazolják, hogy az aláíráshoz használt tanúsítvány az aláírás pillanatában érvényes volt. Megtehetjük, hogy magát a visszavonási információt csatoljuk az aláíráshoz, de az is lehet,

hogy csak egy rá mutató hivatkozást, URL-t teszünk ide. Ha csak a hivatkozás kerül az aláírásba, akkor kisebb aláírást kapunk, de a hivatkozás csatolása sokszor nem elegendő. A szabványok megengedik, hogy a hitelesítés-szolgáltató a visszavont és később lejárt tanúsítványokat ne tüntesse fel a CRL-ben (azért, hogy a CRL-ek idővel ne növekedhessenek kezelhetetlenül nagyra), így ha egy tanúsítvány már lejárt, utólag a CRL alapján nem lehet megállapítani, hogy lejárat előtt visszavonták-e. (Tegyük fel, hogy egy tanúsítványt egy évvel a lejárta előtt visszavontak. A tanúsítvány lejárta után kibocsátott CRL-ekből nem biztos, hogy meg lehet állapítani, hogy a tanúsítvány a lejárta előtt egy héttel érvényes volt-e.) Mivel a hitelesítés-szolgáltató nem köteles a korábbi CRL-eket is közzétenni[16], ezért a hosszú ideig megőrzött aláírásokhoz lehet értelme azon CRL-t hozzácsatolni, amely igazolja, hogy az aláírás érvényes. Ez megkönnyítheti a későbbi ellenőrzéseket. (OCSP-vel történő ellenőrzéskor magát az OCSP választ szokás csatolni az aláíráshoz. Rövid lejáratú OCSP tanúsítvány esetén ennek csak akkor van értelme, ha az OCSP válaszon is időbélyeget helyezünk el, és így már a XAdES-X-L aláíráshoz jutottunk.)

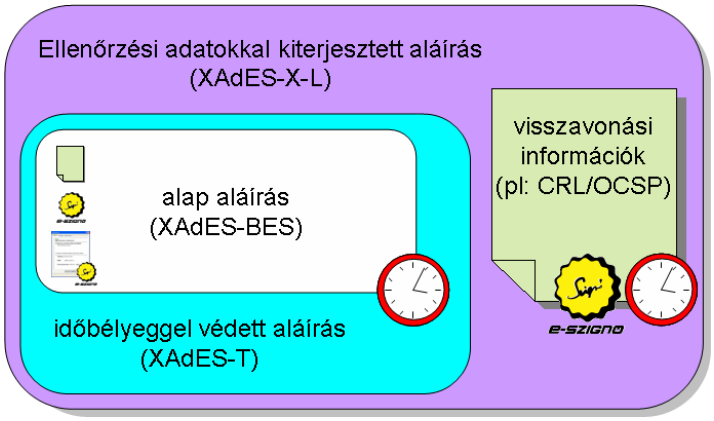
6.4.1.2.5. Időbélyeggel védett visszavonási információkkal kiterjesztett aláírás (XAdES-X-L)

További visszavonási információkat is csatolhatunk a XAdES-C aláíráshoz. Például a XAdES-C aláírás magában foglal egy időbélyeget (ugyanis tartalmazza a XAdES-T aláírást). Az időbélyeg ellenőrzéséhez az időbélyegzőre vonatkozó tanúsítványláncot is fel kell építeni, és az egyes tanúsítványok visszavonási állapotát ellenőrizni kell – az aláíró tanúsítványához hasonló módon. (Lásd: 6.5. fejezet.) Szintén csatolhatjuk a XAdES aláíráshoz az időbélyegre vonatkozó visszavonási listákat is. Attól függően, hogy pontosan milyen információkat csatolunk, különböző „magasabb” XAdES típusú aláírásokhoz juthatunk.

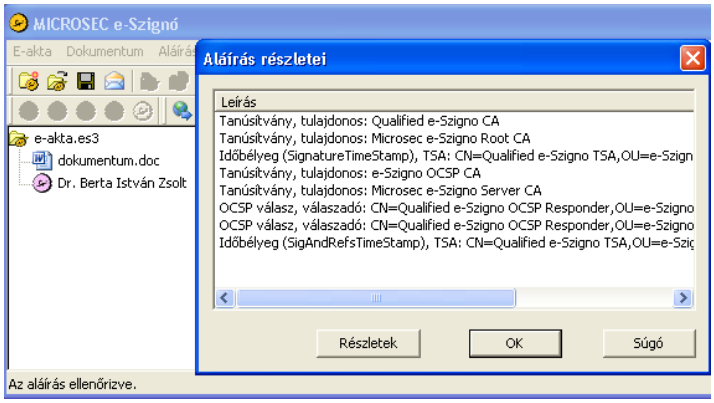
A XAdES aláírásban elhelyezett visszavonási listákon a hitelesítés-szolgáltató elektronikus aláírása szerepel. Ezen aláírás érvényessége – a végfelhasználó által létrehozott aláírások érvényességéhez hasonlóan – addig bizonyítható, amíg a hitelesítés-szolgáltató tanúsítványa érvényes. Ha azt szeretnénk, hogy a visszavonási listák – és így az aláírásunk – érvényessége ezt követően is igazolható legyen, a visszavonási információkat is időbélyeggel kell védenünk.

Ha az aláíráshoz csatolt visszavonási információkat időbélyeggel védjük meg, akkor XAdES-X-L aláírásról beszélünk. (Lásd: 6.17. ábra.) Az ilyen aláírások érvényessége azt követően is bizonyítható, hogy a tanúsítványláncban szereplő hitelesítés-szolgáltatók tanúsítványai lejártak, mert a szükséges visszavonási információk az aláíráshoz csatolva szerepelnek, időbélyeggel ellátva (lásd: 6.18. ábra).

[16] Csak az aktuális CRL-t lehet olyan szabványos módon közzétenni, hogy egy ellenőrző alkalmazás könnyen megtalálhassa a szükséges korábbi CRL-t.



6.17. ábra. **Ellenőrzési adatokkal kiterjesztett aláírás (XAdES-X-L)**



6.18. ábra. **XAdES-X-L aláírás részletei az e-Szignó programmal**

6.4.1.2.6. Archív aláírás (XAdES-A)

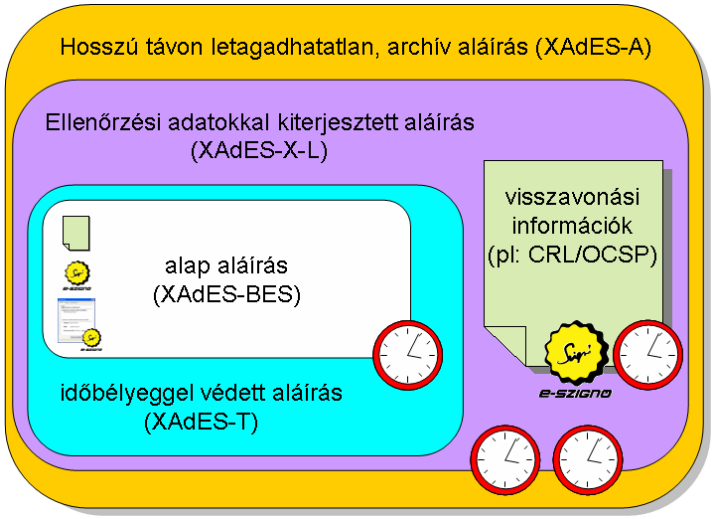
Az ún. archív aláírás magában foglal egy XAdES-X-L aláírást, tehát az aláíró tanúsítványát kibocsátó hitelesítés-szolgáltató közreműködése nélkül is ellenőrizhető. A XAdES-A archív aláírás további időbélyegeket (ún. archív időbélyegeket) tartalmaz a XAdES-X-L-hez képest, amelyek a teljes XAdES-X-L aláírást védik. Miért van erre szükség?

A nyilvános kulcsú infrastruktúra minden szereplőjének (nemcsak az aláíróknak, de a hitelesítés-szolgáltatóknak és az időbélyegzés-szolgáltatóknak is) van nyilvános kulcsa és magánkulcsa. A nyilvános kulcsot minden szereplő hiteles módon – például tanúsítvány formájában – nyilvánosságra hozza, a titkos kulcsot pedig titokban tartja. A rendszer addig működhet helyesen, amíg e titkos kulcsot valóban sikerül titokban tartani. Előfordulhat, hogy a magánkulcs (amely az aláírás-létrehozó adat szerepét is betöltheti) bizalmassága mégis megkérdőjelezhető. Ilyen egyrészt akkor történhet, ha feltételezhető, hogy a magánkulcsot illetéktelen személy megszerezte. Másrészt, akkor is megkérdőjelezhető egy magánkulcs bizalmassága, ha a tudomány és a technika fejlődésével lehetségessé válik a magánkulcs kiszámítása a nyilvános kulcs alapján. Szerencsére, csak nagyon ritkán következik be ekkora áttörés. A kulcsokat (és a kulcsokat használó kriptográfiai algoritmusokat) úgy szokták megválasztani, hogy erre belátható időn belül egyáltalán ne kerülhessen sor. Mégis *előfordulhat, hogy bizonyos méretű kulcsok elavulnak* (vagyis már nem jelentenek biztonságot), és *ilyenkor hosszabb, nehezebben kitalálható kulcsot szokás választani, és gondoskodni kell a korábban aláírt dokumentumokon lévő aláírások védelméről.*

Ha az az algoritmus és kulcsméret (pl: RSA algoritmus, 512 bit hosszú kulccsal), amely segítségével az aláíró az aláírást készítette, elavult, akkor nem biztosított az aláírás „letagadhatatlansága". E pillanattól kezdve ugyanis – hiába tartotta az aláíró titokban a magánkulcsát – más is létrehozhatta az aláírást az aláíró nevében. *Az elavult algoritmussal létrehozott aláírás csak akkor letagadhatatlan, ha* – például időbélyeg segítségével – *bizonyítható, hogy az aláírás akkor készült, amikor az algoritmus még nem avult el.*

Természetesen az időbélyeg is valamilyen kulcs és algoritmus segítségével készül, és ezek is elavulhatnak. Az elavult algoritmussal készült időbélyeg – az elavult algoritmussal készült aláíráshoz hasonlóan – nem rendelkezik bizonyító erővel. Így egy elavult technológiával készült aláírás letagadhatatlanságát csak egy olyan időbélyeg bizonyíthatja, amely friss, biztonságos technológiával készült, de akkor, amikor az aláírás készítésére használt technológia még nem számított elavultnak.

Előfordulhat például a következő eset: Egy olyan XAdES-X-L aláírást, amelyben az aláírás és az időbélyegek 512 bites RSA algoritmussal készültek (ez mára már elavult technológiának minősül) 768 bites RSA algoritmussal készült időbélyeggel védtek meg (mielőtt az 512 bites RSA algoritmus elavult). Ezt követően a 768 bites RSA-val készült időbélyeggel megvédett XAdES-X-L aláírást (tehát egy XAdES-A aláírást) 1024 bites RSA algoritmussal készült



6.19. ábra. **Archív aláírás (XAdES-A)**

archív időbélyeggel védtek meg, mielőtt a 768 bites RSA algoritmus elavult volna. Az 1024 bites RSA-val készült időbélyeggel megvédett XAdES-A aláírást célszerű 2048 bites RSA-val készült időbélyeggel is megvédeni. Ha az 1024 bites RSA elavul, akkor a 2048 bites RSA várhatóan elég hosszú ideig nyújt biztonságot, amíg az aláírást 4096 bites RSA algoritmussal (vagy más, a 2048 bites RSA-nál erősebb algoritmussal) készült időbélyeggel is megvédik.

A XAdES-A aláírás-formátum magában foglalja a XAdES-X-L aláírást, és a teljes XAdES-X-L-t archív időbélyegekkel is védi (lásd: 6.19. ábra). Az archív időbélyegek jellemzően egyre erősebb, biztonságosabb algoritmussal készülnek. Az egyes archív időbélyegek védik az aláírást, a hozzá tartozó tanúsítványláncot, időbélyegeket és visszavonási információkat, védik a korábbi archív időbélyegeket is. Az a helyes, ha egyúttal a korábbi archív időbélyegek visszavonási információit is védik (bár ezek helyét a XAdES specifikáció, nem definiálja).

A XAdES-A aláírás minden tanúsítványt, időbélyeget és visszavonási információt tartalmaz, leszámítva a legkülső archív időbélyegre vonatkozó információkat. XAdES-A aláírás ellenőrzésekor elegendő a külső archív időbélyeg visszavonási állapotát ellenőrizni, a többi ellenőrzés a XAdES-A aláírás alapján is elvégezhető.

A XAdES-X-L aláírással szemben a XAdES-A nem csupán az aláírás formátumát jelenti, hanem azt a *folyamatot* is, amely szerint az aláírást további archív időbélyegekkel kell ellátni. Célszerű új archív időbélyeget elhelyezni az aláíráson, ha a legkülső archív időbélyeg elkészítésére használt technológia várhatóan nemsokára elavul. Bizonyos rendszerességgel is célszerű újabb archív időbélyeget elhelyezni a XAdES-A aláíráson, mert – habár az időbélyegzés-szolgáltatók különösen gondosan védik az időbélyegek aláírására használt kulcsaikat – előfordulhat, hogy az időbélyegző kulcs illetéktelen kezekbe kerül, és ekkor az időbélyegek érvényessége csak az időbélyegzés-szolgáltató napló fájljai alapján bizonyítható.

Akkor célszerű XAdES-A archív aláírást készíteni, ha egy aláírt dokumentumot várhatóan nagyon sokáig (akár évtizedekig) meg kell őrizni, és az aláírás érvényességét évtizedekkel később is biztosítani kell. Természetesen az is elképzelhető, hogy aláíráskor mindig XAdES-T aláírást hozunk létre, és csak akkor alakítjuk XAdES-A aláírássá, ha kiderül, hogy a dokumentumra hosszú távon is szükség van. Ezt az átalakítást bárki elvégezheti, nincs szükség hozzá az aláíróra.

Bizonyos esetekben kötelező gondoskodni az aláírások hosszú távú hiteles megőrzéséről, archiválásáról. Ekkor a XAdES-A formátum, és annak gondozása jelenti az egyik lehetséges megoldást. Ugyanakkor a XAdES-A formátum gondozása az elektronikus aláírás technológiájának folyamatos figyelését és a dokumentumokon lévő aláírások hosszú távú gondozását (újra-időbélyegzését) igényli, ez nem feltétlenül könnyű feladat. Célszerű mérlegelni, hogy mikor érdemes az elektronikus aláírás érvényességének hosszú távú biztosításával erre felkészült, professzionális, a Nemzeti Média- és Hírközlési Hatóság által is nyilvántartásba vett archiválás-szolgáltatót megbízni. Az aláírások archiválásának kérdésével a későbbiekben részletesebben is foglalkozunk. (Lásd: 6.6. fejezet.)

6.4.1.2.7. Melyiket érdemes használni?

- *A XAdES-BES vagy -EPES aláírás jogilag rendelkezik az adott (fokozott biztonságú vagy minősített) elektronikus aláíráshoz fűződő bizonyító erővel, de mindez meginoghat, amint az aláíró tanúsítványa lejár vagy visszavonásra kerül.*
- E probléma megoldódik, ha a XAdES-BES vagy -EPES aláírást időbélyeggel látjuk el. Ezért *célszerű legalább XAdES-T aláírásokat létrehozni.* Ha ez valamilyen okból nem valósítható meg – például mert az aláírást végző gépről nem érhető el időbélyegzés-szolgáltatás – a -BES vagy -EPES aláírást célszerű a lehető leghamarabb kiterjeszteni XAdES-T aláírásra. E kiterjesztés bárki elvégezheti.
- Attól függően, hogy milyen hosszan van szükség az aláírás érvényességének bizonyítására, *az aláírás kiterjeszthető magasabb, akár XAdES-A aláírássá.* Ezen kiterjesztés bármikor elvégezhető, amíg a korábbi típusú aláírás érvényessége megállapítható.
- *Aláírt dokumentumok hosszú távú archiválására XAdES-A formátumot célszerű használni.* Az ilyen aláírás alapján a hitelesítés-szolgáltatók közreműködése nélkül is igazolható az aláírás érvényessége, ilyenkor egyedül a legkülső archív időbélyegre vonatkozó visszavonási információt kell ellenőrizni.

E kérdéskört a későbbiekben (12.4.13. fejezet) részletesebben is körüljárjuk.

6.4.1.3. CMS és CAdES (CMS Advanced Electronic Signature) aláírás

A PKCS#7, illetve CMS (cryptographic message syntax) aláírás-blokkok felépítése az XMLDSIG blokk felépítésével analóg, hasonló információkat tartalmaz. [153] Ugyanakkor az XML és a ASN.1 kódolás technológiák jellegzetességei miatt vannak bizonyos különbségek az XML és ASN.1 formátumok között. Például, XML formátum esetén jellemzően URI alapon, míg ASN.1 esetén jellemzően OID alapon hivatkozhatunk objektumokra. XML-ben, URI alapon könnyebb külső tartalmat meghivatkozni (például ha egy adott webcímen lévő tartalmat írtunk alá), míg ez OID alapon körülményesebb.

A CMS formátum CAdES kiterjesztése az XMLDSIG formátum XAdES kiterjesztésével analóg, ugyanúgy definiál -BES, -EPES, -T, -C, -A stb. aláírás-típusokat, ezek ugyanazokat a csatolt információkat tartalmazzák, és ugyanolyan szerepet töltenek be. A CMS és a CAdES aláírás-formátumokat itt külön nem mutatjuk be.

A PKCS#7 aláírások igen elterjedtek, mert nagyon sok program ezeket támogatja. A levelezőprogramokban lévő aláírások PKCS#7 alapúak, és szintén PKCS#7 aláírás szerepel a Word és Excel fájlokban, valamint a klasszikus PDF aláírás is PKCS#7 alapú. Az egységes aláírás-formátum ellenére szó sincs arról, hogy ezek a szoftverek el tudnák fogadni egymás aláírásait. Egységes aláírás-konténer nélkül meg sem találják az idegen fájltípusban lévő aláírásokat, és ha megtalálnák, akkor sem tudnák megállapítani, hogy az aláírás mire vonatkozik. A régi típusú, a 2007-es változat előtti Word és Excel esetén nemhogy az aláírás-konténer nem egységes, de ott még a fájlformátum sem nyilvános, így nehéz elképzelni, hogy bármilyen más alkalmazás fel tudja dolgozni ezeket az aláírásokat.

Egyes EU tagállamokban, például Olaszországban, a PKCS#7 aláírást használják minősített aláírásként is, és a [XC]AdES aláírások kevésbé terjedtek el. Sokan nem tekintik a PKCS#7 aláírást „komoly” célra alkalmas aláírásnak, mert nem függ az aláírói tanúsítvány lenyomatától, illetve nincsen rajta időbélyeg sem (ezt CAdES kiterjesztéssel lehetne orvosolni).

6.4.1.4. „Melasz-Ready” aláírás

A Melasz-Ready formátumot a Magyar Elektronikus Aláírás Szövetség (MELASZ) dolgozta ki. [111] A Melasz-Ready formátum a XAdES egy részhalmaza, megszorításokat tesz a XAdES tartalmára vonatkozóan. Ezáltal a Melasz-Ready specifikációnak megfelelni kívánó szoftverfejlesztőknek nem a teljes XAdES specifikációt kell implementálniuk, így a különféle alkalmazások könnyebben kezelhetik az egymás által létrehozott aláírásokat. A Magyar Elektronikus Aláírás Szövetség egyúttal átfogó interoperabilitási teszteket végzett, amelyek során több magyar aláírás-létrehozó alkalmazás Melasz-Ready minősítést kapott.

Az alkalmazások Melasz-Ready minősítése még csak annyit jelent, hogy az egyes alkalmazásoknak létezik olyan üzemmódja, amellyel egymással együttműködő formátumot hoznak létre, illetve el tudják fogadni egymás aláírásait. Amikor később az alkalmazásokat

nem ilyen üzemmódban használják, azok mégsem megfelelő formátumú aláírást hoznak létre, vagy nem megfelelő módon ellenőrzik azt, és így nem működnek együtt.

Ráadásul, az egységes aláírás-blokk formátum önmagában még nem jelenti, hogy az aláíró-alkalmazások elfogadnák egymás formátumát. *Egységes aláírás-konténer nélkül az alkalmazások nem feltétlenül találják meg az XML fájlokban az aláírást*, illetve az aláírt tartalmat. Több aláírás esetén nem feltétlenül tudják kibogozni az aláírások egymáshoz való viszonyát sem.

6.4.2. Az aláírás-konténer

Az *aláírás-konténer* meghatározott struktúrába foglal egy vagy több aláírás-blokkot, és egyúttal tartalmazhatja magát az aláírt dokumentumot vagy dokumentumokat is. Az aláírás-konténer formátumának ismeretében találja meg egy aláírás-ellenőrző alkalmazás az aláírás-blokkokat, és esetleg azt is, hogy az aláírások pontosan mire vonatkoznak. (Ez utóbbi van, amikor magából az aláírás-blokkból megállapítható, például egy XAdES aláírás meghivatkozhatja, hogy hol, mely URI-n található az aláírt tartalom. Előfordulhat, hogy a formátum ezt nem tudja meghivatkozni, és a konténert úgy kell értelmezni, hogy az aláírás mindenre vonatkozik, kivéve az aláírás-blokkra. Egyes PDF aláírásokat így kell értelmezni.)

Ha az aláírás-konténer maga is egy dokumentum-formátum (ilyen a PDF vagy a Microsoft Word), akkor a konténer határozza meg, hogy az aláírt tartalmat hogyan kell megjeleníteni. Ha a konténerbe különféle dokumentumokat lehet beilleszteni (ilyen az e-akta), akkor általában az aláírás-blokkokban az aláírt tartalom MIME típusa határozza ezt meg.

A konténerekben több dokumentum is lehet, és a dokumentumokon több aláírás is elhelyezkedhet. Az aláírások különféle viszonyban lehetnek egymással. Párhuzamos aláírásokról akkor beszélünk, ha azok egymástól függetlenek, így bármelyik eltávolítható anélkül, hogy a másik aláírás megsérülne. Ha az egyik aláírás függ a másiktól, akkor ellenjegyzésről beszélünk, azaz az ellenjegyző egy másik aláírást (is) aláírt. (Az e-aktában van ún. keretaláírás, amely az aktában lévő összes dokumentumtól, és összes, nem keret aláírástól függ.) A konténernek szerepe lehet az aláírások viszonyának tisztázása vagy megszorítása.

6.4.2.1. E-Akta

Az elektronikus akta (e-akta) elnevezés alatt az Igazságügyi és Rendészeti Minisztérium részére az elektronikus cégeljárás kapcsán a Microsec Kft. által kifejlesztett olyan aláírás-konténer fájlformátumot értünk, amely dokumentumokat és a dokumentumokon elektronikus aláírásokat és időbélyegeket tartalmaz. A dokumentumokon elhelyezett aláírások szabványos, az ETSI által kidolgozott ETSI TS 101 903 specifikációnak megfelelő XAdES aláírások lehetnek. [36]

Az e-akta specifikáció nem tesz megkötéseket a XAdES aláírásokkal kapcsolatban, hanem azt írja le, hogy egy XML fájlban (az ún. e-aktában) hogyan helyezhetőek el dokumentumok, illetve hogyan helyezhetőek el XAdES formátumú aláírások ezen dokumentumokon, illetve a rajtuk elhelyezett aláírásokon. A gyakorlati alkalmazásokban általában nem elengedő a dokumentumokat önmagukban aláírni, hanem különféle metaadatokat is kell csatolni hozzájuk. Az e-akta specifikáció a Dublin Core Metadata Initiative által meghatározott, szabványos formátumú metaadatok csatolását támogatja, de más metaadatok csatolására is van lehetőség. [35]

Magyarországon az e-akta formátum de facto szabvánnyá vált, számos felhasználói közösség, illetve alkalmazás e-akta formátumú elektronikusan aláírt dokumentumokat kezel. (Lásd: 13.1. fejezet.)

Az e-akta egy XML fájl, amelyben a bináris elemek (pl. dokumentumok, tanúsítványok) base64 kódolással szerepelnek. Egy e-aktában *dokumentumok* helyezkedhetnek el, amelyekhez Dublin Core szerinti *metaadatok* kapcsolódhatnak, és a dokumentumokon XAdES *aláírások* vagy *időbélyegek* lehetnek. Az aláírás vagy időbélyeg vagy csak egyetlen dokumentumhoz, vagy pedig az aktában lévő összes dokumentumhoz kapcsolódik (ez utóbbi esetben *keretaláírásnak* vagy *keretidőbélyegnek* is nevezzük). A keretaláírások (és keretidőbélyegek) az aktában lévő összes dokumentumon, valamint a dokumentumokon lévő (nem keret-) aláírásokon és időbélyegeken helyezkednek el.

Például a következő struktúrát követheti egy e-akta:

```
<es:Dossier ... >
<es:DossierProfile>...</es:DossierProfile>
<es:Documents>
<es:Document>                          <!-- egy beillesztett dokumentum -->
<es:DocumentProfile>...</es:DocumentProfile>
<ds:Object>...</ds:Object>        <!-- a dokumentum base64 kódolással -->
<ds:Signature>...</ds:Signature>  <!-- aláírás a dokumentumon -->
<es:TimeStamp>...</es:TimeStamp>  <!-- időbélyeg a dokumentumon -->
</es:Document>
<ds:Signature>...</ds:Signature>      <!-- keretaláírás -->
<es:TimeStamp>...</es:TimeStamp>      <!-- keretidőbélyeg -->
</es:Documents>
```

Az automatizált feldolgozhatóság érdekében minden e-akta rendelkezik valamilyen *sémával*, amely az e-aktában lévő adatokkal kapcsolatban tartalmazhat megkötéseket. Az e-akta sémája például megkötheti, hogy a sémának megfelelő e-aktákban kizárólag meghatározott számú dokumentum szerepelhet, valamint megkötéseket tartalmazhat a dokumentumok címére, formátumára, illetve a hozzájuk kapcsolódó adatelemekre. Az e-akta sémája (amely egyben

egy XML séma) alapján automatizmus is könnyen tudja ellenőrizni, hogy egy e-akta teljesíti-e ezen követelményeket. Például ellenőrizni lehet, hogy egy aláírt beadvány tartalmazza-e a szükséges mellékleteket, illetve az automatizmus is könnyen meg tudja különböztetni a beadványhoz csatolt egyes dokumentumokat egymástól (pl. meg tudja állapítani, hogy melyik dokumentum a beadvány, és melyik a csatolt melléklet).

Az e-akta kiterjesztése `.es3`, tértivevény esetén `.et3`. Az e-akta IANA által bejegyzett[17] mime típusa `application/vnd.eszigno3+xml`, bár egyes régi alkalmazások még az `application/eszigno3+xml` mime típust használják.

Az e-akta specifikáció két részből áll, az első rész szövegesen, a második rész XML séma (az ún. *alapértelmezett e-akta séma*[18]) alapján mutatja be a formátumot. (Nem minden szükséges követelmény írható le XML séma segítségével, így a séma mellett a szöveges specifikáció is tartalmaz követelményeket, és e két rész együttesen tekinthető az e-akta formátum specifikációjának.)

Az e-akta formátum részletes specifikációja a `http://www.e-szigno.hu/?lap=e-akta` címen érhető el. [36]

6.4.2.2. PDF (Portable Document Format)

6.4.2.2.1. Miért PDF?

Az elektronikus papír dokumentumok kezelésének területén az Adobe által kifejlesztett PDF jelenti az egyik legelterjedtebb, szabványos megoldást. Ezért, ha egy dokumentumot hosszú távon, változatlan formában szeretnénk megőrizni, illetve azt szeretnénk, hogy azt bárki könnyen, ugyanolyan formában tudja megtekinteni, a PDF az egyik legkézenfekvőbb megoldás. Ebből kifolyólag az aláírt dokumentumok nagyon nagy hányada is PDF.

Megtehetjük, hogy a PDF-et egy konténer, például egy e-akta belsejében helyezzük el, és így írjuk alá. Ugyanakkor a PDF maga is tartalmazhat aláírást, így e belső, PDF aláírást is használhatjuk. Ekkor a PDF tölti be az aláírás-konténer szerepét is. Ennek nagy előnye, hogy ekkor sem a PDF kicsomagolásához és megjelenítéséhez, sem az aláírás ellenőrzéséhez nincsen szükség másik alkalmazásra, ezek mindegyike elvégezhető egy PDF megjelenítő, például az Acrobat Reader segítségével. Más formátumú dokumentumokat, például Word fájlokat nem szokás ilyen módon „fontos” feladatokra használni. A PDF – részben a fenti előnyök, részben a specifikáció nyíltsága, részben a formátumot támogató sok alkalmazás miatt – az egyik legkedveltebb aláírás-konténer formátum.

[17] `http://www.iana.org/assignments/media-types/application/vnd.eszigno3+xml`
[18] `https://www.microsec.hu/ds/e-szigno30.xsd`

6.4.2.2.2. Hagyományos PDF aláírás

A hagyományos PDF aláírás, amelyet az ISO 32000-1 szabvány is leír, egy beágyazott PKCS#7 formátumú aláírás-blokkot tartalmaz. E megoldás nagyon elterjedt, de erős korlátokkal rendelkezik. Ennek egyik oka, hogy a PKCS#7 aláíráson alapesetben nincsen időbélyeg, így a hosszú távú letagadhatatlanságuk problémás. Készültek olyan megoldások, amelyek a PKCS#7 aláírásokat CAdES aláírásokká, például időbélyeggel is ellátott, CAdES-T aláírásokká terjesztették ki. Itt erős korlátot jelentett, hogy a PDF specifikációból adódóan az aláírás beillesztésekor előre meg kellett mondani, hogy az aláírás mekkora helyet foglal el, és az aláírás kiterjesztésekor e méretkorlátot nemigen lehetett átlépni. Ebből kifolyólag a hagyományos PDF aláírásokat nem lehet archív aláírásként archiválni, mert az archív időbélyegekből álló lánc nem növekedhet akármeddig. Szintén problémát jelent, hogy a PKCS#7 aláírás alapesetben nem függ az aláíró tanúsítványának a lenyomatától, így egyes álláspontok szerint nem szerencsés aláírás-formátum. A PDF-ben lévő PKCS#7 aláírást úgy kellett értelmezni, hogy a teljes PDF-re vonatkozik, leszámítva magát az aláírást. Ez egyúttal azt is eredményezte, hogy az ilyen módon aláírt PDF-ben már semmilyen más elemet nem lehetett elhelyezni, így például másik aláírást sem, mert a következő aláírás automatikusan elrontotta az elsőt.

A PDF-re épülő megoldásokkal foglalkozó cégek különféle trükköket találtak ki e problémák megoldására vagy enyhítésére, így többek között XAdES vagy CAdES aláírások PDF-be történő illesztésére. Amíg ezek nem váltak a PDF hivatalosan, az Adobe által is támogatott részévé, addig nem jelentettek igazi megoldást, mert egymással sem voltak kompatibilisek, és az aláírást ellenőrző érintett fél Acrobat Readere sem kezelte őket.

6.4.2.2.3. PAdES (PDF Advanced Electronic Signature) aláírás

Az ETSI az Adobe közreműködésével 2009-re fejlesztette ki a PAdES (ETSI TS 102 778) specifikációt. A PAdES a [CX]AdES aláírások korrekt módon történő PDF-be illesztését írja le. Az elnevezése megtévesztő, ugyanis nem egy új, a XAdES és CAdES mellett már harmadik aláírás-blokk formátumot specifikál, hanem egy aláírás-konténer formátumot ír le. [60] A PAdES a következő részekből áll:

1. PAdES Overview – Áttekintés a PAdES keretrendszerről.
2. PAdES Basic – Alap aláírás-profil, amely az ISO 32000 szerinti, hagyományos PDF aláírásokat írja le. Ezzel a PDF hagyományos, PKCS#7-es aláírást az ETSI hivatalosan is „advanced electronic signature”-nek, azaz fokozott biztonságú aláírásnak ismerte el, annak ellenére, hogy ezen aláírás nem függ az aláíró tanúsítványának lenyomatától.
3. PAdES Enhanced – A -BES és -EPES profilok. E specifikáció írja le, hogyan lehet CAdES aláírást elhelyezni a PDF-eken.

4. PAdES Long Term – A PDF-en elhelyezett CAdES (azaz PAdES-3) aláírások hosszú távú archiválásának módját (-A) írja le.

5. PAdES for XML Content – A PDF-be XML tartalmat is lehet ágyazni, és ekkor van értelme XML aláírást használni. Ezért a specifikáció ötödik része azt írja le, hogy hogyan helyezhetünk el XAdES aláírást ezen XML tartalmon.

Összefoglalva, a PAdES specifikáció szerint CAdES aláírásokat lehet elhelyezni a PDF dokumentumokon. Ha a PDF-be XML tartalom kerül, az XML tartalmon lehet XAdES aláírást elhelyezni.

6.4.2.2.4. Látható aláírás

Egy papír alapú dokumentumon elhelyezett kézzel írott aláírás esetén nagyon sok információ leszűrhető abból, hogy az aláíró hol, a dokumentum melyik részén helyezte el az aláírást. A legtöbb dokumentum általában tartalmazza az aláírások helyét is. Ha egy dokumentumot különböző szerepkörök (például megbízó és megbízott) szerint is alá lehet írni, leginkább abból derül ki, hogy milyen szerepkörben írták alá, hogy hova (a megbízó vagy a megbízott nevéhez) került az aláírás. A PDF egy elektronikus papír formátum, így e fogalmak ott is értelmezhetőek.

PDF esetében léteznek ún. „látható aláírások”, ekkor az aláíró aláíráskor megadhatja, hogy az elektronikus aláírása hol jelenjen meg a dokumentumon. Ez természetesen semmilyen kapcsolatban nincsen azzal, hogy az aláírás hol helyezkedik el a PDF fájlban. Mindössze arról van szó, hogy az aláírás mellé kerül egy utasítás, hogy a PDF megjelenítő hol jelenítse meg az aláírást jelképező grafikát. E grafika lehet az aláíró beszkennelt kézzel írott aláírása, lehet egy stilizált aláírás, de az is lehet, hogy egy pecsét, vagy az aláíró adatai jelennek ott meg. E grafikát az aláíró adhatja meg.

Látható aláírások esetén lényeges, hogy a PDF megjelenítő és aláírás-ellenőrző alkalmazás ne a PDF felett jelenítse meg az aláírás ellenőrzésével kapcsolatos információkat. A PDF-ben különböző trükkök is elhelyezhetők, akár aktív tartalom is szerepelhet benne. Lényeges, hogy az aláírás ellenőrzésének eredménye és az aláíró kiléte valahol máshol, a grafikus aláírástól elkülönítve, például egy másik ablakban jelenjen meg. Nem szabad, hogy egy trükkös aláíratlan PDF fájl elhitethesse magáról, hogy azt valaki aláírta. [43]

6.4.2.3. Associated Signatures

Az Associated Signatures, más néven ETSI TS 102 918 az ETSI által kidolgozott aláírás-konténer formátum. A konténer lényegében egy ZIP fájl, amely dokumentumokat és hozzájuk kapcsolódó aláírásokat, időbélyegeket, illetve XML metaadatokat tartalmaz. [61]

Az aláírások vagy időbélyegek vagy egyes fájlokra, vagy – az e-aktában szereplő keretaláírás fogalmához hasonlóan – a teljes aktára vonatkoznak. A konténerben CAdES és XAdES formátumú aláírások helyezhetőek el.

Mivel a konténer egy ZIP fájl, a benne lévő aláírt dokumentum akár speciális alkalmazás nélkül is kicsomagolható és megnyitható. Másrészt az Associated Signatures illeszkedik egyéb ZIP-alapú formátumokhoz is, ilyen például az OpenOffice által is használt ODF (Open Document Format) fájlformátum. Így OpenOffice dokumentumon úgy helyezhető el XAdES aláírás, hogy az továbbra is szabványos, megnyitható OpenOffice dokumentum marad, de egyúttal ETSI TS 102 918 szerinti aláírás-konténerré is válik.

Könyvünk megjelenésekor az Associated Signatures még nagyon fiatal specifikáció, nemigen léteznek hozzá implementációk, de lehet, hogy a jövőben nagy szerep jut majd neki.

6.5. Aláírás ellenőrzése, befogadása

6.5.1. Mit értünk ellenőrzés alatt?

A következőket értjük elektronikus aláírás ellenőrzése alatt:

1. *Műszakilag érvényes-e az aláírás?* Ennek vizsgálatakor:
   a. *kriptográfiai ellenőrzést végzünk*, azaz megvizsgáljuk, hogy összetartozik-e az aláírás, az aláírt dokumentum és az aláíró nyilvános kulcsa;
   b. *PKI ellenőrzést végzünk*, azaz megvizsgáljuk, hogy az adott nyilvános kulcs valóban az aláíró kizárólagos birtokában volt-e az aláírás pillanatában;
2. *Az adott folyamatban elfogadható-e az aláírás az adott célra?* Ennek során figyelembe vehetjük az aláírás biztonsági szintjét, az aláíró szerepét, kilétét, az aláírási szabályzatot, illetve, hogy a műszaki ellenőrzés alapján mennyire lehetünk biztosak az aláírás érvényességében.
3. *Megáll-e, vagy megállna-e az aláírás bíróság előtt?* Ritka, hogy egy elektronikus aláírás bíróság elé kerül. Ennek ellenére, mind az elektronikus, mind a kézzel írott aláírásnak az az értelme, hogy szükség esetén megállja a helyét bíróság előtt. Vita esetén *a bíróságot az érdekli, hogy az aláíró valóban aláírta-e az adott dokumentumot*, és mind a kriptográfiai ellenőrzést, mind a PKI ellenőrzést, mind az aláírás üzleti folyamatban betöltött szerepét ezt alátámasztó (vagy megcáfoló) bizonyítékként kezeli, de ezen túl egyéb tényezőket, egyéb bizonyítékokat is figyelembe vehet.

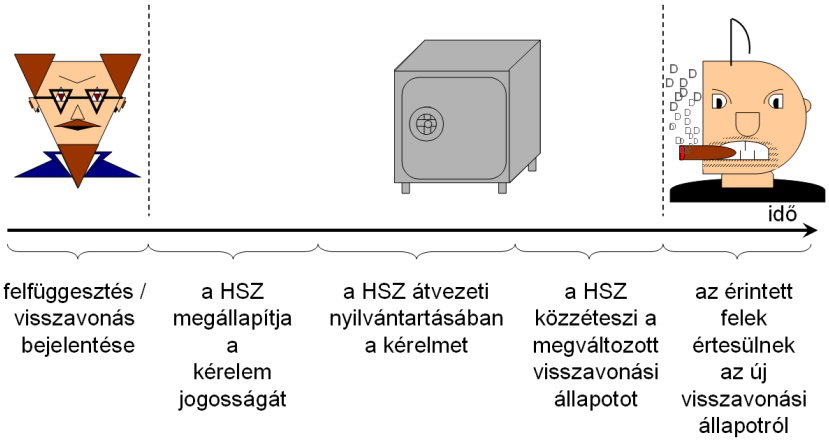
Az aláírás érvényességéről általában egy aláírás-ellenőrző alkalmazás segítségével győződünk meg. A műszaki ellenőrzési lépéseket általában ez az alkalmazás végzi el, és előfordulhat, hogy a későbbi lépések során is segítséget nyújt.

6.5.2. Mennyire egyértelmű az ellenőrzés?

Az aláírás *kriptográfiai ellenőrzése* során a (kriptográfiai értelemben vett) aláírást kódoljuk az aláíró nyilvános kulcsával, és megvizsgáljuk, hogy a kapott eredmény megegyezik-e az aláírt információ lenyomatával. Ez egy egzakt művelet: Ha a két lenyomat megegyezik, akkor az aláírás kriptográfiailag érvényes, és az aláírt dokumentum nem változott meg az aláírás óta. Ha nem egyeznek meg, akkor nem tartozik össze az aláírás az aláírt dokumentummal és az adott nyilvános kulccsal. Ezt okozhatja az, hogy megsérült a dokumentum, vagy szándékosan megváltoztatták, esetleg nem a megfelelő dokumentumból képeztünk lenyomatot, vagy rossz kulcsot használtunk az aláírás ellenőrzéséhez, de az is lehet, hogy az aláírásnak valóban semmi köze nincs az adott dokumentumhoz. Ha a kriptográfiai ellenőrzés sikertelen, akkor az aláírással baj van. Ekkor nem állapítható meg, hogy csak egyetlen bit változott-e meg, vagy az aláírás egészen más dokumentumra vonatkozik.

A *PKI ellenőrzés* során egy tanúsítványláncot (5. fejezet) keresünk egy megbízható gyökértanúsítványig, és megbizonyosodunk róla, hogy e tanúsítványlánc minden egyes eleme érvényes volt az aláírás pillanatában. Megvizsgáljuk, hogy az aláírás időpontja belül esik-e a lánc minden egyes tanúsítványának érvényességi idején, valamint nem voltak-e ezen tanúsítványok az aláírás időpontjában felfüggesztve vagy visszavonva. A PKI ellenőrzés eredménye számos tényezőtől függ. Kérdés, hogy mely gyökeret tekintünk megbízhatónak, milyen módon keressük a tanúsítványláncot, hogyan vizsgáljuk a visszavonási állapotot stb. Ráadásul egy tanúsítvány visszavonási állapota számos emberi tényezőtől is függ: észrevette-e az aláíró a kulcs kompromittálódását, időben és megfelelő módon értesítette-e a hitelesítés-szolgáltatót, a hitelesítés-szolgáltató mennyire gyorsan tette közzé a megváltozott visszavonási állapotot, és mennyire gyorsan jutott el az új állapot az aláírást ellenőrző érintett félhez (lásd: 6.20. ábra) stb. Ezek egy része emberi folyamat, így kevésbé megbízható, és kevésbé gyors, mint a számítógéppel végezhető folyamatok. Ez esetben közel nem beszélhetünk 100%-os biztonságról, legfeljebb a felelősségi viszonyokat tisztázhatjuk (például rögzíthetjük, hogy ha az aláíró nem veszi észre a kulcs kompromittálódását, azért a hitelesítés-szolgáltató nem felel). A PKI ellenőrzés legfeljebb akkor tekinthető objektív, egzakt műveletnek, ha rögzítjük, hogy pontosan milyen paraméterek (gyökerek, visszavonás-ellenőrzés módja stb.) szerint végezzük. Az ellenőrzés rendjét aláírási szabályzatban (6.8. fejezet) szokás rögzíteni.

Felhasználható-e egy adott aláírás egy adott célra? E kérdésre akkor adható egzakt válasz, ha – például aláírási szabályzatban – rögzítve van, hogy pontosan milyen aláírások használhatóak fel az adott célra. Szükség van-e minősített aláírásra? Milyen kivárási időt (6.5.4.4. fejezet) alkalmazunk? Mekkora felelősséget vállal az adott tanúsítványért az őt kibocsátó hitelesítés-szolgáltató? Milyen módon kell meggyőződni az aláíró szerepéről, jogosultságáról? A PKI ellenőrzés alapján mennyire vagyunk biztosak az aláírás érvényességében? A gyakori esetek aláírási szabályzatban kezelhetőek, az egyedi esetekben mérlegelni kell. E kérdéskörben számos emberi és szervezeti tényező is szerepet játszik, általában nem kezelhető teljesen objektíven.



6.20. ábra. **Felfüggesztési vagy visszavonási kérelem feldolgozása**

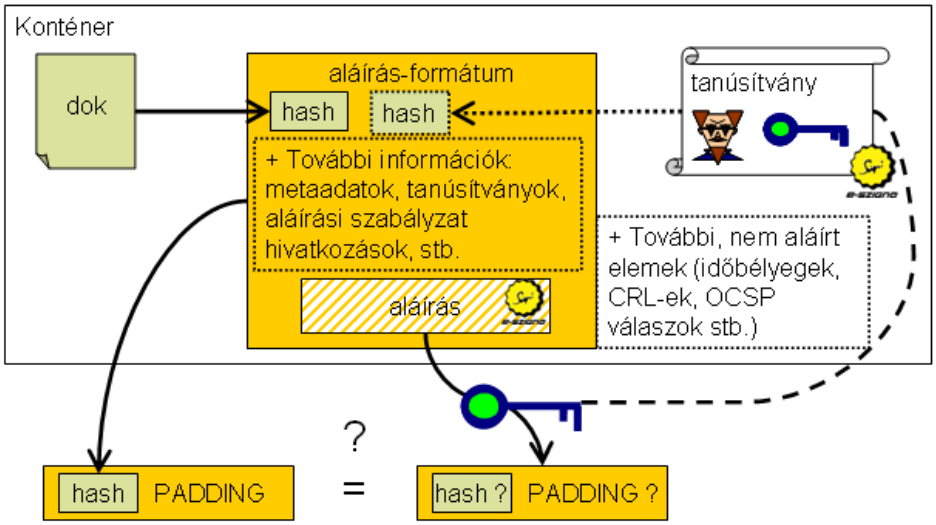
Megáll-e, vagy megállna-e az aláírás bíróság előtt? E kérdéskör végképp nem kezelhető objektív módon, nehéz megmondani, hogy egy bíróság pontosan mit fog dönteni egy adott aláírással kapcsolatban. Lényeg, hogy egy aláírás műszaki érvényességéből nem következik, hogy az garantáltan megállna bíróság előtt (az aláírás jogilag nem „letagadhatatlan", csak bizonyító erő kapcsolódik hozzá), és ha egy aláírás műszakilag érvénytelen, abból sem következik, hogy a bíróság az feltétlenül el fogja utasítani.

Kihangsúlyozzuk, hogy *az aláírást ellenőrző érintett fél úgy ellenőrzi az aláírást, ahogy kedve tartja.* Bárki ellenőrizhet aláírást. Az aláírást ellenőrző érintett fél nem biztos, hogy bármilyen viszonyban van akár az aláíróval, akár a hitelesítés-szolgáltatóval. Nagyon kevés szabály vonatkozik az érintett félre, általánosságban csak az mondható el, hogy ha egy érintett fél nem kellően gondosan jár el egy aláírás ellenőrzése során, akkor ő felel az ebből eredő károkért. Ellenőrzéskor célszerű figyelembe vennie az adott aláírásra vonatkozó aláírási szabályzatot, az aláíró tanúsítványára vonatkozó hitelesítési rendben leírtakat, a hitelesítés-szolgáltató szabályzatait, illetve a nemzetközi ajánlásokat az aláírás ellenőrzésére (ez utóbbiak egy részét az aláírás-ellenőrző alkalmazása végzi el helyette).

6.5.3. Kriptográfiai ellenőrzés

Az aláírás kriptográfiai ellenőrzése során azt vizsgáljuk, összetartozik-e az aláírás az aláírt dokumentummal és az aláíró nyilvános kulcsával. Ekkor a (kriptográfiai értelemben vett) aláírást kódoljuk az aláíró nyilvános kulcsával, és megvizsgáljuk, hogy a kapott eredmény megegyezik-e az aláírt információ lenyomatával.

Az aláírás formátumáról szóló alfejezetben (6.4. fejezet) láthattuk, hogy aláíráskor általában nem közvetlenül az aláírandó dokumentum lenyomatát kódoljuk a magánkulccsal. Az aláírt dokumentum lenyomatát egy aláírás-blokkban helyezzük el (lásd: 6.11. ábra), és ezen blokk (aláírt elemeinek) lenyomatát kódolja az aláíró a magánkulcsával.



6.21. ábra. **Kriptográfiai ellenőrzés – a dokumentum, az aláírás és a nyilvános kulcs összetartozása**

Így az ellenőrzés is ennek megfelelően zajlik le. (Lásd: 6.21. ábra.) Az egyes aláírt dokumentumokból (XAdES vagy XMLDSIG aláírás esetén az aláírás-blokkban szereplő `xmldsig:Reference` elem által hivatkozott tartalmakból) lenyomatot képzünk, és ellenőrizzük, hogy ezen lenyomatok szerepelnek-e az aláírás-blokkban. Az aláírás blokkban szereplő kriptográfiai értelemben vett aláírást kódoljuk az aláíró nyilvános kulcsával, a kapott eredményről eltávolítjuk a paddinget, és megvizsgáljuk, hogy az eredmény megegyezik-e az aláírás-blokk aláírt elemeinek lenyomatával.

6.5.4. PKI ellenőrzés

A kriptográfiai ellenőrzés segítségével arról bizonyosodhatunk meg, hogy az aláírás egy adott nyilvános kulcshoz tartozó magánkulccsal készült. *A PKI ellenőrzés során azt vizsgáljuk, hogy tudjuk-e*[19]*, hogy az adott kulcspár kihez tartozik, és a magánkulcs a fellelhető bizonyítékok alapján ezen fél – az aláíró – kizárólagos birtokában volt-e az aláírás pillanatában.* E vizsgálatot többféleképpen is elvégezhetjük, így a vizsgálat eredménye akkor lesz egyértelmű, ha tudjuk, hogy milyen módon, például egy aláírási szabályzat szerint végeztük.

Az aláírás ellenőrzésére a CWA 14171 fogalmaz meg ajánlásokat. [31] Az aláíró tanúsítványának ellenőrzése tekintetében az RFC 5280 specifikációban[20] leírt algoritmust hivatkozza meg. [152] Ez nem szerencsés.

[19] A PKI ellenőrzés során nem vagyunk rá kíváncsiak, hogy kihez tartozik a kulcspár. Arra vagyunk kíváncsiak, hogy vita esetén tudjuk-e bizonyítani, hogy a kulcspár kihez tartozik, azaz van-e az illetőnek tanúsítványa. Előfordulhat, hogy az aláírónak álneves tanúsítványa van, így nem tudjuk, ki az illető, de vita esetén a hitelesítés-szolgáltató segítségével rá tudjuk bizonyítani, hogy az aláírást ő készítette.

[20] Az 3280 specifikációt hivatkozza, de azt időközben felváltotta az RFC 5280. [143], [152]

A CWA 14171 európai specifikáció, és *elektronikus aláírás* – potenciálisan minősített elektronikus aláírás – ellenőrzéséről szól. Az EU-s jog szerint a minősített aláírás a kézzel írott aláírással egyenértékű, így különösen fontos, hogy ha valaki egy aláírást érvényesnek tekint, akkor mások, máskor ugyanúgy érvényesnek tekinthessék. Ezzel szemben az RFC 5280 nemzetközi specifikáció, amely egy *tanúsítvány* ellenőrzésére ad algoritmust. [152] Ha titkosító tanúsítványt vagy autentikációs tanúsítványt ellenőrzünk, arra vagyunk kíváncsiak, hogy a tanúsítvány érvényes-e *az ellenőrzés pillanatában*. Ugyanez lehet a helyzet, ha ún. pillanatnyi digitális aláírásokat ellenőrzünk, ahol pl. az a kérdés, hogy megsérült-e egy Interneten átküldött fájl; ha az ellenőrzés pillanatában érvényes rajta az aláírás, elfogadjuk a fájlt, és eldobjuk az aláírást, mert soha többet nem lesz rá szükségünk. Ezzel szemben, a CWA 14171 az EU-s, „letagadhatatlan" aláírásokkal foglalkozik, ahol az aláírást ellenőrzés után eltesszük, és esetleg évekkel később egy bizonyítási eljárás során használjuk fel. [31]

Elektronikus aláírás ellenőrzése esetén nem az a kérdés, hogy az aláíró tanúsítványa érvényes-e az ellenőrzés pillanatában – ez általában lényegtelen. Elektronikus aláírás ellenőrzése esetén az a kérdés, hogy az aláíró tanúsítványa érvényes volt-e az aláírás készítésének pillanatában.

> ***6.9. Példa:*** *Alajos (elektronikusan) aláír egy váltót, amelyben igazolja, hogy kölcsön kért Bendegúztól 1 millió forintot. Alajos később nem adja meg a tartozását, ezért Bendegúz bírósághoz fordul. Csakhogy ekkor Alajos tanúsítványa már nem érvényes, mert időközben lejárt. Ez nem baj, mert a tanúsítvány az aláírás pillanatában még érvényes volt, ha ez igazolható, akkor az aláírás érvényes.*

A CWA 14171 részletesen leírja, hogy az ellenőrzést az aláírás időpontjára vonatkozóan kell végezni, viszont olyan algoritmust hivatkozik meg a PKI ellenőrzéssel kapcsolatban, amely csak a jelen időpontra, az ellenőrzés időpontjára vonatkozóan ellenőriz. [31] Más specifikációk, például az ETSI TS 101 903 (XAdES) és az ETSI TS 101 733 (CAdES), további felvilágosítást adnak arról, hogy hogyan kell múltbéli időpontra vonatkozóan ellenőrizni, de ők sem kezelik teljeskörűen a problémát. [51], [49], [150] Ebből adódóan *nincs olyan specifikáció, amely pontosan megmondaná, hogy hogyan célszerű elektronikus aláírást ellenőrizni.*

A továbbiakban a fent hivatkozott specifikációkban szereplő, illetve a belőlük levezethető főbb elveket ismertetjük.

6.5.4.1. Bizonyítékok alapján

Amikor aláírást *ellenőrzünk*, megpróbáljuk levezetni, bebizonyítani az aláírás érvényességét valamilyen követelményrendszer vagy szabályrendszer (pl. aláírási szabályzat) szerint. A bizonyítás során megbízható felek kulcsaira próbáljuk meg visszavezetni az aláírás érvényességét. A bizonyítás *bizonyítékokra*, lehetőség szerint aláírt PKI objektumokra épül, és e bizonyítékok alapján:

1. *Érvényesnek* tekintjük az aláírást, ha az adott aláírási szabályzat szerint sikerül bebizonyítanunk az érvényességét.

2. *Érvénytelennek* tekintjük az aláírást, ha arra utaló bizonyítékot találunk, hogy az adott aláírási szabályzat szerint az aláírás érvénytelen.

3. Attól függően, hogy az ellenőrzést mikor végezzük, előfordulhat, hogy nem sikerül bebizonyítanunk az aláírás érvényességét, de nem találunk arra utaló bizonyítékot, hogy az aláírás érvénytelen lenne. Ha úgy látjuk, hogy a szükséges bizonyítékot egy későbbi időpontban majd be lehet gyűjteni, akkor az ellenőrzés *befejezetlen*.

Az ellenőrzés során csak érvényes bizonyítékokat használhatunk fel, ezért minden felhasznált bizonyítékot is *ellenőrzünk*. Amennyiben aláírt PKI objektumokat (tanúsítványokat, időbélyegeket, visszavonási listákat (CRL-eket), OCSP válaszokat stb.) használunk bizonyítékként, a rajtuk lévő aláírás érvényességét is ellenőriznünk kell a fentieknek megfelelően. Az aláíratlan, nem PKI objektum bizonyítékokat úgyszintén ellenőrizzük egyéb, pl. out-of-band módszerekkel.

> **Megjegyzés:** Nemcsak időbélyeggel igazolhatjuk, hogy egy bitsorozat már létezett egy adott időpontban. Például ha egy megbízható és biztonságos rendszer naplói szerint az adott bitsorozat t időpontban megjelent a rendszerben, akkor e naplót is tekinthetjük bizonyítéknak. Hasonlóképpen, ha az adathordozót t időpontban letétbe helyezzük egy megbízható félnél, és e megbízható fél igazolja, hogy azóta nem fért hozzá senki, azt is tekinthetjük bizonyítéknak. (A már nem hatályos, 7/2005. IHM rendeletben szerepelt egy ilyen alternatív megoldás az elektronikus archiválásra.) Elvileg akár tanúkkal is igazolhatjuk egy bitsorozat létezését, de ez igen körülményes lehet.
>
> Az RFC 5380 szerint a tanúsítványláncban szereplő minden egyes tanúsítvány visszavonási állapotát ellenőrizni kell. Ezen ellenőrzés nem feltétlenül PKI alapon – CRL-ek vagy OCSP válaszok szerint – történik, elvégezhető out-of-band módszerekkel is. Előfordulhat, hogy más úton ellenőrizzük a visszavonási állapotot, például úgy, hogy a hitelesítés-szolgáltató hirdetést tesz közzé a TV-ben vagy egy napilapban egy tanúsítvány érvénytelenségéről. A végfelhasználói tanúsítványok ilyen módon történő ellenőrzése meglehetősen körülményes volna, de köztes[21] szolgáltatói tanúsítványok esetén néhol a valóságban is alkalmaznak out-of-band megoldást. (Igaz, akkor általában nem definiálják, hogy mit jelent az out-of-band ellenőrzés, hanem azt mondják, hogy az adott tanúsítvány „úgyis érvényes”, és „majd valaki szól, ha nem”.)

[21] A gyökértanúsítványokról itt nem beszélünk. Egyrészt, mert a gyökértanúsítvány nem része a tanúsítványláncnak, másrészt, mert a gyökértanúsítványt kizárólag out-of-band módszerrel lehet ellenőrizni.

Az aláíratlan bizonyítékokat általában nem lehet szabványos módon ellenőrizni, így ezek használata erősen körülményes. Annak ellenére, hogy így a PKI ellenőrzés látszólag egyszerűbb, vagy jelentős terhet ró az ellenőrző félre, vagy az ellenőrző nem végez out-of-band ellenőrzést, és így elmaradnak az ellenőrzés egyes kritikus lépései. Összességében a nem PKI alapon történő ellenőrzés a legköltségesebb módja az ellenőrzésnek, e megoldás általában azt is kizárja, hogy az ellenőrzést gép végezze el.

Egyedül a megbízható gyökértanúsítvány olyan, amit kizárólag out-of-band módon lehet ellenőrizni. Célszerű arra törekedni, hogy a gyökereken kívül minden más bizonyítékot szabványos, PKI alapú módon ellenőrizhessünk, és az ellenőrző félnek ne kelljen vele törődnie. Összességben így tehető hatékonnyá és olcsóvá az ellenőrzés.

Megjegyezzük, hogy műszakilag és jogilag eltérő gondolatmenet szerint építünk az aláírásra. Műszakilag úgy gondolkozunk, hogy az aláírást mindaddig nem tekintjük érvényesnek, amíg be nem bizonyítjuk az érvényességét. (A műszaki bizonyítás alatt egy adott követelményrendszer szerint, bizonyítékok egy adott halmaza alapján végzett matematikai levezetést értünk.) Jogilag úgy gondolkozunk, hogy ha egy minősített aláírás műszakilag érvényes[22], akkor jogilag mindaddig érvényesnek tekintjük, amíg valaki be nem bizonyítja az ellenkezőjét. (Lehet, hogy például valaki tanúkkal igazolja, hogy őt fegyverrel kényszerítették, hogy írja alá az adott dokumentumot. Így hiába jött létre érvényes aláírás, a bíróság dönthet úgy, hogy azt nem veszi figyelembe.)

6.5.4.2. Rekurzív algoritmus

Egy aláírás PKI ellenőrzése a következő rekurzív algoritmussal írható le:

1. Határozzuk meg, hogy milyen időpontra nézve kell elvégezni az ellenőrzést! (Az időpontot, amelyre nézve az adott aláírást ellenőrizzük, a szakirodalom *control time*-nak nevezi.) Olyan $t_{control}$ időpontot keresünk, amelyben az aláírás már biztosan létezett.

   a. Ha az aláírást *érvényes* időbélyeg védi, akkor az ellenőrzést az időbélyegben szereplő időpontra nézve végezzük. Az időbélyeg érvényességéről meg kell győződnünk, ugyanezen algoritmus szerint ($\rightarrow$ rekurzió).

   b. Ha más, megbízható bizonyíték (pl. biztonságos naplófájl) alapján biztosak vagyunk benne, hogy az aláírás egy adott időpontban már létezett, akkor az ezen időpontra nézve végezzük az ellenőrzést. (Ne feledjük, ezen bizonyítékot is ellenőrizni kell, csak ezen ellenőrzés jellemzően nem PKI alapon történik.)

[22] Az Eat. úgy fogalmaz, hogy „ha az aláírás érvényességének ellenőrzéséből más nem következik”. [180]

c. Ha a fenti információk egyike sem áll rendelkezésre, akkor csak annyit tudhatunk, hogy az aláírás „most”, azaz az ellenőrzés pillanatában biztosan létezik, így az ellenőrzés pillanatára nézve végezzük az ellenőrzést.

2. Építsük fel a tanúsítványláncot az adott $t_{control}$ időpontra nézve! Követeljük meg, hogy a lánc minden egyes tanúsítványának érvényességi ideje foglalja magába $t_{control}$ időpontot.

3. Ellenőrizzük az aláírást a láncban szereplő minden egyes tanúsítványon ($\rightarrow$ rekurzió).

4. Bizonyítékokat (CRL-eket, OCSP válaszokat) gyűjtünk a láncban szereplő minden egyes tanúsítvány visszavonási állapotára, és ellenőrizzük, hogy ezen bizonyítékokon érvényes-e az aláírás ($\rightarrow$ rekurzió).

5. Az egyes bizonyítékokkal kapcsolatban kivárási időt (6.5.4.4. fejezet) érvényesítünk, ha az aláírási szabályzat megköveteli.

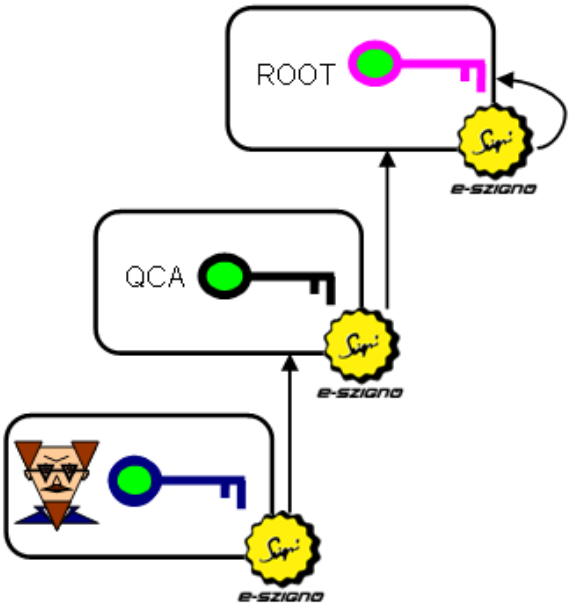
Az aláírási szabályzat eltérő követelményeket támaszthat a kivárási időre a végfelhasználói tanúsítványokkal és a szolgáltatói tanúsítványokkal kapcsolatban, a lánc különböző szintjein lévő szolgáltatói tanúsítványokkal kapcsolatban, a hitelesítés-szolgáltatók és az időbélyegzés-szolgáltatók tanúsítványaival kapcsolatban stb.

Előfordulhat, hogy egy adott tanúsítványlánc vagy egy adott $t_{control}$ időpont vonatkozásában nem sikerül ellenőrizni az aláírást. Vegyük figyelembe, hogy esetleg más fél más lánc és más control time időpont vonatkozásában ellenőriz majd, és lehet, hogy ő érvényesnek tekinti majd az aláírást. Akkor célszerű elutasítani az aláírást, ha minden lehetséges láncot, és minden szóba jöhető időpontot megvizsgáltunk.

6.5.4.3. Tanúsítványlánc keresése

Aláírás ellenőrzésekor egy vagy több megbízható kulcsra – megbízható gyökértanúsítványra – vezetjük vissza az aláírás érvényességét. Olyan, n tanúsítványból álló ún. tanúsítványláncot (5. fejezet) keresünk, amely megfelel a következő követelményeknek:

- A lánc minden i. elemére, igaz, hogy az $i+1$. tanúsítvány az i. tanúsítvány kibocsátójának tekinthető.
- Az 1. tanúsítvány kibocsátójának egy megbízható gyökértanúsítvány tekinthető.
- Az n. tanúsítvány az „aláíró” tanúsítványa, azaz a benne szereplő nyilvános kulcs a kriptográfiai ellenőrzés (6.5.3. fejezet) szerint összetartozik az aláírással és az aláírt dokumentummal.
- A láncban szereplő összes tanúsítvány *érvényes* volt azon $t_{control}$ időpontban, amelyre nézve az ellenőrzést végezzük.



6.22. ábra. **Megvizsgáljuk, hogy az aláírás pillanata a lánc minden elemének érvényességi idején belül van-e. Ezen kívül megvizsgáljuk a lánc minden egyes elemének a visszavonási állapotát.**

A megbízható gyökértanúsítvány nem része a tanúsítványláncnak. A gyökér érvényességét PKI alapon nem lehet, nem tudjuk ellenőrizni, a megbízható gyökeret out-of-band módszerek segítségével lehet beszerezni, és csak így lehet ellenőrizni is. (Lásd: 4.1.4. fejezet.) A láncban szereplő első tanúsítványt – tegyük fel, hogy ez nem rögtön az aláíró, hanem egy köztes hitelesítés-szolgáltató tanúsítványa – a gyökér alapján ellenőrizhetjük. A gyökérben megbízunk, és e tanúsítvány kibocsátásával a gyökér azt állította, hogy ezen köztes szolgáltatói tanúsítványban is megbízhatunk (feltéve, hogy az *érvényes*). A soron következő tanúsítványt mindig az őt megelőző tanúsítvány alapján ellenőrizhetjük, és minden tanúsítvány egy-egy szinttel tovább delegálja a bizalmat, amit a gyökérre alapozunk. Végül eljutunk ahhoz a tanúsítványhoz, amelyet éppen ellenőrzünk, és amely azon nyilvános kulcsot tartalmazza, amely kriptográfiailag összetartozik a kérdéses aláírással[23], illetve az aláírt dokumentummal. (Lásd: 6.22. ábra.)

Az RFC 4158 egy algoritmust ír le, amely szerint megkereshetjük a tanúsítványláncot. [147] Ne feledjük, hogy az aláírás akkor érvényes, ha van olyan lánc, és van olyan időpont, amely szerint érvényes. Előfordulhat, hogy találunk egy láncot, amely nem egy megbízható gyökértanúsítványban ér véget. Előfordulhat, hogy találunk egy láncot, amelynek valamely eleme lejárt, vagy amelyet egy adott időpontban már visszavontak. Az is előfordulhat,

[23]Ne feledjük, hogy az itt leírtakat nemcsak a végfelhasználó által készített aláírások ellenőrzésénél, hanem a tanúsítványokon, CRL-eken, időbélyegeken stb. elhelyezett aláírások ellenőrzésénél is alkalmazhatjuk.

hogy több megfelelő láncot találunk. Az RFC 4158 szerint az összes lehetséges[24] láncot célszerű végigpróbálnunk, és csak akkor tekintendő érvénytelennek az aláírás, ha egyik sem elfogadható.

> ***6.10. Példa:*** *Nem lenne szerencsés, ha egy alkalmazás talál egy láncot, és mivel a láncban szereplő valamely tanúsítvány érvénytelen, egyből elutasítja az aláírást. Előfordulhat, hogy egy másik alkalmazás más módon keres, egy másik láncot talál, és aszerint érvényesnek tekintené ugyanazt az aláírást. Ezért célszerű az összes lehetséges lánc alapján döntenie az alkalmazásnak.*

Az aláírás-ellenőrző alkalmazás mindenképpen ismeri a tanúsítványt, amelyet éppen ellenőriz, és ismeri a megbízható gyökértanúsítványok egy körét. Ezen kívül az aláírás-ellenőrző alkalmazás az *általa ismert* tanúsítványok között keresi a tanúsítványláncot. Az alkalmazás a következő köztes szolgáltatói tanúsítványokat ismerheti:

- Azon tanúsítványokat, amelyeket csatoltak az aláíráshoz. Az aláíráshoz – pl. a XAdES aláírásokhoz – lehet csatolni a tanúsítványláncot, és az aláírás-ellenőrző alkalmazás ellenőrzéskor figyelembe veheti a csatolt láncot, illetve a csatolt tanúsítványokat. Ugyanakkor nem kötelező e lánc alapján végeznie az ellenőrzést.

  > ***6.11. Példa:*** *Aláíráskor az aláíró egy α láncot csatolt az aláírásához. Az α lánchoz tartozó X gyökér időközben lejárt, a hitelesítés-szolgáltató az Y gyökeret használja azóta. A szolgáltató Y gyökere felülhitelesítette a régi X gyökerét, ehhez egy X' ún. rollover tanúsítványt bocsátott ki. Az ellenőrző elfogadhatja az aláírást; akár úgy, hogy olyan időpontra ellenőriz, amikor az X gyökér még érvényes volt; akár úgy, hogy az X' tanúsítvánnyal kiegészített α láncot használja, amely az Y gyökér szerint helyes lánc.*
  >
  > *Szintén előfordulhat, hogy az aláíró a Z gyökér szerint írta alá a dokumentumot, de az ellenőrző a W gyökér szerint ellenőrzi, mert az ő aláírási szabályzata szerint csak a W gyökeret fogadhatja el, a Z gyökeret pedig nem.*

- Azon tanúsítványokat, amelyeket az ellenőrzés során szerez be.

  > ***6.12. Példa:*** *Az aláírás-ellenőrző alkalmazás az X tanúsítványt ellenőrzi, de nem ismer olyan – sem köztes, sem gyökér – tanúsítványt, amelyet az X kibocsátójának tekinthetne. Ekkor dönthet úgy, hogy elutasítja az aláírást. Ugyanakkor megnézheti, hogy az X tanúsítvány tartalmaz-e hivatkozást a kibocsátó tanúsítványának elérhetőségére. A tanúsítvány*

[24] Leszámítva pl. a „hurkokat” tartalmazó láncokat stb, mert az RFC 5280 szerint egy láncban egy tanúsítvány csak egyszer szerepelhet.

authority information access mezejében szerepelhet egy olyan URL, amelyen a kibocsátó tanúsítvány elérhető. Ezen URL-ről letöltheti a kibocsátó Y tanúsítványát, és felhasználhatja azt az ellenőrzés során. Ha az Y tanúsítványt ellenőrizni tudja a Z gyökér szerint, akkor az (X, Y) lánc a Z gyökér szerint érvényes láncnak tekinthető.

- Azon tanúsítványokat, amelyek az alkalmazás tanúsítványtárában szerepelnek. Például a Windows rendelkezik olyan tanúsítványtárral, amelyben köztes tanúsítványok is szerepelnek. Ha a Windows új köztes tanúsítvánnyal találkozik, az bekerül a tanúsítványtárba, így a későbbi ellenőrzések során is felhasználhatja azt.

Az aláírási szabályzatban célszerű rögzíteni, hogy az aláírás-ellenőrző alkalmazás milyen köztes tanúsítványok alapján végezze az ellenőrzést, illetve felhasználhat-e egyéb tanúsítványokat az ellenőrzés során.

6.5.4.4. Visszavonási állapot és kivárási idő

6.5.4.4.1. Aláírói tanúsítvány visszavonási állapotának ellenőrzése

A nyilvános kulcsú infrastruktúra szereplői nem tudnak tökéletesen vigyázni a magánkulcsukra, ezért a tanúsítványok csak véges hosszú ideig érvényesek. Előfordulhat, hogy egy tanúsítványt még a lejárata előtt, soron kívül érvénytelenné kell tenni, vissza kell vonni. Ezért, ha ellenőrzünk egy aláírást, ellenőrizzük a hozzá tartozó tanúsítványláncban szereplő összes tanúsítvány visszavonási állapotát. A visszavonási állapotot jellemzően a hitelesítés-szolgáltató által kibocsátott és közzétett, aláírt visszavonási információk (CRL-ek és OCSP válasz-ok) alapján ellenőrizzük. (Lásd: 4.1.5. fejezet.)

A visszavonási állapotot azon $t_{control}$ időpontra nézve vizsgáljuk, amelyre nézve az adott aláírást ellenőrizzük. Arra keresünk bizonyítékot, hogy a kérdéses tanúsítványok az aláírás időpontjában érvényesek voltak. Ha titkosító tanúsítványt vagy autentikációs tanúsítványt ellenőrzünk, az ellenőrzést tipikusan az ellenőrzés időpontjára nézve végezzük ($t_{control} = most$), és ebből kifolyólag a legfrissebb elérhető visszavonási információra van szükségünk. Ha aláírást ellenőrzünk, akkor az ellenőrzést jellemzően múltbéli időpontra nézve végezzük, és ebből eredően körültekintőbben is eljárhatunk.

Az ellenőrzés során a következő forrásból származó visszavonási információkat használhatjuk fel:

- Az aláíráshoz csatolt visszavonási információkat. A XAdES-C és afeletti aláírásokhoz csatolhatunk CRL-eket és OCSP válaszokat. Az aláírás ellenőrzésekor ezen információkat is használhatjuk, de más információkat is figyelembe vehetünk.

- Az ellenőrzés során beszerzett visszavonási információkat. Az aláírás-ellenőrző alkalmazás az ellenőrzés során letölthet CRL-eket vagy lekérdezhet OCSP válaszokat.
- Minden olyan visszavonási információt, amelyről az aláírás-ellenőrző alkalmazás tud.
- Egyéb visszavonási információkat.

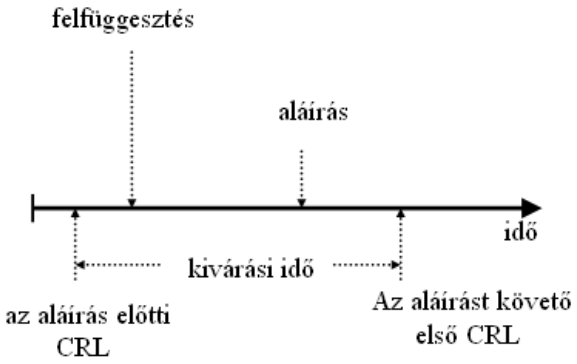
> ***6.13. Példa:*** *Alajost hétfőn halálra gázolja a HÉV. Bendegúz látta a balesetet, Alajos a szeme láttára halt meg. Szerdán Bendegúz kap egy aláírt dokumentumot, amelyen Alajos szerdai aláírása szerepel. A hitelesítés-szolgáltató visszavonási listája szerint Alajos tanúsítványa még érvényes (mert Alajos rokonsága még nem értesítette a hitelesítés-szolgáltatót). Bendegúz tudja, hogy Alajos nem írhatta alá a dokumentumot, ezért nem fogadja el az aláírást.*

> ***6.14. Példa:*** *Manfréd februárban betör az X hitelesítés-szolgáltatóhoz, és megszerzi a szolgáltató egy köztes hitelesítő egységének a magánkulcsát. Olyan ügyesen dolgozik, hogy a szolgáltató hosszú ideig nem veszi észre a betörést. Májusban kiderül, hogy a szolgáltatói kulcs valószínűleg februárban kompromittálódott, és az X hitelesítés-szolgáltató újsághirdetésben értesíti erről az érintett feleket.*
>
> *Bendegúz augusztusban kap egy aláírást, amely egy, az X hitelesítés-szolgáltató által kibocsátott tanúsítvány alapján ellenőrizhető. XAdES-A aláírásról van szó, a rajta lévő időbélyeg szerint márciusban jött létre, és az aláíráshoz csatolva ott szerepelnek az aláírás érvényességét igazoló CRL-ek. Bendegúz tudja, hogy márciusban már kompromittálódott az X hitelesítés-szolgáltató kérdéses kulcsa (csak ekkor e tény még nem volt ismert), így elutasítja az aláírást.*

6.5.4.4.2. A kivárási idő és annak összetevői

Az aláírás ellenőrzésének pillanatában még nem biztos, hogy rendelkezésre áll olyan bizonyíték, amely az aláírási szabályzatnak megfelelően igazolja a kérdéses tanúsítvány érvényességét. Aláírás ellenőrzése esetén előfordulhat, hogy nem elégszünk meg a legfrissebb elérhető visszavonási információval, de a frissebb visszavonási információ még nem szerezhető be. A *kivárási idő* az az időtartam, amit az ellenőrző félnek (a legrosszabb esetben) várnia kell a kellően friss visszavonási információ beszerzéséhez.

Különböző források különböző módon szokták definiálni a kivárási időt. A kivárási idő komponensei egyrészt az aláírási szabályzattól, másrészt a hitelesítés-szolgáltató visszavonási állapot közzétételére vonatkozó eljárásától függenek (lásd: 6.20. ábra):



6.23. ábra. **Kivárási idő, példa**

1. Az aláíró észleli, hogy a magánkulcsa kompromittálódott, és bejelenti ezt a hitelesítés-szolgáltatónak. Nem szokás beleérteni a kivárási időbe, amíg az aláíró nem veszi észre a kompromittálódást. Ez nem a technológia, hanem az aláíró hibája. Ha ezt beleértenénk, a kivárási idő végtelen lenne. Azon időtartamot viszont egyes értelmezések beleérthetik a kivárási időbe, amíg a visszavonási kérelem eljut a hitelesítés-szolgáltatóhoz.

2. A hitelesítés-szolgáltató megállapítja a visszavonási kérelem jogosultságát. Az ehhez szükséges idő egyértelműen beleszámít a kivárási időbe. Ha a hitelesítés-szolgáltató megfelelő folyamatokat épít ki, akkor ennek ideje minimálisra szorítható.

3. A hitelesítés-szolgáltató átvezeti nyilvántartásában a megváltozott visszavonási állapotot, és közzéteszi az új visszavonási információt. Az ezekhez szükséges idő szintén beleszámít a kivárási időbe. Ha a hitelesítés-szolgáltató kellően gyorsan működik, ennek ideje is minimálisra szorítható.

4. Az érintett fél értesül a megváltozott visszavonási állapotról (azaz például már nem a korábban letöltött CRL-eket cache-eli). Az ehhez szükséges időt nem szokás beleszámítani a kivárási időbe, hiszen az érintett félnek már megvan a lehetősége, hogy a friss visszavonási állapotot használja.

5. Lehet, hogy az érintett fél már értesült a megbízható visszavonási állapotról, de még nem rendelkezik pozitív bizonyítékkal, amely igazolná, hogy a kérdéses tanúsítvány $t_{control}$ időpontban érvényes volt, azaz nem tud olyan archív aláírást létrehozni, amely később harmadik fél számára is igazolná az aláírás érvényességét.

6.5.4.4.3. Hogyan ellenőrizzük a visszavonási állapotot?

Többféle megközelítés szerint vizsgálhatjuk egy tanúsítvány visszavonási állapotát. Gyakran célszerű eltérő módszereket használni a végfelhasználói és a szolgáltatói tanúsítványokkal

kapcsolatban. Az alábbiakban a főbb értelmes módszereket foglaljuk össze:

1. *Nem vizsgálunk visszavonási állapotot*, azaz a tanúsítványt egészen addig érvényesnek tekintjük, amíg csak le nem jár.

   Ez olyan értelemben konzisztens hozzáállás, hogy két különböző fél ugyanazon időpontban ilyen módon garantáltan azonos eredményre jut a tanúsítvánnyal kapcsolatban. Végfelhasználói tanúsítványok esetén ez nem szerencsés, mert a végfelhasználók magánkulcsai viszonylag gyakran kompromittálódnak. Ugyanakkor ez sok esetben jó megoldást jelenthet.

   6.15. Példa: *Az e-Szignó Hitelesítés Szolgáltató 10 percig érvényes tanúsítványt bocsát ki az OCSP válaszadói részére.*
   A válaszadó tanúsítványában az `ocspNoCheck` *kiterjesztéssel jelöli, hogy olyan OCSP válaszadói tanúsítványról van szó, amelyhez nem tesz közzé visszavonási információt. Az OCSP válaszadó egységek kulcsai jól védettek, nagyon valószínűtlen, hogy kompromittálódnak. Amennyiben mégis felmerül a gyanúja, hogy egy válaszadó magánkulcsa kompromittálódott, az e-Szignó Hitelesítés Szolgáltató nem bocsát ki több tanúsítványt az adott kulcshoz, hanem új kulcsot generál, és a továbbiakban ezen új kulcshoz bocsát ki OCSP válaszadói tanúsítványokat. Ha egy támadó megszerezte a kulcsot, legfeljebb 10 percig tud visszaélni vele.*

2. *Mindaddig érvényesnek tekintjük a tanúsítványt, amíg tudomást nem szerzünk róla, hogy visszavonták.* Ezen belül:

   a. *Régi visszavonási információt is elfogadunk.* Célszerű definiálni, hogy mennyire lehet régi a visszavonási információ, nem célszerű bármilyen régit elfogadni; a közvetlenül a tanúsítvány kibocsátását követően megjelent CRL alig mond többet, mint maga a tanúsítvány. E megoldás bármikor elvégezhető, de nem biztosítja, hogy két ellenőrző fél ugyanarra az eredményre jusson egyazon aláírással kapcsolatban.

   b. *Csak olyan visszavonási információt fogadunk el, amely még nem járt le*, azaz megköveteljük, hogy a visszavonási információban szereplő `nextUpdate` legyen későbbi, mint az az időpont, amire nézve ellenőrzünk (control time).

      E követelmény szerint olyan visszavonási információt használunk, amelynél még nem biztos, hogy van frissebb. A szolgáltató `nextUpdate`-kor biztosan ki fog bocsátani új visszavonási információt, amíg a `nextUpdate` el nem jön, a meglévőt használhatjuk, és csak `nextUpdate` után kérdezzük majd le a frissebbet. E megoldásnak előnye, hogy bármikor elérhető ilyen frissességű visszavonási információ. Hátránya, hogy előfordulhat, hogy még `nextUpdate` előtt megjelenik

frissebb információ, és ezt nem vesszük észre, így előfordulhat, hogy két ellenőrző fél nem ugyanarra az eredményre jut egyazon aláírással kapcsolatban.

6.16. Példa: *Az X hitelesítés-szolgáltató minden nap éjfélkor bocsát ki CRL-t. Bendegúz reggel 8 órakor letöltötte a CRL-t. Alajos délelőtt 10 órakor veszi észre, hogy ellopták a magánkulcsát, jelzi ezt a szolgáltatónak, és negyed 11-kor megjelenik az új CRL, amely szerint Alajos tanúsítványa már érvénytelen. Manfréd, a támadó, délután 3 órakor ír alá Alajos lopott kulcsával, és lehet, hogy Bendegúz fél 4-kor érvényesnek fogadja el az aláírást, mert ő még mindig a reggel 8 órakor letöltött CRL-t használja (mert a* `nextUpdate` *csak éjfélkor jön el). Cili, aki fél 4-kor tölti le a CRL-t, már érvénytelennek fogja tekinteni az aláírást. Ha Manfréd másnap próbál aláírni, azt már Bendegúz is érvénytelennek fogja tekinteni.*

3. *Meg szeretnénk róla győződni, hogy a tanúsítvány a kérdéses időpontot (pl. az aláírás időpontját) követően még érvényes volt.*

   Tegyük fel, hogy a hitelesítés-szolgáltató azt vállalja, hogy a visszavonási kérelem benyújtását követően $\Delta t_{kivarasi}$ időn belül közzéteszi a visszavonási állapotot. Ezt úgy tehetjük meg, hogy a control time után $\Delta t_{kivarasi}$ időt várunk, és csak ezt követően kérdezzük le a tanúsítvány visszavonási állapotát. Ha az ekkor beszerzett visszavonási információ szerint a tanúsítvány érvényes, akkor érvényesnek tekintjük a tanúsítványt. E gondolatmenet szerint a control time után $\Delta t_{kivarasi}$ idővel már meg kellett volna, hogy jelenjen egy frissebb visszavonási információ. Ekkor a friss visszavonási információ *hiánya* jelenti a bizonyítékot.

   E megoldás már garantálja, hogy két ellenőrző fél azonos eredményre jut az aláírással kapcsolatban.

   Hátránya e megoldásnak, hogy nem végezhető el bármikor, mert így – feltéve, hogy a szolgáltató nem $\Delta t_{kivarasi} = 0$ vállalást tesz – várakozni kell. Egyes esetekben nagyon jelentős költségeket okozhat, ha a folyamatot várakoztatni kell.

   Vigyázat, e megoldás csak akkor alkalmazható, ha az ellenőrző fél maga kérdezi le a tanúsítvány visszavonási állapotát. Ha harmadik féltől kapja a visszavonási információt, nem tudhatja, hogy nem egy korábbi visszavonási információt kapott-e. Szintén problémát jelent, hogy ha nem megbízható csatornán érjük el a visszavonási információt. Ilyenkor nem tudhatjuk, hogy egy támadó nem egy korábbi visszavonási információt[25] játszik-e vissza nekünk.

   6.17. Példa: *Az X hitelesítés-szolgáltató minden nap éjfélkor bocsát ki CRL-t. Vállalja, hogy visszavonás esetén legkésőbb 4 órával a visszavonás*

[25] E probléma CRL-ek esetén súlyos. OCSP esetén a visszajátszás kivédhető, ha mind a kérdésben, mind a válaszban feltüntetünk egy friss nonce elemet.

bejelentését követően soron kívüli CRL-t is kibocsát. Bendegúz reggel 9 órakor kap Alajostól egy aláírást. Az aláíráson nincsen időbélyeg, így az aktuális időpontra nézve ellenőrzi az aláírást.

Bendegúz kivárja a 4 órát, és 13 órakor tölti le a CRL-t a hitelesítés-szolgáltatótól. Ekkor továbbra is az aznap 0 órakor megjelent CRL-t látja, és ebből arra következtet, hogy reggel 9 óráig senki nem nyújtott be a hitelesítés-szolgáltatóhoz érvényes visszavonási kérelmet. Ez alapján érvényesnek tekinti az aláírást.

Bendegúz csatolja az aláíráshoz a CRL-t (XAdES-C aláírássá egészíti ki), és ezt továbbítja Dezsőnek. Dezső önmagából a XAdES-C aláírásból nem tud olyan mértékben meggyőződni az aláírás érvényességéről, mint Bendegúz; nem tudja, hogy Bendegúz a 13 órakor lekérdezett CRL-t küldte-e el neki. Lehet, hogy Alajos hajnali 3-kor kérte a tanúsítvány visszavonását, és így reggel 7-kor már létezett olyan CRL, amely szerint az aláírás érvénytelen, csak Bendegúz nem ezt csatolta. (Talán azért, hogy megtévessze Dezsőt.) Ha Dezső gondosan jár el, neki magának is le kell töltenie a CRL-t.

4. *Harmadik fél felé is bizonyítani szeretnénk, hogy a tanúsítvány a kérdéses időpontot (pl. az aláírás időpontját) követően még érvényes volt.* Ekkor olyan visszavonási információ alapján ellenőrizzük a tanúsítványt, amely későbbi időpontra vonatkozik, mint a control time, azaz a benne szereplő `thisUpdate` érték[26] nagyobb, mint $t_{control}$.

   Így pozitív bizonyítékkal rendelkezünk, ez alapján más is elfogadhatja az aláírást, anélkül, hogy a hitelesítés-szolgáltatóhoz kellene fordulnia. *Archív aláírás esetén ezt a megközelítést célszerű használni.*

   Előnye e megoldásnak, hogy – korrektül működő hitelesítés-szolgáltatók esetén – két ellenőrző fél garantáltan azonos eredményre jut az aláírással kapcsolatban. Hátránya e megoldásnak, hogy várakozást igényel, nem végezhető el bármikor.

5. *Még tovább várakozunk.* Minél tovább várakozunk, annál valószínűbb, hogy kiderül, ha valami baj van az aláírással. Ugyanakkor minél tovább várakozunk, annál könnyebben előfordulhat, hogy már nem érhető el egy szükséges visszavonási információ. Ne feledjük, a hitelesítés-szolgáltató megteheti, hogy a lejárt tanúsítványokat már nem tünteti fel a CRL-ben, illetve már nem biztosít rájuk OCSP szolgáltatást.

   A PKI ellenőrzés során azért sincs értelme túl sokáig várakoznunk, mert az Eat. 14. § (3) szerint „tanúsítvány visszamenőleges visszavonására nem kerülhet sor”. Ez azt jelenti, hogy ha egy szolgáltató kibocsátott olyan visszavonási információt, amely szerint $t_{control}$ időpontban a tanúsítvány érvényes volt (és `thisUpdate`$>t_{control}$), akkor később nem

[26] Az az időpont, amire az adott visszavonási információ vonatkozik.

jelenhet meg olyan információ, amely szerint a tanúsítvány $t_{control}$ időpontban mégis érvénytelen volt.

6.5.4.4.4. Mikor alkalmazzunk kivárási időt?

A kivárási idő jelentette probléma akkor merül fel, ha egy aláírás ellenőrzésekor nem elégszünk meg a legfrissebb elérhető visszavonási információval, hanem valamilyen értelemben „friss" visszavonási információt követelünk meg. Két ok miatt kényszerülhetünk várakozásra. Egyrészt, mert a visszavonási információk egy része[27] csak periodikusan jelenik meg. Másrészt, mert a hitelesítés-szolgáltatók gyakran nem azonnal, hanem csak hosszú idő után dolgozzák fel a visszavonási kérelmeket.

6.18. Példa: *Az X hitelesítés-szolgáltató minden nap éjfélkor bocsát ki CRL-t. Manfréd hajnali 3-kor ellopja Alajos kártyáját, Alajos azonnal észreveszi és jelenti a lopást. A hitelesítés-szolgáltató 3 óra 5 perckor befogadja Alajos visszavonási kérelmét, de a következő CRL csak éjfélkor fog megjelenni. Ha Bendegúz délelőtt 10 órakor kap egy aláírást, és kivárási időt alkalmaz, csak éjfélkor fogadhatja be az aláírást.*

6.19. Példa: *Az X hitelesítés-szolgáltató azt vállalja, hogy 4 órán belül dolgozza fel a visszavonási kérelmeket. Ha Bendegúz kap egy aláírást, amely a rajta lévő időbélyeg szerint 9 órakor készült, és kivárási időt alkalmaz, akkor csak 13 órakor fogadhatja el az aláírást.*

Sok esetben – elsősorban szolgáltatói tanúsítványok esetén – az is elegendő, ha még nem lejárt visszavonási információra támaszkodunk. Akkor mondjuk, hogy *az ellenőrzés során kivárási időt alkalmazunk*, ha megköveteljük, hogy a visszavonási információ vonatkozzon azon időpontra, amelyre nézve az ellenőrzést végezzük (azaz `thisUpdate`$>t_{control}$).

„Fontos" aláírások esetén a végfelhasználói tanúsítványokra mindenképpen célszerű kivárási időt alkalmazni. Ha különösen megbízható rendszert építünk, vagy ha azt szeretnénk, hogy az aláírások érvényességét hosszú távon is ellenőrizni lehessen – ilyenek az archív aláírások –, akkor a szolgáltatói tanúsítványokra is célszerű kivárási időt alkalmazni.

A kivárási idő jelentette probléma leegyszerűsíthető, ha:

- A hitelesítés-szolgáltató gyors visszavonás-kezelés szolgáltatást nyújt, és vállalja, hogy a beérkező felfüggesztési vagy visszavonási kérelmeket gyorsan, elhanyagolhatóan rövid idő alatt dolgozza fel, és vállalja, hogy megtéríti a kárt, ha ez mégsem így történik.

[27] Ez elsősorban CRL esetén van így, de a CRL alapján működő OCSP esetén is igaz.

- A hitelesítés-szolgáltató OCSP segítségével is közzéteszi a tanúsítványok visszavonási állapotát, és az OCSP szolgáltatást olyan módon nyújtja, hogy a t időpontban lekérdezett OCSP válasz a t időpontra nézve releváns információt ad ($t \simeq$ `thisUpdate`).

Ekkor akár közvetlenül az aláírás létrehozását követően OCSP segítségével beszerezhető olyan bizonyíték, amely harmadik fél számára is igazolja, hogy az aláírás pillanatában az aláíró tanúsítványa érvényes volt.

Az ellenőrzés során megbízható gyökértanúsítványokra vezetjük vissza az aláírás érvényességét. Időbélyegek segítségével igazolhatjuk, hogy egy adott aláírás, tanúsítvány, CRL, OCSP válasz vagy időbélyeg egy adott időpontban már létezett, de az időbélyegen lévő aláírást is ellenőriznünk kell. Akárhogyan is járunk el, az utolsó ellenőrzést mindig az aktuális időpontra, az ellenőrzés időpontjára nézve végezzük el. Ezen utolsó ellenőrzésnél viszont elvileg sem lehet kivárási időt alkalmazni.

Az aláírási szabályzatban célszerű rögzíteni, hogy mikor, mely aláírás ellenőrzése során követeljük meg a kivárási idő alkalmazását.

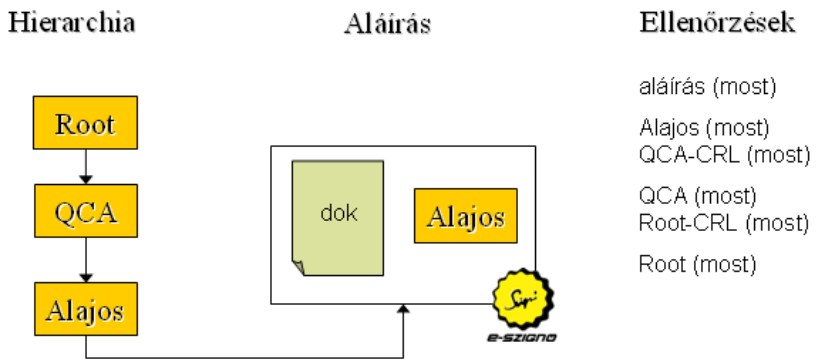
6.5.4.5. Példák

A következőkben néhány példa segítségével illusztráljuk, hogy mikor milyen ellenőrzéseket célszerű elvégezni.

Aláírás ellenőrzése, -BES

Tekintsük a 6.24 ábrán szereplő aláírást és tanúsítványláncot! Sem az aláíráson, sem más információn nincsen időbélyeg, így nincs megbízható információnk arra vonatkozóan, hogy az aláírás mikor jött létre. A PKI ellenőrzés szempontjából lényegtelen, hogy az aláíró állítása szerint mikor készült az aláírás. Lehet, hogy az aláíró hazudik, például lehet, hogy az aláírást egy tolvaj készítette lopott kártyával, csak visszadátumozta. Jobb híján abból indulunk ki, hogy az aláírást épp a most elmúlt másodpercben készítették. Így az aláírást az ellenőrzés időpontjára, azaz a *most* időpontra vonatkozóan ellenőrizzük.

- Ellenőrizzük Root aláírását QCA tanúsítványán (*most*)!
  - Tekinthető Root QCA kibocsátójának?
  - Root megbízható gyökér-e *most*?
  - Mivel Root megbízható gyökér, Root visszavonási állapotát nem lehet, nem kell ellenőrizni.
- Ellenőrizzük QCA aláírását Alajos tanúsítványán (*most*)!



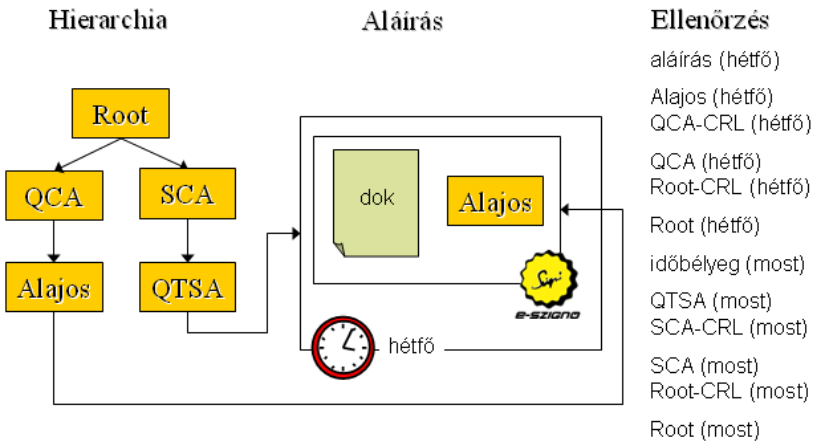
6.24. ábra. **Aláírás ellenőrzése, -BES**

  - Tekinthető QCA Alajos tanúsítványa kibocsátójának?
  - QCA tanúsítványa érvényes a *most* időpontban?
  - A Root által kibocsátott CRL releváns-e a *most* időpontra nézve?
  - A Root által kibocsátott CRL szerint érvényes-e QCA tanúsítványa *most*?
  - Ellenőrizzük Root aláírását az QCA-ra vonatkozó CRL-en *most*!
    - ⋆ Tekinthető-e Root az adott CRL kibocsátójának?
    - ⋆ Root megbízható gyökér-e *most*?
    - ⋆ Mivel Root megbízható gyökér, Root visszavonási állapotát nem lehet, nem kell ellenőrizni.

- Ellenőrizzük Alajos aláírását a „dok” dokumentumon (*most*)!
  - Tekinthető Alajos az „dok” dokumentum aláírójának?
  - Alajos tanúsítványa érvényes a *most* időpontban?
  - A QCA által kibocsátott CRL releváns-e a *most* időpontra nézve?
  - A QCA által kibocsátott CRL szerint érvényes-e Alajos tanúsítványa *most*?
  - Ellenőrizzük QCA aláírását az Alajos tanúsítványára vonatkozó CRL-en (*most*)!
    - ⋆ Tekinthető QCA az adott CRL kibocsátójának?
    - ⋆ Korábban már beláttuk, hogy QCA tanúsítványa *most* érvényes, és *most* nincs visszavonva, így az ellenőrzést nem kell tovább folytatni.

Ezzel beláttuk, hogy Alajos aláírása a „dok” dokumentumon *most* érvényes.

Aláírás ellenőrzése, -T



6.25. ábra. **Aláírás ellenőrzése, -T**

Tekintsük a 6.25 ábrán szereplő aláírást és tanúsítványláncot! Az aláíráson időbélyeg szerepel, miszerint az aláírás, az aláírt dokumentum és Alajos tanúsítványa *hétfőn* már létezett. A többi PKI objektumon lévő aláírást csak az aláírás időpontjára vonatkozóan ellenőrizhetjük.

Ekkor a következő aláírás-ellenőrzéseket kell elvégeznünk:

> Ellenőrizzük, hogy érvényes-e az időbélyeg, azaz támaszkodhatunk-e a benne szereplő időpontra! Nem tudunk olyan megbízható időpontról, amikor az időbélyeg már biztosan létezett (másik időbélyeg nincs rajta), így azt csak az ellenőrzés időpontjára (a *most* időpontra nézve) nézve ellenőrizhetjük.

- Ellenőrizzük Root aláírását SCA tanúsítványán (*most*)!
  - Tekinthető Root SCA kibocsátójának?
  - Root megbízható gyökér-e *most*?
  - Mivel Root megbízható gyökér, Root visszavonási állapotát nem lehet, nem kell ellenőrizni.
- Ellenőrizzük SCA aláírását QTSA tanúsítványán (*most*)!
  - Tekinthető SCA QTSA kibocsátójának?
  - SCA tanúsítványa érvényes a *most* időpontban?
  - A Root által kibocsátott CRL releváns-e a *most* időpontra nézve?
  - A Root által kibocsátott CRL szerint érvényes-e SCA tanúsítványa *most*?
  - Ellenőrizzük Root aláírását az SCA-ra vonatkozó CRL-en *most*!
    - ⋆ Tekinthető-e Root az adott CRL kibocsátójának?
    - ⋆ Root megbízható gyökér-e *most*?

⋆ Mivel Root megbízható gyökér, Root visszavonási állapotát nem lehet, nem kell ellenőrizni.

- Ellenőrizzük QTSA aláírását az időbélyegen (*most*)!
  - Tekinthető QTSA az időbélyeg kibocsátójának?
  - QTSA tanúsítványa érvényes a *most* időpontban?
  - Az SCA által kibocsátott CRL releváns-e a *most* időpontra nézve?
  - Az SCA által kibocsátott CRL szerint érvényes-e QTSA tanúsítványa *most*?
  - Ellenőrizzük SCA aláírását a QTSA tanúsítványára vonatkozó CRL-en (*most*)!
    - ⋆ Tekinthető SCA az adott CRL kibocsátójának?
    - ⋆ Korábban már beláttuk, hogy SCA tanúsítványa *most* érvényes, és *most* nincs visszavonva, így az ellenőrzést nem kell tovább folytatni.

  Meggyőződtünk róla, hogy az időbélyeg érvényes, így a következőkben már alapozhatunk rá, hogy a dokumentum, az aláírás, és Alajos tanúsítványa *hétfőn* már léteztek. Az időbélyeg nem védi QCA tanúsítványát, így QCA tanúsítványát a *most* időpontra nézve ellenőrizzük. Az időbélyeg a CRL-eket sem védi, így azokat is a *most* időpontra nézve ellenőrizzük.

- Ellenőrizzük Root aláírását QCA tanúsítványán (*most*)!
  - Tekinthető Root QCA kibocsátójának?
  - Root megbízható gyökér-e *most*?
  - Mivel Root megbízható gyökér, Root visszavonási állapotát nem lehet, nem kell ellenőrizni.
- Ellenőrizzük QCA aláírását Alajos tanúsítványán (*hétfő*)!
  - Tekinthető QCA Alajos tanúsítványa kibocsátójának?
  - QCA tanúsítványa érvényes volt-e *hétfőn*?
  - A Root által kibocsátott CRL releváns-e a *hétfő* időpontra nézve?
  - A Root által kibocsátott CRL szerint érvényes volt-e QCA tanúsítványa *hétfőn*?
  - Ellenőrizzük Root aláírását a CRL-en (*most*)!
    - ⋆ Tekinthető-e Root az adott CRL kibocsátójának?
    - ⋆ Root megbízható gyökér-e *most*?
    - ⋆ Mivel Root megbízható gyökér, Root visszavonási állapotát nem lehet, nem kell ellenőrizni.

- Ellenőrizzük Alajos aláírását a „dok” dokumentumon (*hétfő*)!
  - Tekinthető Alajos az adott aláírás készítőjének?
  - Alajos tanúsítványa érvényes volt-e *hétfőn*?
  - A QCA által kibocsátott CRL releváns-e a *hétfő* időpontra nézve?
  - A QCA által kibocsátott CRL szerint érvényes volt-e Alajos tanúsítványa *hétfőn*?
  - Ellenőrizzük QCA aláírását az Alajos tanúsítványára vonatkozó CRL-en (*most*[28])!
    - ⋆ Tekinthető-e QCA az adott CRL kibocsátójának?
    - ⋆ QCA tanúsítványa érvényes-e a *most* időpontban?
    - ⋆ A Root által kibocsátott CRL releváns-e a *most* időpontra nézve?
    - ⋆ A Root által kibocsátott CRL szerint érvényes-e QCA tanúsítványa *most*?
    - ⋆ Ellenőrizzük Root aláírását az QCA-ra vonatkozó CRL-en *most*! Történetesen ez a CRL azonos az SCA tanúsítványára vonatkozó CRL-lel, és ennek érvényességét már beláttuk a *most* időpontra nézve, így nem folytatjuk tovább az ellenőrzést.

Meggyőződtünk róla, hogy a fenti bizonyítékok alapján Alajos aláírása a „dok” dokumentumon érvényes a *hétfő* időpontra nézve.

Tegyük fel, hogy a *hétfő* időpontra nézve nem sikerül ellenőriznünk az aláírást (mert pl. nem sikerül meggyőződni a rajta lévő időbélyeg érvényességéről), ekkor megpróbálhatjuk - BES aláírásként, a *most* időpontra nézve is ellenőrizni az aláírást, ami annyit jelent, hogy eltekintünk a rajta lévő időbélyegtől. Ha a *most* időpontra nézve sikerül igazolnunk az aláírás érvényességét, akkor – elvileg[29] – érvényesnek fogadhatjuk el az aláírást. (Egy „jó” aláírást önmagában az nem tesz érvénytelenné, ha ráteszünk egy „rossz” időbélyeget.)

6.5.5. Az adott folyamatban elfogadható-e az aláírás

Az aláírás kriptográfiai ellenőrzése (6.5.3. fejezet) során meggyőződtünk róla, hogy az aláírás összetartozik-e az aláírt dokumentummal és az aláíró nyilvános kulcsával. Az aláírás PKI ellenőrzése (6.5.4. fejezet) során meggyőződtünk róla, hogy az aláíró tanúsítványa, amely az adott nyilvános kulcsot tartalmazza, érvényes volt az aláírás pillanatában, azaz a PKI eszközeivel bizonyítható, hogy az aláírás pillanatában kizárólag az aláíró birtokolta az adott

[28] QCA tanúsítványáról még csak azt láttuk be, hogy *hétfőn* érvényes volt. Külön megvizsgáljuk, hogy *most* is érvényes-e.

[29] Egyáltalán nem biztos, hogy egy adott rendszer valóban elfogadja az ilyen aláírást. Lehet, hogy a befogadó mindenképpen -T aláírásokat követel meg, és nem hajlandó ő alakítani őket át -T aláírássá. (Sok dokumentum átalakítása esetén az ehhez szükséges időbélyegek költsége jelentős tényező lehet.) Az is előfordulhat, hogy egy adott aláírás-ellenőrző alkalmazás az érvénytelen időbélyeg miatt eleve elutasít egy ilyen aláírást, és nem próbálkozik más időponttal.

nyilvános kulcsot. Mindebből még nem következik, hogy elfogadhatjuk az aláírást, illetve az aláírt dokumentumot egy adott folyamatban. [5]

A kérdés teljeskörű kezeléséhez a további, az adott folyamatra jellemző kéréseket is célszerű megvizsgálnunk:

- *Milyen biztonsági szintű az aláírás?*

  Minősített vagy fokozott biztonságú elektronikus aláírásról van szó? (Ez alapvetően meghatározza, hogy mennyire erős bizonyítékot jelent bíróság előtt.) Ha nem minősített az aláírás, akkor történt-e személyes találkozás a regisztráció során?

- *Vállalnak-e felelősséget a hitelesítés-szolgáltatók?*

  Mekkora felelősséget vállal az érintett hitelesítés-szolgáltató a tanúsítvánnyal okozott károkért? (Mekkora kár származhat belőle, ha tévesen fogadunk be egy aláírást? Megtéríti-e valaki a károkat, ha befogadjuk az aláírást, de kiderül, hogy a szolgáltató hibázott, és mégsem az aláíró készítette az aláírást.) Ezen ellenőrzést célszerű elvégezni a tanúsítványláncban szereplő minden egyes hitelesítés-szolgáltatóra.

- *Mennyire vagyunk biztosak a PKI ellenőrzés eredményében?*

  A lánc mely szintjein alkalmaztunk kivárási időt? Előfordulhat, hogy a lánc valamely elemét időközben visszavonták (és pl. meg is jelent erről egy CRL), csak mi nem vettük észre?

- *Melyik gyökérre vezettük vissza az aláírást?*

  Bizonyos aláírásokat csak bizonyos gyökerekre szabad visszavezetni. Például a magyar közigazgatásban elfogadható aláírásokat a Közigazgatási Gyökér Hitelesítés Szolgáltató megbízható gyökértanúsítványára visszavezetve szabad elfogadni.

- *Ki írta alá a dokumentumot?*

  Ha az aláírás megfelelő biztonsági szintű, bizonyítani tudjuk, hogy az aláíráshoz használt magánkulcs a PKI szerint az aláíró kizárólagos birtokában volt.

  Tudjuk, hogy ki az aláíró? Például álneves tanúsítvány esetén nem ismerjük az aláíró kilétét, de ettől még szükség esetén tudjuk bizonyítani, hogy valóban ő készítette az aláírást. Ha nem álneves a tanúsítvány, akkor ismerjük az aláíró nevét. Ha a tanúsítványában annyi szerepel, hogy `CN=Kovács János, C=HU`, az nagyon sok emberre illeszkedhet, ez alig mond többet, mintha álneves tanúsítványt látnánk. Ha az szerepel a tanúsítványban, hogy `CN=Kovács János, SN=szemelyi.ig.szam:123456, C=HU`, akkor egyértelműen azonosítható az illető, bár továbbra sem mond semmit nekünk, mert a kérdéses Kovács Jánost nem a személyi igazolványa alapján ismerjük (hanem pl.

arcról, vagy telefonon beszéltünk vele, esetleg csak annyit tudunk róla, hogy a Kókler Bt-nél dolgozik), és nem tudjuk, mi a személyi igazolványának a száma.

Gyakran nem kell tudnunk, hogy ki az aláíró.

- *Jogosan írta alá az illető a dokumentumot?*

  Itt támaszkodhatunk az aláíró által állított szerepre, vizsgálhatjuk az aláíró tanúsítványában lévő adatokat (pl. az aláíró DN-jében szereplő `title` értéket), vagy vizsgálhatjuk az aláíráshoz csatolt attribútum-tanúsítvány tartalmát is. (Lásd: 11. fejezet.)

- *Milyen formátumú az aláírt dokumentum?*

  Lehet, hogy az aláírás minden szempontból elfogadható lenne, de az adott formátumú aláírt dokumentumot nem tudjuk feldolgozni. Nem létezik olyan informatikai rendszer, amely bármilyen formátumú dokumentumot tud kezelni, ezért aki (aláírt) elektronikus dokumentumok befogadására adja a fejét, annak célszerű meghatároznia és partnerei tudomására hoznia, hogy milyen formátumú dokumentumokat fogad el. Ez ugyanúgy igaz a dokumentum, az aláírás-konténer (6.4.2. fejezet) és az aláírás-blokk (6.4.1. fejezet) formátumára. (Igaz, ha a konténer nem ismert, az aláírás-ellenőrző alkalmazás várhatóan nem találja meg az aláírást, és ha az aláírás-blokk nem ismert, nem fog sikerülni az aláírás-ellenőrzés.)

  Papír alapú iratok esetén is gyakran szabnak meg formai követelményeket az iratok benyújtására vonatkozóan. Például az adóbevallást csak a meghatározott formanyomtatványon lehet benyújtani, és például egy kockás füzetből kitépett lapon nem fogadják el.

Hangsúlyozzuk, *nem lehet általánosan kimondani, hogy egy rendszernek minden elektronikusan aláírt iratot be kell fogadnia.* Ehelyett azt célszerű meghatározni, hogy a rendszer milyen iratokat fogad be, milyen típusú aláírást milyen módon ellenőriz rajtuk.

E kérdések nagy részét célszerű aláírási szabályzatban rögzíteni, ekkor nem szükséges minden egyes aláírás ellenőrzésekor külön döntést hoznunk.

6.5.6. Megáll-e az aláírás bíróság előtt?

A legtöbb elektronikus aláírás soha nem kerül bíróság elé (ahogy a legtöbb kézzel írott aláírás sem). Ugyanakkor mind a kézzel írott aláírásnak, mind az elektronikus aláírásnak az a célja, hogy szükség esetén megállja a helyét bíróság előtt is.

Valószínűtlen, hogy egy bíróság öncélúan az aláírás érvényességét vizsgálná. Sokkal valószínűbb, hogy a vita kézzel fogható problémával kapcsolatos, például: Valóban ez és ez

szerepelt a benyújtott adóbevallásban? Kötöttek-e a felelek egymással szerződést, és pontosan miben egyeztek meg egymással? Átvett-e valaki egy adott elektronikus küldeményt?

Az elektronikus aláírás várhatóan csak egy bizonyíték lesz a vitában, esetleg sok más bizonyíték mellett. Akkor juthat szerephez az elektronikus aláírás, ha olyan kérdés merül fel, hogy valóban az aláíró írta-e alá a dokumentumot, illetve valóban azt a dokumentumot írta-e alá. A bíróságot ekkor sem az aláírás matematikai, kriptográfiai vagy PKI szempontból vett érvényessége érdekli, hanem az aláíró *szándéka*: Valóban alá akarta írni a dokumentumot? Vegyük figyelembe, hogy az elektronikus aláírás jogi értelemben nem „letagadhatatlan". A minősített elektronikus aláírás esetén igaz, hogy ha az aláírás műszakilag érvényes, akkor vélelmezni kell, hogy valóban az aláíró készítette az aláírást, és az aláírt dokumentum nem változott meg az aláírás óta. De még ekkor, a minősített elektronikus aláírás esetében is, van lehetőség ellenkező bizonyításnak.

Előfordulhat, hogy az aláírót megtévesztették, megzsarolták, esetleg tévedésből vagy kényszer alatt írta alá a dokumentumot. Lehet, hogy részeg volt, esetleg begyógyszerezték, de az is lehet, hogy kiskorú vagy nem beszámítható.

Lényeges kiemelni, hogy *az elektronikus aláírás műszaki érvényességéből nem következik, hogy az aláírás garantáltan megáll majd bíróság előtt.* Az elektronikus aláírás pusztán egy – nagyon erős – bizonyítékot jelenthet. Ennek ellenkezője sem igaz: *ha egy elektronikus aláírás műszakilag érvénytelen, abból sem következik, hogy a bíróság ne fogadná el érvényes bizonyítékként.*

> ***6.20. Példa:*** *Adott egy szerződés Alajos és Manfréd között, amelyben Alajos eladja a házát Manfrédnek 25 millió forintért. Mind Alajos, mind Manfréd minősített elektronikus aláírása érvényes a szerződésen. Alajos bírósághoz fordul, azt állítja, hogy nem kapott pénzt, és kényszer alatt írta alá a szerződést. Állítása szerint Manfréd elrabolta a családját, és azzal fenyegette, hogy láncfűrésszel feldarabolja őket, ha Alajos nem írja alá a szerződést. Alajos feljelentést tesz Manfréd ellen.*
>
> *A bíróság a következő (nem PKI) körülményeket is figyelembe veheti az aláírással kapcsolatos döntés során:*
>
> - *Manfréd az alvilág egy ismert alakja, már volt büntetve fegyveres rablás, emberrablás miatt.*
> - *Alajos büntetlen előéletű, és még tilosban sem parkolt.*
> - *Alajos háza jóval többet ér, mint 25 millió forint, és Alajosnak nem volt sürgősen szüksége pénzre. Valószínűtlen, hogy Alajos ennyiért eladta volna.*
> - *Időközben Bendegúz és Cili is feljelentést tettek Manfréd ellen hasonló vádakkal. (A vádak szerint Cilinek a nagymamáját, Bendegúznak pedig az*

aranyhörcsögét akarta feldarabolni Manfréd.) Alajos, Bendegúz és Cili nem ismerik egymást.

- *Manfréd lakásán talált a rendőrség egy véres láncfűrészt, rajta Manfréd ujjlenyomataival.*
- *A rendőrség hosszú és izgalmas autós üldözést követően elfogta Manfrédet és bandáját, akik végül töredelmesen bevallották, hogy valóban erőszakkal kényszerítették Alajost, Bendegúzt és Cilit, hogy írják alá a szerződéseket.*

Lehet, hogy Alajos visszakapja a házát, annak ellenére, hogy műszakilag érvényes az aláírása a szerződésen.

6.21. Példa: *A Q Bank kapott Alajostól egy minősített aláírással ellátott átutalási megbízást, amelyben Alajos minden megtakarítását átutalta Manfréd egy külföldi számlájára. Alajos azt állítja, hogy ő nem adott a banknak ilyen átutalási megbízást. Alajos aláírása érvényes az átutalási megbízáson, és az aláírása minősített elektronikus aláírás, így neki kell bizonyítania állítását.*

Alajos arra hivatkozik, hogy egy Manfréd által készített vírus fertőzhette meg a gépét, és az aláírandó dokumentum lenyomata helyett egy másik dokumentum, egy hamis átutalási megbízás lenyomatát küldte el az aláírás-létrehozó eszköznek. A vírus nem mutatható ki Alajos számítógépén, Alajos ezt azzal magyarázza, hogy a vírus valószínűleg megsemmisítette önmagát. Alajosnak nehéz dolga lesz, ha erről meg akarja győzni a bíróságot.

Tegyük fel, hogy Alajos nincs egyedül, és a Q Bank mintegy 300 000 ügyfele néhány nap leforgása alatt mind átutalta az összes pénzét Manfréd külföldi számlájára. Ebből már nagyon érdekes jogi eset adódhat.

6.22. Példa: *Alajos és Bendegúz krumplival kereskednek. Alajos Interneten keresztül, fokozott biztonságú elektronikus aláírással szokott krumplit rendelni Bendegúztól, aki vasúton küldi el a rendelt mennyiséget. Alajos tanúsítványa már évekkel ezelőtt lejárt, és nem újította meg. Alajos aláírásai már évek óta érvénytelenek, de ez nem zavarja Bendegúzt. Az év során Alajos mintegy 63 megrendelést küldött Bendegúznak, aki 63 rakomány krumplit szállított Alajosnak.*

Egyszer csak Alajos azt állítja, hogy a 47. megrendelést nem ő írta alá, az nem tőle származik. Vissza akarja kapni az érte fizetett pénzt. Arra hivatkozik, hogy nem is érvényes az aláírása a megrendelésen.

Kérdés, hogy ha az előző 46, és az utána következő 16 megrendelésen lévő aláírás ugyanazzal a kulccsal készült, akkor miért pont a 47. aláírását akarja letagadni? Ha kompromittálódott a magánkulcsa, akkor miért használta tovább? Ha nem ő küldte

a megrendelést, akkor miért vette át (az átvételt igazoló dokumentumot ráadásul kézzel írta alá), és miért fizette ki?

Ha bizonyítható, hogy valós tranzakciók történtek, és Alajos feltehetően valóban alá akarta írni a megrendelést, akkor lehet, hogy az aláírás annak ellenére megáll majd a bíróság előtt, hogy az műszakilag nem érvényes.

6.6. Az elektronikus aláírás hosszú távú érvényessége

Az elektronikus aláírásról szóló 2001. évi XXXV. törvény szerint az elektronikus aláírás megfelel az írásba foglaltság követelményeinek, a minősített elektronikus aláírással hitelesített dokumentum pedig a polgári perrendtartásról szóló törvény szerint teljes bizonyító erejű magánokiratnak minősül.

Csak akkor kapcsolódik bizonyító erő az aláíráshoz, ha az *érvényes*, azaz „az aláírás érvényességének ellenőrzéséből más nem következik". Amikor egy aláírás érvényességét vizsgáljuk, ellenőrizzük, hogy az aláírás összetartozik-e az aláírt dokumentummal, illetve aláíró tanúsítványában szereplő nyilvános kulccsal. Ezen túl ellenőrizzük, hogy az aláíró tanúsítványa érvényes volt-e az aláírás készítésének pillanatában. (Lásd: 6.5. fejezet.)

Az aláíró a tanúsítványához tartozó magánkulcs segítségével hozhat létre elektronikus aláírást. Ha aláíráskor érvényes a tanúsítványa, akkor érvényes aláírást hoz létre, és nem szeretnénk, hogy ez az aláírás később mégis érvénytelenné válhasson. Ha az aláírás később érvénytelenné válhat, akkor nem „letagadhatatlan". Attól függ, hogy az aláírásunk érvényes marad-e később is vagy sem, hogy később is tudjuk-e majd bizonyítani – műszaki szempontból – az érvényességét. (Lásd: 6.5. fejezet.)

Előfordulhat, hogy az a tanúsítvány, amely alapján az aláírás készült, érvénytelenné válik – lejár, vagy visszavonják. (A visszavonás történhet azért, mert később, az aláírást követően elvesztettük a tanúsítványhoz tartozó magánkulcsot, vagy mert valamely adatunk megváltozott.) Ha az aláírást ellenőrző érintett fél azt látja, hogy az aláíró tanúsítványa már nem érvényes, önmagában az aláírásból többé nem tudja megállapítani, akkor érvényes volt-e, amikor az aláírást készítették. Lehet, hogy azt meg tudja mondani, hogy a tanúsítvány mikor vált érvénytelenné, de azt nem tudja biztosan, hogy az aláírás mikor készült. (Előfordulhat, hogy az aláírás tartalmaz ugyan utalást az elkészítésének időpontjára, de ezt várhatóan nem fogadja el megbízható időpontnak. Egy személyi számítógép órája egyrészt nem pontos, másrészt nagyon könnyű átállítani, manipulálni.) Ha nem tud valamilyen más módon meggyőződni arról, hogy az aláírás mikor készült, akkor műszaki szempontból nem tudja bizonyítani az aláírás érvényességét, tehát az aláírást érvénytelennek tekinti.

Ha az aláíráson időbélyeget helyezünk el, akkor – egy időbélyegzés-szolgáltató, tehát egy megbízható harmadik fél állítása alapján – bizonyíthatjuk, hogy az aláírás mikor

(pontosabban, mely időpont előtt) készült. Az időbélyeg egy megbízható időbélyegzés-szolgáltató által kibocsátott, aláírt igazolás arról, hogy az időbélyegzett adat (esetünkben az aláírás) az időbélyegzés pillanatában már létezett. Az időbélyegzés-szolgáltatók nagyon vigyáznak az időbélyegek aláírására használt magánkulcsukra, de mégis előfordulhat, hogy ez illetéktelen kezekbe kerül. Mi történik ilyenkor? A támadó, aki megszerezte az időbélyegzés-szolgáltató magánkulcsát, ezentúl tetszőleges időpontot beleírhat az időbélyegekbe. Így ha visszavonják egy időbélyegzés-szolgáltató tanúsítványát, a támadó visszadátumozott időbélyegeket is kibocsáthat, és ezekről már nem lehet megállapítani, hogy a visszavonás után készültek. Hasonló helyzet áll elő, ha az időbélyegzés-szolgáltató tanúsítványa lejár: ha a magánkulcs valahogy mégis illetéktelen kezekbe kerül, a támadó visszadátumozott időbélyegeket bocsáthat ki.

Ha egy időbélyegzés-szolgáltató tanúsítványa már nem érvényes (mert lejárt vagy visszavonták), az adott tanúsítvány szerinti időbélyegek érvényessége PKI alapon nem, legfeljebb csak az időbélyegzés-szolgáltató naplófájljai segítségével bizonyítható.

Ha egy aláíráson időbélyeget helyeztünk el, az időbélyegen új, esetleg más forrásból származó időbélyeget kell elhelyeznünk, ha azt szeretnénk, hogy az aláírás érvényessége hosszú távon is bizonyítható maradjon.

Ha hosszú távú archiválásban gondolkozunk, számolnunk kell azzal, hogy a tudomány és a technológia fejlődése miatt az aláírás készítésekor még biztonságosnak tekintett (aláíró vagy hash) kriptográfiai algoritmusokban megrendülhet a biztonság, és egy korábban még biztonságos aláírás (évekkel, évtizedekkel) később hamisíthatóvá válik. Ha azt szeretnénk, hogy az aláírás érvényessége hosszú távon is bizonyítható maradjon, az aláírt dokumentumon és a rajta lévő időbélyegeken új, fejlettebb technológiával készült időbélyeget kell elhelyeznünk, mielőtt a korábbi technológiában megrendül a biztonság.

Összefoglalva:

- Az aláírás érvényessége akkor bizonyítható, ha az aláíró aláíráskor használt tanúsítványa még érvényes, vagy
- ha más módon, például érvényes időbélyeg segítségével bizonyítani tudjuk, hogy az aláírás akkor készült, amikor az aláíró tanúsítványa még érvényes volt.
- Az időbélyegen is (műszaki szempontból nézve) aláírás van, így egy időbélyeg érvényessége akkor bizonyítható, ha az időbélyegző tanúsítványa még érvényes, vagy
- ha más módon, például egy másik érvényes időbélyeg segítségével bizonyítani tudjuk, hogy az időbélyeg akkor készült, amikor az időbélyegző tanúsítványa még érvényes volt.
- Az időbélyegzők lejárta és a technológia fejlődése miatt rendszeresen – kb. néhány évenként – újra kell időbélyegeznünk az aláírásokat, hogy érvényességük bizonyítható maradjon.

E problémakörről a későbbiekben részletesebben is szólunk (8. fejezet).

6.7. Aláírás megsemmisítése

Az aláírás egy bitsorozat, egy fájl egy számítógépen. Az aláírást úgy lehet megsemmisíteni, hogy a fájlt letöröljük. A törlés általában csak logikai törlést jelent, ha egészen biztosak akarunk lenni benne, hogy a fájl megsemmisült, akkor fizikai törlést, például többszörös felülírást szokás használni. A szakirodalom számos módszert tartalmaz ezzel kapcsolatban. [74]

Az elektronikus aláírás megsemmisítése annyiban speciális, hogy az elektronikusan aláírt dokumentum, mint adat *hiteles*. E hitelesség független attól, hogy a fájl hol, kinek a gépén van, ki fér hozzá, és mennyire biztonságosan őrzik. Az elektronikusan aláírt dokumentum *minden egyes másolata is hiteles*, a másolat egyenértékű az „eredetivel". (Sőt, „eredeti" példány fogalmáról nemigen beszélhetünk elektronikusan aláírt esetben, ekkor minden egyes másolat eredetinek számít.)

Ha elektronikusan aláírt dokumentumot szeretnénk megsemmisíteni, minden egyes példányát meg kell semmisítenünk. Elég, ha egyetlen példány kiszivárog, az aláírt, hiteles dokumentum szivárgott ki, és ennek hitelessége igazolható is. E követelményt az elektronikusan aláírt dokumentumokat kezelő rendszerekkel lehet biztosítani, e rendszerek tervezésekor kell gondolni kell rá, hogy ezt hogyan kezeljük majd.

6.8. Elektronikus aláírási szabályzat

Aláírási szabályzat alatt olyan dokumentumot értünk, amely a hitelesítések, aláírások rendjét szabályozza valamely szervezeten vagy folyamaton belül. Például meghatározhatja, hogy egy szervezet mely munkatársa milyen dokumentumokat írhat alá, ki és mekkora kötelezettséget vállalhat a szervezet nevében, vagy milyen esetben mely osztályoknak kell jóváhagynia egy szerződést.

Az elektronikus aláírási szabályzat az elektronikus hitelesítések rendjét szabályozza. Általában külön dokumentum, de a hagyományos, kézzel írott aláírásokra vonatkozó aláírási szabályzat részét is képezheti. Elvileg az elektronikus aláírásokkal kapcsolatban is ugyanazon pontokat kellene szabályozni, mint a kézzel írott aláírásokkal kapcsolatban, valamint szerencsés, ha az elektronikus és a kézzel írott aláírásokra vonatkozó szabályozás egységes. Ugyanakkor az elektronikus aláírásokkal kapcsolatban számos technikai kérdést is szabályozni kell, és ma még jellemzően csak néhány ponton használ egy szervezet elektronikus aláírást, így az elektronikus aláírási szabályzat ma általában külön dokumentum, és e néhány felhasználási pont technikai kérdéseit szabályozza.

Egy elektronikus aláírás érvényessége nem objektív, csak valamilyen követelményrendszer, aláírási szabályzat (signature policy[30]) kontextusában beszélhetünk róla.

> *ETSI TR 102 272 " The Signature Policy is a set of rules for the creation and validation of an electronic signature, under which the signature can be determined to be valid. A given legal/contractual context may recognize a particular signature policy as meeting its requirements. "*

Az elektronikus aláírási szabályzat elektronikus aláírások létrehozására és ellenőrzésére vonatkozó szabályokat határoz meg, amelyek szerint az aláírás érvényessége vizsgálható. Az -EPES alapú (6.4.1.2.2. fejezet) XAdES és CAdES aláírások meghivatkozhatják, hogy milyen aláírási szabályzat szerint készültek – tartalmazzák a szabályzat azonosítóját, és lenyomatát – így később sem lehet vita róla, hogy mely szabályzat szerint kell ellenőrizni őket.

Az aláírási szabályzat általában szöveges dokumentum, de az *elektronikus aláírási szabályzatok egyes részeit géppel értelmezhető formátumban is le lehet írni*, így egy aláírás-ellenőrző alkalmazás az aláírás ellenőrzése során figyelembe tudja venni a szabályzatban leírtakat. Az ETSI TR 102 272 ASN.1 DER formátumú[31], az ETSI TR 102 038 pedig XML formátumú, géppel értelmezhető aláírási szabályzat formátumot határoz meg. [46], [45] E dokumentumok elsősorban az aláírás technikai ellenőrzéséhez szükséges megkötéseket tárgyalják, míg a szervezeti vonatkozásokra sokkal kisebb hangsúlyt tesznek.

Az aláírási szabályzat mind az aláíróra, mind az aláírást ellenőrző érintett félre vonatkozik. A benne szereplő szabályok egyik része az aláírás-létrehozó, illetve -ellenőrző alkalmazás működésére tesz megkötéseket. Egy másik része a megbízható szolgáltatók (hitelesítés-szolgáltatók, időbélyegzés-szolgáltatók és attribútum-kibocsátók), illetve hitelesítési rendjeik és időbélyegzési rendjeik körére tesz megkötéseket. A szabályok harmadik része pedig a felhasználókra, valamint technikai és szervezeti környezetükre vonatkozik.

A következőkben áttekintjük, hogy milyen kérdéseket célszerű szabályozni egy aláírási szabályzat keretében.

6.8.1. A szabályzat azonosítása

A szabályzat tartalmazza az azonosítóját (OID), verziószámát, a szabályzat kibocsátásának dátumát, a szabályzat kibocsátójának megnevezését és elérhetőségét.

6.8.2. A szabályzat hatálya

A szabályzat tartalmazza saját személyi, tárgyi és időbeli hatályát.

[30] Az angol signature policy elnevezést az aláírási szabályzat mellett aláírási politikának is fordítják.
[31] E formátum RFC 3125 néven is megjelent. [141]

Célszerű rögzíteni, hogy milyen szervezetekre, folyamatokra, és milyen felhasználókra vonatkozik a szabályzat, az aláírásokkal milyen típusú dokumentumokat hitelesítenek, és az aláírások mit jelentenek.

6.23. Példa: *A szabályzat szerinti aláírás szolgálhat például szerződéskötésre, valamely hivatalnak szolgáló adatszolgáltatásra, de igazolhatja például egy dokumentum átvételét is.*

Célszerű rögzíteni, hogy ki készíti az aláírásokat, és ki fogadja be őket. A szabályzatnak szokott lenni érvényességi időtartama – amikor a szabályzat szerinti aláírások készülhetnek. Az érvényességi időtartam vége általában nem meghatározott.

Szintén célszerű rögzíteni, hogy milyen jogszabályok vonatkoznak a szabályzat szerinti aláírásokra. A jogkövetkezménnyel bíró elektronikus aláírásokra mindenképpen vonatkozik az Eat., de emellett más jogszabályok is vonatkozhatnak rá; akár az aláírások létrehozására, akár a megőrzésére.

6.8.3. Milyen biztonsági szintű aláírásokat követelünk meg?

Amennyiben csak minősített elektronikus aláírásokra vonatkozik a szabályzat, célszerű rögzíteni, hogy milyen tranzakciós limitet (4.8.3.3. fejezet) követelünk meg.

Ha a szabályzat szerint fokozott biztonságú elektronikus aláírások is elfogadhatóak, célszerű meghatározni, hogy ezen belül milyen fokozott biztonságú elektronikus aláírások tartoznak a szabályzat hatálya alá.

6.24. Példa:

- *Csak a személyes regisztráció során kibocsátott tanúsítványokra épülő aláírások.*
- *Csak a természetes személy számára kibocsátott tanúsítványokra épülő aláírások.*
- *Csak a nyilvánosan működő hitelesítés-szolgáltatók tanúsítványaira épülő aláírások.*
- *Csak a közigazgatási követelményeknek is megfelelő aláírások. (Sajnos, ezek nemigen terjedtek el.)*
- *Csak a megadott technikai követelményeknek (pl. OCSP szolgáltatás) is megfelelő aláírások.*

6.8.4. Technológiai követelmények

6.8.4.1. Aláírás-formátum

Aki elektronikusan aláírt dokumentumot szeretne fogadni, annak célszerű meghatároznia és a küldő fél tudomására hoznia, hogy milyen formátumú elektronikus aláírást fogad el. Különben abba a helyzetbe kerülhet, hogy olyan aláírást kap, amelyet nem tud elolvasni.

Célszerű rögzíteni a konténer formátumát, valamint az aláírás-blokk formátumát is, ha az a konténerből nem következik.

6.25. Példa: *Gyakori választás az e-akta vagy PDF.*

Viszonylag ritka, hogy valaki egyszerű XML aláírást fogadna be, ehhez az is szükséges, hogy mind az aláírást létrehozó, mind a befogadó félnek legyen egyszerű XML aláírások kezelésére alkalmas és egymással interoperábilis szoftvere, amelyek egyformán kezelik az XML aláírásokat. A tapasztalat azt mutatja, hogy e problémát önmagában nem oldja meg, ha az aláírás-létrehozó alkalmazások interoperabilitása biztosított (ilyen kezdeményezés pl. a MELASZ-Ready), mert ezen túl a két oldal XML tartalmat kezelő alkalmazásainak is együtt kell működnie egymással. Álláspontunk szerint egy közös konténer-formátumra is mindenképpen szükség van.

6.8.4.2. Kriptográfiai algoritmusok, kulcsméretek

Az aláírási szabályzat általában meghatározza az aláírások készítéséhez – beleértve a tanúsítványok, időbélyegek stb. aláírásához is – használt kriptográfiai algoritmusok és paraméterek (pl. kulcsméretek) körét. Ezt nehéz jól meghatározni, ezért célszerű valamilyen létező szabályt vagy ajánlást meghivatkozni.

6.26. Példa:

- *Hivatkozhatunk a Nemzeti Média- és Hírközlési Hatóság Eat. 18. §-a szerinti határozatában foglalt algoritmusok körére. (Eat. hatálya alá tartozó aláírások esetén csak ezek közül szabad válogatnunk.)*
- *Hivatkozhatunk például az ETSI TS 102 176 (ALGO paper) dokumentumára, amely az ETSI algoritmusokkal kapcsolatos ajánlásait tartalmazza. [57]*
- *Gyakori, hogy az RSA algoritmust jelölik meg valamely lenyomatképző algoritmussal (pl. SHA-1, SHA-256).*

Nem célszerű az aláírási szabályzatban túlságosan mély megkötéseket tenni az algoritmusok körére, mert egy esetleges algoritmus-váltás úgy nehezebb feladattá válik. A Hatóság Eat.

18. §-a szerinti határozatában szereplő algoritmusokat és paramétereket célszerű előírni, és a hitelesítés-szolgáltatóktól és az aláírás-létrehozó alkalmazások fejlesztőitől megkövetelni, hogy feleljenek meg a jogszabályoknak.

6.8.4.3. PKI követelmények

Célszerű rögzíteni a megbízható gyökerek körét.

6.27. Példa:

- *A hazai szolgáltatók gyökerei és a Közigazgatási Gyökér Hitelesítés Szolgáltató.*
- *Az Eat. szerint elfogadott, nyilvánosan működő – hazai és külföldi – szolgáltatók gyökerei. Ezek listájának pontos feltérképezése és a lista karbantartása nehéz feladat, az EU-s bizalmi listák (5.4.2. fejezet) kiindulási alapot jelenthetnek, bár azok nem feltétlenül tartalmazzák magukat a gyökereket.*
- *Valamely, zárt körben használható szolgáltató gyökér, például a saját vállalati gyökerünk.*

Célszerű rögzíteni a tanúsítványláncokra (köztes tanúsítványokra) vonatkozó esetleges megszorításokat. (Lásd: 5. fejezet.)

Célszerű rögzíteni, hogy milyen hitelesítési rendek (4.1.3. fejezet) szerinti tanúsítványokat és milyen időbélyegzési rendek (7.7. fejezet) szerinti időbélyegeket fogadunk el.

6.28. Példa: *A minősített szolgáltatók hatósági nyilvántartásban szereplő rendjei, és azok összes korábbi változata.*

Célszerű rögzíteni a visszavonási állapot ellenőrzésének módját a végfelhasználói és köztes szolgáltatói tanúsítványokra, az időbélyegzés-szolgáltatói tanúsítványokra, az attribútum-kibocsátók tanúsítványaira, és az attribútum-tanúsítványokra. Célszerű rögzíteni, hogy hol és mekkora kivárási időt (6.5.4.4. fejezet) követelünk meg. Ezekhez eltérő megoldások tartozhatnak.

6.29. Példa:

- *Mindent OCSP-vel ellenőrzünk, és – a legkülső archív időbélyeg kivételével – mindig megköveteljük, hogy a visszavonási információ frissebb legyen, mint az az időpont, amire nézve ellenőrzünk. Ha az OCSP válaszadói tanúsítványban* `ocspNoCheck` *szerepel, a válaszadó visszavonási állapotát nem ellenőrizzük.*

- *A végfelhasználói tanúsítványokat OCSP-vel ellenőrizzük, és megköveteljük, hogy a visszavonási információ frissebb legyen, mint az az időpont, amire nézve ellenőrzünk. Minden más esetben CRL-t és OCSP-t egyaránt használhatunk, és csak annyit követelünk meg, hogy a visszavonási információ még ne járjon le, azaz a* `nextUpdate` *legyen későbbi, mint az időpont, amire nézve ellenőrzünk.*
- *CRL-t és OCSP-t egyaránt használhatunk, és végfelhasználói tanúsítványok esetén megköveteljük, hogy a visszavonási információ beszerzése legalább 4 órával később történjen, mint az az időpont, amelyre nézve az ellenőrzést végezzük. Minden más esetben csak annyit követelünk meg, hogy a visszavonási információ még ne járjon le, azaz a* `nextUpdate` *legyen későbbi, mint az időpont, amire nézve ellenőrzünk.*
- *CRL-t és OCSP-t egyaránt használhatunk, és csak annyit követelünk meg, hogy a visszavonási információ még ne járjon le, azaz a* `nextUpdate` *legyen későbbi, mint az időpont, amire nézve ellenőrzünk.*

6.8.5. Jogosultság-ellenőrzési követelmények

Célszerű rögzíteni, hogy mely folyamatban mely fél hogyan győződik meg a másik jogosultságáról.

6.30. Példa:

- *A tanúsítvány* `title` *mezeje alapján?*
- *A tanúsítványban szereplő* `organization` *mező alapján győződünk meg a szervezethez tartozásról?*
- *Attribútum-tanúsítvány alapján? (Ekkor célszerű rögzíteni, hogy milyen szerepkör igazolásához milyen attribútum-tanúsítványra van szükség.)*
- *Az aláíró nyilatkozata alapján?*
- *Egyéb out-of-band ellenőrzés alapján?*

6.8.6. Időbélyegzési és archiválási követelmények

Célszerű szabályozni, hogy milyen időbélyegeket fogadunk el vagy követelünk meg?

6.31. Példa: *Legegyszerűbb, ha minden minősített szolgáltató által kibocsátott időbélyeget elfogadunk, de tehetünk korlátozásokat, hogy csak meghatározott kibocsátó vagy meghatározott időbélyegzési rend szerinti időbélyegeket fogadunk el.*

Amennyiben időbélyeggel ellátott AdES aláírásokat használunk (ami az aláírások letagadhatatlansága miatt erősen javasolt), célszerű rögzíteni, hogy kinek a dolga az időbélyegzés, és ki milyen AdES típusú aláírásokat használ.

6.32. Példa:

- *Az aláíró -T típusú aláírást készít, így küldi el őket a befogadónak.*
- *Az aláíró -BES vagy -EPES aláírásokat küld a befogadónak és a befogadó helyez el időbélyeget az aláíráson.*
- *Az aláíró -T típusú aláírást küld a befogadónak, aki -A típusú aláírást készít belőle.*
- *Az aláíró egyből -A típusú aláírást küld a befogadónak.*

Ha magasabb AdES típusú aláírásokat követelünk meg, célszerű figyelembe venni, hogy a visszavonási információk összegyűjtése időt vehet igénybe, különösen ha friss visszavonási listákat követelünk meg.

Célszerű rögzíteni, hogy meddig kell megőrizni az aláírást, és hogyan kell igazolni annak hitelességét. Ha az aláírásokat archiválni kell, célszerű rögzíteni, hogy kinek a feladata:

- összegyűjteni és időbélyeggel ellátni a visszavonási információkat,
- megőrizni az aláírást, illetve az érvényességi láncot,
- rendszeresen újabb időbélyeggel látni el az érvényességi láncot,
- archiválás-szolgáltatónál elhelyezni az érvényességi láncot, ha ez szükséges.

6.9. Összegzés

- Aláírásunkkal mindig valamilyen nyilatkozatunkat hitelesítjük, például kötelezettséget vállalunk, jóváhagyunk, tudomásul veszünk vagy elfogadunk valamit.
- Az aláírásnak az a célja, hogy az aláírással ellátott dokumentumot szükség esetén egy bíróság is elfogadja az aláíró hiteles nyilatkozatának. Ez általában nem az aláíró érdeke, hanem az aláírást felhasználó, elfogadó érintett félé.
- Az elektronikus aláírásról szóló törvényünk különböző biztonsági szintű elektronikus aláírásokat határoz meg:
  - A *minősített elektronikus aláírás*sal ellátott dokumentum teljes bizonyító erejű magánokirat. A minősített elektronikus aláírás minősített hitelesítés-szolgáltató által kibocsátott *minősített tanúsítvány*ra épül, és biztonságos aláírás-létrehozó eszköz (pl. intelligens kártya) segítségével hozható létre.

  - A fokozott biztonságú elektronikus aláírással ellátott dokumentum írásba foglaltnak minősül. A fokozott biztonságú elektronikus aláírásra nagyon kevés követelmény vonatkozik.
  - A fokozott biztonságúnak nem minősülő elektronikus aláírásról mindössze annyit mond az Eat., hogy nem utasítható el pusztán azért, mert elektronikus.

- Aláíráskor nem közvetlenül az aláírt dokumentumot, hanem annak lenyomatát kódoljuk a magánkulcsunkkal.

- Legtöbbször valamilyen szabványos aláírás-blokkot (pl. XAdES, PKCS#7) hozunk létre, amely meghivatkozza, hogy mire vonatkozik az aláírás, tartalmazhat az aláírás ellenőrzéséhez szükséges információkat is. Többféle különböző aláírás-blokk létezik.

- Alkalmazásaink általában nem közvetlenül kezelik az aláírás-blokkokat, hanem ún. aláírás-konténerbe ágyazva használják őket. Konténer például az e-akta, a PDF, a Word dokumentum vagy az aláírt e-mail.

- Aláírás ellenőrzésekor:

  - Megvizsgáljuk, hogy összetartozik-e az aláírás, az aláírt dokumentum és az aláíró tanúsítványában lévő nyilvános kulcs.
  - Egy megbízható időpont szerint felépítünk egy tanúsítványláncot az aláíró tanúsítványától egy megbízható gyökértanúsítványig, és ellenőrizzük a lánc elemeinek visszavonási állapotát.
  - Megvizsgáljuk, hogy az adott folyamatban elfogadható-e az adott típusú aláírás.

- Egy aláírás érvényessége nem objektív fogalom, csak valamilyen követelményrendszer, ún. aláírási szabályzat kontextusában beszélhetünk róla.

- Az aláíráson minél hamarabb helyezzünk el időbélyeget, különben problémássá válhat az aláírás ellenőrizhetősége, ha az aláíró tanúsítványa lejár vagy visszavonásra kerül.

- Ha azt szeretnénk, hogy egy aláírás hosszú távon is hiteles bizonyíték maradjon, megfelelő módon kell tárolnunk, és például rendszeresen archív időbélyegeket kell elhelyeznünk rajta. Ezt magunk is elvégezhetjük, de archiválás-szolgáltatót is megbízhatunk vele.

7. fejezet

Időbélyegzés

„This thing all things devours:
Birds, beasts, trees, flowers;
Gnaws iron, bites steel;
Slays king, ruins town,
And beats high mountain down."

(Ez a valami mindent elemészt,
Madár, vad, fű, fa általa vész,
Vasat, acélt megrág, s a kőből,
Sziklából is lisztet őröl,
Királyt megöl, várost leront,
Magas hegyet a völgybe dönt.)

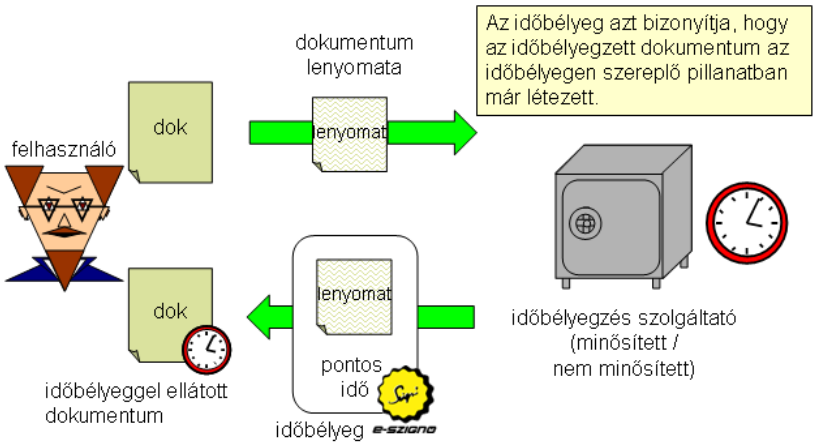
– J.R.R. Tolkien, A Hobbit, Gollam találós kérdése

Az időbélyeg (más néven időbélyegző) a nyilvános kulcsú infrastruktúra (PKI) egyik alapeleme, egy megbízható szervezet – egy időbélyegzés-szolgáltató – aláírt állítása, miszerint egy adott (lenyomatú) dokumentum egy adott időpillanatban már létezett.

Az következő módon helyezhetünk el egy dokumentumon időbélyeget (lásd: 7.1. ábra):

1. Előkészítjük azon dokumentumot[1], amelyen időbélyeget szeretnénk elhelyezni.
2. Egy program (például az e-Szignó) segítségével kiszámítjuk a dokumentum lenyomatát.
3. A lenyomatot a program elküldi az időbélyegzés-szolgáltatónak. (Az időbélyegzés-szolgáltató nem kell, hogy megkapja a teljes dokumentumot, mindössze egy néhány byte hosszú lenyomatot kell eljuttatni hozzá.)

[1] Bármilyen bitsorozaton elhelyezhető időbélyeg. Elektronikus aláírást használó alkalmazásokban jellemzően nem magán az értelmes dokumentumon, hanem az értelmes dokumentumon elhelyezett elektronikus aláíráson helyezzük el az időbélyeget. Az egyszerűség kedvéért az időbélyeggel ellátandó információt a továbbiakban dokumentumnak nevezzük.



7.1. ábra. **Az időbélyegzés-szolgáltató feltünteti az időbélyegben a kapott lenyomatot, a pontos időt (valamint egyéb információkat), majd mindezt aláírja.**

4. Az időbélyegzés-szolgáltató elkészíti az időbélyeget. A lenyomat mellé „odaírja" az aktuális időpontot, esetleg kiegészíti egyéb információkkal, majd aláírásával hitelesíti, és visszaküldi az időbélyeget kérő programnak.

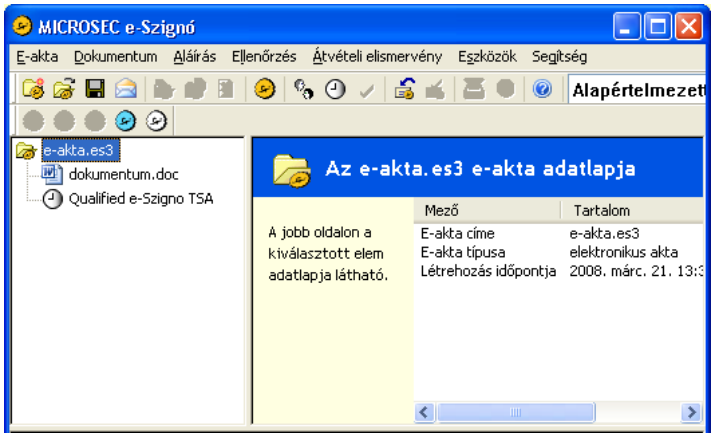
5. Az időbélyeget lekérő program felhasználja az időbélyeget, azaz például eltárolja, vagy csatolja az időbélyegzett dokumentumhoz (lásd: 7.2. ábra).

Az időbélyeg hitelességét jogszabály is elismeri. Az elektronikus aláírásról szóló törvény szerint ha egy dokumentumon minősített időbélyegzés-szolgáltató helyezett el időbélyeget, vélelmezni kell, hogy a dokumentum az időbélyegzés pillanatában már létezett (és annak kell bizonyítania állítását, aki ezt kétségbe vonja).

> *Eat. 2 § „ 16. Időbélyegző: elektronikus dokumentumhoz végérvényesen hozzárendelt vagy azzal logikailag összekapcsolt olyan adat, amely igazolja, hogy az elektronikus dokumentum az időbélyegző elhelyezésének időpontjában változatlan formában létezett. "*

Ha egy dokumentumon elektronikus aláírás, és az aláíráson egy időbélyeg szerepel, akkor nemcsak az bizonyítható, hogy a dokumentumot az a személy írta alá, akinek a tanúsítványa alapján az aláírás ellenőrizhető, hanem az is, hogy a dokumentumot az időbélyegben szereplő időpont előtt írta alá, és azóta sem a dokumentum, sem az aláírás nem változott meg.

Az időbélyegek különösen fontos szerepet töltenek be az elektronikus aláírást használó alkalmazásokban: Előfordulhat, hogy az aláíró tanúsítványát kibocsátó hitelesítés-szolgáltató visszavonja a tanúsítványt (például mert az aláíró elvesztette a magánkulcsot tartalmazó kártyáját). Ha a korábban létrehozott elektronikus aláírásait időbélyeg védi, akkor igazolható, hogy ezen aláírások még a tanúsítvány visszavonása előtt készültek, tehát érvényesek. Ha a



7.2. ábra. **Minősített időbélyeggel ellátott dokumentum az e-Szignó programban**

korábban létrehozott aláírásokon nincsen időbélyeg, akkor kétségbe lehet vonni az aláírás érvényességét; az aláírásból nem lehet megállapítani, hogy érvényes tanúsítvánnyal készültek-e. (Lásd: 6.5. fejezet.)

A fentiek is mutatják, hogy *az elektronikus aláírás csak akkor „letagadhatatlan", ha igazolható, hogy az aláírás mikor (vagy mely időpont előtt) jött létre*; ez legegyszerűbben időbélyeggel biztosítható. Ezért *az elektronikus aláírások érvényességét időbélyegekkel szokták biztosítani*, így az esetek nagy többségében *az elektronikus aláírások érvényességét időbélyegek érvényességére vezetjük vissza.*

7.1. Minősített és nem minősített időbélyegzés

Az Eat. 2 § határozza meg az időbélyeg fogalmát. Időbélyeget időbélyegzés-szolgáltató bocsát ki az Eat. 6 § (1) szerinti elektronikus aláírással kapcsolatos szolgáltatás, az úgy nevezett időbélyegzés-szolgáltatás keretében.

A Nemzeti Hírközlési Hatóság nyilvántartásában minősített időbélyegzés-szolgáltatóként szereplő szolgáltatók által kibocsátott időbélyegeket *minősített időbélyeg* néven is szokás nevezni. A minősített időbélyegekhez az alábbi jogkövetkezmény kapcsolódik:

> *Eat. 4 § „ (6) Amennyiben az időbélyegzést olyan szolgáltató végezte, amely az időbélyegzéskor e szolgáltatás tekintetében a szolgáltatók nyilvántartásában minősítettként szerepelt és az időbélyegző ellenőrzésének eredményéből más nem következik, az ellenkező bizonyításáig vélelmezni kell, hogy a dokumentumban foglalt adatok az időbélyegző elhelyezése óta változatlan formában léteztek. "*

A jogszabályban nem szerepel a minősített időbélyeg kifejezés, de a fogalom precíz elnevezése – olyan szolgáltató által kibocsátott időbélyeg, amely a szolgáltatók nyilvántartásában

az időbélyegzés-szolgáltatás tekintetében minősített szolgáltatóként szerepel – túlságosan körmönfont, nehézkes. A továbbiakban mi is a „minősített időbélyeg" kifejezést használjuk e fogalomra.

A minősített időbélyeghez kapcsolódó jogkövetkezmény analóg a minősített elektronikus aláíráshoz kapcsolódó jogkövetkezménnyel (6.1. fejezet): Ha egy dokumentumon (vagy bitsorozaton) *érvényes* minősített időbélyeg szerepel, akkor abból kell kiindulni, hogy az adott dokumentum (vagy bitsorozat) létezett az időbélyegben szereplő időpontban. Ha valaki ezzel ellentéteset állít, neki kell bizonyítania az állítását.

A minősített elektronikus aláírásnál kevésbé szigorú követelményeknek megfelelő fokozott biztonságú elektronikus aláíráshoz az Eat. jogkövetkezményt kapcsol, az így hitelesített dokumentum írásba foglaltnak minősül. *A nem minősített időbélyeghez*[2] *viszont nem kapcsol jogkövetkezményt az Eat*, így a törvény szerint nincs a fokozott biztonságú elektronikus aláíráshoz hasonló biztonsági szintű időbélyeg.

Az Eat. nem ejt szót a nem minősített időbélyegekről, de levezethető belőle, hogy az elektronikus aláírással kapcsolatos szolgáltatások mindegyike nyújtható minősített és nem minősített szolgáltatásként. Egyedül a fokozott biztonságú aláíráshoz (amelynek a nem minősített hitelesítés-szolgáltatás keretében kibocsátott tanúsítványra alapuló elektronikus aláírás minősül) rendel jogkövetkezményt, így a többi elektronikus aláírással kapcsolatos szolgáltatásnak – köztük a nem minősített időbélyegzés-szolgáltatásnak – a törvény szerint nem sok értelme van. [180] Ettől még egy nem minősített időbélyeg is felhasználható bizonyítékként, csak az Eat. nem határozza meg, hogy milyen erősségű bizonyítékot jelent, és a bíróság úgy is dönthet, hogy egyáltalán nem veszi figyelembe.

Az Eat. szerint a nem minősített időbélyegzés-szolgáltatónak is HSM-mel kell védenie a magánkulcsát, így a nem minősített időbélyegzés-szolgáltatás nem nyújtható lényegesen olcsóbban, mint minősített párja. Ma Magyarországon nincs olyan jogszabály, amely megköveteli ugyan az időbélyeg alkalmazását, de a nem minősített időbélyeget is elfogadja. A fentiekből következően úgy látjuk, a nem minősített időbélyegzés-szolgáltatásnak és a nem minősített időbélyegeknek a jelen jogszabályi környezetben nem sok értelme van.

Mindennek ellenére, a Nemzeti Hírközlési Hatóság által vezetett nyilvántartásban szerepelnek nem minősített időbélyegzés-szolgáltatók, és a Hatóság által közzétett statisztikák szerint e szolgáltatók bocsátanak is ki időbélyegeket, bár minősített időbélyegből lényegesen többet bocsátanak ki.

Az elektronikus aláírásról szóló 1999/93-as számú EU irányelv nem szól időbélyegekről, így időbélyegek esetén nem beszélhetünk arról, hogy az egyik tagállamban kibocsátott

[2] A jogszabályban nem szerepel a „fokozott biztonságú időbélyeg" kifejezés, a „fokozott biztonságú" jelzőt a jogszabály csakis és kizárólag az elektronikus aláíráshoz kapcsolja. A jogszabályban a „nem minősített időbélyeg" kifejezés sem szerepel, de a fogalom precíz elnevezése – olyan szolgáltató által kibocsátott időbélyeg, amely az időbélyegzés-szolgáltatás tekintetében nem szerepel a szolgáltatók nyilvántartásában minősített szolgáltatóként – olyan körülményes, hogy helyette mégis a „nem minősített időbélyeg" kifejezést használjuk.

időbélyegeket másik tagállamban is automatikusan elfogadnák. A legtöbb tagállam szabályozása nem különböztet meg minősített és nem minősített időbélyegeket, hanem csak „időbélyegekről" beszél. A magyar szabályozás mellett például a német és az olasz használja ezen fogalmakat. Ennek ellenére, a minősített időbélyeg fogalmát nem használó tagállamokban is vonatkozhat szigorú szabályozás az időbélyegzés-szolgáltatókra, mert az elektronikus aláírások hosszú távú hitelességét külföldön is időbélyegekkel szokás alátámasztani.

A továbbiakban elsősorban a minősített időbélyegekre és a minősített időbélyegzés-szolgáltatókra vonatkozó hazai előírásokról írunk, amelyek az ETSI TS 102 023 nemzetközi specifikációra alapulnak. [52]

7.2. Időbélyeg készítése

Ha időbélyeget szeretnénk, időbélyegzés-szolgáltatóhoz kell fordulnunk. Lenyomatot képzünk az időbélyeggel ellátandó dokumentumból, a lenyomatot elküldjük a szolgáltatónak, aki visszaküldi az időbélyeget. (lásd: 7.1. ábra)

Szigorúan véve az időbélyeg annyit igazol, hogy egy adott lenyomat egy adott időpontban rendelkezésre állt a szolgáltatónál. Az időbélyegből – műszaki értelemben – nem következik, hogy létezett is olyan lenyomatú dokumentum. Ugyanakkor egy időbélyeget csak úgy lehet felhasználni, ha meg is mutatjuk, hogy melyik dokumentumhoz tartozik. Az időbélyeggel ekkor igazolható, hogy az adott dokumentum már létezett az adott időpontban. Egy dokumentum nélküli, magányos időbélyeg semmit sem igazol.

> ***7.1. Példa:*** *Alajos generál egy véletlen 20 byte hosszú bitsorozatot, amit – mint SHA-1 lenyomatot – elküld az X időbélyegzés-szolgáltatónak. Visszakap egy időbélyeget. Alajos nem ismer olyan dokumentumot, amelynek ez a 20 byte lenne az SHA-1 lenyomata. Lehet, hogy vannak olyan bitsorozatok, amelyeknek ez a 20 byte a lenyomata, de lehet, hogy egyik sem értelmes dokumentum, és az is lehet, hogy egyetlen ilyen bitsorozat sem létezik. Ha nem találunk ilyen dokumentumot – márpedig, ha a hash függvény ősképellenálló, akkor a lenyomat alapján reális erőforrásokkal nem találunk ősképet – akkor ez az időbélyeg semmit nem bizonyít, és Alajos értelmetlenül kérte le.*

Általában az RFC 3161 specifikációban leírt protokoll szerint szerezzük be az időbélyeget. Az RFC 3161 leírja, hogy milyen formátumú időbélyeg-kérést küldhetünk az időbélyegzés-szolgáltatónak, milyen formátumú választ kapunk rá (e válasz maga az időbélyeg), és javaslatot tesz rá, hogy a kérést milyen módon küldhetjük el a szolgáltatónak, illetve milyen módon kaphatjuk vissza a választ.

A legtöbb szolgáltató az RFC 3161 által is javasolt HTTP-n keresztül bocsátja ki az időbélyegeket. Ha az időbélyegért díjat számít fel, valamilyen módon azonosítani szeretné

az időbélyeget kérő felet. Ez történhet pl. HTTP basic autentikációval (felhasználónév-jelszó alapon), vagy tanúsítvány-alapú autentikációval. Utóbbi esetben a HTTP-t SSL-en (10.3.2. fejezet) keresztül kell fűzni, de ez az előbbi esetben is célszerű, hogy a jelszó ne nyíltan utazzon a hálózaton. Olyan szolgáltató is van, amelyik minden ügyfelének külön URL-t biztosít, és az URL alapján állapítja meg, hogy ki kérte le az időbélyeget.

Az időbélyegzés nem a pontos idő lekérdezésére, hanem egy hiteles időpont igazolására szolgál. Vegyük figyelembe, hogy amikor elküldjük a lenyomatot az időbélyegzés-szolgáltatónak, időbe telik, amíg a lenyomat eljut a szolgáltatóhoz. A szolgáltató beszerzi a pontos időt az időforrásából, és összeállítja az időbélyegbe kerülő adatblokkot. Ezt követően időt vesz igénybe, amíg kiszámítja az aláírást, és az is időbe telik, amíg az aláírt időbélyeg eljut az időbélyeget kérő félhez. Ebből következik, hogy:

- Az időbélyegben szereplő időpont egy Δt_1 idővel későbbi, mint a lekérdezés időpontja.
- Amikor az időbélyeget megkapjuk, egy Δt_2 idő már mindenképpen eltelt az időbélyegben szereplő időpont óta.

Az időbélyegzés-szolgáltató garantáltan a pontos időt írja az időbélyegbe, így az időbélyegben szereplő időpont *hiteles*. Ugyanakkor mire az időbélyeg megérkezik a lekérdezőhöz, a benne szereplő időpont már nem a pontos időt tartalmazza. Ha HTTP-n keresztül szerezzük be az időbélyeget, akkor az időbélyeg beszerzésének ideje általában kevesebb, mint az időbélyeg – hazai előírás szerinti – 1 másodperces pontossága. Ugyanakkor előfordulhatnak olyan alkalmazások, amikor ez már nem elfogadható. Az RFC 3161 a HTTP mellett az e-mailt javasolja még az időbélyegek beszerzésére. E-mail esetén akár jelentős késleltetés is előfordulhat.

> ***7.2. Példa:*** *Az X időbélyegzés-szolgáltatóhoz RFC 3161 szerinti időbélyeg-kéréseket lehet beküldeni, és a szolgáltató RFC 3161 formátumú időbélyeget ad válaszul. A szolgáltatóhoz postán, floppy disken lehet beküldeni a kérést, a szolgáltató lovas futárral küldi vissza a disket az időbélyeggel. Általában 2 napba telik, mire eljut egy kérés a szolgáltatóhoz, és szintén 2 napba, mire visszatér a válasz. Az X szolgáltató az RFC 3161 szerint helyesen működik, mert igazolja, hogy egy adott időpontban egy adott lenyomat rendelkezésre állt. (Ennek ellenére nem igazán praktikus a szolgáltatásnak ez a változata.)*

7.3. Az időbélyeg formátuma

Az időbélyeg egy olyan aláírt adat, amely tartalmazza az időbélyegzett dokumentum lenyomatát, az időbélyegzés időpontját, és az időbélyeget egy időbélyegzés-szolgáltató

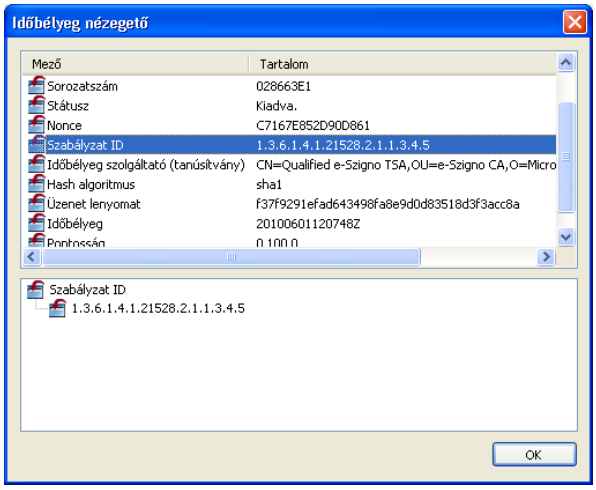
hitelesíti saját aláírásával. Az időbélyegzés-szolgáltatók többsége az RFC 3161 által meghatározott ASN.1 DER formátumú időbélyegeket bocsát ki, de létezhetnek más, pl. XML alapú időbélyeg-formátumok is. A továbbiakban az RFC 3161 által meghatározott időbélyeg-formátumot mutatjuk be. [142]

Az RFC 3161 szerinti időbélyeg egy ASN.1 DER kódolású struktúra, amely a következőket tartalmazza(Lásd: 7.3. ábra.):

- az időbélyeg-formátum verziószáma (az RFC 3161 szerint ez mindig `v1`);
- azon időbélyegzési rend (7.7. fejezet) azonosítója (OID), amely szerint az adott időbélyeget kibocsátották, és amely meghatározza, hogy az adott időbélyeg mire használható fel;
- az időbélyegzett dokumentum lenyomata;
- az időbélyeg sorozatszáma, ami egyedileg azonosítja[3] az időbélyeget;
- az időbélyeg létrehozásának időpontja az időbélyegzés-szolgáltató belső órája alapján, ez az az időpont, amire az időbélyeg vonatkozik, az időbélyegzés-szolgáltató közvetlenül ezen időpontot követően írja alá az időbélyeget; ez egy UTC (Universal Time, Coordinated) idő;
- az időbélyeg pontossága; a hazai minősített szolgáltatók legalább másodperc pontosságot vállalnak;
- az a nonce elem, amelyet a kérdés is tartalmazott (feltéve, hogy a kérdésben volt nonce elem);
- az időbélyegzés-szolgáltató tanúsítványa;
- az időbélyegzés-szolgáltató CMS (6.4.1.3. fejezet) formátumú aláírása. [153]

Az RFC 3161 az időbélyegben szereplő aláírt `TSTInfo` struktúrát határozza meg, aláírás formátuma tekintetében a CMS formátumot leíró RFC 2630-at (amelyet azóta az RFC 5652 váltott fel) hivatkozza meg, ő maga semmit nem ír az aláírás formátumáról. Leírja viszont, hogy a CMS aláírással kiegészített `TSTInfo` struktúrát hogyan kell becsomagolni az időbélyeget leíró `TimeStampResp` struktúrába.

[3] A sorozatszám nem feltétlenül szigorúan monoton növekszik az egymás után kibocsátott időbélyegeken, de feltétlenül *egyedinek* kell lennie.



7.3. ábra. **Időbélyeg tartalmának megjelenítése az e-Szignó programban**

7.4. Időbélyegzés-szolgáltató

Az időbélyegzés-szolgáltató a PKI szempontjából egy speciális végfelhasználó, akinek aláírt állításait az érintett felek elfogadhatják bizonyos megbízható időpontok igazolására.

Az időbélyegzés-szolgáltatóra a hitelesítés-szolgáltatóknál leírtakhoz nagyon hasonló követelmények vonatkoznak (4. fejezet). Az időbélyegzés-szolgáltatónak biztonságos, megbízható rendszert kell üzemeltetnie, fizikailag biztonságos környezetben, tanúsított kriptográfiai modulokban (HSM) kell tárolnia kulcsait, meghatározott bizalmi munkakörökkel kell rendelkeznie, és megbízható, auditálható szervezeti környezetet kell kialakítania. Az időbélyegzés-szolgáltató felelősséget vállal a kibocsátott időbélyegekért, ezért stabil pénzügyi háttérrel kell rendelkeznie, és jogszabályban leírt pénzügyi követelményeket kell teljesítenie.

Az időbélyegzés-szolgáltató olyan speciális tanúsítvánnyal rendelkezik, amelyről az érintett felek felismerhetik, hogy az időbélyeget olyan megbízható fél írta alá, akinek elhihetik, hogy az időbélyegben feltüntetett lenyomatú dokumentum az időbélyegben feltüntetett időpontban már létezett. Az időbélyegzés-szolgáltató tanúsítványát kibocsátó hitelesítés-szolgáltató a `timeStamping` kiterjesztett kulcshasználatot helyezi el az időbélyegzés-szolgáltató tanúsítványában, így jelöli, hogy időbélyegzés-szolgáltatóról van szó.

Az időbélyegzés-szolgáltató mindig valamilyen időbélyegzési rend (7.7. fejezet) szerint bocsátja ki az időbélyeget. Az egyes időbélyegekben feltünteti az időbélyegre vonatkozó időbélyegzési rend azonosítóját.

7.5. Időbélyeg ellenőrzése

Az időbélyeg hitelességét az időbélyegzés-szolgáltató *aláírása* igazolja. Az elektronikus aláírásról szóló törvény szerint *az időbélyegen szereplő aláírás nem elektronikus aláírás.* Így, ha egy időbélyegzés-szolgáltató időbélyeget helyez el egy szerződésen, azzal nem „írja alá" a szerződés tartalmát, nem fogadja el a benne szereplőket magára nézve kötelezőnek, egyedül annyit igazol, hogy az adott szerződés az időbélyegben szereplő időpontban már létezett. Az időbélyegen szereplő aláírás mást jelent, mint az elektronikus aláírás, de ugyanazon technológiára épül, műszakilag ugyanúgy jön létre, és ellenőrizni is pontosan úgy kell, mint egy elektronikus aláírást.

Egy időbélyeg ellenőrzése során:

1. Ellenőrizzük, hogy összetartozik-e az időbélyegzett dokumentum az időbélyeggel és az időbélyegzés-szolgáltató tanúsítványával.

2. Ellenőrizzük az időbélyegen szereplő aláírást (6.5. fejezet), azaz:

   a. Felépítjük a hozzá tartozó tanúsítványláncot, azaz egy megbízható gyökértanúsítványra vezetjük vissza az időbélyegzés-szolgáltató tanúsítványát.

   b. Ellenőrizzük, hogy a tanúsítványlánc minden egyes eleme érvényes volt-e az időbélyeg készítésének pillanatában.

      Vigyázat, *az időbélyeget nem a benne lévő időpontra nézve ellenőrizzük!* Azért ellenőrizzük az időbélyeget, hogy megtudjuk, elhihetjük-e a benne szereplő időpontot. Ebből következik, hogy az időbélyeg ellenőrzésekor nem építhetünk arra, hogy az időbélyeg valóban a benne feltüntetett időpontban készült.

      Ahogy elektronikus aláírás ellenőrzésekor sem az aláírásban[4] feltüntetett időpontot tekintjük az aláírás készítése időpontjának, időbélyeg ellenőrzésekor sem az időbélyegben szereplő időpontot vesszük alapul. Ehelyett pontosan úgy járunk el, mint minden más aláírás ellenőrzésekor: Keresünk egy olyan $t_{control}$ időpontot, amikor az adott időbélyeg bizonyíthatóan létezett (6.5.4.2. fejezet). Ezen időpont vagy egy másik, *érvényes* időbélyegre, vagy egyéb bizonyítékra alapul, ha ilyenek nincsenek, akkor annyit tudunk, hogy az időbélyeg *most*, azaz az ellenőrzés pillanatában létezik, így az aktuális időpontra nézve ellenőrizzük.

3. Ellenőrizzük, hogy az időbélyeg megfelel az adott célra, azaz pontossága, megbízhatósága, a hozzá kapcsolódó időbélyegzés-szolgáltatói felelősségvállalás stb. megfelelő. Ez az időbélyegzési rend (7.7. fejezet) vizsgálatával végezhető el a legkönnyebben.

[4] aláírás-blokkban (6.4.1. fejezet)

Látható, hogy az időbélyeg ellenőrzése során pontosan ugyanúgy járunk el, mint bármely más aláírás ellenőrzése során. Időbélyeg esetén az aláíró egy megbízható időbélyegzés-szolgáltató, aki a magánkulcsát jól védett körülmények között (pl. HSM-ben) őrzi. Az időbélyegzés-szolgáltatói magánkulcsok lényegesen ritkábban kompromittálódnak, mint a természetes személy aláírók magánkulcsai. Így az időbélyegzés-szolgáltatók tanúsítványaira – a hitelesítés-szolgáltatók tanúsítványaihoz hasonlóan – sok esetben nem alkalmaznak kivárási időt.

> ***7.3. Példa:*** *Manfréd 2013. január 23-án megszerzi az X időbélyegzés-szolgáltató magánkulcsát. Az X időbélyegzés-szolgáltató ezt észleli, és kéri szolgáltatói tanúsítványa visszavonását, a hitelesítés-szolgáltató azonnal közzé is teszi, hogy az adott szolgáltatói tanúsítvány 2013. január 23-ától érvénytelen.*
>
> *Manfréd másnap kibocsát egy időbélyeget az X időbélyegzés-szolgáltató nevében. Birtokában van a szolgáltatói magánkulcs, tetszőleges bitsorozatot aláírhat vele, így bármilyen dátumot, bármilyen időpontot beleírhat az időbélyegbe. Olyan időbélyeget bocsát ki, amely a benne szereplő időpont szerint 2009. december 25-én készült. A hitelesítés-szolgáltató visszavonási nyilvántartása szerint az X időbélyegzés szolgáltató tanúsítványa akkor még érvényes volt, aki ezen időpontra nézve ellenőrzi az időbélyeget, az érvényesnek tekinti. Ezért hiba az időbélyeget a benne szereplő időpontra nézve ellenőrizni.*

Vegyük figyelembe ugyanakkor, hogy *időbélyegzés-szolgáltató magánkulcsának kompromittálódása esetén minden időbélyeg visszamenőleg is megkérdőjelezhetővé válik.* Ez azt jelenti, hogy kompromittálódott szolgáltatói kulcs esetén önmagából az időbélyegből nem állapítható meg, hogy nem egy támadó által kibocsátott visszadátumozott időbélyeggel állunk-e szemben. (Az időbélyegzés-szolgáltató kulcsának kompromittálódása esetén egy időbélyeg hitelessége legfeljebb az időbélyegzés-szolgáltató naplófájljai alapján állapítható meg.) Habár az időbélyegzés-szolgáltató kulcs kompromittálódása valószínűtlen esemény, óriási kárt okoz, ha bekövetkezik, ezért sok esetben mégis konzervatív módon, kivárási idővel szokás ellenőrizni az időbélyegeket.

Létezik olyan eset, amikor nem lehet következetesen kivárási időt alkalmazni időbélyeg ellenőrzésekor. Ilyen például a XAdES-A aláírásokban lévő „legkülső” archív időbélyeg ellenőrzése. Ezen időbélyeg kizárólag az ellenőrzés időpontjára[5] (*most*) nézve ellenőrizhető. Így a *most* időpontban kell friss (`thisUpdate`=*most*) visszavonási információt lekérnünk, majd ellenőriznünk kell rajta az aláírást. Ha következetesen járunk el, itt is kivárási időt alkalmazunk, azaz a visszavonási információn lévő aláírásra is friss visszavonási információt kérünk le... majd az új visszavonási információn lévő aláírásra is friss visszavonási információt kérünk le... stb.

E probléma például úgy oldható fel, hogy:

[5] Ha lenne rajta másik időbélyeg, már nem ő lenne a „legkülső”.

- Nem alkalmazunk kivárási időt a legkülső időbélyegre.
- Kivárási időt alkalmazunk, lekérjük a friss visszavonási információt, de a visszavonási információn lévő aláírásra már nem alkalmazunk kivárási időt.
- Kivárási időt alkalmazunk, lekérjük a friss visszavonási információt, és nem ellenőrizzük a visszavonási információt aláíró szolgáltató tanúsítványának visszavonási állapotát. A `thisUpdate`=*most* visszavonási információ OCSP segítségével kérhető le. Az OCSP válaszadók tanúsítványában gyakran szerepel `ocspNoCheck` kiterjesztés, ami azt jelenti, hogy a válaszadó tanúsítványa annyira rövid lejáratú, hogy nem vonatkozik rá visszavonási információ.

7.6. Időbélyeg és időjelzés

Az időbélyeg (timestamp) mellett sok specifikáció az időjelzés (time mark) használatát is megengedi. Igaz, az egyes specifikációk eltérő fogalmat értenek az „időjelzés” kifejezés alatt:

- Biztonságos naplófájlban szereplő bejegyzést, amely valamilyen (lenyomatú) adat vagy dokumentum létezését igazolja egy adott időpontra vonatkozóan. [51] Az ilyen időjelzés nem jelenik meg az aláírás mellett, de az aláírás-ellenőrző alkalmazás figyelembe veheti az aláírás ellenőrzése során.
- Olyan, RFC 3161 szerinti struktúrát, amelyet nem időbélyegzés-szolgáltató írt alá. [96], [84]

A minősített időbélyeg fogalmát az elektronikus aláírásról szóló törvény definiálja. Az ilyen időbélyegre erős biztonsági követelmények vonatkoznak, az Eat. által meghatározott vélelem kapcsolódik hozzá, és az időbélyegzés-szolgáltató felelős az így kibocsátott időbélyegért.

Az időjelzés fogalmát a 193/2005. Kormányrendelet 19. § (2) határozza meg. A rendelet szerint közigazgatási szerv (például egy önkormányzat) saját fokozott biztonságú elektronikus aláírásával bocsátja ki az időjelzést. Nagyon kevés követelmény vonatkozik az időjelzésre; sem az időforrás minőségére, sem az időjelzéseket aláíró kulcs védelmére nem vonatkoznak előírások. Az sem egyértelmű, hogy az időjelzést kibocsátó közigazgatási szerv felel-e, és milyen módon felel az időjelzésért. Az időjelzéshez nem kapcsolódik hozzá vélelem, és legfeljebb közigazgatáson belül használható.

Lényeges, hogy rendszereink elkülönítsék e két fogalmat, mert nagyon különböző biztonsági szintet testesítenek meg. Egyik szabványos megoldás, ha az aláírás-ellenőrző alkalmazás megvizsgálja, hogy az időbélyeg ellenőrzésére használható tanúsítványban szerepel-e a `timeStamping` kiterjesztett kulcshasználat, mert e kulcshasználat – elvileg – csak időbélyegzés-szolgáltató tanúsítványában tüntethető fel.

> **Megjegyzés:** Ha az hitelesítés-szolgáltatók időjelzésre használt tanúsítványban is feltüntetik a `timeStamping` kiterjesztett kulcshasználatot, nehéz lesz elkülöníteni az időjelzéseket az időbélyegektől. Ugyanakkor ha nem tüntetik fel e kulcshasználatot az időjelzésre használható tanúsítványokban, akkor sok alkalmazás el fogja utasítani az időjelzéseket, mert csak megbízható időbélyegzés-szolgáltató által igazolt időpontot fogadnak el. Igaz, ez nem feltétlenül probléma, mert ezt lehet, hogy tényleg nem kellene elfogadni. Álláspontunk szerint az RFC 3161 szerinti formátumú, de nem időbélyegzés-szolgáltató által kibocsátott időjelzés szerencsétlen megoldás, és nehezen illeszthető a PKI fogalmai közé.

Az időbélyegek és az időjelzések elkülönítésére célszerű ellenőrizni, hogy az időbélyegben egy elfogadott időbélyegzés-szolgáltató elfogadott időbélyegzési rendje szerepel-e.

7.7. Időbélyegzési rend

Az elektronikus aláírásról szóló törvény a következő módon definiálja az időbélyegzési rend fogalmát:

> *Eat. 2 § 24. „: [O]lyan szabálygyűjtemény, amelyben egy szolgáltató, igénybe vevő vagy más személy (szervezet) valamely időbélyegző felhasználásának feltételeit írja elő igénybe vevők valamely közös biztonsági követelményekkel rendelkező csoportja, illetőleg meghatározott alkalmazások számára. ”*

Ahogy a hitelesítés-szolgáltatók hitelesítési rend, az időbélyegzés-szolgáltatók időbélyegzési rend szerint működnek. Ahogy egy tanúsítvány elfogadásához a kapcsolódó hitelesítési rendet, úgy egy időbélyeg elfogadásához a kapcsolódó időbélyegzési rendet célszerű megvizsgálni. Az RFC 3161 szerinti időbélyeg tartalmazza azon időbélyegzési rend hivatkozását (OID), amely szerint kibocsátották az adott időbélyeget.

Az időbélyegzési rend az alábbi kérdéseket szokta tisztázni:

- Minősített vagy nem minősített időbélyegről van szó, azaz milyen jogkövetkezmény kapcsolódik hozzá.
- Hogyan védi a szolgáltató a magánkulcsát? A hazai jogszabályok szerint a magánkulcsot HSM-ben kell tárolni, és az időbélyegzés-szolgáltatónak a 3/2005. IHM rendeletben leírt követelményeknek is meg kell felelnie. [80] E követelményrendszer a hitelesítés-szolgáltatókra vonatkozó követelményrendszerrel (4. fejezet) analóg.
- Mekkora felelősséget vállal a szolgáltató az időbélyeggel okozott károkért? A minősített hitelesítés-szolgáltatóknak az Eat. megengedi, hogy korlátozzák a tanúsítvánnyal

vállalható kötelezettség legmagasabb mértékét. Időbélyegzés-szolgáltatók tekintetében nincs a törvényben ilyen rendelkezés.

- Milyen rendelkezésre állást vállal a szolgáltató? A hazai minősített időbélyegzés-szolgáltatók általában éves szinten 99,9%-os rendelkezésre állást vállalnak, ahol a leghosszabb kiesés legfeljebb 3 óra lehet.
- Mennyire pontos az időbélyeg? A hazai minősített szolgáltatók esetén ez jellemzően 1 másodperc.
- Megőrzi-e a szolgáltató a kibocsátott időbélyegeket (vagy azok lenyomatát)? Mit naplóz az időbélyegekről? Meddig őrzi meg mindezt? (Ezekre akkor lehet szükség, ha az időbélyegzés-szolgáltató magánkulcsa kompromittálódik. Ekkor legfeljebb az időbélyegzés-szolgáltató naplófájljai alapján dönthető el, hogy az időbélyeget valóban kibocsátotta-e.)

Az időbélyegzési rend általában az ETSI TS 102 023 specifikációban szereplő struktúrát és tartalmat követi. [52]

Időbélyegzés-szolgáltatók összehasonlításakor célszerű felmérni, hogy képesek-e az elvárt teljesítménnyel bocsátani ki időbélyegeket. Előfordulhat, hogy egy szolgáltató úgy vállalja a rendelkezésre állást, hogy másodpercenként csak 1-2 időbélyeg kibocsátására képes, de ez a teljesítmény sok esetben nem elegendő. Ha nagy mennyiségű időbélyegre van szükségünk, célszerű tesztelni az időbélyegzés-szolgáltatót, hogy valóban képes-e biztosítani az elvárt mennyiséget. Léteznek ingyenes, nyílt forráskódú eszközök időbélyegzés-szolgáltatók teljesítményének tesztelésére.

7.8. Összegzés

- Időbélyeg segítségével az igazolható, hogy egy adott lenyomatú dokumentum egy adott időpontban már létezett.
- Az Eat. egyedül a minősített időbélyeghez kapcsol jogkövetkezményt.
- Egy elektronikus aláírás megbízható ellenőrzéséhez ismernünk kell egy megbízható időpontot, amikor az aláírás már biztosan létezett, így az aláírások biztonsága időbélyegek biztonságára (hiteles időpontokra) épül.
- Ha „letagadhatatlan” elektronikus aláírásról beszélünk, a gyakorlatban az időbélyeget is bele kell érteni.
- Az időbélyegző egységek kompromittálódása visszamenőleg is érintheti az időbélyegeket.

8. fejezet

Elektronikus archiválás-szolgáltatás

„Stat rosa pristina nomine, nomine nuda tenemus."
(A hajdani rózsa név csupán, puszta neveket markolunk.)

– Umberto Eco, A rózsa neve

A papír alapú aláírásokhoz hasonlóan elektronikus aláírás esetén is fennáll az a probléma, hogy a „régen" készült aláírások hitelességét nem könnyű megbízhatóan ellenőrizni.

Az aláíró a tanúsítványához tartozó magánkulcs segítségével hozhat létre elektronikus aláírást. Ha aláíráskor érvényes a tanúsítványa, akkor érvényes aláírást hoz létre, és nem szeretnénk, hogy ez az aláírás később mégis érvénytelenné válhasson. Ha az aláírás később érvénytelenné válhat, akkor nem „letagadhatatlan". Attól függ, hogy az aláírásunk érvényes marad-e később is vagy sem, hogy később is tudjuk-e majd bizonyítani – műszaki szempontból — az érvényességét. (Lásd: 6.5. fejezet.)

Az aláírás érvényessége csak addig bizonyítható, ameddig az aláíró tanúsítványa érvényes. Ha az aláíró tanúsítványa érvénytelenné válik – azaz lejár vagy visszavonják – az aláírás érvényessége megkérdőjelezhetővé válhat. Az aláírás ettől természetesen nem válik meg nem történtté, de esetleg nehéz lesz bizonyítani, hogy az aláíró valóban aláírt egy adott dokumentumot. E probléma nem merül fel, azaz nem sérül az aláírás hitelessége, ha később igazolni tudjuk, hogy az aláírás már valóban létezett olyan adott időpontban, amikor az aláíró tanúsítványa még érvényes volt, azaz az aláírás érvényes tanúsítvány szerint készült. E problémára az időbélyeg alkalmazása jelenti a legnyilvánvalóbb megoldást: a még érvényes aláíráson időbélyeget helyezünk el, így igazoljuk, hogy az aláírás az időbélyegzés pillanatában már létezett.

Vegyük figyelembe, hogy műszaki szempontból az időbélyegen is aláírás van, így az időbélyeg érvényessége is csak addig bizonyítható, amíg az időbélyegzés-szolgáltató tanúsítványa érvényes. Ha az időbélyeg hitelessége sérül, az magával vonhatja, hogy az időbélyegzett aláírás

hitelessége is megsérül, hiszen az aláírás érvényességét az időbélyeg érvényességére vezettük vissza.

E problémára jelent választ az elektronikus archiválás-szolgáltatás. A papír alapú dokumentumokhoz hasonlóan az elektronikus (és elektronikusan aláírt) dokumentumokat is speciális körülmények között, biztonságosan kell tárolni, hogy a dokumentum (és a rajta lévő aláírás) ne sérüljön meg, és a dokumentum hitelessége hosszú távon – akár évtizedekig, sőt évszázadokig is – biztosítható legyen, még akkor is, ha az aláíráshoz használt technológiák időközben gyökeresen megváltoznak, és a korábban készült aláírások könnyen hamisíthatóakká válnak.

Az elektronikus aláírásról szóló törvény értelmében, ha egy aláírt dokumentumot minősített archiválás-szolgáltató archivál, akkor abból kell kiindulni, hogy az archiválást „jól" végzi.

8.1. Mitől válhat egy érvényes aláírás ellenőrizhetetlenné?

Az elektronikus aláírásról szóló törvény az *érvényes* elektronikus aláírásokhoz rendel jogkövetkezményeket, azaz ha „az aláírás ellenőrzésének eredményéből más nem következik". Amikor egy aláírás érvényességét ellenőrizzük, akkor valamilyen aláírási szabályzat szerint próbáljuk levezetni annak érvényességét, *bizonyítékok* (szolgáltatói tanúsítványok, visszavonási listák, OCSP válaszok stb.), *megbízható időpontok* és megbízható gyökértanúsítványok alapján. Ezen megbízható időpontok leggyakrabban érvényes időbélyegekből származnak. Akkor tekintünk egy aláírást érvényesnek, ha az általunk használt követelményrendszer szerint le tudjuk vezetni az aláírás érvényességét.

Az aláírások érvényessége nem objektív fogalom. Ha egy aláírás egy adott aláírási szabályzat szerint érvényes, egy más aláírási szabályzat szerint érvénytelen is lehet. (6.8. fejezet) (Például egy `Microsec e-Szigno Root CA` gyökér szerint érvényes aláírás valószínűleg nem lesz érvényes egy másik gyökér, pl. a Verisign valamely rootja szerint.) A továbbiakban egy másik, hasonló problémáról lesz szó: arról, hogy az aláírás érvényességének igazolhatósága az ellenőrzés időpontjától is függ.

Előfordulhat, hogy egy aláírást egyazon követelményrendszer (aláírási szabályzat) szerint egyik időpontban érvényesnek, egy más időpontban érvénytelennek találunk. A továbbiakban ennek lehetséges okait ismertetjük.

Ha azt szeretnénk, hogy az aláírás hosszú távon is ellenőrizhető maradjon, célszerű azt megfelelő körülmények között, például elektronikus archiválás-szolgáltatás keretében őrizni.

8.1.1. Ha az aláírás időpontja nem igazolható

Ha nem bizonyítható, hogy mikor készült az aláírás, akkor az aláírás érvényessége csak addig igazolható, amíg az aláíró tanúsítványa érvényes. Ha az aláíró tanúsítványa időközben lejárt

vagy visszavonásra került, akkor már nem lehet eldönteni, hogy az aláírás akkor készült-e, amikor a tanúsítvány még érvényes volt.

Ha az aláíró magánkulcsa az aláírás létrehozása előtt kompromittálódott, akkor lehet, hogy egy támadó készítette az aláírást (azaz például egy tolvaj, aki ellopta az aláíró intelligens kártyáját). Ha az aláíró tanúsítványa lejárt, akkor lehet, hogy már nem érhető el rá vonatkozó visszavonási információ, így nem lehet eldönteni, hogy kompromittálódott-e a magánkulcs. Az is előfordulhat, hogy az aláíró tanúsítványa lejárt, és csak ezt követően kompromittálódott a magánkulcs, de a tanúsítvány már érvénytelen volt[1], így senki nem foglalkozott a visszavonásával.

Ha az aláíró magánkulcsa nem kompromittálódott, és a magánkulcsot a tanúsítvány lejártát követően megsemmisítették, és az aláíráshoz használt kriptográfiai algoritmusok még biztonságosak, akkor elvileg a lejárt tanúsítvány alapján is igazolható lehet, hogy az aláírást a tanúsítvány birtokosa készítette. Sajnos, általában nagyon nehéz ezek mindegyikéről meggyőződni.

Ha azt szeretnénk, hogy az aláírás „letagadhatatlan" maradjon, célszerű gondoskodni az ellenőrzéséhez szükséges megbízható időpontról. *A még érvényes aláíráson célszerű időbélyeget elhelyezni, hogy később igazolható legyen egy olyan időpont, amikor az aláírás már létezett, és az aláíró tanúsítványa még érvényes volt.*

Megjegyzés:

1. A már érvénytelen aláíráson hiába helyezünk el időbélyeget, ez az időbélyeg nem sokat segít az aláírás ellenőrzésében.
2. Önmagában az, hogy egy aláíráson időbélyeg van, nem jelenti azt, hogy az aláírás érvényes volt, amikor az időbélyeget elhelyezték rajta. Érvénytelen aláíráson – elvileg – ugyanúgy el lehet helyezni időbélyeget. Igaz, egyes aláírás-létrehozó alkalmazások csak akkor helyeznek el időbélyeget az aláíráson, ha az még érvényes, de nem tudhatjuk, hogy milyen alkalmazás kérte le az időbélyeget az aláíráshoz, és az milyen beállításokkal futott.
3. Az időbélyegre azért van szükség, hogy igazolhassuk, hogy az aláírás egy adott időpontban már létezett. Elvileg bármilyen módon, bármilyen formátumban csatolt időbélyeg megfelel ennek a célnak. Azért célszerű a XAdES-T vagy CAdES-T formátumok használata, mert ekkor egy szabványos aláírás-ellenőrző alkalmazás automatizált módon is ellenőrizni tudja az aláírás érvényességét. Ugyanakkor előfordulhat, hogy valaki sok - BES aláírást összecsomagol például egy ZIP fájlba, majd e ZIP fájlon helyez

[1]Ha egy aláíró tanúsítvány lejár, és a magánkulcsot a továbbiakban nem használjuk (azaz nem újítjuk meg a tanúsítványt), célszerű megsemmisíteni a magánkulcsot.

el egyetlen időbélyeget. Ezen időbélyeg a ZIP fájlban lévő összes aláírás érvényességét igazolja, csak e tényt egy aláírás-ellenőrző alkalmazás nemigen tudja felismerni, és figyelembe venni. Az ilyen aláírások érvényességének vizsgálatához szakértői tevékenység szükséges.

Az aláíró tanúsítványának lejártára előre készülhetünk, így tudhatjuk, hogy az aláíró tanúsítványának lejárta előtt mindenképpen időbélyegeznünk kell az aláírást. Az aláíró tanúsítványának visszavonása viszont rendkívüli, előre nem tervezhető esemény, így erre nem készülhetünk. Extrém esetben az is előfordulhat, hogy az aláíró úgy próbál csalni, hogy kulcskompromittálódást jelent, majd letagadja az aláírását.

Megoldás: *Célszerű a lehető leghamarabb* – lehetőleg rögtön, a létrehozást követően – *időbélyegezni az aláírásokat.*

8.1. Példa: *Bendegúz az Interneten keresztül árul elektronikus tartalmat (pl. mozifilmeket). Alajos aláír egy megrendelést (amelyben megrendel sok-sok filmet), majd elküldi Bendegúznak. Alajos -BES aláírást készít, és arra számít, hogy Bendegúz ellenőrzi az aláírást, de nem tesz rá időbélyeget (azaz nem terjeszti ki -T aláírássá).*

Alajos megvárja, amíg Bendegúz ellenőrzi az aláírást, és teljesíti a megrendelést, elküldi Alajosnak a megrendelt elektronikus tartalmat. Alajos ekkor jelzi a hitelesítés-szolgáltatónak, hogy kompromittálódott a magánkulcsa, és a hitelesítés-szolgáltató visszavonja a tanúsítványt.

Ezt követően Alajos arra hivatkozik, hogy ő nem írta alá a megrendelést. Azt állítja, valaki ellopta a kártyáját, a tolvaj írta alá a megrendelést, és Bendegúz hibázott, mert elfogadta a már érvénytelen, visszavont tanúsítványra épülő aláírást. Ezen túl azt állítja, hogy neki nem kell a kapott elektronikus tartalom, nem hajlandó kifizetni, nem használta fel, és már le is törölte.

Ha Bendegúz tényleg nem helyezett el időbélyeget az aláíráson, nehéz lesz bizonyítania, hogy érvényes volt az aláírás, amikor ő elfogadta.

8.1.2. Ha az időbélyegen lévő aláírás hitelessége megkérdőjelezhető

Az időbélyegen műszaki értelemben szintén aláírás van, amelyet ugyanúgy lehet ellenőrizni, mint minden más aláírást. (Lásd: 7.5. fejezet.) Ebből adódóan, *ha nem bizonyítható egy adott időbélyegről, hogy az mikor készült, akkor az adott időbélyeg érvényessége is csak addig igazolható, amíg az időbélyegzés-szolgáltató tanúsítványa érvényes.*

Egy *érvényes* időbélyeg alapján elfogadhatjuk, hogy az időbélyegzett dokumentum az időbélyegben szereplő időpontban már létezett. Ugyanakkor egy időbélyeg ellenőrzése során

nem vehetjük figyelembe, nem tekinthetjük megbízható időpontnak az időbélyegben szereplő időpontot, épp úgy, ahogy egy aláírás ellenőrzése során nem vehetjük figyelembe az aláírt dokumentumban szereplő, az aláíró által állított időpontot. Ha egy időbélyeget a benne szereplő időpontra nézve ellenőrzünk, akkor hibát követünk el. Az időbélyegben szereplő időpontot csak akkor vehetjük alapul, ha már meggyőződtünk az időbélyegen szereplő aláírás érvényességéről, előtte még nem.

Ha az időbélyegzés-szolgáltató tanúsítványa érvénytelenné válik, az időbélyeg érvényessége már nem igazolható. Ha az időbélyegzés-szolgáltató magánkulcsa kompromittálódik, a támadó bármilyen időbélyeget létrehozhat. Ha az időbélyegzés-szolgáltató tanúsítványa lejár, már „nehéz" ellenőrizni, hogy nem vonták-e azt vissza a lejárat előtt kulcskompromittálódás miatt.

Ha egy aláírás érvényességét egy időbélyeg érvényességére vezettük vissza, és az időbélyeg hitelessége nem állapítható meg, akkor lehet, hogy az aláírás hitelessége sem állapítható meg többé.

> ***8.2. Példa:*** *Adott egy aláírás, amely a rajta szereplő időbélyeg szerint 2006. december 1-én már létezett. Az aláírásra vonatkozó CRL szerint az aláíró tanúsítványa 2006. december 1-én még érvényes volt, de 2006. december 10-én kulcskompromittálódás miatt visszavonásra került. Az időbélyegző tanúsítványát 2008. január 8-án kulcskompromittálódás miatt visszavonták.*
>
> *Alajos 2007. júniusában kapta meg az aláírást, az időbélyegző tanúsítványa ekkor még érvényes volt, így meg tudott győződni az aláírás érvényességéről.*
>
> *Bendegúz 2008. február 2-án kapta meg az aláírást, ő ekkor már nem tudott meggyőződni az aláírás érvényességéről.*
>
> *Manfréd, a támadó, 2008. január 8-án szerezte meg az időbélyegzés-szolgáltató magánkulcsát. Ha a feketepiacon megvásárolta az aláíró ellopott kártyáját, 2008. február 1-én pontosan ilyen aláírást tudott készíteni.*

Az aláírásokon lévő időbélyegeken célszerű új időbélyeget elhelyezni, mielőtt az eredeti időbélyeget kibocsátó időbélyegzés-szolgáltató tanúsítványa érvénytelenné válik. Ilyenkor lényeges, hogy az új időbélyeg más forrásból származzon, mint az eredeti időbélyeg. Ha az új időbélyeg ugyanazon kulccsal, ugyanazon szolgáltatói tanúsítvány alapján készül, akkor ha az eredeti időbélyeg hitelessége megkérdőjelezhetővé válik, akkor egyúttal megkérdőjelezhetővé válik az új időbélyeg hitelessége is.

> ***8.3. Példa:*** *Alajos XAdES-T aláírást készít, azaz időbélyeget helyez el az aláírásán. Az időbélyegzés-szolgáltató tanúsítványa még 10 évig érvényes. Bendegúz XAdES-A aláírást készít, amelynek keretében három időbélyeget helyez el az aláíráson. A három időbélyeg egyazon időbélyegzés-szolgáltatótól származik,*

ugyanarra az időbélyegzés-szolgáltatói tanúsítványra épül, mint az Alajos által elhelyezett időbélyeg. Mindkét aláírás esetén még 10 évig lehet bizonyítani az aláírás érvényességét igazoló megbízható időpontot. Bendegúz XAdES-A aláírásának megbízható időpontja sem igazolható tovább[2].

8.1.3. Ha nem érhető el releváns visszavonási információ

Mind az X.509, mind az RFC 5280 filozófiája szerint elsősorban azon tanúsítványok visszavonási állapotát kell ellenőrizni, amelyek még nem jártak le, hiszen a lejárt tanúsítványok úgyis érvénytelenek. Ebből következően a hitelesítés-szolgáltatóknak sem kell közzétenniük a lejárt tanúsítványok visszavonási állapotát.

ITU-T X.509, 7.3. fejezet „ Expired certificates will normally be removed from the Directory. It is a matter for the security policy and responsibility of the authority to keep old certificates for a period of time if a non-repudiation of data service is provided. ”

RFC 5280, 5. fejezet:
`A complete CRL lists all unexpired certificates, within its scope, that have been revoked for one of the revocation reasons covered by the CRL scope. A full and complete CRL lists all unexpired certificates issued by a CA that have been revoked for any reason.`

A hitelesítés-szolgáltató megteheti, hogy a CRL-en már nem tünteti fel a lejárt tanúsítványok visszavonási állapotát, így ha egy tanúsítvány lejárt, előfordulhat, hogy nem lehet eldönteni, hogy visszavonták-e a lejárta előtt. OCSP esetén annyival jobb a helyzet, hogy az OCSP válasz `archiveCutOff` mezejében fel lehet tüntetni, hogy mennyire régi állapotra vonatkozik a válasz. (Sajnos, erre sem számíthatunk biztosan, mert a szolgáltató nem köteles támogatni az `archiveCutOff` mezőt. Ha az OCSP válaszban nem szerepel `archiveCutOff`, akkor a szolgáltató valószínűleg nem ad releváns választ a lejárt tanúsítványok tekintetében. Ha a szolgáltató a lejárt tanúsítványokra is releváns választ ad, akkor az `archiveCutOff` segítségével jelezheti, hogy mennyire régen lejárt tanúsítványokra válaszol, de üresen is hagyhatja az OCSP válaszban szereplő `archiveCutOff` értéket.)

Ha autentikációs vagy titkosító tanúsítványokról, vagy csak nagyon rövid ideig használt aláírásokhoz tartozó tanúsítványokról van szó, akkor valóban elegendő, ha csak az érvényes tanúsítvány visszavonási állapotát lehet lekérdezni, de a kézzel írott aláírásnak megfelelő elektronikus aláírások esetén általában ennél sokkal többre van szükség. Ha a tanúsítvány

[2]Megjegyezzük, hogy más szempontból a fenti XAdES-A valóban előnyösebb lehet. Például ha három év múlva már nem érhető el Alajos aláírására vonatkozó visszavonási információ, Bendegúz aláírása még mindig ellenőrizhető a csatolt visszavonási információk alapján.

lejárt, akkor – ha csak nem ismerjük a visszavonási állapotot közzétevő szolgáltató működését – a lejárt tanúsítványok korábbi visszavonási állapotának megállapítása problémás.

> ***8.4. Példa:*** *Alajos tanúsítványát 2008. január 1-én bocsátották ki. Manfréd 2008. január 2-án ellopta Alajos tárcáját, benne a tanúsítványához tartozó intelligens kártyáját. Alajos értesítette a hitelesítés-szolgáltatót, aki még 2008. január 2-án közzétette, hogy Alajos tanúsítványa érvénytelen. Manfréd 2008. január 4-én megtalálta Alajos tárcájában a PIN kódot, és XAdES-T aláírást készített.*
>
> *Bendegúz 2008. február 3-án kapta meg a fenti aláírást. Ellenőrizte a szolgáltató által közzétett visszavonási listát, ez alapján érvénytelennek tekintette az aláírást.*
>
> *A tanúsítvány 2010. január 1-én járt le, így a 2010. január 5-én kibocsátott visszavonási listán már nem szerepelt. Cili 2010. január 15-én kapta meg az aláírást. Nem tudott megbizonyosodni róla, hogy a szolgáltató által közzétett visszavonási lista releváns az adott tanúsítványra, így elutasította az aláírást.*
>
> *Cili egy másik aláírást is kapott 2010. január 15-én. Ezen aláírást Dezső készítette egy 2008. január 7-én kibocsátott (és azóta már lejárt) tanúsítvány szerint. Dezső magánkulcsa soha nem kompromittálódott, és az aláírást valóban Dezső készítette, de Cili 2010. január 15-én erről már nem tudott megbizonyosodni, így Dezső aláírását is elutasította – Alajos aláíráshoz hasonlóan.*

A hitelesítés-szolgáltató jellemzően csak az aktuális visszavonási információkat teszi közzé, de az elektronikus aláírásról szóló törvény szerint a tanúsítvány lejártától számított 10 évig köteles megőrizni, hogy mely tanúsítvány mikor vált érvénytelenné.

> *Eat. 9 § „ (7) A hitelesítés-szolgáltató a tanúsítványokkal kapcsolatos elektronikus információkat – beleértve az azok előállításával összefüggőket is – és az ahhoz kapcsolódó személyes adatokat legalább a tanúsítvány érvényességének lejártától számított tíz évig, illetőleg az elektronikus aláírással, illetve az azzal aláírt elektronikusan aláírt elektronikus dokumentummal kapcsolatban felmerült jogvita jogerős lezárásáig megőrzi, valamint ugyanezen határidőig olyan eszközt biztosít, mellyel a kibocsátott tanúsítvány tartalma megállapítható. E megőrzési kötelezettségnek a hitelesítés-szolgáltató minősített archiválási szolgáltató igénybevételével is eleget tehet. ”*

Megjegyzés:

1. A hitelesítés-szolgáltató nem köteles magukat a visszavonási listákat megőrizni, de később meg kell tudja mondani, hogy melyik tanúsítvány mettől meddig volt érvényes.

2. A fenti követelményt az Eat. tartalmazza, így csak az aláíró tanúsítványokra érvényes. A titkosító és autentikációs tanúsítványok esetén a tanúsítványra vonatkozó hitelesítési rendből derül ki, hogy vonatkozik-e rájuk hasonló szabály.

Ha a tanúsítványról a CRL-ek és az OCSP szolgáltatás segítségével nem érhető el a friss visszavonási információ, a hitelesítés-szolgáltató megkérdezésével továbbra is meg lehet állapítani, hogy a tanúsítványt visszavonták-e, és mikor vonták vissza – egészen addig, amíg a fenti adatmegőrzési idő le nem telik. Lehet, hogy ezt automatizmus segítségével nem lehet megállapítani, hanem pl. elektronikus levélben kell felvenni a kapcsolatot a szolgáltatóval. Önmagában az, hogy egy tanúsítvány mikor volt érvényes, és mikor érvénytelen, nyilvános információ, így ezt a hitelesítés-szolgáltató kiadhatja, és ezt az Eat. 9 § (7) szerinti adatmegőrzési idő lejártát követően is megőrizheti.

Az adatmegőrzési időn belül mindenképpen összegyűjthető a szükséges visszavonási információ. Még akkor is, ha a hitelesítés-szolgáltató időközben megszűnik, mert ekkor az Eat. szerint egy másik hitelesítés-szolgáltató (vagy a Nemzeti Média- és Hírközlési Hatóság) kell, hogy átvegye a szükséges információk őrzését (4.3.9. fejezet). Ugyanakkor az Eat. 9 § (7) szerinti adatmegőrzési idő lejártát követően előfordulhat, hogy a tanúsítványra vonatkozó releváns visszavonási információt már egyáltalán nem lehet beszerezni.

Ha egy aláírást a tanúsítvány lejártán[3] *túl is hitelesen meg szeretnénk őrizni, célszerű még addig összegyűjteni a visszavonási információkat, amíg a tanúsítvány érvényes.* Ha ez nem történt meg, az adatmegőrzési időn belül a hitelesítés-szolgáltatótól még beszerezhető a szükséges visszavonási információ, bár ez már jelentősen nehézkesebb. *Ha egy aláírást 10 évnél tovább szeretnénk megőrizni, célszerű összegyűjteni és csatolni hozzá a szükséges visszavonási információkat.*

Megjegyzés: Ha az aláírás ellenőrzésekor kivárási időt alkalmazunk, lehet, hogy rögtön az aláírás létrehozásának pillanatában még nem gyűjthető be a szükséges visszavonási információ.

Ne feledjük, hogy a visszavonási információkon szintén aláírás van, így az rajtuk lévő aláírás hosszú távú hiteles megőrzéséről szintén célszerű gondoskodni – például a rájuk vonatkozó visszavonási információk összegyűjtése, a rendszeres időbélyegzés stb. segítségével. Egy XAdES-A vagy CAdES-A aláírás például jó megoldást jelenthet ezen információk tárolására és hosszú távú hiteles megőrzésére.

[3] Vegyük figyelembe, hogy a tanúsítvány akár másnap is lejárhat.

8.1.4. Ha nem lehet kideríteni, hogy ki volt az aláíró

A tanúsítványban az alany megnevezése (DN-je) szerepel, önmagában a tanúsítvány alapján ennyi állapítható meg az alanyról. Ha a tanúsítvány nem álneves, akkor tartalmazza az alany nevét, valamely személyazonosításra alkalmas okmányában szereplő írásmóddal. (Emellett egyéb adatokat is tartalmazhat, de ezek köre attól függ, hogy pontosan milyen típusú tanúsítványról van szó.) Ha egy nem álneves tanúsítvány csak annyit tartalmaz, hogy az alany neve „Kovács János", akkor esetleg nagyon nehezen kapcsolható össze egy természetes személlyel. Ha a tanúsítvány álneves, akkor nem tartalmazza az alany nevét, és esetleg egyáltalán nem állapítható meg belőle az alany kiléte.

Az Eat. 9 § (7) egyúttal a regisztrációs információkra is vonatkozik, azaz a szolgáltatónak ennyi ideig kell megőriznie, hogy egy tanúsítványt kinek bocsátott ki, azaz mik az illető személyazonosító adatai. Ezen információk nem nyilvánosak, és a hitelesítés-szolgáltató kizárólag az alany beleegyezésével vagy jogszabályban meghatározott esetekben adhatja ki őket. Az Eat. szerint a következő esetekben adhatja ki a hitelesítés-szolgáltató az alany személyes adatait:

> *Eat. 11 § „ (2) A hitelesítés-szolgáltató az elektronikus aláírás felhasználásával elkövetett bűncselekmények felderítése vagy megelőzése céljából, illetőleg nemzetbiztonsági érdekből – az érintett személyazonosságát igazoló, valamint a 12. § (2) bekezdése alapján egyeztetett adatok tekintetében – az adatigénylésre külön törvényben meghatározott feltételek teljesülése esetén adatokat továbbít a nyomozó hatóságnak és a nemzetbiztonsági szolgálatoknak. Az adatátadás tényét rögzíteni kell, az adatátadásról a hitelesítés-szolgáltató az aláírót nem tájékoztathatja.*
>
> *(3) A hitelesítés-szolgáltató a tanúsítvány érvényességét érintő polgári peres, illetve nemperes eljárás során – az érintettség igazolása esetén – az aláíró személyazonosságát igazoló, valamint a 12. § (2) bekezdése alapján egyeztetett adatokat átadhatja az ellenérdekű peres félnek vagy képviselőjének, illetőleg azt közölheti a megkereső bírósággal. "*

Az adatmegőrzési idő lejártát követően a hitelesítés-szolgáltatónak – feltéve, hogy az alany nem adta hozzájárulását, hogy a szolgáltató tovább őrizze az adatait – nincsen jogalapja az alany személyes adatainak kezelésére, így a személyes adatokat le kell törölnie. Ezt követően esetleg nagyon nehéz bizonyítani, hogy ki készítette az aláírást.

> ***8.5. Példa:*** *Egy dokumentumon Kovács János aláírása szerepel. A dokumentum szerint Kovács János, aki Kecskeméten született 1974. június 3-án, és anyja neve Nagy Rozália, 2003-ban eladta a lakását Nagy Józsefnek. A nevezett Kovács János létezik, de tagadja, hogy ő valaha is aláírta volna a dokumentumot, azt*

állítja, hogy a dokumentumot egy másik, Kovács János nevű személy is aláírhatta. 2010-ben a hitelesítés-szolgáltató közreműködésével igazolható, hogy valóban az ő magánkulcsával készült-e az aláírás. 2023-ban ezt már nem lehet megállapítani.

8.6. Példa: *Egy dokumentumon Tallérossy Zebulonné Dr. Felsődióssy-Nagy Annamária Olga aláírása szerepel. Ő nehezen hivatkozhat arra, hogy az aláírást egy másik, azonos nevű személy készíthette. (Különösen, ha tanúsítványában szerepel, hogy ő ügyvéd, az ügyvédi irodájának a neve és címe, és az illetékes ügyvédi igazolványán szereplő lajstromszám is.)*

Az aláírást archiváló fél elvileg megőrizhetne egy hiteles igazolást az aláíró személyes adatairól, de ezen adatokhoz – tipikus esetben – nem férhet hozzá. Ugyanakkor valós folyamatokban ritkán fordul elő, hogy egy tanúsítvány lejáratát követően 10 év elteltével próbálna érvényesíteni valaki egy követelést. Ugyanez a probléma fennáll a kézzel írott aláírásokkal kapcsolatban is: a kézzel írott aláírásból is csak az aláíró neve derül ki.

Ha a tanúsítvány érvényességével kapcsolatban jogvita indul, célszerű értesíteni erről a hitelesítés-szolgáltatót, mert a szolgáltató az Eat. 9 § (7) szerint a jogvita lezártáig köteles megőrizni a tanúsítványra vonatkozó információkat.

8.1.5. Ha a kriptográfiai algoritmusok elavulnak

Amikor egy dokumentum, egy aláírás és egy nyilvános kulcs összetartozását vizsgáljuk, arra építünk, hogy az adott kriptográfiai algoritmus „biztonságos”. Így például feltételezzük, hogy az aláírást a magánkulcs nélkül nem lehetett kiszámítani, és feltételezzük, hogy a magánkulcsot a nyilvános kulcsból nem lehetett kiszámítani. Szintén feltételezzük, hogy a használt hash függvényre teljesülnek az ütközés-ellenállóság és őskép-ellenállóság követelmények (2.4. fejezet).

Előfordulhat, hogy ez idővel megváltozik, és a fentiek valamelyike már nem teljesül. Két okból következhet be, hogy egy kriptográfiai algoritmusban megrendül a bizalom:

- *Technológiai fejlődés.* A számítástechnika fejlődésével a számítógépek egyre gyorsabbak és gyorsabbak, illetve fajlagosan egyre olcsóbbak lesznek, így a támadó egyre több és egyre hatékonyabb gépet tud összekapcsolni a támadáshoz. Ennek következtében egyre nagyobb kulcsteret tud kimerítő kereséssel átvizsgálni. E tényező becsülhető, például a „Moore-törvény” szerint a számítástechnikai eszközök sebessége kb. másfél évente duplázódik meg. A hirtelen, drasztikus sebességnövekedés nagyon valószínűtlen, és ha a technológia fejlődése oda vezet, hogy már nem teljesen irreális, hogy egy adott algoritmust belátható időn belül reális valószínűséggel lehet támadni, akkor célszerű vagy áttérni egy másik algoritmusra, vagy megnövelni a használt kulcsméretet. A

kulcsméret növelésével hatékonyan meg lehet nehezíteni a támadó feladatát, mert a kulcsméret duplázásával a kimerítő kereséshez szükséges idő általában négyzetesen nő meg.

- *Algoritmikus fejlődés.* Előfordulhat, hogy a kriptográfusok, matematikusok az eddig ismert módszereknél jelentősen hatékonyabb módszert találnak egy algoritmus megtámadására. Így előfordulhat, hogy a rendelkezésre álló számítástechnikai kapacitás mellett már nem irreális egy adott algoritmust sikeresen megtámadni. E támadás lehet, hogy csak egy adott algoritmust érint – ilyenkor mondjuk, hogy hibát találtak az algoritmusban –, de egy alapvető számításelméleti áttörés több algoritmus biztonságát is megingathatja. Általában az algoritmikus fejlődés sem teszi hirtelen használhatatlanná az adott kriptográfiai algoritmust, célszerű figyelni támadások fejlődéséről szóló híreket, és kivezetni az adott algoritmust, ha az inogni látszik.

  8.7. Példa: *Biryukov és Khovratovich 2009-es támadása 2^{119} lépésből „töri" a 256 bites kulccsal használt AES-t (a 2^{256} lépés helyett). A szerzők súlyos gyenge pontot találtak az algoritmuson, de a 2^{119} lépés még mindig túl sok, e támadás még nem volt kivitelezhető. [15]*

  8.8. Példa: *Wang, Yin és Yu 2005-ben talált olyan támadást az SHA-1 ellen, amely 2^{69} lépésből talál két különböző ősképet, amelyekhez azonos lenyomat tartozik (a 2^{80} helyett[4]). Így e támadás kb. 2000-szeres gyorsítást jelent a kimerítő kereséshez képest, és már 2005-ben sem volt teljesen irreális. [163], [162]*

  McDonald, Hawkes és Pieprzyk 2009-ben még gyorsabb, 2^{52} lépésben kivitelezhető támadást mutatott az SHA-1 ellen, amely akkor már a gyakorlatban is kivitelezhető volt. [110]

Minden aláíráson, amelynek érvényességét a kriptográfiai algoritmusok elavulása érintheti, célszerű még addig elhelyezni egy új időbélyeget, amíg az adott aláírás érvényessége még igazolható, azaz amíg a kérdéses kriptográfiai algoritmusok még biztonságosnak tekinthetőek. Ez az alapelv nemcsak a végfelhasználók által létrehozott aláírásokra, hanem a szolgáltatók által létrehozott, az időbélyegeken, CRL-eken, OCSP válaszokon lévő aláírásokra is igaz. Ehhez célszerű folyamatosan követni a kriptográfia fejlődését, és ennek függvényében helyezni el az időbélyegeket.

Ha egy már időbélyegzett objektumon helyezünk el új időbélyeget, a következőképpen célszerű eljárni:

[4]Az SHA-1 160 bites (20 byte-os) lenyomattal dolgozik, így összesen 2^{160} különböző lenyomat létezik. Elegendő ennek a gyökét, azaz 2^{80} db. különböző ősképet generálni, köztük már 50 százalék eséllyel találunk két olyat, amelyekhez azonos lenyomat tartozik.

- Az új időbélyeg egyúttal a régi időbélyeget is védje, így az új időbélyeg alapján igazolható a régi időbélyeg érvényessége, a régi időbélyeg alapján pedig így továbbra is igazolható az eredeti megbízható időpont.
- Az új időbélyeg lehetőleg „erősebb" legyen, mint a régi időbélyeg. Szerencsés, ha különböző (esetleg erősebb) kulccsal készül, és célszerű, ha az új időbélyeg olyan kriptográfiai algoritmusokra épül, amelyeket az eredeti időbélyeghez kapcsolódó algoritmusok elavulása nem érint. Szerencsés, ha ugyanez nemcsak az időbélyegre, hanem az időbélyegzés-szolgáltató tanúsítványára és a hozzá kapcsolódó tanúsítványláncra és visszavonási információkra stb. is igaz.

Műszaki szempontból sokszor nem köthető egyértelmű időponthoz, hogy egy kriptográfiai algoritmus mikor avul el. Először általában hibát találnak egy algoritmusban, azaz olyan támadást találnak, amely a véletlen próbálkozásnál jelentősen gyorsabb. E támadás még lehet, hogy közel nem reális, és lehet, hogy csak elméleti jelentősége van. Később egyre gyorsabb és gyorsabb támadások jelennek meg, és közben folyamatosan fejlődik a technológia, így egy támadó egyre több és több erőforrást tud mozgósítani. A kriptográfiai algoritmust általában még jóval azelőtt ki szokták vonni a forgalomból, hogy reálisan végre lehetne hajtani ellene egy támadást.

Nemzetközi szervezetek ajánlásokat szoktak kibocsátani arra vonatkozóan, hogy mely algoritmusokat milyen célra célszerű használni. Az USA-ban az NIST és az NSA bocsát ki ilyen dokumentumokat, az Európai Unióban az ETSI TS 102 176 (ALGO PAPER) számít mértékadó dokumentumnak. [129], [128], [57] A `keylength.com` weboldal különböző, kriptográfiai algoritmuskészletek és kulcsméretek élettartamára vonatkozó ajánlásokat gyűjt össze.

Az Eat. (18 §) szerint Magyarországon a Nemzeti Hírközlési Hatóság határozata szabja meg azon algoritmusok körét, amelyekkel elektronikus aláírást, tanúsítványt vagy időbélyegzőt lehet készíteni.

8.2. A digitális archiválás jogszabályi követelményei

Az elektronikus dokumentumok archiválásának szabályait a digitális archiválásról szóló 114/2007. GKM rendelet határozza meg. [69] E rendelet nem kizárólag az elektronikusan aláírt dokumentumokra, hanem általában az elektronikus információ hiteles megőrzésére vonatkozik.

> *114/2005. GKM r. „ 2. § (1) A megőrzésre kötelezett a megőrzési kötelezettség lejártáig folyamatosan köteles biztosítani, hogy az elektronikus dokumentumok megőrzése olyan módon történjen, amely kizárja az utólagos módosítás lehetőségét,*

> *valamint védi az elektronikus dokumentumokat a törlés, a megsemmisítés, a véletlen megsemmisülés és sérülés, illetve a jogosulatlan hozzáférés ellen. ”*

A rendelet nemcsak követelményeket határoz meg, hanem azt is leírja, hogy e követelményeket hogyan lehet teljesíteni. A rendelet szerint ha valaki elektronikus információt hitelesen szeretne megőrizni a következő módszerek valamelyikét kell alkalmaznia:

1. Az információn legalább fokozott biztonságú elektronikus aláírást és minősített időbélyeget helyez el. Ekkor kriptográfiai módszerekkel biztosítja az információ hitelességét, így az információ hitelességét annak szerkezete biztosítja, attól függetlenül, hogy az információt hol és milyen eszközökkel őrzik.

2. Olyan zárt rendszert alkalmaz, amelyre egy akkreditált tanúsító szervezet igazolja, hogy megfelel a rendeletben szereplő követelményeknek. Ekkor a zárt rendszer biztosítja a hitelességet, de a rendszer csak addig igazol hitelességet, amíg az valóban zárt.

3. Elektronikus adatcsere (EDI, electronic data interchange) rendszert alkalmaz; ezzel a jogszabály egy speciális zárt rendszert nevesít.

A korábbi, már nem hatályos 7/2005. IHM rendeletben egy negyedik lehetőség is szerepelt, miszerint a megőrzésre kötelezett bontásbizos csomagolásban helyezi el az információt, amelyet közjegyző lepecsétel. Ekkor a fizikai biztonság biztosította volna az információ hitelességét. E lehetőség azóta megszűnt, feltehetően azért, mert senki nem élt vele. [81]

A továbbiakban arról a lehetőségről írunk, ha valaki legalább fokozott biztonságú elektronikus aláírással és minősített időbélyeggel kívánja biztosítani az archivált dokumentumok hitelességét.

Az elektronikus aláírás természetesen csak az észrevétlen utólagos módosítást zárja ki, nem biztosítja azt, hogy a dokumentum nem sérül vagy semmisül meg. Aki ezt a megoldást választja, annak továbbra is meg kell őriznie a dokumentumot, de jelentősen egyszerűbb és olcsóbb hozzáférés-védelmi megoldásokat alkalmazhat, mintha a megőrzésre használt környezettel próbálná igazolni, hogy a dokumentumot nem módosították.

Aki legalább fokozott biztonságú elektronikus aláírással és minősített időbélyeggel ellátva szeretné biztosítani a rendelet 2 § (1) pontjában leírt követelményeket, annak a következőképpen kell eljárnia:

1. Ha még nincs aláírva a dokumentum, neki kell aláírást elhelyeznie rajta.

2. Ellenőriznie kell az aláírást, és ha még nincs időbélyegezve a dokumentum, neki kell időbélyeget elhelyeznie rajta.

> **Megjegyzés:** Ha valaki egy XAdES-T aláírással ellátott dokumentumot akar archiválni, akkor e fenti lépéseket nem kell elvégeznie.

3. Ha a megőrzési kötelezettség időtartama hosszabb, mint 11 év, akkor a fentieken túl:

   a. Be kell szereznie minden olyan információt, amely az aláírás érvényességét igazolja.

   b. Minősített időbélyeget kell elhelyeznie a fenti adatokon.

   c. Újabb minősített időbélyeget kell elhelyeznie, ha az előző pontban szereplő időbélyeg az „Eat. szabályai szerint” (azaz a Nemzeti Média- és Hírközlési Hatóság határozata szerint) már nem biztonságos.

A jogszabály alapvetően az aláírás műszaki érvényességének megőrzését írja elő. Ugyanakkor a jogszabályi követelmények néhol eltérnek a műszaki szempontoktól.

- Jogszabályilag elválik a „rövid távú” (11 éven belüli) és a „hosszú távú” (11 évet meghaladó) megőrzés, holott műszakilag nincs ilyen éles határ: a hosszú távon felmerülő problémák – elvileg – akár egy éven belül is felmerülhetnek.

  Ugyanakkor a teljes körű archiválás a visszavonási információk tárolásával, a technológiafigyeléssel és a rendszeres időbélyegzéssel együtt igen költséges lehet. Elfogadható, hogy rövid távon nagyon kicsi a valószínűsége pl. egy algoritmus-elavulásnak vagy egy időbélyegző kulcs kompromittálódásának, és valószínűleg felesleges lenne azt is terhelni e megoldásokkal, akik csak 5 vagy 8 évre őriznek meg egy aláírást.

- A jogszabály külön nem emeli ki az időbélyegzés-szolgáltatói magánkulcsok kompromittálódását, vagy az időbélyegzés-szolgáltatói tanúsítványok lejártát, holott ezek is indokolják, ha ismételt időbélyeget kell elhelyezni egy aláíráson.

- Egy időbélyegzés-szolgáltató tanúsítványa nem feltétlenül érvényes 11 évig. Így előfordulhat, hogy az időbélyegzés-szolgáltató tanúsítványa lejár, és így műszakilag már nem lehet igazolni egy aláírás érvényességét.

- A jogszabályban szereplő 11 éves határvonal vélhetően onnan származik, hogy az Eat. szerinti alapértelmezett adatmegőrzési idő a tanúsítvány lejártától számított 10 év, és a tanúsítvány maximális élettartamaként vett 1 évet adták hozzá.

  Ez olyan értelemben logikus választóvonal, hogy a hitelesítés-szolgáltatótól ennyi ideig még beszerezhetőek az aláírás időpontjára vonatkozó visszavonási információk (8.1.3. fejezet). Ugyanakkor sok szolgáltató nem 1, hanem 2 évre bocsát ki minősített tanúsítványt, így a két határvonal nem esik pontosan egybe.

- Az Eat. szerinti adatmegőrzési idő leteltével már nehéz igazolni, hogy ki volt az aláíró (8.1.4. fejezet). Ezzel a problémakörrel egyáltalán nem foglalkozik a rendelet. (A rendeletben nem is lehetne ezt rendezni, hiszen ahhoz az Eat. szerinti adatkezelést kellene felülbírálni, és ezt egy rendeletben nem lehet megtenni.) Megjegyezzük, ritka, hogy 10 év múlva indul vita arról, hogy annak idején ki volt az aláíró, és 10 év elteltével már igen valószínűtlen, hogy valaki jogilag érvényesíthető követeléssel állhasson elő.

A rendelet szerint a megőrzésre kötelezett maga is elvégezheti a fenti műveleteket, de minősített archiválás-szolgáltatót is megbízhat vele. A kettő között az jelenti a fő különbséget, hogy ha minősített archiválás-szolgáltató archivál egy dokumentumot, akkor az Eat. szerint vélelmezni kell, hogy „jól" végzi az archiválást, és felel azért, hogy az archiválást jó végezze. Ezzel szemben, ha valaki saját maga végzi az archiválást, akkor neki kell bizonyítani, hogy megfelelően jár el (pl. megfelelően helyezi el az időbélyeget), és ő maga felel minden ebből eredő kárért.

8.3. Hogyan biztosíthatjuk az aláírás hosszú távú érvényességét?

Összefoglalva a műszaki (8.1. fejezet) és jogszabályi (8.2. fejezet) szempontokat, a következőket célszerű tenni, ha egy aláírást hosszú távon hitelesen szeretnénk megőrizni:

1. Az aláíráson célszerű időbélyeget elhelyezni a lehető leghamarabb, lehetőleg rögtön a készítését követően. Ezt követően van olyan megbízható időpontunk, amelyre nézve vizsgálhatjuk az aláírás érvényességét. Később is igazolhatjuk, hogy az aláírás létezett az időbélyegben jelzett időpontban, így – ha összegyűjtjük a szükséges bizonyítékokat – az aláírás ellenőrzését később megismételve azonos eredményre juthatunk. A később is igazolható megbízható időpont nélkül nem beszélhetünk egzakt ellenőrzésről.

2. Célszerű összegyűjteni az aláírás érvényességét igazoló információkat (6.5. fejezet) (érvényességi láncot), beleértve a tanúsítványláncot, az aláíró tanúsítványára és a tanúsítványlánc elemeire vonatkozó visszavonási információkat, valamint az időbélyegen, a szolgáltatói tanúsítványokon és a visszavonási információkon lévő aláírás ellenőrzéséhez szükséges információkat (tanúsítványláncokat és visszavonási információkat stb).

3. Az érvényességi láncon is célszerű időbélyeget elhelyezni. Így már garantáltan van egy időbélyeg az érvényességi láncban szereplő minden egyes aláíráson, és így van egy olyan időpont, amelyre nézve az egyes tanúsítványokon, CRL-eken, OCSP válaszokon lévő aláírások érvényességét vizsgálhatjuk.

4. Az érvényességi láncon – beleértve a mindenkori „legkülső" időbélyeget – célszerű új, „erősebb", lehetőleg más forrásból származó, más kulccsal készült, erős algoritmuskészletre alapuló időbélyeget elhelyezni, mielőtt a „legkülső" időbélyeg ellenőrizhetőségével probléma merül fel (azaz mielőtt a tanúsítványa lejár, a kulcsa kompromittálódik, vagy valamely algoritmusa elavul – akár az időbélyegnek, akár az érvényességi láncnak). Ez bizonyos mértékű jövőbelátást igényel. Ezért célszerű nyomon követni a technológia fejlődését, és célszerű nem egyetlen szolgáltatói kulcsra

építeni az archiválást. Például helyes megoldás, ha „baj” esetén a mindenkori legkülső időbélyeg elhagyásával, a mindenkori második legkülső időbélyeg alapján is felépíthető az érvényességi lánc. Lényeges, hogy új időbélyeg elhelyezése előtt célszerű csatolni az érvényességi lánchoz az előző külső időbélyeg érvényességét igazoló visszavonási információt az aktuális „*most*” időpontra nézve.

Az archiválás során n darab, egymásra épülő archív időbélyegből álló láncot építünk. A mindenkori „legkülső”, n. időbélyeget az ellenőrzés időpontjában, a *most* időpontban is ellenőriznünk kell tudni, a *most* időpontra nézve. Az $n-1$. időbélyeget az n. időbélyeg által meghatározott időpontra nézve ellenőrizhetjük, az $n-2$. időbélyeget az $n-1$. időbélyegben szereplő időpontra stb.

8.9. Példa: *A 2010. évben ellenőrzünk XAdES-A aláírást, amelyen három archív időbélyeg van. Az első két archív időbélyeg a 2010-ben már elavult, MD5 algoritmussal készült, így önmagukban már nem ellenőrizhetőek. A 3., legkülső időbélyeg viszont még ellenőrizhető. Ezen időbélyeget 2006-ban helyezték el az érvényességi láncon, és az SHA-1 lenyomatképző algoritmussal készült. E 3. időbélyeg alapján igazolható, hogy a 2. időbélyeg (valamint minden információ, amelyet a 2. időbélyeg véd) nem 2006. után jött létre, és az MD5 akkor még „elfogadható” algoritmus volt.*

Így a 2. időbélyeget is elfogadhatjuk, ha sikeresen ellenőriztük a 3. időbélyeg által meghatározott, 2006. évi időpontra nézve. A 2. időbélyeg igazolja, hogy az 1. időbélyeg 2004-ben jött létre. Az 1. időbélyeg olyan gyökérre vezethető vissza, amelynek magánkulcsa 2005-ben kompromittálódott. Így sem a ma (2010-ben), sem a 3. időbélyeg által meghatározott 2006-os időpontra nézve nem fogadhatjuk el az 1. időbélyeget. Csakhogy a 2. időbélyeg szerint az 1. időbélyeg 2004-ben jött létre, és az adott gyökér akkor még nem kompromittálódott, hanem érvényes, elfogadott gyökér volt.

Így az 1. időbélyeget is elfogadhatjuk, ha sikeresen ellenőrizzük a 2. időbélyeg által meghatározott, 2004. évi időpontra nézve. Az 1. archív időbélyeg igazolja, hogy az aláírás, a rajta lévő (XAdES-T) időbélyeg, és az aláíráshoz csatolt tanúsítványláncok és visszavonási információk 2002-ben jöttek létre. Így ezen információkat annak ellenére elfogadhatjuk, hogy a rajtuk lévő aláírások algoritmusai elavultak, és a gyökerek, amelyekre épültek, már régen nem gyökerek.

A továbbiakban néhány olyan megoldást mutatunk be, amelyek segítségével a fentieknek megfelelő módon archiválhatunk aláírásokat.

8.3.1. Archív aláírás (XAdES-A)

Ha az archiváláshoz szükséges információkat – tanúsítványokat, tanúsítványláncokat, visszavonási információkat, időbélyegeket stb. – az aláíráshoz csatoljuk, archív aláírásról beszélünk. Ilyen technológiát jelent például a XAdES-A (6.4.1.2.6. fejezet) vagy a CAdES-A formátum. A továbbiakban a XAdES-A megoldásról szólunk, de megállapításaink egyaránt érvényesek a CAdES-A megoldásra is.

XAdES aláírás készítésekor először mindig „alap aláírás", azaz XAdES-BES vagy -EPES jön létre. Ezen célszerű a lehető leghamarabb időbélyeget elhelyezni; optimális esetben az aláírás-létrehozó alkalmazás az aláírás készítése keretében a -BES vagy -EPES elkészítése után azonnal beszerez egy időbélyeget az aláírásra, azaz a -BES vagy -EPES aláírás helyett XAdES-T aláírást készít el.

Az időbélyeggel ellátott XAdES-T aláíráshoz célszerű csatolni a visszavonási információkat, és célszerű azokat is időbélyegezni. Ha archív, azaz XAdES-A aláírást hozunk létre, akkor az aláírás érvényességét igazoló összes információt – az aláírt dokumentumot, az aláírást, tanúsítványokat, tanúsítványláncokat, visszavonási információkat, időbélyegeket stb. – egy mindent átfogó időbélyeggel, ún. archív időbélyeggel védünk meg. Az érvényes archív időbélyeg igazolja, hogy az összes általa védett információ létezett az archív időbélyegben megjelölt időpontban, így amikor az ezen információkon lévő aláírásokat ellenőrizzük, az archív időbélyegben szereplő időpontot mindenképpen alapul vehetjük.

> **Megjegyzés:** A XAdES-A aláírás készíthető a XAdES-X-L aláírásból, de készíthető közvetlenül a XAdES-T-ből is, ekkor eggyel kevesebb időbélyegre van szükség.

Az archív időbélyeg önmagában nem különbözik a többi időbélyegtől, nem az időbélyeg típusát, hanem az időbélyeg felhasználásának módját jelöli. Az archív időbélyeget mindig közvetlenül a védendő objektumokon (és nem pl. azok lenyomatain vagy az azokról készült aláírásokon) helyezzük el, azaz az archív időbélyeg készítésekor az összes védendő objektumról együttesen kell lenyomatot képezni. Például a XAdES-T aláírásban szereplő időbélyeget csak magán az *aláíráson* helyezzük el, azaz készítésekor csak az aláírásból képzünk lenyomatot. A XAdES-T időbélyeg így csak az aláírás létezésére vonatkozóan igazol egy megbízható időpontot; az aláírt dokumentumra csak akkor vonatkozik, ha az aláíráshoz használt kriptográfiai algoritmusok még biztonságosak.

> ***8.10. Példa:*** *Az X dokumentumot 2003-ban írták alá, az aláírás készítéséhez az MD5 lenyomatképző algoritmust és a 2048 bites RSA algoritmust használták. Az aláíráson még 2003-ban időbélyeget helyeztek el, azaz XAdES-T aláírássá bővítették. Az időbélyeg készítéséhez az SHA-256 lenyomatképző algoritmust és a*

2048 bites RSA algoritmust használták; az időbélyeg készítésekor csak az aláírásból képeztek lenyomatot.

Az időbélyeg 2010-ben is érvényes, így 2010-ben is igazolható, hogy az aláírás 2003-ban már létezett. Problémát jelenthet viszont, hogy az MD5 2010-ben már nem biztonságos, így nem feltétlenül igazolható, hogy az aláírás eredetileg mely dokumentumra vonatkozott.

Előfordulhat, hogy Alajos is és Bendegúz is mutat egy-egy dokumentumot, a két dokumentum különbözik egymástól, de az aláírás mindkét dokumentumra vonatkozhat. Az MD5 2003-ban még biztonságos volt, így feltehetően csak az egyik dokumentum létezett akkor, a másik utólag készült hamisítvány. Mivel az MD5 már nem biztonságos, a XAdES-T időbélyeg alapján nem lehet megállapítani, hogy melyik dokumentum létezett már 2003-ban.

Ha annak idején kiterjesztették volna az aláírást XAdES-A-vá, e probléma nem merülhetne fel. Akkor a XAdES-A aláírás archív időbélyegéhez a dokumentumról, az aláírásról, a XAdES-T időbélyegről stb. együttesen kellett volna lenyomatot képezni az SHA-256 algoritmussal, és ez az archív időbélyeg igazolná, hogy melyik dokumentum létezett 2003-ban.

Mielőtt az archív időbélyeg érvényességével probléma merülne fel, új archív időbélyeget célszerű elhelyezni a teljes eddigi XAdES-A láncon, azaz az aláírt dokumentumon, az aláíráson, a XAdES-T időbélyegen, a csatolt visszavonási információkon, az esetleges további tanúsítványokon, időbélyegeken, tanúsítványláncokon, a rájuk vonatkozó visszavonási információkon, illetve az összes eddigi archív időbélyegen (és a hozzájuk kapcsolódó információkon).

Új archív időbélyeg elhelyezése előtt célszerű csatolni az előző archív időbélyeg érvényességét igazoló visszavonási információkat. Sajnos, a XAdES specifikáció erre nem biztosít szabványos helyet, azaz nem írja le, hogy az aláírás-ellenőrző alkalmazásnak hol kellene keresnie ezen információkat. Az egyik legjobb nemzetközi gyakorlat e probléma kezelésére, ha ezen információkat az előző archív időbélyeg ASN.1 struktúrájában lévő CMS/CAdES aláíráshoz csatoljuk.

Megjegyzés: Az előző archív időbélyeget később az új archív időbélyegben szereplő időpontra nézve ellenőrizhetjük. Ebből következően az előző archív időbélyegre vonatkozó visszavonási információkat az új archív időbélyegben szereplő időpontra nézve célszerű csatolni. Ha azt szeretnénk, hogy az új archív időbélyeg védje ezen információkat, előbb kell az információkat csatolnunk, és csak utána helyezhetjük el rajta az időbélyeget. Ekkor viszont az új archív időbélyegben szereplő időpont későbbi, mint a visszavonási információk `thisUpdate` időpontja,

és az ellenőrző nem alkalmazhat kivárási időt. Tekintve, hogy archív időbélyegről van szó, ez nem annyira súlyos probléma, mert a legkülső archív időbélyeget jellemzően nem lehet kivárási idővel ellenőrizni[5].

Az archív aláírás technológiája a következő tulajdonságokkal bír:

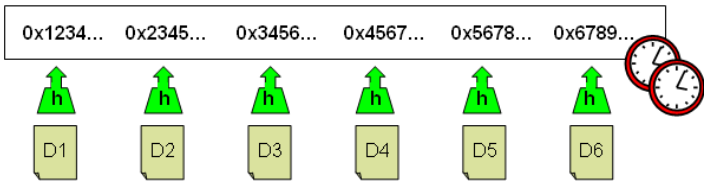
- Nem igényel nagy befektetést, hiszen ingyenesen elérhető eszközök léteznek a XAdES-A aláírásokkal történő archiválásra. Ilyen ingyenes eszköz például az e-Szignó program.
- Minden aláíráshoz külön-külön csatoljuk a visszavonási információkat. Ha sok aláírást archiválunk, könnyen lehet, hogy ugyanazt a visszavonási információt nagyon sokszor tároljuk el.

  Ha sok aláírást ugyanaz az aláíró készített – ilyen pl. egy e-számlázó rendszer –, akkor az aláíró tanúsítványát és az aláíró tanúsítványára vonatkozó visszavonási információt sokszor tároljuk el. Szintén sokszor tároljuk el a tanúsítványláncokat, az időbélyegzés-szolgáltatók tanúsítványait, láncait és visszavonási információit stb.
- Ha visszavonási információként OCSP válaszokat használunk, általában jelentősen kisebb XAdES-A aláírások jönnek létre.
- Minden aláírást egy-egy archív időbélyegre vezetünk vissza. Ha n db aláírást archiválunk, minden egyes újra-időbélyegzéskor n db időbélyeget használunk fel. Ennek jelentős költsége lehet.
- Az aláírásokat egymástól függetlenül archiváljuk, így ha egy aláírást leválasztunk a többiről, az érvényessége önmagában is igazolható.
- Nem minden szükséges információ helyezhető el egy XAdES-A-ban. Például a XAdES-A nem tartalmazza, hogy mely időpontban melyek voltak az akkor elfogadott, biztonságos kriptográfiai algoritmusok, vagy a megbízható gyökértanúsítványok.
- Archív időbélyegzéskor minden egyes archivált aláíráshoz hozzá kell nyúlni; módosítani kell a XAdES struktúrákat. Ez egyes rendszerekben súlyos problémát jelent, például nagyon nehézzé válik a csak egyszer írható (WORM, write once, read many) médiák használata.

8.3.2. Csoportos időbélyegzés (LTANS)

Az archív aláírással történő, aláírásonként egy-egy archív időbélyeg elhelyezésénél jelentősen hatékonyabb módszerek is léteznek. Egyrészt kevesebb időbélyeggel is megoldható lehet az

[5] A legkülső archív időbélyeget csak az ellenőrzés időpontjára (*most*) nézve ellenőrizhetjük. Így bármilyen friss visszavonási információt is használunk, az abban szereplő `thisUpdate` soha nem lesz későbbi, mint a *most*.



8.1. ábra. **Az archiválandó adatobjektumokból lenyomatot képzünk, és a lenyomatokat tartalmazó összesítő fájlon helyezünk el archív időbélyegeket**

archiválás, másrészt ugyanazt a PKI objektumot – pl. ugyanazt a visszavonási listát – elegendő egyszer archiválni.

A legegyszerűbb, „naiv" módszer szerint az archiválandó aláírásokat és az érvényességüket alátámasztó információkat egyetlen fájlba foglaljuk, és e fájlon helyezünk el rendszeresen archív időbélyeget. Így egyetlen archív időbélyeggel bármennyi aláírást lefedhetünk. (Mielőtt az archív időbélyeg érvényességével probléma merül fel, az időbélyeggel ellátott nagy fájlhoz csatoljuk az időbélyeg érvényességét igazoló információkat, és ezen új fájlon helyezünk el archív időbélyeget.) E naiv megoldásnak egy nagyon súlyos hátránya is van. Egy archivált aláírás érvényességének igazolásához az összes archivált aláírásra és PKI objektumra szükség van.

> ***8.11. Példa:*** *Alajos több millió aláírást archivál, a szükséges PKI objektumokat egyetlen nagy ZIP fájlban helyezi el, és e ZIP fájlon helyez el archív időbélyeget. E ZIP fájl mérete jelenleg 500 Gigabyte körül van. Az aláírt dokumentumok között elektronikus számlák is vannak, és az adóhatóság számára az egyik számlán lévő aláírást kellene ellenőrizni. E kérésnek csak úgy tud eleget tenni Alajos, ha a teljes fájlt, minden benne lévő információval együtt átad az adóhatóságnak. Ez nemcsak azért problémás, mert az adóhatóság nehezen fogad be 500 Gigabyte-os fájlokat, hanem azért is, mert így Alajos minden nála lévő aláírt dokumentumot át kell, hogy adjon az adóhatóságnak, és ezt nem szeretné megtenni. Ráadásul lehet, hogy a számla mellett más dokumentumokban mások személyes adatai is szerepelnek, és Alajosnak nincsen jogalapja átadni ezen adatokat harmadik fél részére.*

A naiv megoldás jelentősen használhatóbbá tehető a következő módon: Az összes archiválandó PKI objektumról lenyomatot képzünk, a lenyomatokat helyezzük el egy összesítő fájlban, és e lenyomatokon helyezünk el archív időbélyeget. Ekkor egy aláírás érvényességének igazolásához csak a szükséges PKI objektumokra és a lenyomatokat tartalmazó, archív időbélyegekkel ellátott összesítő fájlra van szükség. Ehhez jelentősen kevesebb információt kell mozgatni, és a lenyomatokból az ellenőrző fél – a lenyomatképző függvény őskép-ellenállósága miatt – nem tud következtetni a többi PKI objektum tartalmára (lásd: 8.1. ábra).

E megoldás esetén nem elegendő, ha rendszeresen új archív időbélyeget helyezünk el az

összesítő fájlon. Az összesítő fájl képzéséhez használt lenyomatképző függvény elavulása esetén új, még „erős” lenyomatképző algoritmus segítségével újra kell képezni az összesítő fájlt, méghozzá úgy, hogy ne csak az archivált PKI objektumok, hanem az előző összesítő fájl létezésének időpontját is igazolja. Az Olaszországban működő archiválás-szolgáltatók ezt az elvet használják. [186]

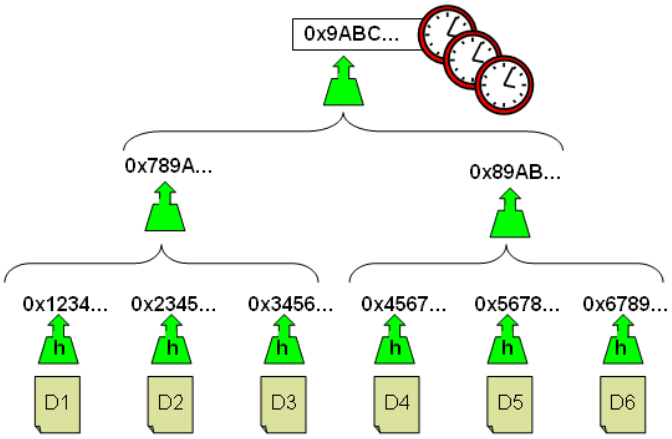
A IETF Long-Term Archive and Notary Services (LTANS) munkacsoportja által kidolgozott módszer a fenti megoldás általánosítására épül. Az LTANS megoldása közvetlenül nem aláírások archiválására, hanem nagy mennyiségű fájlra (bitsorozatra) kínál olyan megoldást, amelynek segítségével e fájlok létezésének időpontját hosszú távon is igazolni lehet.

Az LTANS által kibocsátott RFC 4998 egy hosszú távú archiválási szolgáltatást nyújtó rendszerre vonatkozó főbb követelményeket határozza meg: [148]

- Lehessen objektumokat feltölteni, letölteni és törölni.
- Szabályozott módon, hosszú távú archiválási szabályzat szerint működjön.
- Meg lehessen határozni, hogy mely objektumot mennyi ideig kell archiválni, lehessen az archiválási időt változtatni stb. Az RFC 4810 az „archivált objektumok menedzsmentje” kifejezést használja e funkciókra.
- Őrizzen meg olyan bizonyítékokat (pl. időbélyegeket), amelyek igazolják, hogy az archivált adatok nem változtak meg. (Így ne csak saját fizikai biztonságára és biztonságos folyamataira építse a megőrzést.)
- Gondoskodjon az archivált objektumok bizalmasságáról, és fel lehessen tölteni titkosított információkat is, méghozzá úgy, hogy az archiválás ne csak a titkosított, hanem a nyílt információra is vonatkozzon.
- Át tudja adni az objektumokat és a bizonyítékokat egy másik szolgáltatónak.
- Támogasson objektum-csoportokon végzett műveleteket.

Az LTANS munkacsoport munkájának központi elemét a bizonyítékok megőrzésére javasolt adatformátum jelenti. Az LTANS megoldása ún. Merkle-féle hash-fát épít az archivált objektumokból. [112] A Merkle-fa levelei maguk az archivált objektumok lenyomatai, míg az összes többi nem levél csomópontban a csomópont gyermekeiben szereplő adatok lenyomata található. A Merkle-fa gyökere így egy olyan lenyomat, amely az összes objektum lenyomatától, így az összes objektumtól függ.

Az archiválás során a Merkle-fa gyökerén helyezünk el archív időbélyegeket. Itt is igaz, hogy új archív időbélyeget kell elhelyezni, mielőtt az előző archív időbélyeg érvényessége problémássá válik (lásd: 8.2. ábra).



8.2. ábra. **Az archiválandó $D1$, $D2$, $D3$, ... adatobjektumokból lenyomatokat képzünk. A lenyomatokat csoportokba foglaljuk, és a lenyomatok csoportjaiból ismét lenyomatokat képzünk. A kapott lenyomatokat ismét csoportokba foglalhatjuk, és a csoportokból lenyomatot képezhetünk stb. Végül egy olyan lenyomathoz jutunk, amelynek értéke az összes archiválandó adatobjektumtól függ. Ezen a lenyomaton helyezünk el archív időbélyegeket.**

Problémát jelent, ha a Merkle-fa felépítéséhez használt lenyomatképző algoritmus elavul. Mielőtt ez megtörténik, újra kell építeni a Merkle-fát, az új Merkle-fában a régi fához tartozó objektumokon túl meg kell, hogy jelenjen a régi fa gyökere és a rajta elhelyezett archív időbélyegekből álló lánc.

Az LTANS munkacsoport ASN.1 DER, és XML alapú adatformátumot is meghatározott a fenti információk tárolására. Az ASN.1 DER formátumot az RFC 4998 írja le. [149] Az XML alapú adatformátumról egyelőre nem jelent meg hatályos RFC.

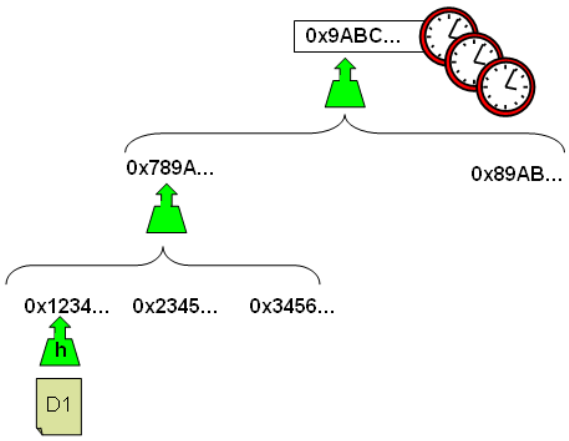
Az LTANS megoldása a következő tulajdonságokkal bír:

- PKI objektumok létezésének időpontját igazolja időbélyegek segítségével, nem foglalkozik azzal a kérdéssel, hogy egy aláírás érvényességét hogyan kell ellenőrizni, és milyen objektumokkal kell igazolni. A Merkle-fa levelein elvileg bármilyen[6] bitsorozat szerepelhet.

- Nagy mennyiségű aláírás archiválásakor sokkal hatékonyabb, mintha egyesével archiválnánk a XAdES/CAdES-A aláírásokat. A fa gyökerén elhelyezett időbélyeggel a teljes fa lefedhető.

- Egy PKI objektum létezésének időpontjának igazolásához a PKI objektum, az archív időbélyegek és a Merkle-fa egy „redukált” változata elegendő (lásd: 8.3. ábra). Elegendő

[6]Aláírások archiválása esetén több jó megoldást is követhetünk. Az egyik szélsőséges megközelítés, ha a Merkle-fa levelei az egyes dokumentumok, aláírások, visszavonási listák stb. A másik szélsőséges megközelítés, ha a Merkle-fa levelei XAdES/CAdES -X-L aláírások.



8.3. ábra. **Az előző ábrán szereplő hash-fát olyan módon redukáltuk, hogy csak a $D1$ objektum létezésének igazolásához szükséges elemeket őriztük meg.**

az adott levéltől a Merkle-fa gyökeréhez vezető hash-eket, és az ő közvetlen gyermekeiket bemutatnunk, ezek alapján már belátható, hogy az adott levél valóban az adott Merkle-fa levele. A redukált Merkle-fa mérete az archivált aláírások számának logaritmusával arányos, így sok archivált objektum esetén is viszonylag kis méretű információra van szükség. A redukciós algoritmust az RFC 4998 írja le részleteiben.

- Ha letörlünk egy objektumot, a fában a lenyomata ott maradhat, így törlés esetén nem kell újraépíteni a Merkle-fát.

- Ha egy új PKI objektumot adunk hozzá az archívumhoz, a Merkle-fát újra kell építeni, így e megoldás nem hatékony, ha dinamikusan változik az archivált objektumok köre.

A Németországban működő archiválás-szolgáltatók használnak Merkle-fákat az archivált dokumentumok és aláírások létezésének biztosításához.

8.4. Archiválás-szolgáltató

Az archiválás-szolgáltató feladata az elektronikusan aláírt dokumentumokon lévő elektronikus aláírások hosszú távú érvényességének biztosítása. Az archiválás-szolgáltatónak aláírásokat, illetve aláírt fájlokat küldhetünk be, és a szolgáltató megbízható, bevizsgált rendszer segítségével ellenőrzi az aláírásokat a vonatkozó szabványok, előírások szerint, majd az archiválás időtartama alatt a jogszabályi előírások szerint folyamatosan biztosítja az archivált aláírások hitelességét, így például rendszeresen időbélyegeket helyez el rajtuk. Az archiválás-szolgáltató ügyfelei kérésére kiadja az archivált dokumentumokat, valamint igazolást állít ki arról, hogy egy adott aláírás érvényes.

Az elektronikus archiválás-szolgáltatást az elektronikus aláírásról szóló 2001. évi XXXV. törvény definiálja, a következő módon:

> *Eat, 6 § „ (4) Az elektronikus archiválás-szolgáltatás keretében a szolgáltató*
> *a) a letagadhatatlanság biztosítása és a dokumentumok hiteles megőrzése céljából archiválja az archiválás időpontjában létező érvényességi láncot;*
> *b) biztosítja az érvényességi lánc sérthetetlenségét az ahhoz tartozó elektronikus aláírások érvényességének hosszú távú ellenőrizhetősége érdekében;*
> *c) az érvényességi láncot az igénybe vevő kérésére részére haladéktalanul átadja;*
> *d) kérelemre igazolást bocsát ki az általa archivált elektronikus dokumentummal vagy érvényességi lánccal kapcsolatban. ”*

Érvényességi lánc alatt a törvény az aláírás érvényességét alátámasztó összes információt érti (Eat. 2. § 14.), esetleg beleértve az aláírt dokumentumot is.

> *Eat, 2 § „ 14. Érvényességi lánc: az elektronikus dokumentum vagy annak lenyomata, és azon egymáshoz rendelhető információk sorozata, (így különösen azon tanúsítványok, a tanúsítványokkal kapcsolatos információk, az aláírás-ellenőrző adatok, a tanúsítvány aktuális állapotára, visszavonására vonatkozó információk, valamint a tanúsítványt kibocsátó szolgáltató elektronikus aláírás ellenőrző adatára és annak visszavonására vonatkozó információk), amelyek segítségével megállapítható, hogy az elektronikus dokumentumon elhelyezett fokozott biztonságú vagy minősített elektronikus aláírás, illetve időbélyegző, valamint az azokhoz kapcsolódó tanúsítvány az aláírás és időbélyegző elhelyezésének időpontjában érvényes volt. ”*

Az elektronikus aláírásról szóló törvény szerint, *ha minősített archiválás-szolgáltató archivál egy aláírást, vélelmezni kell, hogy az aláírást „jól” archiválják.* A minősített archiválás-szolgáltatókról a Nemzeti Média- és Hírközlési Hatóság vezet nyilvántartást, felügyeli őket, és éves rendszerességgel helyszíni ellenőrzést tart náluk – a minősített hitelesítés-szolgáltatókhoz hasonló módon.

A hazai elektronikus archiválás-szolgáltatók működését elsősorban az elektronikus aláírásról szóló törvény, és a 3/2005. IHM rendelet határozza meg. [180], [80] A jogszabályok a szolgáltatás főbb követelményeit rögzítik, az archiválás-szolgáltatás részleteire ajánlások vonatkoznak, amelyeket a Nemzeti Hírközlési Hatóság bocsátott ki. [120], [121], [122]

Az archiválás-szolgáltatás külföldön is ismert fogalom, például Németországban és Olaszországban is léteznek archiválás-szolgáltatók, de ott e fogalom nem az elektronikus aláírás törvényben szerepel, és az elektronikus aláírásról szóló EU direktíva sem szól archiválás-szolgáltatókról. A külföldi archiválás-szolgáltatók nem feltétlenül aláírt iratok

megőrzésével foglalkoznak, hanem bitsorozatok, akár aláíratlan dokumentumok hiteles megőrzésével foglalkoznak, és a hiteles megőrzést jellemzően az elektronikus aláíráshoz kapcsolódó technológiákkal biztosítják. Az archiválás-szolgáltatás nemzetközi viszonylatban is ritka jelenség, nagyon kevés szervezet foglalkozik ilyen tevékenységgel. Viszonylag új területről van szó, amely nem rendelkezik a hitelesítés-szolgáltatáséhoz hasonló letisztult és általános elfogadott interfésszel vagy követelményrendszerrel.

Az ETSI megkezdte egy archiválás-szolgáltatókra vonatkozó követelményrendszer és egy archiválás-szolgáltatók auditálására vonatkozó ajánlás kidolgozását. [48], [44]

Magyarországon a Microsec Kft. 2007 elején elsőként indította el minősített elektronikus archiválás-szolgáltatását, és ma is a Microsec Kft. az egyetlen, aktívan működő minősített elektronikus archiválás-szolgáltató az országban.

8.4.1. Dokumentum elhelyezése az archívumban

8.4.1.1. Dokumentum vagy lenyomat archiválása

Az Eat. két megközelítés szerinti archiválás-szolgáltatást tartalmaz. Az egyik megközelítés szerint az archiválás-szolgáltató az aláírt dokumentumot is megkapja és archiválja az aláírással együtt, míg a másik megközelítés szerint az archiválás-szolgáltató csak az aláírást kapja meg, és magát a dokumentumot nem.

Ez a második megközelítés arra az esetre nyújt megoldást, ha az ügyfél nem szeretné, hogy az archiválás-szolgáltató hozzáférjen a nyílt dokumentumhoz, de e megoldásnak van egy nagyon súlyos gyenge pontja: Ha az aláíráskor használt hash függvény elavul, előfordulhat, hogy valaki előállít egy másik olyan dokumentumot, amelyhez ugyanaz az aláírás tartozik. Ekkor az aláírás hitelessége elvész – nem lehet megállapítani, hogy eredetileg melyik dokumentumot írták alá. Ezért az Eat. előírja, hogy a dokumentumból rendszeresen (újabb és újabb algoritmusok segítségével) lenyomatot kell képezni, és a lenyomatokat rendszeresen be kell küldeni az archiválás-szolgáltatónak. Ezt a megoldást egyrészt bonyolultnak, nehézkesnek tartjuk, másrészt több súlyos problémát is látunk vele kapcsolatban:

- Ha a hash algoritmus „hirtelen” avul el (vagy hirtelen derül fény arra, hogy már régen elavult), előfordulhat, hogy az ügyfélnek nincs ideje beküldeni az új lenyomatot, és az archivált aláírások hitelessége elvész.
- Ha az ügyfél – véletlenül vagy szándékosan – nem jó lenyomatot küld be (vagy összekeveri a beküldött lenyomatokat), szintén elvész az archivált aláírások hitelessége, és ez lehet, hogy csak sokára, akár évtizedekkel később derül ki, amikor már nem lehet orvosolni a problémát.

Habár elvileg megoldható lenne, hogy az aláírást az aláírt dokumentum nélkül archiváljuk, a gyakorlatban ez nagyon nehezen képzelhető el. *A továbbiakban a dokumentum és az*

aláírás együttesen történő archiválásáról írunk, és az „érvényességi lánc" fogalmába az aláírt dokumentumot is beleértjük.

8.4.1.2. Hogyan helyezhetünk el dokumentumot az archívumban?

Az archiválandó dokumentumokat, aláírásokat, érvényességi láncokat valahogy el kell juttatnunk az archiválás-szolgáltatónak. Biztonságos megoldást célszerű választanunk, hogy útközben illetéktelen fél ne férhessen hozzá az archiválandó információkhoz.

Egyik lehetőség, hogy hálózaton keresztül – akár Interneten, akár például bérelt vonalon – küldjük el dokumentumainkat; ekkor használhatunk például HTTPS vagy VPN kapcsolatot vagy védett (pl. titkosított) levelezést. Másik lehetőség, hogy adathordozókon (pl. DVD-n) nyújtjuk be a dokumentumainkat (pl. valamilyen biztonságos eljárásrend keretében).

A szolgáltatók különféle lehetőségeket biztosíthatnak a dokumentumok elhelyezésére, és ezek időben változhatnak. Az aláírás hosszú távú hitelessége szempontjából nincs jelentősége, hogy egy dokumentum milyen módon jutott el a szolgáltatóhoz.

8.4.1.3. Érvényességi lánc felépítése

Amikor az archiválás-szolgáltató befogad egy dokumentumot, ellenőriznie kell az aláírást, és fel kell építenie az „érvényességi láncot", vagyis össze kell gyűjtenie minden olyan információt, amely igazolja az aláírás érvényességét. Ide tartozik a tanúsítványlánc (a végfelhasználói tanúsítványtól valamely megbízható gyökértanúsítványig), valamint a tanúsítványlánc minden elemére vonatkozó visszavonási információ (CRL vagy OCSP válasz), amely igazolja, hogy az adott tanúsítvány az aláírás pillanatában érvényes volt. Ha az aláíráshoz időbélyeg is tartozik, akkor az időbélyegre (vagy időbélyegekre) vonatkozó tanúsítványlánc és visszavonási információk is az érvényességi lánchoz tartoznak. (Lásd: 6.5. fejezet.) Az érvényességi lánc beszerzése is tipikusan valamilyen aláírási szabályzat szerint történik.

A jogszabályi követelmények értelmében az érvényességi láncot három napon belül be kell szerezni. [80] Ez többek között azt is jelenti, három napon belül kell olyan CRL-eket vagy OCSP válaszokat összegyűjteni, amelyeket az aláírás időpontját követően bocsátottak ki, hiszen ha az aláírás hitelességét hosszú távon szeretnénk biztosítani, akkor célszerű kivárási időt alkalmazni. A legtöbb hitelesítés-szolgáltató naponta bocsát ki CRL-t, de egyes szolgáltatók – különösen a gyökér hitelesítés-szolgáltatók – ritkábban. Így előfordulhat, hogy a jogszabályi követelmények tiszteletben tartása mellett egyes szolgáltatók aláírásait nem lehet (vagy nagyon nehéz) archiválás-szolgáltatónál elhelyezni.

8.12. Példa: *A Közigazgatási Gyökér Hitelesítés Szolgáltató (KGYHSZ) például 35 naponta bocsát ki CRL-t. [94] Ez azt jelenti, hogy egy KGYHSZ-re*

visszavezethető aláírást csak akkor lehetne – elvileg – feltölteni egy archiválás-szolgáltatóhoz, ha a KGYHSZ következő CRL-je várhatóan 3 napon belül meg fog jelenni.

CRL alapján történő ellenőrzés esetén nagyon nehéz elvi problémákkal állunk szemben. Az okozza a problémát, hogy a CRL-ek periodikusan jelennek meg, méghozzá az archiválás-szolgáltatótól független módon. Míg az aláírások többsége esetén ez nem okoz problémát, igen egzotikus speciális esetek is előfordulhatnak. A láncban szereplő egyes CRL-eket úgy kell összegyűjteni, hogy azok ne csak az aláírás időpontjára, hanem az láncban alattuk lévő CRL-ek kibocsátási idejére is vonatkozzanak (6.5. fejezet). Ebből adódóan a kivárási idők egyes láncokban összeadódhatnak, így könnyen előfordulhat, hogy egy aláírást a 3 napos időkorlát miatt nem fogadhat be egy archiválás-szolgáltató.

OCSP esetén – a mindig friss OCSP esetén (4.1.5.2. fejezet) – jelentősen egyszerűbb problémával állunk szemben, és ekkor nem jelent problémát a 3 napos korlát sem. A Microsec Kft. archiválás-szolgáltatása kizárólag olyan szolgáltatókra épülő aláírásokat fogad be, amelyek mindig friss OCSP szolgáltatást nyújtanak.

8.4.2. Az archívumban szereplő dokumentumok védelme

8.4.2.1. Fizikai védelem

Az archiválás-szolgáltatónak fizikailag biztonságos környezetben, megbízható rendszert kell üzemeltetnie, meghatározott bizalmi munkakörökkel kell rendelkeznie, és megbízható, auditálható szervezeti környezetet kell kialakítania. Ha a szolgáltatáshoz szolgáltatói kulcsokat használ – bár az archiválás-szolgáltatáshoz nincsen feltétlenül szükség szolgáltatói kulcsokra – azokat tanúsított kriptográfiai modulokban (HSM) kell tárolnia. Az archiválás-szolgáltató felelősséget vállal az archivált dokumentumokért és a rajtuk lévő aláírások hitelességéért, ezért stabil pénzügyi háttérrel kell rendelkeznie, és jogszabályban leírt pénzügyi követelményeket kell teljesítenie.

8.4.2.2. Hitelesség szempontjából

Az aláírások hosszú távú ellenőrizhetősége végett az archiválás-szolgáltató minősített elektronikus aláírással és minősített időbélyeggel látja el az archivált dokumentumokat:

- A szolgáltatási szabályzatában meghatározott időközönként (Eat. 16/G § (2) a)). Ezt célszerű úgy végezni, hogy az aláírások műszaki szempontból (8.1. fejezet) is ellenőrizhetőek maradjanak. Ez azt jelenti, hogy célszerű időbélyegezni:
  - mielőtt az előző aláírást védő, legkülső időbélyegzése lejár.

  - mielőtt az előző aláírást védő, legkülső időbélyegző tanúsítványát visszavonják.
  - mielőtt az előző aláírást védő, legkülső időbélyegzőhöz kapcsolódó kriptográfiai algoritmusokban meginog a bizalom.
  - biztos, ami biztos, néhány évenként.

- Ha a Nemzeti Média- és Hírközlési Hatóság határozatban előírja (Eat. 16/G § (2) b)). A Hatóság várhatóan közelgő algoritmusváltások esetén hoz ilyen határozatot.

A dokumentumok időbélyegzése számos módon elvégezhető, jó megoldást jelenthet például a XAdES-A (6.4.1.2.6. fejezet) és az LTANS ERS (8.3.2. fejezet). Érdekes, hogy az érvényességi láncokat nemcsak időbélyegzővel, hanem minősített elektronikus aláírással is el kell látni. Az aláírások létezési időpontjának igazolása szempontjából ennek nincsen jelentősége, erre azért van szükség, hogy az archiválás-szolgáltató eljárásaiban e lépés egyértelműen hozzárendelhető legyen egy adott személyhez.

Alapesetben sem a XAdES-A, sem az LTANS ERS nem támogatja, hogy az érvényességi láncot minősített aláírással lássuk el, ezért önmagában egyik technológia sem használható egy az egyben. A XAdES-A esetén további problémát jelent, hogy itt az archiválás (archív időbélyegzés) egyesével történik, és nem reális, hogy valaki egyesével minősített aláírással lásson el nagy tömegű dokumentumot. (Sok biztonságos aláírás-létrehozó eszköz megköveteli, hogy az aláíró minden egyes aláírás készítésekor gépelje be a PIN kódját, és ez több millió aláírás készítése esetén jelenthet némi problémát.)

Az archiválás-szolgáltatók jellemzően a XAdES-A (vagy CAdES-A) és az LTANS ERS egyes elemeinek kombinációját használhatják, és ezeket esetleg további eszközökkel egészíthetik ki.

8.4.2.3. Bizalmasság szempontjából

Az Eat szerint az archiválás-szolgáltató (az ügyfél felhatalmazása nélkül) nem ismerheti meg az archivált dokumentumok tartalmát. Ez a követelmény úgy értelmezhető, hogy a szolgáltató *munkatársai* nem ismerhetik meg a dokumentumok tartalmát, hiszen a szolgáltatónak mindenképpen kezelnie kell a nyílt dokumentumot, ugyanis az Eat. szerint ellenőriznie kell a dokumentumon lévő elektronikus aláírást.

Megoldást jelenthetne, ha az archiválás-szolgáltató olyan fájlokat archiválna, amelyeket először titkosítottak, és utána helyeztek el rajtuk elektronikus aláírást, de ekkor nagyon nehéz lehet például annak a bizonyítása, hogy milyen nyílt fájlt írt valaki alá. Ha egy dokumentumon először helyezünk el aláírást, majd ezt követően titkosítjuk, az korrekt, tiszta megoldás, de ha az archiválás-szolgáltató nem tudja feloldani a titkosítást, akkor nem tudja ellenőrizni az aláírást sem (holott ezt jogszabály írja elő a számára).

Egy dokumentumot célszerű először aláírni, majd utána az aláírással együtt titkosítani. Ha ezt fordítva tesszük, az problémákat vet fel (9.5. fejezet).

Megoldást jelent, ha az ügyfél titkosított (pl. SSL) csatornán küldi el a nyílt, aláírt dokumentumot az archiválás-szolgáltatónak. Az archív szolgáltató ellenőrzi az aláírást, kiterjeszti archív aláírássá, időbélyeget helyez el rajta, majd titkosítja az e-aktát. Ez a titkosítás már történhet olyan módon, hogy a titkosított e-aktát csak az arra jogosult ügyfél fejtheti vissza. Igaz, az archiválás-szolgáltatónak néha (nagyon ritkán) szüksége lehet a nyílt aláírásra az aláírások hitelességének biztosítása végett, így célszerű, ha fenntartja magának a lehetőséget, hogy – egy különleges eljárás keretében, több, bizalmi munkakört betöltő munkatárs együttes jelenlétében – visszanyerhesse a nyílt dokumentumokat.

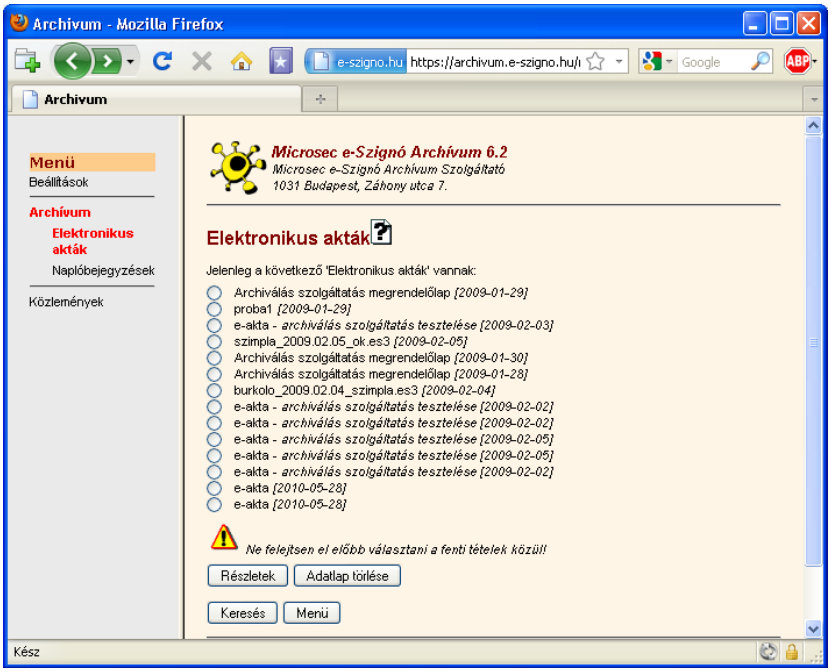
Célszerű figyelembe venni, hogy az archiválás-szolgáltató – az archiválás természetéből adódóan – olyan hosszú időtávon működik, amelyen az aláírásra használt kriptográfiai algoritmusok elavulhatnak. Ebből adódóan, *a titkosításhoz használt kriptográfiai algoritmusok is elavulhatnak az archiválás során*, így előfordulhat, hogy egy korábban még biztonságosan titkosított dokumentum hosszú idő után visszafejthetővé válik. Ezek szerint, ha feltöltünk egy titkosított dokumentumot az archiválás-szolgáltatóhoz, fel kell tételeznünk, hogy az archiválás időtartama alatt a szolgáltató vissza tudja fejteni a dokumentumot. A fentiekből egyúttal az is következik, hogy az archiválás-szolgáltatónak a titkosított dokumentumok bizalmasságát is védenie kell.

> ***8.13. Példa:*** *Alajos elhelyez egy dokumentumot az archiválás-szolgáltatónál 2010-ben. A szolgáltató titkosítva tárolja a dokumentumot, a titkosítást 2048 bites RSA-val végzi. Manfréd, a támadó 2015-ben betör az archiválás-szolgáltatóhoz, és megszerzi a titkosított dokumentumot, de nem tudja visszafejteni. A használt titkosítási algoritmus idővel kezd elavulni, ezért 2025-ben a szolgáltató újratitkosítja a dokumentumot 4096 bites RSA-val.*
>
> *Tételezzük fel, hogy a 2032-ben elérhető technológiákkal már néhány év alatt törhető*[7] *lesz az 2048 bites RSA. Igaz, hogy a szolgáltatónál már 4096 bites RSA-val van titkosítva a dokumentum, de Manfréd még a 2048 bites változatot szerezte meg. Manfréd megkezdi a titkosítás feltörését.*
>
> *2035-re Manfréd feltöri a titkosítást, és visszafejti a nyílt dokumentumot. A szolgáltató közben újratitkosítja a dokumentumot 8192 bites RSA-val, de ennek már nincs jelentősége, mert a nyílt dokumentum már rossz kezekben van.*

8.4.3. Érvényességi lánc elérhetőségének biztosítása

Az archiválás-szolgáltatónak biztosítania kell, hogy a jogosult felek elérjék az archívumban szereplő érvényességi láncokat. Ez általában azt jelenti, hogy a szolgáltatás igénybe vevői kereshetnek az archívumban, és letölthetik feltöltött dokumentumaikat. (Lásd: 8.4. ábra.)

[7] Ezt csak a példa kedvéért tételezzük fel, ilyen időtávra nagyon nehéz megalapozott állítást tenni a kriptográfiai algoritmusok erejéről.



8.4. ábra. **Az archivált dokumentumok például webes felületen keresztül tölthetőek le a minősített archiválás-szolgáltatótól**

Kérdés, hogy az érvényességi lánc pontosan mit is jelent. Ha az archiválás az aláírt dokumentummal együttesen történik, feltehetően beletartozik az aláírt dokumentum is, és feltehetően az érvényességi lánc részét képezi az archiválás-szolgáltató által a *feltöltés pillanatában* összegyűjtött érvényességi lánc. Nem feltétlenül tartozik ide a *mindenkori* érvényességi lánc, azaz a letöltés pillanatában, az esetleg sok évvel korábbi aláírás érvényességét alátámasztó összes információ. Ha az archiválás-szolgáltató XAdES/CAdES-A alapon működik, megoldható, hogy letölthető legyen az aktuális – azaz az időbélyegekkel kiegészített – XAdES/CAdES-A, LTANS ERS alapú archiválás esetén pedig az adott dokumentumra redukált Merkle-fa. Ugyanakkor nem valószínű, hogy letölthető vagy elérhető az archiválás-szolgáltató által elhelyezett minősített elektronikus aláírás, és a szolgáltató egyéb belső adatai. A technológia ezen része még nem tisztult le olyan mértékben, mint például ahogy az a hitelesítés-szolgáltatás vagy időbélyegzés-szolgáltatás esetén történt. Vegyük figyelembe, hogy a régi elektronikus aláírások ellenőrzése nem könnyű feladat, és a régi aláírások teljes érvényességi láncára nincsenek igazi, szabványos formátumok. E kérdéskört a következő fejezetben járjuk körül részletesen.

Az archiválás-szolgáltató célja az aláírások hiteles megőrzése. Hosszú távon, nagy biztonsággal kell működnie, ebből adódóan nem lehet olyan rugalmas, mint egy dokumentum-kezelő rendszer. A hozzáférési jogosultságok kevéssé rugalmasan módosíthatóak (különösen, ha titkosított dokumentumokról van szó), és a szolgáltató letöltéshez kapcsolódó díjai is lehet, hogy túl magasak ahhoz, hogy dokumentumainkat minden egyes alkalommal az archiválás-

szolgáltatóból nyissuk meg. Az archiválás-szolgáltató elsősorban arra szolgál, hogy ha elhelyeztük nála a dokumentumunkat, akkor szükség esetén egészen biztosan hozzá tudunk férni az eredeti, hiteles dokumentumhoz. A napi munka során valószínűleg nem a hosszú távú archívumot célszerű használni.

8.4.4. Igazolás kibocsátása

Ha egy archiválás-szolgáltató hosszú ideig (pl. 50 évig) tárol egy archív aláírást, rendszeresen új archív időbélyegekkel kell kiegészítenie. Előfordulhat, hogy az archiválás időtartama alatt változások következnek be. Változhatnak az elfogadott hitelesítés-szolgáltatók, időbélyegzés-szolgáltatók, az időbélyegek előállításához használható kriptográfiai algoritmusok, az aláírásokra és az archiválásra vonatkozó szabványok, jogszabályok és egyéb előírások. Így például megváltozhat az archív időbélyegek formátuma (pl. RFC 3161), az archív időbélyegek elhelyezésére és az archív aláírások formátumára vonatkozó specifikáció (pl. XAdES vagy CAdES), az archiválásra (pl. az archív időbélyegek elhelyezésére) vonatkozó követelmények, de akár időközben az is gyökeresen megváltozhat, hogy mit értünk elektronikus aláírás alatt.

8.14. Példa: *Alajos 2008-ban elhelyezte Bendegúz egy elektronikus aláírását egy minősített archiválás-szolgáltatónál. Az aláírás az e-Szignó program 3.1-es verziójával készült, XAdES v1.2.2 szerint. A szolgáltató 2010-ben kiegészíti egy archív időbélyeggel, ekkor az e-Szignó 3.2-es verzióját használja, és az új archív időbélyeg a XAdES v1.3.1 szerint készül.*

Tegyük fel, hogy majd 2015-ben új archív időbélyeget helyez el az aláíráson, az e-Szignó 4.7-es változata segítségével, az akkor elterjedt XAdES v2.0.6 szerint. Később, 2019-ben az új archív időbélyeg kerül az aláírásra az e-Szignó 6.1 segítségével, XAdES v3.2.1 szerint. Lehet, hogy ekkorra már nem RFC 3161 szerinti időbélyeg lesz az aláíráson, mert e formátumot akkor már nem használják. Lehet, hogy 2028-ra kihal a XAdES formátum, így az akkori archív időbélyeg majd – az e-Szignó 6.9-es változatával – az akkori YAdES 2.3 szerint készül.

Tekintve, hogy sem az eredeti elektronikus aláírás bitjei, sem a csatolt archív időbélyegek bitjei nem változtak meg, Dezső 2030-ban egy olyan aláírást ellenőriz, amelyhez:

- *nincs olyan aláírás formátum, amely egy az egyben ilyen lenne.*
- *nincs olyan program-változat, amely ilyen aláírásokat hozna létre.*

Az aláírást minősített archiválás-szolgáltató archiválja, így vélelmezni kell, hogy az aláírás érvényes. A szolgáltató gondosan járt el, így szükség esetén be tud mutatni minden olyan bizonyítékot, amely alapján az aláírás érvényessége műszakilag

> *visszavezethető az eredeti aláírás bitjeire, és megállapítható, hogy az aláírás érvényes.*
>
> *Ugyanakkor az aláírás érvényességének belátása nem egyszerű feladat, szakértői tevékenységet igényel.*

Archív aláírás segítségével igazolható, hogy az eredeti, kriptográfiai értelemben vett aláírás bitjei egy adott időpontban már léteztek, és az aláíráshoz használt tanúsítvány ezen időpontban még érvényes volt. Hosszú távon az archív aláírás mellett további információkra is szükség van (az egyes időpontokban elfogadott kriptográfiai algoritmusokról, megbízható gyökerekről stb.), illetve az aláírás hitelességének bizonyítása nagyon nehéz feladat lehet. Így előfordulhat, hogy hiába tölti le valaki az adott időpontban vett érvényességi láncot, ez olyan összetett struktúrát jelent majd, amit senki nem tud ellenőrizni.

E probléma feloldható, mert az archiválás-szolgáltatótól kérhető olyan igazolás, hogy egy adott dokumentumot archivál, és a dokumentumon lévő elektronikus aláírás vagy időbélyeg a feltöltés pillanatában érvényes volt. Ezen igazolást kiadhatja a szolgáltató papír alapon vagy elektronikusan, minősített elektronikus aláírással, az adott pillanatban elfogadott technológia szerint.

A minősített archiválás-szolgáltató által kiállított igazolás alapján bárki megállapíthatja, a szolgáltató szerint az aláírás érvényes, így kapcsolódik hozzá az Eat. szerinti vélelem. Így az aláírás elfogadásához nem szükséges a „muzeális” aláírásokat ellenőrizni, ezek vizsgálatára egyedül akkor van szükség, ha valaki meg akarja dönteni a minősített archiválás-szolgáltatáshoz kapcsolódó vélelmet, mert ekkor a szolgáltatónak be kell mutatnia a szükséges bizonyítékokat.

8.4.5. Érvényességi lánc törlése

A 3/2005. IHM rendelet előírja, hogy az igénybe vevő kérése vagy a szolgáltatási szerződés megszűnése esetén az archiválás-szolgáltatónak visszaállíthatatlanul törölnie kell az érvényességi láncot. [80] E törlés különösen nehéz feladat, mert a szolgáltatónak minden mentésből is törölnie kell az adott érvényességi láncot (beleértve a dokumentumot is). Ez azt a célt szolgálja, hogy sehogyan, semmilyen módon ne kerülhessen elő a szolgáltatótól az eredeti érvényességi lánc.

8.4.6. Megjeleníthetőség, értelmezhetőség biztosítása

Aláíráskor műszaki értelemben mindig egy bitsorozatot írunk alá, amely valamely számítógépes programmal létrehozott, a program segítségével látható, olvasható, értelmezhető állomány. Ha az aláírt bitsorozatot ezzel a programmal megnyitjuk, akkor a program értelmezi a bitsorozatot, és valamilyen értelmes tartalmat jelenít meg nekünk. Előfordulhat,

hogy ha ugyanazt a bitsorozatot másik programmal (vagy ugyanannak a programnak egy másik verziójával, esetleg ugyanannak a programnak egy másképpen konfigurált változatával) nyitjuk meg, a bitsorozat másképpen jelenik meg, esetleg más tartalommal rendelkező értelmes dokumentum jelenik meg. Hiába bizonyítható, hogy milyen bitsorozatot írtunk alá, az aláírásunk „letagadhatatlanságához" az is szükséges, hogy az aláírt tartalmat hogyan kell értelmezni, megjeleníteni.

A fejlett aláírás-blokk formátumok szerint éppen ezért nemcsak magát a dokumentumot kell aláírni, hanem például a dokumentum megjelenítésére vonatkozó információkat (pl. MIME típus) is. Ez néhány éven belüli archiválás esetén (jellemzően) elegendő, viszont hosszú távú – több évtizeden átívelő – archiválás esetén nem feltétlenül igaz. Előfordulhat, mint ahogy korábban már többször előfordult, hogy a használt fájlformátumok „kihalnak", és a jövőben megjelenő platformokon nem lehet majd megjeleníteni őket. Az Eat. szerint az archiválás-szolgáltató olyan szolgáltatást is nyújthat, amely szerint bizonyos fájlformátumok esetén vállalja, hogy az archivált dokumentumokat hitelesen meg tudja jeleníteni. Ehhez a szolgáltató meg kell, hogy őrizzen a fájl hiteles megjelenítéséhez szükséges szoftver és hardver eszközöket.

8.5. Mire jó az archiválás-szolgáltatás?

Az elektronikus archiválás-szolgáltatás egyrészt az elektronikus dokumentumok megőrzésére szolgál, másrészt azon túl, hogy a dokumentum rendelkezésre áll, e megőrzés hiteles, és a dokumentumokon elhelyezett elektronikus aláírások hosszú távú ellenőrizhetőségét is biztosítja.

Az elektronikus dokumentumok hiteles archiválásáról szóló rendelet szerint, ha elektronikus aláírás segítségével hosszú távon szeretnénk biztosítani egy dokumentum hitelességét, két dolgot tehetünk: archiválás-szolgáltatót bízunk meg e feladattal, vagy saját magunk látjuk el az archiválás-szolgáltató feladatkörét (pl. archív aláírással). [69] Bármelyik megoldást is választjuk, az aláírás mind műszaki, mind jogi szempontból hiteles marad, ilyen szempontból nincsen különbség e két megoldás között. Az jelent különbséget, hogy ha egy aláírt dokumentumot minősített archiválás-szolgáltató archivál, akkor egy bíróságnak vélelmeznie kell, hogy az aláírás valóban érvényes. A „házi" megoldásokhoz nem kapcsolódik ilyen jogkövetkezmény, ott előfordulhat, hogy az archiválónak kell bizonyítania, hogy valóban helyesen végzi az archiválást, és egy adott aláírás valóban érvényes.

Az elektronikusan aláírt dokumentumok hitelességének megőrzése nehéz feladat, nemcsak szakértelmet, de jelentős erőforrásokat igényel; többek között folyamatosan figyelemmel kell kísérni az elektronikus aláírással kapcsolatos technológiák fejlődését. Egy minősített archiválás-szolgáltató egységesen, a jogszabályoknak megfelelő módon végzi el mindezt, teljes körű szolgáltatást nyújt az aláírt dokumentumok megőrzésével kapcsolatban. Ha valaki minősített archiválás-szolgáltatónál helyez el egy dokumentumot, semmilyen további teendője

nincs a dokumentum hitelességével kapcsolatban. Az archiválás-szolgáltató egységesen felelőssé tehető az archiválásért, és egyúttal arra is garanciát jelent, hogy az archiváló nem járt el hanyagul: az elérhető legmagasabb szintű, bevizsgált, professzionális megoldást választotta.

8.6. Összegzés

- Ha azt szeretnénk, hogy egy elektronikusan aláírt dokumentum hitelessége hosszú távon is ellenőrizhető maradjon, a dokumentumot speciális módon kell megőriznünk.
- Minél hamarabb időbélyeget kell elhelyezni az aláíráson, különben az aláírás érvényessége megkérdőjelezhetővé válik, amint az aláíró tanúsítványa érvényét veszti (lejár vagy visszavonják).
- Az aláírást, illetve a dokumentumot újabb, erősebb időbélyegzővel célszerű ellátni, ha várható, hogy a közeljövőben az aláírást, illetve a dokumentumot védő „legkülső” archív időbélyeggel kapcsolatban az alábbiak valamelyike bekövetkezik:
  - Az időbélyegzés-szolgáltató tanúsítványa lejár.
  - Az időbélyegzés-szolgáltató tanúsítványát visszavonják.
  - Az időbélyeg készítéséhez használt kriptográfiai algoritmusokban megrendül a bizalom.
- Az aláírásokat időbélyegezhetjük egyenként is (így működnek az archív aláírások), vagy csoportosan is (így működik például az LTANS ERS).
- Az aláírások hitelességét védhetjük magunk is, de archiválás-szolgáltatót is megbízhatunk vele.
- Ha egy dokumentumot minősített archiválás-szolgáltató archivál, vélelmezni kell, hogy az archiválást „jól” végzi. Ha az archiválást magunk végezzük, nekünk kell bizonyítanunk, hogy megfelelően járunk el.
- Az archiválás-szolgáltató a dokumentumok olvashatóságának, értelmezhetőségének biztosítását is vállalhatja.

9. fejezet

Titkosítás PKI alapon

„You never realize how good your crypto is until you lose your key."
(Akkor érted meg igazán, mennyire erős kriptográfiát használtál, amikor elveszíted a [dekódoló] kulcsodat.)

– *Marcus Ranum, The Sourcefire Computer Security Calendar*

„Gentlemen do not read each others mail."
(Úriemberek nem olvassák el egymás leveleit.)

– *Henry Lewis Stimson*

A titkosítás azt jelenti, hogy egy dokumentumot olyan speciális módon kódolunk, hogy annak tartalmát illetéktelen személyek ne olvashassák el. A titkosítás általában valamilyen titok (jelszó vagy kulcs) segítségével oldható fel, amelyhez illetéktelen személyek nem férhetnek hozzá.

Egyszerű megoldás, ha ketten közös titokban állapodnak meg. Lesz egy közös jelszavuk (vagy kulcsuk), és aki titkos üzenetet szeretne küldeni a másiknak, az ezzel titkosíthatja az üzenetet, és a címzett is ugyanezen kulccsal fejtheti vissza. Ezt a megoldást nevezik titkos kulcsú vagy szimmetrikus kulcsú titkosításnak. Jelentős korlátja e megoldásnak, hogy a feladónak és a címzettnek valamilyen biztonságos módon kell megállapodnia a közös titokban. Ha a közös titokban nem biztonságos csatornán állapodnak meg, illetéktelen személy is megismerheti azt, aki így később megfejtheti a titkosított üzeneteiket.

Megoldást jelent e problémára, ha a titkosításhoz tanúsítványokat használunk. A titkosítás (kódolás) ekkor a tanúsítványban szereplő nyilvános kulcs alapján történik (e technológia neve nyilvános kulcsú titkosítás), ha valakinek a nyilvános kulcsával kódolunk valamit, azt ő a hozzá tartozó magánkulcsával tudja visszafejteni. Nyilvános kulcsú titkosítás segítségével *úgy is küldhetünk valakinek titkos üzenetet, hogy előtte semmilyen közös titkos jelszóban, kulcsban*

nem állapodtunk meg. Így akár olyannal is létesíthetünk titkos, biztonságos kapcsolatot, akit még nem is ismerünk; elég, ha a tanúsítványa alapján megállapítjuk, hogy mi az ő nyilvános kulcsa.

A továbbiakban a nyilvános kulcsú infrastruktúra (PKI) használatával történő titkosításról lesz szó.

9.1. Dokumentum titkosítása

Ha PKI alapon szeretnénk valakinek titkosított üzenetet küldeni, be kell szereznünk az illető fél tanúsítványát, amely tartalmazza az ő nyilvános kulcsát. Az üzenetet a nyilvános kulcs alapján titkosíthatjuk a korábban (2. fejezet) bemutatott algoritmusok segítségével. A továbbiakban elsősorban a titkosítás PKI-vel kapcsolatos vonatkozásait mutatjuk be.

9.1.1. Titkosító tanúsítvány

Azon X.509 tanúsítványokat nevezzük *titkosító tanúsítvány*nak, amelyeket titkosításra használnak. Formailag nagyon kevésben különbözik a titkosító tanúsítvány az aláíró tanúsítványtól, elsősorban annyiban, hogy a tanúsítványt kibocsátó hitelesítés-szolgáltató a tanúsítványban szereplő kulcshasználati (key usage) bitekben jelezheti, hogy a tanúsítványt titkosításra célszerű használni.

A titkosító tanúsítványban általában a `keyEncipherment` és a `dataEncipherment` kulcshasználati bitek szerepelnek, vagy együttesen, vagy csak egyikőjük. A `dataEncipherment` azt jelenti, hogy a nyilvános kulccsal magát az üzenetet titkosítjuk, míg a `keyEncipherment` azt jelenti, a nyilvános kulcsot az üzenet titkosítására használt szimmetrikus kulcs titkosítására használjuk. Gyakran szerepelnek a titkosító tanúsítványban kiterjesztett kulcshasználat (extended key usage) állítások is, például az e-mailek titkosítására használt tanúsítványokban a `secureEmail` állítás.

Az titkosításra használt tanúsítványt és kulcspárt célszerű szétválasztani az aláírásra használt tanúsítványtól és kulcspártól, mert:

- Az Eat. 13. § (4) tiltja, hogy az aláíró az aláírásra használt magánkulcsot aláíráson kívül bármi másra használhassa.
- A titkosító tanúsítványnak egészen más az életciklusa. Például titkosító tanúsítvány esetén létezhet kulcsletét szolgáltatás, azaz előfordulhat, hogy a tanúsítványhoz tartozó magánkulcs nem kizárólag a tanúsítvány alanyának a birtokában van. (Lásd: 3.3.3. fejezet.) Aláírás esetén ez nem engedhető meg.

A titkosításra használt tanúsítványt és kulcspárt célszerű szétválasztani az autentikációra használt tanúsítványtól és kulcspártól, mert:

- Autentikációkor véletlen kihívást kódolunk a magánkulcsunkkal, ha ugyanazt a kulcsot egyúttal titkosításra is használjuk, egy támadó a véletlen kihívás helyett akár dekódoltathat velünk egy nekünk szóló üzenetet is. (Lásd: 10.2. fejezet.) Ez ellen valamelyest védelmet nyújt, hogy az autentikáció és a titkosítás más paddinget használ.

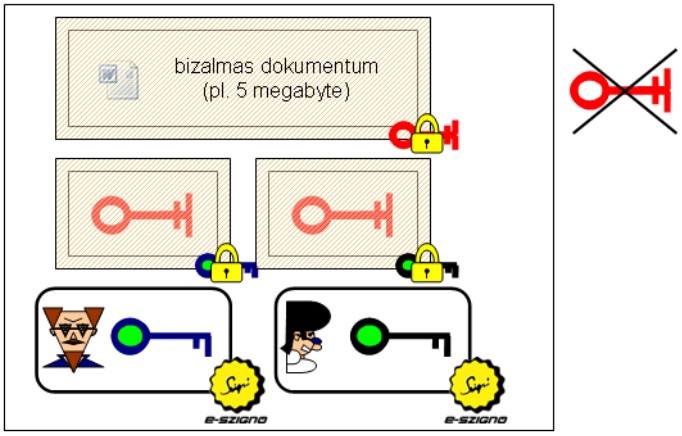
Az aláíró és autentikációs tanúsítványokat lejártuk után nem lehet, nincs értelme használni. A hozzájuk tartozó magánkulcsokat – ha a tanúsítvány nem kerül megújításra – célszerű megsemmisíteni. A titkosító tanúsítvány ebben is eltér: *titkosító tanúsítvány esetén előfordulhat, hogy a tanúsítvány lejárta után használni kell a magánkulcsot*, a korábban titkosított dokumentumok elérésére, visszafejtésére. Az is előfordulhat, hogy egy adott üzenetet csak a tanúsítvány lejárta után kap meg a címzett, így már csak érvénytelen tanúsítvány esetén használja a magánkulcsot.

> ***9.1. Példa:*** *Alajos titkosított üzenetet küld Bendegúznak, az üzenetet a Bendegúz tanúsítványában lévő nyilvános kulccsal titkosítja. Bendegúz szabadságon van, egy héttel később kapja meg az üzenetet, a tanúsítványa addigra már lejárt. Ez nem baj, mert a titkosító tanúsítvány magánkulcsát a tanúsítvány lejárta után is szabad használni, így visszafejtheti az üzenetet.*

Lényeges, hogy *az elektronikus aláírásról szóló törvény nem vonatkozik a titkosító tanúsítványokra.* Így, ha egy hitelesítés-szolgáltató titkosító tanúsítványt bocsát ki, nem vonatkozik rá az Eat. Ekkor kizárólag a szolgáltató szolgáltatási szabályzatai és a hitelesítési rend alapján dönthetjük el, hogy mit várhatunk el a tanúsítványtól. Ennek ellenére *sok szolgáltató nagyon hasonlóan kezeli a titkosító (és autentikációs) tanúsítványokat, mint az aláíró tanúsítványokat.*

Azt követően, hogy egy aláíró vagy autentikációs tanúsítvány visszavonásra került, már nem lehet visszaélni a magánkulccsal. A titkosító tanúsítványokra ez nem igaz. Ha a támadónál már vannak lehallgatott titkosított üzenetek amikor megszerzi a magánkulcsot, akkor ezen üzeneteket vissza tudja fejteni, és ebben nem akadályozza őt meg, ha a titkosító tanúsítványt visszavonják. A visszavonással csak az érintett feleket értesítjük, hogy a későbbiekben ne ezzel a nyilvános kulccsal titkosítsák a nekünk szóló üzeneteket, mert a hozzá tartozó magánkulcs már illetéktelen kezekben lehet. A titkosító tanúsítvány visszavonása nem szünteti meg a visszaélés lehetőségét, csak mérsékeli a várható károkat.

Akármilyen követelmények is vonatkoznak az aláírásra, titkosításra és autentikációra használható kulcspárokra, akármi is szerepel a tanúsítványban, egy tanúsítványt ellenőrző érintett fél olyan tanúsítványt fogad el, amilyet csak kedve tartja. Így egy érintett fél akár elfogadhat aláírói tanúsítványt is titkosításra. (Az aláíró tanúsítvány alapján titkosított üzenetet viszont az címzett lehet, hogy nem tudja majd kicsomagolni, mert például az intelligens kártyája az aláíráshoz használt paddinget követeli meg.)



9.1. ábra. **Titkosított üzenet összeállítása**

9.1.2. Titkosított üzenet összeállítása

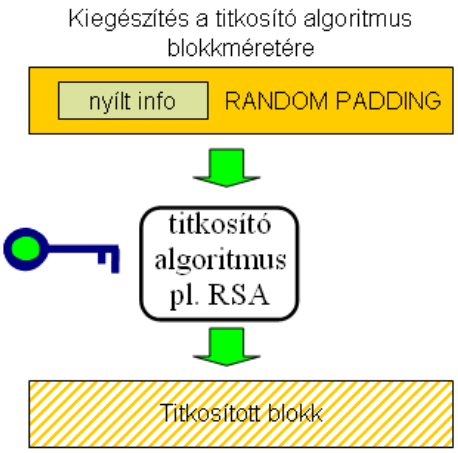
A feladó a címzett nyilvános kulcsa segítségével titkosítja az üzenetet, de általában nem közvetlenül a nyilvános kulccsal titkosít. A feladó egy friss, véletlen szimmetrikus kulcsot választ ki, és e szimmetrikus kulccsal titkosítja az üzenetet. A titkosítás általában valamilyen blokkrejtjelező-üzemmód (pl. CBC, cipher block chaining) szerint történik; a blokkrejtjelező-üzemmódokról a „Kriptográfia és alkalmazásai” című könyvben olvashatunk részletesen. [21] A titkosított üzenet a szimmetrikus kulccsal titkosított nyílt üzenetből és a címzett nyilvános kulcsával titkosított szimmetrikus kulcsból áll. A feladó más célra nem használja e szimmtrikus kulcsot, és annak nyílt változtatát megsemmisíti. (Lásd: 9.1. ábra.)

A címzett a saját magánkulcsával vissza tudja fejteni a szimmetrikus kulcsot, a szimmetrikus kulcs segítségével pedig hozzájut a nyílt üzenethez.

E megoldásnak több előnye is van azzal szemben, ha a teljes üzenetet a címzett nyilvános kulcsával titkosítanánk. Egyrészt, az üzenet akár nagyon hosszú is lehet, és a nyilvános kulcsú műveletek lassúak. Sokkal gyorsabb a szimmetrikus kulccsal titkosítani a hosszú üzenetet. Másrészt, a támadó így kevesebb olyan blokkot figyelhet meg, amelyet a nyilvános kulccsal titkosítottak, így kevesebb információja van a nyilvános kulcs támadásához. Harmadrészt, így hatékonyan titkosíthatunk több címzett számára is: a szimmetrikus kulcsot külön-külön titkosíthatjuk minden egyes címzett nyilvános kulcsával. (Lásd: 9.1. ábra.) Ha az üzenet hosszú, a több címzettnek titkosított üzenet nem lesz jelentősen hosszabb, mintha csak egyetlen címzettnek titkosítottuk volna.

A szimmetrikus kulcs általában rövidebb, mint a nyilvános kulcsú titkosító algoritmus blokkmérete, ezért a szimmetrikus kulcsot paddinggel egészítjük ki. (Lásd: 9.2. ábra.) *Titkosítás esetén a padding véletlen* biteket is tartalmaz. (Lásd: 9.3. ábra.) Általában is igaz, hogy titkosításhoz randomizálásra van szükség: ha teljesen determinisztikus algoritmussal[1]

[1]RSA esetén emiatt különösen fontos szerep jut a paddingnek. Az ElGamal titkosítások már eleve



9.2. ábra. **Titkosított blokk összeállítása**

titkosítunk, akkor a támadó felismeri, ha valaki azonos kulccsal azonos üzenetet titkosított.

9.2. Példa: *Bendegúz kincset keres, Alajos távolról, titkosított üzenetekkel irányítja. Manfréd, a támadó lehallgatja a csatornát, valamint figyeli Bendegúzt, és megpróbálja kideríteni, hogy Alajos mikor mit üzen Bendegúznak.*

Alajos első üzenete: „Menj száz lépést Észak felé." Az üzenetet Bendegúz nyilvános kulcsával titkosítva küldi el. Manfréd hiába hallgatja le az üzenetet, nem tudja visszafejteni.

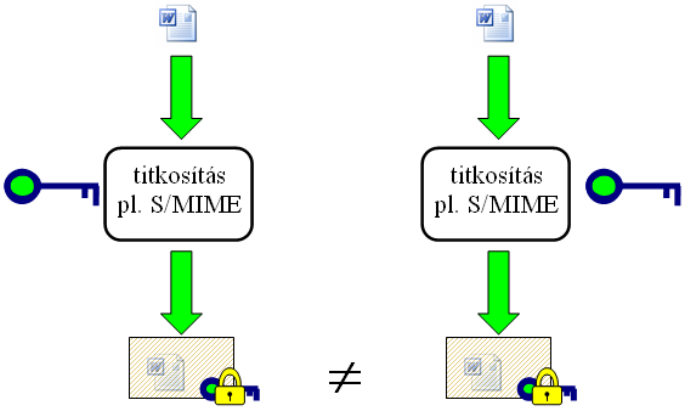
Alajos második titkosított üzenete: „Menj húsz lépést Kelet felé." Továbbra is igaz, hogy csak Bendegúz tudja elolvasni az üzenetet, Manfréd nem.

Alajos harmadik üzenete: „Menj száz lépést Észak felé." Ha Alajos nem használt randomizálást a titkosítás során, Manfréd észreveheti, hogy Alajos ugyanazt a bitsorozatot küldte el, mint az első esetben. Első alkalommal látta, hogy Bendegúz száz lépést ment Észak felé, így rájöhet, hogy most is ugyanazt kell tennie. Ez alapján Manfréd Bendegúz elé kerülhet, és hamarabb találhatja meg a kincset.

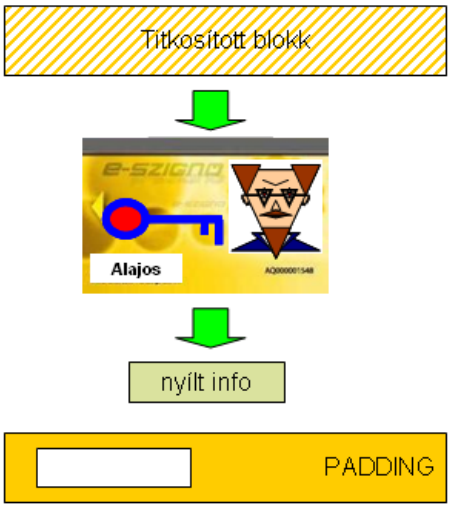
Ha Alajos randomizált a titkosítás során, Manfréd nem jöhet rá, hogy megint ugyanaz az üzenet haladt át a csatornán.

A PKI alapú, nyilvános kulcsú titkosítást használó megoldások mind a fenti elvek szerint működnek. Ilyen megoldás például a levelezőprogramok által használt S/MIME, és így működik a World Wide Web Consortium által kidolgozott XML Encryption is. [146], [193]

tartalmaznak randomizálást, így ott ez kevésbé lényeges.



9.3. ábra. **Ha egy fájlt kétszer titkosítok, nem feltétlenül ugyanazt a bitsorozatot kapom eredményül**



9.4. ábra. **Titkosított dokumentum visszafejtése**

9.2. Titkosított dokumentum visszafejtése

Aki titkosított üzenetet kap, az megvizsgálja, hogy a szimmetrikus kulcs csatolt titkosított változatai közül melyiket tudja visszafejteni saját magánkulcsával. Ha sikerül a visszafejtés (ez abból derül ki, hogy érvényes paddinget kap), eldobja a paddinget, és a kapott szimmetrikus kulcs segítségével visszafejti a titkosított üzenetet. (Lásd: 9.4. ábra.)

9.3. Titkosított dokumentum archiválása

Titkosított dokumentumok hosszú távú megőrzése esetén a következőkre célszerű tekintettel lenni:

- A titkosított dokumentumokon kívül egyúttal meg kell őrizni a visszafejtésükre szolgáló magánkulcsot is, mert ha a magánkulcs elvész, már nem lehet kideríteni, mi volt titkosítva. A magánkulcsot nem őrizhetjük ugyanúgy, ahogy a titkosított dokumentumokat őrizzük, mert ha egy támadó hozzáfér a titkosított dokumentumhoz, akkor így a magánkulcshoz is hozzáfér, azaz nem volt értelme a titkosításnak. A dekódoló magánkulcsot máshogy, máshol, esetleg magasabb biztonsági követelmények között célszerű őrizni.

- Célszerű végiggondolni, milyen problémák merülhetnek fel, ha a titkosító tanúsítványunk változik. Ha az új titkosító tanúsítványban ugyanaz a magánkulcs szerepel, akkor a korábban titkosított dokumentumokat továbbra is vissza tudjuk fejteni. (Vigyázat, nem minden alkalmazás támogatja, ha a címzett tanúsítványa nem azonos a mi aktuális tanúsítványunkkal, sok alkalmazás nem veszi észre, hogy a magánkulcs ennek ellenére használható.)

  Ha az új titkosító tanúsítványhoz más kulcspár tartozik, akkor is célszerű végiggondolni, hogyan fogjuk elérni a korábban titkosított dokumentumainkat. Egyik lehetőség, hogy megőrizzük a korábbi magánkulcsunkat is; ez azt jelenti, hosszú távon sok magánkulcsot kell őriznünk. Másik lehetőség, hogy a korábban titkosított dokumentumokat áttitkosítjuk, hogy ezentúl az új kulcspárral lehessen dekódolni; sok dokumentum esetén ez jelentős erőforrásokat igényel.

- Hosszú ideig tartó megőrzés esetén előfordulhat, hogy a titkosító algoritmusok elavulnak, és ekkor a korábban még biztonságos titkosítás nem feltétlenül jelent elég erős védelmet. (Ez nem feltétlenül jelenti azt, hogy valóban vissza tudjuk fejteni a titkosított dokumentumokat a kulcs nélkül, csak annyit jelent: már nem bízhatunk abban, hogy a támadó nem tudja visszafejteni őket.)

9.4. Titkosított dokumentum megsemmisítése

A titkosított dokumentumok megsemmisítésének egyik módja, ha a dekódolásra használt kulcsokat megsemmisítjük. Ekkor egyáltalán nem lehet visszafejteni a titkosított dokumentumokat – egészen addig, amíg a titkosításra használt algoritmusok biztonságosak maradnak. E megoldáshoz az szükséges, ha *minden* olyan kulcsot megsemmisítünk, amellyel visszaállítható a nyílt dokumentum, beleértve a letétbe helyezett magánkulcsokat is.

9.5. Titkosítás és aláírás

A titkosítás és az aláírás során is megváltoztatjuk a dokumentumot (aláíráskor ez főként azt jelenti, hozzáillesztjük az aláírást), és a kódolt dokumentum szerkezet biztosítja a titkosságot,

illetve a hitelességet. E két technológia együttes alkalmazásakor célszerű figyelembe venni az alábbi szempontokat.

- Ha aláírt dokumentumot[2] titkosítunk, a titkosítás miatt nem lehet ellenőrizni az aláírást. Az aláírás ellenőrzéséhez először vissza kell fejteni a titkosítást.
- Ha titkosított dokumentumot írunk alá, akkor ugyan lehet ellenőrizni az aláírást, de csak annyi állapítható meg, hogy az aláíró aláírt *valamit*. Esetleg nagyon nehéz lehet bizonyítani, hogy pontosan milyen értelmes, nyílt dokumentumra vonatkozik az aláírás.
  - E bizonyításhoz a titkosítás címzettje fel kell, hogy fedje a dekódoló magánkulcsát. Ezt valószínűleg nem szívesen teszi meg, mert ekkor más, neki szóló titkosított üzenetek is könnyen dekódolhatóakká válhatnak. Ha a kulcsot például intelligens kártya védi, lehet, hogy a magánkulcs egyáltalán nem nyerhető ki belőle.
  - Ha ismét titkosítjuk a nyílt üzenetet a címzett nyilvános kulcsával, akkor a randomizálás miatt nem az aláírt titkosított üzenetet fogjuk visszakapni, hanem egy másik bitsorozatot.
  - A titkosító algoritmusoknak nem célja a hitelesség biztosítása, nem erre valók. Előfordulhat, hogy a titkosított üzenetet egy másik dekódoló kulccsal egy másik nyílt üzenetté dekódolhatjuk, és ekkor az aláírás ezen másik dokumentumhoz is tartozhatna.

    Például ha tökéletes titkosítást – one-time-pad-et (2.5.2. fejezet) – használunk titkosításra, és az így titkosított üzenetet írjuk alá, akkor bármilyen (megfelelő hosszúságú) üzenethez található olyan dekódoló kulcs, amely alapján az aláírt üzenet az adott üzenetbe dekódolható.

    9.3. Példa: *Alajos one-time-pad segítségével az X bitsorozatot titkosította a K kulcs-bitsorozattal, és az Y bitsorozatot kapta eredményül. Alajos az Y üzenetet írta alá. Manfréd az X' bitsorozatról szeretné bizonyítani, hogy Alajos aláírása arra vonatkozik, így keres egy olyan K' kulcsot, amellyel ha titkosítjuk az X' üzenetet, akkor az Y üzenetet kapjuk eredményül. Az alábbi módon számíthatja ki e K' kulcsot:*
    $K' = Y \oplus X'$
    ahol az $\oplus$ művelet bitenkénti modulo2 összeadást jelent.

- Ha kapunk valakitől egy titkosított és ezt követően aláírt dokumentumot, abból nem következik, hogy aki az aláírást készítette, az ismerte a nyílt dokumentum tartalmát.

  9.4. Példa: *Alajos felad egy rejtvényt, 1 millió forintot fizet annak, aki először elküldi neki a megfejtést. Titkosítva és aláírva várja a rejtvény*

[2]Például egy fájlt és a rajta lévő aláírást egyaránt tartalmazó e-aktát.

9.5. ábra. **Titkosítás és aláírás**

megfejtését; a titkosítás biztosítja, hogy senki nem „lopja el" más eredményét, és Alajos az aláírás alapján dönti el, hogy ki küldte a megfejtést.

Bendegúz fejti meg először a rejtvényt, először titkosítja, és utána aláírja a megfejtést, és így küldi el Alajosnak.

Manfréd elkapja az üzenetet a hálózaton. A titkosítás miatt nem tudja megállapítani, hogy milyen megfejtést küldött Bendegúz, de ez nem gátolja meg abban, hogy levegye Bendegúz aláírását a titkosított üzenetről, és helyette ő maga írja alá.

Az így kapott üzenet alapján Alajost azt fogja hinni, hogy Manfréd fejtette meg a rejtvényt.

Ha titkosítást és aláírást együtt kell használni, először mindig az aláírást helyezzük el a dokumentumon, és csak az aláírt dokumentumot titkosítsuk.

9.6. Titkosítás és biztonságos csatorna

Többféle módszer létezik a titkosított kommunikációra. Egyik lehetőség, hogy magát az üzenetet titkosítjuk, és e titkosított üzeneteket küldjük el – akár nyílt csatornán. Jelen fejezetben elsősorban ezt a megoldást mutatjuk be. Másik lehetőség, hogy titkosított csatornát építünk ki a címzettel, és e titkosított csatornán keresztül küldöm el az üzenetet. E megoldásról az autentikációról szóló fejezetben (10. fejezet) írunk.

Ha titkosított üzenetet küldök, a titkosított üzenet szerkezete biztosítja, hogy csak a címzett olvashatja el, így a titkosság független attól, hogy ki és hol tárolja a titkosított üzenetet. Mindez egészen addig teljesül, amíg valaki külön el nem menti a visszafejtett, nyílt üzenetet. Vegyük figyelembe, hogy a titkosított üzenet visszafejtéséhez szükség van a megfelelő magánkulcsra. E megoldás nem biztosít hitelességet, ehhez például elektronikus

aláírás szükséges. E megoldás annak felel meg, ha egy lezárt páncélkazettában küldünk el egy levelet, amelyet kizárólag a kulcs birtokában lehet kinyitni.

Ha titkosított biztonságos csatornán küldöm el az üzenetet, akkor az üzenet kizárólag addig titkos, amíg a csatornán halad keresztül. Amint kikerül a csatornából, már más módszerekkel kell védeni. E megoldás egyúttal bizonyos mértékű hitelességet is biztosít. Akihez biztonságos csatornán keresztül érkezik egy üzenet, az biztos lehet benne, hogy a csatorna másik végén lévő fél küldte neki, és az üzenet útközben nem változott meg. Mindezt viszont már nem tudja bizonyítani harmadik fél számára. (Ha ezt is szeretnénk, ahhoz elektronikus aláírásra van szükség.) A biztonságos csatorna mindig csak két fél között épül ki (mert az elterjedt protokollok csak ezt támogatják). E megoldás annak felel meg, ha egy védett csőben küldöm át a levelet a címzettnek.

A titkosítás és a biztonságos csatorna jól alkalmazható akár együttesen is; például titkosított üzenet is küldhető biztonságos csatornán keresztül.

9.7. Összegzés

- A titkosító tanúsítványok életciklusa egészen más, mint az aláíró vagy autentikációs tanúsítványoké.
- Általában nem a teljes dokumentumot titkosítjuk nyilvános kulcsú algoritmussal, hanem friss, véletlen szimmetrikus kulcsot sorsolunk ki, ezzel titkosítjuk a dokumentumot, és a szimmetrikus kulcsot titkosítjuk nyilvános kulcsú algoritmussal.
- A gyakorlatban titkosítás során mindig kell használni randomizálást is; ilyen a véletlen szimmetrikus kulcs, és ilyen a véletlen padding.
- A titkosítva tárolt dokumentumok kezeléséhez menedzselni kell a dekódolásukra használható kulcsokat is.
- Ha a titkosítást aláírással kombináljuk, célszerű alaposan végiggondolni, hogy mit várunk e megoldástól. Célszerű először aláírni, és utána titkosítani.

10. fejezet

Autentikáció (partner hitelesítés) PKI alapon

> *„If you wake up at a different time, in a different place, could you wake up as a different person?"*
> *(Ha felébredhetsz máskor és máshol, felébredhetsz-e másvalakiként?)*
>
> – *Chuck Palahniuk, Harcosok klubja*

Az autentikáció, más néven *partner hitelesítés* vagy *biztonságos azonosítás* azt jelenti, hogy valamilyen biztonságos módon, jellemzően kódolási, kriptográfiai módszerek segítségével megbizonyosodunk róla, hogy azzal kommunikálunk-e, akivel szeretnénk. E fejezetben arról lesz szó, hogy a PKI milyen segítséget nyújt ebben, így a továbbiakban az autentikációnak elsősorban a PKI vonatkozásairól lesz szó.

A PKI alapú autentikáció általában a következő elvek szerint történik:

1. Megszerezzük a másik fél tanúsítványát, és meggyőződünk a tanúsítvány érvényességéről. Így hitelesen hozzájutottunk a tanúsítványba foglalt nyilvános kulcsához, biztosak lehetünk benne, hogy az valóban az ő nyilvános kulcsa[1].

2. Generálunk egy friss véletlen számot, ezt nevezzük *kihívásnak*. E véletlen kihívást küldjük el a másik félnek.

3. A másik fél – akit biztonságosan azonosítani szeretnénk – megválaszolja a kihívást: a kihívásban szereplő véletlen számot a saját tanúsítványához tartozó magánkulcsával kódolja. (E kódolást csak ő tudja elvégezni, mert az ő tanúsítványához tartozó magánkulcs kizárólag az ő birtokában van.) A kódolás eredményét visszaküldi nekünk.

[1] Pontosabban, abban lehetünk biztosak, hogy a tanúsítványba foglalt nyilvános kulcshoz tartozó magánkulcs kizárólag a tanúsítványban feltüntetett fél birtokában van.

4. Ellenőrizzük, hogy a másik fél helyesen válaszolt-e a kihívásunkra: a tanúsítványa – pontosabban a tanúsítványában lévő nyilvános kulcsa – segítségével ellenőrizhetjük, hogy a kódolást a tanúsítványhoz tartozó magánkulccsal végezték-e el.

Az ilyen módon történő biztonságos azonosítást *kihívás és válasz* alapú azonosításnak is nevezik.

Szorosan nem tartozik ugyan az autentikációhoz, de az autentikáció során általában biztonságos – kriptográfiai módszerek segítségével titkosított és hitelesített – csatorna is felépül köztünk és a másik fél között. Így az autentikációt követően e csatornán biztonságos módon cserélhetünk információt a másik féllel. Ilyen alapokon nyugszik az SSL (vagy TLS), az SSH, és sok tanúsítvány-alapú VPN megoldás is.

Szinte minden Internet-felhasználó használt már tanúsítvány-alapú autentikációt. Amikor a webböngészőnkbe `https://` kezdetű címet írunk be (például: `https://www.e-szigno.hu`), akkor SSL segítségével kapcsolódunk a webszerverhez, böngészőprogramunk megszerzi és ellenőrzi a webszerver tanúsítványát, és kihívás és válasz alapú azonosítás segítségével ellenőrzi, hogy a webszerver rendelkezik-e a tanúsítványához tartozó magánkulccsal. A legtöbb böngészőprogram egy kis lakatot jelenít meg a jobb alsó sarokban, ezzel jelzi, hogy a webszerver (pl: `www.e-szigno.hu`), amellyel kommunikálunk valóban rendelkezik az adott címre (esetünkben a `www.e-szigno.hu` címre) szóló tanúsítvánnyal, és a kapcsolat a szerverrel biztonságos.

10.1. Mi alapján győződhetünk meg valakinek a kilétéről?

E célra három fő módszer létezik:

- *Tudás alapon* azt vizsgáljuk, hogy partnerünk ismer-e olyan információt, amelyet kizárólag a jogosult fél ismer. Ilyen megoldás például a jelszó alapú beléptetés: minden felhasználónak van egy jelszava, és aki be akar lépni a felhasználó nevében, annak ismernie kell a felhasználó jelszavát. Belépéskor a rendszer megkérdezi a jelszót, és ha a belépő fél meg tudja mondani, akkor beengedi az illetőt az adott felhasználó nevében. Nemcsak jelszót használhatunk tudás alapú beléptetéshez, hanem kriptográfiai kulcsok ismeretét is vizsgálhatjuk.

  10.1. Példa: *Tegyük fel, hogy Bendegúz egy ajtót őriz. Aki be akar lépni, attól megkérdezi a jelszót. Kizárólag azt engedi be, aki a helyes jelszót mondja neki. Bendegúz nem azért kérdezi meg a jelszót, mert kíváncsi volna rá. Ő nem a jelszóra kíváncsi, hanem azt szeretné tudni, a belépő ismeri-e a jelszót. Ennek legegyszerűbb módja, hogy megkérdezi a jelszót, de e megoldásnak*

hátulütői is vannak: például így valaki illetéktelen is meghallhatja a jelszót, amikor a belépő kimondja.

Nyilvános kulcsú kriptográfiai eszközökkel nemcsak az oldható meg, hogy a belépésre jogosító magánkulcsot nem ismeri meg a hallgatózó támadó, hanem ezen a módon Bendegúz úgy tud meggyőződni róla, hogy a belépő ismeri-e a magánkulcsot, hogy azt még a kérdező Bendegúz sem tudja meg. A korábban bemutatott kihívás és válasz alapú azonosítás erre egy megoldás.

A tudás alapú megoldásoknak problémája, ha az információ kiszivárog, vagy a támadó kitalálja azt. Szintén problémát jelent, ha a jogosult felhasználó elfelejti a kérdéses információt. A jelszó vagy PIN kód alapú megoldásoknak gyenge pontja, hogy a felhasználók által még elég könnyen kezelhető jelszavakat számítógéppel sokszor túlságosan is könnyű végigpróbálni. E megoldások ma már leginkább csak akkor életképesek, ha korlátozott, hogy egy felhasználó hányszor próbálkozhat, vagy egy próbálkozás elég sok időt vesz igénybe ahhoz, hogy ne lehessen akárhányszor próbálkozni.

- *Tulajdon alapon* azt vizsgáljuk, hogy partnerünk rendelkezik-e egy olyan tárggyal, amellyel kizárólag a jogosult fél rendelkezik. Ilyen megoldás például ha egy ajtót kulcsra zárunk. Kizárólag a belépésre jogosultaknak van megfelelő kulcsa, így csak ők tudják kinyitni az ajtót. A kulcsokra, kártyákra, tokenekre épülő módszerek tulajdon alapúak.

  A tulajdon alapú megoldásoknak problémája, ha a felhasználó elveszíti a kérdéses tulajdont, vagy ellopják tőle azt. Szintén veszélyt jelent, ha e tulajdon (kulcs, kártya) másolható.

- *Biometriai alapon* azt vizsgáljuk, hogy partnerünk rendelkezik-e olyan élettani jellemzővel, amellyel kizárólag a jogosult fél rendelkezik. Ilyen megoldás például egy ujjlenyomat alapú beléptető rendszer, amely ismeri a jogosult felhasználók ujjlenyomatát, és belépéskor megvizsgálva a belépő felek ujjlenyomatát csak azt engedi be, aki saját belső adatbázisa szerint jogosult erre. Biometriai alapon csak embert lehet azonosítani. Az embernek lényegében bármely része egyedi, a biometriai módszerek közül használnak ujjlenyomatot, hangot, arcfelismerést, írisz vagy retina vizsgálatot stb.

  E megoldások arra épülnek, hogy a rendszer az élő emberből vesz mintát, és azt más módon nem lehet előállítani, és nem lehet visszajátszani. Az emberek biometriai jellemzői általában nem titkosak, az embereket le lehet fényképezni, az ujjlenyomataikat össze lehet gyűjteni stb. Ezen információk alapján a támadók próbálhatnak jeleket visszajátszani, hamisítani. Támadók néha durvább módszerekhez is folyamodnak, például ujjlenyomat alapú beléptetésen néha a jogosult fél levágott ujjával próbálnak átjutni.

A biometriai megoldások sokszor érzékenyek a visszajátszásra, illetve előfordulhat, hogy egyes biometriai jellemzőket elég jól lehet hamisítani, és így a mérő eszközök becsaphatóak. [39] A biometriai rendszerek biztonsága sokszor arra épül, hogy a támadó nem ismeri, hogy a rendszer pontosan hogyan méri, vizsgálja az ember biometriai jellemzőjét, és ha a támadó ennek birtokába jut, esetleg könnyebben be tudja csapni a rendszert. Biometriai rendszerek esetén sokszor csak a gyártók állításaira, nyilatkozataira építhetünk, amelyekről néha kiderül, hogy nem helytállóak. [108] A biometriai rendszerekkel jogi problémák is felmerülhetnek, nem szabad korlátozás nélkül tárolni biometriai jellemzőket, mert azok az adatvédelmi törvény szerint személyes adatnak minősülnek.

Ha erős autentikációra van szükség, a fenti autentikációs módszerek közül többet szokás kombinálni. Akkor beszélünk *kétfaktoros* autentikációról, ha legalább két különböző módon autentikáljuk a felhasználót. Például ha autentikációhoz kártyát kell használni és emellett egy PIN kódot is meg kell adni.

A továbbiakban PKI alapú autentikációs módszerekről szólunk, ekkor a felhasználó a tanúsítványa és magánkulcsa segítségével igazolja kilétét. Az ilyen autentikáció a magánkulcs bizalmasságára épül, így tekinthető tudás alapú autentikációnak. Ha a magánkulcsot intelligens kártya védi, és a kulcs nem nyerhető ki a kártyából, akkor ez tulajdon alapú autentikációnak is tekinthető. PKI alapon könnyen megvalósítható a kétfaktoros autentikáció, például egy intelligens kártyával és annak PIN kódjával.

10.2. Autentikációs tanúsítvány

Autentikációs tanúsítványnak azon X.509 tanúsítványokat nevezzük, amelyeket autentikációra használnak.

Formailag nagyon kevésben különbözik az autentikációs tanúsítvány az aláíró tanúsítványtól, elsősorban annyiban, hogy a tanúsítványt kibocsátó hitelesítés-szolgáltató a tanúsítványban szereplő kulcshasználati (key usage) bitekben jelezheti, hogy a tanúsítványt autentikációra célszerű használni.

Az autentikációs tanúsítványban általában a `digitalSignature` és a `keyAgreement` kulcshasználati bitek szerepelnek. Ezen kívül gyakran megjelennek benne kiterjesztett kulcshasználat (extended key usage) állítások is, SSL szerver tanúsítványokban pl. `serverAuthentication`, SSL kliens tanúsítványokban pl. `clientAuthentication`.

A `digitalSignature` bit jelenléte további magyarázatot igényel. A `digitalSignature` bit *digitális aláírást* jelent, vagyis azt, hogy egy bitsorozatot a magánkulccsal kódolva olyan bitsorozatot hozunk létre, amelyről – a nyilvános kulcs segítségével – igazolható, hogy csak a magánkulcs birtokában lehetett létrehozni. Az elektronikus aláírást, azaz a hosszú távú

elkötelezettséget, a jogilag is kötelező erejű aláírást – amelyről az Eat., illetve az elektronikus aláírásról szóló EU direktíva is szól – a `nonRepudiation`[2] bit jelenti. Autentikáció esetén nagyon rövid ideig használatos, pillanatnyi jelenlétet igazoló *digitális aláírás*ról van szó, valamilyen véletlen kihívást kódolunk a magánkulcsunkkal, így a `digitalSignature` bit meg szokott jelenni az autentikációs tanúsítványban, a `nonRepudiation` bit nem.

Bizonyos értelmezések szerint a pillanatnyi jelenlétet igazoló digitális aláírás és a hosszú távú elkötelezettséget igazoló elektronikus aláírás fogalmai összemosódhatnak, és mivel a két funkcióhoz ugyanaz a padding kapcsolódik, előfordulhat, hogy egyazon kulcspár és egyazon tanúsítvány mindkét funkcióra használható. *A magyar jogszabályok (Eat, 13. § (4)) szerint ez tilos, az aláírásra használt magánkulcsot nem szabad más célra, így például autentikációra használni.*

Bizonyos értelmezések szerint a `digitalSignature` bit jelenti, hogy a tanúsítvány alanya a magánkulcsot digitális aláírásra használja, a *mellette* szereplő `nonRepudiation` pedig azt jelöli, hogy ez egyúttal hosszú távú elkötelezettséget, elektronikus aláírást is jelent(het). Más értelmezések szerint a `nonRepudiation` önmagában jelenti, hogy a tanúsítvány elektronikus aláírásra szolgál, és kizárja a `digitalSignature` jelenlétét. [75]

Az autentikációra használt tanúsítványt és kulcspárt célszerű szétválasztani az aláírásra használt tanúsítványtól és kulcspártól, mert:

- Az Eat. 13. § (4) tiltja, hogy az aláíró az aláírásra használt magánkulcsot aláíráson kívül bármi másra használhassa.

- Autentikációkor véletlen kihívást kódolunk a magánkulcsunkkal, ha ugyanazt a kulcsot egyúttal aláírásra is használjuk, egy támadó a véletlen kihívás helyett akár értelmes dokumentumot is aláírathat velünk .

  10.2. Példa: *Alajos be szeretne lépni a Szevér által üzemeltetett szerverre. Jelzi a szervernek, hogy be szeretne lépni, mire a szerver küld neki egy r véletlen számot, mint kihívást. Alajos magától nem írná alá a d dokumentumot, ezért Manfréd, a támadó ki akar csalni Alajostól egy aláírást a d dokumentumra. Manfréd elkapja a hálózaton a Szevértől jövő kihívást, és helyette a d dokumentum $h(d)$ lenyomatát teszi az Alajosnak címzett csomagba. Alajos aláírja a Szevértől kapott csomagban szereplő véletlen számot, és visszaküldi Szevérnek. Manfréd ezt a csomagot is elkapja, és így hozzájutott Alajos aláírásához a d dokumentumra. Szevér nem kapja meg a megfelelő választ a kihívására, így nem engedi be Alajost a szerverre. Alajos nem érti, mi történt, de nem gyanakszik rá, hogy most éppen aláírt valamit.*

[2] újabb nevén `contentCommitment`

Az autentikációra használt tanúsítványt és kulcspárt célszerű szétválasztani a titkosításhoz használt tanúsítványtól és kulcspártól, mert:

- A titkosító tanúsítványnak egészen más az életciklusa. Például titkosító tanúsítvány esetén létezhet kulcsletét szolgáltatás, azaz előfordulhat, hogy a tanúsítványhoz tartozó magánkulcs nem kizárólag a tanúsítvány alanyának a birtokában van. (Lásd: 3.3.3. fejezet.)

- Autentikációkor véletlen kihívást kódolunk a magánkulcsunkkal. Ha ugyanazt a kulcsot egyúttal dekódolásra is használjuk, akkor egy támadó a véletlen kihívás megválaszolása helyett egy nekünk szóló, titkosított dokumentumot is kicsomagoltathat velünk.

> ***10.3. Példa:*** *Alajos be szeretne lépni a Szevér által üzemeltetett szerverre. Jelzi a szervernek, hogy be szeretne lépni, mire a szerver küld neki egy r véletlen számot, mint kihívást. Manfréd elfogott egy Alajosnak címzett, $E_A(m)$ titkosított üzenetet, de ezt csak Alajos magánkulcsával lehet kinyitni. Ezért Manfréd elkapja a hálózaton a Szevértől jövő kihívást, és helyette az $E_A(m)$ üzenetet teszi az Alajosnak címzett csomagba. Alajos digitálisan aláírja a Szevértől kapott csomagban szereplő véletlen számot, és visszaküldi Szevérnek. Manfréd ezt a csomagot is elkapja, és így hozzájutott az Alajosnak szóló m üzenethez. Szevér nem kapja meg a megfelelő választ a kihívására, így nem engedi be Alajost a szerverre, Alajos nem érti, mi történt, de fel sem merül benne, hogy épp most dekódolt valamit Manfréd számára.*

Lényeges, hogy *az elektronikus aláírásról szóló törvény nem vonatkozik autentikációs tanúsítványokra.* Így, ha egy hitelesítés-szolgáltató autentikációs tanúsítványt (pl. webszerver tanúsítványt) bocsát ki, nem vonatkozik rá az Eat. Ekkor kizárólag a szolgáltató szolgáltatási szabályzatai és a hitelesítési rend alapján dönthetjük el, hogy mit várhatunk el a tanúsítványtól.

Ennek ellenére *sok szolgáltató nagyon hasonlóan kezeli az autentikációs (és titkosító) tanúsítványokat, mint az aláíró tanúsítványokat.*

Akármilyen követelmények is vonatkoznak az aláírásra, titkosításra és autentikációra használható kulcspárokra, akármi is szerepel a tanúsítványban, egy tanúsítványt ellenőrző érintett fél olyan tanúsítványt fogad el, amilyet csak kedve tartja. Így egy érintett fél akár elfogadhat aláírói tanúsítványt is autentikációra. Bonyolítja a helyzetet, hogy sok böngészőprogram felkínál autentikációra minden olyan tanúsítványt, amelyben `digitalSignature` kulcshasználat szerepel, így véletlenül autentikálhatunk egy fokozott biztonságú[3] aláírás létrehozására alkalmas, `digitalSignature` és `nonRepudiation` bitet

[3]Magyarországon a minősített tanúsítványokban kizárólag `nonRepudiation` szereplhet.

egyaránt tartalmazó tanúsítvány magánkulcsával. Az ilyen tanúsítványt a webszerverek is el szokták fogadni.

10.3. Biztonságos csatorna

10.3.1. Autentikációt követően biztonságos csatorna is létrejöhet

Az autentikáció önmagában annyit jelent, hogy meggyőződünk róla, valóban a kívánt féllel vagyunk-e kapcsolatban. Ha személyesen találkozunk valakivel, az igazolványa alapján meggyőződhetünk az illető kilétéről. Ha az illető ezt követően mond nekünk valamit, pontosan tudjuk, hogy azt ki mondta.

Ha távoli partnert hálózaton keresztül autentikálunk, általában nem elegendő meggyőződni az illető kilétéről, hanem azt is tudni szeretnénk, hogy a következő üzenetek is ugyanezen féltől érkeznek. Az autentikációt követően ezért általában ún. *biztonságos csatorna* is kiépül.

Autentikációkor a felek nyilvános kulcsú kriptográfiai módszerek alapján meggyőződnek egymás kilétéről, és egyúttal megállapodnak egy *közös titkokban* is, amelyet kizárólag az egymást autentikáló felek ismernek, és amelynek a csatornán hallgatózó támadó sem juthatott birtokába. Erre például a Diffie-Hellman protokoll nyújt megoldást, amelynek elliptikus görbék feletti változatát korábban (2.6.2.5. fejezet) bemutattuk.

A felek e közös titokból szimmetrikus kulcsokat származtatnak, és az ezt követő kommunikációt ezen szimmetrikus kulcsok segítségével titkosítják, illetve hitelesítik. A csatorna titkosított, így a csatornán hallgatózó támadó nem ismeri meg, hogy ki milyen üzenetet küldött. (A csatorna típusától függően a támadó megismerheti, hogy ki kivel építette ki a csatornát.) A csatorna hitelesített, így a csatornát manipuláló aktív támadó nem tudja észrevétlenül módosítani az üzeneteket. Ha az A felhasználó biztonságos csatornát épített ki a B felhasználóval, és ezen a csatornán a B felhasználó az x üzenetet kapja, akkor biztos lehet benne, hogy az x üzenetet valóban az A felhasználó küldte, és senki nem módosította útközben.

Ugyanakkor *a biztonságos csatornán küldött üzenet csak addig titkos és csak addig hiteles, amíg a csatornán áthalad.* Az előbb említett B felhasználó hiába biztos benne, hogy az x üzenetet az A felhasználó küldte, ezt már nem tudja bebizonyítani a C felhasználónak. A csatornából kivett üzenetről nem állapítható meg, hogy az korábban a csatornán érkezett. Ehhez hasonlóan, ha az A felhasználó elmenti számítógépén az x üzenetet, annak bizalmasságát már nem védi, hogy az eredetileg biztonságos csatornán érkezett.

10.3.2. Secure Socket Layer (SSL, TLS)

A Secure Socket Layer (SSL), újabb elnevezéssel Transport Layer Security biztonságos csatorna kiépítésére szolgáló kriptográfiai protokoll. Az SSL-t a Netscape fejlesztette ki, majd

TLS néven internetes RFC-ként jelent meg. Ennek legfrissebb változata az RFC 5246. [151]

Többféle módon használható az SSL. Legtöbbször csak a szerver igazolja kilétét tanúsítvány alapon (PKI alapon), és a kliens anonim módon építi fel a kapcsolatot. (Ez nem zárja ki, hogy később a már felépített SSL kapcsolaton belül felhasználónév és jelszó alapján azonosítsa magát.) Másik lehetőség az ún. kétoldali autentikáció, amikor mindkét fél, mind a kliens, mind a szerver tanúsítvány alapon igazolja kilétét.

Az SSL csatorna felépítése egy „handshake" művelettel kezdődik, melynek során egy *kliens* és egy *szerver* meggyőződnek egymás kilétéről, majd közös algoritmuskészletekben és közös szimmetrikus kulcsokban állapodnak meg. Az SSL kapcsolat felépítése úgy kezdődik, hogy a kliens kapcsolatba lép a szerverrel, és elküldi neki az általa támogatott kriptográfiai algoritmuskészletek (rejtjelezők és hash függvények) listáját. A szerver kiválasztja a listából az általa is támogatott, legerősebb algoritmusokat, és értesíti erről a klienst. A szerver elküldi a kliensnek saját szerver tanúsítványát, amely egy autentikációs tanúsítvány (jellemzően `serverAuthentication` kiterjesztett kulcshasználattal). Ha a kliens meggyőződött a szerver tanúsítványának érvényességéről, a kliens egy véletlen számot titkosít a szerver tanúsítványában lévő nyilvános kulccsal, és e titkosított véletlen számot küldi el a szervernek. Az így kapott titkosított véletlen számot csak a szerver tudja dekódolni a saját magánkulcsa segítségével. E véletlen számból a felek szimmetrikus kulcsokat származtatnak le, amelyek segítségével titkosítják és hitelesítik a későbbi üzeneteket. A handshake ezzel véget ér.

Az kliens (C) és a szerver (S) közötti SSL handshake a következő módon zajlik le:

`ClientHello` $(C \to S)$: A kliens elküldi a szervernek az általa támogatott legfrissebb TLS verziószámát, egy friss véletlen számot és az általa javasolt kriptográfiai algoritmuskészletek (rejtjelezők és hash függvények) és tömörítő algoritmusok listáját.

`ServerHello` $(S \to C)$: A szerver válasza tartalmazza a kiválasztott TLS protokoll verziót, egy friss véletlen számot és a kiválasztott algoritmuskészletet és tömörítő algoritmust.

`Certificate` $(S \to C)$: A szerver elküldi a tanúsítványát is.

Kétoldali autentikáció esetén:

`CertificateRequest` $(S \to C)$: A szerver a klienstől is tanúsítványt kér.

`ServerHelloDone` $(S \to C)$: A szerver jelzi, hogy részéről véget ért az egyeztetés.

Kétoldali autentikáció esetén:

`Certificate` $(C \to S)$: A kliens is elküldi a tanúsítványát a szervernek.

A szerver ellenőrzi a kliens tanúsítványának érvényességét (felépíti a tanúsítványláncot, és ellenőrzi a láncban szereplő tanúsítványok visszavonási állapotát), és megvizsgálja, hogy az adott tanúsítvánnyal rendelkező kliens jogosult-e az adott kapcsolat létrehozására. (Lásd: 10.6. fejezet.)

`ClientKeyExchange` $(C \to S)$: A kliens egy PreMasterSecret üzenetet küld a szervernek, amely egy véletlen számot tartalmaz a szerver nyilvános kulcsával titkosítva.

Az e lépésben titkosítva átküldött véletlen számból származtatja majd mindkét fél a csatorna titkosításához és hitelesítéséhez szükséges szimmetrikus kulcsokat.

Kétoldali autentikáció esetén:

`CertificateVerify` $(C \to S)$: A kliens lenyomatot képez az összes eddigi handshake üzenetről, e lenyomatot aláírja[4] magánkulcsával, és az így kapott blokkot elküldi a szervernek. A szerver ebből tudja megállapítani, hogy a kliens valóban birtokolja a kliens által korábban bemutatott tanúsítványhoz tartozó magánkulcsot.

`ChangeCipherSpec` $(C \to S)$: A kliens jelzi, hogy mostantól minden kommunikációt az előzőekben egyeztetett algoritmuskészletekkel és szimmetrikus kulccsal véd.

`Finished` $(C \to S)$: A kliens elküldi az első titkosított és hitelesített üzenetét, amely egy kriptográfiai ellenőrzőösszeget tartalmaz az összes előző handshake üzenetre.

`ChangeCipherSpec` $(S \to C)$: A szerver is jelzi, hogy mostantól minden kommunikációt az előzőekben egyeztetett algoritmuskészletekkel és szimmetrikus kulccsal véd.

`Finished` $(S \to C)$: A szerver is elküldi az első titkosított és hitelesített üzenetét, amely egy kriptográfiai ellenőrzőösszeget tartalmaz az összes előző handshake üzenetre.

A handshake véget ért, felépült a biztonságos csatorna.

Megjegyzés: A protokoll nem szimmetrikus, a kliens a szerver tanúsítványa szerint titkosított blokkot küld (a szerver úgy igazolja kilétét, hogy ki tudja bontani ezt a blokkot, és válaszolni tud a belőle származtatott kulcsokkal), majd szintén a kliens írja alá a handshake üzeneteket (és a szerver ezen digitális aláírásból állapítja meg, hogy a kliens birtokolja a szükséges magánkulcsot).

Ha a felek egy már felépített kapcsolat esetén módosítani szeretnének a kapcsolat jellemzőin – például egy anonim SSL kapcsolatban a kliens olyan erőforráshoz kíván hozzáférni, amelyhez a szerver a kliens tanúsítványát kéri – ismételt handshake-et futtathatnak. Ezt a műveletet nevezik „renegotiation”-nek.

Lényegében bármilyen protokoll (HTTP, FTP) futtatható SSL-en keresztül, de az SSL (TLS) talán legnagyobb, legismertebb felhasználási területét a webszerverekkel való biztonságos kapcsolat jelenti.

[4] Ez csak *digitális aláírás*, elektronikus aláírás, jogi értelemben nem jelent elkötelezettséget.

10.4. Autentikáció a weben

10.4.1. Webszerver tanúsítványok

Az autentikációs tanúsítványok közül a webszerver tanúsítványok a leggyakoribbak. Ha egy webszervernek tanúsítványa van, akkor a szerveren lévő weboldalakat látogató felhasználóknak lehetősége van biztonságos – titkosított és hitelesített – kapcsolatot létesíteni a szerverrel, meggyőződhetnek róla, hogy valóban azzal a szerverrel kommunikálnak, amelyikkel szeretnének, és biztosak lehetnek benne, hogy a kommunikációt más nem hallgathatja le. Ma általánosan elfogadottnak számít, hogy bizonyos webszervereknek tanúsítványa van, és ezt a weboldalak felhasználói gyakran el is várják.

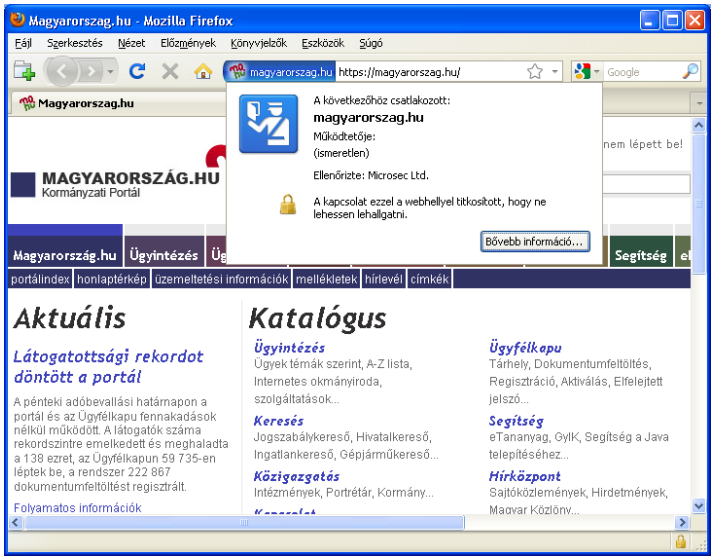
Ha egy böngészőbe `https://` kezdetű címet írunk be, akkor a böngésző megpróbál SSL (secure socket layer) kapcsolatot létesíteni az adott szerverrel. Az SSL kapcsolat keretében elkéri és ellenőrzi a szerver tanúsítványát, összehasonlítja a tanúsítványban szereplő és a böngészőbe beírt címet (URL-t), majd a tanúsítványban lévő nyilvános kulcsot felhasználva közös szimmetrikus kulcsokban állapodnak meg. A későbbi kommunikációt ezen szimmetrikus kulcsokkal titkosítják, illetve hitelesítik.

Ha SSL kapcsolatot létesítettünk, a böngészők általában egy kis lakatot[5] jelenítenek meg. (Lásd: 10.1. ábra.) A lakat vagy az URL (a weboldal címe) mellett, vagy a böngésző jobb alsó sarkában jelenik meg. (Ha a lakat magán a weboldalon jelenik meg, az nem jelenti, hogy az oldallal valóban biztonságos lenne a kapcsolat. Csaló weboldalak szoktak ilyen, és ehhez hasonló trükkökkel próbálkozni.)

Mikor fontos, hogy egy oldalnak SSL tanúsítványa legyen? Elsősorban akkor, ha az oldallal érzékeny információkat cserélünk. Például internet bank estében különösen fontos, hogy SSL kapcsolaton keresztül jelentkezzünk be, mert ekkor a hálózaton hallgatózó támadók sem a belépési jelszavunkat, sem a banki adatainkat nem hallgathatják le. Hasonlóan érzékeny a helyzet, ha Interneten fizetünk, és hitelkártyánk adatait adjuk meg a weben keresztül. Akkor is fontos lehet az SSL kapcsolat, ha nem adunk ugyan meg személyes adatokat, viszont valami miatt *különösen* fontos, hogy a weboldalon szereplő információk valóban arról az oldalról származnak, mint ahogy hisszük. (Gyakori, hogy a szerverrel nem cserélünk érzékeny adatokat, ekkor teljesen felesleges SSL kapcsolatot létesíteni, a kódolási műveletekkel ekkor csak feleslegesen terhelnénk a gépünket. Ha nem cserélünk érzékeny adatokat, az SSL hiánya nem jelent veszélyt.)

A támadók számára nem nehéz eltéríteni a kapcsolatainkat, ekkor hiába írjuk be a böngészőnkbe, hogy a `http://kedvencoldalam.hu` oldallal szeretnénk kommunikálni, a támadó egészen egyszerűen átirányíthat bennünket az ő csaló, `http://sunyivagyok.hu` című

[5] A lakatot nagyon sok felhasználó ismeri, keresi, és használja. Ugyanakkor a Firefox a 4-es verziójától a Mozilla megszüntette a lakatot, mert – álláspontjuk szerint – sok felhasználó számára a lakat azt jelentette, hogy ha „lakatos” oldalt látnak, biztonságban vannak; így úgy vélték, a lakat hamis biztonságérzetet kelt.



10.1. ábra. **A képen látható böngésző a jobb alsó sarokban megjelenő lakat ikonnal és az URL színezésével jelzi, hogy biztonságos kapcsolatot építettünk ki a webszerverrel. Az URL elején kékre színezve külön megjeleníti a weboldal címét a szerver tanúsítványa alapján. Ha e címre rákattintunk, részleteket is megtudhatunk a tanúsítványról.**

weboldalára.

Ha ugyanazt a címet `https://` kezdettel írjuk be (`https://kedvencoldalam.hu`), akkor a támadó hiába téríti el a kapcsolatot a `sunyivagyok.hu` oldalra. Böngészőnk ekkor ugyanis elkéri az oldal tanúsítványát, megvizsgálja, hogy a tanúsítvány a `kedvencoldalam.hu` címre szól-e, majd az SSL protokoll szerint – kihívás és válasz alapú megoldás segítségével – ellenőrzi, hogy a webszerver rendelkezik-e az adott – `kedvencoldalam.hu` címre szóló – tanúsítványhoz tartozó magánkulccsal. A hitelesítés-szolgáltatók a tanúsítvány kibocsátása előtt meg kell, hogy győződjenek róla, hogy a tanúsítvány alanya – esetünkben a webszerver üzemeltetője – birtokolja-e a tanúsítványba kerülő címet (domaint), valamint rendelkezik-e a tanúsítványhoz tartozó magánkulccsal, és a kulcs nem volt-e illetéktelen kezekben. Ez a megoldás biztosítja, hogy ha a böngészőnkbe a `https://kedvencoldalam.hu` címet írjuk be, akkor a kapcsolat a `kedvencoldalam.hu`-val jön létre, és – tekintve, hogy a kapcsolat titkosított és hitelesített – a kommunikációt illetéktelen fél nem módosíthatja, és nem hallgathatja le.

A webszerver tanúsítvány azt igazolja, hogy a tanúsítványhoz tartozó magánkulccsal kizárólag a tanúsítványban lévő cím birtokosa rendelkezik. Így a kérdéses webszerverrel titkosított és hitelesített kapcsolatot lehet létesíteni.

A következőkre legyünk tekintettel ezzel kapcsolatban:

- A webszerver tanúsítvány nem jelenti azt, hogy a szerver biztonságos. Ha a szervert feltörték, és az a támadó irányítása alatt áll, hiába létesítünk vele biztonságos

kapcsolatot, a támadóval kommunikálunk.

- Az SSL kapcsolatot a tanúsítványban – és a böngészőnk címsorában – megjelölt szerverrel létesítjük. A címből nem mindig könnyű kideríteni, hogy valóban azzal kommunikálunk, akivel szeretnénk. Az `xyz.hu` címből gyakran nehéz meghatározni, hogy a cím mögött milyen cég vagy szervezet áll.

- Ha valakinek érvényes tanúsítványa van, az nem jelenti azt, hogy az illető jóindulatú. A támadó, mint a `sunyivagyok.hu` domain birtokosa, nyugodtan vásárolhat erre a címre tanúsítványt. Hiába győződünk meg róla, hogy az oldalnak tanúsítványa van, nem szerencsés, ha hitelkártyánk adatait közvetlenül a támadónak adjuk meg. (Sőt, a támadó, mint a `sunyivagyok.hu` domain birtokosa, vásárolhat tanúsítványt a `kedvencoldalam.sunyivagyok.hu` címre is.)

  Célszerű megtekinteni az oldal tanúsítványát, és megnézni a tanúsítványban szereplő adatokat, ez alapján – például a tanúsítvány alanya szervezetének nevéből – könnyebb eldönteni, hogy valóban azzal kommunikálunk, akivel szeretnénk.

- Gyakori, hogy amikor az X cégtől szeretnénk vásárolni valamit, az átirányít az Y cég weboldalára, mert a fizetéssel kapcsolatos lépéseket az Y cég bonyolítja le. Ilyenkor általában nem akarnak becsapni bennünket, de akár az URL-t, akár a tanúsítványban lévő adatokat próbáljuk ellenőrizni, zavarba jöhetünk, amikor egy vadidegen cég címét és adatait találjuk ott.

- Legyünk óvatosak, ha a böngészőnk nem ismeri fel, vagy nem tudja ellenőrizni egy oldal tanúsítványát. Ez önmagában nem jelenti azt, hogy csaló oldalról van szó, de – ha más módon nem győződünk meg a tanúsítvány érvényességéről, – nem lehetünk biztosak benne, hogy valóban azzal kommunikálunk, akivel szeretnénk.

- A böngészők időnként elemi lépéseket kihagynak a tanúsítványok ellenőrzésekor. Például előfordul, hogy nem ellenőrzik a tanúsítvány visszavonási állapotát stb.

- Ha SSL kapcsolaton kommunikálunk egy szerverrel, a kapcsolatba harmadik fél nem tud közbeékelődni (nem tudja azt sem lehallgatni, sem módosítani), de ez nem mindig előnyös. Például előfordulhat, hogy egy rosszindulatú weboldallal (pl. a támadó irányítása alatt álló, feltört oldallal) kommunikálunk, és az oldalról SSL kapcsolaton keresztül rosszindulatú programot (pl. vírust) töltünk le. Ekkor hiába védi a hálózatunkat olyan tűzfal, amely esetleg képes volna kiszűrni a rosszindulatú tartalmat, a saját tűzfalunk sem lát bele a titkosított adatfolyamba.

10.4.2. Támadások webszerver tanúsítványok ellen

Ha az SSL kapcsolat erős kriptográfiai algoritmusok segítségével épül fel a megfelelő felek kulcspárjai alapján, akkor a felek magánkulcsait nem ismerő támadónak a tudomány és a technológia mai állása mellett nincs számottevő esélye a kapcsolatot lehallgatni, vagy a kapcsolaton küldött információkat észrevétlenül módosítani.

Ennek ellenére, ahogy az internetes kereskedelem virágzik, hasonlóan virágzó ágazattá vált az internetes csalás is. Annak ellenére, hogy a kriptográfiai algoritmusokat a támadók nem tudják ésszerűen támadni, a lehetséges nyereség által motiválva igenis megkeresik a rendszer azon pontjait, ahol mégis sikerrel járhatnak. Nem a kriptográfiai algoritmust támadják, mert azt – reális erőforrásokkal – nem lehet. Ehelyett támadják a felhasználót, támadják az alkalmazásokat, és támadják a hitelesítés-szolgáltatót.

10.4.2.1. A felhasználók megtévesztése

A felhasználók a webszerverek SSL tanúsítványa alapján győződhetnek meg arról, hogy biztonságos kapcsolaton kommunikálnak a szerverrel (pl. a bankjukkal). A böngésző programok egy apró lakatot jelenítenek meg az állapotsorban, így jelzik, hogy a felhasználó SSL kapcsolaton keresztül néz egy oldalt. Ez a lakat mindössze annyit jelent, hogy *valakivel* SSL kapcsolaton keresztül (titkosított és hitelesített csatornán) kommunikálunk. Ahhoz, hogy a kommunikációt valóban biztonságosnak nevezhessük, azt is meg kell vizsgálni, hogy kicsoda az, akivel a biztonságos csatornát kiépítettük. Nem sok értelme van az SSL kapcsolatnak, ha nem tudjuk, ki az, akivel titkosítva és hitelesítve kommunikálunk.

Az erre az egyik lehetőség, hogy ellenőrizzük a böngészőben annak oldalnak a címét, amelyikkel kapcsolatba léptünk. El kell döntenünk, hogy valóban annak az oldalnak a címe szerepel-e (HTTPS-sel) a böngésző címsorában, mint amelyiken lenni szeretnénk. A másik, talán biztonságosabb megoldást az jelenti, ha rákattintunk a lakatra, és megnézzük a webszerver tanúsítványát is. A tanúsítványból kiderül, hogy a tanúsítványt pontosan kinek vagy minek a számára bocsátották ki. (Itt fontos megjegyezni, hogy csak igazi, megbízható hitelesítés-szolgáltató által kibocsátott tanúsítványban bízhatunk igazán. Bármelyik csaló bármikor kibocsáthat saját maga számára tanúsítványt. Ha figyelmen kívül hagyjuk a böngésző program azon figyelmeztetését, hogy a tanúsítványt nem megbízható szolgáltató bocsátotta ki, akkor lehet, hogy az SSL nyújtotta védelem semmit sem ér.) Mindkét esetben az a probléma, hogy nem könnyű eldönteni, hogy a böngészőben lévő cím vagy a tanúsítványban szereplő megnevezés valóban azt a bankot vagy szervezetet jelenti-e, amelyiknek az oldalán lenni szeretnénk. Például sok phising üzenet `www.xyzbank.net` címet tartalmaz, holott a bank valódi címe pedig `www.xyzbank.hu` volt. Onnantól kezdve, hogy a támadó valóban megszerezte a `www.xyzbank.net` domaint, nem kerül neki komoly erőfeszítésbe webszerver tanúsítványt szerezni hozzá. Az a tapasztalat, hogy – habár a böngészőkben vannak biztonsági megoldások

– a laikus felhasználók nem ismerik, és így nem tudják kihasználni őket.

A felhasználók egy része nincs tudatában a veszélynek, és nem is számít a támadásra. A felhasználók egy másik része tisztában van a veszéllyel, és még azt is tudja, hogy lakatot kell keresni, de ennek ellenére megtéveszthető. Egy harvardi tanulmány ezen megtévesztési módszereket elemzi. [190] Eszerint van olyan felhasználó...

- akit meggyőz, ha egy banki oldal pont úgy néz ki, mint a bank igazi oldala.
- aki figyelmen kívül hagyja a böngésző figyelmeztetését, miszerint a weboldal tanúsítványa érvénytelen,
- aki azt is elfogadja, ha a weboldalon szerepel a lakat (nem pedig a böngésző interfészén), és fel sem merül benne, hogy azt a támadó is odatehette.
- aki keresi és felismeri a lakatot, de nem nézi meg, hogy milyen oldalon jár.
- aki a weboldal címét is ellenőrzi, de nem nézi meg, hogy milyen névre (DN) szól a tanúsítvány.
- aki a tanúsítvány tartalmát is ellenőrzi, de ennek ellenére megtéveszthető.

A tanulmány azt a következtetést vonja le, hogy a felhasználók megtéveszthetősége teljesen független a koruktól, nemüktől, iskolai végzettségüktől, és még attól is, hogy naponta mennyit használnak számítógépet. Az egyik csaló weboldal a felmérésben részt vevők mindegyikét sikeresen megtévesztette. A `www.bankofthevvest.com` oldal érvényes tanúsítvánnyal rendelkezett, tartalma megegyezett a helyes (`www.bankofthewest.com`) oldallal, és látszatra még a címe is hasonlított[6].

Egy másik támadás arra épít, hogy már nemcsak az angol ABC betűi szerepelhetnek a domain nevekben. Így a támadó olyan domain névre szerezhet – érvényes – tanúsítványt, amelynek írásmódja „hasonlít” a megtámadott domainhez. Például a `www.paypal.com` cím helyett a `www.paypäl.com` címre szerez tanúsítványt, vagy olyan más címre, amelynek valamely karaktere nagyon hasonlít az paypal egyik betűjéhez. Csak a rendkívül éles szemű felhasználónak tűnik fel az ilyen különbség.

> **Megjegyzés:** Az igazán paranoiás Internet felhasználó csak az olyan címekben bízhat meg, amelyeket ő maga gépelt be.

A támadás ellen a hitelesítés-szolgáltatók védekezhetnek azzal, hogy ellenőrzik, hogy az igényelt tanúsítványban csak olyan karakterek szerepelnek-e, amelyek a megadott top-level-domain alatt érvényesek. Például a magyar ABC-ben nincsen ł betű, így nem illik olyan `.hu` végű címre bocsátani ki tanúsítványt, amelyben ł karakter szerepel (pl. `xyzhivatał.hu`).

[6] A csaló címben duplavé helyett két darab szimpla „v” betű szerepelt.

Szintén unicode karakterekre épül a következő, igen galád támadás. [105] A támadó igényel egy tanúsítványt a saját, `*.tamadovagyok.cn` pl. kínai domainjére. (A `*` jel azt jelenti, ún. wildcard tanúsítványról van szó, amely az adott domain minden első szintű subdomainjére illeszkedik (10.4.4. fejezet).) Ezt követően, olyan, különösen hosszú subdomainre irányítja a felhasználót, mint pl. a

`www¸xyzbank¸hu/services/beleptetes/kutykurutty/.../login¸jsp.tamadovagyok.cn`

A címben a subdomain eleje hasonlít a megszemélyesíteni kívánt, `www.xyzbank.hu` címhez, és annak valódi, beléptető oldalához, csak „`.`” és „`/`” helyett azokhoz – a böngészőprogramok címsoraiban – nagyon hasonlónak látszó karakter jelenik meg. A teljes cím nem fér ki a böngésző címsorában, így a felhasználó nem látja a végén megbúvó, támadásra utaló domaint.

10.4.2.2. A böngészők hibáit kihasználva

Egyes böngészők több, mint nagyvonalúan ellenőrzik a webszerverek tanúsítványait. Előfordulhat, hogy nem ellenőrzik a webszerver tanúsítvány visszavonási állapotát, a tanúsítványláncban szereplő valamely szolgáltatói tanúsítvány visszavonási állapotát, vagy hibásan vezetik vissza a tanúsítványt a megbízható gyökérre.

Például előfordult, hogy egyes böngészők nem vizsgálták a szolgáltatói tanúsítványokban szereplő `basicConstraints` (3.4. fejezet) kiterjesztést, így olyan webszerver tanúsítványt is elfogadtak, amelyet nem hitelesítés-szolgáltató, hanem egy végfelhasználó írt alá (5.5.3.4. fejezet). [104]

10.4.2.3. A felhasználó számítógépét manipulálva

Ha a böngésző helyesen ellenőrzi a webszerver tanúsítványt, akkor is előfordulhat, hogy a felhasználó számítógépére kerül egy vírus, amely átveszi az irányítást. Az ún. man-in-the-browser támadás úgy működik, hogy amikor a felhasználó a böngészőjével valamilyen módon biztonságos csatornát alakít ki egy szerverrel, akkor a felhasználó gépére települt rosszindulatú program (a „brazil trójai”[7]) a felhasználó böngészőjéből a felhasználó nevében küldi el a támadó által meghatározott parancsokat.

> ***10.4. Példa:*** *Alajos bejelentkezik a webbankjába. Bejelentkezéskor a kapcsolat a bank webszerver tanúsítványa alapján épül fel, majd a bank SMS-ben elküld Alajosnak egy titkos kódot. Alajos beírja a weboldalra a titkos kódot, és megadja a saját jelszavát is. Az Alajos gépére települt rosszindulatú program felismeri, hogy Alajos bejelentkezett a webbankba, és Alajos böngészőprogramja segítségével – a*

[7] A szerző korábban nem hitte volna el, hogy ilyen szókapcsolat valaha is létrejöhet. Akkor neveznek egy rosszindulatú programot trójainak, ha az jóindulatú programnak álcázza magát, és úgy éri el, hogy a felhasználó önként futtassa le a számítógépén. Man-in-the-browser módon működő trójai programokat brazíliai bankok ellen használtak egyes nevezetes esetekben, innen a furcsa elnevezés.

biztonságos kapcsolatot meglovagolva – átutalást kezdeményez Manfréd, a támadó számlájára.

Az ilyen támadások ellen alapvetően semmilyen autentikációs (partner hitelesítési) módszer nem véd. Megoldást jelent viszont, ha nemcsak a partner, hanem a tranzakció hitelességéről is gondoskodunk. Például nemcsak a jelszót, hanem a tranzakció lényeges elemeit is megerősítjük SMS-ben.

Kevésbé körmönfont támadás, ha a felhasználó számítógépére került rosszindulatú program úgy módosítja a felhasználó böngészőjét, hogy az a csaló weboldalt is érvényes oldalnak fogadja el (és ehhez például új gyökértanúsítványt telepít a felhasználó számítógépére).

10.4.2.4. A hitelesítés-szolgáltatók megtámadása

Legegyszerűbb a hitelesítés-szolgáltatók regisztrációs eljárását támadni, és így szerezni „valódi" csaló tanúsítványt. Ez nehéz feladat, de előfordul, hogy például a szolgáltató hanyagsága miatt mégis kivitelezhető. Nagy port kavart fel, amikor a közelmúltban egy Mozilla-aktivista vásárolt egy webszerver tanúsítványt a `mozilla.com` domainre egy nagy nemzetközi hitelesítés-szolgáltató egy viszonteladójától. A tanúsítványt nem a domain tulajdonosa vásárolta, és az illető aktivista egyáltalán nem volt jogosult a Mozilla címére tanúsítványt igényelni. Állítása szerint semmilyen ellenőrzés nem történt a tanúsítvány kibocsátása során, a szolgáltató viszonteladója kérdés nélkül kibocsátotta a számára az SSL tanúsítványt. Mindebből azt vonta le, hogy az adott viszonteladó általában így szokott eljárni, és így akár bárki bármilyen domainre igényelhetne SSL tanúsítványt. [125]

Ritka, hogy a támadás kriptográfiai alapon történik, de erre is volt már példa, ilyen például a „rogue CA" néven elhíresült eset. [173] Egy kutatócsoport – akik támadásukkal egy biztonsági hibára hívták fel a figyelmet – kerestek egy olyan CA-t, amelyiknek a gyökér tanúsítványa széles körben elterjedt, sok böngésző elfogadja, és még mindig bocsátott ki olyan tanúsítványokat, amelyek az akkor már elavult, MD5 lenyomatképző algoritmusra (2.4. fejezet) épülő aláírásokat tartalmaznak. Generáltak egy kulcspárt, és PKCS#10 tanúsítványkérelem (4.5.2. fejezet) alapján igényeltek és kaptak a CA-tól egy legális végfelhasználói tanúsítványt. Az MD5 hibája miatt speciális PKCS#10-et tudtak konstruálni: A visszakapott legális végfelhasználói tanúsítványon lévő aláírás egyúttal egy másik, csaló tanúsítványhoz is helyes aláírás lett. Így létre tudtak hozni egy csaló CA tanúsítványt („MD5 Collisions Inc." néven) egy saját kulcspárhoz. Innentől, amit e kulcspár magánkulcsával aláírnak, azt a böngészők mind-mind elfogadták...

Léteznek olyan támadások, amelyeket állami szervek hajthatnak végre. Például az SSL kapcsolatot lehallgatni kívánó ország kötelezhet egy – az adott országban működő – hitelesítés-szolgáltatót (bármelyik hitelesítés-szolgáltatót azok közül, amelyekben a célpont felhasználó vagy az ő alkalmazása megbízik) egy csaló tanúsítvány kibocsátására. Amikor a

felhasználó az X weboldallal szeretne https kapcsolatot kiépíteni, a kapcsolatot lehallgatni kívánó ország kormányzata, titkosszolgálata „szerez” magának egy csaló tanúsítványt az X weboldal címére, megszemélyesíti az X weboldalt, majd a felhasználó kéréseit továbbítja a weboldalnak, a weboldal válaszait pedig visszaküldi a felhasználónak (azaz ún. man-in-the-middle támadást hajt végre). Egy kormányzat esetleg kötelezhet egy hitelesítés-szolgáltatót egy ilyen tanúsítvány kibocsátására, bár ha egy hitelesítés-szolgáltató részt vesz egy ilyen műveletben, azzal az egzisztenciáját kockáztatja, végképp elveszhet benne a bizalom. Rengeteg hitelesítés-szolgáltató létezik, és sok alkalmazás (pl. Internet Explorer, Mozilla) nagyon-nagyon sok szolgáltató gyökértanúsítványát elfogadja. Elég egyetlen együttműködő szolgáltatót találni, az alkalmazás összes felhasználója támadhatóvá válik. A cikk egyik fő érve arra épül, hogy létezik olyan, kormányzatok számára hirdetett berendezés[8], amely a fenti támadás kivitelezésére szolgál.

E támadás technikailag természetesen kivitelezhető. Már régóta léteznek megoldások, amelyek ehhez hasonló módon teszik lehetővé, hogy egy szervezet/vállalat szűrhesse a tűzfalrendszerén átmenő titkosított forgalmat, vagy monitorozza a titkosított csatornán történő rendszergazdai belépéseket (pl. SSH). Sok szervezet használ ilyen technológiát, például azért, hogy a felhasználóik ne töltsenek le vírusokat SSL-en keresztül. Ez egy teljesen korrekt, „fehér” megoldás, ha az érintett felhasználók is tudnak róla.

Vannak országok, amelyek masszívan korlátozzák vagy figyelik saját állampolgáraik Internet-használatát.

Ugyanakkor nem valószínű, hogy szabad országok állampolgárait ilyen módon titokban, tömegesen figyeljék. A csaló tanúsítvány ekkor ugyanis rákerül a megfigyelt felhasználó számítógépére. Ha a megfigyelt felhasználónak feltűnik, hogy az oldalnak nem olyan tanúsítványa van, amint amilyen szokott lenni (pl. a `paypal.com` tanúsítványát a csalafinsztáni állami CA bocsátotta ki), akkor a megfigyelő nagyon csúnyán lebukik, és nemigen tudja kimagyarázni magát – sem ő, sem a CA. A megfigyelt aláírt bizonyítékhoz jut a megfigyelésről, és világraszóló botrányt csaphat.

10.4.2.5. Az SSL protokoll hibáját kihasználva

Volt már rá példa, hogy magában az SSL protokollban találtak hibát. Az alábbi eset nagy port vert fel, annak ellenére, hogy csak nagyon keveseket érintett.

Az SSL kapcsolat felépítésekor lefolytatott kulcsegyeztetést (itt nemcsak kulcsokat, de pl. kriptográfiai algoritmuskészleteket is egyeztetnek egymással) később is megismételhetik a résztvevők, ez az ún. *renegotiation*. Ez történhet azért, hogy lecseréljék a régen használt kulcsokat, azért, hogy erősebb kriptográfiai algoritmusokra térjenek át, vagy azért, hogy anonim SSL kapcsolatról (amikor csak a szerver mutatja be a tanúsítványát) kétirányú

[8] `http://www.wired.com/threatlevel/2010/03/packet-forensics/`

autentikációra váltsanak (amikor mindkét félnek van tanúsítványa), vagy vissza.

A probléma ezen renegotiation lehetőséggel kapcsolatos. Az SSL protokoll nem biztosította, hogy renegotiation előtt és után ugyanazon felek vesznek részt a kommunikációban. Így például előfordulhat a következő támadás:

A támadó anonim SSL kapcsolatot létesít egy webszerverrel, majd elküld neki egy kérést, amihez már autentikációra van szükség. A szerver ezért renegotiationt kezdeményez, és elkéri a kliens tanúsítványát. A támadó ekkor egy olyan legális klienssel köti össze a webszervert (man-in-the-middle támadással), aki rendelkezik megfelelő autentikációs tanúsítvánnyal, és épp hozzá akarna kapcsolódni a webszerverhez. A szerver és a legális kliens elvégzik a renegotiationt, kulcsot cserélnek egymás tanúsítványai alapján, és innentől kezdve a támadó már nem lát bele a köztük lévő kapcsolatba. A problémát az jelenti, hogy a szerver az egészből annyit lát, hogy küldtek neki egy kérést, ő autentikációt kért, és az autentikáció sikeres volt, így végrehajthatja a kérést. Ez annyit jelent, hogy a támadó – aki nem rendelkezik megfelelő autentikációs tanúsítvánnyal – be tudott szúrni egy legális kliens autentikált SSL kapcsolata elejére valahány byte-ot.

A hiba magában az SSL protokollban volt, nem pedig valamelyik implementációban, így elvileg minden SSL-t használó szerver érintett lehetett. [154]

10.4.3. Webszerver tanúsítványok biztonsági szintjei

10.4.3.1. Domain validated (DV) tanúsítványok

A domain validated (DV) tanúsítványok a legalacsonyabb biztonsági szintű webszerver tanúsítványok. Ekkor a hitelesítés-szolgáltató mindössze azt ellenőrzi, azt biztosítja, hogy a tanúsítvány alanya valóban kontrollálja-e az adott domaint. Ez történhet például úgy, hogy elküld az adott domainhez tartozó valamely kiemelt e-mail címre egy levelet, és meghatározott választ vár. Ez az ellenőrzés automatizált módon elvégezhető; a DV tanúsítványok esetén más ellenőrzés nem történik. A DV tanúsítványok nagyon egyszerűen és gyorsan bocsáthatóak ki, de alacsony biztonsági szintet jelentenek.

10.4.3.2. Organization validated (OV) tanúsítványok

Az organization validated (OV) tanúsítványok olyan DV tanúsítványok, amelyek esetén a hitelesítés-szolgáltató nemcsak azt ellenőrzi, hogy kontrollálja-e a domain-t, aki a tanúsítványt igényli, hanem ellenőrzi az igénylő szervezet létezését, illetve, hogy az igénylő valóban az adott szervezethez tartozik-e.

10.4.3.3. Extended Validation (EV) tanúsítványok

Az extended validation (EV) tanúsítványok jelentik ma a webszerver tanúsítványok legmagasabb biztonsági szintjét. Az EV tanúsítványok olyan OV tanúsítványok, amelyek teljesítik a CA/Browser Forum által meghatározott, EV tanúsítványokra vonatkozó követelményeket is. [22] E követelmények között szerepel például:

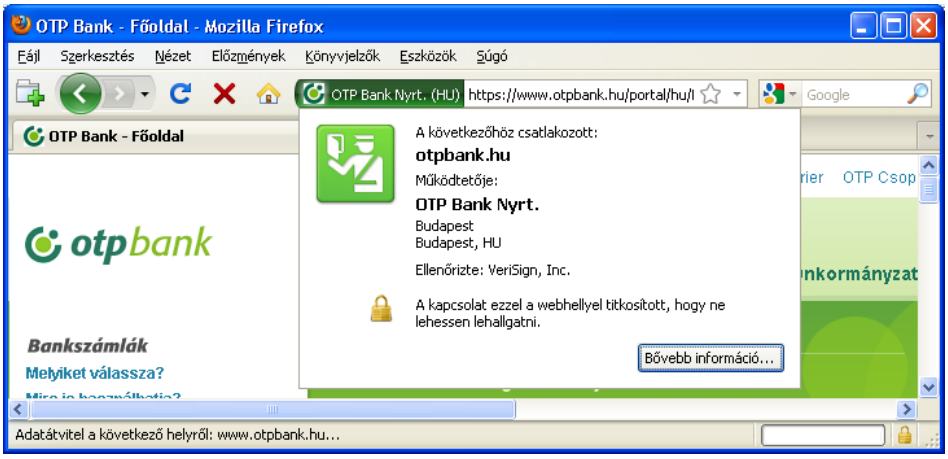
- A regisztrációs eljárásra vonatkozó követelményrendszer;
- A tanúsítványban szereplő mezők kitöltésének adott módja (így könnyen értelmezhető, hogy mely mező mit jelent);
- A kibocsátáshoz használt algoritmusokra vonatkozó követelmények;
- A tanúsítványt kibocsátó hitelesítés-szolgáltatóra vonatkozó pénzügyi követelmények.

A műszaki és eljárásrendbeli követelmények jellemzően nem haladják meg az EU-s, minősített hitelesítés-szolgáltatókra vonatkozó követelményeket. Ugyanakkor az EV követelmények egészen más módon, talán sokkal praktikusabban közelítik meg a problémát. Míg az EU-s megközelítés az igénylő személyének azonosítására, a személyes regisztrációra, és az adatok közhiteles adatbázissal való egyeztetésére összpontosít, az EV követelményekben ezek sokkal kisebb hangsúlyt kapnak. Az EV regisztráció kevésbé az igénylő személyére összpontosít, hanem arra, hogy az igénylést jogosult fél hagyta-e jóvá. Ezen kívül olyan regisztrációs szempontok is megjelennek, mint például, hogy az igénylő (szervezet) elérhető-e a telephelyén, elérhető-e a megadott telefonszámon, és megfelelő üzleti háttérrel rendelkezik-e. (Például három évnél frissebb szervezetek esetén megfelelő pénzügyi fedezettel kell rendelkeznie az igénylőnek.)

Az EV követelmények az EU-s specifikációknál sokkal részletesebben leírják, hogyan kell meggyőződni a különféle típusú szervezetektől érkező igénylések hitelességéről. A következő négy típust különítik el:

- Cégek (akiket cégnyilvántartásban, vagy más hasonló nyilvántartásban lehet ellenőrizni);
- Állami szervezetek (akiket általában jogszabályok hoznak létre);
- Egyéb üzleti szereplők (akikről nincsen közhiteles nyilvántartás, de attól még léteznek);
- Nemzetközi non-profit szervezetek.

Az EV követelmények ország-független regisztrációs eljárást határoznak meg, amelyek lehetővé teszik, hogy egy amerikai szolgáltató egy cseh szervezet részére bocsásson ki tanúsítványt – a személyes regisztrációhoz hasonló biztonsági szinten. E regisztrációs eljárás



10.2. ábra. **EV tanúsítvány esetén a Firefox az URL mezőt zöld színnel jelöli meg, és feltünteti benne a tanúsítvány alanya megnevezésében (DN) szereplő szervezet nevét.**

egy – jellemzően az igénylő országában élő – jogász vagy könyvelő[9] szakvéleményére épít, aki nyilatkozik róla, hogy a megadott regisztrációs lépéseket elvégezte. A regisztrációt ekkor e jogász vagy könyvelő végzi, a szolgáltató feladata pedig a jogász vagy könyvelő kilétének és szakvéleményének ellenőrzése. [22]

A pénzügyi követelmények a globálisan működő, nagy, nemzetközi szolgáltatókra illeszkednek, kisebb országok szolgáltatói nem valószínű, hogy meg tudnak felelni ezeknek.

Az EV tanúsítvány kibocsátására vonatkozó alkalmasságot WebTrust vagy ETSI TS 102 042 alapú audittal lehet igazolni. Az alkalmas szolgáltatók általában a CA/Browser Forum tagjai. [189], [53] Az egyes alkalmazásoknak (Internet Explorer, Mozilla) külön-külön el kell fogadnia, hogy egy adott megbízható gyökértanúsítvány alkalmas EV tanúsítványok kibocsátására.

Az EV tanúsítványokat a böngészők megkülönböztetett módon, például zöld színű URL-lel és az alany szervezetének feltüntetésével jelölik (lásd: 10.2. ábra).

10.4.4. Wildcard (*-ot tartalmazó) tanúsítványok

A *wildcard SSL tanúsítványok* olyan webszerver (SSL szerver) tanúsítványok (speciális autentikációs tanúsítványok), amelyek `"*"` (csillag) karaktert tartalmazó címre szólnak.

A webszerver tanúsítvány azt igazolja, hogy a tanúsítványhoz tartozó magánkulccsal kizárólag a tanúsítványban lévő cím birtokosa rendelkezik. Így a kérdéses webszerverrel titkosított és hitelesített kapcsolatot lehet létesíteni. A webszerver tanúsítványban eredetileg egy és pontosan egy cím szerepelhet (pl: `www.kedvencoldalam.hu`); ez alól a wildcard tanúsítványok jelentenek kibúvót.

[9] Az EV követelmények a Webtrust [189] követelményekre építenek, amelyeket egy kanadai könyvelőket tömörítő szervezet (AICPA/CICA) hozott létre.

A wildcard tanúsítványokban a csillag karakter joker (wildcard) karakterként működik, így a tanúsítványban szereplő cím (pl: `*.kedvencoldalam.hu`) több címre is illeszkedhet. E megoldás előnye, hogy elegendő egy wildcard tanúsítványt vásárolnunk, így több különböző címünkre is van tanúsítványunk. Ugyanakkor a wildcard tanúsítványok általában drágábbak, mint a kizárólag egyetlen címre szóló tanúsítványok.

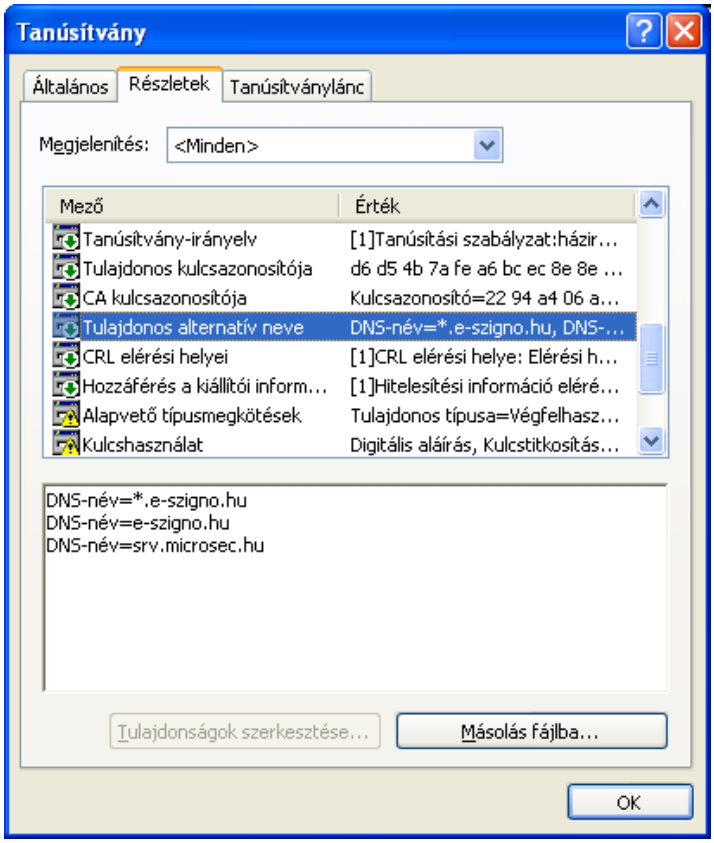
Vegyük figyelembe a következő szempontokat, mielőtt a wildcard tanúsítvány mellett (vagy ellen) döntenénk:

- A `*.xyz.hu` címre szóló tanúsítvány illeszkedik a `https://www.xyz.hu`, a `https://mail.xyz.hu`, a `https://biztonsagos.xyz.hu` címre, de *nem illeszkedik* a `https://xyz.hu` címre (azaz, ha nincs subdomain).
- Csak egy szint írható le a "`*`" karakterrel. A "`*`" karakter nem illeszkedik a "`.`" (pont) karakterre, így a `*.e-szigno.hu` címre kibocsátott tanúsítvány nem illeszkedik az `egyik.masik.e-szigno.hu` címre. A böngészők nem fogadják el a több csillagot tartalmazó címeket sem, így például a `*.*.e-szigno.hu` címre kibocsátott tanúsítvány sem működik.
- Ha a különböző címeink `https://www.xyz.hu`, a `https://mail.xyz.hu` különböző gépeken vannak, akkor ugyan elláthatjuk az összes címet egyetlen tanúsítvánnyal (a tanúsítványhoz tartozó magánkulcsot minden egyes gépünkre fel kell másolnunk), de *bármelyik gépünket feltörik, az összes gép magánkulcsa egyszerre kerül illetéktelen kezekbe*. A támadó ekkor bármelyik gépünket megszemélyesítheti, bármelyik gépnek kiadhatja magát (amíg a tanúsítvány visszavonásra nem kerül).
- Ha a wildcard tanúsítványunk magánkulcsa kompromittálódik, a támadó olyan címekre is szerezhet tanúsítványt, amilyen címekkel mi nem is rendelkezünk. (Például sok adathalász oldal használ `https://nagyonhiresbank.xyzkft.com` jellegű címeket. Ekkor a támadó az XYZ Kft. szerverét törte fel, és e szerver tanúsítványa segítségével próbálja becsapni a bank ügyfeleit.)

A wildcard tanúsítvány akkor igazán jó választás, ha egy gépen üzemeltetünk sok különböző subdomaint, és ezen subdomainekre szeretnénk tanúsítványt. Ekkor egyetlen wildcard tanúsítvánnyal lefedhető az összes subdomain. (Az, hogy hány domain esetén éri meg a wildcard tanúsítvány, az a wildcard tanúsítvány árától függ.)

10.4.5. UCC (több címre szóló) tanúsítványok

Az *UCC tanúsítvány* (Unified Communication Certificates, UCC) olyan webszerver tanúsítvány, amely egyszerre több címre (így esetleg több domainre is) szól, például `www.valami.hu`, `webmail.valami.hu`, `www.valahol.net`. A *wildcard-UCC tanúsítvány* olyan



10.3. ábra. **UCC tanúsítványban az alany alternatív nevei között lehet felsorolni a címeket. Az ábrán olyan Wildcard-UCC tanúsítvány látható, amely illeszkedik az https://e-szigno.hu címre, az e-szigno.hu subdomainjeire, és az srv.microsec.hu címre is.**

webszerver tanúsítvány, amely egyszerre több domain összes subdomainjére is szólhat, például `*.valami.hu`, `*.valahol.net`."

Webszerver tanúsítványok esetén az alany DN-jének `common name` mezejében szerepel, hogy milyen címre szól a tanúsítvány. Ha ehelyett a címeket az alany alternatív nevei (subject alternative names, SAN) között soroljuk fel, akkor *UCC tanúsítvány*ról beszélünk. (Lásd: 10.3. ábra.) A legtöbb böngésző támogatja az UCC tanúsítványokat. (Például a Internet Explorer 7 és 8, valamint a Mozilla Firefox 3.5 feletti verziói igen, de például a Google Chrome 3 még nem.) Ez azt jelenti, hogy amikor az SSL kapcsolat felépítése során a böngésző ellenőrzi, hogy a webszerver tanúsítványa valóban arra a címre szól-e, amelyhez a böngésző kapcsolódik, akkor is elfogadja a tanúsítványt, ha az alany alternatív nevei között szerepel az adott cím. (Ha szerepel a tanúsítványban az alany alternatív nevei (SAN) mezőben DNS név, akkor a common name mezőt figyelmen kívül hagyja, és csak az alternatív nevek között keresi az adott címet.)

Miért van szükség több címet tartalmazó tanúsítványra? Igaz, hogy létezik olyan webszerver,

amelyikkel megoldható, hogy ugyanazon a porton több különböző domaint (illetve subdomaint) szolgáljunk ki, miközben az egyes címekhez külön-külön tanúsítvány tartozik. Ugyanakkor e megoldás csak akkor használható, ha a böngészőprogram még az SSL kapcsolat felépítése előtt is elküldi, hogy pontosan milyen címre szeretne csatlakozni. Tekintve, hogy például az Internet Explorer ezt nem teszi meg, e megoldás nemigen használható a gyakorlatban. *Ha egyazon portról több különböző domaint/subdomaint szeretnénk kiszolgálni, akkor egy tanúsítvánnyal kell lefednünk az összes domaint és subdomaint.* E tanúsítvány lehet joker ("`*`") karaktert tartalmazó wildcard tanúsítvány vagy több címet tartalmazó UCC tanúsítvány.

Az UCC tanúsítványok a wildcard tanúsítványokhoz hasonlóan több címre szólnak, így a wildcard tanúsítványoknál szereplő biztonsági megfontolások nagy része itt is érvényes. Például tegyük fel, hogy több domainünk van, mindegyik domaint külön gépről szolgáljuk ki, és a domainekhez egyetlen UCC tanúsítványt használunk. Ekkor, ha kompromittálódik az UCC tanúsítványhoz tartozó magánkulcs, akkor az összes gépen le kell cserélni a tanúsítványt (és a kulcspárt), mert a támadó az UCC tanúsítvány magánkulcsával a tanúsítványban feltüntetett összes domaint megszemélyesítheti.

Az jelenti a különbséget a wildcard és az UCC tanúsítványok között, hogy míg a wildcard tanúsítvány a "`*`" jelet tartalmazó kifejezéssel illeszkedik több címre, addig az UCC tanúsítványban fel kell sorolni, hogy az mely címekre szól.

Így, ha egy wildcard tanúsítványhoz tartozó magánkulcs kompromittálódik, akkor a támadó az adott domain (pl: `e-szigno.hu`) bármennyi subdomainjét (pl. `archivum.e-szigno.hu`) megszemélyesítheti, olyanokat is, amelyeket a tanúsítvány birtokosa nem is használ. Ha egy UCC tanúsítványhoz tartozó magánkulcs kompromittálódik, akkor a támadó kizárólag a tanúsítványban felsorolt címeket személyesítheti meg.

A tanúsítvány birtokosára ugyanúgy érvényes e korlátozás. UCC tanúsítvány esetén a tanúsítvány igénylésekor el kell dönteni, hogy pontosan milyen címekre (például: `www.e-szigno.hu`, `archivum.e-szigno.hu`) kérjük a tanúsítványt, és ezen később csak a tanúsítvány cseréjével, azaz a tanúsítványt kibocsátó hitelesítés-szolgáltató bevonásával lehet változtatni. A szolgáltatónak vissza kell vonnia a régi tanúsítványt, és új tanúsítványt kell kibocsátania. Az adminisztratív lépések mellett ennek anyagi vonzatai is lehetnek. Wildcard tanúsítványok esetén a tanúsítvány birtokosa később is változtathat a subdomainjein, és ehhez nincsen szükség a hitelesítés-szolgáltató bevonására – feltéve, hogy az új domainek és subdomainek illeszkednek a wildcard tanúsítványban szereplő, "`*`" jelet tartalmazó kifejezésre.

Wildcard tanúsítvánnyal mindig csak egyetlen domain subdomainjei fedhetőek le. Például a `*.valami.hu` tanúsítvánnyal lefedhető a `www.valami.hu`, a `webshop.valami.hu` és a `webmail.valami.hu`, de már nem fedhető le a `www.valahol.net`. Egy UCC tanúsítványban viszont egymástól markánsan eltérő domainek is felsorolhatóak; egyetlen UCC tanúsítvánnyal lefedhető a fenti összes domain és subdomain. A tanúsítvány elvileg bármekkora lehet, így

egyetlen UCC tanúsítvánnyal elvileg bárhány cím lefedhető.

Fontos megjegyezni, hogy a `*.valami.hu` címre szóló tanúsítvány nem illeszkedik a `https://valami.hu` címre, csak a domain első szintű subdomainjeit fedi le (pl. a `https://www.valami.hu` címet). (Lásd: 10.4. ábra.)

Ha egyetlen tanúsítvánnyal szeretnénk lefedni a `www.valami.hu` és a `valami.hu` címet, akkor UCC tanúsítványt kell használnunk.

A wildcard tanúsítványok kombinálhatóak az UCC tanúsítványokkal. A *wildcard-UCC* tanúsítványok olyan webszerver tanúsítványok, amelyekben több címet is felsorolhatunk (UCC), de e címek "`*`" jelet is tartalmazhatnak. Wildcard-UCC tanúsítvánnyal oldható meg, hogy egyazon tanúsítvány illeszkedjen a valami.hu és a `www.valami.hu` címre, valamint a domain összes subdomainjére.

Wildcard-UCC tanúsítvánnyal oldható meg, hogy egyazon tanúsítvány fedje le a `valami.hu`, a `valahol.net` domaineket és ezek összes lehetséges subdomainjeit, így pl. a `www.valami.hu`, `webshop.valami.hu`, `webmail.valahol.net` címeket. Ebben az esetben a tanúsítványban az alany alternatív nevei között a következőket kell feltüntetni:

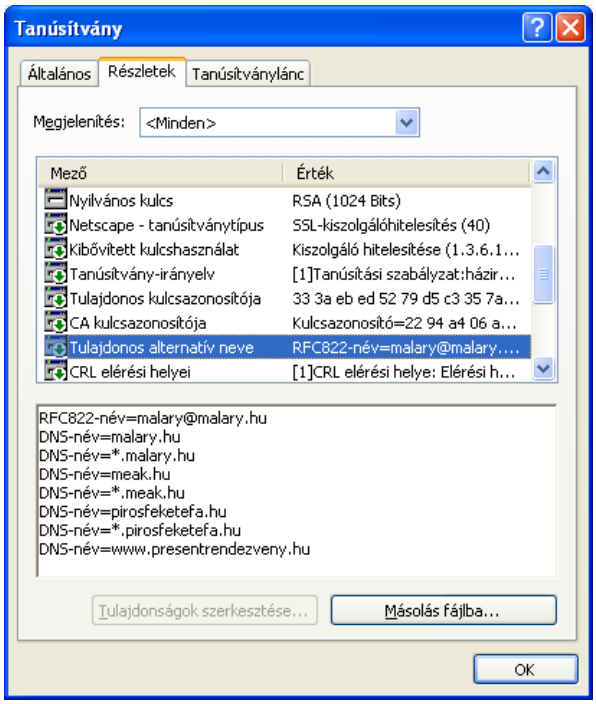
```
DNS=valami.hu
DNS=*.valami.hu
DNS=valahol.net
DNS=*.valahol.net
```

Az UCC, illetve wildcard-UCC tanúsítvány akkor igazán jó választás, ha egy gépen több különböző domaint (illetve subdomaint) szolgálunk ki ugyanazon a porton. Ekkor egyetlen UCC tanúsítvánnyal vagy wildcard-UCC tanúsítvánnyal lefedhető az összes domain és subdomain. Például webhosting szolgáltatóknak javasoljuk e megoldást.

10.5. Tanúsítványok programok aláírásához (code signing)

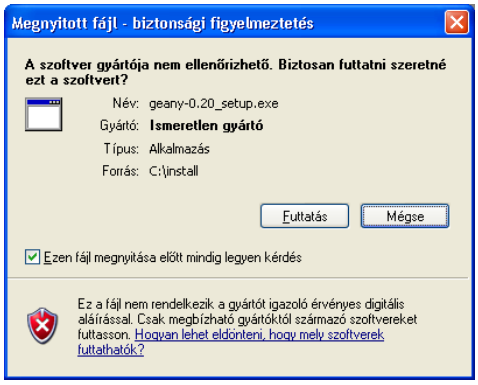
Code signing tanúsítványnak azon tanúsítványokat nevezzük, amelyek számítógépes programok aláírására alkalmasak. Az ezen tanúsítványok szerint aláírt alkalmazásokat egyes programok vagy rendszerek (pl. Windows, Java Virtual Machine) megbízható forrásból származó programoknak ismerik el (lásd: 10.5. ábra). Míg a nem megbízható forrásból származó programok telepítésekor figyelmeztető üzeneteket jelenítenek meg, a code signing tanúsítvány szerint aláírt programok esetén kevesebb és kevésbé fenyegető hibaüzenet jelenik meg (lásd: 10.6. ábra).

A code signingot itt nem tekintjük elektronikus aláírásnak. Egyrészt mert nem jogilag kötelező érvényű aláírásról van szó, hanem a „digitális aláírás” csak a program eredetét és sértetlenségét igazolja. A programot aláíró fél nem szerződést kötött, hanem megjelölte a programot, mint az adott gyártótól származó elfogadható, megbízható programot. Másrészt, mert míg



10.4. ábra. **A fenti wildcard-UCC tanúsítvány illeszkedik a malary.hu címre és a malary.hu domain összes subdomainjére, a meak.hu címre és a meak.hu összes subdomainjére, a pirosfeketefa.hu címre és a pirosfeketefa.hu domain összes subdomainjére. Szintén illeszkedik a www.presentrendezveny.hu címre, de a presentrendezveny.hu más subdomainjeire nem illeszkedik.**

10.5. ábra. **A Windows felismeri, hogy aláírt, ismert gyártótól származó program kerül telepítésre**



10.6. ábra. **A Windows nem talál aláírást a telepítő csomagon, veszélyt jelez, hogy ismeretlen gyártótól származó kód kerül telepítésre**

elektronikus aláírást bárki létrehozhat, a code signing csak megbízható felek által végezhető. Ha egy bűnöző vírust ír, és code signing tanúsítvány szerint aláírja, akkor a vírust sok program, operációs rendszer kérdés nélkül (vagy kevés kérdés alapján) lefuttatja. (Az is előfordult már, hogy a vírust érvényes code signing tanúsítvány lopott magánkulcsával írták alá. [172]) Így a code signing tanúsítványt például – a webszerver tanúsítványhoz hasonlóan – visszavonja a szolgáltató, ha valaki „rossz célra” használja.

A code signing tanúsítványok onnan ismerhetőek fel, hogy speciális, `codeSigning` kiterjesztett kulcshasználatot tartalmaznak.

Ma általánosan elfogadottá válik, hogy a „jó” alkalmazásokon érvényes code signing tanúsítvány alapján ellenőrizhető aláírás van. Sőt egyes platformokra kizárólag olyan programok telepíthetőek, amelyeket „megfelelő” code signing tanúsítvány szerint írtak alá.

10.6. Milyen tanúsítványt engedjünk be?

Tegyük fel, hogy egy szervert üzemeltetünk, és a klienseket tanúsítványuk alapján szeretnénk beengedni. Hogyan járhatunk el?

Ha saját CA-t működtetünk, lehet, hogy úgy döntünk, azokat engedjük be, akik az adott CA-tól származó tanúsítvánnyal rendelkeznek. Ellenben, ha rendszerünk egy más rendszerhez is csatlakozik, vagy ha professzionális szolgáltatót használunk, már nem biztos, hogy mindenkit be akarunk engedni, akinek érvényes autentikációs tanúsítványa van. Ekkor elválik egymástól az, hogy valakinek érvényes tanúsítványa van (ő birtokolja a hozzá tartozó magánkulcsot) attól, hogy az illető jogosult-e az adott erőforrás használatára.

A tanúsítványok között vagy a kibocsátó és a tanúsítványok sorozatszáma, vagy a tanúsítványok DN-je alapján szelektálhatunk. Utóbbi esetben felsorolhatjuk a legális felhasználók teljes DN-jét, de az is lehet, hogy csak a DN bizonyos elemeire teszünk megkötéseket.

10.5. Példa: *Az XY Kft. több professzionális szolgáltató gyökértanúsítványait elfogadja. Egy adott webes szolgáltatásba viszont csak a saját alkalmazottait szeretné beengedni. Ezt elérheti úgy, hogy csak olyan tanúsítványokat enged be, amelyeknek a DN-je…*

- *megfelel az XY Kft. valamely alkalmazottja DN-jének.*
- *az* `O=XY Kft.` *elemet tartalmazza.*
- *olyan e-mail címet tartalmaz, amelynek a vége* `@xykft.hu`*.*

Ne feledjük, hogy a tanúsítványok változnak. Ha a tanúsítványokra sorozatszámukkal hivatkoztunk, akkor a hozzáférési szabályokat frissítenünk kell, amint egy tanúsítvány változik (pl. lejár, és megújításra kerül). Ha a felhasználókra a teljes DN-nel hivatkoztunk, a hozzáférési szabályokat frissítenünk kell, amint egy felhasználó DN-je megváltozik (pl. férjhez megy, és felveszi a férje nevét). Ha a tanúsítványokra generikus szabályokkal (pl. megadott `O` elem, megadott mintára illeszkedő e-mail cím) hivatkoztunk, a hozzáférési szabályokat lehet, hogy egyáltalán nem kell frissítenünk, de gondolnunk kell rá, ha később valaki nem jogosult is olyan tanúsítványhoz juthat, amely illeszkedik a szabályra (pl. később egy külsős alvállalkozó is céges e-mailt kap).

A következő, 11. fejezetben részletesen is körüljárjuk a tanúsítványok és szerepkörök kapcsolatából eredő problémát.

10.7. PKI és single sign-on

A „single sign-on” kifejezés azt jelenti, hogy egy decentralizált rendszerben egyszer, egy ponton, egy módon kell bejelentkezni, és ezen bejelentkezést később más helyen, más rendszerben vagy a rendszer más komponenseiben is felhasználhatjuk. A single sign-on rendszerek előnye, hogy kevesebb jelszóra, kevesebb bejelentkezésre van szükség, alapvetően elég egy jelszót fejben tartani, és ezzel sok helyre beléphetünk. A hátrányuk ugyanez: ha azt az egy jelszót elfelejtjük, egyszer csak sehova nem tudunk belépni, és ha az az egy jelszó rossz kezekbe kerül, egyszerre sok helyen személyesíthetnek meg bennünket.

A single sign-on rendszerekben ún. Identity Providerek azonosítják a felhasználókat, és vannak ún. Service Providerek, akik szolgáltatásokat nyújtanak, és ennek keretében elfogadják az Identity Providerek által végzett azonosítást, és ezen azonosítás alapján engednek hozzáférést a szolgáltatásaikhoz. Ha egy felhasználó igénybe akar venni egy szolgáltatást, a Service Provider átirányítja az Identity Providerhez, aki meggyőződik róla, hogy a felhasználó valóban az, akinek mondja magát. Ennek a módja, illetve biztonsági szintje az adott Identity Providertől függ, történhet jelszóval, autentikációs tanúsítvánnyal vagy akár biometriai módszerekkel. A sikeres azonosítást követően az Identity Provider visszairányítja a

felhasználót a Service Providerhez, illetve az Identity Provider esetleg igazolhatja a felhasználó egyes adatait, attribútumait (11. fejezet) is.

Léteznek teljesen decentralizált single sign-on rendszerek, ahol az Identity Provider csupán annyit vállal, hogy ha egy felhasználóról azt állítja, hogy ő az X azonosítóval rendelkezik, akkor egy másik felhasználóról nem fogja ugyanezt állítani. Ekkor a Service Provider mindössze annyi információhoz jut, hogy meg tudja különböztetni az adott felhasználót mindenki mástól, de nem tudja meg, hogy az illető kicsoda. Ilyen nyitott, pszeudonim technológia például az OpenID[10], ahol jellemzően minden Service Provider minden Identity Providert elfogad.

Más single sign-on rendszerekben az egymásban megbízó, jellemzően azonos biztonsági követelményekkel rendelkező Identity Providerek és Service Providerek ún. föderációkba szerveződnek. Ekkor a föderáció Identity Providerei nem feltétlenül csak álnevet adnak meg a felhasználótól, hanem a felhasználó kilétét igazolják a Service Provider számra. Például, az A egyetem hallgatója az A egyetem Identity Providere által végzett azonosítás alapján léphet be a B egyetem könyvtárának szerverére, nem szükséges ott külön felhasználói fiókkal rendelkeznie. Ilyen, föderáció alapú technológia például a Shibboleth[11].

E két megközelítés között nem mindig éles a határ, és vannak olyan technológiák is, amelyekkel mindkét megközelítés szerinti rendszer építhető. Ilyen megoldás például a Microsoft által kidolgozott Information Cards[12].

Ellison és Schneier 2000-ben megjelent cikkükben a PKI-t és a single sign-ont egymással ellentétes megoldásnak állították be [40], mert míg a PKI folyamatos azonosítást követel meg, a single sign-on alaptétele, hogy egy azonosítás elegendő. Álláspontunk szerint a probléma ennél lényegesen összetettebb, e két technológia egyáltalán nem mond ellent egymásnak.

A legtöbb single sign-on megoldás masszívan támaszkodik PKI-re, mind a felhasználó és az Identity Provider közötti kapcsolat, mind az Identity Provider és a Service Provider közötti kapcsolat esetén. Gyakori, hogy a sikeres azonosítás tényéről a Service Provider vagy egy HTTPS kapcsolaton keresztül értesül[13], vagy – például Information Cards esetén – az Identity Provider a saját webszerver[14] tanúsítványának magánkulcsával digitálisan aláír egy igazolást[15], amelyet a felhasználó mutat be a Service Providernek.

Ezen túl, mivel a single sign-on megoldás elfedi a Service Provider elől azt, hogy a felhasználó pontosan hogyan azonosítja magát, a single sign-on rendszerkbe viszonylag könnyű bevezetnei egy PKI-s, kártya alapú beléptetést, mert ez csak az Identity Providert érinti, a Service Providereket nem. Így a szolgáltatók úgy használhatják a tanúsítvány-alapú autentikáció

[10] `http://openid.net/`

[11] http://shibboleth.internet2.edu/

[12] http://informationcard.net/foundation

[13] OpenID esetén a sima HTTP is megengedett, de a HTTPS használata jelentősen biztonságosabb.

[14] Ne feledjük, amerikai specifikációról van szó, a webszerver által létrehozott digitális aláírás nem feltétlenül tekinthető jogkövetkezménnyel is bíró, elektronikus aláírásnak.

[15] Information Cards esetén ez egy XML formátumú igazolás, amelyen XMLDSIG aláírást helyeznek el.

előnyeit, hogy nem kell foglalkozniuk a tanúsítvány-alapú autentikáció olyan részleteivel, mint a megbízható gyökértanúsítványok kezelése, a visszavonási listák ellenőrzése, a felhasználók és autentikációs tanúsítványok összerendelése, a tanúsítványcserék adminisztrálása stb.

10.8. Összegzés

- Autentikációnak (partner hitelesítésnek) azt nevezzük, amikor távolról, biztonságosan győződünk meg valakinek a kilétéről.
- Ilyenkor általában biztonságos (titkosított és hitelesített) csatornát is ki szoktunk építeni. A kulcscserét nyilvános kulcsú alapon, a további kommunikációt a kicserélt szimmetrikus kulcsok alapján végezzük.
- Az SSL az egyik legelterjedtebb PKI alapú autentikációs megoldás. A webszerverek SSL tanúsítványai jelentik a PKI talán legszélesebb körű alkalmazását.
- Az tanúsítvány alapú autentikáció teljesen független az elektronikus aláírástól, az elektronikus aláírást szabályozó jogszabályok nem vonatkoznak rá.

11. fejezet

Tanúsítvány és szerepkör – Attribútum-tanúsítványok

„A bölcs Sir Bedevir volt az első, aki beállt Arthur királyhoz lovagnak. De hamarosan követte őt sok más neves személyiség is: Sir Lancelot a bátor, Sir Galahad, az érintetlen, Sir Robin, a nem annyira bátor, mint Sir Lancelot; aki majdnem legyőzte az angnori sárkányt, aki majdnem kiállt az ádáz bristoli tyúkok ellen, és aki berezelt a Badon Hill-i csatában."

– *Monty Python, Gyaloggalopp*

A tanúsítvány igazolja, hogy egy adott kulcspár ilyen és ilyen nevű felhasználóhoz tartozik. Gyakran nemcsak arra vagyunk kíváncsiak, hogy az aláírói tanúsítvány alanya kicsoda, hanem azt is szeretnénk tudni, hogy az illető milyen szerepkörrel, jogosultsággal, tulajdonsággal rendelkezik, azaz milyen minőségben használja a tanúsítványát. Lehet, hogy egyszerűen magánszemélyként kívánja használni a tanúsítványt, de az is lehet, hogy valamely szervezet tagjaként vagy munkatársaként, egy vállalat képviselőjeként, valamilyen hivatással (ügyvéd, közjegyző) rendelkező személyként, vagy egy szolgáltatás előfizetőjeként.

Egyik lehetőség, hogy ezen információkat is a tanúsítvány alapján állapítjuk meg. Amikor a hitelesítés-szolgáltató beírja a tanúsítványba a felhasználó megnevezését, a neve mellett egyéb adatokat is feltüntethet benne. Ezen egyszerű megoldásnak jelentős korlátai vannak: például nem szerencsés, ha mindenki, aki a tanúsítványunkat kezeli, egyúttal minden adatunkról, jogosultságunkról is tudomást szerezhet. Másrészt, ha a tanúsítványban szereplő adataink bármelyike megváltozik, a hitelesítés-szolgáltató vissza kell, hogy vonja, le kell, hogy cserélje a tanúsítványt, ami esetleg túl sok felesleges adminisztrációt jelentene.

Másik lehetőség, hogy adatainkat, tulajdonságainkat, jogosultságainkat (együttesen: *attribútum*) nem a hitelesítés-szolgáltató igazolja, és nem a tanúsítványban szerepeltetjük, hanem minden attribútumot az igazol, aki az adott attribútumot kezeli, és ezen igazolásokat

csatoljuk a tanúsítványunkhoz, ha az szükséges. Például a munkáltatói igazolást állítsa ki a munkahelyünk, pénzügyi helyzetünkről az igazolást állítsa ki a bankunk, egészségügyi adatainkat pedig a megfelelő orvos igazolja. Mindezen adatokhoz semmi köze sincs a hitelesítés-szolgáltatónak.

Ha ezen igazolást olyan, szabványos, géppel értelmezhető dokumentumként hozzák létre, amely könnyen összekapcsolható tanúsítványunkkal, akkor úgynevezett *attribútum-tanúsítvány*ról beszélünk.

E fejezetben az attribútum-tanúsítványok nyújtotta lehetőségeket járjuk körül. Bár az itt szereplő megállapítások többsége egyaránt vonatkozhat aláíró, titkosító és autentikációs tanúsítványokra, e fejezetben elsősorban az aláírói tanúsítványokról szólunk.

11.1. Szerepkör megállapítása tanúsítvány alapján

Attól függően, hogy egy hitelesítés-szolgáltató milyen információkat, valamint hol és hogyan tüntet fel valakinek a tanúsítványában, az illető jogosultságai más és más módon állapíthatóak meg a tanúsítvány alapján. Ez gyakran interoperabilitási problémákhoz vezet: előfordulhat, hogy egy rendszer nem engedi be a jogosult felhasználót, de az is lehet, hogy jogosulatlan felhasználót enged be, vagy nem a megfelelő jogosultságokkal enged be valakit.

Bemutatjuk, hogy milyen megoldások terjedtek el a szerepkörök, jogosultságok, tulajdonságok – együttesen: attribútumok – tanúsítványban való szerepeltetésére. Egyúttal azt is megmutatjuk, hogy ezek a megoldások milyen korlátokkal rendelkeznek, miért nem skálázhatóak, miért nem működnek nagy rendszerekben, ahol sok hitelesítés-szolgáltató, sok felhasználó és sok attribútum van jelen.

A felvetett problémákra az jelenti a választ, ha az attribútumok nem a tanúsítványban kapnak helyet, hanem a tanúsítvány független marad attól, hogy mire használják. Később az *attribútum-tanúsítvány*ok technológiáját mutatjuk be, mint az egyik legjobb megoldást e területen.

11.1.1. Implicit kapcsolat

Implicit kapcsolatról akkor beszélünk, ha valamilyen rendszerben csak a „jogosult” személyek rendelkeznek tanúsítvánnyal. Ekkor, ha valakinek tanúsítványa van, az egyben azt is jelenti, hogy az illető rendelkezik a rendszer használatához szükséges szerepkörrel, jogosultsággal.

Tegyük fel, hogy egy baráti társaság saját mini hitelesítés-szolgáltatót hoz létre, és ez a szolgáltató kizárólag a baráti társaság tagjainak bocsát ki tanúsítványt. Ha érvényesnek találunk egy tanúsítványt vagy aláírást a baráti társaság gyökértanúsítványa alapján, akkor nemcsak arról győződünk meg, hogy a tanúsítványhoz tartozó magánkulcsot az illető személy birtokolja, hanem egyben arról is, hogy ő a baráti társaság tagja.

Azzal, hogy az autentikáció (meggyőződünk róla, hogy valóban ő az) és autorizáció (jogosultságot adunk valamihez) fogalmait összemossuk, egyszerű és könnyen kezelhető rendszerhez jutunk. Elegendő a tanúsítvány érvényességét ellenőriznünk, az érvényes tanúsítvány egyben a baráti társaság tagsági viszonyát is igazolja, így az alanyt feljogosítja arra, hogy a társaság erőforrásait használja.

E megoldásnak előnye, hogy az alany jogosultságai a tanúsítványával együtt, egy helyen visszavonhatóak.

Ugyanakkor ennek a megoldásnak a következő hátrányai vannak:

- ilyen módon csak egyetlen szerepkör, csak egyetlen attribútum kezelhető,
- e megoldás nehezen kapcsolható más rendszerekhez, ezért elsősorban zárt közösségben használható.

Hogyan fogadhatja el az egyik közösség egy másik közösség tanúsítványát, és hogyan biztosítható ilyen esetekben a rendszerek együttműködése?

A következő felsorolás néhány megoldási lehetőséget[1] mutat az implicit kapcsolat problémáinak kezelésére, figyelemmel a hátrányos következményekre is:

1. *Az egyes közösségek megbíznak egymás gyökértanúsítványaiban. Így minden egyes közösség és minden egyes jogosultság egy-egy gyökértanúsítványt jelent.*

   Ezáltal nagyon nehézzé válik egy új közösség/jogosultság bevezetése, mert az adott gyökértanúsítványt minden egyes végfelhasználó gépére telepíteni kell. A megszűnt vagy kompromittálódott gyökértanúsítványok eltávolítása is problémákhoz vezet, mert azokat külön-külön minden egyes végfelhasználónál törölni kell a rendszerből.

   A megoldás hátrányos tulajdonsága még, hogy a gyökértanúsítványok számának növekedésével egyenes arányban a felmerülő problémák száma is növekszik.

2. *Az egyes közösségek gyökértanúsítványaival rendelkező CA-k kereszthitelesítik (5.2.1. fejezet) egymást (tanúsítványokat bocsátanak ki egymás számára).*

   Ezáltal az egyes közösségeken kívüli személyek tanúsítványait is lehet majd ellenőrizni a közösség gyökértanúsítványa alapján.

   A megoldás hátrányos tulajdonsága, hogy a fent ismertetett módszerrel nem lehet megállapítani, hogy ki milyen szerepkörrel, jogosultsággal rendelkezik, hiszen a kereszthitelesítés miatt több PKI közösségnek is tagjává válik.

3. *A közösségek új, közös CA-t hoznak létre, amely tanúsítványt bocsát ki az egyes közösségi CA-k számára.*

[1] A hitelesítés-szolgáltatói hierarchiák összekapcsolásának módjait a tanúsítványláncokról szóló fejezetben (5. fejezet) mutattuk be részletesen.

Ebben az esetben a jogosultságok, illetve szerepkörök a tanúsítványláncokban fellelhető köztes CA-k tanúsítványai alapján vezethetők le.

A megoldás hátrányos tulajdonsága, hogy ekkor az egyes alkalmazásokba „bele kell drótozni" a köztes tanúsítványokat is azért, hogy az alkalmazások meg tudják állapítani a megfelelő szerepköröket, illetve jogosultságokat. A köztes tanúsítványok lejárta, visszavonása, cseréje az alkalmazásokat is érinti, ezáltal a művelet biztonságos elvégzése nagyon körülményessé válik. Az is előfordulhat, hogy valaki több közösségnek is tagja, ezért az egyes közösségektől továbbra is külön-külön tanúsítványokat kell beszereznie. Ráadásul a megoldás nem szabványos, a tanúsítványlánc jellegzetességeiből kívülálló nem tud az alany jogosultságaira következtetni.

Összefoglalva: *Nagy rendszerekben – ahol több közösség, több jogosultság jelenik meg – az implicit kapcsolaton alapuló megoldás nem alkalmazható.*

11.1.2. Az attribútum a tanúsítványban szerepel

Másik lehetőség, hogy a tanúsítvány tartalmaz olyan mezőket, amelyekből az alany szerepköre vagy jogosultsága megállapítható. Ez például a következő pontokon tüntethető fel (3.4. fejezet):

1. Az alany megnevezésében (DN, distinguished name): Ekkor a jogosultság általában a `title` (emellett esetleg az `organization`, `organization unit` stb.) mezőben jelenik meg. Például az aláírói tanúsítványban lévő Subject DN `title` elemének értéke „ügyvezető".

   A megoldás hátrányos tulajdonsága, hogy a szöveges leírásokat nehéz géppel automatizáltan feldolgozni, és az ott feltüntetett adatok csak egy adott nyelvterületen belül értelmezhetőek. Minél több hitelesítés-szolgáltató és regisztrációs szervezet működik egy PKI közösségben, a Subject DN ilyen módon történő használata annál több interoperábilitási problémát eredményezhet.

   > ***11.1. Példa:*** *A tanúsítványban „ügyvezető" helyett szerepelhet „Ügyvezető", „ÜGYVEZETŐ", „ügyvezető igazgató", „vezérigazgató" stb. is. Ráadásul, a szöveg akár különböző (latin-2, UTF-8) kódolással, vagy ékezet nélkül is szerepelhet.*

2. A tanúsítványra vonatkozó valamely hitelesítési rendben (`certificate policies`): A tanúsítvány tartalmazza (tartalmazhatja) a rá vonatkozó hitelesítési rendek azonosítóját. A hitelesítési rend pedig szövegesen írja le, hogy az adott rendnek megfelelő tanúsítvány alanya mely attribútummal rendelkezik. A megoldás előnyös tulajdonsága,

hogy a rendre történő hivatkozás OID alapján történik, így számítógép is könnyen fel tudja dolgozni.

A megoldás hátrányos tulajdonsága, hogy amennyiben gépi feldolgozásra nincs lehetőség, természetes személy nagyon nehezen tudja az OID-be kódolt adatokat értelmezni. Másfelől pedig túlságosan körülményes minden szerepkörhöz, jogosultsághoz, tulajdonsághoz külön-külön hitelesítési rendet felvenni.

3. A szerepkör, illetve jogosultság egyéb helyeken (pl. `subjectDirectoryAttributes` kiterjesztés) van feltüntetve.

   A megoldás hátrányos tulajdonsága, hogy nagyon kevés szolgáltató, és így nagyon kevés alkalmazás támogatja a megoldást.

A magyar közigazgatásban a fenti 1. és 2. megoldás keverten[2] jelentkezik, mert:

- a hitelesítési rend alapján el lehet dönteni, hogy az alany közigazgatási szerepkört tölt-e be[3],
- további finomítás pedig az alany DN-je alapján lehetséges.

Ha a tanúsítványt több célra is szeretnénk használni, várhatóan többféle szerepkört, illetve jogosultságot is fel kell tüntetni a tanúsítványban. (Például valaki egy cég ügyvezetője, de emellett egy egyesület elnökségi tagja, és egyúttal ügyvéd is.) E megoldásnak hátrányos következményei vannak:

- Nehéz megállapítani, hogy az alany éppen melyik szerepében, melyik jogosultsága szerint használja/használta a tanúsítványát.
- Ha egy tanúsítványban szereplő bármely szerepköre, jogosultsága, tulajdonsága megváltozik, a tanúsítványt vissza kell vonni, és helyette újat kell kibocsátani.
- A tanúsítvány lecserélését a hitelesítés-szolgáltatónak kell végeznie, aki jellemzően nem ugyanaz a fél, mint aki az alany szerepköreiről, jogosultságairól dönt. Ez már önmagában is jelentősen megnehezíti, megdrágítja a folyamatot.
- Nem helyes, ha olyan adatbázisok jönnek létre, amelyekben valakinek minden attribútuma összegyűlik. A hitelesítés-szolgáltatónak nincs köze az alany attribútumaihoz, csak közvetítő szerepet játszik, amikor az attribútumokat igazolja. Elvileg nem is kellene tudnia az attribútumokról.

[2] A közelmúltban megjelent 78/2010. Kormányrendelet az itt leírtakon kívül más eseteket is megenged, de a műszaki specifikációk még a korábbi, 194/2005. Kormányrendelet szerinti jogi szabályozásra épülnek.

[3] A közigazgatás számára más hitelesítési rendek szerint lehet tanúsítványt kibocsátani, mint a közigazgatás ügyfelei számára.

- A tanúsítvány lecserélésére – elsősorban minősített tanúsítvány esetén – összetett szabályok vonatkoznak. Az új tanúsítványt jellemzően másik magánkulcshoz kell kibocsátani, és esetleg a teljes – a személyes találkozót is igénylő – regisztrációs eljárást meg kell ismételni.

- Az adatvédelmi szabályok miatt egy tanúsítványból közvetlenül[4] nem határozható meg, hogy az alany kicsoda. Ezért a célrendszerekben valamilyen „másodlagos regisztrációt" kell használni, nyilvántartásba kell venni az alany tanúsítványát, és fel kell jegyezni, hogy az melyik alanyhoz tartozik. Ha az alany tanúsítványa gyakran változik, ezt a körülményes műveletet is gyakran meg kell ismételni.

- Ha az alany sok attribútummal (szerepkörrel, jogosultsággal) rendelkezik, nem szeretné feltétlenül, hogy a tanúsítványából (és így például minden aláírásából) kiderüljön, hogy pontosan milyen attribútumai vannak. Sőt, a tanúsítványt kibocsátó hitelesítés-szolgáltató tanúsítványtárából esetleg még visszamenőleg is megállapítható lehet, hogy kinek mikor milyen attribútumai voltak, melyiket mikor szerezte, és mikor veszítette el.

11.1.3. Az attribútum az alany állításából derül ki

Elsősorban papír alapú rendszerekben gyakori, hogy amikor valaki aláír egy dokumentumot, ő maga nyilatkozik róla, hogy milyen szerepkörben írja alá. A dokumentumot felhasználó fél elfogadhatja az aláíró állítását, de úgy is dönthet, hogy – egy esetleg igen körülményes eljárás keretében – utánajár az aláíró szerepkörének, jogosultságának. E megoldásnak kockázata, hogy az aláíró hazudhat, olyan attribútumot is állíthat magáról, amellyel nem rendelkezik. Ekkor a dokumentumot felhasználó fél bíróság előtt felelősségre vonhatja az aláírót a hamis állítást tartalmazó aláírt dokumentum alapján.

Megjegyezzük, e megoldás megdöbbentően jól működik egyes papír alapú rendszerekben.

11.1.4. Az attribútumot más informatikai rendszer tartalmazza

Kézenfekvő megoldás, ha a szerepkört mindig az igazolja, aki az adott szerepkörről dönt, illetve aki azt igazolni jogosult. Ahogy papír alapon, úgy elektronikusan is váljon el az aláírás attól, hogy az aláíró milyen minőségben használja az aláírását. Ebben az esetben az alany tanúsítványa egyedül azt igazolja, hogy az alany valóban birtokolja az aláírói tanúsítványhoz tartozó magánkulcsot, míg a kérdéses szerepkört az erre jogosult szervezet vagy intézmény igazolja.

11.2. Példa: *Tegyük fel, hogy papír alapon benyújtok egy beadványt az X hivatalhoz. Ha igazolnom kell, hogy*

[4] A tanúsítvány alanya személyazonosságának meghatározásához a hitelesítés-szolgáltató által felvett, nem nyilvános regisztrációs adatok is szükségesek.

- *mérnök vagyok, akkor a diplomámat nyújtom be, amelyet a műszaki egyetem állított ki, és írt alá.*
- *nincsen tartozásom az adóhatóságnak, akkor az adóhatóság által kiállított és aláírt igazolást kell benyújtanom.*
- *van munkahelyem, akkor a munkahelyem által kiállított munkáltatói igazolást nyújtom be.*

Ebből az következik, hogy az aláírói tanúsítvány ellenőrzését követően valamilyen más rendszertől származó bizonyítékok alapján kell ellenőrizni, hogy az alany valóban rendelkezik-e a megfelelő jogosultsággal, illetve szerepkörrel.

A megoldás vitathatatlan előnye, hogy a felhasználóknak elegendő egyetlen (aláírói) tanúsítványt birtokolniuk, és azt minden alkalmazási területen, minden szerepkörben felhasználhatják. Nem kell a tanúsítványt visszavonni és újat kibocsátani, ha az alany valamely attribútuma megszűnik, vagy ha az alany új attribútumhoz jut. A tanúsítványt kibocsátó hitelesítés-szolgáltatónak nem kell nyilvántartania, hogy az egyes alanyok milyen attribútumokkal rendelkeznek. A szerepkörök, jogosultságok kiosztása, illetve megszüntetése ott történik, ahol az attribútumok használatáról döntenek, és arról megfelelő nyilvántartást vezetnek.

A tanúsítvány felhasználásakor (például aláírás létrehozásakor) kapcsolatba kell lépni azokkal a rendszerekkel, amelyek az alany kérdéses attribútumait nyilvántartják, és az adott szerepkört, jogosultságot bizonyító, a szervezet elektronikus aláírásával ellátott igazolásokat szolgáltatnak.

E kapcsolatot felépítheti a tanúsítvány alanya vagy a tanúsítványt elfogadó érintett fél. A kapott igazolás lehet pillanatnyi, de akár hosszú ideig is érvényes lehet. Elképzelhető olyan rendszer, ahol a kapcsolatot minden egyes felhasználáskor online fel kell építeni, de elképzelhető, hogy egy hosszú ideig érvényes (vagy egy időbélyeggel ellátott) igazolást elengedő egyszer beszerezni.

Az attribútum-tanúsítványok ezen harmadik féltől származó igazolások kezelésére nyújtanak rugalmas és szabványos megoldást.

11.2. Mit nevezünk attribútum-tanúsítványnak?

A félreértések elkerülése végett a továbbiakban az aláírásra, autentikációra és titkosításra használható, nyilvános kulcsot tartalmazó tanúsítványokra „nyilvános kulcsú tanúsítvány” néven hivatkozunk.

Az attribútum-tanúsítvány olyan igazolás, amely egy nyilvános kulcsú tanúsítványhoz, vagy a nyilvános kulcsú tanúsítvány alanyához kapcsolódik, és alkalmas a nyilvános kulcsú

tanúsítvány alanyához tartozó egy vagy több szerepkör, jogosultság, tulajdonság (együttesen: attribútum) igazolására.

Az alany nyilvános kulcsát és „kilétét”[5] a nyilvános kulcsú tanúsítvány alapján lehet megállapítani[6], míg szerepköreit, jogosultságait, tulajdonságait (attribútumait) pedig attribútum-tanúsítványai tartalmazzák.

Az attribútum-tanúsítvány hivatkozást tartalmaz a nyilvános kulcsú tanúsítványra (vagy annak alanyára), így egy adott attribútum-tanúsítványról és egy adott nyilvános kulcsú tanúsítványról megállapítható, hogy a két tanúsítvány alanya megegyezik.

Az alany nyilvános kulcsát a nyilvános kulcsú tanúsítvány tartalmazza. *Az attribútum-tanúsítvány nem tartalmaz kulcsot.* Mivel az attribútum-tanúsítványban nincs kulcs, az attribútum-tanúsítvány *használatához intelligens kártyára sincs szükség.*

11.2.1. Hogyan kapcsolódik az attribútum-tanúsítvány az alanyhoz?

Az attribútum-tanúsítvány többféleképpen kapcsolódhat a nyilvános kulcsú aláírói tanúsítványhoz. [144]

Tartalmazhatja:

1. a nyilvános kulcsú aláírói tanúsítvány kibocsátójának megnevezését (DN) és a nyilvános kulcsú aláírói tanúsítvány sorozatszámát. (`baseCertificateID`)

2. a nyilvános kulcsú aláírói tanúsítvány alanyának megnevezését, azaz DN-jét. (`entityName`)

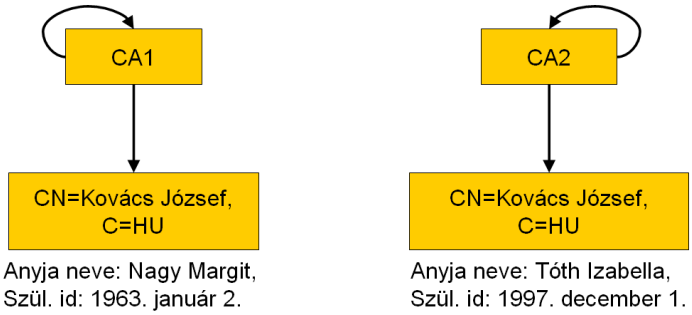
3. a nyilvános kulcsú aláírói tanúsítvány kriptográfiai lenyomatát. (`objectDigestInfo`[7])

Az 1. és a 3. megoldás magára a nyilvános kulcsú tanúsítványra hivatkozik, így ha az alany új tanúsítványt kap, azzal már nem használható az attribútum-tanúsítványa. A 2. megoldás esetén az attribútum-tanúsítvány a nyilvános kulcsú tanúsítvány alanyára, és nem az alany tanúsítványára hivatkozik. Így, ha a nyilvános kulcsú tanúsítvány lejár (vagy visszavonásra kerül), és az alany új tanúsítványt kap (azonos DN-nel), az attribútum-tanúsítvány az új nyilvános kulcsú tanúsítvánnyal is használható. A hitelesítés-szolgáltató ugyanazt a DN-t nem rendelheti másik alanyhoz, tehát az adott szolgáltatón belül a DN egyértelműen meghatározza, hogy kiről van szó.

[5] A személyazonosság megállapításához a nyilvános kulcsú tanúsítvány nem elegendő, a hitelesítés-szolgáltató által felvett regisztrációs adatok is szükségesek hozzá. Jogvita estén a hitelesítés-szolgáltató segítségével állapítható meg a nyilvános kulcsú tanúsítvány alanyának személyazonossága.

[6] Ahogy azt az előző fejezetekben leírtuk, a nyilvános kulcsú tanúsítvány is tartalmazhat attribútumokat.

[7] Az `objectDigestInfo` segítségével nemcsak az aláírói tanúsítványra, hanem a benne lévő nyilvános kulcsra is hivatkozhatunk. Ekkor az attribútum-tanúsítvány az adott kulcshoz tartozó összes tanúsítvánnyal használható.



11.1. ábra. **A DN csak hitelesítés-szolgáltatón belül egyedi, globálisan nem az.**

Míg a hitelesítés-szolgáltatók egyazon DN-nel csak egyetlen alanynak bocsáthatnak ki tanúsítványt, a DN globálisan nem egyedi. Magyarországon például több, egymástól független hitelesítés-szolgáltató is működik, és semmi sem garantálja, hogy két hitelesítés-szolgáltató nem ugyanazt a DN-t adja két különböző személynek. (lásd: 11.1. ábra)

A hitelesítés-szolgáltatók DN-je sem feltétlenül egyedi. Még kisebb PKI közösségekben is könnyen előfordulhat, hogy két szolgáltatónál egy köztes hitelesítő egység véletlenül azonos DN-t kap. (Lásd: 11.1. ábra.) Világviszonylatban pedig mindenképpen számolni kell ezzel a körülménnyel.

Egyedül a 3. megoldás (objectDigestInfo) segítségével lehet globálisan egyedi módon hivatkozni egy nyilvános kulcsú aláírói tanúsítványra. E megoldás támogatása a szabványok szerint nem kötelező, így sok alkalmazás – például az attribútum-tanúsítványokat egyébként támogató Acrobat Reader – nem is támogatja.

11.2.2. Nemzetközi műszaki specifikációkban

11.2.2.1. Főbb mértékadó specifikációk

Az attribútum-tanúsítványok szintaxisát az X.509 ajánlás tartalmazza (így az „X.509 tanúsítvány” fogalom elvileg nyilvános kulcsú tanúsítványt és attribútum-tanúsítványt is jelentene). [191] Az attribútum-tanúsítványok profilját, kezelését az RFC 3281 is meghatározza. [144] Az RFC 3281 egyúttal az attribútum-tanúsítóra és az ő aláírói tanúsítványára vonatkozó követelményeket is meghatároz:

> RFC 3281:
> `4.5 Profile of AC issuer's PKC`
>
> `The AC issuer's PKC MUST conform to [PKIXPROF], and the keyUsage extension in the PKC MUST NOT explicitly indicate that the AC issuer's public key cannot be used to validate a digital signature.`

```
In order to avoid confusion regarding serial numbers and
revocations, an AC issuer MUST NOT also be a PKC Issuer. That is,
an AC issuer cannot be a CA as well. So, the AC issuer's PKC MUST
NOT have a basicConstraints extension with the cA BOOLEAN set to
TRUE.
```

Vagyis az attribútum-tanúsítvány aláírásához használt tanúsítvány nem lehet hitelesítés-szolgáltatói tanúsítvány. Így az attribútum-tanúsítványt nem hitelesítés-szolgáltató, hanem a hitelesítés-szolgáltatás végfelhasználója bocsátja ki, és elektronikus aláírással látja el. Ezért az attribútum kibocsátó végfelhasználó tanúsítványának alkalmasnak kell lennie elektronikus aláírás létrehozására.

Az ETSI TR 102 044 az RFC 3281-re hivatkozva az attribútum-tanúsítványokra vonatkozó általános követelményeket írja le. Ismerteti az attribútum-tanúsítványokra vonatkozó európai gyakorlatot, valamint javaslatot tesz az egyes főbb általános attribútumok szabványos megnevezésére (OID segítségével történő jelzésére), és bemutat egy olasz példát is. [54]

Az ETSI TS 102 158 olyan szabályozási követelményeket határoz meg attribútum-tanúsítók (attribute authority, AA) számára, amelyek a minősített tanúsítványokat kibocsátó minősített hitelesítés-szolgáltatókéval egyenszilárdságú biztonságot jelentenek. [55]

Az ETSI 101 903 (XAdES) az XML aláírások formátumát határozza meg. A dokumentum leírja, hogy az XML aláírásban hogyan lehet attribútum-tanúsítványt szerepeltetni. Az attribútum-tanúsítvány az aláírt elemek közé kerül, így aláírásával az alany megerősíti, hogy éppen melyik szerepkörében, melyik attribútuma szerint ír alá. Így az aláíráshoz csatolt, az aláíró által is aláírt – esetleg harmadik féltől származó – igazolás bizonyítja az aláírónak azt a szerepkörét, jogosultságát (attribútumát), amelyet az aláírás idejében használt. [51]

11.2.2.2. Általános attribútum-tanúsító

Az ETSI dokumentumai – elsősorban az ETSI TS 102 158 – a hitelesítés-szolgáltatáshoz hasonló, általános attribútum-tanúsítás szolgáltatást képzelnek el, ahol az attribútum-tanúsító olyan megbízható harmadik fél (TTP, trusted third party), amely az egyes felhasználók attribútumaira a hitelesítés-szolgáltatókhoz hasonló módon ad ki igazolást. A TTP-jellegű, általános attribútum-tanúsító többféle – akár tetszőleges – attribútumot tanúsíthat, és az érintett felek a minősített tanúsítványban igazolt állításhoz hasonlóan „komolyan veszik” az attribútum-tanúsító állításait, és megbíznak abban. [55]

Mivel az általános attribútum-tanúsító bármilyen attribútumot tanúsíthat, legalább olyan szintű biztonsággal kell működnie, amely a legérzékenyebb attribútum esetén szükséges. Ez magyarázza az ETSI TS 102 158 erős követelményeit. Ugyanakkor számos attribútum esetén közel sincs szükség ilyen erős bizonyítékokra. Annyira, hogy az ETSI TR 102 044 ún. „claimed role”-t is megenged, ahol valaki saját maga számára állít ki attribútum-tanúsítványt – saját

attribútumairól nyilatkozik a korábban ismertetett gondolatmenet szerint. (Lásd: 11.1.3. fejezet.) Bizonyos attribútumok esetén vagy bizonyos helyzetekben ilyen szintű garancia is elegendő. Így – álláspontunk szerint – egyes esetekben túlzó követelményeknek számítanak az ETSI TR 102 158 által megfogalmazott előírások.

A specifikációkban elkülönül, hogy ki rendelkezik a felhasználók attribútumairól (AGA, attribute granting authority vagy AIA, attribute issuing authority), illetve ki az attribútum-tanúsítványt kibocsátója, aláírója (AA, attribute authority).

Ha AA és AGA elkülönülnek egymástól, akkor AA pontosan olyan helyzetben van, mint egy hitelesítés-szolgáltató: harmadik féltől származó igazolásokra támaszkodva kell igazolnia az attribútumokat.

Egy AA – a hitelesítés-szolgáltatók esetén elfogadott megoldáshoz hasonlóan – több AGA-val is kapcsolatban lehet.

11.2.3. Milyen joghatással rendelkezik az attribútum-tanúsítvány?

A hazai jogszabályok nem definiálják az „attribútum-tanúsítvány" fogalmát, így az attribútum-tanúsítvány az elektronikus aláírásról szóló törvény (Eat.) szerint értelmezhető. Az Eat. a következő módon definiálja a „tanúsítvány" fogalmát:

> *Eat., 2. § „ 21. Tanúsítvány: a hitelesítés-szolgáltató által kibocsátott igazolás, amely az aláírás-ellenőrző adatot a 9. § (3), illetőleg (4) bekezdése szerint egy meghatározott személyhez kapcsolja, és igazolja e személy személyazonosságát vagy valamely más tény fennállását, ideértve a hatósági (hivatali) jelleget. "*

A fent ismertetett ajánlások szerinti attribútum-tanúsítvány nem fér bele az Eat. szerinti tanúsítvány fogalmába, hiszen nem tartalmazza az aláírás-ellenőrző adatot, így nem kapcsolja azt meghatározott személyhez sem. Az attribútum-tanúsítványokat merőben másképp kell kibocsátani vagy felhasználni, mint az Eat. szerinti nyilvános kulcsú aláírói tanúsítványokat. Az attribútum-tanúsítvány önmagában – egy az Eat. szerinti nyilvános kulcsú aláírói tanúsítvány nélkül – nem is használható.

Megmutattuk, hogy az attribútum-tanúsítványt nem hitelesítés-szolgáltató, hanem aláírásra képes végfelhasználó bocsátja ki. Ebből következik, hogy az Eat. értelmében az attribútum-tanúsítvány mindössze egy elektronikus aláírással ellátott elektronikus dokumentum, amelyen egy végfelhasználó – az Eat. fogalmai szerint egy *aláíró* – helyez el fokozott biztonságú vagy minősített elektronikus aláírást. Ez a dokumentum egy szabványos, géppel is értelmezhető formátumú igazolás, amely az igazolást kérő személy szerepkörét, jogosultságát, illetve tulajdonságát igazolja.

Az attribútum-tanúsítványon elektronikus aláírás van, így – fokozott biztonságú elektronikus aláírás esetén – egyenértékű lehet egy írásba foglalt igazolással, amely például a következőket

tartalmazza:

> ***11.3. Példa:*** *„Alulírott Gipsz Jakab (szig. szám: XY123456), a Kókler Kft. ügyvezetője meghatalmazom Zebulont, az 012345...6789 SHA-256 lenyomatú tanúsítvány alanyát, aki a Kókler Kft. munkatársa, hogy a Kókler Kft. címére érkező küldeményeket átvegye. Jelen meghatalmazás 2008 január 7. és 2008. január 8. között érvényes.*
>
> *(dátum, aláírás)"*

Így egy attribútum-tanúsítvány a fenti meghatalmazással azonos joghatással rendelkezik.

Kérdés: *Ki jogosult attribútum-tanúsítványt kiállítani, és milyen attribútumokat jogosult valaki igazolni?*

A fenti papír alapú igazolással egyenértékű attribútum-tanúsítványt az állíthat ki, és olyan attribútumokról, amelyről a fentihez hasonló igazolást egyébként papíron is jogában áll kiadni.

Összefoglalva:

- A jogszabályok külön nem definiálják az attribútum-tanúsítvány fogalmát, így az attribútum-tanúsítvány egyszerűen csak egy aláírt dokumentumnak minősül.

- *Semmi akadálya annak, hogy valaki attribútum-tanúsítvány formájában állítson ki igazolást olyan tényről, amelyről papír alapú igazolást is kiadhat.* Ez esetben AA és AGA megegyezik. Az attribútum-tanúsítvány szabványos, számítógéppel is értelmezhető, ellenőrizhető szerkezettel rendelkező, írásba foglalt igazolás.

- *Általános célú, TTP-jellegű, az ETSI specifikációkban elképzelt attribútum-tanúsító megjelenése ugyanakkor nehezen képzelhető el Magyarországon* a jelen jogszabályi környezetben.

  Ennek oka, hogy ha AA és AGA nem esik egybe, akkor az AA által aláírt igazolásokat, attribútum-tanúsítványokat nem lehet megbízható módon ellenőrizni – legalábbis nem egyszerű[8] AGA-ra visszavezetni, következésképpen az attribútum-tanúsítványokban feltüntetett szerepköröket, jogosultságokat sem könnyű elfogadni.

[8] Az ellenőrzési eljárásban azt kellene bizonyítani, hogy AA jogosult egy adott attribútummal kapcsolatban igazolást kiállítani. Ez történhetne ugyan egy AGA által AA számára kiállított attribútum-tanúsítvány alapján is, de AA szerepe éppen az volna, hogy AGA-nak ne kelljen attribútum-tanúsítványokkal foglalkoznia.

11.3. Modell az attribútum-tanúsítványok felhasználására

Az attribútum-tanúsítványok használatához nyilvános kulcsú aláírói tanúsítványokra is szükség van. Ezért, amíg csak kevesen használnak aláíró tanúsítványokat, addig az attribútum-tanúsítványok sem fognak ugrásszerűen elterjedni. Ugyanakkor ha az új nyilvános kulcsú tanúsítványra épülő rendszereket attribútum-tanúsítványok nélkül építik ki, az gyakran oda vezet, hogy a felhasználók szerepköreit, jogosultságait, tulajdonságait a nyilvános kulcsú (aláíró) tanúsítványukban helyezik el. Így problémássá válik, ha az adott nyilvános kulcsú (aláírói) tanúsítványokat egy másik rendszerben is használni szeretnék, és gyakran az a megoldás merül fel, hogy a másik rendszerben nem szabad ugyanazt a tanúsítványt használni, hanem új tanúsítványra van szükség. (A tapasztalat azt mutatja, e probléma leginkább az elektronikus aláírással kapcsolatban leglelkesebb felhasználóknál merül fel, és jelentősen csökkenti e felhasználók lelkesedését.) Álláspontunk szerint ez a rossz tendencia jelentősen gátolja a nyilvános kulcsú infrastruktúra terjedését.

Az alábbiakban egy olyan, attribútum-tanúsítványokra épülő módszert mutatunk be, amely viszonylag egyszerűen bevezethető, és egyúttal elválasztja a nyilvános kulcsú tanúsítványokat az attribútumoktól. A körvonalazott modellben több, független hitelesítés-szolgáltató, és több, független attribútum-tanúsító (AA) működik. Modellünk az attribútum-tanúsítványokra vonatkozó specifikációk szűkítéséből, pontosításából, illetve a hazai helyzethez történő illesztéséből származik. Az itt leírt modellen kívül természetesen számos más módon is alkalmazhatnánk attribútum-tanúsítványokat, csak egy lehetséges megoldást mutatunk be.

A körvonalazott rendszerben hitelesítés-szolgáltatók adják ki a végfelhasználók (köztük az attribútum-tanúsítók) aláírói tanúsítványait, az AA-k pedig a végfelhasználók aláírói tanúsítványaihoz kapcsolódó szerepkör, jogosultság igazolásokat (attribútum-tanúsítványokat) bocsátanak ki.

Azok a szervezetek működnek attribútum-tanúsítóként, amelyek jogosultak a végfelhasználók attribútumainak igazolására, és ismerik az adott attribútummal rendelkező végfelhasználók valamelyik nyilvános kulcsú aláírói tanúsítványát (illetve annak DN-jét vagy kriptográfiai lenyomatát).

Az egyes attribútum-tanúsítók kizárólag saját hatáskörükben bocsátanak ki attribútum-tanúsítványokat, és kizárólag olyan attribútumokat igazolnak, amelyekkel kapcsolatban papíron is kiadhatnak igazolásokat.

Ha egy végfelhasználó valamelyik szerepkörében szeretné a nyilvános kulcsú aláírói tanúsítványát használni, akkor az adott szerepkörének igazolására jogosult attribútum-tanúsítóhoz fordul. Az attribútum-tanúsító online bocsátja ki az attribútum-tanúsítványt.

Az attribútum-tanúsítvány olyan „rövid” élettartammal rendelkezik, hogy nem kell vizsgálni a visszavonási állapotát. Ha egy felhasználótól megvonnak egy attribútumot, akkor az attribútum-tanúsító a visszavonást követően már nem ad ki tanúsítványt. (Lásd: 11.3.4.4.

fejezet.)

Ha egy felhasználó nyilvános kulcsú aláírói tanúsítványához tartozó magánkulcsa kompromittálódik, a hitelesítés-szolgáltató visszavonja az aláírói tanúsítványt, így a nyilvános kulcsú aláírói tanúsítványhoz tartozó attribútum-tanúsítványok sem használhatóak többé.

11.3.1. A modell jellemzői

1. A végfelhasználóknak elég egyetlen nyilvános kulcsú aláírói tanúsítványt vásárolni, azt több célra is felhasználhatják.

2. A rendszer elosztott, decentralizált. A jogosultságok, szerepkörök (attribútumok) kezelését nem hitelesítés-szolgáltató végzi, hanem a szerepkört, jogosultságot kezelő szervezet.

3. A rendszer nyitott, könnyű hozzá csatlakozni. Az attribútum-tanúsítványok érvényességét könnyű ellenőrizni.

4. A rendszer nem fogalmaz meg felesleges követelményeket az attribútum-tanúsítókra. Az igazolt attribútumok jellege, érzékenysége határozza meg, hogy az adott attribútum esetén milyen biztonsági követelmények indokoltak[9].

5. A rendszer skálázható, sok hitelesítés-szolgáltató, sok attribútum-tanúsító, sok felhasználó és sok attribútum esetén is működőképes.

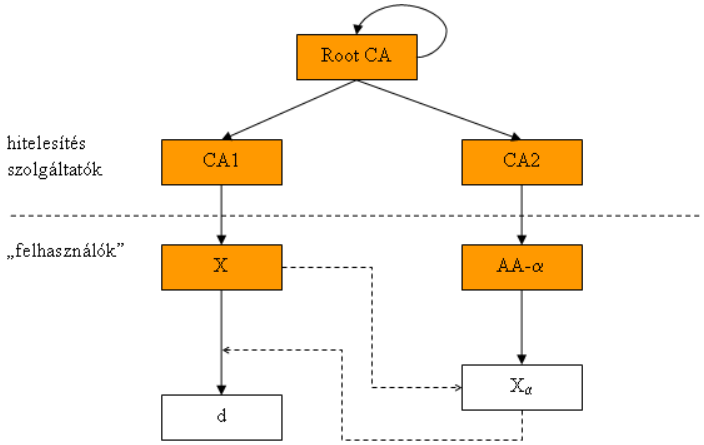
6. A rendszer elveiben hasonlít a jelenleg működő, papír alapú ügyvitelhez.

7. A jelenlegi jogszabályi környezetnek megfelel.

8. A rendszer szereplői csak olyan információkhoz jutnak hozzá, amelyek kezelésére jogosultak. Az érintett fél csak olyan ismeretekhez jut az aláíró tulajdonságaival, attribútumaival kapcsolatban, amelyeket közvetlenül az aláíró juttat el neki.

11.3.2. Végfelhasználók: aláírás létrehozása, ellenőrzése

A következőkben azt részletezzük, hogy hogyan használhatóak az attribútum-tanúsítványok elektronikus aláírás létrehozására alkalmas nyilvános kulcsú aláírói tanúsítványok mellett. Más célra szolgáló nyilvános kulcsú tanúsítványok esetében is ehhez hasonló megoldások alkalmazhatók.



11.2. ábra. **Az X felhasználó aláírja a d dokumentumot. Ez az aláírás tartalmazza az X felhasználó tanúsítványához kapcsolódó (annak lenyomatát tartalmazó) X_α attribútum-tanúsítványt. Az attribútum-tanúsítványban $AA-\alpha$ igazolja, hogy az X felhasználó rendelkezik az α szerepkörrel.**

11.3.2.1. Aláírás létrehozása

Tegyük fel, hogy X felhasználó az α szerepkörével szeretne aláírni egy dokumentumot. X rendelkezik egy nyilvános kulcsú aláírói tanúsítvánnyal, és AA-α (az α szerepkör igazolására jogosult attribútum szolgáltató) ismeri X ezen nyilvános kulcsú tanúsítványát.

Az eljárás lépései a következők:

1. X felhasználó üzenetet küld AA-α-nak, amelyben igazolást kér arról, hogy ő rendelkezik α szerepkörrel.
2. AA-α kiállítja az X_α igazolást (attribútum-tanúsítvány). Nyilvántartásában megnézni X tanúsítványát, és annak lenyomatát beleírja az X_α igazolásba. Az attribútum-tanúsítványt rövid távra (kb. 10 percre) állítja ki.
3. AA-α elküldi X_α-t X-nek.
4. Amikor X aláírja a d dokumentumot, egyúttal aláírja saját X aláírói tanúsítványát, és az X_α attribútum-tanúsítványt is. (Lásd: 11.2. ábra.)
5. X időbélyeget helyeztet el az elkészült aláíráson.

11.4. Példa: *Dr. Gipsz Jakab ügyvédként szeretne ellenjegyezni egy dokumentumot, ehhez igazolnia kell, hogy ő ügyvéd. A Magyar Ügyvédi Kamarától ezért igazolást (attribútum-tanúsítványt) kér arról, hogy ő valóban ügyvéd, majd ezt az igazolást csatolja a XAdES aláírásához.*

[9]Túl szigorú követelmények általános előírása költségessé teszi az attribútum-tanúsítványok elterjedt használatát.

Megjegyzés:

- *AA*-α-nak nem kell vizsgálnia X aláírói tanúsítványának érvényességét. Visszavont, lejárt tanúsítványhoz is kiállíthat X_α attribútum-tanúsítványt, mert az attribútum-tanúsítvány (igazolás) érvényes aláírói tanúsítvány nélkül amúgy sem használható. [56]
- *AA*-α bárkinek elküldheti az X_α attribútum-tanúsítványt, mivel azt úgyis csak az X tanúsítványhoz tartozó magánkulccsal lehet használni. *AA*-α egyébként kérheti a kérelmezőket arra, hogy azonosítsák magukat, mert:
  - nem szeretné, hogy akárki lekérdezze, hogy kik rendelkeznek α szerepkörrel,
  - védekezni kíván a szolgáltatás leterhelésére irányuló támadások ellen,
  - díjazáshoz kívánja kötni az attribútum-tanúsítványok kiállítását.
- Ha valakinek fontos, hogy különböző szerepkörben elkészített aláírásairól ne lehessen megállapítani, hogy ugyanaz a személy hozta őket létre, ezt az általunk körvonalazott modellben is meg tudja valósítani. Az X aláírói tanúsítvány alanya rendelkezhet további X' vagy X'' stb. aláírói tanúsítványokkal is, esetleg másik hitelesítés-szolgáltatótól is beszerezheti azokat. Ezek az aláírói tanúsítványok más lenyomattal rendelkeznek, így az X_α attribútum-tanúsítvány nem használható velük. Ha X nem szeretné a δ szerepkörben ugyanazt a kártyát vagy magánkulcsot használni, beszerezhet egy X' aláírói tanúsítványt, és ehhez kérheti az X'_δ attribútum-tanúsítványt. Ekkor X aláírói tanúsítványát (és a hozzá tartozó magánkulcsot) α szerepkörben, X' aláírói tanúsítványát δ szerepkörben használhatja. Aki nem tudja, hogy X és X' aláírói tanúsítványok alanya megegyezik, nem tudja megállapítani, hogy X_α és X'_δ attribútum-tanúsítványok ugyanahhoz a személyhez tartoznak. Így nem tudja megállapítani, hogy e személy α és δ szerepkörrel is rendelkezik, és így nem tudja összekapcsolni az α és a δ szerepkörben létrehozott aláírásait sem.

11.3.2.2. Egy érintett fél ellenőrzi az aláírást

Tegyük fel, hogy az Y érintett fél megkapja a fent létrehozott d dokumentumot, és ellenőrizni szeretné az aláírást, valamint meg szeretné állapítani, hogy az aláíró rendelkezik-e az α szerepkörrel. A következőket kell tennie:

1. Ellenőrzi az elektronikus aláírást a megfelelő aláírási szabályzat (6.8. fejezet) szerint, így például[10]:

[10] Az aláírások ellenőrzésének kérdéskörét az elektronikus aláírásról szóló fejezetben mutatjuk be részletesen.

- ellenőrzi, hogy az aláírás az X felhasználó magánkulcsával készült-e,
- meghatározza az ellenőrzéshez szükséges megbízható időpontot,
- felépíti a tanúsítványláncot egy megbízható gyökértanúsítványig ($Root \rightarrow CA1 \rightarrow X$), és
- ellenőrzi a lánc elemeinek visszavonási állapotát a megbízható időpontra vonatkozóan.

2. Megvizsgálja, hogy az aláírt adatok között szerepel-e attribútum-tanúsítvány, és megtalálja X_α-t.

3. Ellenőrzi, hogy AA-α aláírása érvényes-e X_α-n. Ezt az aláírást ahhoz hasonlóan – ugyanazon aláírási szabályzat keretében – ellenőrzi, mint X aláírását, így várhatóan ekkor is:

   - ellenőrzi, hogy az X_α attribútum-tanúsítványon lévő aláírás AA-α magánkulcsával készült-e,
   - meghatározza az ellenőrzéshez szükséges megbízható időpontot (mivel egyazon időbélyeg védi X aláírását és az attribútum-tanúsítványt, valószínűleg az X aláírása ellenőrzéséhez is használt megbízható időpontot választja majd);
   - felépíti a tanúsítványláncot az X_α attribútum-tanúsítványtól a gyökérig ($Root \rightarrow CA2 \rightarrow AA-\alpha$).

4. Az X_α attribútum-tanúsítvány „rövid lejáratú”, így nem kell ellenőriznie, hogy AA-α visszavonta-e X_α-t.

5. Ellenőrzi, hogy X_α érvényes volt-e (nem járt-e már le) az aláírás létrehozásának pillanatában.

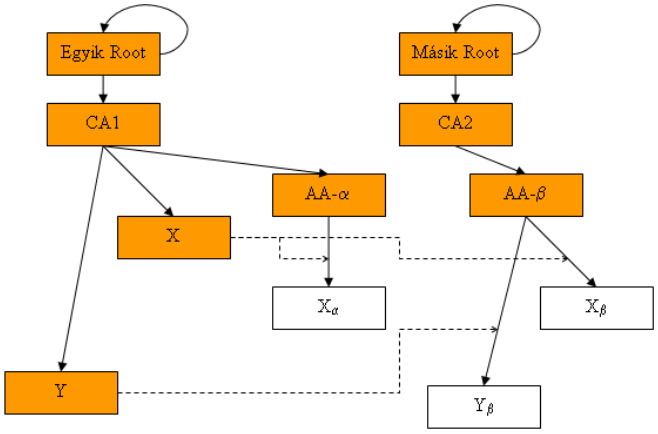
6. Ellenőrzi, hogy AA-α jogosult-e az α szerepkör igazolására

11.3.3. Elosztott rendszer

Amikor több szerepkör is megjelenik a rendszerben (lásd: 11.3. ábra), a felhasználóknak (alanyoknak) elég egyetlen tanúsítvánnyal rendelkezniük, ezt használhatják valamennyi szerepkörükben. Ugyanakkor aki kívánja, megteheti, hogy különböző szerepköreiben más és más tanúsítványt és magánkulcsot használ.

Az itt körvonalazott rendszerben az RFC 3281 szerinti „push” modellt használjuk az attribútum-tanúsítványok továbbítására.[144] E modell szerint az attribútum-tanúsítványok beszerzése az aláírói tanúsítvány alanyának feladata[11]. Szintén az aláírói tanúsítvány alanya

[11] A másik, ún. „pull” modell szerint az attribútum-tanúsítvány beszerzése az ellenőrző fél feladata. Ez ugyan egyszerűsíti az aláíró feladatát, de adatvédelmi kérdéseket vet fel; például milyen alapon fér hozzá az ellenőrző fél az attribútum szolgáltató nyilvántartásához.



11.3. ábra. **Az X felhasználó egyaránt rendelkezik α és β szerepkörrel, mindkét szerepkörében ugyanazt a tanúsítványt használja. Abból derül ki, hogy melyik szerepkörében ír alá, hogy X_α vagy X_β attribútum-tanúsítványát csatolja az aláírásához. Az Y felhasználó csak β szerepkörrel rendelkezik.**

kell, hogy eljuttassa az attribútum-tanúsítványt az aláírást ellenőrző érintett félnek. (Mindez úgy zajlik le, hogy az attribútum-tanúsítvány az aláírt elemek közé kerül.)

A „push” modell előnye, hogy az érintett félnek nem kell kapcsolatba kerülnie AA-α-val, nem kell ismernie AA-α elérhetőségét, és nem kell adatokat (attribútum-tanúsítványokat) lekérdeznie az egyes AA-któl. Így az érintett fél kizárólag azon információkhoz jut hozzá, amelyeket az aláírói tanúsítvány alanya a rendelkezésére bocsát. Így nem jelentkeznek olyan problémák, hogy mely érintett fél jogosult lekérdezni az egyes felhasználók α, β, γ stb. attribútum-tanúsítványait, és így nem kell kezelni a felhasználók adott attribútumaival kapcsolatos adatokat.

11.3.4. Az attribútum-tanúsítók

11.3.4.1. Ki tanúsíthat attribútumot?

Az attribútum-tanúsítók olyan végfelhasználók, akik elektronikus aláírással látják el az online kibocsátott rövid élettartamú attribútum-tanúsítványokat[12]. Az attribútum-tanúsítványokon jellemzően automata segítségével helyeznek el *fokozott biztonságú elektronikus aláírást.*

Az itt körvonalazott modellben az, aki az attribútum kiosztására jogosult (AGA) és az, aki az attribútum-tanúsítványt kibocsátja (AA) ugyanaz a szereplő.

Bármelyik végfelhasználó felléphet attribútum-tanúsítóként, pontosan ugyanolyan jogosultságot, felhatalmazást (igazolást) adhat ki elektronikusan, mint amilyet papíron, kézzel írott aláírással is kiadhat. Aki elektronikusan olyan szerepköröket, jogosultságokat igazol, amelyekre nincs meg a megfelelő felhatalmazása, pontosan olyan következményekkel

[12] Az attribútum-tanúsítványok elektronikus aláírással aláírt igazolások.

számolhat, mintha azt papír alapon adta volna ki. Az ETSI TS 102 044[54] szellemében a végfelhasználó saját attribútumairól is nyilatkozhat „claimed role”-t tartalmazó attribútum-tanúsítvány formájában. Az attribútum-tanúsító egy kulcspárral több szerepkört is igazolhat, mert az attribútum-tanúsítványok kibocsátása pusztán dokumentumok aláírását jelenti, így elvileg bármilyen dokumentumot aláírhat saját aláírói magánkulcsával.

11.3.4.2. Attribútum-tanúsítási rend (ACP)

Érzékenyebb attribútumok esetén az attribútum-tanúsító *attribútum-tanúsítási rend*et fogalmazhat meg. Ez a rend pontosan leírja az attribútum-tanúsítvány kibocsátásának és felhasználásának feltételeit. Az ETSI TS 102 158 specifikáció ilyen rendet (rendeket) határoz meg, bár ez a specifikáció elsősorban az általános célú, TTP-jellegű attribútum-tanúsítóra vonatkozik, amely a minősített tanúsítványokkal egyenszilárdságú[13] biztonságot és garanciát nyújt az AGA-któl kapott információk alapján.

Elsősorban akkor van értelme attribútum-tanúsítási rendet megfogalmazni, ha az érintett félnek – a hitelesítési rendhez hasonlóan – a jogszabályok szerint is figyelembe kell azt vennie. Az attribútum-tanúsítási rendek használatát jó megoldásnak tartjuk, így használatukat annak ellenére javasoljuk, hogy a jelen jogszabályi környezetben még nem lehet őket a hitelesítési rendekhez hasonlóan alkalmazni, és az attribútum-tanúsítványban sem lehet őket szabványos módon jelölni.

11.3.4.3. Hogyan történik az attribútum-tanúsítvány kibocsátása?

AA-α nyilvántartásba veszi azoknak a felhasználóknak az aláírói tanúsítványát[14], akik rendelkeznek az α attribútummal. AA-α-nak csak a saját nyilvántartását kell megnéznie; azt kell ellenőriznie, hogy az X tanúsítvány alanyához valóban tartozik-e az α szerepkör. Ha igen, kiadhatja a tanúsítványt. Ha X már nem rendelkezik az α szerepkörrel, AA-α nem ad ki több X_α attribútum-tanúsítványt. Más teendője ezen kívül nincs[15].

11.3.4.4. Attribútum-tanúsítványok visszavonásának közzététele

Az itt bemutatott modell szerinti attribútum szolgáltató „rövid” lejáratú attribútum-tanúsítványokat ad ki. Az attribútum-tanúsítványok élettartama olyan rövid, hogy nem szükséges, illetve nincs értelme a rájuk vonatkozó attribútum-visszavonási listát közzétenni[16].

Ez szélsőséges esetben akár 10 percig érvényes attribútum-tanúsítványokat is jelenthet, de gyakran nincsen szükség ilyen rövid élettartamra. Lényeg, hogy az attribútum-tanúsítvány

[13] Nem minden attribútum esetén van szükség ilyen szintű garanciákra.
[14] A tanúsítvány lenyomatát is elegendő nyilvántartásba venni.
[15] AA-α-nak nem kell vizsgálnia a nyilvános kulcsú tanúsítvány lejártát, visszavonási állapotát. [56]
[16] OCSP válaszadó tanúsítványokkal kapcsolatban terjedt el ehhez hasonló gyakorlat (4.1.5.2. fejezet).

élettartama annál legyen rövidebb, mint amilyen gyakran az attribútum változik, azaz AGA dönt az attribútumról.

> ***11.5. Példa:*** *Tegyük fel, hogy a Zabhegyező Egyesület alapszabálya értelmében a tagok felvételét és kilépését az elnökség hagyja jóvá, amely havonta egyszer ülésezik. Így a hónap közepén elvileg sem szűnhet meg valakinek a tagsága, azaz a tagságot igazoló attribútum-tanúsítványt hónap közepén nem kellhet visszavonni. Így jó megoldást jelenthet, ha az egyesület hónap elején kibocsátja a tagok attribútum-tanúsítványait, és nem tesz közzé hozzájuk visszavonási állapotot. Nincs szükség technikai szempontból finomabb felbontásban kezelni az attribútumot, mint ahogy az a valóságban történik.*
>
> *Ehhez hasonló gondolatmenet szerint, az élethosszig tartó „nyugdíjas" attribútumot igazoló attribútum-tanúsítvány elvileg nagyon hosszú érvényességgel is kiadható.*

Jelen dokumentumban a 10 perces attribútum-tanúsítvány élettartamot javasoljuk, de egyes esetekben ennél hosszabb élettartam is elég rövid lehet. Például ha egy AGA-ω működéséből adódóan csak hetente egyszer születhet olyan döntés, amely az ω attribútumot személyekhez rendeli (és esetleg megfoszt valakit az ω attribútumától), akkor nem sok értelme van egy hétnél rövidebb (pl. 1 órás) élettartamú attribútum-tanúsítványokat kibocsátani. AA-α felelőssége olyan rövid lejáratot meghatározni, amely az adott attribútummal kapcsolatban minimalizálja a visszaélés kockázatát, egyúttal használható szolgáltatást eredményez. AA-α az RFC 3281 4.3.6. fejezetében leírt, szabványos módon, a „No Revocation Available" kiterjesztéssel jelöli, hogy az adott attribútum-tanúsítvánnyal kapcsolatban nincs elérhető visszavonási információ. [144]

11.3.4.5. Biztonság

Egy attribútum-tanúsítónak olyan fizikai, logikai és szabályozási biztonságot kell megvalósítania, amely az adott attribútumhoz indokolt. Például valamely cég képviseletére vonatkozó jogosultság érzékeny attribútum, ahol erős biztonsági követelmények alkalmazása is indokolt. Ezzel szemben egy evezősklub-tagság kevésbé érzékeny attribútum, ahol felesleges szigorú követelményeket támasztani.

AA-α (vagy AGA-α) dönt arról, hogy az α attribútum mennyire érzékeny, és hogyan állapítja azt meg, hogy az X felhasználó valóban rendelkezik-e a kérdéses attribútummal. Ő dönti el, hogy milyen adatbázisok adataira támaszkodik, találkozik-e személyesen X-szel, és miként bizonyosodik meg arról, hogy X valóban birtokában van az aláírói tanúsítványához tartozó magánkulcsnak. AA-α felel az általa kibocsátott igazolásokért is.

Az attribútum-tanúsító végfelhasználók sok helyen a magánkulcsot – a hitelesítés-szolgáltatóknál megszokotthoz képest – gyenge biztonsági környezetben őrzik, így

mindenképpen biztosítani kell az attribútum-tanúsítók aláírói tanúsítványainak felfüggesztési, illetve visszavonási lehetőségét.

Nagyobb szervezetek várhatóan szabályozni fogják, hogy az általuk tanúsítható attribútumokat milyen módon lehet tanúsítani, és saját szervezetükön belül követelményeket fognak megfogalmazni attribútum-tanúsítóik működésére.

11.3.4.6. Felelősség

AA-α felelős az általa aláírt attribútum-tanúsítványokért, mint ahogy az X felhasználó is felelős az általa aláírt dokumentumokért, nyilatkozatokért.

11.3.5. Az attribútum-tanúsítvány felépítése

Az alábbiakban az írjuk le, hogy az előbbiekben felvázolt rendszerben szereplő attribútum-tanúsítványokban hogyan használjuk az RFC 3281 által meghatározott mezőket.

11.3.5.1. Kötelező mezők

- `version`: Ide a „v2” érték kerül az RFC 3281 szerint. [144]
- `holder`: Az attribútum-tanúsítványhoz kapcsolódó nyilvános kulcsú tanúsítvány alanyára a korábban (11.2.1. fejezet) leírt gondolatmenet szerint `objectDigestInfo` segítségével hivatkozunk. Itt a nyilvános kulcsú tanúsítványnak (publicKeyCert) az SHA-256 algoritmussal képzett lenyomata szerepel.
- `issuer`: Az attribútum-tanúsítványt aláíró attribútum-tanúsító megkülönböztetett neve (DN), pontosan úgy, ahogy az az attribútum-tanúsító nyilvános kulcsú aláírói tanúsítványában szerepel. A két DN azonosságát az RFC 5280 szerint kell vizsgálni. [152]
- `signature` és *signatureAlgorithm*: Az „sha256withRSA” algoritmus.
- `serialNumber`: A kibocsátó attribútum-tanúsító által meghatározott egyedi sorozatszám. A sorozatszám az adott attribútum-tanúsító kulcsának kontextusában kell, hogy egyedi legyen.
- `attrCertValidityPeriod`: „Rövid” élettartam. Az korábban leírt gondolatmenet szerint az élettartam olyan rövid kell, hogy legyen, hogy ne legyen szükség visszavonási állapotot ellenőrizni (11.3.4.4. fejezet).
- `attributes`: Egy vagy több attribútum, lásd: 11.3.5.3. fejezet.
- `issuerUniqueID`: Nincs kitöltve, mert ez felesleges megkötést tenne az attribútum-tanúsító nyilvános kulcsú tanúsítványára.

11.3.5.2. Kiterjesztések

- `Audit Identity`: Opcionális, tartalmát az attribútum-tanúsító határozza meg.
- `AC Targeting`: Nincs kitöltve.
- `Authority Key Identifier`: Kitöltve, megegyezik az attribútum-tanúsító nyilvános kulcsú tanúsítványában lévő Subject Key Identifier értékkel. Mind az RFC 5280, mind az RFC 3281 ajánlja ezen mező használatát.
- `Authority Information Access` (Hozzáférés a kiállítói információkhoz): Az RFC 5280 szerinti hivatkozást tartalmaz az attribútum-tanúsító nyilvános kulcsú aláírói tanúsítványára (3.4. fejezet). Más hivatkozás (pl. OCSP) nem kerül ide.
- `CRL Distribution Points` (CRL elérési helyei): Nincs kitöltve, mert modellünkben nem kell ellenőrizni az attribútum-tanúsítványok visszavonási állapotát.
- `No Revocation Available`: Ez az állítás szerepel az attribútum-tanúsítványban.

11.3.5.3. Hogyan jelenik meg az attribútum az attribútum-tanúsítványban?

Az attribútumokat az X.509, az ETSI TS 102 044 és az RFC 3281 által meghatározott szintaxis szerint tüntethetjük fel az attribútum-tanúsítványban. Az attribútumokat célszerű olyan módon jelölni, hogy azokat automaták segítségével is fel lehessen dolgozni. Így célszerű őket URI vagy OID segítségével jelölni. Ugyanakkor előnyös, ha az attribútumok szövegesen is megjelennek az attribútum-tanúsítványban, így akkor is fel lehet dolgozni azokat, ha a feldolgozó által használt alkalmazás nem tudja értelmezni az adott URI-t vagy OID-t.

Azt javasoljuk, hogy az attribútum-tanúsítvány mindkét módon tartalmazza az attribútumot. Ilyenkor az attribútum-tanúsító felelőssége, hogy a szöveges és az automaták által feldolgozható megjelölés konzisztens legyen, és ne lehessen az attribútumot kétféleképpen értelmezni. A következő megoldást javasoljuk:

- Az attribútumok megnevezésként, azaz DN-ként szerepeljenek az attribútum-tanúsítványban.
- A DN `title` eleme tartalmazza az attribútumot szövegesen, magyarul. Emberi felhasználó e mező alapján tudja értelmezni az attribútum-tanúsítványt.
- A DN `description` eleme egy URI-t tartalmaz, amely az attribútum egyedi azonosítója. Ha géppel dolgozzuk fel az attribútum-tanúsítványt, e mezőt célszerű vizsgálni. Előfordulhat, hogy az attribútum `title` mezőben szereplő elnevezése változik: például átnevezésre kerülhet, de az is lehet, hogy csak más írásmódra térünk át valami miatt. Ha az attribútum érdemben ugyanaz, az URI-ja nem változik.

A hivatkozott URI-n elhelyezhető szöveges magyarázat az attribútum pontos jelentéséről, akár több nyelven is. Az URI-n található információ nem kerül aláírásra, csak az URI maga.

- Ha az attribútumhoz kapcsolódik valamilyen azonosító szám, az a `serialNumber` elembe kerül. Így ide kerül, hogy pontosan mire vonatkozik az attribútum.

  11.6. Példa: *Ha az attribútum egy cég képviseletét jeleni, a `serialNumber` mezőbe kerül a cég cégjegyzékszáma.*

- Ha az attribútumhoz valamilyen érték kapcsolódik, akkor az a `commonName` elembe kerül.

  11.7. Példa: *Ha az attribútum azt igazolja, hogy valakinek az anyja neve vagy születési helye ez és ez, akkor ide kerülhet ez az információ.*

- A DN további információkat is tartalmazhat, de ezek érdemben nem módosíthatják az attribútum jelentést, a fentieket nem írhatják felül, csak kiegészítő információként szolgálnak.

Megjegyezzük, jelenleg nincsen egységes nomenklatúra az attribútumok elnevezésére, sem EU-s, sem hazai, sem egyéb szinten. Úgy véljük, ilyen – belátható időn belül – nem is jöhet létre, mert az attribútumok világunkban annyira sokfélék, annyira gyorsan változnak, és teljesen reménytelen (és értelmetlen) lenne az összes lehetséges attribútumot egyetlen helyen felsorolni. Ha egy szervezet megadott attribútumokat igazolni kíván, például a fenti módszert követheti.

11.3.6. Hogyan szerzi be a felhasználó az attribútum-tanúsítványt?

Az RFC 3281 szerinti „push” modellt javasoljuk az attribútum-tanúsítványok beszerzésére, ami azt jelenti, hogy az attribútum-tanúsítványt a felhasználó, azaz a nyilvános kulcsú tanúsítvány alanya szerzi be, és juttatja el az attribútum-tanúsítványt ellenőrző érintett félhez. Ennek az a kellemes következménye, hogy az attribútum-tanúsítvány beszerzésének módja az alany és a attribútum-tanúsító „magánügye”, azaz a megoldásnak nem szükséges széles körben összehangoltan működnie[17]. Az egyes rendszerekben olyan megoldások használhatóak, amelyek ott könnyen megvalósíthatóak és jól használhatóak.

A továbbiakban egy javaslatot fogalmazunk meg.

11.3.6.1. Az attribútum-tanúsító címe

Az alany tudja, hogy milyen attribútumokkal rendelkezik, és minden α attribútumához ismeri azt az URL-t, amelyről $AA-\alpha$ attribútum-tanúsítványt kérhet.

[17] Míg a „pull” modellben specifikálni kellene, hogy az érintett fél hogyan tudja meg az attribútum-tanúsító címét, hogyan igazolja jogosultságát az attribútum-tanúsítvány lekérdezésére stb.

11.3.6.2. Protokoll az attribútum-tanúsítvány beszerzésére

- Az attribútum-tanúsítványt HTTPS kapcsolaton keresztül kell letölteni. A HTTPS szervere SSL tanúsítvány segítségével azonosítja magát, amit a kliens ellenőriz.
- AA dönti el, hogy a kliensnek kell-e azonosítani magát. AA vagy tanúsítvány-alapú, vagy basic (felhasználónév és jelszó alapú) autentikációt írhat elő a kliens számára.
- A kliens elküldi az attribútum-tanúsítvány kérelmet AA-nak. A kérelemnek kell tartalmaznia az alany nyilvános kulcsú tanúsítványának lenyomatát, a használt lenyomatképzési algoritmus megnevezését, és annak az attribútumnak vagy attribútumoknak a megnevezését is, amelyekre attribútum-tanúsítványt kér.
- AA megvizsgálja, hogy a kliens rendelkezik-e a kívánt attribútumokkal, és AA jogosult-e ezeknek az attribútumoknak a tanúsítására. Ha igen, AA elkészíti a rövid lejáratú attribútum-tanúsítványt.
- AA elküldi az attribútum-tanúsítványt a kliensnek a HTTPS kapcsolaton keresztül.
- A kliens ellenőrzi az attribútum-tanúsítványt, és ellenőrzi, hogy valóban azok az attribútumok szerepelnek-e benne, amelyeket kért.
- Ezt az eljárást a felhasználó aláírás-létrehozó alkalmazása végzi el, az emberi felhasználó mindebből annyit érzékel, hogy egy listából ki kell választania, hogy éppen milyen szerepkörben szeretne aláírni.

11.3.6.3. Az attribútum-tanúsítvány kérelem formátuma

Az „Attribute Certificate Request Message Format” PKIX draft specifikáció leír egy lehetőséget az attribútum-tanúsítványok formátumára. E formátum analóg a PKCS#10 tanúsítványkérelem formátummal: az attribútum-tanúsítványba kért adatokat tartalmazza, az attribútum-tanúsítvány szintaxisának megfelelően. Az alábbiakban egy erre épülő attribútum-tanúsítvány kérelmet írunk le. [137]

Modellünkben egyszerre egyetlen attribútum-tanúsítványt lehet lekérdezni (amely több attribútumot is tartalmazhat), így a fenti specifikáció szerinti `AttrCertReqMessages` egyetlen `AttrCertReqMsg` elemet tartalmaz. Az `AttrCertReqMsg` kizárólag `AttrCertRequest` elemet tartalmaz. Ezen kívül nem használjuk sem a `regInfo`, sem a `controls` elemeket. Az `AttrCertTemplate` mezőben a kérelmező lényegében az attribútum-tanúsítvány mezőit tölti ki. Jelen dokumentumban azt a gondolatmenetet követjük, hogy a kérelmező csak a rá vonatkozó, a modellben is felhasznált mezőket adja meg, a többi kitöltése AA feladata.

- `version`: „v2”

- `holder`: Az attribútum-tanúsítványba kerülő `holder` érték kerül ide, azaz az `objectDigestInfo`, amely a nyilvános kulcsú aláírói tanúsítvány lenyomatát tartalmazza.
- `issuer`: Nincs kitöltve, AA határozza majd meg.
- `signature`: Nincs kitöltve, AA határozza majd meg.
- `validityPeriod`: Nincs kitöltve, AA határozza majd meg.
- `attributes`: A kérelmező kitölti, hogy mely attribútumokra vonatkozó attribútum-tanúsítványt kér (11.3.5.3. fejezet).

  Előfordulhat, hogy a kérelemben szereplő és az AA által visszaadott attribútum nem egyezik meg bitről bitre. A kérelemnek automaták által feldolgozható szerkezettel kell rendelkeznie, így például egy céghez kapcsolódó attribútumot nem a cég neve, hanem a cég cégjegyzékszáma alapján kell lekérdezni. Ugyanakkor jó, ha az attribútum-tanúsítvány szövegesen is tartalmazza a cég nevét. Azt javasoljuk, hogy AA a kérelem géppel is értelmezhető mezőit vegye figyelembe, ezek alapján ő maga határozza meg a szövegesen kitölthető mezőket.

  Például: A kérelemben szereplő attribútumban csak a cégjegyzékszám szerepel, míg a válaszban megjelenik mind a cégjegyzékszám, mind a cég neve.
- `extensions`: Nincs kitöltve, a kérelem nem tartalmaz kiterjesztéseket. A kiterjesztéseket AA határozza majd meg.

11.3.7. Hogyan ellenőrzi az érintett fél, hogy ki milyen attribútumot jogosult tanúsítani?

A korábban leírt gondolatmenet szerint az attribútum-tanúsító végfelhasználó (az Eat. terminológiája szerint „aláíró”), aki legalább fokozott biztonságú elektronikus aláírással lát el elektronikus dokumentumokat, és a dokumentumokban igazolásokat bocsát ki az egyes aláírói tanúsítványok alanyainak attribútumaival kapcsolatban. Jogi szempontból nincsen kitüntetett szerepe, műszaki szempontból pedig csak elektronikus aláírás létrehozására alkalmas tanúsítvánnyal kell rendelkeznie. Az RFC 3281 szerint a befogadónak közvetlenül meg kell bíznia AA tanúsítványában. Ez a gondolat szintén TTP-jellegű AA-t tételez fel.

> RFC 3281:
> `5. Attribute Certificate Validation`
> [...]
> `4. The AC issuer MUST be directly trusted as an AC issuer (by configuration or otherwise).`

Nem tudunk olyan szabványos lehetőségről, amely szerint egy aláírói tanúsítványban feltüntethetnénk, hogy az egy attribútum-tanúsító aláírói tanúsítványa. Amennyiben lenne is ilyen lehetőség, ez is legfeljebb általános célú, TTP-jellegű attribútum-tanúsító megjelölésére lenne alkalmas. Olyan modellben, ahol sok attribútum-tanúsító van jelen, de egy-egy tanúsító csak egy-egy szerepkört jogosult tanúsítani, azt kellene megjelölni az attribútum-tanúsító aláírói tanúsítványában, hogy az adott attribútum-tanúsító mely attribútumokkal kapcsolatban jogosult attribútumokat kibocsátani.

A fentiek alapján az attribútum-tanúsító tanúsítványát „közönséges" aláírói tanúsítványnak kell tekinteni, és az attribútum-tanúsító aláírását ennek megfelelően kell ellenőrizni. A következő két megoldást javasoljuk annak eldöntésére, hogy egy attribútum-tanúsító jogosult-e egy adott attribútumot tanúsítani:

- *Ha automata dönt, vagy a papír alapú rendszereknél jelentősen magasabb szintű biztonságot követelünk meg, akkor kizárólag az ismert tanúsítvánnyal rendelkező attribútum-tanúsítókat szabad elfogadni.*

  Ekkor jellemzően nagy mennyiségű aláírásról, attribútum-tanúsítványról kell dönteni, és az attribútum-tanúsítványok az attribútum-tanúsítók zárt halmazából származnak. Az RFC 3281 szerint közvetlenül meg kell bízni az attribútum-tanúsítókban, tehát nyilvántartást kell vezetni az egyes attribútumok tanúsítására jogosult attribútum-tanúsítók aláírói tanúsítványairól.

  Például a Cégtörvény szerint kizárólag ügyvédek, közjegyzők, és jogtanácsosok küldhetnek be a cégbíróságra elektronikus cégbejegyzési kérelmet. A cégbejegyzési kérelmeket befogadó automaták elég, ha e három attribútumot (szerepkört) ismerik, mert csak ezeket kell ellenőrizniük. Ha az ellenőrzés attribútum-tanúsítvány alapján történik, az automatáknak az AA-ügyvéd, AA-közjegyző és AA-jogtanácsos aláírói tanúsítványokat kell ismerniük. Így az „ügyvéd" szerepkör tanúsítását kizárólag az AA-ügyvéd attribútum-tanúsítótól fogadják el, stb.

- *Ha ember dönt, a papíron alkalmazott eljáráshoz hasonlóan kell eljárni, és az attribútum-tanúsító nevéből kell megállapítani, hogy az jogosult-e egy adott attribútum tanúsítására.*

  Papír alapú ügyintézés során az emberi ügyintézők – megfelelő képzés, vagy csak „józan ész" alapján – az igazolást kibocsátó neve alapján döntik el, hogy az jogosult-e egy adott igazolást kibocsátani. Amennyiben kétség merül fel, a papír alapú rendszerekben már kialakult szabályok vannak, hogy hogyan kell a megfelelő bizonyítékokat beszerezni.

  Például elfogadjuk, ha „Budapesti Műszaki és Gazdaságtudományi Egyetem" igazolja, hogy Gipsz Jakab „okleveles villamosmérnök". Azt is elfogadjuk, ha a „Fővárosi Cégbíróság" igazolja, hogy Gipsz Jakab jogosult a „Kókler Bt-t" képviselni. Ezzel szemben rögtön kétség merül fel, ha ezeket az igazolásokat „Nagy Zebulon" magánszemély bocsátja ki.

A megoldás előnyös tulajdonsága, hogy nem kell nyilvántartást vezetni az elfogadott attribútum-tanúsítók aláírói tanúsítványairól. Nem kell foglalkozni azzal sem, ha az attribútum-tanúsítók aláírói tanúsítványai megváltoznak, vagy új attribútum-tanúsító jelenik meg.

Egyedül az álneves tanúsítványok (amelyek nem az alany valódi nevét tartalmazzák) jelenthetnek problémát: álneves tanúsítvány esetén az attribútum-tanúsító megnevezéséből (DN) nem következtethetünk annak kilétére. Megoldásunkban ezért nem fogadhatunk el olyan attribútum-tanúsítókat, amelyek álneves tanúsítvánnyal rendelkeznek. Az Eat. szerint egyértelműen jelölni kell egy tanúsítványban, ha az álnevet tartalmaz, így e probléma kivédhető.

11.3.8. A hitelesítés-szolgáltatók szerepe a modellben

Az eddigiekben leírt modell nem érinti a hitelesítés-szolgáltatókat, nem támaszt követelményeket velük szemben. A hitelesítés-szolgáltatóknak továbbra is aláírói tanúsítványokat kell kibocsátani, ahogyan ezt most is teszik. Az attribútum-tanúsítványok terjedésével várhatóan csökken a nyilvános kulcsú aláírói tanúsítványokban szereplő attribútumok jelentősége, és e tanúsítványok az alany nevén kívül – hasonlóan a kézzel írott aláíráshoz – a jövőben mást már nem fognak tartalmazni[18].

Ha egy nyilvános kulcsú aláírói tanúsítványt több célra, több szerepkörben is fel lehet használni, akkor várhatóan többen találják majd gazdaságosnak a nyilvános kulcsú infrastruktúra használatát, így mindez elősegíti majd az elektronikus aláírásra szolgáló nyilvános kulcsú tanúsítványok terjedését. E modellben semmilyen megkötést nem támasztunk az aláírói tanúsítványokra nézve, mert lenyomat alapján hivatkozunk rájuk, így modellünkben valamennyi hitelesítés-szolgáltató tanúsítványa használható.

11.4. Összegzés

- Egy felhasználó szerepköreit, jogosultságait, tulajdonságait egységes elnevezéssel az ő *attribútumai*nak nevezzük.
- Ha az attribútum a (nyilvános kulcsú) tanúsítványban szerepel, az több szempontból is hátrányos:
  - a kezelése, változtatása nehézkes,
  - interoperabilitási problémákat okoz,
  - súlyos adatvédelmi problémákat vet fel,

[18] Legfeljebb néhány kötelező elemet, mint pl. az ország megjelölése, illetve az e-mail cím, amely a levelezőprogramok használatához szükséges.

  - gátolja, hogy egyazon nyilvános kulcsú tanúsítványt többféle célra is használni lehessen, így növeli a felhasználóra nehezedő költségeket, és ezzel gátolja a PKI terjedését.

- Az attribútum-tanúsítvány olyan szabványos formátumú, géppel is értelmezhető, aláírt igazolás, amely az attribútumokat egy nyilvános kulcsú tanúsítványhoz (vagy annak alanyához) kapcsolja.

- Az Eat. nem definiálja az attribútum-tanúsítvány fogalmát, így az attribútum-tanúsítvány pusztán egy aláírt dokumentumnak tekinthető.

- A nemzetközi specifikációkban szereplő, általános célú, TTP-jellegű attribútum-tanúsító nehezen képzelhető el a hazai jogszabályi környezetben.

- Javaslatot tettünk egy modellre, amelyben:

  - mindenki a saját maga által igazolható attribútumokról állít ki attribútum-tanúsítványokat,
  - a felhasználók fordulnak az attribútum-tanúsítókhoz, ők kérik le az attribútum-tanúsítványaikat, az adott attribútum-tanúsítónál működő módon,
  - elektronikus aláírás esetén az aláíró csatolja az attribútum-tanúsítványt az aláírandó dokumentumhoz, és az attribútum-tanúsítványt is aláírja,
  - az attribútum-tanúsítványok „rövid” lejáratúak, így nem szükséges ellenőrizni a visszavonási állapotukat.

- Célszerű attribútum-tanúsítványokkal igazolni a szerepköröket, jogosultságokat, mert ezzel elkerüljük, hogy olyan szigetmegoldást alakítsunk ki, amelyben a tanúsítványokat máshol, más rendszerben nem lehet felhasználni.

12. fejezet

A PKI gyakorlati alkalmazása

„In theory, theory and practice are the same. In practice, they are not."
(Az elmélet és a gyakorlat elméletileg megegyezik egymással. Gyakorlatilag nem.)

– *Albert Einstein*

Egy elmélet gyakorlatba való átültetése mindig sok problémát vet fel. Nem elég, ha az elektronikus aláírást pusztán matematikai, technológiai, jogi vagy folyamatszervezési szempontból közelítjük, a sikeres alkalmazáshoz ezen szempontokat együttesen kell figyelembe venni. Az eddigi fejezetekben láttuk, hogy az elektronikus aláírás és a PKI eszköztára korántsem egyszerű, de a valódi nehézségeket mégsem maguk a technológiai problémák, hanem a valós, emberi környezethez való illesztés szokta jelenteni. A technológiai problémákra számos jól bevált megoldás ismert, míg az elektronikus aláírás emberi kérdései még közel nincsenek olyan letisztult állapotban.

E fejezet első alfejezetben megvizsgáljuk, hogy az elektronikus világban milyen eszközök léteznek a hitelesség biztosítására, és ezek között az elektronikus aláírás hol helyezkedik el. Bár elektronikus aláírást használni látszólag sokkal körülményesebb és drágább, minta nem használnánk elektronikus aláírást, ezen összehasonlítás csalóka. Ha egy rendszerben lévő adatok hitelességét szeretnénk biztosítani, az elektronikus aláírás költségeit azzal kell összemérni, hogy mibe kerülne, ha a hitelességet más eszközökkel biztosítanánk. A második alfejezetben kiemelünk néhány pontot, amelyekre feltétlenül ügyelnünk kell, ha be valahol be szeretnénk vezetni az elektronikus aláírást. A harmadik alfejezetben az elektronikus aláírás néhány legnagyobb mai felhasználási területét vázoljuk fel, a negyedik alfejezetben pedig a PKI-vel kapcsolatban gyakran felmerülő kérdéseket gyűjtjük csokorba, és megpróbáljuk megválaszolni őket. Végül azt a kérdést vizsgáljuk meg, hogy lehet-e hamisítani az elektronikus aláírást, azaz mekkora biztonságot jelent ez a technológia a gyakorlatban.

12.1. Hitelesség elektronikusan

„Quis custodiet ipsos custodes?"
(Ki őrzi az őröket? Ki őrködik az őrség felett?)
– Decimus Iunius Iuvenalis

'Who watches the watchmen? Me, Mr. Pessimal.'
'Ah, but who watches you, your grace?' said the inspector, with a brief smile.
'I do that, too. All the time,' said Vimes. 'Believe me.'

(– Ki őrzi az őröket? Én, Mr. Pessimal.
– De Önt ki őrzi, Fennség? – mosolygott a felügyelő.
– Saját magamat is én őrzöm. Folyamatosan – mondta Vimes – Higgye el nekem.)
– Terry Pratchet „Thud!" című könyvéből

Mind az elektronikus, mind a papír alapú világban két koncepció szerint biztosíthatunk hitelességet:

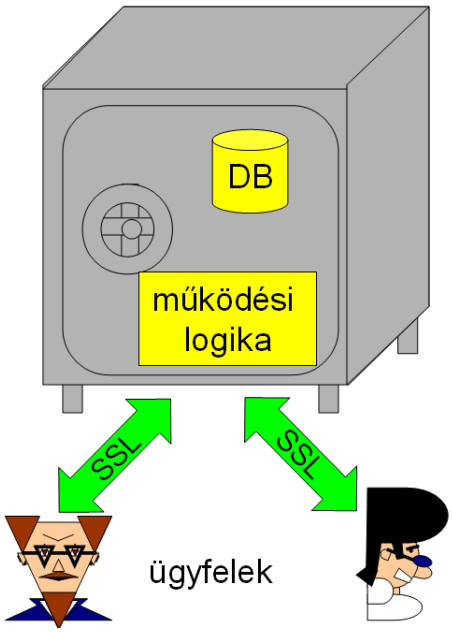
- *Megbízható, zárt, tanúsított rendszert* vagy rendszereket használunk, és e megbízható rendszereket *biztonságos csatornán* keresztül érjük el.
- *Hiteles, aláírt okiratokat* használunk, és a hitelesség ezen okiratokból ered. Papír alapú esetben aláírt, lepecsételt okiratokról, elektronikus esetben elektronikus aláírással ellátott okiratokról beszélünk.

A következőkben e két irányt vetjük össze.

12.1.1. Megbízható, zárt, tanúsított rendszerek biztonságos csatornán

E koncepció szerint az információt megbízható rendszerek kezelik. A rendszerben van egy adatbázis, és van valamilyen üzleti logika (működési logika), amely az adatbázist kezeli, és szabályozza a hozzáférést a benne lévő adatokhoz. A rendszer zártsága biztosítja, hogy sem fizikailag, sem logikailag nem lehet illetéktelenül hozzáférni a benne lévő adatokhoz.

Ha távolról kell hozzáférni az adatokhoz, az kizárólag biztonságos csatornán keresztül lehetséges. Ekkor a biztonságos csatorna kiépítéséhez használt megoldás – ami elektronikus esetben pl. SSL (10.3.2. fejezet) lehet – alapján a rendszer meggyőződhet róla, hogy melyik felhasználóval van kapcsolatban, a felhasználó meggyőződhet róla, hogy valóban a rendszerrel van kapcsolatban, valamint mindketten meggyőződhetnek róla, hogy a csatornán kapott információ valóban a másik féltől származik, és senki nem módosította útközben. (Lásd:



12.1. ábra. **Zárt rendszer**

12.1. ábra.) A rendszer naplózza, hogy mikor, mely felhasználótól milyen adatot kapott. Feltételezzük, hogy a rendszer helyesen működik, és például nem csapja be a felhasználókat.

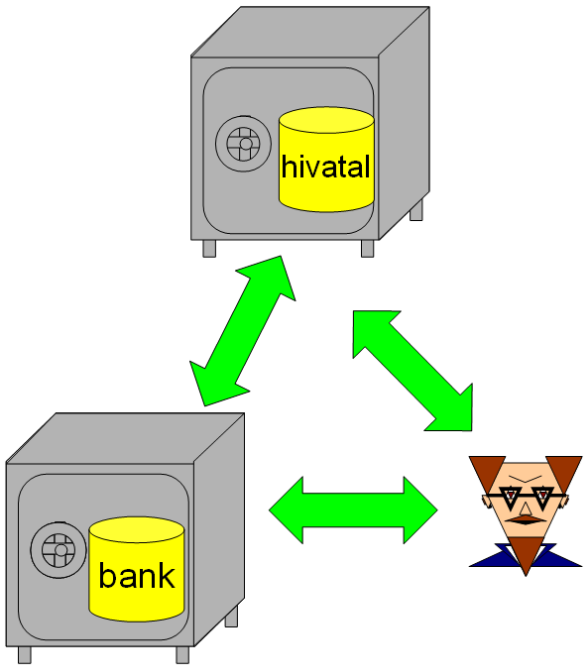
A rendszer adatbázisában szereplő információt tekintjük hitelesnek, és akinek erre szüksége van, az a rendszertől – biztonságos csatornán keresztül – szerezheti ezt be. Vita esetén a rendszer naplói bizonyítják, hogy ki, mikor, milyen műveletet végzett, és ki, mikor és kinek, milyen adatot küldött.

Különösen fontos, hogy az ilyen rendszer valóban zárt legyen, valóban megbízhatóan működjön, és senki se módosíthasson illetéktelenül sem az adatbázison, sem a naplókon. Ezért nagy a hangsúly a fizikai biztonságon, és a biztonságos, szabályzott üzemeltetésen. Az ilyen rendszert tanúsíttatni, auditáltatni szokás, hogy külső, független fél is megerősítse, valóban helyesen, a szabályzatai szerint működik.

Léteznek ilyen rendszerek, de mind a kiépítésük, mind az üzemeltetésük nagyon drága, és e rendszerek méretének növelésével rohamosan nőnek a költségek.

Általában több ilyen rendszer működik egymás mellett, és e rendszerek biztonságos csatornán keresztül kapcsolódnak egymáshoz. Ha az egyik rendszer küld valamit a másik rendszernek, biztonságos csatorna épül ki közöttük, meggyőződnek egymás kilétéről, és mindketten tudják, hogy a másik rendszerből jövő információt senki nem módosította útközben. Ha az egyik rendszernek szüksége van valamilyen információra a másik rendszerből, biztonságos csatornán kérdezi meg.

Tegyük fel, hogy egy felhasználónak szüksége van valamilyen információra az A rendszerből. Biztonságos csatornán keresztül hozzáfér a rendszerhez, amely átadja neki a szükséges x



12.2. ábra. **Zárt rendszerek kapcsolata**

információt. A felhasználó tudja, hogy a rendszer megbízható, és a csatorna biztonságos, így elhiszi, hogy az x információ valóban hiteles, azonos a rendszer adatbázisában lévő etalonnal. Tegyük fel, hogy a felhasználó a B rendszerben is meg akarja adni az A rendszerből származó, hiteles x információt. Biztonságos kapcsolatot létesít a B rendszerrel és elküldi x-et. A B rendszer tudja, hogy a csatorna biztonságos, így az információt valóban ez a felhasználó küldi. Arról viszont nem tud meggyőződni a B rendszer, hogy a felhasználó valóban az A rendszerből kapott információt adta tovább. Ezért a B rendszernek közvetlenül az A rendszerhez kell fordulnia, és onnan kell lekérni az x információt, csak így juthat hozzá hitelesen. (Lásd: 12.2. ábra.)

> ***12.1. Példa:*** *Alajos két megbízható, zárt rendszerrel van kapcsolatban: az egyik egy bank, a másik az adóhivatal. Biztonságos csatornán kapcsolatba lép a bankjával, utasítja a bankot, hogy utaljon át egy meghatározott összeget az adóhivatalnak. A bank visszajelzi, hogy az átutalás megtörtént. Hiába kapta meg Alajos a bank visszaigazolását, hiába tudja, hogy az üzenet a banktól jött, az adóhivatalnak ezzel még nem tudja bizonyítani, hogy átutalta a pénzt. Hiába mutatja be a banktól jött üzenetet az adóhivatalnak, az adóhivatal nem tudja megállapítani, hogy az valóban a banktól jött, és nem Alajos állította elő.*

A zárt rendszerre épülő megoldás esetén a hitelesség a rendszer adatbázisából vagy naplójából ered. Mindenkinek mindig a rendszert kell megkérdeznie, ha meg akar győződni egy információ

hitelességéről. A rendszer állítását el kell fogadni, mert a rendszernek mindig igaza van. A rendszer állítását kétségbe vonni csakis a rendszerből származó bizonyítékok alapján lehetne. Így egy magánszemélynek nem lehet bizonyítéka, nem lehet igaza a rendszerrel szemben.

Mivel az információ attól hiteles, hogy a megbízható rendszer megbízható csatornán küldte nekem, a hitelesség csak addig áll fent, amíg a rendszer működik és online. Ha a rendszer nem érhető el, nincs kit megkérdeznem a hiteles információról.

Szintén probléma merül fel, ha a rendszer valami miatt mégsem működik helyesen. Ha a rendszer adatbázisa vagy naplója megváltozik, nehéz kideríteni, hogy mi a valós információ, és nehéz azt igazolni. A változás történhet véletlen hiba vagy szándékos visszaélés miatt; mindkettő előfordulhat. Nincsenek tökéletes rendszerek, nincsenek hibátlan működési folyamatok. Ha eseti változások történnek, az érintett felhasználók nagyon nehéz helyzetbe kerülnek, mert csak ezen adatbázis, illetve naplók segítségével bizonyíthatnák igazukat. Ha elterjed a rendszerről, hogy az adatbázisa, illetve naplói nem hitelesek, minden benne tárolt vagy általa naplózott információ hitelessége megkérdőjeleződik, és – más bizonyíték híján – nagyon nehéz lesz kideríteni, pontosan mi történt. [100]

> ***12.2. Példa:*** *Manfréd rendszergazda egy hivatalban. Megharagszik Alajosra, ezért töröl róla minden adatot az adatbázisból, és kitörli azon naplóbejegyzéseket, miszerint Alajos valaha is kapcsolatba lépett a hivatallal. Bendegúz köztisztviselő a hivatalban, feltűnik neki, hogy Alajos még nem lépett kapcsolatba a hivatallal, holott kötelessége lett volna bizonyos adatszolgáltatást megküldenie. Felelősségre vonja Alajost, és eljárást indít ellene. Alajosnak nagyon nehéz lesz bizonyítania az igazát.*

Ha egy dokumentum elhagyta a rendszert, már nem hiteles. Ezért minden szereplőnek mindig a rendszerben lévő információkhoz kell hozzáfernie. Ez nagyon intenzív kommunikációt jelent, a rendszer nagy terhelésnek van kitéve. Elektronikus esetben, nagy tömegben ez csak számítógépek, automatizmusok segítségével képzelhető el. Egy tanúsított, zárt központ könnyen szűk keresztmetszetté válhat. Ugyanakkor ha megnöveljük a központ kapacitását, az nagyon drága, és – az egyre több eszköz, az egyre nagyobb terület, az egyre több üzemeltető stb. miatt – egyre nehezebb biztosítani a zártságot. A központok automatái intenzíven kommunikálnak egymással, ehhez egyszerre kell online lenniük, egyszerre kell elkészülniük, és kompatibilisnek kell lenniük egymással, azaz érteniük kell egymás üzeneteit.

> ***12.3. Példa:*** *Tegyük fel, hogy 100 rendszer működik. E rendszerek markánsan különböznek egymástól, így például különböző adatokat kezelnek. Ez a 100 rendszer kell, hogy kommunikáljon egymással és a felhasználóikkal. Ha azt szeretnénk, hogy mind a 100 rendszer kommunikálhasson az összes többivel, ez már önmagában 100*99/2 = 4950 kommunikációs csatornát jelent, amelyet egyeztetni kell, és*

valamilyen módon – pl. XML sémával – le kell tudni írni. A szükséges rendszereket ki kell tudni fejleszteni, és a fejlesztéseknek (lényegében) egyszerre kell elkészülniük, mert ha van olyan kommunikációs csatorna, ahol a két rendszer még nem kompatibilis, akkor az adott csatornát használó folyamatok egyáltalán nem tudnak működni.

Ez óriási feladat, és a felhasználókkal való kommunikáció még szóba sem került. Vegyül figyelembe, a 100 nagyon kicsi szám, pusztán a magyar közigazgatásban ennél lényegesen több rendszer fordul elő.

12.1.2. Hiteles, aláírt okiratok alapján

Ha a hitelesség aláírt okiratokra épül, akkor az információ hitelességéről nem az alapján győződök meg, hogy kitől kaptam, hanem azt nézem meg, hogy sértetlen-e az okirat, és ki írta alá. Az aláírt okirat bárhol, bárkinél hiteles, attól függetlenül, hogy hol van. (Lásd: 12.3. ábra.)

A rendszereknek továbbra is van adatbázisa, és vannak naplóik, de a fontos információkról aláírt okiratokat is megőriznek. Az okiratok egy részét maguk hozzák létre, egy másik részét az ügyfeleik írják alá. Az ügyfél mindig aláírt okiratban fordul a rendszerhez, és a rendszer is aláírt okiratban válaszol.

A rendszer elszámoltatható. Bárkinek, bármilyen felhasználónak, ügyfélnek lehet olyan bizonyítéka, amely szerint a rendszer ekkor és ekkor, ezt és ezt állította, és ezt a rendszer nem, vagy csak nagyon nehezen tudja letagadni.

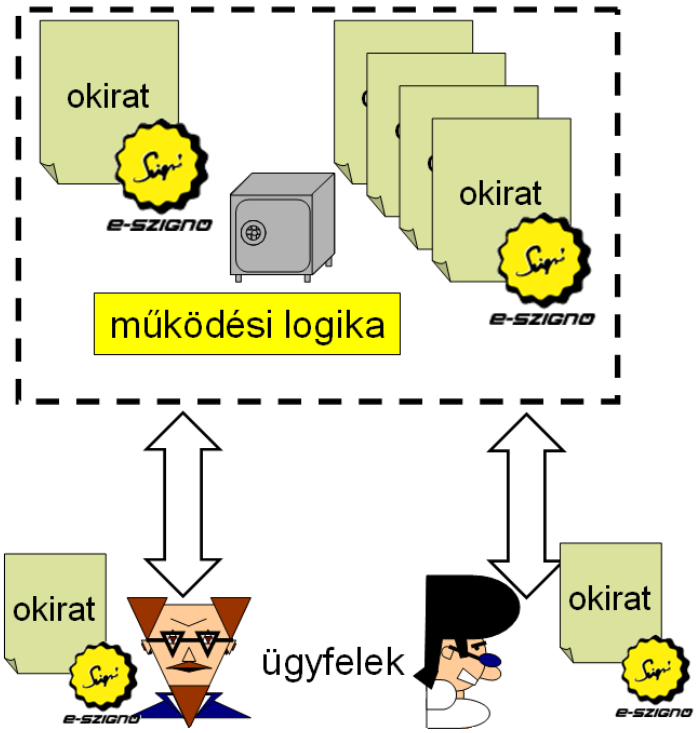
Ha egy felhasználó számára az A rendszer kiállít egy okiratot, a felhasználó ezt bemutathatja a B rendszernek, aki meg tudja állapítani, hogy az okiratot valóban az A rendszer állította ki. Nincs szükség rá, hogy a B rendszer (vagy bárki más) az okirat ellenőrzése végett ismét az A rendszerhez forduljon. (Lásd: 12.4. ábra.)

Az információk hitelessége magukból az aláírt okiratokból megállapítható, így a rendszereket sokkal kevesebbet kell kérdezgetni. Ennek következtében egyrészt kisebb, egyszerűbb, olcsóbb rendszerek állíthatóak fel, másrészt nem jelent katasztrófát, ha valamelyik rendszer nem érhető el (akár azért, mert éppen elromlott, vagy azért, mert az adott hivatal pl. csak hétfőn van nyitva).

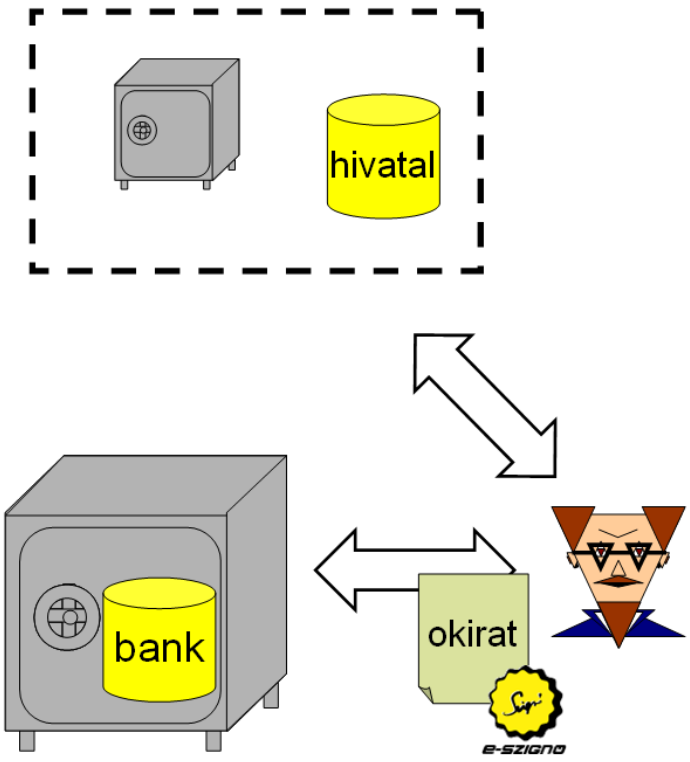
Akkor sem történik tragédia, ha egy rendszer valami miatt megzavarodik. Az aláírt okiratok hitelessége könnyen megállapítható, és belőlük rekonstruálható, hogy mi történt. Ha minden okirat megsemmisült a rendszerben, akkor is támaszkodhatunk a rendszerből kikerült, a felhasználóknál lévő okiratokra.

E megoldásnak is vannak korlátai:

- Az egyes rendszerek nem tudják letagadni a hibáikat. Emiatt az egyes rendszerek sok esetben „tartanak” a hiteles, „letagadhatatlan” dokumentumoktól. Az is igaz, hogy ha



12.3. ábra. **Hiteles dokumentumok alapján működő rendszer**



12.4. ábra. **Kommunikáció hiteles dokumentumok alapján**

egy, az előző pontban leírt zárt rendszer rájön, hogy hibás adatot adott ki, javítja az adatbázisát, és onnantól kezdve a hiba megszűnt: ha legközelebb megkérdezik, már a helyes választ adja. Ugyanakkor e megoldás nemcsak javítja, hanem letagadja, eltussolja a hibát. Emiatt nagyon kényelmetlen helyzetbe kerülhetnek azok a felhasználók, akik a hibás információ alapján jártak el.

A hibásan kiadott okiratokat sokkal nehezebb javítani, mert az már ott van a felhasználónál, és nincs a rendszer fennhatósága alatt. Igaz, ekkor a jóhiszemű felhasználó nem kerülhet bajba, mert ő hiteles dokumentummal rendelkezik.

Az okiratok nemcsak a felhasználót védik, hanem a rendszert is. A rendszer hivatkozhat arra, hogy azért végzett el egy adott műveletet, mert a felhasználó írásban, hiteles okiratban utasította.

> ***12.4. Példa:*** *Alajos azt látja, hogy a bankja átutalt a számlájáról 1 millió forintot Bendegúz számlájára. Megkérdezi a bankot, hogy milyen alapon utalta át a pénzt.*
>
> *A bank válaszolhatja, hogy a naplói szerint Alajos ekkor és ekkor lépett be a rendszerébe, és ilyen és ilyen átutalást indított. Alajos ezt tagadja; vagy azt állítja, hogy akkor nem lépett be, vagy azt, hogy ő bizony nem indított olyan átutalást. Vita indul, ahol Alajos állítása áll a bank állításával szemben. A bíróság valószínűleg a bank naplóinak fog hinni, de a banknak sem szerencsés az ilyen vita.*
>
> *Másik lehetőség, hogy a bank bemutatja az átutalási megbízást, amelyet Alajos aláírt, és amelyben utasítja a bankot az összeg átutalására. Egyértelműen megállapítható, hogy Alajos aláírása van-e az átutalási megbízáson. Alajos ezt is vitathatja, de a bank sokkal tisztább helyzetben van, ő tudja bizonyítani, hogy Alajos utasítása alapján járt el.*
>
> *Ha a bank átutalási megbízást kell, hogy mutasson, Alajos el is várhatja ezt. Megkövetelheti, hogy a bank mutassa be azon aláírt okiratot, amely alapján végrehajtotta a tranzakciót. Aláírt okiratok alapján tisztábbá válik a kapcsolat a felek között.*

- Az okiratokat valamilyen módon hitelesíteni, illetve a hitelességet ellenőrizni kell. Papír alapon a hitelesítés aláírással és bélyegzővel szokott történni, elektronikus esetben elektronikus aláírással.

  Vegyük figyelembe, hogy ez nem plusz költséget jelent, hanem a hitelességet biztosítja; ezáltal kapacitás spórolható meg, valamint jelentősen csökkenthetőek a rendszer zártságát biztosító erőforrások. Okiratok esetén a hitelesítéshez használt elemeket (a pecsétnyomót, elektronikus esetben a magánkulcsot) kell védeni.

- A hiteles okiratokon nem lehet változtatni. Nemcsak a rossz szándékú, hanem a jó szándékú változtatás sem lehetséges. Papír alapon nehezebb elképzelni[1] jó szándékú változtatást, elektronikusan ilyen lehet például: ha átkonvertáljuk a dokumentumokat az újabb szoftverek által támogatott formátumba, vagy ha kiderül, hogy hibás volt a régi dokumentumok formátuma, és ki akarjuk javítani őket. Mivel elektronikus esetben bármely bit megváltozása sérti az aláírást, nem könnyű különbséget tenni jó és rossz szándékú módosítás között.

  Elektronikus esetben úgy lehet javítást végezni, hogy másik, javító okiratot bocsátunk ki, vagy javító aláírt záradékot csatolunk egy meglévő dokumentumhoz. A régi, aláírt dokumentumban nem lehet módosítani (illetve az az eredeti aláírás megsértését jelenti).

- A felhasználónak sokszor kényelmesebb, ha kapcsolatba lép az A zárt rendszerrel, és az „megoldja” a problémáját, azaz kapcsolatba lép az összes többi zárt rendszerrel, és biztonságos csatornán összegyűjti a szükséges információkat, igazolásokat.

  > ***12.5. Példa:*** *Alajos kapcsolatba lép az adóhivatallal, beadja az adóbevallását, mire az adóhivatal kapcsolatba lép Alajos bankjával, és automatikusan leemeli Alajos bankszámlájáról a bevallott összeget. Ez kényelmes megoldás.*
  >
  > *Igaz, Alajos lehet, hogy nyugodtabban alszik, ha ő maga rendelkezik a saját bankszámlája felett.*

  A kényelmi kérdésekkel adatvédelmi kérdések állnak szemben. Ha a rendszerek nem okiratok alapján kommunikálnak, közvetlen kapcsolat szükséges közöttük. Mi alapján van joga az egyik zárt rendszernek hozzáférni a felhasználó másik zárt rendszerben lévő adataihoz? Biztos, hogy a felhasználó szeretné ezt a hozzáférést? Hogyan kell engedélyezni ezt a hozzáférést? Elegendő, ha az A megbízható rendszer állítja, hogy neki szüksége van valamilyen adatra a B rendszerben? Ha valamelyik zárt rendszer megzavarodik, akkor a többi rendszerben lévő adatok sincsenek biztonságban?

  A meglévő, papír alapú rendszerek hiteles okiratok alapján működnek, így a felhasználók már hozzászoktak ehhez a kényelmetlenséghez.

12.1.3. Meglévő rendszereink hitelessége

A meglévő, papír alapú rendszerek hiteles, aláírt okiratokra épülnek, mert hiteles okiratok alapján lehet később bizonyítani, hogy mi történt. A zárt rendszerrel fennálló biztonságos kapcsolat a szóban, négyszemközt kötött szerződésnek felel meg: a megbeszélés folyamán

[1] Leszámítva olyan extrém eseteket, mint amikor egy nagyon régi, szabad szemmel már olvashatatlan okiratot restaurálnak.

mindkét fél tudja, hogy kivel áll szemben, mindkét fél tudja, hogy amit a másik mond, az valóban tőle származik, és senki nem módosítja útközben. De a következő pillanatban lehet, hogy már másképp emlékeznek, vagy valamelyikőjük le akarja tagadni, hogy miről volt szó. A szóban kötött szerződések is érvényesek (leszámítva bizonyos speciális eseteket, amikor a jog írásba foglalást ír elő), de az emberek mégis leírják a fontos megállapodásokat, akkor is, ha az nem kötelező. Így kevesebb a félreértés, és könnyebb bizonyítani, hogy miről szólt a megállapodás.

A meglévő, papír alapú rendszerek *papír* alapúak, azaz *írásra*, aláírásra épülnek. A szó pozitív értelmében vett bürokratikus rendszerek, amelyek írott szabályzatokra, írásos utasításokra, munkaköri leírásokra, írott, aláírt feljegyzésekre, aláírt megállapodásokra és okiratokra épülnek. Ezek nem csak tárgyak, nem csak eszközök, hanem egy gondolkodásmód, egy paradigma is kapcsolódik hozzájuk, és e paradigma régóta, évszázadokban mérhető ideje működik, és a korábban ismert megoldásoknál sokkalta hatékonyabban működik.

Ha a papír alapú okiratok helyett elektronikus okiratok jelennek meg, továbbra is a meglévő paradigmát követjük, és ha az eszközök meg is változnak, elveiben csak nagyon kicsit változik a rendszer.

Ezzel szemben, ha az elektronikus kommunikációt a meglévőtől gyökeresen eltérő paradigmákra építjük, akkor a folyamatok elektronizálása mellett (ami már önmagában sem kis kihívás) egyúttal elveiben is valami egészen másba vágunk bele. A meglévő folyamatok, a meglévő rendszerek, és a bennük részt vevő emberek nincsenek felkészülve rá, hogy valami meghökkentően mást tegyünk.

Ha egy kritikus ügyet a papír alapú világban írásban intézünk, akkor elektronikusan helyes megoldás ugyanazon ügyet csupán SSL csatornán keresztül, írásbeliség nélkül bonyolítani?

Ha az elektronizálás mellett egyúttal a hitelességet is más alapokra helyezzük, ezentúl nem aláírt okiratok, hanem adatbázisszerverek bizonyítják a fontos tényeket, és a hitelességet nem közjegyzők és ügyvédek, hanem szoftverfejlesztők és rendszergazdák igazolják.

A papír alapú okiratokat megszoktuk. Ha az okiratok elektronikussá válnak, az már önmagában nagy változást jelent. De ha ezen túlmenően még az okirat szerepét is egy adatbázis-mező egy értéke veszi át, olyan területre tévedünk, ahol nincsenek több száz éves tapasztalataink.

Elektronikus aláírás alapon okiratokat, elektronikus okiratokat kezelünk. Az elektronikusan aláírt okiratoknak vannak bizonyos speciális tulajdonságai – például minden másolata hiteles, eredeti –, de továbbra is okirat, amelynek sértetlensége, hitelessége önmagában is megállapítható.

A meglévő, papír alapú okiratokra épülő folyamatok viszonylag könnyen leképezhetőek elektronikusan aláírt okiratokra épülő folyamatokká. Léteznek megoldások a papír alapú

okirat hiteles elektronikus okirattá való konverziójára[2], és léteznek megoldások az elektronikusan aláírt okirat hiteles papír alapú okirattá való konverziójára[3].

Ha elektronikus aláírásra építünk egy elektronikus rendszert, a meglévő folyamatok szerint dolgozhatunk, a rendszernek nem kell zártnak lennie, mert a rendszert elhagyó okiratok továbbra is hitelesek maradnak, és már akkor is élvezhetjük az elektronikus világ bizonyos előnyeit, ha az egyes rendszereink nem egyszerre készülnek el.

Elektronikus aláírás alapon jelentősen könnyebb eredményt felmutatni.

12.2. Az elektronikus aláírás bevezetése

Tegyük fel, hogy egy meglévő rendszerbe szeretnénk bevezetni az elektronikus aláírást. Lehet, hogy egy papír alapú rendszert szeretnénk elektronizálni, de az is lehet, hogy egy elektronikus rendszer egyes funkcióihoz szeretnénk elektronikus aláírást kapcsolni. Melyek azok a pontok, ahova különös figyelmet célszerű fordítani, hogy a projekt sikerrel záruljon?

1. *Szerezzük meg a kulcsszereplők támogatását!*
   Ez alapvetően fontos; bárhol, bármilyen rendszerben, bármilyen változtatást is szeretnénk eszközölni, bírnunk kell a kulcsszereplők támogatását. [23] Át kell gondolni, hogy egy adott rendszer tekintetében kik számítanak kulcsszereplőnek, de mindenképpen a kulcsszereplők közé tartozik:
   - aki a rendszer üzleti folyamataiért felel,
   - a rendszer felhasználója, aki az elektronikus aláírással kapcsoltba kerül (akár belső felhasználó, akár külső ügyfél, akár aláírást készít, akár ellenőrzi az aláírást), és
   - aki az elektronikus aláírás technológia költségeit fedezi.
2. *Ne csináljunk felesleges felfordulást!*
   Az elektronikus aláírás bevezetése így is, úgy is nagy változást jelent. Csak annyit változtassunk, amennyit szükséges, és vigyázzuk, hogy a projekt scope-ja ne nőjön meg hirtelen, azaz ha a feladat a rendszer elektronizálása volt, ne tervezzük újra a rendszerben se az informatikai, se az üzleti folyamatokat.
3. *Az elektronikus aláírás szervesen épüljön be a folyamatba.*
4. *Legyen aláírási szabályzatunk!* (Lásd: 6.8. fejezet.)
5. *Megfelelő aláírás-formátumot válasszunk!* (Lásd: 6.4. fejezet.)
6. *Tisztázzuk az elektronikus folyamatok kapcsolatát a papír alapú rendszerekkel!*

[2]Erre szolgál pl. a 3/2005. IHM rendelet. [80]

[3]E megoldások is arra épülnek, hogy valaki nyilatkozik róla, hogy a két okirat megegyezik

12.2.1. A rendszer kulcsszereplőinek támogatása

A rendszerben működő üzleti folyamatokért felelős személyt az motiválhatja, hogy:

- Az elektronikus aláírás használata miatt a rendszerben lévő folyamatok elszámoltathatósága megnő.
- Pénzt takaríthat meg. Például egy elektronikus számla (12.3.1. fejezet) kibocsátása sokkal olcsóbb, mint egy papír alapú számláé, illetve egy irat elektronikusan történő archiválása sokkal olcsóbb, mint ugyanez papír alapon.
- Előírták neki (pl. jogszabályban) az elektronikus aláírás használatát.

Az elektronikus aláíráshoz mindenképpen kapcsolódnak költségek is. Egy elektronikus aláírásra épülő rendszer költségei összevethetőek egy tanúsított, auditált zárt rendszer (12.1.1. fejezet) költségeivel, de ne is próbáljuk összevetni egy olyan rendszer költségeivel, amely se aláírással, se szigorúan auditált biztonsági szabályokkal nem biztosít hitelességet. Ha nem hitelesen működik a rendszer, rövid távon egészen biztosan olcsóbb lesz. Hosszú távon különféle problémák adódhatnak abból, hogy a rendszer nem hitelesen működik, mérlegelni kell, hogy ezek kockázata milyen arányban áll akár az elektronikus aláírás, akár a zárt rendszer költségeivel.

Egyik gyakori megoldás, hogy a rendszert üzemeltető szervezet látja el elektronikus aláírással a felhasználókat, a másik pedig, hogy a felhasználók (vagy ügyfelek) külön-külön szereznek maguknak elektronikus aláíráshoz szükséges eszközöket a rendszer használatához.

Miért dönt úgy valaki, hogy elektronikus aláírással szeretne használni egy rendszert? A tapasztalat azt mutatja, a felhasználókat *nem* motiválják a következők:

- Elektronikus aláírással kényelmesen, sorbanállás nélkül intézheti ügyeit.
  Ha valaki csak évi egy-két alkalommal használ egy rendszert, azért nem éri meg neki (minősített aláírás készítéséhez szükséges) tanúsítványt vásárolni, inkább bevállal némi sorbanállást. Magyarországon még az sem terjedt el, hogy a közüzemi díjakat átutalással (aminek költsége van) fizetik, helyette az emberek – ha nem akarnak csoportos beszedési megbízással fizetni – általában azt választják, hogy a postán állnak sorba (ami „ingyen” van).

  Ráadásul, az elektronikus aláírás használatával is járnak kényelmetlenségek, és az emberek többsége szívesebben választja az ismert kényelmetlenséget az új, ismeretlen helyett.

- Elektronikus aláírással biztonságosabb.
  Ez csak egy mítosz. Az elektronikus aláírás valóban növelheti a biztonságot, de

elsősorban nem az aláíró, hanem az aláírást befogadó fél számára. Aki választhat, hogy aláír egy nyilatkozatot, vagy csak úgy, aláíratlanul mond valamit, az várhatóan az utóbbit választja, mert ha baj van, úgy később nem lehet rábizonyítani.

> **Megjegyzés:** Dönthet valaki azért az aláírt nyilatkozat mellett, mert így később bizonyítani tudja majd, hogy pontosan mit is állított.
> Ez hibás érvelés. Egy aláírt dokumentumból csak az következik, hogy az aláíró valóban aláírta. Az nem következik belőle, hogy át is adta valakinek.

> ***12.6. Példa:*** *Tegyük fel, hogy egy bírósági vitában:*
> - *Bendegúz felmutat egy aláírt árajánlatot, amelyben Alajos X forintért hajlandó eladni az autóját Bendegúznak. Ez bizonyítja, hogy Alajos valóban megtette ezt az árajánlatot.*
> - *Alajos felmutat egy aláírt árajánlatot, amiben ő (Alajos) X forintért hajlandó eladni az autóját Bendegúznak. Ez nem bizonyítja, hogy Alajos valóban megtette ezt az árajánlatot. Lehet, hogy Alajos szóban Y forintot mondott Bendegúznak, az X forintról szóló árajánlatot pedig aláírta ugyan (hogy szükség esetén tudja mutogatni), de soha nem adta át Bendegúznak. Az aláírás (szinte mindig) a befogadót védi.*
>
> *(Ha Bendegúz aláírtan visszaigazolta volna, hogy az adott árajánlatot átvette, az már Alajos számára is bírna előnnyel.)*

Az (elektronikus) aláírás biztonsági előnyeit szinte mindig a befogadó fél élvezi, és nem az aláíró számára jelent biztonságot. A felhasználó nem a biztonság miatt fog elektronikus aláírást vásárolni.

- Az elektronikus aláírás még másra is jó lesz.
  Ez igaz lenne, és erős érv lenne az elektronikus aláírás irányában, de a hazai piac még nem tart itt. Általában valakinek egyetlen szolgáltatás miatt van szüksége elektronikus aláírásra.

- A felhasználó fanatikus elektronikus aláírás rajongó, aki mindenhol, mindenképpen elektronikus aláírást akar használni.
  Van ilyen ember, de ez mindig is egy elenyésző kisebbség lesz.

Mégis, miért dönt úgy valaki, hogy elektronikus aláírással szeretne használni egy rendszert? Akkor dönt az elektronikus aláírás mellett, ha az valamilyen „lényeges előnyt” nyújt a számára, azaz például:

- Feltétlenül használni akarja a rendszer egy olyan szolgáltatását, amely csak elektronikus aláírással érhető el.

Papír alapon általános, hogy a fontos nyilatkozatokat írásba kell foglalni. Ez elektronikus esetben is nyugodtan elvárható lehet, és csak a legalább fokozott biztonságú elektronikus aláírással ellátott dokumentum tekinthető írásba foglaltnak.

- Elektronikus aláírással kedvezőbben (pl. olcsóbban) veheti igénybe a szolgáltatást, és ez arányban áll az elektronikus aláírás költségeivel. Ne feledjük, ma tipikusan egyetlen szolgáltatás miatt vesz valaki elektronikus aláírást, ritka, hogy valaki több elektronikus aláírást befogadó szolgáltatással állna kapcsolatban.

  Az elektronikus aláírás tipikusan az aláírást befogadó félnek jó, ő van nagyobb biztonságban. Ha papír helyett elektronikusan fogadhat be erős bizonyítékot jelentő okiratokat, akkor azokat sokkal könnyebben tudja menedzselni (tárolni, továbbítani, keresni stb). A befogadó, és nem az aláíró fél az, aki gazdasági előnyhöz jut az elektronikus aláírás miatt. Ha az elektronikus aláírás költségei az aláírónál jelentkeznek, míg az előnyöket a befogadó élvezi, akkor az aláíró nem lesz motiválva, hogy elektronikus aláírást használjon.

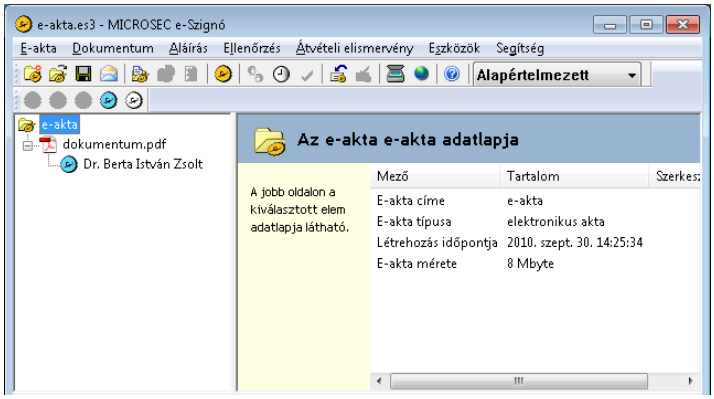
  Ez vagy úgy oldható fel, hogy a befogadó átenged bizonyos előnyöket az aláírónak, vagy úgy, hogy ő finanszírozza az aláírást.

- Elektronikus aláírással közvetlen költségeket takaríthat meg, és ez arányban áll az elektronikus aláírás költségeivel.

  A papír alapú dokumentumok kezelése – nyomtatása, sokszorosítása, továbbítása – drága. Sok esetben már azzal jelentős megtakarítás érhető el, ha a papír helyett elektronikus dokumentumokat használunk. Például egy elektronikus számlát sokkal olcsóbb létrehozni és elküldeni, mint egy papír számlát. Ha valahol sok oldalnyi dokumentumot (esetleg sok példányban) kell benyújtanunk, esetleg már azzal jelentős megtakarítást érünk el, hogy nem papíron, hanem elektronikusan adjuk be.

Ha egy rendszerben ugyanaz a művelet elvégezhető elektronikus aláírással is és elektronikusan, de aláírás nélkül is, és ugyanazon feltételekkel vehető igénybe, akkor a felhasználók várhatóan az aláírás nélküli változatot fogják használni, mert az egyszerűbb, és állításaikat így később nem lehet rájuk bizonyítani. Ha ez elviselhető kockázatot jelent, akkor a rendszerben valószínűleg nincsen szükség aláírásra.

> ***12.7. Példa:*** *Tekintsünk egy webes könyvesboltot. A felhasználó regisztrálhat, létrehoz egy fiókot, betesz könyveket a bevásárló kosarába, majd megrendeli őket. A posta kiszállítja a könyveket, és a felhasználó utánvéttel fizet. A rendszer jelenleg elektronikus aláírás nélkül működik. Vizsgáljuk meg, milyen előny származik az elektronikus aláírás bevezetéséből ebben a rendszerben?*
>
> *Valószínűleg semmilyen. Világszerte számos bolt működik így, és jól működnek. A kereskedő számára nem jelent túl nagy kockázatot, hogy kiküldi a könyvet, és azt*



12.5. ábra. **A PDF dokumentumot elhelyeztük egy e-aktában, és ezen e-aktát írtuk alá. Magán a PDF-en nem látszik az aláírás, az aláírás csak a e-aktában jelenik meg.**

> *nem fizetik ki, nem bombázzák őt „vicces” emberek tömegei, akik kamu címekre rendelnek drága és nehéz könyveket. A kereskedőnek így is jó, és nem akarja akár egy kicsit is nehézkesebbé tenni a rendszer felhasználói felületét. A felhasználónak is jó így, mert a rendszer használata egyszerű, nem igényel speciális eszközt, és így bármikor dönthet úgy, hogy nem veszi át a szállítmányt.*

Ott kell megkövetelni az aláírást, ahol olyan érdemi műveletet végzünk, amelyhez aláírt nyilatkozatot kérünk a felhasználótól. Ott kell megkövetelni az aláírást, ahol nem viselhető el annak a kockázata, hogy a felhasználó egyszer csak letagadja az állítását. Ezeken a pontokon viszont igenis meg kell követelni, és ezeken a pontokon nem szabad aláíratlan állítást elfogadni.

12.2.2. Hogyan járhat a legkisebb változtatással?

Több, különböző megközelítés szerint támaszkodhatunk elektronikusan aláírt okiratokra. Egyik megoldás, ha a hagyományos, papír alapú dokumentumhoz hasonló iratokat – azok nyomtatási képét vagy a kinyomtatott dokumentumok beszkennelt változatát – írjuk alá, például e-aktába ágyazva (lásd: 12.5. ábra). Másik lehetőség, ha kifejezetten elektronikus, géppel is értelmezhető, automatizáltan is feldolgozható okiratokat hozunk létre. Vizsgáljuk meg, melyik lehetőség milyen tulajdonságokkal bír, milyen és mekkora változtatások szükségesek hozzájuk.

12.2.2.1. Elektronikus papír (pl. PDF)

Elektronikus papírról akkor beszélünk, ha az iratok nyomtatási képével vagy beszkennelt változatával dolgozunk, és ezeket írjuk alá. Ilyen formátum például a PDF. Az okiratokat pontosan úgy használjuk, mintha papíron lennének, csak nem nyomtatjuk ki őket.

Ennek következtében a papír alapú folyamatok (lényegében) egy az egyben leképezhetőek elektronikus folyamattá.

E megoldás könnyebben megvalósítható, könnyebben bevezethető, mert a meglévő folyamatokon legfeljebb minimális mértékben kell változtatni, és a felhasználóknak sem kell gyökeresen új folyamatot megtanulniuk. A felhasználók ekkor továbbra is „kézzel" dolgozzák fel az iratokat.

Ha később változik a folyamat, az új iratok formátuma könnyen kialakítható, és – mivel emberi felhasználó fogadja be az iratot – a formátum-változás a befogadó oldalon sem igényel jelentős módosítást.

Az elektronikus papír okiratokat csak ember tudja olvasni (bár egy nyomtatási kép iratban már lehet géppel keresni), de az ember könnyen elolvassa, és más rendszerben is, akár új, ismeretlen rendszerben is könnyen befogadja az iratot, mert az elektronikus papír befogadásához nincs szükség speciális szoftverre. Ebből kifolyólag a régen készült, régi formátumú okiratok olvasása, értelmezése is egyszerű feladat.

12.2.2.2. Intelligens nyomtatvány (pl. XML)

Intelligens nyomtatványról beszélünk, ha géppel is értelmezhető okiratot írunk alá. Az intelligens nyomtatványt vagy eleve automata készíti, vagy emberi felhasználó tölti ki egy nyomtatványkitöltő program segítségével. Az intelligens nyomtatványt gép is be tudja fogadni, ember pedig egy megjelenítő programmal tudja értelmezni.

Az automatizálható feldolgozás miatt intelligens nyomtatványok alapján nagyon hatékonyan működhet a folyamat, de használatuk jelentős előkészítést igényel. Az intelligens nyomtatványok kialakításához a működési folyamatok alapos ismerete, feltérképezése szükséges, specifikálni kell a nyomtatványok formátumát, és ki kell fejleszteni a nyomtatványokat készítő és feldolgozó automatákat, valamint felhasználói programokat.

Az intelligens nyomtatványokat nehezebb illeszteni a külső, papír alapú folyamatokhoz, mert a beszkennelt formanyomtatványokat nehéz géppel feldolgozni.

Ha valami miatt változtatni kell a működési folyamatokon, az szoftverfejlesztést igényel mind a kibocsátó, mind a befogadó oldalon, és e fejlesztésnek egyszerre kell elkészülnie.

A régen készült intelligens nyomtatványok olvasása hosszú távon problémás lehet, mert szükséges hozzá a régi formátum specifikációja, és a régi formátum megjelenítéséhez szükséges szoftverkörnyezet is.

12.2.2.3. Hibrid megoldások

A korábban leírt megoldások kombinálhatóak egymással. Intelligens nyomtatványokkal például nagyon hatékonyan lehet dolgozni, azaz egységnyi idő alatt sokkal több nyomtatványt

lehet feldolgozni, mint kézzel. Ugyanakkor kiépítésük igen bonyolult és költséges lehet. Sokszor nem ismeri egy szervezet kellőképpen saját folyamatait, és a szükséges szoftverfejlesztések sokszor nem készülnek el (időre). Az elektronikus papír kevésbé hatékony, de nagyon könnyen bevezethető, és rugalmasan változtatható megoldás.

Egy lehetséges kombinációt jelent például az elektronikus papír alapú elektronikusan aláírt okiratok bevezetése, majd az elektronikus papír nyomtatványok fokozatos kiváltása intelligens nyomtatványokkal. Nem kell megcélozni azonnal az összes nyomtatvány kiváltását, elég, ha a gyakran használt elektronikus papír okiratokat kezdjük el lecserélni intelligens nyomtatvány okiratokra. Egyes pontokon esetleg egyáltalán nincsen értelme kiváltani az elektronikus papírt. Azt sem kell megcélozni, hogy a küldő és a befogadó oldalon egyszerre álljon elő az automatizált feldolgozást végző szoftverkörnyezet. Lehet, hogy a küldő egy programmal tölti ki az intelligens nyomtatványt, a befogadó pedig egy XML stíluslap (XSLT) segítségével jeleníti meg, elektronikus papír okiratként. Az is lehet, hogy a küldő egy általános PDF kezelő programmal tölt ki egy PDF űrlapot, amit a befogadó automatizáltan dolgoz fel.

Az elektronikus cégeljárás (13.1. fejezet) is elektronikus papír alapokon, beszkennelt és elektronikusan aláírt okiratok alapján indult el, és a rendszer beindulása óta fokozatosan mozdul el az intelligens nyomtatványok irányába.

A zárt rendszer és a hiteles okiratokra épülő rendszer közötti határvonal sem mindig annyira éles. Szerencsés, ha az elektronikusan aláírt okiratokra épülő rendszer is zárt, és fizikailag biztonságos. Másrészt célszerű, ha a zárt rendszer bizonyos esetekben elektronikus aláírással ellátott elektronikus okiratot állít ki, vagy azt követel meg. Általában kimondható, hogy egy valóban megbízható rendszertől ma már elvárható, hogy:

- a „lényeges” kimenő dokumentumokat aláírja,
- a „lényeges” utasításokat hitelesen, aláírtan fogadja el,
- a befogadott dokumentumokra aláírt tértivevényt adjon válaszul,
- elszámoltatható legyen, azaz meg lehessen állapítani, hogy az utasításokat maradéktalanul végrehajtotta-e.

Eltérő koncepciókat mutattunk be, szélsőséges megoldásokon keresztül. A legjobb megoldások nagyon gyakran az adott alkalmazási terület igényeihez illeszkedő megfelelő ötvözésükből születnek.

12.2.3. Az elektronikus aláírás szervesen épüljön be a folyamatba

Hangoljuk össze az elektronikus aláírást a folyamat más részeivel, találjuk meg köztük a megfelelő egyensúlyt: Ha a projekt célja a rendszer elektronizálása, akkor a projekt ne az

elektronikus aláírásról szóljon. Az elektronikus aláírásra eszközként tekintsünk, és használjuk a már meglévő, kész, kifejlesztett eszközöket. Egy aláírás-létrehozó alkalmazás fejlesztése óriási feladat, kerüljük el, hogy ilyet kelljen fejleszteni.

Ne használjunk speciális, egyedi formátumokat, és ne követeljünk meg speciális tanúsítvány-profilokat. Csak arra használjuk az aláírást, amire való: legyen bizonyítható, hogy az adott személy aláírta a dokumentumot. Ne próbáljunk az aláírásból vagy a tanúsítványból más információt kinyerni, például, hogy kicsoda az illető, és az adatait se a tanúsítványból olvassuk ki. Ha ilyen információkra van szükségünk, használjunk inkább attribútum-tanúsítványt (11. fejezet). Ne írjunk elő speciális követelményeket az aláírásokra, használjunk egy már meglévő, „konzerv" aláírási szabályzatot. Ezek mind-mind csökkentenék annak esélyét, hogy az alkalmazott elektronikus aláírás technológia máshol, más rendszerben is használható legyen, és ezáltal éppenséggel megnövelnék az elektronikus aláíráshoz kapcsolódó költségeket – mind a mi számunkra, mind ügyfeleink számára.

Az elektronikus aláírás szolgáljon, és ne uralkodjon el! Ez csak egy technológia, egy eszköz, a lényeg a folyamat maradjon. Ugyanakkor tudatosítsuk a nem elektronikus aláírással foglalkozókban, hogyan kell illeszkedniük az elektronikus aláíráshoz. Legegyszerűbb, ha az elektronikus aláírást ugyanúgy, és ugyanarra használjuk, mint a papír alapú aláírást. Egy nagyon fontos különbség van köztük, amit mindenkinek tudnia kell: ami egyszer aláírásra/időbélyegzésre került, azt soha többet nem lehet megváltoztatni.

Az elektronikus folyamatok ne éljenek külön életet! Ha ugyanazt a műveletet el lehet végezni papír alapon is és elektronikusan is, a két folyamat legyen összhangban, és legyen egy kézben. Sőt, optimális esetben ne külön folyamat végezze az elektronikus és a papír alapú dokumentumok feldolgozását. Különösen fontos, hogy az elektronikus és a papír alapú műveletekhez azonos biztonsági szint kapcsolódjon.

12.2.4. Legyen aláírási szabályzatunk!

Nem feltétlenül kell az aláírási szabályzatnak írott, formális, nyilvános dokumentumnak lennie. Ellenben a rendszer tervezésekor mindenképpen végig kell gondolni az aláírási szabályzatokban szereplő (6.8. fejezet) főbb pontokat, és rögzíteni kell, hogy melyik kérdésre milyen választ adunk. Ha ezeket alaposan végiggondoljuk, a nagy buktatókat elkerüljük.

> **Megjegyzés:** Ha kikényszerítjük, hogy mindenhol alkalmazás-specifikus, XML vagy ASN.1 formátumú, géppel értelmezhető aláírási szabályzatokat hozzanak létre, azzal éppen gátoljuk az elektronikus aláírás terjedését. Nem az XML vagy ASN.1 formátumú aláírási szabályzat elkészítése a lényeg, hanem az, hogy még a rendszer indítása előtt végiggondoljuk és rögzítsük, milyen típusú aláírásokat szeretnénk, illetve tudunk befogadni.

12.2.5. Megfelelő aláírás-formátumot válasszunk!

> *„Pilátus feliratot is készíttetett, és rátétette a keresztre. Ez volt ráírva: A NÁZÁRETI JÉZUS, A ZSIDÓK KIRÁLYA. A zsidók közül sokan olvasták ezt a feliratot, amely héberül, latinul és görögül volt írva, ugyanis közel volt a városhoz az a hely, ahol megfeszítették Jézust. A zsidók főpapjai akkor szóltak Pilátusnak: "Ne azt írd: A zsidók királya! – hanem ahogyan ő mondotta: A zsidók királya vagyok." Pilátus így válaszolt: "Amit megírtam, megírtam."”*
>
> *– János 19,19-22*

Lehet, hogy ma CRL alapon ellenőrizzük az aláírást, 4 óra kivárási idővel, holnap pedig OCSP-t fogunk használni 5 másodperc kivárási idővel. Lehet, hogy ma alkalmazunk kivárási időt a szolgáltatói tanúsítványokra is, de holnap esetleg majd nem fogunk. A folyamatok változnak, a környezetünk változik, így lehet, hogy (az elektronikus aláírás területén is) sok ponton, és gyakran kell változtatnunk.

Az aláírt dokumentum és az aláírás formátuma (6.4. fejezet) viszont olyan, amit célszerű jól kitalálni. Amit megírtak, azt megírták, amit aláírtak, azt aláírták, azon soha többet nem lehet változtatni, legalábbis az aláírás hitelességének elvesztése nélkül. Esetleg át lehet konvertálni más formátumba, csatolni lehet hozzá az eredeti, aláírt formátumot, de minden konverzió kockázatot jelent, mert esetleg megváltozhat az, amit az aláíró eredetileg aláírt.

Próbáljunk olyan formátumot választani, amely nemcsak a ma, hanem a holnap igényeinek is meg fog felelni. A formátumokon is lehet módosítani, de talán ez a legfájdalmasabb. Ameddig egy rendszernek meg kell őriznie a dokumentumait, addig olvasni és értelmezni kell tudni a korábbi formátumokat, és ellenőrizni kell tudni a korábbi aláírásokat is. Több jó megoldás is létezik, de például ne válasszunk olyan formátumot, amit később elvileg sem lehet majd archiválni (8. fejezet).

12.2.6. Tisztázzuk a kapcsolatot a papír alapú rendszerekkel!

Ma még nincsenek tisztán elektronikusan hiteles rendszerek. Egy mai, elektronikus aláírásra épülő rendszerben számolni kell azzal, hogy papír alapú, hiteles dokumentumok is jelennek meg, és ezeket is fel kell tudni dolgozni. A rendszer tervezésekor fel kell mérni, hogy milyen papír alapú, hiteles dokumentumok érkezhetnek, és el kell dönteni, hogy milyen módon illesztjük be őket az elektronikus folyamatunkba. Erre számos lehetőség van, például:

- Felvesszük a kapcsolatot a papír alapú dokumentum forrásával, és megegyezünk benne, hogy nekünk elektronikusan aláírt, elektronikus dokumentumokat küldjön. Ez nem mindig reális lehetőség.

- A beérkező, papír alapú iratokat a 13/2005. IHM rendelet szerint digitalizáljuk, és elektronikus aláírással látjuk el. [79] E megoldásnak hátránya, hogy a beszkennelt iratot nehéz lesz géppel értelmezni.
- Egy kollégánk elolvassa a papír alapú iratot, és tesz egy elektronikusan aláírt, esetleg géppel is értelmezhető nyilatkozatot az eredeti irat tartalmáról. Az eredeti iratot ezt követően papíron megőrizhetjük, ha szükséges.

12.3. Felhasználási területek

12.3.1. Elektronikus számlázás

Akkor beszélünk elektronikus számláról, ha a kibocsátó a számlát elektronikusan hozza létre, és a számla elektronikusan jut el a befogadóhoz. Az így létrejött elektronikus számlát mind a kibocsátó, mind a befogadó elektronikusan őrzi meg.

Az elektronikus számla fogalmát az ÁFA törvény 175. §-a vezeti be. A törvény kimondja, hogy számlát tisztán elektronikus úton is ki lehet bocsátani, az erre vonatkozó követelmények két módon teljesíthetőek:

- az elektronikus aláírásról szóló törvény szerinti legalább fokozott biztonságú elektronikus aláírás és minősített időbélyegzés-szolgáltató által kibocsátott időbélyeg alkalmazásával;
- elektronikus adatcsere (EDI) rendszer segítségével. (Az EDI esetén további feltétel, hogy a kibocsátónak és a befogadónak előzetesen írásban meg kell állapodnia az elektronikus számlázásról, és havonta papíron összesítő jelentéseket kell készíteni.)

Elektronikus aláírással kibocsátott elektronikus számla esetén nem írja elő jogszabály, hogy a befogadónak előzetesen bele kell egyeznie, hogy számára elektronikus számlát bocsáthassanak ki (igaz, a gyakorlatban célszerű erről megállapodni a befogadóval, mert különben előfordulhat, hogy a befogadó azért nem tudja fogadni a számlát, mert nincsen számítógépe). Az elektronikusan aláírt számlák esetén a befogadónak nem kell bonyolult informatikai rendszerrel rendelkeznie (nem kell EDI üzeneteket fogadnia), csak a számlán elhelyezett elektronikus aláírást kell tudnia ellenőrizni. (E lépés általában egyszerű, ingyenes alkalmazásokkal elvégezhető.) Így elektronikusan aláírt elektronikus számla akár magánszemély számára is kibocsátható.

Az elektronikus számla bármilyen formátumú lehet, a jogszabályok sem a számla, sem a rajta lévő elektronikus aláírás formátumára nem tesznek megkötéseket. Ugyanakkor a befogadó el kell, hogy tudja olvasni a számlát, ezért célszerű ismert, elterjedt formátumokat alkalmazni. Általában az e-Szignó program által használt e-akta, vagy PDF formátumban szokás kibocsátani az aláírt számlákat.

A 46/2007. PM rendelet szerint, ha az elektronikus számlát be kell mutatni az adóhatóságnak, a bemutatási kötelezettségnek az elektronikus úton kibocsátott számlákra vonatkozó egyes rendelkezések értelmezéséről szóló APEH közleményben szereplő valamely formátumban kell eleget tenni. [2] (E közlemény szerint a számla XML, TXT, CSV, DBF, MDB, XLS vagy ún. print fájlformátumban mutatható be. Az XML formátum sémáját a közlemény 3. számú melléklete tartalmazza. A közleményben a print formátumra vonatkozóan leírt kikötések alapján e formátum ugyanaz, mint a TXT. A PDF formátum nem szerepel a közleményben.) A 46/2007. PM rendelet 3. §-a kimondja, hogy a bemutatáskor a számla kibocsátásakor érvényes APEH közlemény szerinti formátumnak kell eleget tenni, így a közleményben megfogalmazott szabályok esetleges jövőbeli változása a már kibocsátott számlákat nem érinti. Az elektronikusan aláírt, ügyfélnek kiküldött számla formátumára sem e rendelet, sem az APEH közlemény nem tesz megkötést, de amennyiben a számla nem az APEH közleményben szereplő formátumban van, ellenőrzéskor felmerülhet az a kérdés, hogy mi garantálja, hogy az APEH-nek bemutatott számlaadatok megegyeznek az ügyfélnek kiküldött, elektronikusan aláírt számla adataival.

Ezen túl, a 47/2007. PM rendelet által módosított 24/1995. PM rendelet szerint, az elektronikus számlát olyan számlázó programmal kell elkészíteni, amely kihagyás és ismétlés nélkül, folyamatosan biztosítja a sorszámozást, és e programhoz a számla kibocsátója olyan dokumentációval kell, hogy rendelkezzen, amely biztosítja a program működésének ellenőrizhetőségét, a program működésére, használatára vonatkozó részletes leírást, valamint a program készítője által a számla kibocsátójának címzett írásos nyilatkozatát arról, hogy az maradéktalanul megfelel a vonatkozó jogszabályi előírásoknak. (24/1995. PM r. 1/E § (1) a), 1/E. § (2)-(3) és 1/G §)

Adóellenőrzés esetén az adóhatóság rendelkezésére kell bocsátani a számla olvashatóságához szükséges eszközöket (ez lehet pl. e-Szignó program vagy PDF olvasó szoftver), a fenti dokumentációt, és meg kell adni a számla olvasásához szükséges eszközök használatához szükséges felvilágosítást. Ezen túl az adóhatóság jogosult próbaszámlázást is kérni. (24/1995. PM r. 1/H §)

A fent összefoglalt, kifejezetten elektronikus aláírásra vonatkozó követelményeken túl, az elektronikus számláknak is teljesíteniük kell a számlákra vonatkozó általános számviteli követelményeket.

12.3.1.1. Az elektronikus számla kibocsátójának feladatai

1. Létre kell hoznia a számlaadatokat tartalmazó fájlt (pl. XML vagy PDF formátumban).

2. Legalább fokozott biztonságú elektronikus aláírással és minősített időbélyeggel kell ellátnia a számlatartalmat. Ekkor jön létre az elektronikus számla.

3. El kell juttatni a számlát a befogadóhoz. A kibocsátó általában vagy e-mailben küldi el a számlát a befogadóhoz, vagy egy honlapot üzemeltet, amelyen a befogadó megtekintheti a számlát. Gyakori a két megoldás kombinációja is: a kibocsátó e-mailben értesíti a befogadót, hogy új számlája érkezett, és az értesítés egy linket tartalmaz a kibocsátó honlapján (portálján) található számlára.

4. A kibocsátónak meg kell őriznie a számlát.

12.3.1.2. Az elektronikus számla befogadójának feladatai

- Ha a számla befogadója nem köteles megőrizni a számlát (pl. magánszemély), akkor semmilyen teendője nincs az elektronikus számlával.

- Ha a számla befogadója számlamegőrzésre kötelezett, akkor az alábbiakban leírtak szerint kell megőriznie a számlát. Az elektronikus számlához kapcsolódó kontíradatokat az APEH közlemény szerint egyértelmű hozzárendeléssel, elválaszthatatlan módon, az utólagos módosítás lehetőségét kizárva kell csatolni a számlához. A közlemény szerint a csatolás részletes technikai kérdéseit (azon belül is kiemelten a hitelesség és a felelősség kérdését) a gazdálkodónak kell szabályoznia, a belső működési rendjében.

12.3.1.3. Az elektronikus számla megőrzése

A számviteli törvény 169. § (5) szerint az elektronikus formában kiállított bizonylatokat – és így az elektronikus számlákat – az elektronikus archiválásra vonatkozó jogszabály (amely a 114/2007. GKM rendelet) szerint elektronikus formában kell megőrizni.

Az elektronikus archiválásról szóló 114/2007. GKM rendelet szerint elektronikus adatra vonatkozó megőrzési kötelezettség teljesíthető, ha az adaton legalább fokozott biztonságú elektronikus aláírást és minősített időbélyegzőt helyezünk el. Tekintve, hogy az elektronikus aláírással létrehozott elektronikus számlán már eleve van fokozott biztonságú aláírás és minősített időbélyegző, ezeket nem kell külön elhelyezni rajta. A rendelet 4. § (4) csak akkor szab további feltételeket, ha a megőrzési kötelezettség hosszabb, mint 11 év, de a számlák – és így az elektronikus számlák – esetén a megőrzési kötelezettség csak 8 év, így a 4. § (4) követelményei e területre nem vonatkoznak.

A megőrzésre vonatkozó követelmények úgy is teljesíthetőek, hogy a megőrzésre kötelezett az elektronikus aláírásról szóló törvény szerinti minősített archiválás-szolgáltatót bíz meg a megőrzéssel. E megoldásnak jelentős előnye, hogy ekkor nem a megőrzésre kötelezettnek kell bizonyítania, hogy a követelményeknek eleget tesz, hanem ellenkező bizonyításig vélelmezni kell, hogy az archivált aláírás érvényes, és az archiválás „jól” történt. (Eat. 4. § (7))

12.3.2. Papír alapú számlák másodpéldányainak elektronikus megőrzése

Ha a befogadó papíron kapja meg a számlát, akkor nem elektronikus számláról, hanem hagyományos, papíralapú számláról beszélünk. Ugyanakkor a számítógéppel előállított papíralapú számlák esetén, a számla kibocsátójának nem muszáj kinyomtatnia és megőriznie a számla másodpéldányait, a másodpéldányok elektronikusan – elektronikus aláírással ellátva – is megőrizhetőek.

Erről a 24/1995. PM rendelet rendelkezik (a 47/2007. PM rendelet szerinti módosítás alapján). A 24/1995. PM rendelet 1/F. § (2) kimondja, hogy a "számítástechnikai eszköz útján előállított és papírra nyomtatott számla kibocsátónál maradó példánya – nyomtatás helyett – elektronikus adatállományként is megőrizhető", ha a nyomtatott változattal mindenben megegyező számlaképet legalább fokozott biztonságú elektronikus aláírással és időbélyegzővel látták el, és a vonatkozó jogszabályok szerint megőrzik.

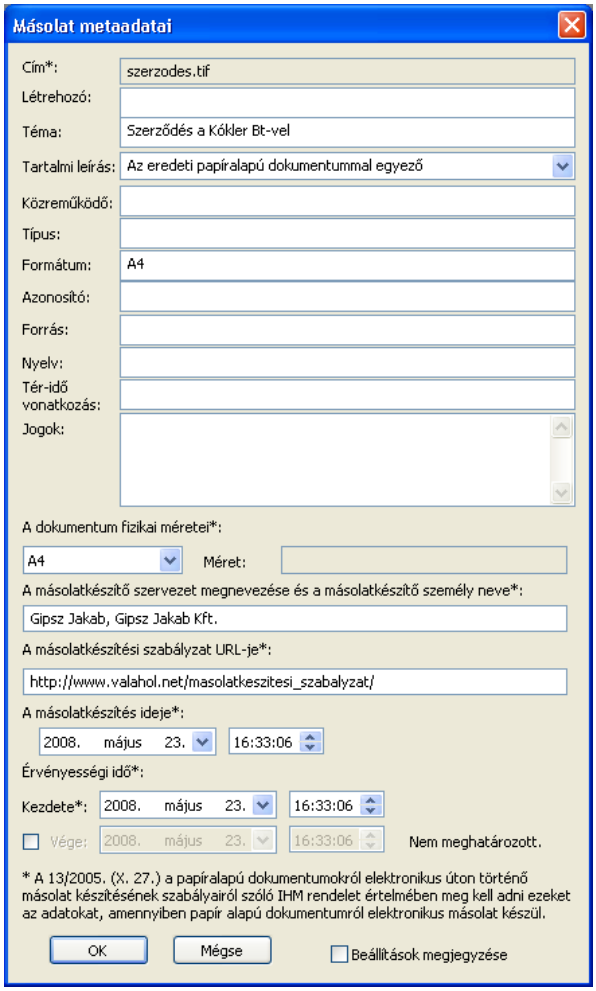
A nyomtatott változattal mindenben megegyező számlaképet kell aláírni és időbélyegzővel ellátni, így az nem megoldás, ha pl. egy adatbázisban lévő számlaadatokat látunk el aláírással. E követelmény leginkább aláírt PDF-fel vagy képfájllal teljesíthető.

Míg az elektronikus számlák esetén a számlákat tipikusan külön-külön kell aláírni és időbélyegezni (hogy a számlák elválaszthatóak legyenek egymástól, és több különböző ügyfélnek ki lehessen küldeni őket), a másodpéldányokat jellemzően nem kell elválasztani egymástól, így célszerű lehet őket együttesen aláírni és időbélyegzővel ellátni.

Habár az így kapott számlák papíralapú számlának – a számítástechnikai eszköz útján előállított és papírra nyomtatott számla elektronikus változatainak (24/1995. PM r. 1/F. § (3)) – minősülnek, a kibocsátónál csak elektronikusan állnak rendelkezésre, az elektronikus számlákhoz nagyon hasonló formátumban. Az elektronikus formában rendelkezésre álló számlákra, és az őket létrehozó, megjelenítő alkalmazásokra vonatkozó követelmények ezért sokszor egybeesnek az elektronikus számlákra (elektronikusan kiállított számlákra) vonatkozó követelményekkel. Így a papíralapú számlák elektronikus másodpéldányaira is érvényesek a folyamatos sorszámozásra, és a kiállító alkalmazás dokumentációjára vonatkozó követelmények (24/1995. PM r. 1/E § (1) a), 1/E. § (2)-(3) és 1/G §), illetve, hogy adóhatósági ellenőrzéskor a hatóság rendelkezésére kell bocsátani a számlák értelmezéséhez szükséges eszközöket, információkat stb. (24/1995. PM r. 1/H §)

12.3.3. Papír alapú dokumentumok elektronikus archiválása

A 13/2005. IHM rendelet lehetővé teszi, hogy papír alapú dokumentumról hiteles elektronikus másolatot készítsünk (lásd: 12.6. ábra). A rendelet szerint ha papír alapú dokumentumról hiteles elektronikus másolatot szeretnénk készíteni, a dokumentumot be kell szkennelni, ki kell egészíteni különféle adatokkal (pl. a dokumentum fizikai méretei, a másolatkészítés helye



12.6. ábra. **A 13/2005. IHM rendeletben előírt kötelező adatok megadása az e-Szignó program segítségével.**

és ideje, a másolatkészítő személy neve stb.), majd a kapott adatokat legalább fokozott biztonságú elektronikus aláírással és minősített időbélyegzővel kell ellátni. [79]

Az így kapott, elektronikus aláírással ellátott dokumentum a másolatkészítő fél aláírt nyilatkozata, miszerint az adott dokumentumot az adott időpontban beszkennelte, és a kapott elektronikus dokumentum képe megegyezik az eredeti papír alapú dokumentuméval. Az elektronikus aláírás és az időbélyegzés technológiája biztosítja, hogy ilyen nyilatkozatot nem lehet más nevében készíteni, és nem lehet visszadátumozni sem. Ugyanakkor a kapott elektronikus dokumentum hitelessége attól függ, hogy mennyire bízunk meg a másolatkészítő félben.

12.8. Példa: *Manfréd készít egy papír alapú szerződést, amely szerint megvette Alajos házát 10 millió forintért. Aláírja a szerződést, és odafénymásolja Alajos aláírását is. Az így kapott hamisítványt a 13/2005. IHM rendelet szerint*

digitalizája, majd az eredetit megsemmisíti. Ha később ezzel kívánja igazolni, hogy a ház az övé, Alajos várhatóan tiltakozik majd, és azt állítja, hogy az eredeti szerződés soha nem létezett. Manréd nagy valószínűséggel nem tudja majd érvényesíteni követelését.

Nincs, és várhatóan nem is lesz olyan jogszabály, amely általánosan kimondaná, hogy minden így digitalizált okirat elektronikus változata egyenértékű a papír alapú eredetivel, és a papír alapú dokumentum minden esetben nyugodtan megsemmisíthető. Előfordulhat, hogy egy vitás esetben a bíróság a papír alapú dokumentum bemutatását kéri. Ugyanakkor sok esetben nem az aláírás hitelessége a vita tárgya, és sok esetben a 13/2005. IHM rendelet szerinti hiteles másolat is elegendő. A gyakorlatban célszerű mérlegelni az eredeti okirat megsemmisítésének kockázatát, és ez alapján választani az alábbiak közül:

- Az eredeti okiratot megőrizzük, de napi szinten egy beszkennelt, aláíratlan, nem hiteles változatával dolgozunk. Ennek jelentős kockázata lehet, mert ekkor az érdemi döntésekt egy nem hiteles változat alapján hozzuk meg.
- Az eredeti okiratot megőrizzük, de a napi munkát a 13/2005. IHM rendelet szerint digitalizált változatával végezzük. Ekkor már hiteles okirat alapján hozzuk a döntéseket.
- Az eredeti okiratot nem semmisítjük meg, de valahol messze, egy olcsó vidéki raktárban tároljuk, és nem törekszünk arra, hogy könnyen, hatékonyan előkereshető legyen. A napi munkát a 13/2005. IHM rendelet szerint digitalizált változatával végezzük. Ezáltal a papír alapú megőrzés költségeit jelentősen csökkentettük. Ha a hiteles dokumentumra van szükségünk, a 13/2005. IHM rendelet szerint digitalizált változatot mutatjuk be. Ha egy hatóság látni szeretné, hogy az eredeti okiratokat valóban megőrizzük, meg tudjuk mutatni a raktárat. Ha valaki egy konkrét okirat eredeti példányát követeli rajtunk, akkor – idővel – elő tudjuk keresni a megfelelő dokumentumot.
- Dönthetünk úgy is, hogy az eredeti dokumentumot megsemmisítjük, és csak a 13/2005. IHM rendelet szerint digitalizált változatát őrizzük meg. A legtöbb esetben ez elegendő[4] kell, hogy legyen, de ha valaki kifejezetten az eredeti, papír alapú okiratot követeli rajtunk, akkor nehéz helyzetbe kerülhetünk.

12.3.4. Felhasználó-azonosítás

Az elektronikus aláírásról szóló törvény szerint az aláírói tanúsítványhoz tartozó magánkulccsal csak aláírni szabad, sem dekódolásra, sem challenge and response autentikációra nem szabad használni. [180]

[4]Sokan nem ismerik a 13/2005. IHM rendeletet, és nem tudnak mit kezdeni egy digitalizált dokumentummal, így nem fogadják el, ha ilyet szeretnénk benyújtani. Gyakran ez a tudatlanság gátolja a technológia használatát.

> *Eat. 13. § „ (4) Az aláíró az aláírás-létrehozó adatot kizárólag az aláírás létrehozására használhatja, betartva a tanúsítványban jelzett esetleges egyéb korlátozásokat is. ”*

Ezt időnként próbálják úgy megkerülni, hogy belépéskor aláíratnak egy olyan dokumentumot, miszerint „alulírott XYZ be szeretnék lépni erre és erre az oldalra, egyedi azonosító: ...”. Egy időben az Ügyfélkapu is használt ilyen megoldást. Ekkor az aláírás nem egy hosszú távon letagadhatatlan dokumentum hitelességét védi, hanem azt igazolja, hogy az aláíró birtokolja a tanúsítványhoz tartozó magánkulcsot. A szerver eldönti, hogy „ismeri’-e az aláíró tanúsítványát, és ha igen, beengedi az aláírót.

Ha tudjuk, hogy az adott aláíró tanúsítvány kihez tartozik, dönthetünk úgy, hogy az aláírást az adott személy készítette; ezzel semmi probléma nincs. Az Eat-ben valóban szerepel néhány helyen az aláírás kifejezés mellett az azonosítás szó is, de ezzel óvatosan kell bánni, az aláíró tanúsítványból jellemzően nem dönthető el, hogy kicsoda az aláíró. Sőt, léteznek az ún. álneves tanúsítványok, ahol kifejezetten az a cél, hogy magából a tanúsítványból ne lehessen megmondani, hogy ki az aláíró. Az elektronikus aláírás elsősorban arra szolgál, hogy *vita esetén utólag* el lehessen dönteni, hogy ki készítette az aláírást, és nem arra, hogy ezt bárki bármikor megmondhassa.

> *Eat. 2.§ „ 6. Elektronikus aláírás: elektronikusan aláírt elektronikus dokumentumhoz azonosítás céljából logikailag hozzárendelt vagy azzal elválaszthatatlanul összekapcsolt elektronikus adat. ”*

Ellenben, ha az aláírás egyetlen szerepe az, hogy megbizonyosodjuk róla, hogy az aláíró birtokolja a magánkulcsot, akkor lényegében kihívás és válasz alapú autentikációról van szó (ahol a kihívás a dokumentumban szereplő egyedi azonosító, a válasz pedig az aláírás). Ha a dokumentumot korrektül íratjuk alá a felhasználóval – pl. aláírás előtt megmutatjuk, aláírást követően letölthetővé tesszük – akkor jogilag valószínűleg nem lehet belekötni a megoldásba, de olyanra használtuk az aláírást, amire nem való. Lényegében challenge and response autentikációt valósítottunk meg, csak ezt nem egy szabványos technológiával (pl. SSL, IPSEC stb), hanem egy saját protokoll szerint tettük. Biztonsági szempontból sem szerencsés, ha a felhasználó megszokja, hogy folyton be kell gépelnie a PIN kódját; az a helyes, ha az aláíró akkor készít aláírást, amikor egy dokumentum hosszú-távú hitelességét szeretné biztosítani.

Egyes területeken azért szokás mégis ezt a megoldást használni, mert az aláíró tanúsítványokhoz, illetve az aláírásokhoz kapcsolódik jogkövetkemény, míg az SSL autentikációhoz nem. *Általában sokkal egyszerűbb SSL-t használni beléptetéshez, és ekkor a felhasználónak sem kell azzal foglalkoznia, hogy mit kódol a magánkulcsával.*

12.3.5. Dokumentum-kezelés

Bármilyen elektronikus dokumentum-kezelő rendszerben használhatnánk elektronikus aláírást a kézzel írott aláírás helyett. Az elektronikus dokumentumok könnyebben továbbíthatóak, jobban menedzselhetőek, így az elektronikus dokumentumokkal megtakarítás érhető el.

Egyik lehetőség, hogy egy elektronikus dokumentum-kezelő rendszer egyetlen helyen kezel aláírásokat. Itt ellenőrzi a bejövő dokumentumokon elhelyezett elektronikus aláírást, majd iktatja e dokumentumokat, illetve itt látja el aláírással a rendszerben szereplő dokumentumokat – a rendszer nevében. Így aláírásra kerülhet egy ügy összes irata, ha az ügy lezárul, illetve aláírásra kerülhet a rendszerből kimenő összes dokumentum. Ilyenkor általában automatizmus által készített fokozott biztonságú elektronikus aláírás jön létre.

Másik lehetőség, hogy a rendszer felhasználói a saját kártyáikkal készítenek aláírást, pontosan úgy, ahogyan ezt a papír alapú munkafolyamatokban tették. Ez jelentősen nehezebb feladat, mert az egyes munkafolyamatok alapos ismeretét jelenti.

Szintén figyelembe kell venni, hogy a papíroknak vannak bizonyos előnyei. A papír kézzel fogható, bármilyen eszköz nélkül olvasható, rá lehet írni, könnyen hordozható, és a rávitt információt hosszú távon, változatlan formában megőrzi.

Előfordulhat, hogy ügyvitelünk során például munkatársaink ráírnak egyes papírokra, és ez megnehezíti a folyamat ilyen módon történő elektronizálását. Hasonló problémát jelenthet, ha munkatársaink rendszeresen átviszik egyik szobából a másikba a papírokat, és ez elektronikusan nem oldható meg egyszerűen és rugalmasan, például egy elektronikus jogosultságkezelő rendszer miatt[5].

Egy központi aláíró automatizmussal könnyebb elindulni, és ez már jelentős lépést jelent az elektronikus aláírás bevezetése felé. Ha azt szeretnénk, hogy az egyes felhasználóink egyenként írjanak alá, az a folyamatok alaposabb ismeretét igényli, ami sok helyen, sok esetben nincsen meg.

12.3.6. Veszélyes környezetben való munkavégzés dokumentálása

Léteznek olyan rendszerek, ahol egymástól távoli pontokon kell gyors döntéseket hozni, amelyeknek jelentős anyagi vonzata van, illetve emberéletre is hatással lehetnek. Ilyen esetekben a kommunikáció többnyire – a gyorsaság miatt – telefonon zajlik, az írásos utasítások dokumentálásra pedig sokszor csak utólag kerül sor. Egyrészt a telefonon, szóban elhangzottakat nagyon könnyű félreérteni. Másrészt e megoldás csak addig működőképes, amíg minden rendben történt. Ha baj történt, baleset történt, vagy megsérült valaki, senki nem akarja majd utólag aláírni az ezt okozó utasítást, és magára vállalni a felelősséget.

[5]Ez egyúttal azt is jelenti, munkatársaink a munkájuk elvégzéséhez rendszeresen megkerülik az elektronikus jogosultságkezelő rendszert, ami egyáltalán nem biztos, hogy szerencsés.

Megoldást jelenthet, ha ilyen környezetben elektronikus aláírást használunk. Ekkor az elektronikus aláírással ellátott utasítások gyorsan továbbíthatóak, és a munkát végző felek az eredeti, hiteles dokumentumok alapján hoznak döntéseket. Nem válik el az az információ, ami alapján dolgozunk attól, ami a hitelességet biztosítja.

Figyelembe kell venni, hogy az elektronikus aláírással kapcsolatos szolgáltatásokat nyújtó szolgáltatók rendelkezésre állása jellemzően 99,9%. Míg ez egy irodai környezetben többnyire bőven elég, kritikus rendszerek esetén gondolni kell rá, hogy mind e szolgáltatások, mind maga az Internet kieshetnek, és gondoskodni kell tartalék megoldásról. A jelenleg használt telefon és utólagos dokumentálás betöltheti ezt a szerepet.

12.3.7. Szerződéskötés

„I am altering the deal. Pray I don't alter it any further."
(Módosítom a megállapodásunkat. Ne kívánd, hogy még jobban módosítsam.)

– A birodalom visszavág; Darth Vader szavai

A kézzel írott aláíráshoz hasonlóan elektronikus aláírással is hitelesíthetünk szerződéseket. Az Eat. 3. § (2) bekezdése néhány területen – elsősorban ahol egymással közeli kapcsolatban élő emberek közötti jogviszonról van szó, akik hozzáférhetnek egymás aláírás-létrehozó eszközéhez és PIN kódjához – kizárja az elektronikus aláírás használatát, de minden más területen használhatjuk e technológiát.

Szerződést többnyire szóban is lehet kötni, de néhány speciális esetben a jog előírja, hogy kötelező írásba foglalni a szerződést. Akkor is célszerű leírni a szerződést, ha ez nem előírás, mert így később nem lehet abból nézeteltérés, hogy pontosan miben állapodtak meg a felek egymással.

Az elektronikus aláírással való szerződéskötés egyelőre nem terjedt el, elsősorban azért, mert ritka, hogy a felek már rendelkeznének elektronikus aláírással.

A következő okok miatt lehet értelme elektronikus aláírást használni szerződéskötésre:

- Jogszabáy előírja, hogy írásba kell foglalnunk a szerződést, és ennek meg akarunk felelni, de nem szeretnénk sok papír alapú szerzdődést tárolni. Ekkor bármilyen, legalább fokozott biztonságú elektronikus aláírás megfelelő. Megoldást jelenthet erre a problémára, ha az a szerződő fél, akinek érdeke az elektronikus „megszervezi", hogy a másik fél rendelkezzen elektronikus aláírással.

  12.9. Példa: *Alajos egy nagy cég munkatársa, sok beszállítóval áll kapcsolatban. A beszállítók tipikusan egy webes portálon lépnek be Alajos cégének rendszerébe, jellemzően nem találkoznak Alajossal. A beszállítókkal sok szerződést kell kötni, és Alajos nem szeretné ezt papíron, postán intézni.*

Alajos „meggyőzi" a beszállítóit, hogy a továbbiakban elektronikus aláírással szerződjenek. Összegyűjti az elektronikus aláírás igényléseiket, megszerzi írásos jóváhagyásukat, hogy a tanúsítványigénylés során eljárjon a nevükben, eljuttatja az igényléseket egy hitelesítés-szolgáltatónak, aki kibocsátja a tanúsítványokat. (Ezek jellemzően személyes találkozás nélkül kibocsátott tanúsítványok.)

Lényeges, hogy több segítséget nyújt Alajos a beszállítóinak, annál nagyobb kontrollal rendelkezhet az igénylések, illetve a beszállítók magánkulcsai felett. Habár az így aláírt szerződések írásba foglaltnak minősülnek, vita esetén lehet, hogy az aláírások nem állnak meg bíróság előtt.

Ennek szélsőséges esete, ha Alajos cége – azaz az egyik szerződő fél – a hitelesítés-szolgáltató regisztrációs szervezetévé válik. Ekkor az Alajos és egy beszállító közti vitában a beszállító akár azt is állíthatja, hogy ő soha nem igénylet tanúsítványt, hanem Alajos találta ki az egészet.

- Azt szeretnénk, hogy a szerződések biztosan megálljanak bíróság előtt. Ekkor célszerű minősített aláírást használni, és célszerű azt biztosítani, hogy a hitelesítés-szolgáltató független fél marad. (Így például nem szerencsés, ha az egyik fél maga a hitelesítés-szolgáltató vagy annak regisztrációs szervezete.)

  Itt is előfordulhat, hogy az egyik fél kikényszeríti, hogy a másik minősített aláírással kell, hogy rendelkezzen a szerződéskötéshez, de a minősített aláírás költsége várhatóan akkor térül meg, ha az aláírást a későbbiekben más célokra is használják.

- Előfordulhat, hogy nem mindkét fél látja el elektronikus aláírással a szerződést. Például, általános szerződési feltételek esetén kezd terjedni az a megoldás, hogy egy cég elektronikus aláírással látja el általános szerződési feltételeit, és így teszi őket elérhetővé honlapján. Ez erősítheti benne a bizalmat, mert a szerződéseket letöltő, és eltevő ügyfél így aláírt bizonyítékkal rendelkezhet arról, hogy egy adott pillanatban melyek voltak a cég általános szerződési feltételei. [116]

12.3.8. Szerzői jogok védelme

Az elektronikus alárás célja elsősorban nem a szerzői jogok védelme, de e technológia akár erre a célra is használható. Szerzői jogi vitákban gyakran kell azt bizonyítani, hogy a kérdéses mű nálunk volt jelen először. Ezt megtehetjük elektronikus aláírással, illetve az aláíráson lévő időbélyeggel. [41]

12.10. Példa: *Manfréd megjelentet egy cikket egy tudományos folyóiratban. Alajos azt állítja, hogy a cikket ő írta, Manfréd pedig ellopta tőle. Megpróbálja*

bebizonyítani, hogy a cikk már az ő birtokában volt, mielőtt az megjelent a folyóiratban. Segítségére lehet, ha rendelkezésére áll egy általa aláírt és időbélyegzett változat, amelyen az időbélyeg a folyóirat megjelenését megelőzően készült.

Ilyenkor az időbélyeg igazolja, hogy a dokumentum egy adott időpontban már létezett, és az elektronikus aláírás igazolja, hogy a dokumentum akkor az aláíró birtokában volt. (Szigorúan véve az aláírás csak azt igazolja, hogy a dokumentum *lenyomata* volt az aláíró birtokában.) Önmagában nem elegendő, ha csak időbélyeget helyezünk el a dokumentumon.

12.11. Példa: *Alajos ír egy szimfóniát, és a kottáját időbélyeggel látja el, hogy később igazolhassa, hogy a szimfónia az ő műve. Megmutatja a szimfóniát barátjának, Manfrédnak, aki ellopja tőle. Manfréd felkeres egy lemezkiadót, és a szimfónia az ő nevén jelenik meg, ezáltal híres zeneszerző és elismert, gazdag ember lesz.*

Alajos felkeresi Manfrédot, és követeli, Manfréd ismerje el, hogy ellopta a szimfóniát, és térítse meg az Alajosnál keletkezett kárt. Megfenyegeti, hogy bepereli Manfrédot, ha nem tesz eleget követeléseinek, és megmutatja, hogy rendelkezik a szimfónia időbélyegzett változatával. Manfréd kidobja Alajost, aki bírósághoz fordul.

Mielőtt az ügy bíróság elé kerül, Manfréd nyilvánosan bejelenti, hogy megvádolták, és – bizonyítandó, hogy a szimfóniát ő írta – honlapján nyilvánosságra hozza az időbélyegzett változatot, amit Alajos mutatott neki. Azt állítja róla, hogy az időbélyeget ő helyezte el rajta.

A bíróság előtt Manfréd nem kérdőjelezi meg az időbélyeg valódiságát, csak annyit állít, hogy az időbélyeget ő kérte le a szimfóniára, és az Alajos által bemutatott „bizonyíték" valójában Manfréd honlapjáról származik. Mivel az időbélyegből nem állapítható meg, hogy ki kérte, Alajos nehéz helyzetbe kerül.

Aki először mutatja fel az időbélyeget, az állíthatja azt, hogy a többiek nem tudnák felmutatni. Aki másodjára mutatja fel ugyanazt az időbélyeget, az nem tudja bizonyítani, hogy az időbélyeg nála korábban is jelen volt, mint az első felmutatónál.

Lényeges, hogy az aláírás és az időbélyeg is csak egy bizonyíték a sok közül. Ha valaki be tud mutatni egy aláírt és időbélyegzett változatot, az nem azt bizonyítja, hogy ő készítette művet, csak annyit jelent, hogy az időbélyegben szereplő időpontban a mű már az ő birtokában volt. Lehet, hogy valaki korábbi időbélyeget tud felmutatni, de az is lehet, hogy nem vitatják, hogy nála volt: például lehet, hogy hivatalosan megkapott egy művet bírálatra vagy lektorálásra.

12.12. Példa: *Manfréd begépeli a Háború és béke című könyvet a számítógépébe, majd elektronikus aláírással és időbélyeggel látja el. Könnyen lehet, hogy sem*

Tolsztoj, sem más nem tud nála korábbi időbélyeget felmutatni a művön. Ennek ellenére, senki nem fog hinni Manfrédnek, ha azt állítja, ő írta a Háború és békét.

12.3.9. Biztonságos kézbesítés

„A mátraszentannai postahivatal ablaka az udvarra nyílt. Alatta, az íróasztaltól kartávolságnyira esővizes hordó állt, tele megzöldült, megkocsonyásodott vízzel. Gyuri atyus, miután gondosan átnézte a napi postát, ide dobálta a megsemmisülésre ítélt leveleket.
Ide került, ugyanazon a napon, egy aranyszegélyű meghívó, mely Cipriani professzort és nejét a kormányzói pár gardenpartyjára invitálta, valamint az a vöröskeresztes sürgöny is, mely Tót Gyula halálhírét hozta. Ezzel nemcsak a professzorék kaptak egy fricskát, hanem a kedves Tótékat is sikerült megkímélni a gyászhírtől. A világ egyensúlya helyreállt."

– Örkény István, Tóték

A feladó elküld a címzettnek egy elektronikus üzenetet. Az üzenet az Interneten halad keresztül, így út közben elveszhet, megsérülhet, de az is lehet, hogy csak késve ér célba. A feladó és a címzett esetleg ellenérdekeltek. Előfordulhat, hogy a címzett nem akarja átvenni az üzenetet, vagy később szívesen letagadná, hogy valóban átvette azt. Szintén előfordulhat, hogy a feladó azt állítja, hogy ekkor és ekkor elküldött valamit a címzettnek, de azt nem akkor, vagy esetleg soha nem küldte el.

12.13. Példa: *Alajos meteorológus. Kiszámítja, hogy néhány óra múlva az évszázad vihara fog kitörni a városban. Tudja, hogy Bendegúz épp akkorra szervezett tűzijátékot, amelyet rengeteg ember néz majd a szabadban. Alajosnak figyelmeztetnie kell Bendegúzt, mielőtt elszabadul a pokol. Elküldte Alajos a figyelmeztetést? Időben küldte el? Megérkezett Bendegúzhoz az üzenete? Meggyőződött róla Alajos, hogy Bendegúz valóban megkapta?*

12.14. Példa: *Alajos egy közbeszerzési eljárásban ajánlatot tesz, az ajánlatot Bendegúznak küldi el. Bendegúz nem veszi figyelembe Alajos ajánlatát, arra hivatkozik, hogy az nem, illetve csak a határidő után érkezett meg hozzá. Valóban elküldte Alajos az ajánlatot? Időben küldte el? Időben érkezett meg Bendegúzhoz? Ha igen, tudja ezt Alajos bizonyítani? Ha nem, tudja ezt Bendegúz bizonyítani?*

Az Interneten küldött üzenetek elvileg elveszhetnek vagy késlekedhetnek, de azok nagyon nagy része szinte azonnal célba ér. A megbízhatatlan Interneten keresztül is kommunikálhatunk megbízhatóan és biztonságosan – megőrizve az Internet nyújtotta gyorsaságot –, ha a célba ért üzeneteinkről igazolni tudjuk, hogy azok valóban célba értek. A biztonságos kézbesítésnek

nem az a célja, hogy mindig minden üzenet érkezzen meg (ez irreális célkitűzés lenne), hanem az, hogy a fenti vitás kérdések nagyon egyszerűen tisztázhatóak legyenek.

A „biztonságos kézbesítés" azt jelenti, bizonyítható, hogy a feladó valóban elküldött egy üzenetet, illetve a címzett valóban átvette azt. Lényeges, hogy e bizonyítás a kézbesítés során létrejövő *bizonyítékokra* épüljön, és e bizonyítékokat utólag *ne lehessen módosítani*, és *bíróság előtt is meg kell, hogy állják a helyüket.* [131]

E probléma megoldásához egy *megbízható harmadik fél*, azaz egy *kézbesítési szolgáltató* bevonása szükséges. A következő követelményeket fogalmazhatjuk meg a kézbesítési szolgáltató (vagy szolgáltatók) biztonságával kapcsolatban:

1. A kézbesítésről szóló *bizonyítékokat még a kézbesítési szolgáltató se módosíthassa utólagosan.*

   E követelmény az elektronikus aláírásról szóló törvényben szereplő technológiákkal (elektronikus aláírás és időbélyegzés együttes alkalmazásával) teljesíthető a legegyszerűbben és legolcsóbban: A címzett elektronikus aláírással ír alá egy tértivevényt, amelyben igazolja, hogy átvette az adott üzenetet. Ha a címzett nem képes vagy nem hajlandó átvenni az üzenetet, a kézbesítési szolgáltató állít ki erről egy elektronikusan aláírt igazolást. (E megoldással nemcsak az igazolható, hogy a címzett átvett egy üzenetet, az is bizonyítható, hogy pontosan milyen tartalmú üzenetet vett át.)

   A kézbesítés ekkor annyit jelent, hogy a feladó elküldi az üzenetet a címzettnek, cserébe megkapja a tértivevényt, és e cserét „fair" módon bonyolítják le. A „fair" szakkifejezés azt jelenti, hogy vagy mindkét fél hozzájut a kívánt információhoz (a címzett az üzenethez, a feladó a tértivevényhez), vagy egyikőjük sem. Így a címzett nem kaphatja meg az üzenetet, amíg át nem adta az aláírt tértivevényt, de csak akkor kell átadnia a tértivevényt, amikor már megkapta az üzenetet. Papír alapú iratok esetén e fair cserét a két irat egyidejű átadásával szokás megoldani. Elektronikus esetben ez nem ennyire egyszerű, mert ekkor a feladó és a címzett egymástól távol helyezkednek el, és csak Interneten vannak kapcsolatban. Ekkor a fair csere nem egyidejű átadással, hanem például kriptográfiai módszerekkel oldható meg; a szakirodalomban számos módszer szerepel a fair cserék lebonyolítására. [101]

2. Ne kerülhessen a kézbesítési szolgáltató a mindentudó „nagy testvér" szerepébe, azaz:

   a. *A kézbesítési szolgáltató ne ismerhesse meg az üzenetek tartalmát.* Gondoljunk bele, mi történne, ha valaki egy központi helyen beleolvashatna például bármely közbeszerzési eljáráshoz beérkező bármely ajánlatba. Még ha nem is történik visszaélés, már az is rontja a bizalmat, ha ennek műszakilag megvan az elvi lehetősége.

Egyik lehetőség, hogy a kézbesítési szolgáltatón titkosított üzenet halad keresztül, olyan módon titkosítva, hogy azt még maga a kézbesítési szolgáltató sem tudja visszafejteni. Másik lehetőség, hogy olyan kézbesítési megoldást választunk, amely szerint a kézbesítési szolgáltató csak az üzenettel kapcsolatos adatokat kezeli, de maga az üzenet el sem jut a kézbesítési szolgáltatóhoz. [101]

b. *Senki ne térképezhesse fel, hogy ki, mikor és kinek küld üzenetet.* Meglepő, mennyi információ nyerhető ki önmagában abból, hogy valaki kiknek és mikor üzen. Előfordulhat, hogy pusztán a fenti információkból, az üzenetek tartalmának ismerete nélkül következtetni lehet például valakinek jövőbeli terveire, vagyoni helyzetére, egészségügyi állapotára stb. Egy dolog, ha egy konkrét üzenetből kinyerhető, hogy azt ki, mikor és kinek küldte, ehhez képest minőségileg mást jelent, ha mindez mindenkiről egyetlen központban könnyen összegyűjthető.

3. *A kézbesítési szolgáltatás ne jelentsen szűk keresztmetszetet.* Országos szinten óriási mennyiségű üzenetet küldünk egymásnak. Nem szabad, hogy ezek leterheljék a kézbesítési szolgáltatást. Ezen kívül egy kézbesítési szolgáltató leállása, kiesése miatt nem állhat le minden ügyintézés az országban.

A biztonságos elektronikus kézbesítés alapvetően a tértivevényes postai levelek kiváltására alkalmas, náluk jelentősen gyorsabb és olcsóbb alternatívát kínálnak. Egyrészt egy kiküldött e-mail szinte azonnal eljut a címzetthez, aki azonnal dolgozhat a hiteles elektronikus dokumentummal, illetve akár azonnal megkaphajtuk a kézbesítés tényét igazoló elektronikus tértivevényt. Másrészt az elektronikusan aláírt levelek és tértivevények jelentősen olcsóbbak a tértivevényes postai leveleknél. Nagy mennyiségű levél esetén az aláírásokhoz kapcsolt időbélyeg jelenti az egyetlen érdemi költséget, és ez nagyságrendileg századannyi, mint egy postai tértivevényes levél feladása.

A jelen jogszabályi környezetben egyetlen olyan funkció van, amely elektronikusan nem teljesíthető: a kézbesítési vélelem elektronikusan még nem állítható be[6], így ha a címzett nem hajlandó átvenni a levelet, akkor elektronikusan nem tudjuk egyértelműen igazolni, hogy a levelet valóban elküldtük.

Célszerű feltérképezni azon pontokat, ahol a ma tértivevényes postai leveleket váltó felek elektronikusan aláírt leveleket és elektronikus tértivevényeket is válthatnak. E pontokat célszerű elektronizálni, mert néhány ezer tértivevényes levél esetén az elektronikus út már milliós nagyságrendű megtakarítást jelent. Ha egy levelet egyáltalán nem sikerül elektronikusan kézbesíteni (azaz nem érkezik vissza az aláírt tértivevény), akkor és csak akkor célszerű a postai kézbesítést használni.

[6]Leszámítva a hivatalos iratok elektronikus kézbesítéséről és az elektronikus tértivevényről szóló 2009. évi LII. törvény szerinti Állami Elektronikus Kézbesítési Szolgáltató használatát, amely nem jött létre, így a gyakorlatban nem használható.

12.4. Gyakori kérdések

„Deep Thought: Okay. The answer to the ultimate question of life, the universe, and everything is... (wild cheers from audience, then silence)
Deep Thought: 42."

(Bölcs Elme: A válasz az Élet, a Világmindenség és Minden nagy kérdésére... (a közönség ujjong, majd néma csend)
Bölcs Elme: 42.)

– *Douglas Adams, Galaxis útikalauz stopposoknak*

12.4.1. Aláíráshoz intelligens kártyát használjak?

Az intelligens kártya használatának fő előnye, hogy az aláírásra használt magánkulcsot nem lehet kinyerni belőle. Így az aláíró biztos lehet benne: ahol a kártyája, ott, és csakis ott van az ő magánkulcsa is. Ameddig a kártya az aláíró zsebében van, addig biztos lehet benne, hogy senki nem él vissz az ő magánkulcsával.

Az intelligens kártya egyik fő hátránya az, ami az előnye is. A magánkulcsot nem lehet kinyerni a kártyából, így biztonsági másolatot sem lehet készíteni róla. Ha az aláíró elutazik a nyaralójába, és a kártyáját otthon hagyja, akkor nyaralójában nem tud aláírást készíteni (hacsak nincs ott egy másik kártyája).

Az intelligens kártyák másik fő hátránya a különböző operációs rendszerek és meghajtóprogramok kompatibilitási kérdésiből ered. A legtöbb kártyának csak Windows platformra van támogatása, és a meghajtóprogramjaik gyakran összeakadnak a Windows-zal vagy más hardver eszközökkel. Van olyan kártya, amelynek „megbolondul" a meghajtóprogramja, ha egyszerre két kártyát csatlakoztatunk egy számítógéphez, és nagyon kevés meghajtóprogram tud korrektül működni, ha ugyanarra a számítógépre másik típusú kártyát is telepítettek. (Ilyen esetekben sajnos nagyon nehéz kideríteni, hogy melyik kártya melyik meghajtóprogramja a bűnös.)

A tapasztalat azt mutatja, hogy a felhasználók nagyon nehezen tudnak vigyázni a „szoftveres" tanúsítványaik magánkulcsaira. Ekkor általában nem is csak egy fájlra kell vigyázniuk, hanem valamely szoftver (pl. Windows, Mozilla) tanúsítványtárában elhelyezett tanúsítványra. Csak hozzáértő felhasználók tudják biztosítani, hogy a magánkulcsot valóban bizalmasan kezelik (más nem férhet hozzá), ugyanakkor nem veszítik el, nem semmisítik meg véletlenül. (Például a számítógép újratelepítése esetén külön gondolni kell a Windows tanúsítványtárában lévő magánkulcsok mentésére.) Ha a magánkulcsról mentés készül, gondolni kell rá, hogy más ne férhessen hozzá a mentésben lévő magánkulcshoz. Ha a magánkulcsot jelszóval védve[7]

[7]Ekkor az adott jelszóból képzett szimmetrikus kulccsal titkosítva mentjük el a magánkulcsot.

mentjük el, meg kell őrizni a jelszót, mert a jelszó elfelejtése a magánkulcs végérvényes megsemmisülését jelentheti. Megjegyezzük, a felhasználók többsége nem tud erős jelszakavat megfelelően biztonságosan kezelni.

Ha a felhasználónak nem csak egy magánkulcsa és tanúsítványa van – mert például van külön aláíró, titkosító és autentikációs – akkor több magánkulcsról kell gondoskodnia.

A legtöbb felhasználó már rendelkezik bankkártyával, így kártyák esetén kialakult kultúrája van, hogy hogyan kell vigyázni egy kártyára és a hozzá tartozó PIN kódra, illetve, hogy a kártya elvesztése esetén a kártyát le kell tiltani.

Az intelligens kártya tekintetében a következőt javasoljuk:

- *Ha az aláíró vagy felhasználó természetes személy (ember), akkor egyértelműen javasoljuk a kártyák használatát*, mert a kártyák egyszerűbbé, érthetőbbé, kézzel foghatóbbá teszik az elektronikus aláírás és a PKI használatát.
- *Ha a felhasználó egy automata, akkor a kártyák valószínűleg felesleges korlátozást jelentenek.* Ha az automata egy szervezet biztonságos szervertermében van, akkor a szervezet valószínűleg más eszközökkel is meg tudja védeni a magánkulcsot. Ekkor különösen fontos, hogy a kulcsról mentés készüljön, mert például egy elektronikus számlázáshoz használt kulcs megsemmisülése esetén leállhat a szervezet számlázórendszere. Ha a kulcs bizalmasságát kriptográfiai célhardverrel is védeni szeretnénk, akkor használjunk HSM-et.

Ha minősített elektronikus aláírást szeretnénk készíteni, ahhoz mindenképpen szükséges speciális hardver eszköz, egy biztonságos aláírás-létrehozó eszköz. Ez általában intelligens kártya.

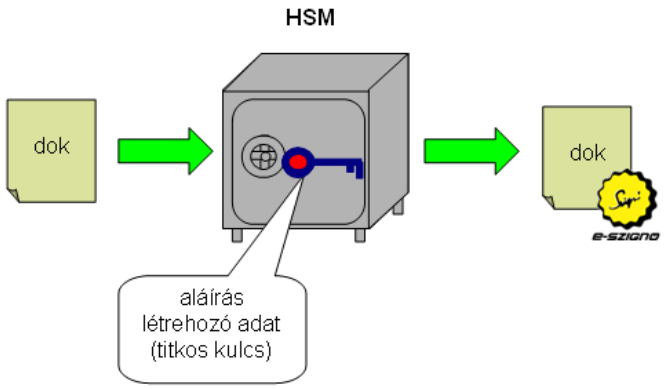
12.4.2. Minősített vagy fokozott biztonságú aláírást használjak?

Természetes személyek esetén az intelligens kártya használatát javasoltuk. Ha valaki már intelligens kártyát használ, általában kevés oka lehet rá, hogy ne minősített elektronikus aláírást készítsen. Bizonyos helyeken csak minősített aláírást fogadnak el, ezért nem javasoljuk, hogy valaki fokozott biztonságú elektronikus aláírás készítésére rendezkedjen be. (Lásd: 6.1. fejezet.)

Természetes személyek számára a minősített elektronikus aláírás használatát javasoljuk.

12.4.3. Vásároljak HSM-et?

A HSM (hardware security module) az intelligens kártyához hasonlóan a magánkulcsunkat védheti meg. Ha megfelelően azonosítjuk magunkat a HSM felé, a HSM aláírja, amit küldünk neki, de a magánkulcsot nem lehet – nyíltan – kinyerni belőle. Szemben az intelligens



12.7. ábra. **A HSM aláírja a bemenetet, és visszaadja a kapott aláírást. A HSM nem tudja vizsgálni, hogy milyen dokumentumot ír alá.**

kártyákkal, a HSM-ből ki lehet nyerni a magánkulcsot, de csak titkosított formában, és az így titkosított magánkulcsot – megfelelő kártyák, kulcsok stb. birtokában – be lehet tölteni egy másik HSM-be.

Akkor célszerű HSM-et használni, ha magas biztonsági szinten, nagy tömegű elektronikus aláírást szeretnénk készíteni. A helyesen üzemeltett HSM-ben védett kulcsot sem a rendszergazdánk, sem a szervert feltörő támadó nem tudja lemásolni vagy elvinni.

Ugyanakkor *a HSM aláírja a bemenetet, amit küldünk neki.* Így *akár a rendszergazda, akár a szervert feltörő támadó, bármilyen dokumentumot aláíratha a HSM-ben tárolt magánkulcsunkkal, ez ellen a HSM nem nyújt, nem nyújthat védelmet* (lásd: 12.7. ábra). Ennek következtében a HSM nem váltja ki egy biztonságos, szabályozott üzemeltetési környezet kialakítását és fenntartását.

Aki magas biztonsági szinten készít aláírást, de csak keveset, annak azt javasoljuk, inkább minősített aláírást, és intelligens kártyát használjon. Akinek elegendő alacsonyabb biztonsági szintű aláírás is, annak azt javasoljuk, szoftveres tanúsítványt használjon, és inkább e szoftveres kulcsot védje meg.

A HSM-ről gyakran mondják, hogy nagy tömegű aláírást nagyon gyorsan tud létrehozni. Tapasztalatunk azt mutatja, hogy a számítógépek valós környezetben sokszor gyorsabbak a HSM-eknél. Ezen túl egy HSM árából általában sok nagy teljesítményű szerver számítógép vásárolható, így a számítógépek átalában sebesség szempontjából időnként költséghatékonyabbnak mondhatóak.

Célszerű végiggondolni, hogy a HSM-re szánt többletköltséget nem célszerűbb-e az üzemeltetési környezet biztonságára fordítani.

12.4.4. Mikor bízhatok meg az aláíráshoz használt számítógépben?

„Sose bízz olyan számítógépben, amit nem tudsz kidobni az ablakon."

– Ismeretlen

Elektronikus aláírások készítéséhez és ellenőrzéséhez számítógép, vagy más, hasonló berendezés segítségét kell igénybe vennünk. Olyan biztonságos számítógépet és olyan szoftvereket célszerű használni, amelyek működésében megbízunk. Csak így biztosítható, hogy valóban a kívánt dokumentumot írjuk alá, és valóban pontos információhoz jutunk egy aláírás érvényességét illetően.

Egy számítógép biztonságos üzemeltetése nem könnyű feladat, de nemcsak az elektronikus aláíráshoz van szükség biztonságos számítógépre. Kizárólag biztonságos számítógépen végezhető felelősséggel bármilyen olyan munka, amelynek során érzékeny adatokat kezelünk.

Ha azt szeretnénk, hogy egy számítógépet alappal tekinthessünk megbízhatónak, célszerű követni az alábbiakban leírt útmutatást. A felsorolt, kilenc pontot a `cert.org` által kibocsátott, „Home Computer Security" c. dokumentumban megfogalmazott ajánlások alapján állítottuk össze. [24]

1. *Használjon vírusvédelmi szoftvert.* Úgy állítsa be, hogy az folyamatosan védje a számítógépet, és rendszeresen frissítse a szoftver vírus-adatbázisát.

2. *Rendszeresen frissítse a számítógépén futó operációs rendszert és egyéb szoftvereket.* A legtöbb szoftvergyártó rendszeresen bocsát ki biztonsági frissítéseket termékeihez. Minél hamarabb telepíti e frissítéseket, annál hamarabb válik gépe védetté a legújabb ismert támadásokkal szemben.

3. *Körültekintően járjon el e-mailben érkező csatolt fájlok megnyitásakor.* Gondolja végig: Ismeri a feladót? Kapott már tőle levelet? Vár tőle levelet az adott tárgyban? Van értelme a levél tárgyának, illetve a levélnek? Jelez veszélyt akár a vírusvédelmi szoftver, akár a levelezőprogram?

4. *Használjon személyi tűzfal programot.* A számítógépre telepített személyi tűzfal meggátolhatja, hogy illetéktelenek hálózaton keresztül hozzáférjenek a számítógépéhez, valamint meggátolhatja, hogy a számítógépre felkerült rosszindulatú alkalmazás adatokat küldjön el a gépéről.

5. *Rendszeresen készítsen biztonsági mentést a számítógépén lévő adatokról.* Az elektronikusan aláírt fájlok mentése különösen fontos, mert ezen hiteles dokumentumokból lehet, hogy egyáltalán nincsen papír alapú (hiteles) példány. Az elektronikusan aláírt fájlokat célszerű minősített elektronikus archiválás-szolgáltatónál elhelyezni.

6. *Erős jelszavakkal vagy intelligens kártyával férjen hozzá adataihoz.*

7. *Körültekintően járjon el új programok telepítésekor.* Győződjön meg róla, hogy megbízható forrásból származó szoftvert telepít. Gondolja végig, hogy megbízik-e az adott szoftvergyártóban, és győződjön meg róla, hogy valóban az adott gyártó termékét telepíti-e.

8. *Használjon hardveres tűzfalat.* Már egy egészen egyszerű tűzfal-berendezés is jelentős védelmet nyújthat: az Internetről érkező támadások nem közvetlenül az Ön számítógépét, hanem a tűzfal-berendezést érik.

9. *Csak megbízható személyek férhessenek hozzá a számítógépen lévő programokhoz, érzékeny adatokhoz.* Nem célszerű olyan gépen készíteni elektronikus aláírást, amelynek nem bízik meg a rendszergazdájában, vagy amelyet sok ismeretlen is használ.

Az elektronikus aláírás biztonságos, felelősségteljes használatához biztonságos, megbízható számítógépre van szükség. Ez attól függetlenül igaz, hogy Ön szoftveres aláíró kulcsot vagy intelligens kártyát (vagy például HSM-et vagy USB tokent) használ aláírások létrehozására, illetve hogy a kártya (vagy HSM, USB token stb) pontosan milyen biztonsági funkciókkal bír és milyen tanúsítással rendelkezik.

Egy jó minőségű intelligens kártya vagy egy PIN pades kártyaolvasó bizonyos támadásokat elháríthat vagy megnehezíthet, de nem változtat azon, hogy az elektronikus aláíráshoz biztonságos számítógépes környezetre van szükség.

12.4.5. Milyen adatok feltüntetését kérjem a tanúsítványomban?

„Persze nem biztos, hogy jót tesz neked,
Ha kiszolgálják az ízlésedet”

– *Bonanza Banzai, Induljon a banzáj!*

Javasoljuk, hogy tájékozódjon, hogy milyen alkalmazásokban szeretné használni a tanúsítványt, és hogy ezen alkalmazások milyen adattartalmat várnak el. Például van olyan alkalmazás, amely csak ügyvédi aláírásokat fogad el, és megköveteli, hogy a tanúsítvány `title` mezejében szerepeljen az „ügyvéd” szöveg. A befogadó alkalmazások által elvárt adattartalom feltétlenül szerepeljen a tanúsítványban.

> **Megjegyzés:** Szerencsétlen esetekben előfordulhat, hogy a különböző befogadó rendszerek által támasztott elvárások ellentmondanak egymásnak. Ekkor sajnos nem elegendő egyetlen tanúsítványt használni.

Javasoljuk, hogy *a kötelező elemeken túl a lehető legkevesebb információ feltüntetését kérje.* Az egyénileg kért információkat a befogadók általában nem tudják figyelembe venni, így ritka, hogy ezek valóban jól használhatóak. Ugyanakkor ezen információk problémát jelenthetnek, mert a hitelesítés-szolgáltatónak ellenőriznie kell ezen információkat, mielőtt feltünteti a tanúsítványban, illetve a tanúsítványt vissza kell vonni, hogyha bármilyen információ megváltozik. A visszavonás és az új tanúsítvány kibocsátása adminisztratív terhet és költséget is jelenthet.

> ***12.15. Példa:*** *Alajos ügyvéd. Tanúsítványában szerepel a neve (`CN=Alajos`), a tény, hogy ő ügyvéd (`Title=ügyvéd`), az illetékes ügyvédi kamara megnevezése és a lajstromszáma (`SN=Budapesti Ugyvedi Kamara, lajstromszam=1234`), Alajos ügyvédi irodájának neve és címe (`O=Dr. Alajos Ügyvédi Iroda, Budapest, Hosszú u. 3/b`), az e-mail címe (`EA=alajos@valahol.net`).*
>
> *E tanúsítványt vissza kell vonni, ha például megváltozik Alajos neve vagy e-mail címe, ha megváltozik az ügyvédi iroda neve (pl. mert egyesül Bendegúz irodájával), vagy ha átköltözik a Budapest, Hosszú u. 3/c alá.*

A tanúsítványban szereplő megnevezés (DN) lehetőleg a következő adatokat tartalmazza:

- az alany nevét, ahogy azt a papír alapú aláírás is tartalmazza,
- az alany e-mail címét, mert sok alkalmazás (pl. a levelezőprogramok) ezt megkövetelik, és enélkül nem működnek,
- a `Country` mezőt (ami Magyarországon `HU`), amiből meg lehet állapítani, hogy a tanúsítványt a magyar jogrend szerint kell értelmezni.

Javasoljuk, hogy a további adatokat, így például a munkahelyet, beosztást, szerepköröket, attribútumokat attribútum-tanúsítvány segítségével igazoljuk, és ne a tanúsítványban szerepeltessük (11. fejezet).

12.4.6. Célszerű hozzájárulnom a tanúsítványom nyilvánosságra hozatalához?

A nyilvános kulcsú infrastruktúra szereplőinek van magánkulcsa, amit mindenki titokban tart, és nyilvános kulcsa, amit nyilvánosságra hozhat. Műszaki szempontból az a „természetes”, ha a nyilvános kulcs nyilvános. Jogi, adatvédelmi szempontból érthető, ha valaki nem szeretné, hogy adatai között bárki böngészhessen, és még azt sem feltétlenül szeretné nyilvánosságra hozni, hogy mely hitelesítés-szolgáltatótól vagy szolgáltatóktól van tanúsítványa.

Magyarországon a hitelesítés-szolgáltató csak akkor hozhatja nyilvánosságra a tanúsítványt, ha az alany hozzájárul (ekkor viszont köteles is nyilvánosságra hozni). (Lásd: 3.3.5. fejezet.)

Van értelme a nyilvános kulcsú infrastruktúrának úgy, hogy a nyilvános kulcsokat tartalmazó tanúsítványok nem nyilvánosak?

- Az aláíró tanúsítványokat nem szükséges nyilvánosságra hozni. Ha valaki aláírást készít, a szabványos formátumú aláíráshoz úgyis csatolja a tanúsítványát, így az aláírást befogadó érintett fél várhatóan az aláírást tartalmazó aláírás-blokkból (6.4.1. fejezet), és nem a szolgáltató tanúsítványtárából szerzi be a tanúsítványt.

- Az autentikációs tanúsítványokat sem szükséges nyilvánosságra hozni. Az autentikációs tanúsítványt általában olyan protokollban használjuk, ahol a felek a kapcsolatfelvétel során először elküldik egymásnak tanúsítványaikat, és utána használják fel őket.

- Egyedül titkosító tanúsítványok esetén célszerű nyilvánosságra hozni a tanúsítványt. Titkosító tanúsítványok esetén a titkosított üzenetet küldő kezdeményező félnek hozzá kell jutnia a címzett tanúsítványához, csak így tud titkosított üzenetet küldeni. Ekkor is megoldható, hogy a felek elküldik egymásnak tanúsítványaikat, de ez esetleg túl körülményes. Titkosító tanúsítvány esetén elképzelhető, hogy a kezdeményező fél a hitelesítés-szolgáltató tanúsítványtárában keresi a címzett tanúsítványát.

A mai gyakorlat szerint a szolgáltatók általában nyilvánosságra hozzák a tanúsítványokat, de ennek elsősorban titkosító tanúsítványok esetén van értelme. Aláíró és autentikációs tanúsítványok esetén leginkább kíváncsiságból keres valaki a tanúsítványtárban.

12.4.7. Mit kezdjek a bejövő tanúsítványban szereplő DN-nel?

Ha egyazon szolgáltató egyazon hitelesítő egysége kibocsát két tanúsítványt, amelynek azonos a DN-je, akkor biztosak lehetünk benne, hogy a két tanúsítvány ugyanahhoz az alanyhoz tartozik. Ha a két DN egy bitben is eltér, akkor már nem.

Ezen túl ha a tanúsítvány nem álneves, akkor célszerű csak szövegesen értelmezni a DN tartalmát, úgy, mintha egy papír alapú dokumentumon ez a szöveg szerepelne az aláírás mögött. Ha papír alapon azt látom, hogy „Kovács János, Kókler Bt.”, akkor sem tudom, hogy az illető személy milyen viszonyban áll a Kókler Bt-vel, egyedül annyit tudok, hogy a Kókler Bt. neve szerepel a személy neve mögött. Ne vonjunk le ennél több következtetést egy DN alapján sem.

Ha a tanúsítvány álneves, akkor még az előző elv sem alkalmazható, semmiben nem lehetek biztos a tanúsítvány alanyának adataival kapcsolatban. Egyedül annyit tudok, hogy jogvita esetén megtudhatom, hogy ki volt az illető.

Ha a pontos értelmezésre van szükségünk, el kell olvasni a tanúsítványra vonatkozó hitelesítési rendet.

12.4.8. Elfogadhatok álneves tanúsítványt?

Álneves tanúsítvány esetén a tanúsítványban nem az aláíró valódi neve, hanem álnév szerepel, így magából az aláírásból nem (vagy nem feltétlenül) deríthető ki az aláíró kiléte. (Lásd: 3.2.4. fejezet.)

Nem befolyásolja az aláíráshoz kapcsolódó bizonyító erőt, ha az álneves tanúsítványra épül. Akár minősített tanúsítvány is kibocsátható álnévre, és e minősített tanúsítvány alapján – biztonságos aláírás-létrehozó eszköz segítségével – minősített elektronikus aláírás hozható létre.

Az álneves tanúsítvány nyugodtan elfogadható, ha például:

1. Aláírás ellenőrzésekor egyáltalán nem érdekel bennünket az aláíró kiléte. Előfordulhat, hogy elég annyit tudnunk, hogy a dokumentumot „valaki” – egy természetes személy – aláírta, és bár nem tudjuk az illető kilétét, jogvita esetén a hitelesítés-szolgáltató bevonásával ki tudjuk deríteni, és rá tudjuk bizonyítani az illetőre az aláírást.

2. Már tudjuk, hogy kihez tartozik a tanúsítvány, így nem a tanúsítványból, nem az aláírásból szeretnénk megtudni az aláíró kilétét.

3. Aláírás ellenőrzésekor nem az érdekel bennünket, hogy ki az aláíró, hanem az, hogy jogosan írta-e alá az adott dokumentumot. Például ha az aláírásból annyit tudunk meg, hogy az aláíró az XYZ Kft. ügyvezetője, aki a Kft. nevében önállóan cégjegyzésre jogosult, nem feltétlenül kell tudnunk az ő nevét vagy más személyes adatait. Hasonló helyzet, ha egy technikai kérdést tartalmazó levelünkre a megkeresett cég „support munkatársa” válaszol. Nem az ő személyes adataira vagyunk kíváncsiak, hanem arra, hogy a cég nevében egy kompetens személy írt nekünk választ, és szükség esetén igazolni tudjuk, hogy mit küldött nekünk az illető. Az aláíró szerepe vagy jogosultsága ekkor kiderülhet magából a tanúsítványból vagy egy hozzá kapcsolódó attribútum-tanúsítványból (11. fejezet) is.

Nagyon kevés szabály vonatkozik az álneves tanúsítványokra, így nem lehet tudni, hogy az alany tanúsítványban szereplő megnevezésének (DN) mekkora része az álnév, és mekkora része tartalmaz valós információkat, illetve az álnevet ki és milyen módon határozta meg.

Például ha egy álneves tanúsítványban annyi szerepel, hogy „XYZ Kft.” és „support munkatárs”, akkor lehet, hogy:

- Az XYZ Kft. valós információ, a hitelesítés-szolgáltató meggyőződött róla, hogy a tanúsítványt az adott cég munkatársa igényelte. A hitelesítés-szolgáltató a cég állítása alapján írta a tanúsítványba a „support munkatárs” álnevet, így számíthatunk rá, hogy az illető valóban ilyen szerepkörrel rendelkezik.

- Az XYZ Kft. valós információ, a hitelesítés-szolgáltató meggyőződött róla, hogy a tanúsítványt az adott cég munkatársa igényelte. A többi információt a hitelesítés-szolgáltató az alany választása alapján töltötte ki, a cégnél volt, aki a „Micimackó", volt, aki a „Malacka", és volt, aki a „support munkatárs" álnevet választotta magának. Lehet, hogy az illető takarítóként dolgozik a cégnél, és nem kompetens az adott műszaki kérdésben.

- A teljes név álnév, az illetőnek semmi köze az XYZ Kft-hez, és nem tölt be „support munkatárs" szerepkört.

Az sem egyértelmű, hogy mit nevezünk álneves tanúsítványnak. Egyes álláspontok szerint, ha nem az alany ékezethelyes neve szerepel a tanúsítványban, akkor a tanúsítvány álneves. Ha a Kovács János Béla nevű felhasználó tanúsítványában a Kovacs Janos szöveg szerepel, az lehet, hogy álnév, de mégis egészen más, mintha a „Micimackó" szöveg szerepelne ott.

Előfordulhat, hogy az álnév félrevezető. Például ha valaki az Alajos nevű felhasználótól vár levelet, és olyan levelet kap, amelyben az aláíró (aki valójában Manfréd, a támadó) az Alajos álnevet használta, lehet, hogy azt hiszi, hogy a levél valóban Alajostól jött. Igaz, az Eat. értelmében a hitelesítés-szolgáltató köteles egyértelműen feltüntetni, ha egy tanúsítványban álnév szerepel, de nincs előírva, hogy hogyan. A hazai szolgáltatók gyakorlata rendkívül szerteágazó, az egyes szolgáltatók szabályzatainak részletes tanulmányozása alapján lehet kideríteni, hogy mely szolgáltató hol és milyen módon tüntetni fel, hogy a tanúsítvány álneves.

Fent felsoroltuk, hogy mikor fogadhatunk el nyugodtan álneves tanúsítványt. Az 1. eset nagyon ritka, a 3. eset teljesen bizonytalan, és a 2. esetben is általában sokkal nyugodtabb az az aláírást elfogadó fél, aki az aláíró valódi nevét is látja. A tapasztalat azt mutatja, a felhasználók félnek az álnevektől, illetve a szabályozatlanság olyan bizonytalanságot jelent, hogy az álneves tanúsítványok nagyon nehezen használhatóak.

12.4.9. Használjak álneves tanúsítványt?

Az előző fejezetben leírtuk, hogy bár az álneves tanúsítványok elvileg egyenértékűek a nem álnevesekkel, a befogadók általában bizalmatlanok az álneves tanúsítványokkal szemben, és nem fogadják el őket. Például a közigazgatás sem fogad el álnévre kibocsátott tanúsítványokat. [181], [97]

Mivel általában azt szeretnénk, hogy egy tanúsítványt minél több helyen használhassunk, és sok helyen nem fogadják el az álneves tanúsítványokat, jelenleg *nem tudjuk javasolni az álneves tanúsítványok használatát.*

12.4.10. Milyen biztonsági szintű tanúsítványokat, aláírásokat célszerű elfogadni?

Aki úgy dönt, elektronikus aláírásokat szeretne befogadni, annak célszerű – pl. aláírási szabályzat keretében – meghatároznia, hogy milyen aláírásokat fogad el. Ennek keretében célszerű meghatároznia, hogy milyen aláírói tanúsítványokat fogad el.

Felhívjuk a figyelmet, hogy a fokozott biztonságú elektronikus aláírás önmagában nagyon keveset jelent, például nem jelenti azt, hogy a hitelesítés-szolgáltató találkozott az aláíróval. Fokozott biztonságú elektronikus aláírás létrehozására alkalmas tanúsítvány akár postán is igényelhető. Általában célszerű ennél részletesebben meghatározni, hogy milyen tanúsítványokat fogadunk el. Erre adunk most néhány példát.

- Akkor célszerű minden, nyilvános körben használható, legalább fokozott biztonságú elektronikus aláírás létrehozására alkalmas tanúsítványt elfogadunk, ha tisztában vagyunk vele, hogy ez nagyon-nagyon keveset jelent.

- Gyakori megoldás a személyes találkozás során kibocsátott tanúsítványok elfogadása, ide tartoznak:

  - a minősített tanúsítványok,
  - a „ketes”, azaz a közigazgatásban használható tanúsítványok,
  - az ún. III. hitelesítési osztályba[8] tartozó tanúsítványok, illetve minden olyan tanúsítvány, ahol a hitelesítés-szolgáltató vállalja (pl. a hitelesítési rendben), hogy személyes találkozás során bocsátja ki a tanúsítványt.

- Megkövetelhetjük, hogy a személyes találkozás során kibocsátott tanúsítványok magánkulcsát kriptográfiai hardver eszköz, pl. intelligens kártya védje. E követelménynek megfelelő tanúsítványok:

  - a minősített tanúsítványok,
  - a „ketes” nem minősített tanúsítványok közül az EHR+ hitelesítési rendnek megfelelő tanúsítványok, [83]
  - minden olyan tanúsítvány, amelyre a hitelesítés-szolgáltató vállalja, hogy a tanúsítványt kriptográfiai hardver eszközön bocsátja ki.

- Megkövetelhetjük, hogy csak minősített aláírást (azaz minősített tanúsítványt) fogadunk be, mert ekkor tartozik a legmagasabb bizonyító erő az aláíráshoz. Vigyázat, ekkor kizártuk az automatizmussal történő aláírásokat, és ekkor a felhasználóink minden

[8] E terminológia a Verisigntól származik, de egyes magyar szolgáltatók is követik.

egyes aláíráskor PIN kódot kell, hogy gépeljenek. Előfordulhat, hogy egyes aláírás-létrehozó alkalmazások nem működnek együtt minősített tanúsítványokkal, így ekkor ezen alkalmazásokat is eleve kizárjuk.

Megkövetelhetjük, hogy csak olyan minősített tanúsítványokat fogadunk be, amelyekben meghatározott tranzakciós limit szerepel. Ekkor ha a hitelesítés-szolgáltató hibát vét a tanúsítvánnyal kapcsolatban, e tranzakciós limitig meg kell térítenie a kárunkat.

Az aláírások minősítettségét könnyű szabványosan, szolgáltató-függetlenül ellenőrizni, és a tranzakciós limit mértéke is könnyen, szabványos, szolgáltató-független módon kinyerhető a minősített tanúsítványból. Mindezekre a minősített tanúsítványok ETSI TS 101 862 által meghatározott QCStatement kiterjesztése ad lehetőséget. [50]

12.4.11. Milyen aláírásformátumot használjak (PDF, XAdES, S/MIME stb)?

E téren a következő tanácsokat fogalmazhatjuk meg:

- Nem javasoljuk az egyes célalkalmazások (pl. Word, Excel stb.) által használt aláírás-formátumokat. Az ilyen aláírások általában csak egyetlen fájltípust fednek le, csak egyetlen problémát oldanak meg. Ha egy rendszert ilyen aláírásokra építünk, nem biztos, hogy más fájlokat is kezelni tudunk majd.

  A célalkalmazások fájlformátuma sokszor nem nyilvános, és ekkor az sem nyilvános, hogy milyen fajta aláírást is használunk. Ha megszűnik az alkalmazás támogatása, esetleg egyáltalán nem tudjuk majd ellenőrizni az aláírásokat sem, illetve ha változik az alkalmazás fájlformátuma, az mélyen hat az aláírás-kezelő rendszerre is. Végül, ha a célalkalmazás formátumát használjuk, ki vagyunk szolgáltatva a célalkalmazás aláírási szabályzatának, azaz csak úgy és olyan módon tudunk aláírást készíteni vagy ellenőrizni, ahogy a célalkalmazás azt lehetővé teszi.

- Elektronikus levelezés aláírása esetén az S/MIME az elterjedt megoldás, ha csak a levelet írjuk alá, ezt célszerű használni. E megoldást általában nem jogilag is kötelező aláírásokra és nem minősített aláírásokra szokás használni, hanem pusztán a levelek integritásának védelmére. A levelezőprogramok jellemzően nem támogatják az időbélyegzést és a visszavonási információk csatolását, így a levelezés aláírása az EU-s, minősített elektronikus aláírás koncepció szerint nemigen használható.

- A XAdES (és CAdES, illetve a PKCS#7 és XMLDSIG) aláírások önmagukban, konténer nélkül nehezen használhatóak. Például egy XAdES aláírás konténer nélkül egy XML fájl. A legtöbb operációs rendszer a kiterjesztése alapján kapcsolja a fájlokat alkalmazásokhoz, és nem feltétlenül szerecsés a XAdES aláírást egy általános XML

megjelenítővel jeleníteni meg, és szintén nem szerencsés, ha minden XML fájlt XAdES aláírásként nyitunk meg. Nehéz meghatározni, hogy az aláírás-blokkban (6.4.1. fejezet) szereplő aláírás pontosan mire vonatkozik, azaz hol kezdődik a dokumentum, és hol végződnek a rá vonatkozó metaadatok. Még azt sem könnyű eldönteni, hogy egy beérkezett XML fájlban minden alá van-e írva. (Minden elem nem lehet aláírva, mert az aláírás önmagára nem vonatkozhat, így minden „lényeges" elemről meg kell vizsgálni, hogy az az elem is alá van-e írva.) A gyakorlati folyamatokban nem csak egy aláírás szerepel, hanem aláírások egymáshoz viszonyulnak, egymásra (és különféle dokumentumokra) hivatkoznak. Ezek bonyolult hálót képezhetnek, amit nem könnyű kibogozni.

A XAdES aláírások (és más aláírás-blokkok) mellett a valós felhasználáshoz mindenképpen szükség van egy konténerre is, amely a fenti problémákat kezeli, és a szerteágazó lehetőségeket valamilyen mederbe tereli.

- Magyarországon a legelterjedtebb konténerek az e-akta (6.4.2.1. fejezet) és a PDF (6.4.2.2. fejezet). E kettő közül a következő szempontok alapján választanunk:

  – Ha rendszerünkben minden esetben igaz, hogy egy fájlban egy aláíró egy dokumentumot ír alá, és azt nem kell hosszú távon archiválni (ilyen pl. az elektronikus számlázás esete), akkor a PDF talán azért lehet jobb választás, mert a megnyitásához szükséges Acrobat Reader nagyon elterjedt, nem ró terhet az érintett félre. Igaz, e probléma e-aktával is megoldható.

  – Ha az előző pontban felsoroltak valamelyike nem teljesül, például több, egymáshoz kapcsolódó dokumentumot több felhasználó ír alá, ezeket később időbélyegezni kell, illetve archív aláírássá kell kiterjeszteni, vagy hosszú távon archiválni kell, illetve összetett aláírási szabályzatot szeretnénk érvényesíteni, akkor egyértelműen az e-aktát javasoljuk.

    A klasszikus „PDF aláírás" (amelyet az ISO 32000 határoz meg) esetén a kiterjeszthetőség problémás, mert az aláírások mérete nem növelhető utólag. Ezt a PAdES (PDF Advanced Electronic Signatures) specifikációja megoldotta, de még kevés implementáció támogatja e megoldást. [87], [60] Könyvünk készítésekor PDF vagy PAdES konténerben lévő aláírást még nem lehet magyar archiválás-szolgáltatóhoz beküldeni.

12.4.12. Mikor célszerű különálló aláírást használni?

Akkor beszélünk különálló aláírásról, ha az aláírás és az aláírt dokumentum(ok) külön fájlban helyezkednek el.

Különálló aláírás esetén az aláírt dokumentumot könnyebb megnyitni, e-akta esetén a

megnyitáshoz először ki kell csomagolni őket. Ugyanakkor ha különálló aláírás esetén valami miatt elválik egymástól a dokumentum és az aláírás, vagy megsérül a dokumentum, az aláírás végérvényesen elromolhat. Ha több dokumentum van aláírva, e problémák fokozottan jelentkezhetnek Ha több aláírás van, e problémák még fokozottabban jelentkezhetnek

A dokumentumokat és aláírásokat egybe foglaló e-akta sokszor könnyebben kezelhető, és a fenti problémák elkerülhetőek.

Ha az aláírással az a célunk, hogy a címzett felhasználó könnyen meg tudja nyitni a dokumentumot, de azt is állíthassuk, hogy aláírt dokumentumot küldtünk, akkor a különálló aláírás is jó megoldás lehet. Nagyon nagy fájlok esetén is célszerű lehet a különálló aláírást választani, mert a nagy e-akták sokszor nehezen kezelhetőek. Egyébként célszerű egységbe (e-aktába) foglalni a dokumentumot és az aláírást, így sokkal könnyebb később feldolgozni. Akkor ajánlott különálló aláírásokra támaszkodni, ha van olyan célrendszerünk, amely gondoskodik az aláírások és az aláírt dokumentumok megfelelő összepárosításáról.

12.4.13. Milyen XAdES aláírástípust használjak?

Azt javasoljuk, a következő elvek alapján válasszuk ki a nekünk megfelelő XAdES-típust:

- Mindig legalább időbélyeggel ellátott (XAdES-T) aláírást hozzunk létre, mert csak így biztosítható, hogy az aláírás érvényessége később is igazolható marad. Ha nem mi hozzuk létre az aláírást, hanem csak befogadjuk, akkor ellenőrizzük, hogy van-e rajta időbélyeg, és tegyünk rá, ha még nincs.
- Ha 11 évnél hosszabb időre kell archiválnunk, vagy különösen fontosnak ítéljük, hogy az aláírás érvényessége mindenképpen igazolható maradjon, akkor használjunk archív (XAdES-A) aláírást, vagy forduljunk archiválás-szolgáltatóhoz.
- Kizárólag akkor hozzunk létre „alap” (XAdES-BES vagy -EPES) aláírást, ha nincs más lehetőségünk (pl. nincs online kapcsolat az időbélyegzés-szolgáltatóval). Ez is csak ideiglenesen maradjon így, minél hamarabb gondoskodjunk az aláírás időbélyegzéséről – XAdES-T aláírássá való kiterjesztéséről. Vigyázat, az időbélyeg nélküli „alap” aláírás bármikor ellenőrizhetetlenné válhat!

A következőkben a fent leírt elveket indokoljuk meg.

12.4.13.1. Aláírás-típusok

Az aláírás ellenőrizhetőségének szempontjából a következő aláírás-típusokat tartjuk különösen fontosnak:

- „Alap” aláírás: a dokumentumot aláírással látjuk el, és nem helyezünk el rajta időbélyeget. Ilyen például a XAdES-BES vagy a XAdES-EPES.

- Időbélyeggel ellátott aláírás: olyan egyszerű aláírás, amelyen időbélyeget helyeztünk el. Az időbélyeg igazolja, hogy amikor az aláírás készült, akkor az aláíró tanúsítványa még érvényes volt. Ilyen típus a XAdES-T.

- Az időbélyeggel ellátott aláírásokhoz különféle információkat csatolhatunk, és ezeket időbélyeggel láthatjuk el. Ilyen típusok például a XAdES-C, XAdES-X és XAdES-X-L.

- Archív aláírás: olyan egyszerű aláírás, amelyen időbélyeget helyeztünk el, majd csatoltunk hozzá minden szükséges visszavonási információt – beleértve a szolgáltatói tanúsítványokra és az időbélyegekre vonatkozó visszavonási információkat is – majd *rendszeresen* további archív időbélyegeket helyezünk el rajtuk. Az egyes archív időbélyegek igazolják, hogy a korábbi archív időbélyegek (illetve a dokumentum és az aláírás) mikor készültek, így az archív aláírás érvényessége akkor is igazolható, ha az ezek készítéséhez használt technológia már elavult, vagy ha az ezek készítéséhez használt kulcs időközben kompromittálódott.

12.4.13.2. Meddig igazolható ezen aláírás-típusok érvényessége?

A fenti aláírás-típusoknak megfeleltethető ugyan egy-egy XAdES-típus, de az aláírások elvi ellenőrizhetősége nem a konkrét formátumtól (hanem az aláíráshoz tartozó időbélyegek és egyéb információk logikai kapcsolatától) függ, így az alábbiakban leírt megállapítások nem XAdES-specifikusak, más aláírás-formátumok esetén is teljesülnek.

- *Az „alap" elektronikus aláírások szinte bármikor letagadhatóak.* Ha az aláíráshoz használt tanúsítvány érvénytelenné válik, később – önmagában, az aláírás alapján – nem lehet bizonyítani, hogy a tanúsítvány még érvényes volt, amikor az aláírás készült. Így nehéz lehet kivédeni egy olyan állítást, amely szerint az aláírás a már érvénytelen tanúsítvánnyal, kompromittálódott kulccsal készült. A tanúsítvány mindenképpen érvénytelenné válik, ha lejár (általában legfeljebb 2 évig lehet érvényes), de bármikor visszavonhatják, például ha az aláíró arról értesíti a tanúsítványt kibocsátó hitelesítés-szolgáltatót, hogy a kulcsa kompromittálódott.

  A digitális archiválásról szóló 114/2007. GKM rendelet nem engedi meg a csak „alap" aláírással történő archiválást. [69]

- Az időbélyeggel ellátott aláírás érvényessége addig igazolható, amíg az időbélyegzés-szolgáltató tanúsítványa érvényes. Jogi szempontból a digitális archiválásról szóló rendelet szerint *az időbélyeggel ellátott aláírás a legfeljebb 11 évre szóló archiválásra megfelelő.* Az időbélyegzés-szolgáltatói tanúsítványok általában sokáig érvényesek, és nagyon ritkán vonják vissza őket, az a célszerű, ha az időbélyegzők tanúsítványa mindig legalább 11 évig érvényes.

Ugyanakkor a XAdES-T csak azt igazolja, hogy az aláírás mikor készült, önmagában nem tartalmazza az aláírásra vonatkozó visszavonási információkat. A XAdES-T aláírásból nem állapítható meg, hogy az időbélyegzés pillanatában valóban érvényes volt-e a tanúsítvány, ezen információkat a tanúsítványt kibocsátó hitelesítés-szolgáltatótól kell beszerezni. Amíg a tanúsítvány érvényes, addig ezen információk könnyen, szabványos módon (pl. CRL vagy OCSP segítségével) beszerezhetők, de a tanúsítvány lejártát követően igen körülményessé válhat ezen információk összegyűjtése. A hitelesítés-szolgáltató a tanúsítvány lejártát követően 10 évig (Eat 9. § (7)) köteles megőrizni ezen információkat, utána akár meg is semmisítheti[9] őket.

- Az archív aláírást is időbélyegek védik, így itt is az mondható, hogy az archív aláírás érvényessége addig igazolható, amíg a külső időbélyeghez tartozó időbélyegzés-szolgáltató tanúsítványa érvényes. A XAdES-A annyiban több, mint a XAdES-T, hogy:

  – A XAdES-A archív aláírás tartalmazza az aláíráshoz és az aláíráson lévő időbélyegekhez kapcsolódó minden (végfelhasználói és szolgáltatói) tanúsítvány visszavonási információit, így ezek beszerzéséhez nem szükséges később a hitelesítés-szolgáltatóhoz fordulni. (A külső időbélyeg visszavonási állapotáért továbbra is valamely hitelesítés-szolgáltatóhoz kell fordulnunk, de már nem függünk az eredeti hitelesítés-szolgáltatótól.)

  – Ha *rendszeresen* (amíg a korábbi időbélyeg érvényessége még igazolható) csatoljuk a XAdES-A aláíráshoz a külső időbélyeg aktuális visszavonási információit, majd újabb – jellemzően más kulccsal vagy más technológiával készült – külső időbélyeget helyezünk el az aláíráson, akkor a XAdES-A aláírás érvényessége akkor is bizonyítható, ha a belső aláírások, tanúsítványok, időbélyegek érvényessége már nem igazolható.

  Ezért mondható el, hogy *ha az archív (XAdES-A) aláírást rendszeresen gondozzuk, felül-időbélyegezzük, akkor érvényessége bármeddig igazolható marad.* A digitális archiválásról szóló rendelet 11 éves megőrzési kötelezettség felett előírja az archív aláírás (pl. XAdES-A) használatát.

  Ugyanakkor ha egy XAdES-A aláírás archív időbélyegeit ugyanazon technológiával készítjük, mint a XAdES-T időbélyegét, és később nem látjuk el további időbélyegekkel a XAdES-A aláírást, akkor a XAdES-A aláírás érvényessége pontosan addig igazolható, mint a XAdES-T-é.

  Az archív aláírások megfelelő gondozása rendszeres munkát, és jelentős szakértelmet igényel, ezért ehelyett célszerű minősített archiválás-szolgáltatót igénybe venni e célra.

[9] A hitelesítés-szolgáltató 10 évet követően *köteles* megsemmisíteni ezen adatokat, kivéve, ha az aláíró hozzájul, hogy az adatkezelés a kötelező 10 évnél is tovább tartson.

12.4.14. Biztonságos csatornával ki tudom váltani az aláírást?

Ha két fél biztonságos csatornát (pl. SSL) épít ki egymással, mindkét fél biztos lehet a másik kilétében, és mindkét fél meg tud győződni róla, hogy a csatornán kapott üzeneteket valóban a másik fél küldte. *A csatorna hitelessége nem váltja ki a dokumentumok hitelességét.* Abban a pillanatban, hogy egy dokumentum elhagyja a csatornát, már nem igazolható, hogy az honnan származik. Más szóval, ha biztonságos csatornán kapunk egy dokumentumot, hiába tudjuk, hogy ki küldte, hiába tudjuk, hogy az illető pontosan ezt a dokumentumot küldte, mégsem tudjuk ezt bizonyítani harmadik félnek.

A biztonságos csatorna nem váltja ki az aláírást. (Lásd: 12.1. fejezet.)

12.4.15. Időbélyeggel ki tudom váltani az aláírást?

Az időbélyeg azt bizonyítja, hogy egy adott lenyomatú dokumentum adott időpillanatban már létezett (7. fejezet). *Az időbélyeg nem igazolja, hogy a dokumentum hol, és kinél létezett.*

Egy időbélyegzés-szolgáltató több ügyfél számára is szolgáltat, szolgáltathat időbélyeget, az egyes ügyfelek ugyanolyan értékű időbélyegeket hozhatnak létre.

> ***12.16. Példa:*** *Alajos is és Bendegúz is ügyfele az X időbélyegzés-szolgáltatónak, és adott egy d dokumentum, amelyen az X időbélyegzés-szolgáltató a t időpontban elhelyezett egy időbélyeget. Az időbélyegből nem lehet megmondani, hogy Alajos vagy Bendegúz rendelkezett-e az adott dokumentummal a t időpontban. Az időbélyeg egyedül annyit igazol, hogy a d dokumentum a t időpontban már létezett. Az időbélyeget lekérhette Alajos is, Bendegúz is, a szolgáltató bármely másik ügyfele is, de akár maga a szolgáltató is tesztelési célból.*

Az időbélyegzés-szolgáltató nem feltétlenül vezet nyilvántartást arról, hogy mely időbélyeget mely ügyfélnek adta ki, mert az időbélyegzés-szolgáltatás nem erre való. Az időbélyegzés-szolgáltató lehet, hogy semmilyen nyilvántartást sem vezet az időbélyegekről, és egyedül csak annyit tud megmondani, hogy a t időpontban valóban kiadott egy időbélyeget. Az időbélyegzés-szolgáltató nem feltétlenül köti autentikációhoz az időbélyegek kibocsátását, elvileg ingyenesen, anonim módon is adhatja az időbélyegeket.

Elképzelhető olyan nyakatekert megoldás, amely szerint egy időbélyegzés-szolgáltató csakis és kizárólag egyetlen ügyfél számára nyújt időbélyegzés-szolgáltatást, ekkor az időbélyegből következik, hogy a dokumentum (illetve annak a lenyomata) valóban rendelkezésre állt az ügyfélnél az időbélyeg elhelyezésének időpontjában. (Ekkor igazolni kell tudni, hogy a szolgáltató valóban csak annak az egy ügyfélnek nyújt időbélyegzés-szolgáltatást, és ekkor is lehet, hogy az illető csak a lenyomattal rendelkezett, magával a dokumentummal.) Ebben a nagyon speciális esetben indirekt módon következik az időbélyegből, hogy a dokumentum lenyomata ekkor és ekkor rendelkezésre állt a szolgáltató ügyfelénél.

Ha azt szeretnénk igazolni, hogy adott dokumentum adott szereplőnél rendelkezésre állt, *elektronikus aláírás*t célszerű használni. Ekkor nem indirekt következtetésekre, hanem az Eat. által meghatározott jogkövetkezményekre támaszkodhatunk, és az illető sokkal nehezebben tagadhatja le ezt a tényt. Jogi értelemben nem jelent semmilyen kötelezettséget, ha valaki elhelyez egy időbélyeget egy dokumentumon, az aláírásnak világos következményei vannak.

Az időbélyeggel időpontot lehet igazolni, e technológia erre alkalmas. Extrém esetekben más is következhet időbélyegekből, de nem helyes, ha erre építünk. Az időbélyegzés technológiáját, illetve az időbélyegzés nevű jogi fogalmat arra kell használni, amire való.

12.4.16. Hogyan határozzam meg, hogy milyen aláírásokat fogadjak be?

Célszerű meghatározni a következőket:

- Az aláírás biztonsági szintjét (minősített/fokozott), minősített esetben a minimális tranzakciós limitet is.
- Az elfogadott gyökereket. (Esetleg meghatározhatóak az elfogadott tanúsítványláncok is.)
- Az aláírás típusát („alap” aláírás, időbélyeggel ellátott aláírás stb).
- A visszavonási állapot ellenőrzésének módját.
- A kivárási idő mértékét. Külön kivárási idő érvényesíthető az aláíró tanúsítványokra és az egyes szolgáltatói tanúsítványokra.
- Az aláírás formátumát. Itt konténer formátumot célszerű meghatározni, az aláírás-blokk formátumának rögzítése általában kevés, önmagából egy XAdES aláírásból sokszor nagyon nehéz kihámozni, hogy mi is van aláírva.
- Az aláírási jogosultság igazolásának módját. (Pl. Adott `title` értéknek kell szerepelnie a tanúsítványban, attribútum-tanúsítványt kell csatolni, a befogadó által regisztrált (lenyomatú) tanúsítvány szerint kell aláírni stb.)

12.4.17. Milyen kivárási időt használjak aláírás ellenőrzéskor?

Célszerű mérlegelni, hogy mennyire gyorsan kell dönteni egy aláírás elfogadásáról, és mekkora kockázatot jelent, ha egy érvénytelen aláírást fogadok el (pl. vissza lehet-e vonni a döntés következményeit, ha kiderül, hogy tévesen fogadtam ez az aláírást). Sajnos a gyakorlatban nagyon nehéz e kockázatokat elemezni, mérlegelni. Gyakran nem tudjuk megmondani, mekkora kár keletkezhet egy aláírás téves elfogadásából. A legrosszabb esetben általában a kár óriási, de többnyire jelentősen kisebb. Nehéz megmondani a legrosszabb eset bekövetkezési

valószínűségét. A szolgáltatói kulcsok kompromittálódása pedig ritka esemény, a bekövetkezési valószínűségét nagyon nehéz megbecsülni. Ugyanakkor ha későn értesülünk egy szolgáltatói kulcs kompromittálódásáról, az érvénytelen aláírások tömeges elfogadását okozhatja, amihez különösen nehéz kárértéket rendelni.

Ennek függvényében jó döntést jelenthetnek például a következők:

- Nem használok kivárási időt, hanem mindig a legfrissebb visszavonási információt fogadom el. Ez akkor alkalmazható, ha gyorsan kell döntenem, és kis kockázatot jelent egy érvénytelen aláírás téve elfogadása. Ez minden szolgáltatóval megvalósítható.
- A végfelhasználói tanúsítványra érvényesítek kivárási időt, minden más esetben (szolgáltatói tanúsítványokra, visszavonási információkra, időbélyegekre és ezek tanúsítványláncaira) a legfrissebb visszavonási információt fogadom el. Ez a legtöbb szolgáltató esetén megvalósítható, és kb. 4-24 óra várakozást jelent. E megoldás sok esetben jó kompromisszumot jelent a biztonság és a várakozás között, de van, amikor nem. Van, amikor gyorsabban kell döntenem, illetve nagyon nehéz megítélni annak a kockázatát, ha későn értesülök egy szolgáltatói kulcs kompromittálódásáról.
- Kivárási időt alkalmazok, ahol csak lehet: végfelhasználói tanúsítványra, az aláíráson lévő időbélyegre, a tanúsítványláncaikra, illetve a rájuk vonatkozó visszavonási információkra. Egyedül a külső (archív) időbélyegre és az ő tanúsítványláncára (és a rá vonatkozó visszavonási információkra) nem lehet kivárási időt érvényesíteni. Ezzel a lehető legnagyobb mértékben eliminálták a kivárási időből származó PKI kockázatot. Ez azért előnyös, mert így a felhasználónak nem kell számára ismeretlen PKI kockázatokkal foglalkoznia. Hátránya e megoldásnak, hogy nem minden szolgáltató esetén valósítható meg. Ehhez lényegében arra van szükség, hogy a tanúsítvány hierarchia minden szintjén „mindig friss OCSP” legyen elérhető, amihez a szolgáltatónak gyors visszavonás-kezelés szolgáltatást kell nyújtania.

12.4.18. Mire kell figyelni titkosító tanúsítványokkal kapcsolatban?

1. Aki üzenetet akar küldeni nekünk, az meg kell, hogy kapja a tanúsítványunkat. Ennek egyik módja, ha elküldjük neki, de sok szereplő esetén az nem megoldás, hogy mindenki mindenkinek küldje el a titkosító tanúsítványát. Ha a feladó a hitelesítés-szolgáltató tanúsítványtárában keresi a tanúsítványt, lehet, hogy több érvényes titkosító tanúsítvány is talál. Mi alapján választ közülük? Ha bármelyikkel titkosíthat, kérdés, hogy számunka elérhető-e éppen annak a magánkulcsa. (Lehet, hogy éppen az a kártya nincsen nálunk.)
2. Gondolni kell rá, mi történik, ha a titkosító tanúsítványunk megváltozik. Értesíteni kell erről az összes érintett felet. Ez megtörténik, ha a hitelesítés-szolgáltató visszavonja a

korábbi titkosító tanúsítványunkat. Hogyan jut el ekkor az új tanúsítvány ahhoz, aki üzenetet akar küldeni nekünk?

3. Ha megváltozik a titkosító tanúsítványunk, kérdés, hogy hogyan érjük el az adott tanúsítvánnyal titkosított korábbi üzeneteket. Ha az új tanúsítvány is a régi kulcspárhoz tartozik, elvileg birtokoljuk a szükséges magánkulcsot. Kérdés, hogy az alkalmazásunk észreveszi-e, hogy bár nem az a tanúsítványunk, amely szerint az üzenetet titkosították, de mégis birtokoljuk a hozzá tartozó magánkulcsot. Nem minden alkalmazás készült fel erre.

4. Célszerű valamilyen módon biztonsági másolatot készíteni vagy készíttetni magánkulcsunkról, hogy titkosított üzeneteink ne váljanak olvashatatlanná, ha az megsemmisül. Célszerű például kulcsletét szolgáltatást igénybe venni.

5. Ha az új tanúsítvány új kulcspárhoz tartozik, gondoskodnunk kell a régi kulcspár szerint titkosított üzenetek olvashatóságáról. Ekkor a következőket tehetjük:

   - Megőrizzük a régi kulcspárt is. (Több régi kulcs esetén ez oda is fajulhat, hogy sok kártyát hordozunk magunkkal, hogy korábbi üzeneteinket is olvashassuk.)
   - A régi kulcspárral titkosított üzenetet áttitkosítjuk az új kulcspár szerint. Ez körülményes művelet lehet, különösen sok üzenet esetén.

   Itt megoldást jelenthet, ha az üzenet csak addig van titkosított formában, amíg el nem jut hozzánk, és utána az üzenetet kicsomagoljuk, és a nyílt üzenetet tároljuk.

A PKI alapú titkosítás sok problémát vet fel, és ezek közül sokra nincsenek egyszerű és gyakorlatban jól használható megoldások. Leginkább olyan rendszerben használható, ahol automaták vesznek részt, akiknek a tanúsítványa ritkán változik, és a magánkulcsa ritkán kompromittálódik.

12.4.19. Hozzáférésmenedzsment titkosító tanúsítványok alapján?

Megtehetjük, hogy dokumentumainkat PKI alapon titkosítva tároljuk, minden dokumentumot annak a nyilvános kulcsával titkosítunk, aki a dokumentum olvasására jogosult. E megoldás működőképes, de legyünk tekintettel a következőkre:

- A hozzáférési jogosultságok változása esetén újra kell titkosítani a dokumentumainkat. Sok dokumentum esetén ez nagyon fájdalmas lépés lehet.
- A tanúsítványok változása és kulcsok esetleges kompromittálódása miatt lehet, hogy sokszor újra kell titkosítani a dokumentumainkat.

- A titkosítás nem biztosítja, hogy ha valaki már hozzáfért egy dokumentumhoz, az nem adja tovább másnak a nyílt dokumentumot.

Ha a hozzáférési jogosultságok, illetve a tanúsítványok gyorsan változnak, e megoldás igen rugalmatlanná válhat.

12.4.20. Hozzáférésmenedzsment autentikációs tanúsítványok alapján?

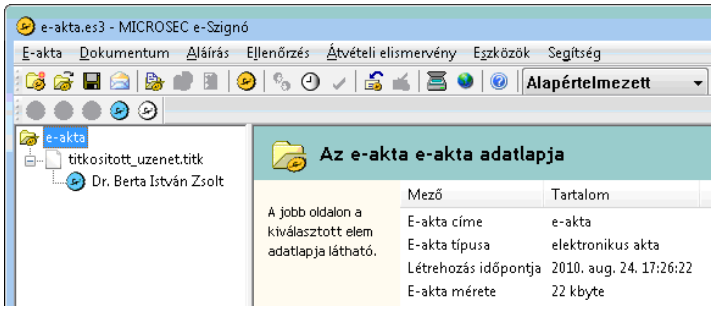
Egy rendszer sokféle módon győződhet meg felhasználói kilétéről, ezt autentikációs tanúsítvány alapján is megteheti. Ugyanakkor az autentikációs tanúsítványt és annak tartalmát célszerű elválasztani attól, hogy az illető milyen jogosultságokkal rendelkezik. Ha a felhasználó szerepkörét vagy jogosultságait beleírjuk a tanúsítványba, azzal összekeverjük az autentikációt és az autorizációt, és megnehezítjük, hogy az autentikációs tanúsítványt más rendszerben is használni lehessen.

Az a szerencsés, ha a tanúsítványt kizárólag annak igazolására használjuk, hogy egy adott entitás birtokolja az adott nyilvános kulcshoz tartozó magánkulcsot. Ha szükségünk van a felhasználó személyes adataira, azt célszerű valamilyen más módon begyűjteni, nem pedig az autentikációs tanúsítványban keresni. Ha meggyőződtünk róla, hogy egy adott felhasználót valóban be akarunk engedni a rendszerünkbe, szerezzük meg a tanúsítványát, és a továbbiakban ezen tanúsítvány alapján bizonyosodhatunk meg róla, hogy ő akar-e belépni.

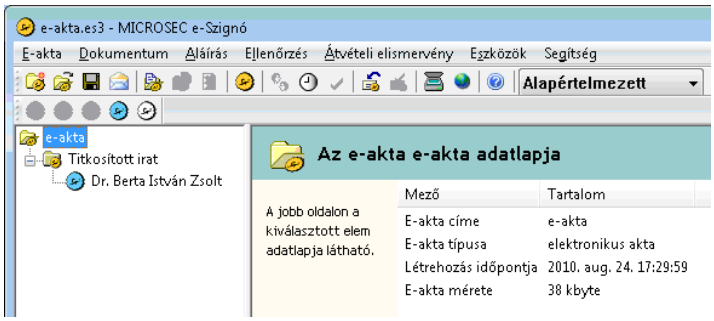
Azt se felejtsük el, hogy az autentikációs tanúsítványok változnak. Lejárhatnak, visszavonásra kerülhetnek, és az új tanúsítványban lehet, hogy már más adatok fognak szerepelni. Ha a rendszerünk a felhasználó tanúsítványát, annak lenyomatát vagy a tanúsítvány sorozatszámát tartja nyilván, akkor a nyilvántartásunkat a tanúsítvány minden változásakor frissíteni kell. Ha a rendszerünk a felhasználó tanúsítványában szereplő megnevezést tartja nyilván, akkor a tanúsítvány (lejárat miatt történő cseréje, azaz) megújítása miatt nem kell frissíteni, hanem csak akkor kell változtatni rajta, ha a megnevezés változik. Ha a megnevezésben szerepel a felhasználó valamilyen egyértelmű azonosítója, elég ezen azonosítót nyilvántartanunk, és ekkor nyilvántartásunkat nem kell gyakran változtatnunk. E megoldás szolgáltató-függő, de több magyar szolgáltató is feltünteti egy a felhasználót egyértelműen azonosító értéket a megnevezés `serialNumber` elemében.

12.4.21. Elhelyezhetek titkosított dokumentumot archiválás-szolgáltatónál?

Az elektronikus aláírásról szóló, 2001. évi XXXV. törvény (Eat) értelmében működő elektronikus archiválás-szolgáltató köteles ellenőrizni az archiválásra beküldött e-aktán lévő elektronikus aláírást. Így kizárólag olyan e-aktákat fogadhat be, amelyeken az aláírást ellenőrizni tudja.



12.8. ábra. **Egy titkosított fájlon helyeztem el elektronikus aláírást**



12.9. ábra. **Egy titkosított fájlon helyeztem el elektronikus aláírást, amely maga is e-akta. A titkosított e-aktában további fájlok, dokumentumok és aláírások lehetnek**

> *Eat, 16/E. § „ Az elektronikus dokumentumok vagy lenyomatok átvételekor a szolgáltató ellenőrzi az elektronikus aláírást, majd beszerzi az elektronikus aláírás hosszú távú érvényesítéséhez szükséges információkat, így különösen:*
> *a) a tanúsítvánnyal kapcsolatos információkat, az aláírás-ellenőrző adatot, valamint a tanúsítvány aktuális állapotára, visszavonására vonatkozó információkat;*
> *b) az a) pontban felsoroltakon túl a tanúsítvány kibocsátójának szolgáltatói aláírás-ellenőrző adataira és annak visszavonására vonatkozó információkat. ”*

Elképzelhető, hogy az e-aktában lévő aláírt fájl önmagában nem egy értelmes, olvasható dokumentum, hanem egy titkosított üzenet, amelyet csak valamilyen kulcs vagy jelszó birtokában lehet elolvasni. (Lásd: 12.8. ábra.) Az is elképzelhető, hogy a titkosított üzenet maga is egy e-akta, amelyben esetleg további fájlok, dokumentumok, és rajtuk aláírások helyezkedhetnek el. (Lásd: 12.9. ábra.)

Természetesen, ilyen e-akta is elhelyezhető archiválás-szolgáltatónál, de ekkor tekintettel kell lenni arra, hogy az Eat. szerinti archiválás-szolgáltatás nem pusztán az adatok megőrzéséről, hanem a rajtuk lévő elektronikus aláírások hiteles megőrzéséről szól.

Vegyük figyelembe, hogy az archiválás-szolgáltatás kizárólag azokra az aláírásokra vonatkozhat, amelyekhez az archiválás-szolgáltató hozzáfér. Így egy archiválás-szolgáltatóhoz feltöltött e-aktában lévő titkosított fájl belsejében lévő további aláírásokra az archiválás-szolgáltatás nem vonatkozik, ezen aláírások tekintetében az archiválás-szolgáltató elvileg sem tudja ellátni az Eat. 16/E §-ában leírt kötelezettségeit, nem tud igazolást kiállítani (3/2005. IHM r. 43. §) ezen aláírások őrzéséről, így ezen aláírásokhoz nem kapcsolódik az Eat. 4. § (7) szerinti vélelem.

A titkosított fájlok belsejében lévő aláírásokkal kapcsolatban az archiválás-szolgáltatás értelme vész el.

Az archiválás-szolgáltatónál elhelyezett titkosított fájlok belsejében lévő információkra, dokumentumokra, aláírásokra vonatkozóan bármilyen állítás legfeljebb egyedi szakértői vizsgálat keretében tehető. Előfordulhat, hogy egyáltalán nem igazolható a titkosított fájl belsejében lévő aláírás érvényessége, vagy – az alkalmazott titkosítási módszertől függően – esetleg még az sem, hogy e fájl létezett az archiválás pillanatában.

Az elektronikus archiválás-szolgáltatás arra a feltételezésre épül, hogy a hosszú ideig tartó megőrzés során az aláíráshoz használt kriptográfiai algoritmusok elavulhatnak. Ebből adódóan azzal is számolni kell, hogy ezen időtartam alatt a titkosításhoz használt algoritmusok szintén elavulhatnak, így az archiváláskor „erős” titkosítást lehet, hogy később már könnyű visszafejteni. Ebből adódóan hosszú távon nem jelent igazi védelmet, ha titkosított fájlt helyezünk el egy archiválás-szolgáltatónál.

Álláspontunk szerint nem célszerű titkosított információt aláírni, és nem célszerű titkosított fájlban helyezni el az aláírásokat egy archiválás-szolgáltatónál. Ezen aláírásokkal kapcsolatban elvész az archiválás-szolgáltatás értelme, és az aláírások hiteles megőrzésével kapcsolatban minden felelősség visszahull a végfelhasználóra. Úgy látjuk, az archiválás-szolgáltatásnak elsősorban akkor van értelme, ha a szolgáltató megkapja mind a nyílt dokumentumot, mind a rá vonatkozó aláírást.

Nyílt dokumentumok archiválása esetén viszont különösen fontossá válik, hogy az archiválás-szolgáltató „megbízható harmadik fél” szerepet töltsön be, és a szolgáltató ügyfele megbízzon az archiválás-szolgáltatóban.

12.4.22. Van értelme saját, vállalati CA-t működtetni?

A tanúsítvány a hitelesítés-szolgáltató által kiállított igazolás arról, hogy egy adott nyilvános kulcs egy adott végfelhasználóhoz tartozik. A tanúsítvány kiállításához a hitelesítés-szolgáltatónak azonosítania kell a végfelhasználót, ellenőriznie kell a tanúsítványba kerülő adatokat, és meg kell győződnie róla, hogy a kérdéses magánkulcs valóban a végfelhasználó birtokában van-e. Egy szervezet általában ismeri a saját dolgozóit, valamilyen módon már meggyőződött a kilétükről, ismeri a dolgozók adatit, és általában biztonságos módon el tud

juttatni hozzájuk egy magánkulcsot. Jogosan merül fel a kérdés, miért fizessen ezért egy harmadik félnek, van-e értelme, illetve mikor van értelme bevonni ebbe egy harmadik felet is. Miért fizessen valaki egy hitelesítés-szolgáltatónak olyan információk igazolásáért, amelyeket ő már úgyis jobban tud?

12.4.22.1. Aláírás, titkosítás és autentikáció esete

Aláíró tanúsítványok esetén a belső, vállalati CA-ra visszavezethető aláírásokhoz nem feltétlenül fűződik jogkövetkezmény. Egy belső CA legfeljebb zárt körben használható fokozott biztonságú elektronikus aláírás létrehozására alkalmas tanúsítványt bocsáthat ki, de ezt is csak akkor, ha betartja az elektronikus aláírásról szóló törvényben és a hozzá kapcsolódó rendeletben szereplő követelményeket. Ilyen követelmény például a tanúsított HSM használata, ilyenek a regisztrációs eljárásra vonatkozó követelmények (pl. meg kell adni az előírt tájékoztatást az aláírónak), illetve az aláírókulcsot az aláíró csak aláírás létrehozására használhatja. [180], [80]

A belső, vállalati, zárt körben működő, nem minősített hitelesítés-szolgáltatóra épülő, zárt körben használható fokozott biztonságú elektronikus aláírásokkal kapcsolatban probléma lehet, ha az aláírást olyan jogvitában kívánjuk felhasználni, ahol a hitelesítés-szolgáltató nem független fél. Ilyen lehet például egy munkaügyi per a vállalat és a dolgozója között. Itt előfordulhat, hogy a dologzó arra hivatkozva tagad le egy aláírást, hogy azt a vállalat rendszergazdája hozta létre, akár úgy, hogy hozzáfért az ő magánkulcsához, akár úgy, hogy a hozzájárulása nélkül bocsátott ki egy tanúsítványt, és annak magánkulcsával készítette az aláírást. Hasonló eset lehet például egy, a vállalat és az ügyfele közötti jogvita.

A titkosításhoz és az autentikációhoz nem fűződik jogkövetkezmény, a kulcsnak itt csak műszaki szerepe van. Titkosító tanúsítvány esetén ekkor az is biztosítható, hogy a tanúsítványhoz tartozó magánkulcs a vállalatnál van lététben, és nem egy harmadik félnél. Titkosító és autentikációs tanúsítványok esetén egy külső hitelesítés-szolgáltató jogi szempontból kevés hozzáadott értéket jelenthet.

Akkor juthat mégis szerephez egy külső hitelesítés-szolgáltató, ha az is fontos, hogy a vállalaton kívüli, harmadik felek is zökkenőmentesen elfogadják a tanúsítványokat. Például webszerver tanúsítványt célszerű professzionális CA-tól vásárolni, mert a belső CA által kibocsátott tanúsítványokat a weblapot látogatók nem fogják elfogadni. Szintén célszerű külső CA-hoz fordulni olyan, elektronikus levelezésre használható tanúsítványokkal kapcsolatban, amelyeket sok ügyfélnek kell használnia. (Például ha aláírt e-maileket küld valaki az ügyfeleinek, ezt olyan tanúsítványra célszerű építeni, amelyet elfogadnak az ügyfelek levelezőprogramjai.) Ha csak kevés partnerrel tartunk kapcsolatot, megtehetjük, hogy egyeztetjük velük a saját, belső CA-ink által kibocsátott tanúsítványokat, és ők beállítják őket saját alkalmazásaikban. Ha ez nem jelent túl nagy kényelmetlenséget, e megoldás is jól használható, nem jelent kisebb biztonságot, mint a külső hitelesítés-szolgáltató használata.

12.4.22.2. Felülhitelesítés, kereszthitelesítés

Szóbajöhető megoldás, ha saját CA-nkat felülhitelesíttetjük egy professzionális hitelesítés-szolgáltatóval. Ekkor a tanúsítványokat ellenőrző érintett fél úgy fogja látni, mintha a saját CA-nk által kibocsátott tanúsítványt e professzionális hitelesítés-szolgáltató bocsátotta volna ki:

- Ha egy Eat. szerinti, nyilvánosan működő hitelesítés-szolgáltató hitelesíti felül saját CA-nkat, akkor a belső CA-nk által kibocsátott tanúsítványok alapján is Eat. szerinti, bárki által elfogadható elektronikus aláírások hozhatóak létre.

- Ha egy olyan hitelesítés-szolgáltató hitelesíti felül saját CA-nkat, amelynek gyökerét alapértelmezetten tartalmazzák egyes alkalmazások, akkor – az adott alkalmazások korlátait figyelembe véve – az általunk kibocsátott tanúsítványokat is alapértelmezetten el fogják fogadni ezen alkalmazások.

E megoldásnak súlyos korlátja is van: a saját tanúsítványainkat az érintett fél ezentúl úgy látná, mintha a másik hitelesítés-szolgáltató bocsátotta volna ki, így a másik hitelesítés-szolgáltató valamilyen értelemben mindenképpen felel a mi tevékenységünkért. Egy professzionális hitelesítés-szolgáltató csak akkor fog belemenni a felülhitelesítésbe, ha meg tud bizonyosodni róla, hogy a mi CA-nk is korrektül, azaz a rá vonatkozó szabályoknak megfelelően működik. A felülhitelesítő szolgáltató bele fog szólni, hogy milyen algoritmusokat használunk, milyen eljárásrend szerint regisztráljuk felhasználóinkat, milyen eljárásrend szerint bocsátjuk ki a tanúsítványokat. Meg fogja követelni, hogy nyilvános hitelesítési rend szerint bocsássuk ki a tanúsítványokat, és követelményeket fogalmazhat meg e hitelesítési rendre is. Egy Eat. szerint működő felülhitelesítő szolgáltató meg fogja követelni, hogy tételesen tartsuk be az Eat. követelményeit, így pl. minősített HSM védje a magánkulcsunkat stb. Végül a felülhitelesítő szolgáltató várhatóan ellenőrizni akarja, hogy valóban be is tartjuk-e az előírásokat, így vagy rendszeresen auditálni akarja saját belső CA-nkat, vagy megköveteli, hogy rendszeresen végeztessünk el egy ugyanolyan auditot, mint amilyet neki is el kell végeztetnie, és rendszeresen mutassuk be az ezen audit során képződő audit jelentést.

A Nemzeti Média- és Hírközlési Hatóság tájékoztatója szerint, ha valaki nyilvánosan működő hitelesítés-szolgáltatást nyújt, akkor ezt be kell jelentenie a Hatóságnak, és kérni kell a nyilvántartásba vételt, mert ha valaki egy nyilvánosan működő hitelesítés-szolgáltatóval láncoltatja magát, akkor maga is nyilvános szolgáltatást nyújt. (A felülhitelesítésnek csak akkor van értelme, ha valaki nyilvánosan használható tanúsítványokat bocsát ki.) [118]

Egy professzionális hitelesítés-szolgáltató által való felülhitelesítés jelentős adminisztratív terhet jelent, sok esetben jobban megéri megvásárolni a szükséges tanúsítványokat.

12.4.22.3. Milyen lehetőségek jönnek szóba?

Egy belső CA viszonylag egyszerűen kiépíthető, akár a Microsoft CA, akár egy ingyenesen letölthető CA szoftver (pl. tinyCA, OpenCA, EJBCA) segítségével. Ennél jelentősen több erőforrást, és a műszaki lépéseken túl jogi, szabályozási lépéseket is igényel, ha azt szeretnénk, hogy e CA megfeleljen az Eat. szerinti, zárt körben működő, nem minősített hitelesítés-szolgáltatókra vonatkozó követelményeknek. A következő lépcső, ha a CA nyilvánosan szeretne működni, azaz nem csak egy zárt kör által elfogadott aláírások, hanem bárki által elfogadható aláírások létrehozására alkalmas tanúsítványokat szeretne kibocsátani. A nyilvános szolgáltatás indítását be kell jelenteni a Nemzeti Média- és Hírközlési Hatóságnak. E lépcsőhöz már célszerű igénybe venni egy professzionális hitelesítés-szolgáltató segítségét. Egy minősített hitelesítés-szolgáltató létrehozása messze meghaladja egy vállalati CA-ra szánható erőforrásokat, a minősített tanúsítványokat általában sokkal olcsóbb egy meglévő minősített hitelesítés-szolgáltatótól beszerezni.

A következő alternatívák merülnek fel:

- A vállalat minden tanúsítványt egy, a Nemzeti Média- és Hírközlési Hatóság nyilvántartásában szereplő szolgáltatótól szerez be.

- A vállalat az aláíró tanúsítványokat egy, a Nemzeti Média- és Hírközlési Hatóság nyilvántartásába vett szolgáltatótól szerzi be, a titkosító és autentikációs tanúsítványokat a belső, vállalati CA-val hozza létre.

  A belső CA-t vagy felülhitelesítteti egy széles körben elterjedt gyökérrel rendelkező hitelesítés-szolgáltatóval, vagy nem.

- A vállalat csak a minősített tanúsítványokat szerzi be a Nemzeti Média- és Hírközlési Hatóság nyilvántartásában szereplő szolgáltatótól, a titkosító és autentikációs tanúsítványokat, valamint a zárt körben használható aláíró tanúsítványokat a belső, vállalati CA hozza létre.

- A vállalat csak a minősített tanúsítványokat szerzi be az Nemzeti Média- és Hírközlési Hatóság nyilvántartásában szereplő szolgáltatótól, a titkosító és autentikációs tanúsítványokat, valamint az aláíró tanúsítványokat a belső, vállalati CA hozza létre.

  A belső CA-t vagy felülhitelesítteti egy, széles körben elterjedt gyökérrel rendelkező hitelesítés-szolgáltatóval, vagy nem. Ha e szolgáltató Eat. szerinti, nyilvánosan működő szolgáltató, akkor a belső CA is azzá válik, és a belső CA által kibocsátott tanúsítványokra épülő aláírások is nyilvánosan használhatóakká válnak.

- A vállalat minden tanúsítványt a saját CA-jával hoz létre, és kezdeményezi, hogy a saját CA kerüljön bele a Nemzeti Média- és Hírközlési Hatóság nyilvántartásába.

Ha minősített tanúsítványt is ki szeretne bocsátani, nagyon költséges lehet teljesítenie a szükséges követelményeket. Vagy megkezdi saját gyökerének terjesztését, vagy felülhitelesítteti magát egy széles körben elterjedt gyökérrel rendelkező hitelesítés-szolgáltatóval.

12.4.23. Végezhetem én a regisztrációt a HSZ helyett?

Előfordulhat, hogy egy vállalat saját maga akarja azonosítani a dolgozóit, ügyfeleit, viszont nem akarja ellátni a hitelesítés-szolgáltató műszaki feladatait. Ekkor a következő megoldások merülnek fel:

- Egyik lehetőség, hogy a vállalat egy meglévő hitelesítés-szolgáltató ún. regisztrációs szervezete lesz. Ehhez olyan szerződést kell kötnie a szolgáltatóval, amely szerint a hitelesítés-szolgáltató regisztrációját e vállalat is elvégezheti. Ekkor kívülről nézve a szolgáltató bocsátja ki a tanúsítványokat, de az azonosítással és ügyintézéssel kapcsolatos lépéseket e vállalat maga végzi. Lényeges, hogy ekkor a szolgáltató felelőssé válik e vállalat által végzett regisztrációért, így várhatóan ellenőrizni akarja majd a vállalat munkáját. Szintén megemlítendő, hogy a regisztrációt végző vállalat ekkor az Eat. szerint a szolgáltató részének tekinthető, így a szolgáltatóra vonatkozó szabályok rá is vonatkoznak majd. Például a Nemzeti Média- és Hírközlési Hatóság is kiszállhat a regisztrációt végző vállalathoz, és ellenőrizheti a munkáját.

- Másik lehetőség, hogy a vállalat formailag hitelesítés-szolgáltatóvá válik, de a technikai feladatok ellátásával egy már meglévő, professzionális hitelesítés-szolgáltatót bíz meg. E professzionális szolgáltató várhatóan jelentős segítséget tud nyújtani a hitelesítés-szolgáltatóvá válás adminisztratív lépéseiben. E megoldás szerint a professzionális hitelesítés-szolgáltató lesz a vállalat alvállalkozója, így várhatóan majd ő akarja ellenőrizni, hogy a professzionális hitelesítés-szolgáltató valóban megfelelően működik-e.

  Ettől függetlenül, a két szolgáltató bármilyen struktúrában felülhitelesítheti, illetve kereszthitelesítheti egymást, és ez további alá- és felérendeltségi viszonyokat eredményezhet.

Megjegyezzük, hogy a CA hierarchiát tekintve nem feltétlenül van különbség a két megoldás között. Az első esetben is elfordulhat, hogy a hitelesítés-szolgáltató egy dedikált hitelesítő egységet hoz létre, és a regisztrációt végző vállalat tanúsítványait ezen egységgel bocsátja ki. Ez a hierarchia pont olyannak látszik, mintha amikor a második esetben a professzionális hitelesítés-szolgáltató felülhitelesíti az új hitelesítés-szolgáltatót.

12.4.24. Mi a teendő algoritmusváltás esetén?

Algoritmusváltásra akkor van szükség, amikor várható, hogy az eddig használt kriptográfiai algoritmusok hamarosan elavulnak.

- A minősített archiválás-szolgáltatónál elhelyezett dokumentumokkal és aláírásokkal nincsen teendő, esetükben a szolgáltató végzi az archiválást.

  Az is megoldás, ha egy közelgő algoritmusváltás esetén minden dokumentumunkat elhelyezzük egy minősített archiválás-szolgáltatónál.

- A nem archiválás-szolgáltatónál archivált aláírásokon új időbélyeget célszerű elhelyezni. Lényeges, hogy az új időbélyeg ne az elavuló algoritmusokra épüljön, mert különben nem sokat ér a felülidőbélyegzés.

  Ennek akkor van értelme, ha az új időbélyeg védi az elavuló algoritmussal védett elemeket. Például ha az aláíráshoz használt hash algoritmus avul el, akkor az elavuló algoritmussal védett összes dokumentumot újra kell hash-elni, és az új időbélyegnek az új hash-eket is védenie kell.

  Például a következő megoldások merülnek fel:

  – A kiterjeszthető aláírásokat (pl. XAdES) egyenként is lehet időbélyegezni, ekkor azok külön-külön is ellenőrizhetőek maradnak. Ez általában azt jelenti, hogy az AdES-A aláírások érvényességi láncát egy új időbélyeggel hosszabbítjuk meg, illetve a még nem AdES-A aláírásokból AdES-A aláírásokat készítünk.
  – A nem kiterjeszthető aláírásokat (pl. ISO 32000 szerinti PDF aláírás) az aláírt dokumentumokkal együtt például ZIP fájlokba fogjuk össze, és ezen fájlokon helyezünk el időbélyeget. Így kevesebb időbélyeget fogyasztunk, és e megoldás logikailag megfelelő. Ugyanakkor az így archivált aláírásokat nem lehet majd szabványos módon ellenőrizni, érvényességük legfeljebb szakértői tevékenység keretében látható be.

    Az ilyen megoldásokat célszerű inkább az archiválás-szolgáltatókra hagyni. Ezen kívül általánosságban is igaz, hogy amikor a hosszú távú archiválás szempont, akkor célszerű kerülni a nem kiterjeszthető aláírás-formátumokat.

12.4.25. Mire kell ügyelni PKI rendszerek tesztelésekor?

Nem könnyű feladat megbízható rendszereket fejleszteni és üzemeltetni. Ezért az éles rendszerek mellett külön teszt rendszereket szokás működtetni, és minden változást először a teszt rendszeren szokás kipróbálni, mielőtt az az éles rendszeren is megtörténik.

A PKI, illetve elektronikus aláírás alapú rendszerek tesztelése különös megfontolásokat igényel. Ekkor az információ nem attól hiteles, hogy az az éles rendszerben van, hanem például

attól, hogy magánkulcsunkkal kódolva elektronikus aláírást helyeztünk el rajta, és okiratba foglaltuk. E gondolatmenetet követve, *ha egy éles rendszerről készítünk egy pontos másolatot*, és az így kapott rendszerben az éles kulcsok szerepelnek, akkor *a kapott rendszer nem teszt rendszer, hanem egy másik éles rendszer* lesz, hiszen éles, hiteles okiratokat készít.

Tesztelésre használt rendszerben sokszor nem használhatjuk az éles rendszer kulcsait. A magánkulcsokat azért nem, mert azzal éles okiratokat hoznánk létre. A nyilvános kulcsokat azért nem, mert várhatóan a rendszer olyan funkcióit is tesztelni szeretnénk, amelyekhez nem rendelkezünk éles bemeneti dokumentumokkal. Szintén problémát jelenthet, ha egy rendszernek éles tanúsítványokkal szeretnénk teszt bemeneteket adni, mert ekkor a tesztelést végző személyeknek esetleg olyan dokumentumokat kell – éles – aláírással ellátni, amelyeket egyébként nem írnának alá. Ha egyes funkciók tesztelésekor minősített aláírást kell létrehozni, vegyük figyelembe, hogy ez minden egyes alkalommal PIN begépelésével járhat, így a nagy tömegű tesztelés minősített tanúsítványokkal nem feltétlenül oldható meg automatizáltan.

Egy PKI-re épülő rendszer teszteléséhez általában szükség van egy teszt CA hierarchiára, amely teszt tanúsítványokat szolgáltat a rendszer számára. A teszt rendszert teszt kulcsokkal kell ellátni, hogy ne éles, hanem teszt okiratokat hozzon létre, és teszt tanúsítványok alapján teszt bemeneti adatokat kell generálni a rendszer számára, és azt kell vizsgálni, hogy a teszt rendszer megfelelően dolgozza-e fel őket.

A legtöbb szolgáltató biztosít teszt tanúsítványokat és teszt hierarchiát. Ugyanakkor egy korrektül működő szolgáltató várhatóan nem lesz hajlandó az éles hierarchiájával olyan tanúsítványt kibocsátani, amely nem valós adatokat tartalmaz, és ezt akkor sem teheti meg, ha nekünk csak tesztelési célból lenne szükségünk a tanúsítványra.

Ha egy rendszer teszt tanúsítványokkal működik, abból nem minden esetben következik, hogy éles tanúsítványokkal is működni fog. Előfordulhat, hogy a teszt és az éles tanúsítványok miatt markánsan másképp kell konfigurálni az éles és a teszt rendszert (más gyökereket kell beállítani, máshonnan kell beszerezni a CRL-eket, OCSP válaszokat és időbélyegeket, másképp kell használni intelligens kártyákat). E kérdéskörre gondosan fel kell készülni, és akkor nem ér bennünket hirtelen meglepetés, amikor egy kifejlesztett rendszert éles üzembe akarunk állítani.

12.5. Lehet elektronikus aláírást hamisítani?

„Hic sunt dracones”
(Itten sárkányok vannak)

– titokzatos, ismeretlen területek jelölése régi térképeken

Gyakran merül fel a kérdés: Lehet-e hamisítani az elektronikus aláírást?

A válasz egyszerű: Igen, lehet. Általában is igaz, hogy amit ember alkotott, azt ember meg is „hackelheti".

Az aláíráshoz használt kriptográfiai algoritmusok nem tökéletesek (2. fejezet), előfordulhat, hogy valaki sikeres támadást intéz ellenük. Ahogy telik-múlik az idő, fejlődik a tudomány és a technológia, a korábban biztonságosnak számító algoritmusokról gyakran ki is derül, hogy már nem nyújtanak kellő biztonságot, ezért speciális óvintézkedésekre lehet szükség (8. fejezet). De az elektronikus aláírás hamisítása nem feltétlenül a kriptográfiai algoritmusok támadását jeleni. Az aláírás biztonságában számos egyéb tényező is szerepet játszik, és általában sokkal könnyebb ezeket támadni.

Mielőtt részletesebben megvizsgáljuk a kérdést, definiáljuk, mit is értünk az aláírás hamisítása alatt!

Alajos a jó szándékú aláíró, Bendegúz az aláírást ellenőrző érintett fél. Alajos nem szeretné aláírni a d dokumentumot. Manfréd, a támadó, nem ismeri Alajos magánkulcsát, de azt szeretné, hogy Bendegúz azt higgye, Alajos aláírta a d dokumentumot. E cél érdekében szeretne konstruálni egy aláírást a d dokumentumra, illetve e cél érdekében próbál becsapni más szereplőket.

Lényeges, hogy Manfréd célja nem az, hogy kriptográfiailag érvényes aláírást hozzon létre Alajos nevében, hanem az, hogy sikeresen becsapja Bendegúzt. Látni fogjuk, hogy ezen utóbbi cél sokkal könnyebben elérhető. Az alábbiakban egy úgy nevezett „támadási fát" mutatunk be. [160] E támadási fa a lehetséges támadások egyfajta strukturált csoportosítására és esetleg értékelésére, összehasonlítására szolgál.

A fa gyökerében a támadó célja szerepel. Minden egyes csomópont (beleértve a gyökeret) gyermekeiben azon alternatívák szerepelnek, ahogy a csomópontban megfogalmazott cél megvalósítható. Az egyes alternatívák tovább és tovább oszthatóak, így a fa leveleiben a konkrét támadások szerepelnek. A levelekben lévő konkrét támadásokhoz hozzárendelhető a támadás költsége, így az egyes csomópontokban lévő célok költsége a gyermek csomópontokban lévő költségek minimuma. Így a gyökérhez, azaz a támadó céljának eléréséhez szükséges költség a legalacsonyabb támadáshoz szükséges költséget jelenti. Ha a támadó racionálisan gondolkozik, várhatóan ezt a lehetőséget fogja választani.

Az alábbi támadási fával nem kívánunk minden lehetséges támadást teljes körűen felsorolni, pusztán illusztrálni szeretnénk, hogy a fának milyen rafinált ágai-bogai lehetnek.

Manfréd célja: *Bendegúz higgye azt, hogy Alajos aláírta a d dokumentumot.*

1. *A kriptográfiai algoritmusok támadása*:

    a. A nyilvános kulcsú kriptográfiai algoritmus támadása:

        i) Alajos magánkulcsának meghatározása Alajos nyilvános kulcsa alapján.

ii) Egy Alajos nyilvános kulcsa szerint érvényes aláírás kiszámítása az Alajos által korábban készített aláírások alapján.

b. A hash függvény támadása:

i) Olyan d' dokumentum konstruálása, amely tartalmában megegyezik a d dokumentummal (így Alajos magától nem írná alá), és lenyomata megegyezik egy Alajos által korábban már aláírt dokumentum lenyomatával. Ez a hash függvény második őskép-ellenállóságának (2.4. fejezet) támadását jelenti.

ii) Olyan d', és x dokumentum konstruálása, ahol d' tartalmában megegyezik a d dokumentummal (így Alajos magától nem írná alá), míg x „ártatlan" dokumentum, amelyet Alajos szívesen aláír, és d' és x lenyomata megegyezik. Ez a hash függvény ütközés-ellenállóságának (2.4. fejezet) támadását jelenti.

Ha a felhasznált kriptográfiai algoritmuskészlet biztonságos, akkor a tudomány és a technológia mai állása szerint a fentiek mindegyike[10] nagyon nehéz feladat. Ez azt jelenti, hogy Manfréd csak elenyésző valószínűséggel és/vagy irreális erőforrások befektetésével érhetne el sikert.

2. *Alajos aláírás-létrehozó eszközének (intelligens kártyájának) támadása*:
Ez azt jelenti, hogy Manfréd valahogy hozzáfér Alajos aláírás-létrehozó eszközéhez (pl. el kell lopnia), majd:

a. kinyeri az eszközből a PIN kódját.

b. kinyeri az eszközből a magánkulcsot.

c. olyan módon manipulálja az eszközt, hogy az a PIN kód nélkül is készítsen aláírást.

d. megpróbálja kitalálni az eszköz PIN kódját.

Az intelligens kártyákat úgy alakították ki, hogy azok nagyon erősen védik a magánkulcsot, és a PIN kódot kinyerni vagy az ellenőrzést megkerülni nagyon nehéz (6.3.3.2. fejezet). E támadások végrehajtása vélhetően könnyebb, mint a kriptográfiai algoritmusokat támadni, de mély ismereteket és drága eszközöket igényel, így nagyon nagy szervezetek számára lehet reális.

Ezzel szemben, bárki megpróbálhatja kitalálni a PIN kódot. Ha Manfréd egy 6 jegyű PIN-t próbál kitalálni, és 3-szor próbálkozhat, akkor véletlenszerű próbálkozással 3: 1 000 000 eséllyel ér el sikert. Ez akkor teljesül, ha Alajos jó PIN kódot választott. Nem szerencsés, ha Alajos PIN kódja az ő születési dátuma, vagy a PIN 123456, mert könnyen lehet, hogy a támadó az ehhez hasonló kódokat próbálja ki, és nem véletlenszerűen próbálkozik. E kérdéskörrel az aláírót érő támadásoknál foglalkozunk.

[10]Közülük ma talán a hash függvény ütközés-ellenállóságának támadása tűnik a legkönnyebb, míg a második őskép-ellenállóság támadása tűnik a legnehezebb feladatnak, de ez nem változtat azon, hogy a fenti problémák mindegyike nagyon-nagyon nehéz, csak irreális erőforrásokkal lenne megoldható.

Lényeges, hogy Manfrédnek jellemzően nincsen korlátlan ideje e támadás végrehajtására. Ha Alajos észleli, hogy Manfréd hozzáfért az aláírás-létrehozó eszközhöz, akkor kérheti a hitelesítés-szolgáltatót, hogy függessze fel a tanúsítványt.

3. *Alajos számítógépének támadása*:
   A támadó megpróbálhatja az irányítása alá vonni Alajos számítógépét. (Az egyszerűség kedvéért tekintsük Alajos számítógépe részének minden olyan szoftver és hardver eszközt, amelyet Alajos az aláírás készítéséhez használ, leszámítva a kulcsot tároló intelligens kártyát.) Ez történhet tisztán szoftveresen, amikor egy vírust vagy trójait juttat a gépre, de történhet hardveresen is, amikor beépít valamit a gépbe, vagy például lecserél bizonyos alkatrészeket. Ekkor:

   a. Lehallgathatja Alajos PIN kódját, így ha később ellopja a kártyát, alá tud írni Alajos nevében. Ha Alajos PIN pades kártyaolvasót használ, azzal jelentősen megnehezítheti a támadást, mert a PIN ekkor nem jut be a számítógépbe. Manfréd ekkor is próbálkozhat a kártyaolvasó manipulálásával vagy lecserélésével.

   b. Ha sikeresen lehallgatta Alajos PIN kódját, akár Alajos tudta nélkül is aláírathat a kártyájával, anélkül, hogy azt ellopta volna Alajostól. A PIN pades olvasó ezen támadás ellen is segít: ha a PIN-t az olvasó PIN padén kell megadni, a támadó nem írhat alá Alajos tudta nélkül.

   c. Aláíráskor lecserélheti az intelligens kártyának küldött lenyomatot. Így, ha Alajos az x dokumentumot próbálja aláírni, a támadó az x dokumentum lenyomatát a d dokumentum lenyomatára cseréli, és ezt küldi el a kártyának. Ha elég ügyesen manipulálta Alajos számítógépét, akkor azt is elérheti, hogy ha Alajos megvizsgálja akár aláírás előtt, hogy mit készül aláírni, akár aláírás után, hogy mit írt alá, akkor a számítógépe az x dokumentumot, illetve az x dokumentumhoz tartozó aláírást mutatja majd.

      E támadás ellen egyedül az segít, ha Alajos egy megbízható eszközön tudja ellenőrizni, hogy mit fog aláírni. Ha nem áll rendelkezésre megbízható eszköz, és Manfréd Alajos minden szoftverét vagy hardverét az irányítása alá vonta, akkor Alajosnak nagyon-nagyon nehéz kivédenie a támadást. [10]

   Ha a támadó csak szoftveresen támad, akkor is támadhatja az aláírás-létrehozó alkalmazást (manipulálhatja, vagy írhat egy másikat, amelyik úgy néz ki, mintha az igazi lenne), az intelligens kártya meghajtó programjait, az operációs rendszert is. Az aláíró számítógépének támadása sokkal könnyebb feladat az eddigieknél. Jelentősen kevesebb speciális szaktudást igényel, a támadás nagy része jól dokumentált felületeken hajtható végre.

4. *Az aláíró támadása*:

A támadó megpróbálhatja rávenni az aláírót, hogy mégis írja alá a d dokumentumot.

a. Megtéveszteti, például:

   i) Manfréd írhat Alajosnak egy e-mailt Alajos titkárnője nevében, és kérhet egy aláírást a csatolt dokumentumra. Ha szerencséje van, Alajos nem olvassa el, mit készül aláírni.

   ii) Manfréd telefonon felhívhatja Alajost, mint Alajos főnöke, és nyomást gyakorolhat Alajosra, hogy írja alá a d dokumentumot.

b. Megzsarolhatja vagy megfenyegetheti, megkínozhatja.

c. Megvesztegetheti. (Megjegyezzük, Alajos számára talán ez a megoldás a legkellemesebb.)

d. Ismerve Alajost, megpróbálhatja kitalálni, milyen jelszót/PIN-t választott, hol tartja a kártyáját stb. Ehhez betörhet Alajos lakásába, átkutathatja a tárcáját stb.

Az eddigi támadásokhoz képest ez az egyik legkönnyebb, egyes támadások semmilyen technikai felkészültséget nem igényelnek. A biztonsági rendszerek egyik leggyengébb pontja szinte mindig az ember, a végfelhasználó.

Megjegyezzük, a fenti támadások jelentős része papír alapon is ugyanúgy működik.

5. *Az aláírást ellenőrző érintett fél támadása*:

   a. A támadó megpróbálhatja rávenni az érintett felet, tekintse úgy, mintha Alajos aláírta volna a d dokumentumot. Az Alajos ellen bevethető módszereket (megtévesztés, zsarolás, megvesztegetés) itt is alkalmazhatja.

   b. A támadó kihasználhatja, ha az érintett fél nem elég körültekintően ellenőrizi az aláírásokat.

      i) Ha Manfréd tudja, hogy Bendegúz nem ellenőriz visszavonási állapotot, akkor Alajos egy régi magánkulcsa és visszavont tanúsítványa szerint is támadhat.

      ii) Ha Manfréd tudja, hogy Bendegúz csak az aktuális visszavonási állapotot nézi, és nem alkalmaz kivárási időt, esetleg a nyílt utcán kirabolhatja Alajost, elveheti a kártyáját, és kiverheti belőle a PIN kódot is. Alajos hiába függeszti fel azonnal a tanúsítványát, ha Bendegúz ezt nem veszi észre.

      iii) Manfréd egészen egyszerű támadásokkal is próbálkozhat. Például készíthet egy képet, amely egy e-aktát ábrázol, benne a d dokumentummal és rajta Alajos aláírásával. A képet elnevezi `nyilatkozat.es3.png`-nek, és elküldi Bendegúznak.

      Bendegúz levelezőprogramja nem jeleníti meg a fájl kiterjesztését, így nem látja, hogy PNG képfájlt kapott. Kettőt kattint a csatolmányon, és látja, hogy

Alajos aláírta a d dokumentumot, és ha nem figyel oda, esetleg nem is megy tovább, hanem becsukja az ablakot, és nem veszi észre, hogy csak egy képfájlt látott.

iv) Manfréd a saját álneves tanúsítványa szerinti aláírást is küldhet, amelyben az általa választott álnév „Alajos". Az Eat. értelmében a hitelesítés-szolgáltatónak fel kell tüntetnie, hogy álnév szerepel a tanúsítványban, de Manfréd alapozhat arra, hogy Bendegúz nem veszi észre, hogy álneves tanúsítvánnyal van dolga.

c. A támadó manipulálhatja az érintett fél számítógépét, hogy azt jelezze, hogy a d dokumentumot Alajos valóban aláírta. Itt is támadhatja a hardvert és a szoftvert is. Utóbbi esetben akár az operációs rendszert, akár az aláírás-ellenőrző alkalmazást.

E támadások hasonló erőforrásokat igényelnek, mint az aláíró ellen indított támadások.

6. *A nyilvános kulcsú infrastruktúra szolgáltatóinak támadása*:

a. Egy hitelesítés-szolgáltató támadása:

i) A regisztrációs folyamat támadása:
A támadó megpróbál tanúsítványt szerezni Alajos nevében. Ehhez:

α) Megpróbálhatja becsapni a hitelesítés szolgáltató regisztrációs munkatársát. Például magát Alajosnak maszkírozva Alajos lopott személyi igazolványával vagy egy hamis személyi igazolvánnyal érkezik a személyes regisztrációra.

β) Igényelhet olyan, alacsony biztonsági szintű tanúsítványt, amelyhez nincsen szükség személyes találkozásra, ekkor könnyebb dolga van. Az ilyen aláírást viszont nem biztos, hogy Bendegúz elfogadja majd, bár Manfréd arra is építhet, hogy esetleg Bendegúz nem ellenőrzi elég körültekintően az aláírásokat.

γ) Olyan eszközökkel is próbálhatja támadni a hitelesítés-szolgáltató regisztrációs munkatársát, amelyekkel az aláírót is támadta (zsarolás, megvesztegetés, fenyegetés). Megjegyezzük, egy jól működő hitelesítés-szolgáltató esetén egyetlen személy egyedül nem adhat ki tanúsítványt, így feltehetően többüket kell támadnia.

ii) A visszavonás-kezelés folyamat támadása:
Tegyük fel, hogy Manfréd sikeresen megszerezte Alajos kártyáját, de nem tudja a PIN kódját. Megpróbálja megtudni a PIN-t is valahogy (például elcsábította Alajos feleségét, és megpróbálja kiszedni belőle), de ehhez időre van szüksége. Meg akarja gátolni, hogy Alajos felfüggeszthesse vagy visszavonhassa a tanúsítványát. Megpróbálhatja túlterhelni a szolgáltató felfüggesztő telefonvonalait, vagy annyiszor hibásan próbálkozni Alajos nevében, hogy

a szolgáltató tagadja meg a felfüggesztési/visszavonási kérelmeket, vagy megpróbálhatja visszaállítani a felfüggesztett tanúsítványokat.

b. Egy időbélyegzés-szolgáltató támadása:
Tegyük fel, hogy Manfréd ellopta Alajos kártyáját, és sikerült megtudnia a PIN kódját. (Tudta, hogy Alajos ugyanezt a kódot használja jelszóként egy közösségi portálon, ott bárhányszor próbálkozhatott Alajos jelszavával, így három nap után megtudta a PIN-t.) Csakhogy, mire megtudta a PIN-t, Alajos már észrevette, hogy eltűnt a kártyája, és a tanúsítványt a hitelesítés-szolgáltató már felfüggesztette.
A támadó készíthet egy aláírást a visszavont tanúsítvány szerint, majd megpróbálhat megtámadni egy időbélyegzés-szolgáltatót, hogy egy visszadátumozott időbélyeget szerezzen az aláírásra, amely (hamisan) igazolná, hogy az aláírás akkor készült, amikor a tanúsítvány még érvényes volt.
Ez nehéz feladat, valószínűleg egy időbélyegzés-szolgáltató több, bizalmi munkakört betöltő munkatársát kellene egyszerre támadnia, vagy be kellene hatolnia a szolgáltató jól védett informatikai rendszerébe.

c. Egy archiválás-szolgáltató támadása:
A támadó próbálhat szerezni egy archiválás-szolgáltatótól olyan igazolást, miszerint a szolgáltató archiválja a d dokumentumot, rajta Alajos érvényes aláírásával. Ez legalább az időbélyegzés-szolgáltató támadásához hasonló nehézségű feladatot jelent.

A szolgáltatók támadása vélhetően jelentősen nehezebb, mind az aláíró vagy az érintett fél támadása, de jelentősen könnyebb, mint a kártya vagy az algoritmusok támadása.

A fenti támadásokhoz általánosságban nehéz költséget rendelni, de azok egymáshoz viszonyított nagyságrendjét a következő módon illusztrálhatjuk: Tegyük fel, hogy a kriptográfiai algoritmusok támadásának költsége 1 millió „egység”. Ekkor az intelligens kártya támadásának költsége – amely függ a kártya típusától, az alkalmazott technológiáktól – kb. 10 ezer és 100 ezer egység között lehet. A kártyát ért támadások közül a PIN vaktában történő kitalálása jelentősen olcsóbban megoldható, de ez csak igen kis eséllyel jár sikerrel. A szolgáltatók támadása jól működő szolgáltatók esetén kb. 1000 vagy 500 egységet igényelhet, míg az aláíró, az aláíró számítógépe és az érintett fél támadása – a támadó felkészültségétől, számítástechnikai ismereteitől, és személyes meggyőző erejétől függően – jócskán 100 egységnyi költség alatt elvégezhető.

Felhívjuk a figyelmet, hogy ezen értékek csak becslésen, „hasra ütésen” alapulnak, nem lehet, illetve nagyon nehéz egzakt módon összevetni a kriptográfiai algoritmusok támadhatóságának költségét egy felhasználó befolyásolásának költségével. A kriptográfiai algoritmusok és az intelligens kártya hardver „érdemi” megtámadása jellemzően akkora költséggel járnak, hogy ezek a reálisan szóba jöhető támadók számára gyakorlatilag lehetetlenek.

E fejezetben mindössze azt kívántuk érzékeltetni, hogy:

- Elektronikus aláírást is lehet hamisítani.
- Ez jellemzően nem a kriptográfiai algoritmusok támadását, hanem a kriptográfiai védelem valamilyen megkerülését jelenti.
- Az elektronikus aláírás sem csodaszer, ahogy a papír alapú aláírások esetén, úgy várhatóan elektronikus aláírásokkal kapcsolatban is lesznek vitás esetek, amelyeket majd egyedi szakértői vizsgálattal, egy bizonyítási eljárás keretében lehet tisztázni.

„Cryptography is typically bypassed, not penetrated"
(A kriptográfiát [a támadók] általában megkerülik, nem pedig megtörik.)
– *Adi Shamir*

12.6. Összegzés

- Akár papíron, akár elektronikusan tároljuk az információt, két koncepció szerint biztosíthatjuk hitelességét:
  - Az információt tároló rendszer zártságát biztosítjuk, és a rendszerben lévő, illetve a rendszerből biztonságos csatornán kinyert információt tekintjük hitelesnek.
  - A dokumentumokat aláírt okiratokba foglaljuk, és az aláírt okiratokat tekintjük hitelesnek, attól függetlenül, hogy azok hol vannak.
- Zárt rendszer esetén:
  - Amint elhagyja az információ a zárt rendszert (illetve a vele kiépített biztonságos csatornát), már nem hiteles.
  - Ha más is támaszkodik a zárt rendszerben lévő információ hitelességére, a rendszernek folyamatosan online kell lennie, és olyan kapacitással kell rendelkeznie, hogy bárki bármikor elérhesse.
  - Ha a rendszer zártsága megszűnik, a benne lévő információ hitelessége megkérdőjelezhetővé válik.
  - Ha a rendszer leáll, senki más nem tudja megállapítani az információ hitelességét. Ha a rendszer megszűnik, valakinek át kell venni a hitelesség biztosítását.
- Meglévő papír alapú rendszereink aláírt okiratokra épülnek.

- Ha meglévő papír alapú rendszereinket elektronizálni szeretnénk, a „zárt rendszer” koncepció gyökeres változtatásokat igényel. Ha a zárt rendszer útját követjük, nem egyszerűen elektronizálni kell, a hitelesség forrásának megváltozása miatt *az elektronizálással együtt egyúttal újra kell tervezni a rendszert érintő összes üzleti folyamatot*. A tapasztalat szerint, az ilyen próbálkozás nagyon meg szokott bukni.

- Elektronikus aláírás alapon történő elektronizálás esetén a meglévő folyamatok használhatóak, mert ekkor továbbra is okiratok jelentik a hitelesség forrását. Kevesebb változtatásra van szükség, és nem kell alapvetően felforgatni a rendszert.

- Elektronikus aláírás bevezetésekor:

  1. Szerezzük meg a kulcsszereplők támogatását, motiváljuk a felhasználókat az aláírás használatára!
  2. Csak annyit változtassunk az üzleti folyamatokon, amennyit feltétlenül szükséges!
  3. Az elektronikus aláírás szervesen épüljön be a folyamatainkba!
  4. Legyen aláírási szabályzatunk, vagy legalább gondoljuk végig, mi kerülne bele!
  5. Megfelelő, perspektivikus aláírás formátumot válasszunk!
  6. Tisztázzuk a kapcsolatot a papír alapú rendszerekkel!

- Ma az elektronikus számlázás jelenti az elektronikus aláírás talán legnagyobb felhasználási területét, de lényegében bárhol használhatnánk a kézzel írott aláírás kiváltására.

- Az elektronikus aláírás sem tökéletes, de sok szempontból jelentősen nagyobb biztonságot jelent, mint a papír alapú aláírás. Kevésbé kérdőjelezhető meg, és a hitelességgel kapcsolatos viták egy része sokkal könnyebben eldönthető.

13. fejezet

Esettanulmányok

FEHERUUARU REA MENEH HODU UTU REA

– A tihanyi apátság alapítóleveléből

13.1. e-Cégeljárás

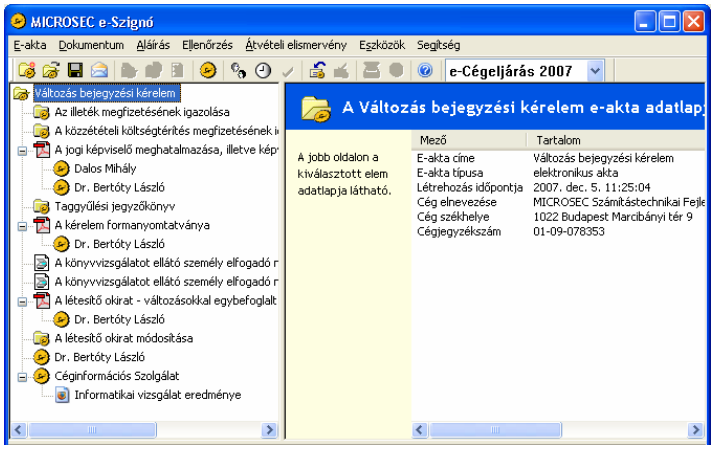
Az elektronikus cégeljárás rendszere az elektronikus aláírás technológia segítségével teszi lehetővé, hogy Magyarországon a cégek, illetve a jogi képviselőjük cégbírósági ügyeiket elektronikus úton is intézhessék. E körbe tartozik például a cégalapítás, a változásbejegyzés, a céginformáció lekérdezése, illetve a cégek beszámolóinak elektronikus benyújtása és közzététele. Ma az elektronikus cégeljárás jelenti a hazai e-aláírás alkalmazások zászlóshajóját.

A cégbíróságokon az 1990-es évek elejétől elektronikus nyilvántartó rendszer működik, és e rendszerből 1995-től az Interneten keresztül is lekérdezhető a céginformáció. A cégbejegyzési eljárás 1998. óta „egyablakos”, azaz elegendő egyetlen ponton kezdeményezni egy új cég alapítását, és a rendszer elvégzi a szükséges kommunikációt az érintett hatóságokkal. E rendszer még az elektronikus aláírásról szóló törvény előtt készült, de már akkor is aláírtan (és titkosítottan), PGP-vel kommunikált az érintett társintézményekkel (APEH, Központi Statisztikai Hivatal).

Később e rendszerbe került integrálásra a PKI alapú elektronikus aláírás technológiája.

2002. óta a pénzintézetek a cégek bankszámlaszám-változásait már fokozott biztonságú elektronikus aláírással jelentik a cégbíróságoknak. Az így beküldött számlaszám-változások eleinte emberi közreműködéssel, később automatizáltan kerültek bele a cégjegyzékbe.

2005. szeptember 1-jétől minősített elektronikus aláírással ellátva, elektronikus úton is be lehet nyújtani a cégbíróságokhoz a cégek bejegyzési és változásbejegyzési kérelmeit. Ez például a következő módon történhet:



13.1. ábra. **Egy cégbíróságra beküldött kérelem, amelyet a cég jogi képviselője elektronikus aláírással látott el, majd a Céginformációs Szolgálat elektronikusan visszaigazolta a kérelem átvételét**

1. Cég alapításához jogi képviselő (ügyvéd, közjegyző vagy jogtanácsos) segítségét kell igénybe venni. A jogi képviselő összeállít például egy változásbejegyzési kérelmet. Az összeállított e-aktát, amely állhat elektronikusan aláírt, vagy a jogi képviselő által szkennelt iratokból ill. ezek kombinációjából is, a jogi képviselő ellátja minősített elektronikus aláírásával, valamint időbélyeggel. Az elektronikus aktát elektronikus aláírás védi, így tartalma nem változtatható meg anélkül, hogy a rajta lévő aláírás sértetlen maradjon. Az elektronikus aktának tartalmaznia kell a Magyar Államkincstár szintén elektronikusan aláírt igazolását a befizetett eljárási illetékről és költségről. Az eljáró cégbíró ebből tudja, hogy az ügyhöz tartozó illetéket és költséget megfizették. [6]

2. A jogi képviselő beküldi az elektronikus aktát a Közigazgatási és Igazságügyi Minisztérium[1] Céginformációs Szolgálatának. A Céginformációs Szolgálat automatizmusai alapvető formai ellenőrzést végeznek a benyújtott e-aktán: ellenőrzik az aktán lévő aláírásokat, illetve ellenőrzik, hogy a előírt című dokumentumok szerepelnek-e az aktában. Ezt követően a Céginformációs Szolgálat automatizmusa fokozott biztonságú aláírással és minősített időbélyeggel ellátott tértivevényt küld a jogi képviselőnek az e-akta átvételéről. A tértivevény tartalmazza a benyújtott e-akta lenyomatát, így a jogi képviselő igazolni tudja, hogy pontosan mit nyújtott be. (Lásd: 13.1. ábra.)

3. A Céginformációs Szolgálat továbbküldi az e-aktát a területileg illetékes cégbíróságra, és az akta egy bíróhoz kerül.

4. A cégbíró minősített elektronikus aláírással ellátott végzést hoz, és a változás bekerül a cégjegyzékbe.

[1]Korábban: Igazságügyi Minisztérium, illetve Igazságügyi és Rendészeti Minisztérium

5. A rendszer elküldi a végzést a jogi képviselőnek, aki minősített elektronikus aláírással igazolja a végzés átvételét.

A dokumentumok szkennelésének, digitalizálásának feladata a cégbíróságokról az ügyvédekhez került át, ami megtakarításokat jelentett. A jogalkotó ösztönözni kívánta az elektronikus cégeljárást, így (azon kívül, hogy lecsökkent a sorban állás) lényeges előnyöket biztosított az elektronikusan eljáró jogi képviselők számára: A közzétételi költségtérítés mértékét cégbejegyzés esetén 25 000 Ft-ról 5 000 Ft-ra, változásbejegyzés esetén 15 0000 Ft-ról 3 000 Ft-ra csökkentette. Ezáltal ha egy ügyvéd egy évben 2-3 cégügyet intézett, akkor már megérte beszereznie az elektronikus aláíráshoz szükséges felszerelést.

2006. július 1-től a Kkt-t, Bt-k, és Kft-k ügyeiben egyszerűsített cégeljárás valósult meg, és elektronikus ügyintézés esetén a bejegyzési határidő 2 napra csökkent. Az új iratok egyre inkább digitalizáltan kerültek be a rendszerbe, a régi cégiratok pedig digitalizálásra kerültek, így a cégiratokat 2007-től 10 évre visszamenőleg elektronikusan is meg lehetett tekinteni. A bejegyző végzések, illetve a cégkivonatok egyre inkább elektronikusan, elektronikus aláírással születtek, így más rendszereknek is fel kellett készülniük az elektronikus dokumentumok fogadására. Mivel az elektronikus eljárás jelentősen kedvezőbb volt a jogi képviselők részére, egyre többen választották ezt a megoldást.

2008. július 1-jétől csak minősített elektronikus aláírással ellátva, elektronikus úton lehet benyújtani a kérelmeket a cégbíróságra, így az elektronikus eljárás e téren kizárólagossá vált. A bejegyzési kérelmek kb. felét már akkor is elektronikusan nyújtották be, amikor e megoldás még nem volt kizárólagos.

2008-ban megváltozott a benyújtandó dokumentumok formátuma is. Míg korábban szkennelt, jellemzően PDF dokumentumokat kellett benyújtani, 2008-tól a bejegyzési kérelem formanyomtatványát már XML formátumban kell beküldeni, amelyet a jogi képviselők nyomtatványkitöltő programok segítségével készítenek. Az XML nyomtatványokat a rendszer automatizáltan feldolgozhatta, így – a jogszabály értelmében – 1 órán belül be kellett jegyezni a céget. [68] Az eljárásban több gyártó nyomtatványkitöltő programját, és több gyártó aláírás-létrehozó alkalmazását lehetett használni.

2008 végétől bekapcsolódtak a rendszerbe a felszámolók, végelszámolók, illetve az önálló bírósági végrehajtók is.

2008. második felében hetente kb. 6 000 elektronikusan aláírt beadvány érkezett a cégbíróságokra, ebből kb. 2 000 esett abba a kategóriába, amelyet 1 órán belül kellett bejegyezni. Hetente kb. 6 000 elektronikusan aláírt végzés született, ennek 95-97%-át elektronikusan vette át az ügyvéd, és csak a fennmaradó esetekben kellett postai tértivevényes levélben kiküldeni a végzést. Ez pusztán a postaköltségen havi 10 milliós megtakarítást eredményezett a cégbíróságokon.

A megoldás azért lehetett sikeres, mert fokozatosan, kis lépésekben került bevezetésre, és

megfelelően motiválta a résztvevőket:

- Először az érintett hatóságok egymás között kezdtek elektronikus aláírással kommunikálni.
- Később megteremtették a lehetőségét, hogy jogi képviselők is küldhessenek be elektronikusan aláírt beadványokat.
- Az így keletkezett megtakarítások egy részét átengedték a jogi képviselőknek, akik érdekeltté váltak az elektronikus eljárásban.
- A rendszer először elektronikus papír iratokkal, PDF-ekkel indult el, a meglévő, papír alapú folyamatok szerint. Ezáltal gyorsan el lehetett indítani a rendszert.
- Az XML alapú intelligens nyomtatványok később, ettől függetlenül kerültek bevezetésre, amikor a rendszer már kb. 3 éve sikeresen működött.
- A folyamatok alapvetően ma is az okirat-alapú megközelítés szerint működnek. Majdnem ugyanaz történik, mint papíron, „csak” elektronizálták, és ezáltal gyorsabbá, olcsóbbá, hatékonyabbá tették.
- A rendszer nyílt specifikációkra épül. XAdES aláírásokat és e-aktákat, valamint nyilvános XML sémákat használ. [51], [36] Nem egyes alkalmazásokat tettek kötelezővé, a rendszerhez bármilyen eszközzel csatlakozni lehet, amely a megadott formátumú fájlokat létre tudja hozni. Ezáltal több hitelesítés-szolgáltató tanúsítványai, időbélyegei, és több alkalmazásfejlesztő aláírás-létrehozó alkalmazásait és nyomtatványkitöltő programjai közül választhatunk.
- A rendszert kötelezővé tették, de csak akkor, amikor már évek óta jól működött, és a beadványok fele már elektronikusan érkezett, a papír alapú beadványok pedig már csak felesleges terhet, visszahúzó erőt jelentettek.

13.2. Önálló bírósági végrehajtók és pénzintézetek kapcsolata

A végrehajtás során az önálló bírósági végrehajtó megkeresi a pénzintézeteket, hogy megtudja, az érintett személynek mely pénzintézeteknél van számlája.

Magyarországon a legtöbb embernek a néhány legnagyobb banknál van bankszámlája, és alig van, aki kisebb pénzintézeteknél, például vidéki takarékszövetkezeteknél tartja a pénzét. Az összes pénzintézet megkeresése munkát és költséget jelentett a végrehajtónak: külön-külön levelet kellett írniuk minden pénzintézetnek, és postai tértivevényes levélben kellett volna elküldeni őket. E kiadás többnyire felesleges volt, mert a legtöbb pénzintézettől úgyis negatív válasz érkezett.

Ezért, amíg a folyamat papíron zajlott, a végrehajtók többnyire nem keresték meg az összes pénzintézetet, hanem csak néhány nagy bankkal vették fel a kapcsolatot. Ennek következtében a kis pénzintézetnél számlát vezető adós esetén a végrehajtó kénytelen volt az inkasszónál költségesebb végrehajtási cselekményt (pl. foglalás, árverés) foganatosítani.

Az önálló bírósági végrehajtók 2009. óta már elektronikusan állnak kapcsolatban a pénzintézetekkel, és üzeneteiket elektronikus aláírás védi. [26] A végrehajtó egyetlen, elektronikus kérelmet ír meg, ezt minősített elektronikus aláírással és időbélyeggel látja el. A kérelmet elküldi egy közvetítő rendszernek, amely szétküldi a kérelmet minden egyes pénzintézetnek. A pénzintézetek elektronikusan, elektronikusan aláírva válaszolnak a közvetítő rendszernek, amely visszajuttatja a válaszokat a végrehajtóknak.

A rendszerben XML formátumú üzenetek közlekednek, így azokat automatizmusok is fel tudják dolgozni. Ezen XML formátuma rögzített, a szereplők különféle gyártóktól származó alkalmazásokat használnak az üzenetek előállítására. Mind végrehajtói, mind a pénzintézeti oldalon több fejlesztő alkalmazása versenyzik egymással.

A végrehajtók célalkalmazásokat használnak az XML üzenetek feldolgozására, és a pozitív válaszok kiválogatására. A rendszerben a negatív válaszok is fontos szerepet töltenek be, mert a végrehajtó így tudja igazolni, hogy valóban megkeresett minden pénzintézetet. A pénzintézetek egy része a banki adatbázishoz hozzáférő automatizmus segítségével tölti ki az XML-t, egy másik részük esetén ember viszi be az adatokat a végrehajtókat válaszoló rendszerbe.

Az XML fájlok átadását-átvételét a felek elektronikus aláírással nyugtázzák, az üzenetek bizalmasságáról PKI alapú titkosítási megoldás gondoskodik.

A rendszer hatására:

- megnőtt a végrehajtás hatékonysága, nehezebb elrejteni a pénzt a végrehajtók elől, ezáltal javult a jogbiztonság.
- a szereplők jelentős összeget takarítanak meg a postaköltségen, tértivevényes levél helyett elektronikus aláírást és időbélyeget használnak. [132]
- ha egy pénzintézet automatizmussal válaszol a megkeresésekre, megtakaríthatja az emberi erőforrásokra fordított költséget.

13.3. Elektronikus aláírás a közigazgatásban?

A közigazgatási hatósági eljárásról szóló 2004. évi CXL. törvény (Ket.) megteremtette a jogi alapját, hogy elektronikus aláírással is lehessen ügyet intézni a közigazgatásban. A törvény két alternatív elektronikus megoldást is adott a papír alapú ügyintézés mellé: az egyik az elektronikus aláírással, a másik a Központi Rendszeren (Ügyfélkapun) keresztül történő

ügyintézés. [181] Később megjelentek a törvényhez kapcsolódó végrehajtási rendeletek, köztük a közigazgatásban használt elektronikus aláírásokra és tanúsítványokra vonatkozó 194/2005. Kormányrendelet és az elektronikus ügyintézést lehetővé tevő informatikai rendszerekről szóló 195/2005. Kormányrendelet. [97], [98]

A jogszabályok megkülönböztettek közigazgatási területen (is) használható tanúsítványokat és elektronikus aláírásokat, és ezekre további követelményeket írtak elő. A közigazgatásban használható tanúsítványokat a kereskedelmi hitelesítés-szolgáltatók bocsátották ki, ehhez auditáltatniuk kellett magukat, hogy megfeleljenek a közigazgatási követelményeknek. Néhány kiemelt személy tanúsítványát nem a kereskedelmi szolgáltatók, hanem az IHM Biztonsági Hitelesítés Szolgáltató bocsátotta (volna) ki. Az alkalmasnak talált szolgáltatókat a Közigazgatási Gyökér Hitelesítés Szolgáltató (KGYHSZ) felülhitelesítette. [94]

Az Informatikai és Hírközlési Minisztérium műszaki specifikációkat is kibocsátott a közigazgatási célú aláírásokkal kapcsolódóan. A specifikációk többsége nagyon alapos, jó minőségű, a nemzetközi szabványoknak megfelelő dokumentum, amely azóta is mértékadónak számít a hazai PKI területén.

Hat hitelesítési rend jelent meg [83] (3 a közigazgatást, és másik 3 a közigazgatás ügyfelei tanúsítványainak számára), elkészült többek között a közigazgatási tanúsítványok formátumára vonatkozó specifikáció [82], elkészült a közigazgatásban használható aláírás formátumára vonatkozó specifikáció [85] és a közigazgatási időbélyegzést leíró specifikáció is. Egy „viszontazonosítás” nevű protokollt is kidolgoztak, amelynek segítségével [86] a közigazgatás meggyőződhetett róla, hogy egy adott tanúsítvány egy adott természetes személyhez tartozik-e.

A nagy erőfeszítés ellenére a rendszer soha nem indult el igazán. A közigazgatási szervek nem (vagy csak elvétve) indítottak elektronikus aláírásra épülő szolgáltatásokat, így az ügyfelek sem vásároltak közigazgatásban használható tanúsítványokat. A kereskedelmi szolgáltatók alig néhány közigazgatásban használható tanúsítványt bocsátottak ki, az IHM Biztonsági Hitelesítés Szolgáltató pedig soha nem kezdte meg a működését.

Miért fulladt kudarcba ez a kezdeményezés?

- A közigazgatás részéről nem volt meg az elkötelezettség az elektronikus aláírás mellett.

  Sem a közigazgatási szerveket, sem az ügyfeleket nem motiválta semmi az elektronikus aláírás használatára. A közigazgatási szervek kibújhattak az e-ügyintézés alól (pl. az önkormányzatok határozatot hozhattak, hogy nem biztosítanak elektronikus ügyintézést), és ezt meg is tették.

  A közigazgatás inkább az Ügyfélkapura épülő e-ügyintézést favorizálta, de – tekintve, hogy itt a hitelességet egyedül a központi rendszer biztonságos és megbízható működése nyújtotta [100] – ez ellenállásba ütközött, és e megoldás sem ért el áttörést.

- Teljesen új PKI rendszert akart létrehozni, és nem vette figyelembe a már meglévő PKI alkalmazásokat.

  A szabályozás arra számított, hogy a meglévő (pl. az e-cégeljárásban használt) PKI előbb-utóbb magától átvált a közigazgatási PKI-be. Ugyanakkor az új rendszer elsősorban a közigazgatás számára lett volna használható: a viszontazonosítás csak a közigazgatás számára volt elérhető, mások számára nem oldotta meg a problémát, és a szabályozás nem fordított gondot arra, hogy mi történik a közigazgatási PKI-n kívül.

  Sikeresebb lehetett volna e megoldás, ha épített volna a meglévő PKI közösségekre, illetve tekintettel lett volna a közigazgatáson kívüli felhasználásra.

- Túlságosan komplex szabályrendszert hozott létre, amelyet csak nagyon kevesen értettek meg. Egy egyszerűbb, kevesebb opciót megengedő rendszer valószínűleg kedvezőbb fogadtatásra talált volna.

2009. októberében a Ket-ből kikerült az elektronikus aláírással történő ügyintézés lehetősége. Később új végrehajtási rendelet jelent meg, amely alapvetően felforgatta a korábbi szabályozást, de a műszaki specifikációk ezt nem követték. [99] Várhatóan hamarosan ismét változni fog a közigazgatási elektronikus aláírásra vonatkozó szabályozás.

13.4. e-Aláírás az útlevelekben

Az Európai Unióban, illetve Magyarországon kibocsátott új típusú (bordó, más néven „burgundi vörös” színű) útleveleken mikrochip helyezkedik el, amelyet az olvasó berendezések távolról, rádiófrekvenciás kapcsolaton keresztül érhetnek el. Az útlevelekben használt biztonsági megoldások nagy mértékben támaszkodnak a nyilvános kulcsú infrastruktúra (PKI) eszköztárára; a továbbiakban ezen elemeket mutatjuk be.

Az EU útleveleinek biztonsági előírásait az Európai Unió Tanácsa 2252/2004/EK rendelete határozza meg. [63] E rendelet, illetve annak műszaki melléklete a Nemzetközi Polgári Repülési Szervezet (ICAO) 9303-as számú (Machine Readable Travel Documents) dokumentumában szereplő követelményekre épül. [130] E követelményrendszert a külföldre utazásról szóló 1998. évi XII. törvényt módosító 2006. évi XXI. törvény építette be a magyar jogrendszerbe. [183]

A következőkben az ICAO által felvázolt rendszer PKI vonatkozásait tekintjük át. Ahol a 9303-as ICAO dokumentumban leírtak nem egy az egyben valósultak meg, ott a gyakorlatban megvalósult megoldást mutatjuk be. A 9303-as ICAO dokumentum egyes elemei opcionálisak, itt a magyar útlevelekkel kapcsolatban megvalósult megoldást írjuk le.

Az ICAO által leírt megoldás két lépcsőben valósult meg: A biometrikus útlevelek első generációján az útlevélen grafikusan is megjelenő adatok szerepelnek elektronikusan, digitálisan aláírva. A biometrikus útlevelek második generációján az útlevél birtokosának

ujjlenyomata is szerepel, de az ujjlenyomathoz kizárólag arra jogosult olvasó berendezések férhetnek hozzá, tanúsítvány-alapú autentikációt követően, biztonságos csatornán keresztül.

13.4.1. Első generáció: Aláírt adattartalom

Magyarország 2006. augusztus 29. óta bocsát ki első generációs biometrikus útleveleket. Ezek elektronikusan is tartalmazzák az útlevélen grafikusan feltüntetett adatokat, így többek között az útlevél birtokosának nevét, állampolgárságát, születési helyét és idejét, nemét, illetve az arcképét. Az első generációs útlevelek esetén az arckép jelenti az egyetlen biometriai információt.

Az elektronikus útlevelekkel az érintésnélküli kártyáknál is használt ISO 14443 szabványban leírt módon lehet rádiófrekvenciás kapcsolatot létesíteni. Az útlevelet úgy alakították ki, hogy e kapcsolat csak akkor épülhet fel, ha az útlevél nyitva van; ez az ún. alapszintű hozzáférés ellenőrzés (basic access control). Az elektronikus információk kiolvasása titkosított kapcsolaton keresztül történik. A titkosított kapcsolat felépítéséhez szükséges kriptográfiai kulcsok származtatásához szükség van az útlevél géppel olvasható területén (machine readable zone) grafikusan feltüntetett információkra is. Ez biztosítja, hogy akinek nem mutattuk meg a nyitott útlevelünket, az elektronikusan sem tud beleolvasni. E kapcsolat felépítése során az olvasókészülék is meggyőződik róla, hogy valóban egy útlevélchippel áll-e kapcsolatban. [27]

Az útlevélchipen szereplő adatokat az útlevelet kibocsátó hatóság aláírással látja el. E digitális aláírás nem az elektronikus aláírásról szóló 2001. évi XXXV. törvény, illetve nem az elektronikus aláírásról szóló EU irányelv szerinti elektronikus aláírást jelent, azaz nem az aláíró kötelezettségvállalását, hanem az információ eredetének hitelességét bizonyítja. Ezen aláírás ellenőrzésével az olvasókészülék arról győződhet meg, hogy valóban létezik-e útlevél az adott adattartalommal. Maga az útlevél nem tartalmaz aláírás-létrehozó vagy ellenőrző funkcionalitást, mindössze aláírt adatokat tárol, így e funkciót passzív autentikációnak is nevezik.

Minden útlevél-kibocsátó ország létrehozott egy-egy dedikált gyökér hitelesítés-szolgáltatót (CSCA, country signing certification authority). Az adott ország által kibocsátott útlevélchipeken szereplő aláírt adatok ezen gyökér egység önhitelesített tanúsítványa alapján ellenőrizhetőek. A CSCA gyökerek „hosszú ideig” érvényesek, a hozzájuk tartozó magánkulcsokat az országok sokáig használják. (Lásd: 13.2. ábra.)

Az egyes országok CSCA gyökér hitelesítés-szolgáltatóihoz tartozó hierarchiák nem kapcsolódnak egymáshoz. Az országok egyenként, biztonságos diplomáciai csatornán küldik el egymásnak a CSCA gyökértanúsítványaikat. Minden ország minden CSCA tanúsítványt telepít minden okmány-ellenőrző készülékre.

A CSCA-k különösen biztonságos környezetben, offline működnek, ők bocsátják ki az okmányokat aláíró egységek (DS, document signer) tanúsítványait. Az útlevélchipekre

13.2. ábra. **Az útlevelekben szereplő adatok aláírására az egyes országok dedikált hierarchiákat használnak, amelyek nem állnak kapcsolatban egymással. Az ábrán szereplő sárga téglalapok az egyes entitásokat jelentik, az általuk készített aláírásokat, illetve a kibocsátott tanúsítványokat nyilakkal jelöltük.**

kerülő adatokon a DS egységek helyeznek el PKCS#7 formátumú aláírást. A DS egységek biztonságos környezetben, és nem a nyilvános Interneten működnek, a kulcsaikat, tanúsítványaikat „gyakran" cserélik.

> ***13.1. Példa:*** *Tegyük fel, hogy egy német készülék egy magyar útlevélen ellenőrzi a digitális aláírást. A német készülékben eleve szerepel a magyar CSCA megbízható gyökértanúsítványa. Az útlevelet aláíró DS egység tanúsítványa az útlevélen lévő PKCS#7 aláírás-blokkban szerepel, így az ellenőrző készülék fel tudja építeni a tanúsítványláncot.*

A Magyar Köztársaság által kibocsátott útlevelek kiállítására használt rendszert – beleértve a CSCA gyökér hitelesítés-szolgáltatót és a DS egységeket – a Közigazgatási és Elektronikus Közszolgáltatások Központi Hivatala üzemelteti.

Tekintve, hogy ezen PKCS#7 aláírásokon nincsen időbélyeg, az útleveleken szereplő aláírás hitelessége csak addig állapítható meg, amíg az aláíró (DS egység) tanúsítványa érvényes. Ezért a DS egységek tanúsítványainak a kibocsátott útlevelek élettartamáig érvényesnek kell maradniuk, de a hozzájuk tartozó magánkulcsot csak rövid ideig használják, utána megsemmisítik. A tanúsítvány ezt követően is érvényes marad. (Bár a tanúsítvány érvényes, a magánkulcsát már megsemmisítették. Így az új útleveleket a DS egység már egy másik kulcs és másik tanúsítvány szerint írja.)

Ha valahol egy DS egység magánkulcsa kompromittálódik, az adott ország CSCA-ja visszavonási listát (CRL-t) bocsát ki. A CRL-eket az adott ország külön-külön juttatja el minden más országnak. A CSCA-k és DS egységek, valamint az offline működéséből adódóan ezek a CRL-ek nincsenek kint az Interneten, hanem közvetlen csatornákon terjednek. Ha egy DS egység tanúsítványát visszavonják, akkor – mivel az útlevélchipeken lévő PKCS#7

aláírásokon nincsen időbélyeg – az adott egység által kibocsátott összes útlevél érvénytelenné válik. E rendszerek mindegyike, beleértve a CSCA és a DS egységeket, biztonságos körülmények között működik, így nagyon kicsi az esély a kulcsok kompromittálódására.

Az eredeti tervek szerint az ICAO működtetett volna egy központi adatbázist a DS tanúsítványok és a CRL-ek elosztására. Ezen adatbázis végül nem jött létre.

13.4.2. Második generáció: Ujjlenyomat kiolvasása tanúsítvány-alapú autentikációt követően

Magyarország 2009. június 28. óta bocsát ki második generációs biometrikus útleveleket. Ezek annyiban nyújtanak többet az első generációs útleveleknél, hogy az útlevél birtokosának ujjlenyomatát is tartalmazzák. Az ujjlenyomat érzékeny személyes adat, így garantálni kell, hogy kizárólag arra jogosult olvasó berendezések olvashatják ki az útlevélből.

Az útlevél és az olvasó készülék között tanúsítvány alapú autentikációt követően biztonságos, azaz titkosított és hitelesített csatorna épül ki. Az útlevél birtokosának ujjlenyomatát kizárólag egy megfelelő tanúsítvánnyal rendelkező olvasó berendezés olvashatja ki, kizárólag e biztonságos csatornán keresztül.

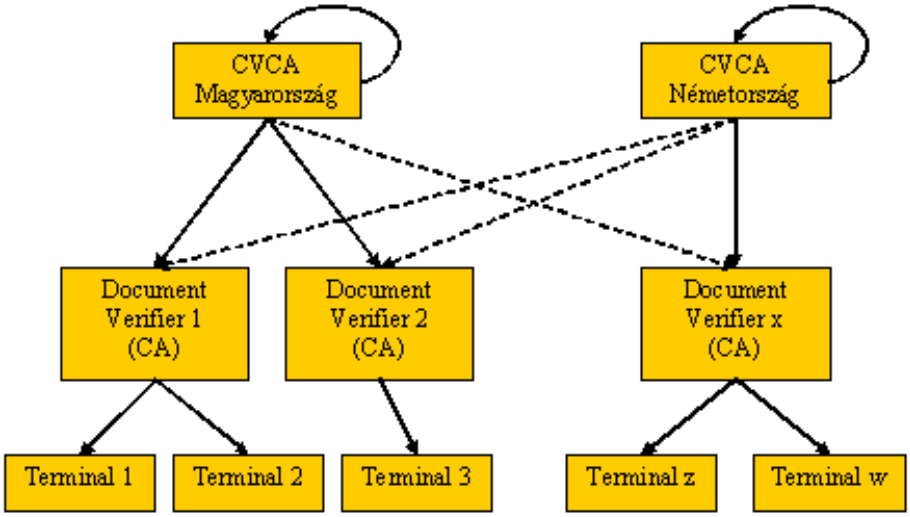
Az autentikáció és a biztonságos csatornák kiépítése PKI alapon történik, dedikált, nem nyilvános tanúsítvány-hierarchiák alapján. E célból minden útlevél-kibocsátó ország létrehozott egy-egy dedikált gyökér hitelesítés-szolgáltatót (CVCA, country verifying certification authority), és az adott ország által üzemeltetett vagy elfogadott olvasó berendezéseinek tanúsítványai ezen megbízható gyökértanúsítványokra vezethetőek vissza.

E CVCA-k biztonságos környezetben működnek, ők köztes szolgáltatói tanúsítványokat bocsátanak ki ún. DV (document verifier) hitelesítés-szolgáltatók számára. Az útlevél-ellenőrző berendezések (terminálok) tanúsítványait e DV köztes hitelesítő egységek bocsátják ki.

> ***13.2. Példa:*** *Tegyük fel, hogy X ország kibocsát egy DV CA tanúsítványt az okmányirodái, egy másik DV CA tanúsítványt pedig a határőrsége számára. Az okmányirodákban működő olvasó berendezéseknek az előbbi, a határőrségnél működő olvasó berendezéseknek az utóbbi bocsát ki eszköz-tanúsítványokat.*

Minden ország a saját CVCA tanúsítványát telepíti az általa kibocsátott útlevelekre, az útleveleken lévő chipek kizárólag ezt a CVCA tanúsítványt tekintik majd megbízható tanúsítványnak, csak a rá visszavezethető tanúsítvánnyal rendelkező olvasó berendezéseknek engedik kiolvasni az ujjlenyomatot.

Az országok kereszthitelesítik egymás DV CA-it, azaz újabb tanúsítványt bocsátanak ki a másik ország DV CA-i számára, így az országok egymás olvasó berendezéseit is elfogadják.



13.3. ábra. **Minden ország CVCA-ja tanúsítványt bocsát ki minden, általa elfogadott DV CA számára. Így minden DV CA több tanúsítvánnyal fog rendelkezni, mindig azt a tanúsítványát mutatja be, amely ország útlevelét éppen ellenőrzi.**

(Lásd: 13.3. ábra.) E kereszthitelesítéssel az egyes országok meghatározhatják, hogy más országok mely DV CA-i által felülhitelesített egységeknek mennyi időre, és milyen típusú hozzáférést adnak az ujjlenyomatokhoz. E kereszthitelesítések 2010. áprilisában még nem történtek meg, előkészítő tesztelések folynak.

> ***13.3. Példa:*** *Tegyük fel, hogy Y ország is kibocsát egy DV CA tanúsítványt az X ország okmányirodáinak berendezéseit felülhitelesítő DV CA számára, és szintén kibocsát egy másik DV CA tanúsítványt az X ország határőrségein működő berendezéseket felülhitelesítő DV CA számára. Így ezen berendezések (legalább) két tanúsítvánnyal fognak rendelkezni, az egyik az X ország, a másik az Y ország CVCA-jától fog származni.*

Ezen autentikációs tanúsítványokat az útleveleken lévő mikrochipen futó alkalmazások kell, hogy ellenőrizzék, így e területen nem a szokásos X.509 tanúsítványokat, hanem egyszerűbb felépítésű, kompaktabb, ún. CV (card verifiable) tanúsítványokat használnak. Az autentikációs tanúsítványokhoz szükséges hierarchiák nem RSA-t, hanem az elliptikus görbék elméletére épülő kriptográfiai algoritmusokat (ECC) használnak, mivel ECC segítségével sokkal kisebb kulcsmérettel el lehet érni a megfelelő szintű biztonságot. A használt ECC kulcsméret legalább 224 bit, és vagy az NIST, vagy az európai „Brainpool” munkacsoport által javasolt görbéket kell használni.

Az útlevél és az olvasó között felépülő biztonságos (titkosított és hitelesített) csatornát megvalósító protokoll nagyon hasonlít az SSL-re, de kialakításakor figyelembe vették,

hogy a kártyákon (és az útleveleken) lévő chipek teljesítménye messze elmarad az asztali számítógépeké mögött. E protokoll tömörebb, kevesebb opciót és egyeztetést tartalmaz, és például nem épít arra, hogy a felek Internet kapcsolattal vagy órával, időforrással rendelkeznének.

13.4. Példa: *Tegyük fel, hogy a német autópálya-rendőrség olvasó-berendezése hozzá szeretne férni egy magyar útlevélben lévő ujjlenyomathoz. Érzékeli, hogy az útlevél magyar, így az autentikációs tanúsítványához kapcsolódó „magyar" tanúsítványláncot mutatja be az útlevélnek. E lánc a következő elemekből áll: (1) a német autópálya-rendőrség DV CA-ja által kibocsátott végtanúsítvány, (2) a magyar CVCA által a német autópálya-rendőrség DV CA-ja számára kibocsátott köztes szolgáltatói tanúsítvány, és a lánc a magyar CVCA gyökértanúsítványában végződik, amely eleve telepítve van a magyar útlevélre. E láncot a magyar útlevél felismeri, és az e lánc szerinti autentikációt követően kiépülő biztonságos csatornán keresztül engedélyezi a hozzáférést a kártyabirtokos ujjlenyomatához.*

14. fejezet

Összefoglalás

„Aber das ist eine andere Geschichte und soll ein andermal erzählt werden."
(De ez már más történet, és elbeszélésére más alkalommal kerül majd sor.)

– Michael Ende, A végtelen történet

A „Web 2.0" korszakát éljük. Egyre több és több ügyünket intézhetjük különféle webes portálokon, egyre több szolgáltatást érhetünk el az Interneten keresztül. Az igazán fontos, érdemi műveletekhez viszont még mindig papírra van szükség, mert általában még a papír alapú, kézzel írott aláírás biztosítja, hogy a művelet „hiteles" legyen. Várható, hogy a közeljövőben az érzékeny műveleteket is egyre inkább elektronizálni fogják, és ekkor kulcskérdéssé válik, hogy az elektronikus világban mi és hogyan biztosítja a hitelességet.

Csak látszólag könnyebb az az út, ha egyes portálokról egyszer csak kimondjuk, hogy azok hitelesek, és mostantól érzékeny ügyeink kezelésére is alkalmasak. Az így kapott megoldás kényelmes ugyan, de ha az információ pusztán attól hiteles, hogy a portálon van, akkor teljesen kiszolgáltatottá válunk portállal, annak üzemeltetőivel és a náluk dolgozó rendszergazdákkal szemben.

A könyvünkben bemutatott elektronikus aláírás, illetve PKI technológia jelenti azt a megoldást, amely szerint elektronikusan is hozhatunk létre írásba foglalt, bizonyító erővel rendelkező, hiteles okiratokat. Elektronikus aláírás használata esetén nem az biztosítja az információ hitelességét, hogy az hol, melyik rendszerben található, hanem – a papír alapon már megszokott és bevált megoldáshoz hasonlóan – az információt tartalmazó okiraton elhelyezett aláírás. Az elektronikus aláírással ellátott okirat attól függetlenül hiteles, hogy az hol van, kinél található, így nem mindig a rendszernek van igaza, hanem a kisembernek is lehet bizonyítéka a „Nagy Testvér" ellen.

Bemutattuk, hogy az elektronikus aláírás működéséhez szükséges nyilvános kulcsú infrastruktúra (PKI) szereplői úgy nevezett kriptográfiai kulcsok és kriptográfiai algoritmusok

segítségével biztosíthatják az információ titkosságát, illetve hitelességét. A hitelesítés-szolgáltatók által kibocsátott tanúsítványokra támaszkodva megállapíthatják, illetve később bizonyíthatják, hogy egy kriptográfiai kulcs kihez tartozik. A tanúsítvány alapján küldhetünk a birtokosának titkosított üzenetet, illetve a tanúsítványa alapján ellenőrizhetjük valakinek az elektronikus aláírását. Az aláírások „letagadhatatlanságának" biztosításához megbízható időpontokra, időbélyegekre van szükség. Leírtuk, hogyan használható az elektronikus aláírás, bemutattunk egy aláírások létrehozására és ellenőrzésére szolgáló szoftvert, és áttekintettünk néhány már létező, elektronikus aláírásokra épülő, a valóságban is működő rendszert.

Jósolni nehéz, különösen, ha a jövőről van szó. Lehet, hogy egészen más technológiák fognak elterjedni az elektronikus hitelesség biztosítására, de lehet, hogy az itt bemutatott elektronikus aláírás lesz az a technológia, amellyel a jövőben elektronikus okiratainkat hitelesíteni fogjuk. Az idő el fogja dönteni.

Függelék

A függelék

Az e-Szignó programcsalád

„Nem irom pennával,
Fekete téntával,
De szablyám élivel,
Ellenség vérivel,
Az én örök hiremet.”

– Zrínyi Miklós verse

A.1. Az e-Szignó bemutatása

Az e-Szignó egy tanúsított[1], professzionális elektronikus aláírás-létrehozó és -ellenőrző alkalmazás, amelyet a Microsec Kft. fejlesztett ki. Az úgy nevezett e-akta formátum eredetileg az e-Szignó natív formátuma volt, de mára más aláírás-létrehozó és aláírás-ellenőrző alkalmazások is kezelik, és Magyarországon de facto szabvánnyá vált.

A továbbiakban az e-Szignó programon keresztül mutatjuk be egy aláírás-létrehozó alkalmazás működését, hiszen ma az e-Szignó talán a legelterjedtebb hazai aláírás-létrehozó és -ellenőrző alkalmazás, illetve bizonyos változatai ingyenesen is elérhetőek, így a könyvünkben leírtak többségét bárki ingyenesen kipróbálhatja a `www.e-szigno.hu` oldalról igényelhető teszt tanúsítványok segítségével. Jellemzően más aláírás-létrehozó és -ellenőrző alkalmazások is hasonló funkciókkal rendelkeznek.

Többféle formában találkozhatunk az e-Szignóval:

[1] Az e-Szignó magját képező XadesSigner modul bevizsgált, tanúsított aláírás-létrehozó szoftver. A szoftver tanúsítását a MATRIX Vizsgáló, Ellenőrző és Tanúsító Kft, a Nemzeti Hírközlési Hatóság nyilvántartásában szereplő, elektronikus aláírás termékek tanúsítását végző szervezet végezte.

- Az *e-Szignó grafikus kliens* változata segítségével felhasználói felületen keresztül hozhatóak létre aláírások és e-akták. A grafikus kliens Windows és Linux alatt működik, ingyenesen letölthető a `www.e-szigno.hu` oldalról, és egyes funkciói – köztük az aláírás-ellenőrzés, és a Microsec által kibocsátott éles és teszt tanúsítványok szerinti aláírások létrehozása – ingyenesen használhatóak.

- Az *e-Szignó Automata* olyan szoftvercsomag, amely könnyen integrálható informatikai rendszerekbe, ahol aláírások nagy tömegű, automatizált feldolgozását teszi lehetővé. Az e-Szignó Automata használható parancssoros alkalmazásként vagy programkönyvtárként, segítségével tágabb funkcionalitás érhető el, mint a grafikus kliens segítségével.

- A *Webes e-Szignó* (`www.e-szigno.hu/webszigno`) böngészőprogramok segítségével használható, anélkül, hogy a kliens számítógépre szoftvert installálnánk. A böngésző ilyenkor egy mindössze néhány kilobyte-os ActiveX controlt vagy Java appletet tölt le, amely a kliensen vagy klienshez kapcsolt eszközökön (pl. intelligens kártyán) lévő magánkulcsokat kezeli. Az e-Szignó funkcionalitást a webszerver és a rajta futó e-Szignó Automata biztosítja. (Lásd: A.3.6. fejezet.)

A.2. A XadesSigner mag

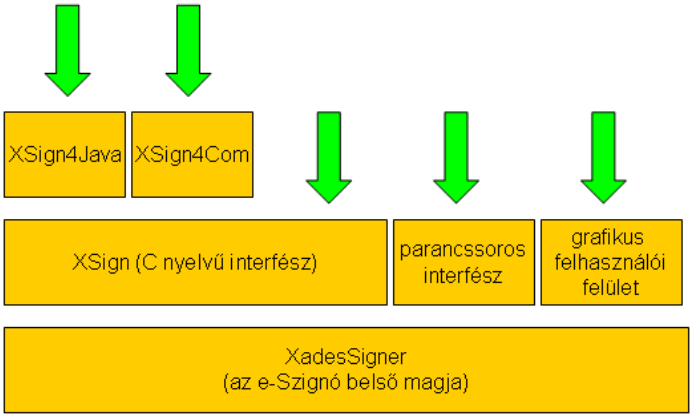
A.2.1. A XadesSigner mag és interfészei

Az e-Szignó magja a XadesSigner modul, ez valósítja meg az e-Szignó funkcióit. A modul funkcionalitása különböző interfészeken keresztül érhető el. A grafikus kliens e-Szignó és a parancssoros változat közvetlenül a XadesSigner modulra épülnek. Szintén ezt a modult használja az e-Szignó XSign nevű, standard C interfésze is. Az XSign-ra épülnek az e-Szignó magas szintű, Java és COM interfészei is. Az e-Szignót integráló programok közvetlenül nem a XadesSigner modult érik el, hanem vagy a parancssoros alkalmazást használják, vagy az XSign, XSign4Java és XSign4Com interfészek egyikét használják.

A.2.2. e-Szignó az egyes platformokon

Az e-Szignó sokféle különböző platformon elérhető. Windows alatt működhet *windowsos üzemmód*ban vagy *nem windowsos üzemmódban.*

Windowsos üzemmód esetén az e-Szignó támaszkodhat a Windows tanúsítványtárára, és használhatja a Windows alatt elérhető CryptoAPI-t. A CryptoAPI-n keresztül el tudja érni mindazon kriptográfiai hardver eszközöket, például intelligens kártyákat, HSM-eket, amelyek támogatják a CryptoAPI-t. A CryptoAPI jóvoltából bizonyos grafikus felülettel is rendelkezik, például meg tudja kérdezni a felhasználót, hogy melyik tanúsítvány szerint szeretne aláírni.



A.1. ábra. **Az e-Szignó interfészei**

Windowsos üzemmód esetén az e-Szignó az Internet Explorer segítségével kezeli az Internetet, így használja az Explorer beállításait (pl. proxy, megjegyzett jelszavak stb). A Windowos üzemmód csak Windows alatt érhető el.

Nem windowsos üzemmódban az e-Szignó csak a paraméterként megadott tanúsítványokat használja, és a hálózati kapcsolatokhoz szükséges adatokat (pl. jelszavakat) meg kell adni neki. A magánkulcsokat ekkor vagy PKCS#11 könyvtárakon keresztül éri el, vagy szoftveres tanúsítványok PFX fájlban lévő magánkulcsait tudja használni. Linux, Solaris, AIX és WinCE alatt csak a nem windowsos üzemmód érhető el, de Windows alatt is tud nem windowsos üzemmódban működni az e-Szignó.

A XadesSigner a következő bemenetek alapján dolgozik:

- Meg kell adni, hogy mely megbízható gyökértanúsítványok (`trusted_certs`) alapján ellenőrizze az aláírásokat. Ha a Windows tanúsítványtárát használjuk, akkor az e-Szignó a Windowsban szereplő megbízható gyökértanúsítványokat fogadja el. Ha nem a Windows tanúsítványtárát használjuk, akkor beállítható, hogy egy megbízható gyökértanúsítványt ne fogadjon el minden célra, hanem például csak aláírásokat, csak időbélyegeket vagy csak OCSP válaszokat vezessen vissza rájuk az e-Szignó.

- Meg kell adni, hogy mely köztes tanúsítványokat használjon az e-Szignó; windowsos üzemmódban a Windows tanúsítványtárában szereplő köztes tanúsítványokat, egyébként a megadott könyvtárban lévő köztes tanúsítványokat használja. A köztes tanúsítványok tárában megadott tanúsítványokon kívül az e-Szignó le tudja tölteni a tanúsítványok AIA mezejében hivatkozott szolgáltatói tanúsítványokat is, ezeket is felhasználhatja a tanúsítványlánc felépítéséhez.

- A magánkulcsot használó műveletekhez (pl. aláírás, dekódolás) meg kell adni, hogy mely tanúsítvány magánkulcsát szeretnénk használni. Windowsos üzemmódban az

e-Szignó fel tudja ajánlani a felhasználónak, hogy válasszon a Windowsba „saját” tanúsítványként telepített tanúsítványok közül. Nem windowsos üzemmód esetén a megadott magánkulcsot használja az e-Szignó.

Tanúsítványlánc felépítésekor az e-Szignó az ellenőrizni kívánt tanúsítványból indul ki, és a köztes szolgáltatói tanúsítványok alapján keres tanúsítványláncot egy önhitelesített tanúsítványig. Ha talál ilyen láncot, megvizsgálja, hogy a talált önhitelesített tanúsítvány megbízható gyökértanúsítvány-e. A lánc keresésekor használja a köztes szolgáltatói tanúsítványok tárát, a megbízható gyökértanúsítványok tárát, és az Internetről letölti az AIA mezőben meghivatkozott tanúsítványokat. Ha talál olyan láncot, amely az adott célra megfelelő megbízható gyökértanúsítványban végződik, akkor megáll, egyébként tovább próbálkozik, és új láncot keres. Ha egy adott ideig nem talál megfelelő láncot, elutasítja az aláírást.

1.1. Példa: *Az e-Szignó Alajos tanúsítványát ellenőrzi, ennek során felépíti a tanúsítványláncot Alajos tanúsítványától egy megbízható gyökérig. Az Alajos tanúsítványát kibocsátó szolgáltató tanúsítványa nem szerepel sem a megbízható gyökerek, sem a köztes szolgáltatói tanúsítványok között. Az e-Szignó megtalálja Alajos tanúsítványában a kibocsátó tanúsítványának URL-jét, ez alapján letölti a kibocsátó hitelesítés-szolgáltató X tanúsítványát, és köztes szolgáltatói tanúsítványként kezeli. (Ha ez egy önhitelesített tanúsítvány, akkor – mivel nem megbízható gyökértanúsítvány – az e-Szignó elutasítja az aláírást.)*

Az e-Szignó tovább építi a tanúsítványláncot az Alajos tanúsítványát kibocsátó szolgáltató X tanúsítványától, talál a megbízható gyökértanúsítványok között egy olyan Y tanúsítványt, amely tekinthető az X tanúsítványt kibocsátó szolgáltató tanúsítványának. Az e-Szignó talált egy tanúsítványláncot, az aláírást ez alapján fogja ellenőrizni.

Az e-Szignó mindig valamilyen beállítások szerint ellenőrzi az aláírást, ilyen beállítás lehet például, hogy ellenőrizze-e a visszavonási állapotot, érvényesítsen-e kivárási időt stb. A beállítások megadhatóak paraméterként, de egy részük XML aláírási szabályzatként is. Alapesetben minden opció ki van kapcsolva, külön be kell kapcsolni őket.

Aláírás ellenőrzése során az e-Szignó *bizonyítékok*at gyűjt az aláírás érvényességére. Az aláírást *érvényes*nek tekinti, ha az aláírás érvényességét le tudja vezetni a rendelkezésre álló bizonyítékok alapján, a használt beállítások szerint. Ha nem tudja levezetni, de későbbi időpontban előállhatnak még szükséges információk (pl. CRL-ek), akkor az ellenőrzés *befejezetlen*, ha a szükséges információk később sem állhatnak elő, akkor *érvénytelen*nek tekinti az aláírást. Ha talál olyan bizonyítékot, amely értelmében az aláírás érvénytelen, akkor is *érvénytelen*nek tekinti az aláírást.

A bizonyítékok elektronikusan aláírt PKI objektumok. Az e-Szignó ellenőrzi a rajtuk lévő aláírást is a megadott beállítások szerint, ehhez további bizonyítékokat kell figyelembe vennie, és az ezeken lévő aláírásokat is ellenőriznie kell stb. E bizonyítékok köztes szolgáltatói tanúsítványok, visszavonási listák, OCSP válaszok, és időbélyegek lehetnek.

Amennyiben az adott aláíráson van érvényes időbélyeg, úgy az időbélyegen szereplő időpontra nézve ellenőrzi az aláírást. Ehhez az időbélyeg érvényességét, azaz az időbélyegen lévő aláírást is ellenőrizni kell. Ha az adott aláíráson nincsen érvényes időbélyeg, akkor az aktuális időpontra (*most*) nézve ellenőrzi, az aktuális időpontot a számítógép órája alapján határozza meg.

Bekapcsolhatjuk, hogy az e-Szignó kivárási időt érvényesítsen. Ekkor megköveteli, hogy a visszavonási információ legyen frissebb, mint az időpont, amire ellenőrzünk, azaz a visszavonási információban szereplő `thisUpdate` időpont legyen későbbi, mint az az időpont, amelyre nézve az aláírást ellenőrizzük (`controlTime`). Kivárási idő bekapcsolása esetén az e-Szignó kivárási időt érvényesít a végfelhasználói aláírásokra, a szolgáltatói tanúsítványokon lévő aláírásokra, az időbélyegeken lévő aláírásokra, az OCSP válaszokon lévő aláírásokra, de CRL-eken lévő aláírásokra nem érvényesít kivárási időt. (Nem vizsgálja, hogy a CRL kibocsátójára vonatkozó visszavonási információ későbbi-e, mint maga a CRL.)

> ***1.2. Példa:*** *Adott egy XAdES-C aláírás. A végfelhasználó aláírását és a tanúsítványát időbélyeg védi, és az aláíráshoz csatolták a végfelhasználói tanúsítványra vonatkozó CRL-t. A végfelhasználó tanúsítványát egy köztes szolgáltatói egység bocsátotta ki, amelynek egy megbízható gyökér bocsátott ki tanúsítványt. Sem e szolgáltatói tanúsítványokat, sem a gyökér által kibocsátott CRL-t nem védi az időbélyeg. Ezt az aláírást szeretnénk kivárási idővel ellenőrizni.*
>
> *Az e-Szignó az időbélyegben szereplő időpontra nézve ellenőrzi a végfelhasználó aláírását és a végfelhasználó tanúsítványán lévő aláírást is. Megköveteli, hogy a csatolt CRL későbbi legyen, mint az időbélyegben szereplő időpont.*

Az e-Szignó teljes funkcionalitása csak regisztrált változat esetén használható ki. Ugyanakkor az e-Szignó regisztrálatlanul is működik, de ekkor csak korlátozott funkcionalitás érhető el. E funkcionalitás eltér a kliens és az automata e-Szignó esetén:

- Az e-Szignó grafikus kliens regisztrálatlanul is használható „éles” célra. Regisztrálatlanul csak a Microsec Kft. által kibocsátott tanúsítványok szerint lehet vele aláírást létrehozni, de az így készült, éles tanúsítványok szerint készült aláírások teljesen egyenértékűek a regisztrált e-Szignó által létrehozott aláírásokkal. A regisztrálatlan kliens segítségével bármilyen[2] tanúsítvány, bármilyen aláírás ellenőrizhető, függetlenül a tanúsítvány kibocsátójától. A regisztrálatlan grafikus kliens reklámokat jelenít meg.

[2]Bármilyen, a vonatkozó szabványoknak megfelelő tanúsítvány, feltéve, hogy a szükséges gyökereket telepítettük a megbízható gyökerek közé.

- Az e-Szignó Automata regisztráció nélkül csak tesztelésre használható, éles célra nem. Ekkor csak a Microsec Kft. által kibocsátott teszt tanúsítványok szerint lehet vele aláírást készíteni, és a beillesztett dokumentumokat „teszt” prefix-szel látja el.

Az e-Szignó regisztrációja mindig egy géphez kötődik, a gép paramétereit tartalmazza. A regisztrációs fájl korlátozhatja, hogy milyen funkciók érhetőek el az e-Szignóban.

A.3. Az e-Szignó funkcióinak bemutatása

Az e-Szignó funkciói az alábbi csoportokra oszthatók:

- Dokumentumok kezelése az e-aktában
- Aláírások és időbélyegek elhelyezése
- Aláírások ellenőrzése, listázása
- Titkosítás
- Egyéb funkciók

Minden funkció használatához meg kell adni bizonyos alapvető paramétereket:

- Minden e-Szignó művelet esetén meg kell adni a munkakönyvtárat (`work_dir`), a regisztrációs fájlt (`reg_file`), a naplózás mértékét (`silent`, `verbose`, `debug`).
- Aláírási műveletek esetén meg kell adni, hogy milyen kulccsal írunk alá, és meg kell adni a kulcsot védő PIN kódot vagy jelszót is.
- Aláírás ellenőrzésekor meg kell adni a megbízható tanúsítványok és a köztes szolgáltatói tanúsítványok körét (Windows tanúsítványtár vagy egy könyvtár).
- Időbélyegzés (vagy időbélyeges aláírás) és OCSP-alapú aláírás-ellenőrzés esetén meg kell adni a szolgáltatás elérhetőségét és a szolgáltatás eléréséhez szükséges tanúsítványt és magánkulcsot vagy felhasználónevet és jelszót.
- Titkosításkor a titkosító tanúsítványokat, visszafejtéskor a magánkulcsot és az őt védő PIN kódot vagy jelszót.

A.3.1. Dokumentumok kezelése az e-aktában

- Dokumentum beillesztése az e-aktába (`insert_doc`) – E funkcióval egy új dokumentumot adhatunk hozzá egy e-aktához, illetve új e-aktát is ezzel a funkcióval

hozhatunk létre. Ha egy e-aktán már van keretaláírás, akkor már nem helyezhetünk el benne további dokumentumokat. Beillesztéskor célszerű megadni a dokumentum MIME típusát is.

- Dokumentum kimásolása az e-aktából (egy dokumentum esetén `export_doc`, az összes dokumentum esetén `export_docs`) – E művelettel egy vagy több dokumentumot kimásolhatunk az e-aktából, és elmenthetjük őket a fájlrendszerben. Ha titkosított e-aktáról van szó, akkor e művelettel lehet kititkosítani is.

- Dokumentum törlése az aktából (`delete_doc`)

- Másik e-akta tartalmának beemelése (`merge_dossier`) – E művelettel nem egy másik e-aktát, hanem annak tartalmát emelhetjük bele az aktuális e-aktába. A másik e-aktában lévő dokumentumok és a dokumentumokon lévő aláírások (a keretaláírások nem) belekerülnek az aktuális aktába. Ha az A és a B e-aktában is egy-egy dokumentum, és rajta egy-egy aláírás található, akkor ha a B aktába beemeljük az A akta tartalmát, akkor a B aktában két dokumentum lesz, rajtuk egy-egy aláírással.

- Dokumentumok/akták átnevezése

A.3.2. Aláírások és időbélyegek elhelyezése

- Dokumentum aláírása (`sign_doc`) – Aláírás elhelyezése a dokumentumon. Meg lehet adni, hogy milyen XAdES típusú aláírást szeretnénk létrehozni. A magasabb XAdES típusokhoz (-T-től fölfelé) meg kell adni az időbélyeg szolgáltatás elérhetőségét; ha OCSP válaszokat is csatolni szeretnénk az aláíráshoz (pl. -A), akkor meg kell adni az OCSP elérhetőségét is (kiéve, ha az a tanúsítvány AIA mezejéből kiolvasható helyen, autentikáció nélkül érhető el).

- Keretaláírás az aktán (`sign_dossier`) – A keretaláírás az e-aktában lévő összes dokumentumra és a rajtuk lévő összes (nem keret) aláírásra és időbélyegre vonatkozik.

- Ellenjegyzés, keretaláírás ellenjegyzése (`countersign_doc`, `countersign_dossier`) – E művelettel más aláírásokon vagy keretaláírásokon helyezhetünk el ellenjegyző aláírást. Az ellenjegyzés csak a kérdéses kriptográfiai aláírást védi, azaz ezen aláírások kiterjeszthetőek, archiválhatóak maradnak.

- Dokumentum időbélyegzése, illetve keretidőbélyegzés a teljes aktára.

- Nem e-akták esetén használható funkciók:

  - Különálló aláírás készítése (`sign_files`) – Az aláírás ekkor egy külön fájlba kerül.

  - XML aláírás készítése (`sign_xml`) – Ekkor egy XML fájl egyes elemeit írhatjuk alá, és az aláírást ugyanezen XML fájlban helyezhetjük el.
  - PDF aláírás készítése (`pdf_sign`).
  - CMS aláírások kezelése (`cms_sign`/`cms_verify`).

A.3.3. Aláírások ellenőrzése, listázása

- Aláírás ellenőrzése (`validate_sig`) – A megadott aláírást ellenőrizhetjük vele. Meg kell adni, hogy milyen módon ellenőrizzen (CRL, OCSP, OCSP/CRL, kivárási idő stb). E funkcióval egyúttal XAdES típust is lehet váltani, ehhez meg kell adni, hogy milyen XAdES típusra szeretnénk váltani, és szükségesek lehetnek hozzá időbélyeg (pl. -BES $\rightarrow -T$) vagy OCSP (pl. -T $\rightarrow$ -A, OCSP alapon) elérési adatok is.

- Csak tanúsítvány ellenőrzése (`validate_cert_file`) – A megadott fájlban lévő tanúsítványt ellenőrizhetjük. Ekkor csak tanúsítványlánc-építés és visszavonási állapot ellenőrzés történik, az aktuális időpontra nézve.

- Az e-akta tartalmának listázása (`list_dossier`) – XML vagy TXT listát kaphatunk az e-akta tartalmáról. E funkció egyúttal ellenőrizni is tudja az aláírásokat, ekkor meg kell adni az ellenőrzéshez szükséges információkat (visszavonási állapot ellenőrzésének módja, kivárási idő stb).

- A teljes e-akta ellenőrzése (`validate_dossier`) – Az e-aktában lévő összes aláírást ellenőrzi. E művelet rekurzívan is tud ellenőrzini (ha az aktában további akták vannak, azokat is ellenőrzi), és egyúttal ki is tudja terjeszteni az aláírásokat magasabb XAdES típussá; de csak azokat az aláírásokat tudja kiterjeszteni, amelyeket nem véd másik aláírás vagy időbélyeg. E művelet úgy is felhívható, hogy hibát jelezzen, ha aláíratlan dokumentum van az e-aktában, illetve egy XML struktúrát is vissza tud adni az e-akta szerkezetéről.

A.3.4. Titkosítás

- Titkosítás (`encrypt_doc`, `encrypt_dossier`) – Titkosításkor új e-akta jön létre, amelybe a titkosított adatok S/MIME titkosított e-aktaként kerülnek bele. [146] A titkosításhoz meg kell adni a titkosító tanúsítványokat is.

- Visszafejtés (`export_doc`) – A titkosított információk visszafejtéséhez meg kell hivatkozni a visszafejtéshez használni kívánt magánkulcsot is.

A.3.5. Egyéb funkciók

- Átvételi elismervények készítése, ellenőrzése.
- Webes aláíráshoz (A.3.6. fejezet) szükséges műveletek, például az aláírandó hash kinyerése vagy az aláírt hash beillesztése.
- Az e-akta felépítését lekérdező műveletek, például információk az aláírásokról, aláírói tanúsítványokról,
- Viszontazonosítás.

A.3.6. Webes aláírás

A webes aláírás azt jelenti, hogy az aláírást egy böngészőprogramon keresztül készítjük el, és például egy webes űrlapon megadott információkat írunk alá. A böngészőprogram ekkor egy kis méretű ActiveX controlt vagy Java appletet tölt le, és a webszerveren lévő e-Szignó Automatára támaszkodik.

Webes aláíráskor a következő folyamat zajlik le:

1. A felhasználó egy webes űrlapon megadja az aláírandó adatokat, megtekinti és jóváhagyja az aláírásra kerülő információkat. Az ActiveX control vagy Java applet segítségével kiválasztja az aláíráshoz használandó tanúsítványt, és elküldi a szervernek.
2. A szerver az adatok és a tanúsítvány alapján összeállítja az aláírandó e-aktát, és benne az aláírás-formátum blokkot (amelyből még hiányzik a kriptográfiai értelemben vett aláírás), ebből lenyomatot képez, és ezt a hash-t visszaküldi a böngészőnek.
3. Az ActiveX control vagy Java applet aláíratja a hash-t a felhasználóval (ehhez szükség esetén a felhasználó kártyájához fordul), majd az így kapott, kriptográfiai értelemben vett aláírást elküldi a szervernek.
4. A szerver elhelyezi a kriptográfiai értelemben vett aláírást az aláírás-formátum blokkba, és létrejön az aláírt e-akta. A szerver eltárolja az aláírt e-aktát, és egyúttal visszaküldi a felhasználónak, hogy a felhasználó is megkapja, hogy mit írt alá.

Webes aláírás esetén a felhasználó magánkulcsa végig a felhasználó számítógépén van, ugyanakkor felhasználónak nincsen szüksége e-Szignóra. Elegendő, ha a böngészőjébe letöltődik az ActiveX vagy a Java applet (néhány száz kilobyte).

Megjegyezzük, webes aláírás esetén egy rosszindulatú szerver be tudja csapni a felhasználót, és bármit aláíratha vele. Igaz, egy rosszindulatú alkalmazás (vagy a bele integrált e-Szignó) amúgy is bármit aláíratha a felhasználóval. *A webes aláírás technológia akkor használható, ha a felhasználó megbízik a webszerverben.*

A.4. Támogatott aláírás-formátumok

Az e-Szignó a következő aláírás-, illetve konténer formátumokat támogatja:

- e-akta,
- XML aláírások (XMLDSIG és XAdES),
- ASN.1 alapú aláírások (PKCS#7, CMS és CAdES),
- PDF aláírás (PDF, benne CAdES).

A.5. Hogyan célszerű az e-Szignót integrálni?

- Egyik lehetőség, hogy a *grafikus kliens e-Szignó*t használjuk, ekkor nincs szükség integrációra. Ha csak aláírás-ellenőrzésről van szó, akkor a regisztrálatlan változat is elegendő lehet.
- Másik lehetőség, hogy egy *központi e-Szignó szervert állítunk fel*. Ez esetben a szerveren e-Szignó automatát célszerű futtatni, és e szerver lát el minden aláírással kapcsolatos funkciót. Így például egy nagy rendszerben egy ponton, egységesen lehet aláírásokat ellenőrizni vagy készíteni. Ez az automata egy (vagy kevés) tanúsítvánnyal rendelkezik, és általában szoftveres magánkulccsal vagy HSM-mel ír alá. Ezt az utat szokás követni például egy elektronikus számlákat kiállító rendszerben, vagy ha például egy archívumban szeretnénk aláírt dokumentumokat elhelyezni.
- Harmadik lehetőség, hogy az egyes kliensgépeken futó szoftverekbe integrálunk e-Szignó Automatát. Ekkor a kliensgépeken futó szoftvereket használó ügyintézők önálló kártyákkal/tanúsítványokal rendelkeznek, a meglévő szoftvereiket használják, és esetleg nem is látják, hogy e-Szignóval dolgoznak. A szükséges műveleteket az e-Szignó Automata a háttérben végzi el.
- Negyedik lehetőség, hogy a kliensek a böngésző-programukban írnak alá, webes aláírással. Ekkor a kliensek önálló kártyákkal/tanúsítványokkal rendelkezhetnek, és például egy központi webszerveren lévő e-Szignó Automatát használhatnak.

A.6. Alapműveletek parancssoros e-Szignóval

A következőkben néhány alapvető műveletet mutatunk be az e-Szignó program parancssoros változatával Linux operációs rendszeren. Az alábbi parancsok e-aktákat használnak.

Az e-Szignó programot a `/usr/local/eszigno3/` könyvtárba telepítettük, és beállítottuk az e-Szignó működéséhez szükséges paramétereket tartalmazó környezeti változókat. A path és

a megosztott könyvtárak keresési útvonalának beállításán kívül megadjuk, hogy az e-Szignó hol keresse a regisztrációs fájlt, melyik könyvtárba írja az ideiglenes műveletek eredményeit, és ne írjon ki felesleges technikai információkat.

```
export PATH=$PATH:/usr/local/eszigno3/distbin/
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/eszigno3/distbin/
export ES3_REG_FILE=/usr/local/eszigno3/server_reg.xml
export ES3_WORK_DIR=work/
export ES3_SILENT=yes
```

Az aláírások ellenőrzésére vonatkozó – aláírási szabályzat által meghatározott – szabályokat egy e-Szignó konfigurációs fájlban (`policy.conf`) határoztuk meg. Ezeket célszerű egy helyen megadni, így minden egyes aláírás-létrehozáskor vagy aláírás-ellenőrzéskor nem szükséges külön megadni őket.

```
[eszigno3_global_options]
# Az ebben a könyvtárban lévő tanúsítványokat fogadjuk el megbízható gyökérnek.
trusted_cert_dir = /usr/local/eszigno3/trusted_certs/

# Tanúsítványláncok felépítésekor az itt szereplő köztes tanúsítványokat
# is felhasználjuk (az aláíráshoz csatolt és a tanúsítványokban
# meghivatkozott köztes tanúsítványok mellett).
intermediate_cert_dir = /usr/local/eszigno3/intermediate_certs/

# Először OCSP-vel, és ha ez nem sikerül, CRL-lel vizsgáljuk
# a visszavonási állapotot a teljes láncra.
rev_check_mode = ocsp/crl

# Mindkét esetben kivárási időt alkalmazunk, azaz megköveteljük, hogy
# a visszavonási információ frissebb legyen, mint az az időpont,
# amelyre nézve az aláírást ellenőrizzük. A kivárási időt a teljes
# láncra érvényesítjük.
wait_for_grace_peridod = yes

# Időbélyeggel ellátott aláírásokat hozunk létre, valamint
# megköveteljük, hogy minden aláíráson legyen időbélyeg;
# ha egy aláíráson nincsen időbélyeg, mi helyezünk el egyet.
xades_type = t
```

```
# Ha valahova időbélyeget kell tennünk, azt ebből a forrásból szerezzük be,
# és ilyen módon authentikáljuk magunkat.
timestamp_url_list = "https://btsa.e-szigno.hu/tsa2"
http_auth_list = isti:kurtykurutty@btsa.e-szigno.hu

# Az aláírások és az időbélyegek létrehozásához az SHA-256 lenyomatképző
# algoritmust használjuk. A meglévő aláírások esetén más algoritmust is
# elfogadunk.
digest_algorithm = sha256
```

Az `only_qualified_signatures` paraméterrel adhatnánk meg ha csak minősített aláírásokat akarunk elfogadni, valamint az `accepted_cp_list` paraméterrel beállíthatjuk, hogy csak megadott hitelesítési rendeket fogadunk el.

A következőkben bemutatott példa parancsok a fenti beállításokat veszik alapul, azoknak megfelelően működnek.

- Hozzunk létre egy üres e-aktát!

  ```
  eszigno3 insert_doc -new_dossier yes -out aktam.es3
  ```

- Tegyünk bele egy dokumentumot az e-aktába!

  ```
  eszigno3 insert_doc -in aktam.es3 -doc dokumentum.txt -out aktam.es3
  ```

- Helyezzünk el keretaláírást az e-aktán a fenti szabályok (azaz a `policy.conf` fájl) szerint! Példánkban szoftveres kulcsot használunk, és megadjuk a fájlhoz tartozó jelszót.

  ```
  eszigno3 sign_dossier -conf policy.conf -in aktam.es3 \
  -signer_key pfx/isti.pfx -signer_pass 1234 -out aktam.es3
  ```

- Vizsgáljuk meg az akta tartalmát!

  ```
  eszigno3 list_dossier -in aktam.es3
  ```

  A kimenet első sora az akta nevét és azonosítóját tartalmazza. A második sor azt jelenti, hogy az aktán adott személy keretaláírást helyezett el, illetve tartalmazza az aláírás azonosítóját. (Az aláírás azonosítójával hivatkozhatunk az aláírásra, például ha kifejezetten ezt az aláírást szeretnénk ellenőrizni, vagy kiterjeszteni a `validate_sig` paranccsal.) A parancs kilistázza az aktában lévő dokumentumokat és azok azonosítóját is. (A dokumentum azonosítóját akkor kell használni, ha az adott dokumentumot szeretnénk aláírni a `sign_document` paranccsal, vagy ki szeretnénk írni a fájlrendszerbe az `export_doc` paranccsal.) A kimenet:

```
aktam.es3 Object0
Dr. Berta István Zsolt S78bb79d2-1dd2-11b2-bb71-e9d6abde21ef

dokumentum.txt O02876eba-1dd2-11b2-bb71-e9d6abde21ef
```

- Ellenőrizzük az e-aktán lévő aláírásokat! A `validate_dossier` parancs megvizsgálja, hogy az e-akta nem üres-e, illetve ellenőrizi az e-aktában lévő összes aláírást. Ha mindent rendben talált, az e-Szignó visszatérési értéke 0, különben az e-Szignó dokumentációja szerinti értékkel tér vissza. A `list_out` paraméterrel XML formátumú lista is kinyerhető az aláírás ellenőrzésének menetéről, és az esetleges hibákról. [37]

  ```
  eszigno3 validate_dossier -conf policy.conf -in aktam.es3 -out aktam.es3
  ```

  A kimenetként kapott e-akta az ellenőrzött, így esetleg kiterjesztett aláírásokat tartalmazza. Példánkban megőrizzük a kiterjesztés során összgyűjtött információkat, de sok esetben el is dobhatjuk őket.

- Másoljuk ki az e-aktában lévő fájlokat a fájlrendszerbe!

  ```
  mkdir ide/
  eszigno3 export_docs -in aktam.es3 -outdir ide/
  ```

Ezen műveletek az e-Szignó más interfészein is hasonló módon érhetőek el.

A.7. Összegzés

- Az e-Szignó alkalmazásnak van egy felhasználói felülettel rendelkező, grafikus kliens változata, és egy más szoftverekbe integrálható, e-Szignó Automata változata.
- Az e-Szignó többféle platformon is elérhető. Windows alatt képes használni a Windows PKI funkcionalitását (tanúsítványtár, CSP-k stb), más platformokon a megadott tanúsítványokat, szoftveres magánkulcsokat és PKCS#11 könyvtárakat használja.
- Az e-Szignó kliens regisztrálatlan, ingyenes változata éles környezetben is használható aláírások ellenőrzésére, illetve a Microsec által kibocsátott tanúsítványok szerint történő aláírásra.
- Az e-Szignó Automata többféle módon integrálható rendszerekbe; akár központi aláírás-kezelő szerverként, akár egy kliens programba épített aláíró modulként, akár webes technológiával.

- Az e-Szignó Automata parancssoros változata nagyon gyorsan integrálható, és egy különálló egységet képez. A programkönyvtárakból elérhető változat integrációja általában tovább tart, de így sokkal jobb teljesítmény érhető el.

B függelék

Hivatkozások

[1] J. Almási – L. Balázs – P. M. Erdősi – Á. Kovács – B. Rátai – J. Schvéger: Elektronikus hitelesség, elektronikus aláírás. ISBN: 978 963 06 8727 0, OTY StarTel Kft., 2009.

[2] Az Adó- és Pénzügyi Ellenőrzési Hivatal közleménye az elektronikus úton kibocsátott számlákra vonatkozó egyes rendelkezések értelmezéséről. `http://www.apeh.hu/adoinfo/afa080101_hatalyos/elektronikus_szamla.html`, 2009.

[3] Dirk Balfanz – Ed Felten: Hand-Held Computers Can Be Better Smart Cards. Proceedings of USENIX Security '99 Washington, DC., 1999.

[4] B. Bencsáth – I. Vajda: Collecting randomness from the net. Proceedings of the IFIP TC6 and TC11 Joint Working Conference on Communications and Multimedia Security 2001, Kluwer, May, 2001, pp. 105-111., 2001.

[5] I. Zs. Berta: Mi alapján fogadhatunk el egy elektronikus aláírást? Híradástechnika, 2006, vol. LXI, 2006.

[6] I. Zs. Berta – A. Barsi: Hiteles iratok készítése elektronikus aláírás segítségével. Intelligens Települések Országos Szövetsége (http://www.itosz.hu) & e-Írástudásért Közhasznú Alapítvány (http://www.e-irastudasert.hu), ISBN 978-963-06-5356-5, 2008., 2008.

[7] I. Zs. Berta – L. Buttyán – I. Vajda: Mitigating the untrusted terminal problem using conditional signatures. CrySyS Lab Technical Report, `http://www.crysys.hu/publications/files/BertaBV2004condsig.pdf`, 2003.

[8] I. Zs. Berta – L. Buttyán – I. Vajda: A framework for the revocation of unintended digital signatures initiated by malicious terminals. IEEE Transactions on Secure and Dependable Computing, vol. (Vol. 2, No. 3), pp. 268-272, July-September, 2005.

[9] I. Zs. Berta – I. Vajda: Documents from Malicious Terminals. SPIE Microtechnologies for the New Millenium 2003, Bioengineered and Bioinspired Systems, Spain, 2003.

[10] I. Zs. Berta – I. Vajda: Limitations of humans at malicious terminals. Tatra Mountains Mathematical Publications, vol. 29, pp 1-16, 2004.

[11] I. Zs. Berta – L. Buttyán – I. Vajda: Standards for Product Security Assessment. Handbook of Information Security, Chapter 55, Edited by Hossein Bidgoli, John Wiley and Sons, accepted, to appear, 2005.

[12] I. Zs. Berta – Z. Á. Mann: Programozható chipkártyák - elmélet és gyakorlati tapasztalatok. Magyar Távközlés, 2000, vol 4, 2000.

[13] I. Zs. Berta – Z. Á. Mann: Smart Cards – Present and Future. Híradástechnika, Journal on C^5, 2000., vol 12, 2000.

[14] I. Zs. Berta – Z. Á. Mann: Evaluating Elliptic Curve Cryptography on PC and Smart Card. Periodica Polytechnica, Electrical Engineering, 2002, vol. 46/1-2, pp. 47-75, Budapest University of Technology and Economics, 2002. 4.

[15] A. Biryukov – D. Khovratovich: Related-key cryptanalysis of the full aes-192 and aes-256. `https://cryptolux.org/mediawiki/uploads/1/1a/Aes-192-256.pdf`, 2009.

[16] D. Boneh: Twenty years of attacks on the RSA cryptosystem. Notices of the American Mathematical Society (AMS), Vol. 46, No. 2, pp. 203-213, 1999, 1999.

[17] D. Boneh – M. K. Franklin: Identity-based encryption from the weil pairing. Advances in Cryptology - Proceedings of CRYPTO 2001, 2001.

[18] I. Bowman: The History of Electronic Signature Laws. `http://www.isaacbowman.com/the-history-of-electronic-signature-laws`, 2009.

[19] R. Bragg: Cross certification trusts. Microsoft Certified Professional Magazine Online, `http://mcpmag.com/articles/2003/11/01/cross-certification-trusts.aspx`, 2003.

[20] F. Buccafurri – G. Caminiti – G. Lax: The Dali Attack on Digital Signature. Journal of Information Assurance and Security 3 (2008) 185-194, 2008.

[21] L. Buttyán – I. Vajda: Kriptográfia és Alkalmazásai. Typotex, Budapest, 2004.

[22] CA/Browser Forum: Guidelines for the issuance and management of Extended Validation certificates. `http://www.cabforum.org/Guidelines_v1_2.pdf`, 2007.

[23] C. A. Carnall: Managing change in organisations. Prentice Hall, 2003.

[24] Cert.org, Home Computer Security. http://www.cert.org, 2002.

[25] Dwaine Clarke – Blaise Gassend – Thomas Kotwal – Matt Burnside – Marten van Dijk – Srinivas Devadas – Ronald Rivest: The Untrusted Computer Problem and Camera-Based Authentication, 2002.

[26] Computerworld: Elektronikus eljárás a bírósági végrehajtásban. http://computerworld.hu/elektronikus-eljaras-a-birosagi-vegrehajtasban.html, 2009.

[27] Computerworld: Második generációs biometrikus útlevelek. Számítástechnika, 2009.12.16., 2009.

[28] CEN 14167-1 munkacsoport egyezmény: "Biztonsági követelmények elektronikus aláírásokkal kapcsolatos tanúsítványokat kezelő rendszerek megbízható rendszereire".

[29] CEN CWA 14169: Secure signature-creation devices “EAL 4+”, March 2004.

[30] CEN CWA 14170: Security Requirements for Signature Creation Applications.

[31] CEN CWA 14171: Procedures for Electronic Signature Verification.

[32] W. Diffie – M. E. Hellman: New directions in cryptography. IEEE Transactions on Information Theory, volume IT-22, number 6, pp 644–654, November 1976, citeseer.ist.psu.edu/diffie76new.html, 1976.

[33] H. Dreifus – J. Thomas Monk: Smart cards: guide to building and managing smart card applications. John Wiley and Sons, Inc. ISBN 0-471-15748-1, 1998.

[34] DSA-1571-1 openssl – predictable random number generator. Debian Security Advisory, DSA-1571-1, http://www.debian.org/security/2008/dsa-1571, 2008.

[35] Dublin core metadata initiative. http://www.dublincore.org/, 2010.

[36] Az e-akta formátum specifikációja, v1.0, Microsec Kft. http://www.e-szigno.hu/?lap=eakta3/, 2008.

[37] e-Szignó3 Használati Útmutató. v3.2, 2011.

[38] ECC Brainpool Standard Curves and Curve Generation, http://www.ecc-brainpool.org/, 2005.

[39] The Economist: The Difference Engine: Dubious security. http://www.economist.com/blogs/babbage/2010/10/biometrics, 2010.

[40] Carl Ellison – Bruce Schneier: Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure. Computer Security Journal, Volume XVI, Number 1, 2000, 2000.

[41] P. Erdősi: Az elektronikus aláírás avagy a szerzői jogok védelme. NJSZT Mi újság, 2007. június, http://www.njszt.hu/files/neumann/miujsag/2007/Mi_Ujsag_2007_junius.pdf, 2007.

[42] Electronic Signatures in Global and National Commerce Act. Public Law 106–229—June 30, 2000, http://www.fca.gov/download/public law 106-229 e-sign.pdf, 2000.

[43] ETSI SR 003 232 PDF Advanced Electronic Signature Profiles (PAdES); Printable Representations of Electronic Signatures, 2011.

[44] ETSI TR 101 533-2 Information Preservation Systems Security - Part 2 Guidelines for Assessors, 2011.

[45] XML format for signature policies, 2002.

[46] ASN.1 format for signature policies, 2003.

[47] ETSI TS 101 456 Minősített tanúsítványokat kibocsátó hitelesítés-szolgáltatókra vonatkozó szabályozási követelmények.

[48] ETSI TS 101 533-1 Information Preservation Systems Security - Part 1: Requirements for Implementation and Management, 2011.

[49] ETSI TS 101 733 CMS Advanced Electronic Signatures (CAdES), 2004.

[50] ETSI TS 101 862 Qualified Certificate Profile.

[51] ETSI TS 101 903 XML Advanced Electronic Signatures (XAdES), 2004.

[52] ETSI TS 102 023: Policy requirements for time-stamping authorities.

[53] ETSI TS 102 042 Policy requirements for certification authorities issuing public key certificates.

[54] ETSI TR 102 042 Requirements for role and attribute certificates.

[55] ETSI TR 102 044 Requirements for role and attribute certificates.

[56] ETSI TS 102 158 Policy requirements for Certification Service Providers issuing attribute certificates usable with Qualified certificates, v1.1.1, 2003.

[57] ETSI TS 102 176-1 V2.0.0 (2007-11) Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms, 2007.

[58] ETSI TS 102 231 Provision of harmonized Trust-service status information, http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=31945, 2006.

[59] ETSI TS 102 280: X.509 V.3 Certificate Profile for Certificates Issued to Natural Persons (v1.1.1; 2004-03).

[60] ETSI TS 102 778 PDF Advanced Electronic Signatures, PAdES, 2009.

[61] ETSI TS 102 918 Associated Advanced Electronic Signatures (AAdES), 2011.

[62] EU Directive 1999/93/EC of the European Parliament and the council of 13 December 1999 on a Community framework for electronic signatures;, 1993.

[63] Európai Unió Tanácsa 2252/2004/EK rendelete a tagállamok által kiállított útlevelek és úti okmányok biztonsági jellemzőire és biometrikus elemeire vonatkozó előírásokról, 2004.

[64] Security requirements for cryptographic modules. Federal Information Processing Standards Publication 140-1, 1994, http://csrc.ncsl.nist.gov/fips.

[65] FIPS PUB 186-3 - Digital Signature Standard (DSS), Federal Information Processing Standards Publication, National Institute of Standards and Technology, http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf, 2009.

[66] J. L. Fisher: Side-effects of cross-certification.
4th Annual PKI R&D Workshop, NIST, Gaithersburg MD, USA, http://middleware.internet2.edu/pki05/proceedings/fisher-cross_cert.pdf, 2005.

[67] FPKI: Federal Public Key Infrastructure (FPKI) Architecture Technical Overview. http://www.cio.gov/fbca/documents/FPKIAtechnicalOverview.pdf, 2005.

[68] G. Gadó: Ítéletre várva. HVG, 2010.12.16., 2010.

[69] 114/2007. (XII. 29.) GKM rendelet a digitális archiválás szabályairól, 2007.

[70] O. Goldreich: The Foundations of Modern Cryptography. In Proceedings of Crypto97, Springer's Lecture Notes in Computer Science, Vol. 1294, http://theory.lcs.mit.edu/ ~oded/frag.html, 1997.

[71] Johann Großschädl: The Chinese Remainder Theorem and its Application in a High-Speed RSA Crypto Chip. ACSAC '00 Proceedings of the 16th Annual Computer Security Applications Conference, 2000.

[72] Government smart card interoperability specification. NIST, Interagency Report 6887, http://csrc.nist.gov/publications/nistir/nistir-6887.pdf, 2003.

[73] P. Gutmann: Everything you Never Wanted to Know about PKI but were Forced to Find Out. http://www.cs.auckland.ac.nz/ pgut001/pubs/pkitutorial.pdf.

[74] P. Gutmann: Secure deletion of data from magnetic and solid-state memory. Sixth USENIX Security Symposium Proceedings, San Jose, California, July 22-25, 1996, http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html, 1996.

[75] P. Gutmann: X.509 Style Guide.
http://www.cs.auckland.ac.nz/ pgut001/pubs/x509guide.txt, 2000.

[76] P. Gutmann: Pki: It's not dead, just resting. Computer, vol. 35, no. 8, pp. 41-49, Aug. 2002, doi:10.1109/MC.2002.1023787, 2002.

[77] L. Györfi – S. Győri – I. Vajda: Információ és kódelmélet. Typotex, ISBN: 9639132845, 2000.

[78] D. Husemöller: Elliptic Curves (Graduate Text in Mathematics). Springer, ISBN: 0387954902, 1987.

[79] 13/2005. IHM rendelet a papíralapú dokumentumokról elektronikus úton történő másolat készítésének szabályairól, 2005.

[80] 3/2005. IHM rendelet az elektronikus aláírással kapcsolatos szolgáltatásokra és ezek szolgáltatóira vonatkozó részletes követelményekről, 2005.

[81] 7/2005. (VII. 18.) IHM rendelet a digitális archiválás szabályairól, valamint az információs társadalommal összefüggő szolgáltatásokkal kapcsolatos elektronikus archiválás szabályairól, 2005.

[82] Az Informatikai és Hírközlési Minisztérium ajánlása a közigazgatásban alkalmazható végfelhasználói tanúsítványok szerkezetének és adattartalmának műszaki specifikációjára, 2006., 2006.

[83] Az Informatikai és Hírközlési Minisztérium ajánlása a közigazgatásban alkalmazható hitelesítési rendekre, 2006.

[84] Az Informatikai és Hírközlési Minisztérium ajánlása a közigazgatásban alkalmazható időbélyegzés formátum műszaki specifikációjára, 2006.

[85] Az Informatikai és Hírközlési Minisztérium ajánlása a közigazgatásban alkalmazható elektronikus aláírás formátumok műszaki specifikációjára, 2006., 2006.

[86] Az Informatikai és Hírközlési Minisztérium ajánlása a közigazgatásban a hitelesítés-szolgáltatók által végzett viszontazonosítás protokolljának műszaki specifikációjára, 2006., 2006.

[87] ISO 32000 Document management – Portable document format – Part 1: PDF 1.7, 2008.

[88] ISO/IEC 7816 Part 4: Interindustry command for interchange.

[89] Információ technológia - Nyílt rendszerek kapcsolódása - Könyvtár: Nyilvános kulcs és attribútum tanúsítvány keretrendszer.

[90] D. Johnson – A. Menezes – S. Vanstone: The Elliptic Curve Signature Algorithm (ECDSA). `http://www.comms.scitech.susx.ac.uk/fft/crypto/ecdsa.pdf`, 2001.

[91] D. B. Johnson – A. J. Menezes: Elliptic curve DSA (ECDSA): an enhanced DSA. http://citeseer.nj.nec.com/276964.html, 1998.

[92] Burton S. Jr. Kaliski: A layman's guide to a subset of ASN.1, BER, and DER. An RSA Laboratories Technical Note, Revised November 1, 1993, `http://luca.ntop.org/Teaching/Appunti/asn1.html`, 2009.

[93] A. Kerckhoffs: La Cryptographie Militaire. Journal des Sciences Militaires, 1883, Jan, 1883.

[94] A Közigazgatási Gyökér Hitelesítés Szolgáltató (KGYHSZ) hitelesítési rendje, http://www.kgyhsz.gov.hu/KGYHSZ_HR_v1.0.pdf, 1.0.

[95] N. Koblitz: Elliptic Curve Cryptosystems. Mathematics of Computation, Vol. 48. No. 177, 1987, 203-209., 1987.

[96] 193/2005. Korm. rendelet, 2005.

[97] 194/2005. (IX.22.) Korm. rendelet a közigazgatási hatósági eljárásokban használt elektronikus aláírásokra és az azokhoz tartozó tanúsítványokra, valamint a tanúsítványokat kibocsátó hitelesítés szolgáltatókra vonatkozó követelményekről., 2005.

[98] 195/2005. (IX.22.) Korm. rendelet az elektronikus ügyintézést lehetővé tevő informatikai rendszerek biztonságáról, együttműködési képességéről és egységes használatáról., 2005.

[99] 78/2010. (III. 25.) Korm. rendelet az elektronikus aláírás közigazgatási használatához kapcsolódó követelményekről és az elektronikus kapcsolattartás egyes szabályairól, 2010.

[100] P. Kovács: Ügyfélkapu apraja falván. Ügyvédek lapja, XLVIII. évf, 2. szám, 2009, március., 2009.

[101] S. Kremer – O. Markowitch – J. Zhou: An intensive survey of fair non-repudiation protocols. Computer Communications, 25(17): pp. 1606–1621. November 2002., 2002.

[102] B. Kéki: Az írás története. Gondolat, 1971.

[103] L. Lamport: Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, Oct. 1979., 1979.

[104] M. Marlinspike: Internet explorer ssl vulnerability 08/05/02. `http://www.thoughtcrime.org/ie-ssl-chain.txt`, 2002.

[105] M. Marlinspike: New tricks for defeating SSL in practice. Blackhat Conference, 2009, `http://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf`, 2009.

[106] M. Matsui: Linear cryptanalysis method for des cipher. Advances in Cryptology – EUROCRYPT 1993., 1993.

[107] T. Matsumoto: Human-Computer cryptography: An attempt. In ACM Conference on Computer and Communications Security, pp 68-75, 1996.

[108] T. Matsumoto – H. Matsumoto – K. Yamada – S. Hoshino: Impact of Artificial Gummy Fingers on Fingerprint Systems. Proceedings of SPIE Vol. 4677, Optical Security and Counterfeit Deterrence Techniques IV, 2002.

[109] U. Maurer: A universal statistical test for random number generators. Journal of Cryptology, vol.5, no. 2., 1992, pp. 89-105., 1992.

[110] C. McDonald – P. Hawkes – J. Pieprzyk: SHA-1 collisions now 2^{52}. `http://eurocrypt2009rump.cr.yp.to/837a0a8086fa6ca714249409ddfae43d.pdf`, 2009.

[111] Egységes melasz formátum elektronikus aláírásokra verzió: 2.0. Melasz Munkacsoport Megállapodás, MMM 001:2008, 2008.

[112] R. Merkle: Protocols for public key cryptosystems. Proceedings of the 1980 IEEE Symposium on Security and Privacy (Oakland, CA, USA)", pages 122-134, April 1980., 1980.

[113] Microsoft Root Certificate Program. `http://technet.microsoft.com/en-us/library/cc751157.aspx`, 2010.

[114] V. S. Miller: Use of Elliptic Curves in Cryptography. CRYPTO '85, volume 12, 1985.

[115] Mozilla CA Certificate Policy.
http://www.mozilla.org/projects/security/certs/policy/, 2011.

[116] A. Mártonffy: Virtuális szerződés valódi biztonsággal. IT Business, 2010. 09. 07, 20-21. oldal, 2010.

[117] Moni Naor – Adi Shamir: Visual Cryptography. Lecture Notes in Computer Science, vol 950, pp 1–12, 1995, http://citeseer.nj.nec.com/naor95visual.html, 1995.

[118] A Nemzeti Hírközlési Hatóság Hivatalának tájékoztatója a láncolt hitelesítés-szolgáltatásokról és más alárendelt elektronikus aláírással kapcsolatos szolgáltatásokról. http://www.nhh.hu/dokumentum.php?cid=12060, 2007.

[119] A Nemzeti Hírközlési Hatóság Hivatalának tájékoztatója az elektronikus aláírás célú tanúsítványok kibocsátásakor végrehajtandó regisztrációs tevékenységről, az ezért való felelősségről, valamint ennek során a közjegyzői közreműködésről (2007. május 22.). http://www.nhh.hu/dokumentum.php?cid=12059, 2007.

[120] Elektronikus archiválási szolgáltatással kapcsolatos hatósági tájékoztató, Nemzeti Hírközlési Hatóság Hivatala, 2008. június.

[121] Ajánlás eljárásrendi követelményekre elektronikus aláírás felhasználásával végzett elektronikus archiválási szolgáltatások szolgáltatói számára, Nemzeti Hírközlési Hatóság Hivatala, 2008. június.

[122] Ajánlás elektronikus archiválási szolgáltatások nyújtásához felhasznált megbízható rendszerekre vonatkozó biztonsági követelményekre, Nemzeti Hírközlési Hatóság Hivatala, 2008. június.

[123] Minősített tanúsítvány visszavonással és felfüggesztéssel kapcsolatos hatósági állásfoglalás, Nemzeti Hírközlési Hatóság, ügyiratszám: HL-16624-2/2005, http://www.nhh.hu/dokumentum.php?cid=9182.

[124] E. Nigg: Untrusted certificates. Blog, https://blog.startcom.org/?p=145, 2008.

[125] E. Nigg: Untrusted Certificates. StartCom blog, https://blog.startcom.org/?p=145, 2008.

[126] NIST – Cryptographic Hash Algorithm Competition. http://csrc.nist.gov/groups/ST/hash/sha-3/incryptographic hash Algorithm Competitionindex.html, 2010.

[127] Randomized hashing for digital signatures. Special Publication 800-160m, csrc.nist.gov/publications/nistpubs/800-106/NIST-SP-800-106.pdf , 2009.

[128] Recommendation for key management. Special Publication 800-57 Part 1, NIST, 03/2007, http://csrc.nist.gov/groups/ST/toolkit/key_management.html, 2007.

[129] NSA Suite B Cryptography. http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml, 2009.

[130] International Civil Aviation Organization: Machine readable travel documents, part 1 machine readable passports, volume 2 specifications for electronically enabled passports with biometric identification capability. 6th Edition, 2006.

[131] Origo.hu: Elektronikus kézbesítés III. - Informatikai biztonsági aggályok. http://www.origo.hu/uzletinegyed/jog/uzleti/20091210-jog-kezbesit-elektronikus.html, 2009.

[132] Origo.hu: Százmilliós megtakarítás a végrehajtók elektronikus üzenetközvetítő rendszere. http://www.origo.hu/uzletinegyed/jog/uzleti/20101108-szazmillios-megtakaritas-a-vegrehajtok-elektronikus-uzenetkozvetito-rendszere.html, 2010.

[133] PKCS#10 v1.7: Certification Request Syntax Standard. http://www.rsalabs.com, 2000.

[134] PKCS#11: Cryptographic Token Interface Standard. http://www.rsalabs.com.

[135] PKCS#15. http://www.rsalabs.com, 2004.

[136] PKCS#7: Cryptographic message syntax standard. http://www.rsalabs.com, 1993.

[137] Attribute Certificate Request Message Format, PKIX draft, 2002.

[138] 46/2007. PM rendelet az elektronikus számlával kapcsolatos egyes rendelkezésekről, 2007.

[139] W. Rankl – W. Effing: Smart Card Handbook. John Wiley & Sons, 2nd edition, ISBN: 0471988758, 1997.

[140] RFC 2560: Online Certificate Status Protocol (OCSP).

[141] Electronic signature policies. http://tools.ietf.org/html/rfc3125, 2001.

[142] RFC 3161: Time-Stamp Protocol (TSP).

[143] RFC 3280 (Internet X.509 Nyilvános kulcsú infrastruktúra - tanúsítvány és tanúsítvány visszavonási lista profil).

[144] RFC 3281 An Internet Attribute Certificate Profile for Authorization, 2002.

[145] RFC 3647 (Internet X.509 Nyilvános kulcsú infrastruktúra - tanúsítványtípus és szolgáltatási szabályzat keretrendszer).

[146] Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1, Message Specification, http://tools.ietf.org/html/rfc3851, 2004.

[147] RFC 4158 Internet X.509 Public Key Infrastructure: Certification Path Building, http://www.ietf.org/rfc/rfc4158.txt, 2005.

[148] RFC 4810 Long-Term Archive Service Requirements http://www.rfc-archive.org/getrfc.php?rfc=4810, 2007.

[149] RFC 4998 Evidence Record Syntax (ERS) http://www.ietf.org/rfc/rfc4998.txt, 2007.

[150] CMS Advanced Electronic Signatures (CAdES), 2008.

[151] RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2, http://tools.ietf.org/html/rfc5246, 2009.

[152] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, 2008.

[153] RFC 5652 Cryptographic Message Syntax (CMS), http://www.ietf.org/rfc/rfc5652.txt, 2009.

[154] Transport Layer Security (TLS) Renegotiation Indication Extension.

[155] R. Rivest: Chaffing and winnowing: Confidentiality without encryption. MIT Lab for Computer Science, 1998-03-22, (http://theory.lcs.mit.edu/~rivest/chaffing.txt), 2008.

[156] R. Rivest – A. Shamir – L. M. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. MIT/LCS/TM-82, http://citeseer.nj.nec.com/rivest78method.html, 1978.

[157] RSALabs: PKCS#1 v2.1: RSA Cryptography Standard. http://www.rsalabs.com, 2002.

[158] Z. Révay: Titkosírások – Fejezetek a rejtjelezés történetéből. LAZI Könyvkiadó, Szeged, 1978, ISBN: 963 9227 82 X, 1978.

[159] L. Rónyai: Elliptikus görbék és a Fermat-sejtés. Matematikai Lapok, 1995.

[160] B. Schneier: Attack Trees. Dr. Dobb's Journal December 1999, http://www.schneier.com/paper-attacktrees-ddj-ft.html, 1999.

[161] B. Schneier: One-time pads. Crypto-Gram Newsletter, October, 2002, http://www.schneier.com/crypto-gram-0210.html#7, 2002.

[162] B. Schneier: Cryptanalysis of SHA-1. Schneier on Security, http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html, 2005.

[163] B. Schneier: SHA-1 Broken. Schneier on Security, http://www.schneier.com/blog/archives/2005/02/sha1_broken.html, 2005.

[164] Bruce Schneier: Applied Cryptography. John Wiley & Sons, ISBN: 0471117099, 1996.

[165] Bruce Schneier: The Solitaire Encryption Algorithm. http://www.counterpane.com/solitaire.htm, 1999.

[166] A. Shamir: How to share a secret. Communications of the ACM 22 (11): 612–613, doi:10.1145/359168.359176, 1979.

[167] A. Shamir: Identity-based cryptosystems and signature schemes. Advances in Cryptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science, 7:47–53, 1984 http://www.iseca.org/modules/mydownloads/visit.php?cid=56&lid=33, 1984.

[168] C. E. Shannon: Communication Theory of Secrecy Systems. Bell Systen Technical Journal, vol 28, pp 656–715, 1949, 1949.

[169] Gesetz über Rahmenbedingungen für elektronische Signaturen (Signaturgesetz - SigG). Signaturgesetz vom 16. Mai 2001 (BGBl. I S. 876), das zuletzt durch Artikel 4 des Gesetzes vom 17. Juli 2009 (BGBl. I S. 2091) geändert worden ist http://bundesrecht.juris.de/sigg_2001/BJNR087610001.html, 2001.

[170] Gustavus J. Simmons: A „weak" privacy protocol using the RSA cryptoalgorithm. Cryptologia, 7(2), Apr 1983., 1983.

[171] Simon Singh: The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. Anchor Books; ISBN: 0385495323, 2000.

[172] New Stuxnet-Related Malware Signed Using Certificate from JMicron, http://news.softpedia.com/news/New-Stuxnet-Related-Malware-Signed-Using-Certificate-from-JMicron-148213.shtml, 2010.

[173] A. Sotirov: Creating a rogue CA certificate. http://www.phreedom.org/research/rogue-ca/, 2008.

[174] Tage Stabell-Kulo – Ronny Arild – Per Harald Myrvang: Providing Authentication to Messages Signed with a Smart Card in Hostile Environments. Usenix Workshop on Smart Card Technology, Chicago, Illinois, USA, May 10-11, 1999., 1999.

[175] Sun Microsystems Inc.: Java Card (TM) 2.1.1 Application Programming Interface. Sun Microsystems, Inc. 901 San Antonio Road Palo Alto, CA 94303 USA 650 960-1300, 2000. 5. http://java.sun.com/javacard.

[176] Digital tachograph system European root policy. `http://dtc.jrc.it/docs/SPI0416.pdf`, 2004.

[177] Ken Thompson: Reflections on Trusting Trust. Communication of the ACM, Vol 29. No. 8, August, 1984 pp 761-763, 1984.

[178] J. Tumbull: Cross-certification and PKI policy networking. Entrust, `http://www.entrust.com/resources/pdf/cross_certification.pdf`, 2000.

[179] 1992. évi LXIII. törvény a személyes adatok védelmérõl és a közérdekû adatok nyilvánosságáról, 1992.

[180] 2001. évi XXXV. törvény az elektronikus aláírásról (a 2004. évi módosításokkal), 2001.

[181] A közigazgatási hatósági eljárásról szóló 2004. évi CXL. törvény, 2004.

[182] 1991. évi XLI. törvény a közjegyzőkről, 1991.

[183] 1998. évi XII. törvény a külföldre utazásról, 1998.

[184] 1952. évi III. törvény a polgári perrendtartásról, 1952.

[185] 2007. évi CXXVII. törvény az általános forgalmi adóról, 2007.

[186] Supporting Interoperability in Preservation and Retrieval of digital Objects (SInCRO). UNI U30110600, 2010.

[187] US Government Family of Protection Profiles Public Key-Enabled Applications For Basic Robustness Environments (v2.8, May 2007).

[188] P. Wayner: Disappearing Cryptography, Third Edition: Information Hiding: Steganography and Watermarking. Morgan Kaufmann, 2008.

[189] AICPA/CICA WebTrust sm/tm Program for Certification Authorities. `http://www.webtrust.org/item27804.pdf`, 2000.

[190] Why Phishing Works? `http://people.deas.harvard.edu/~rachna/papers/why_phishing_works.pdf`.

[191] ITU X.509 "Információ technológia - Nyílt rendszerek kapcsolódása - Könyvtár: Nyilvános kulcs és attribútum tanúsítvány keretrendszer" ajánlás 5. kiadás, 2005.

[192] Information technology – ASN.1 encoding rules: Specification of basic encoding rules (BER), canonical encoding rules (CER) and distinguished encoding rules (DER). ITU-T Recommendation,
`http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf`,
2009.

[193] XML Encryption Syntax and Processing. W3C Recommendation 10 December 2002, `http://tools.ietf.org/html/rfc5246`, 2002.

[194] P. Zimmermann – A. Johnston – J. Callas: The ZRTP internet draft.
`http://zfoneproject.com/zrtp_ietf.html`, 2008.

[195] J. L. Zoreda – J. M. Oton: Smart cards. Artech House, ISBN: 0890066876, 1994.

C függelék

Szabványok

Az alábbiakban a PKI szabványokban való eligazodáshoz adunk rövid útmutatót.

X.509: A tanúsítványok, tanúsítványláncok, visszavonási listák és attribútum-tanúsítványok szerkezetét és a tanúsítványláncok ellenőrzésének módját leíró X.509 specifikáció az egyik legfontosabb PKI szabvány. [191] **ISO 9594-8** néven nemzetközi szabványként is kibocsátották. [89]

RFC 5280: Az X.509 egy részhalmaza, az Interneten használható tanúsítványok és rájuk vonatkozó visszavonási listák szerkezetét és a tanúsítványláncok ellenőrzését írja le. [152]

RFC 5652 (CMS): Olyan (ASN.1) üzenet-formátumot határoz meg, amely szerint be lehet csomagolni egy titkosított üzenetet, illetve aláírást lehet fűzni egy üzenethez. Az IETF a **PKCS#7** specifikációt emelte át, minimális eltérésekkel. [153]

RFC 3161: Az időbélyegek formátumát és az időbélyegek lekérésére vonatkozó protokollt írja le. Nem foglalkozik az időbélyegen lévő aláírással, e tekintetben az RFC 5652-t hivatkozza meg. [142]

RFC 2560: Az OCSP protokollt írja le. (Az X.509 nem szó az OCSP-ről.) [140]

RFC 3647: A hitelesítési rendek (és szolgáltatási szabályzatok) szerkezetére ad sablont. A szolgáltatók rendjei és szabályzatai általában az itt leírt struktúrát követik. [145]

PKCS#1: Az RSA algoritmust, illetve az RSA alapon titkosított és aláírt blokk szerkezetét és az alkalmazható padding megoldásokat írja le. [157]

PKCS#7: Kriptográfiai üzenetformátumot (titkosított, illetve aláírt üzenetet) ír le, minimálisan tér el az RFC 5652 szerinti formátumtól. [136]

PKCS#10: A tanúsítványkérelem (nyilvános kulcs és megnevezés, a benne lévő nyilvános kulcshoz tartozó magánkulccsal aláírva) formátumát írja le. [133]

PKCS#11: Olyan interfészt ír le, amelyen keresztül egy kriptográfiai token (intelligens kártya vagy HSM) funkcionalitása érhető el. [134]

CWA 14171: Az aláírás ellenőrzésének módját írja le, így pl. szól a kivárási időről is. Sok tekintetben az RFC 5280-at (illetve annak korábbi változatát, az RFC 3280-at) hivatkozza meg. [31]

CWA 14168: A biztonságos aláírás-létrehozó eszközökre vonatkozó Common Criteria védelmi profilt (SSCD PP) határozza meg; a BALE intelligens kártyák többségét ma e specifikáció szerint tanúsítják. [29]

ETSI TS 101 456 (QCP, QCP+SSCD): Minősített tanúsítványokra vonatkozó hitelesítési rendeket határoz meg. [47]

ETSI TS 102 042 (NCP+, NCP, LCP): Nem minősített tanúsítványokra vonatkozó hitelesítési rendeket határoz meg. [53]

ETSI TS 101 862: A minősített tanúsítványok formátumát határozza meg. Leírja, hogy a minősített tanúsítványok miben különböznek más tanúsítványoktól. [50]

ETSI TS 102 280: A természetes személyek számára kibocsátott tanúsítványok (beleértve a minősített tanúsítványokat is) formátumát határozza meg. [59]

ETSI TS 101 733 (CAdES): Az RFC 5652, illetve PKCS#7 specifikációkra épülő „EU-s" aláírás-formátumot határoz meg, amely kiterjeszthető időbélyeggel és visszavonási információkkal, valamint archiválható. [49] **RFC 5126** néven is kibocsátották. [150]

ETSI TS 101 903 (XAdES): A W3C **XMLDSIG** szerinti XML aláírás formátumára épülő „EU-s" aláírás-formátumot határoz meg, amely kiterjeszthető időbélyeggel és visszavonási információkkal, valamint archiválható. [51]

ETSI TS 102 778 (PAdES): Leírja, hogyan helyezhető el XAdES, illetve CAdES aláírás PDF dokumentumokban. [60]

ETSI TS 102 176-1 (Algo paper): Az aláírás készítésére, illetve tanúsítványok és időbélyegek aláírására használható kriptográfiai algoritmusokra fogalmaz meg ajánlásokat. [57]

D függelék

Rövidítések

AA: attribute authority;

AC: attribute certificate;

AES: advanced encryption standard;

AGA: attribute granting authority;

AIA: authority information access – mező az X.509 tanúsítványokban;

AIX: Advanced Interactive eXecutive – az IBM Unix alapú operációs rendszere;

AKS: Agrawal - Kayal - Saxena – determinisztikus prímteszt, amely a három feltalálójáról kapta a nevét;

APDU: application protocol data unit – intelligens kártyáknak ilyen módon küldhetünk parancsokat;

APEH: Adó- és Pénzügyi Ellenőrzési Hatóság;

ASN: abstract syntax notation – leírónyelv, amelyen adatobjektumok formátumát lehet meghatározni;

BALE: biztonságos aláírás-létrehozó eszköz;

BES: basic electronic signature;

BMP: bitmap – fájlformátum;

CA: certification authority – hitelesítés-szolgáltató;

CAdES: CMS advanced electronic signature – az ETSI által kidolgozott aláírás-formátum, amely a CMS-re épül, így ASN.1 alapú;

CBC: cipher block chaining – a blokkrejtjelezők egy működési üzemmódja;

CD: compact disc;

CDP: CRL distribution point – mező az X.509 tanúsítványban;

CGI: common gateway interface – egy webszerveren futó alkalmazás így éri el a böngésző által küldött információkat;

CICA: Canadian Institute of Chartered Accountants;

CMS: cryptographic message syntax;

CN: common name – az X.509 tanúsítványban szereplő megnevezés egyik eleme;

COM: Component Object Model;

CP: certificate policy – hitelesítési rend;

CRC: cycle redundancy check – hibadatektáló kód;

CRL: certificate revocation list – visszavonási lista;

CRT: chinese remainder theorem – kínai maradéktétel;

CSCA: country signing CA – az egyes EU tagállamok útlevél-kibocsátó rendszereinek tanúsítványait kibocsátó hitelesítés-szolgáltató;

CSP: certification service provider – hitelesítés-szolgáltató (időnként más bizalmi szolgáltatót is értenek e kifejezés alatt);

CSV: comma separated values – fájlformátum, amely vesszővel (vagy pontosvesszővel) elválasztott értékeket tartalmaz;

CV: card verifiable;

CVCA: card verifiable (certificate issuing) CA – az egyes EU tagállamok útlevél-ellenőrző rendszereinek tanúsítványai előállításához használt gyökér hitelesítés-szolgáltató;

CWA: CEN workshop agreement – az európai szabványosítási testület, a CEN (Comité Européen de Normalisation) által kibocsátott, szabvány-jellegű dokumentumok elnevezése;

DER: data encoding rules – az ASN.1 nyelven leírt adatstruktúrák a DER segítségével bitsorozatokká képezhetőek le;

DES: data encryption standard – egy blokkrejtjelező elnevezése;

DLP: discrete logarithm problem – diszkrét logaritmus probléma;

DN: distinguished name – az X.509 tanúsítványokban ilyen módon hivatkozhatunk a tanúsítvány alanyára és kibocsátójára;

DNS: domain name service;

DS: digital signature;

DSA: digital signature algorithm – egy DLP-re épülő aláíró algoritmus elnevezése;

DV: domain validated: olyan webszerver tanúsítvány, amely esetén a hitelesítés-szolgáltató csak annyit ellenőrzött, hogy a tanúsítvány alanya birtokolja-e a tanúsítványban szereplő domaint; VAGY document verifier – pl. útlevél-ellenőrző készülék;

DVD: Digital Versatile Disc;

EA: e-mail address – az X.509 tanúsítványban szereplő megnevezés egyik eleme;

EC: elliptic curve;

ECB: electronic codebook – a blokkrejtjelezők egy működési üzemmódja;

ECC: elliptic curve cryptography – az elliptikus görbék elméletére épülő kriptográfiai megoldások összefoglaló neve;

ECDH: elliptic curve Diffie-Hellman;

ECDLP: elliptic cure discrete logarithm problem – elliptikus görbék felett értelmezett DLP;

ECDSA: elliptic curve digital signature algorithm – elliptikus görbék felett értelmezett DSA;

EDI: electronic data interchange;

EEPROM: Electrically Erasable Programmable Read-Only Memory;

EK: Európai Közösség;

EMC: electromagnetic compatibility;

EPES: explicit policy electronic signature – olyan aláírás, amely meghivatkozik egy aláírási szabályzatot;

ERS: evidence record syntax – az LTANS munkacsoport által kidolgozott archív-megoldás;

ESI: Electronic Signatures and Infrastructures – az ETSI elektronikus aláírással foglalkozó munkacsoportja;

ETSI: European Telecommunications Standardization Institute – Európai Távközlési Szabványosítási Intézet;

EU: Európai Unió;

EV: extended validation – tanúsítványok egy biztonsági osztálya, elsősorban webszerver tanúsítványokkal kapcsolatban;

FIPS: federal information processing standard – amerikai szabványok egy csoportja;

FMH: fizetési meghagyás;

FTP: file transfer protocol;

GF: Galois field – Galois test;

GKM: Gazdasági és Közlekedési Minisztérium;

GNU: GNU is not Unix – szabad szoftverekkel kapcsolatos rekurzív rövidítés;

GPG: GNU Privacy Guard – a PGP szabad változata;

GSC: Government Smart Card;

HSM: hardware security module – kriptográfiai hardvermodul;

HSZ: hitelesítés-szolgáltató;

HTML: hypertext markup language – a weboldalak nyelve;

HTTP: hypertext transfer protocol – így kommunikál a böngésző a webszerverekkel;

HTTPS: HTTP secure – HTTP SSL-en keresztül, a HTTP biztonságos változata;

IANA: Internet Assigned Numbers Authority;

IC: integrated circuit – integrált áramkör;

ICAO: International Civil Aviation Organization;

ID: identifier;

IETF: Internet Engineering Task Force – ez a nemzetközi szabványosítási testület bocsátja ki az RFC-ket;

IFP: integer factorization problem – az egész számok törzstényezőkre bontásának problémája, erre épül az RSA;

IHM: Informatikai és Hírközlési Minisztérium;

IP: internet protocol;

IPSEC: IP security – virtuális magánhálózatok kialakítására használt hálózatbiztonsági megoldás;

ISO: international standardization organization;

ITSEC: IT security – biztonsági termékek tanúsításához használt szempontrendszer;

ITU: International Telecommunication Union;

JPG: Joint Picture Expert Group – képfájl-formátum, amely a formátumot kidolgozó munkacsoportról kapta a nevét;

KAP: Kihelyezett Adatküldő Program – a PSZÁF által működtetett rendszer;

KEKKH: Közigazgatási és Elektronikus Közszolgáltatások Központi Hivatala;

KGYHSZ: Közigazgatási Gyökér Hitelesítés Szolgáltató;

LCP: Lightweight Certification Policy – ETSI hitelesítési rend;

LDAP: lightweight directory access protocol;

LTANS: Long-term Archive and Notary Services – egy archiválással foglalkozó IETF munkacsoport neve;

LZW: Lempel-Ziv-Welch – tömörítési algoritmus;

MAC: message authentication code – kriptográfiai ellenőrzőösszeg;

MD5: message digest 5 – egy hash függvény neve;

MELASZ: Magyar Elektronikus Aláírás Szövetség;

MIME: Multipurpose Internet Mail Extensions;

NATO: North Atlantic Treaty Organization;

NCP: Normalized Certification Policy – ETSI hitelesítési rend;

NHH: Nemzeti Hírközlési Hatóság – az NMHH jogelődje;

NIST: National Institute of Standardization – amerikai szabványosítási testület;

NR: non-repudiation – letagadhatatlanság;

NSA: National Security Agency;

NSS: Network Security Services;

OCF: Open Card Framework;

OCSP: online certificate status protocol – online tanúsítvány-állapot protokoll;

ODF: Open Document Format – az OpenOffice által is használt dokumentum-formátum;

OID: object identifier – számokból és pontokból álló, globálisan egyedi azonosító;

OU: organization unit – az X.509 tanúsítványban szereplő megnevezés egyik eleme;

OV: organization validated – olyan (webszerver) tanúsítvány, amely esetén a hitelesítés-szolgáltató ellenőrizte a tanúsítványban feltüntetett szervezet kilétét;

PAdES: PDF advanced electronic signature;

PDF: Portable Document Format – az Adobe által kifejlesztett dokumentum-formátum;

PFX: Personal Information Exchange – fájlformátum, amely tanúsítványt és magánkulcsot is tartalmazhat;

PGP: Pretty Good Privacy – egy nyilvános kulcsú kriptográfiát megvalósító megoldás, illetve szoftver;

PIN: personal identification number;

PKC: public key certificate – nyilvános kulcsú tanúsítvány;

PKCS: public key cryptography standard – az RSA nevéhez fűződő szabványcsalád;

PKI: public key infrastructure – nyilvános kulcsú infrastruktúra;

PKIX: Public Key Infrastructure for X.509 Certificates – IETF munkacsoport;

PM: Pénzügyminisztérium;

PNG: portable network graphics – képfájl-formátum;

PP: protection profile – védelmi profil (Common Criteria fogalom);

PSS: probabilistic signature schme – véletlen padding használata aláíráskor;

QCA: qualified CA – minősített hitelesítés-szolgáltató;

QCP: Qualified Certificate Policy;

QTSA: qualified TSA – minősített időbélyegzés-szolgáltató;

RA: registration authority – regisztrációs szervezet, aki pl. az ügyfelek azonosítását végzi egy hitelesítés-szolgáltató számára;

RAM: random access memory;

RFC: request for comments – az IETF ilyen néven teszi közzé szabványjellegű specifikációit;

RSA: Rivest - Shamir - Adleman – az RSA nyilvános kulcsú kriptográfiai algoritmus a három feltalálójáról kapta nevét;

SAN: subject alternative names – a tanúsítvány alanyának alternatív nevei, mező az X.509 tanúsítványban;

SAP: Systeme, Anwendungen, Produkte in der Datenverarbeitung – német vállalat;

SHA: secure hash algorithm – az NSA által kifejlesztett lenyomatképző algoritmus;

SMS: Short Message Service;

SN: serial number – sorozatszám;

SSCD: secure signature creation device – biztonságos aláírás-létrehozó eszköz, BALE;

SSH: secure shell – biztonságos távoli bejelentkezési mód;

SSL: secure socket layer – egy kriptográfiai protokoll;

ST: security target – biztonsági előirányzat (Common Criteria fogalom);

TAJ: társadalombiztosítási azonosító jel;

TLS: transport layer security – az SSL újabb elnevezése;

TR: technical specification – ETSI specifikációk egy típusa;

TS: time stamp;

TTP: trusted third party – megbízható harmadik fél;

UCC: Unified Communication Certificates;

URI: uniform resource identifier – pl. egy webes cím;

URL: uniform resource location – az URI-hoz hasonló, de kicsit szűkebb fogalom;

USB: universal serial bus – számítógép-perifériák egy illesztési módja;

UTC: universal time, coordinated;

UTF: Unicode Transformation Format;

VOIP: voice over IP – pl. internetes telefonálási megoldások;

VPN: virtual private network – virtuális magánhálózat;

WORM: write once media – egyszer írható média;

XAdES: XML advanced electronic signature – az ETSI által kidolgozott aláírás-formátum, amely az XMLDSIG-re épül, így XML alapú;

XML: extended markup language;

XMLDSIG: XML digital signature – egy XML alapú aláírás-formátum elnevezése;

XSLT: XML style sheet – egy XML állomány megjelenítésére vonatkozó információkat tartalmazó másik XML állomány;