Diplomamunka

Farkas Róbert

Debrecen
2007

Debreceni Egyetem
Informatikai Kar
Számítógéptudományi Tanszék

Nyilvános Kulcsú Infrastruktúrák

Témavezető:
Prof. Pethő Attila
egyetemi tanár
Az Informatikai Kar dékánja

Készítette:
Farkas Róbert
Programtervező matematikus
hallgató

Debrecen
2007

Tartalomjegyzék

1. Bevezetés

A személyi számítógépek óriási fejlődésével és a szélessávú Internet elterjedésével mára mindenki, akinek van saját gépe, böngészik, folytat levelezést, készíti el adóbevallását, esetleg vásárol kényelmesen ülve otthon a karosszékéből, miközben teát szürcsölget. Felgyorsult és egyre inkább automatizált világunkban ma már elképzelhetetlen az élet számítógépek nélkül. Virágzik az e-business, az e-government, és az e-buy. Hálózatba kötött számítógépek milliói szolgáltatják a mindennapi életünk részét alkotó tevékenységeket. Azonban felhasználóknak csak egy csekély része ügyel arra, hogy a kommunikáció, amit folytat bizalmas és biztonságos legyen.

Kérdezhetnénk, miért kellene a levelezéseimet, adataimat számolási kapacitás költséges kódolásokkal védenem. Ami természetesen rengeteg időbe kerül és bonyolult is. A válasz kézenfekvő. Ma a legtöbb kommunikáció elektronikusan folyik. Az USA területén 107 milliárd levelet kézbesítettek, de ez a szám nem vetekedhet az e-mailek 4 trillióra becsült számával. A legintimebb beszélgetésünket is lehallgathatják a tudtunk nélkül. Az e-maileket észrevétlenül és automatikusan lehet szűrni bizonyos kulcsszavakra, és ugye mindig akad valami érdekes. A postai levelezéseinket sem nyílt levelezőlapon folytatjuk, hanem egy zárt borítékba rakjuk a levelet és így küldjük el. Ez egy természetes igényünk, hogy privát levelezéseinket vagy bizalmas adatainkat ne olvassa el más. Ez a magánélethez való jog egy része.

Igen könnyű elektronikus levelet hamisítani, így sohasem lehetünk teljesen bizonyosak abban, hogy a kapott üzenet csakugyan a feladóként feltüntetett személytől származik. Továbbá mindennapi munkánk, ügyeink intézése során különböző

elektronikus dokumentumokat továbbítunk üzleti partnereink felé. Könnyű elképzelni, miféle bonyodalmak származhatnak abból, ha valaki engedély nélkül módosítja a továbbított állományokat. Ma már a szerződések és egyéb hivatalos dokumentumok többsége is számítógépen készül, s egyre inkább megnyílik az elektronikus ügyintézés lehetősége.

A felsorolt problémákra a megoldást, a digitális világ szereplőinek hitelt érdemlő azonosítása, a fellépő adatcsere esetleges titkosítása, s az átadott információk változatlanságának védelme biztosítja. Ezt a megoldást a nyilvános kulcsú infrastruktúra (eredeti, angol elnevezésének elterjedten használt rövidítésével PKI) jelenti.

A mai személyi számítógépek rendelkeznek olyan számítási kapacitásokkal, amelyek nagy része kihasználatlan. Rengeteg olyan kis ingyenes alkalmazás, program van, amik kiteszteltek, megbízhatóak, és a célnak tökéletesen megfelel.

Az eszközök tehát rendelkezésre állnak, és még pénzt sem kell költeni rájuk. Így csak a társadalomnak kell rászánnia magát, hogy a technológia széles körben elterjedjen. Vannak olyan üzleti szektorok, ahol már régóta alkalmazzák ezeket a technológiákat. A banki életben ugye elképzelhetetlen lenne a kommunikáció, a személyes adataink és a pénzünk kezelése megfelelő szintű, és megfelelő minőségű titkosítások, biztonsági rendszerek nélkül. Elektronikus vásárlás esetén is megtörténnek a megfelelő biztonsági lépések, hogy személyes adataink és legfőképpen a bankkártyánk adatai titokban maradjanak. Ma már Magyarországon is törvényi szintű szabályozás vonatkozik arra, hogy egy nyilvános kulcsú infrastruktúrának, milyen kritériumoknak kell eleget tennie, és mit kell, esetleg mit nem kell tudnia.

Diplomamunkám célja, hogy néhány alapfogalom definiálása és a kriptológia történelmi áttekintés után, a biztonsági szolgáltatások köréből kiemelve, bemutassam a nyilvános kulcsú infrastruktúrát. Továbbá, hogy megmutassam az algoritmusok mögött húzódó elméleti hátteret, és ezek matematikai megalapozását. Ezt azért tartom fontosnak, mert az ember addig nem nagyon bízik egy alkalmazásban és az azt használó infrastruktúrában, míg nem tudja, hogy mit is csinál az a háttérben. Szeretnék rávilágítani az ilyen rendszerek sebezhetőségeire. Ezzel is felhívva a figyelmet azokra a sarkalatos pontokra, melyeket egy ilyen alkalmazás tervezésénél, kivitelezésénél nagyon meg kell fontolni, és csak megfelelő körültekintéssel szabad megvalósítani.

Diplomamunkám részét képezi még egy program elkészítése, amely egy ilyen infrastruktúrát valósít meg persze nagyon sematizált módon. Ennek a programnak a bemutatására és használatára szentelem az utolsó fejezetet. Melyben mindezek mellett az elkövetett implementációs hibáim is elemzésére kerülnek.

2. Kriptográfiai alapok

A huszadik század második feléig a kriptográfiát kizárólag katonai és diplomáciai alkalmazásokban használták. A kriptográfia ebben az időszakban egyet jelentett a titkosírással és rejtjelezéssel. Olyan tényeket és adatokat kezeltek melyek nyilvánosságra kerülése veszélyeztette az állam érdekeit. Napjainkban a kriptográfia már megjelent az üzleti életben is. A titkosság mellett fontossá vált az integritásvédelem, a hitelesítés, a letagadhatatlanság és még sorolhatnánk. A nyilvános kutatások a 1970-es évek közepétől egyre növekvő intenzitással folynak. A kriptográfia mára már külön tudományterületté fejlődött.

2.1. Tudományágak

Mi is ez a varázslatos kriptográfia? A titkos és védett kommunikáció tudománya a kriptológia. Ennek van két egymással állandóan rivalizáló ága a kriptográfia és a kriptoanalízis. A kriptográfia azon elvek, és gyakorlati technikák tanulmányozásával foglalkozik, melyek lehetővé teszik az üzenetek, adatok, információk olyan módon történő tárolását és továbbítását, hogy ahhoz csak a jogosult fél férjen hozzá. Biztosítja a hitelességet, a titkosságot és a védettséget. Eszközei a matematikai módszereket alkalmazó algoritmusok, melyek pontos használatáról a kriptográfiai protokollok adnak útmutatást. A kriptoanalízis területe viszont pont azzal foglalkozik, hogy az egyes titkosított üzenetekből miként fejthető vissza az eredeti üzenet. Többnyire illetéktelen megfejtésekre, feltörésekre irányuló eljárásokat készítenek a kriptoanalízist művelő emberek. A kriptoanalízis eredményei azonban visszahatnak a védelmi rendszerek fejlődésére, hiszen azokat úgy kell elkészíteni, hogy az ismert támadási módszereknek ellenálljanak.

2.2. Személyek

A kommunikációban legalább két személy vesz részt. Aki az üzenetet, adatot el karja küldeni küldőnek, feladónak (sender) nevezzük. Aki megkapja az információt az a fogadó vagy címzett (receiver). Ezen kívül létezik még legalább egy harmadik személy is, aki a támadó (intruder, attacker, adviser). Ezt a betolakodót megkülönböztetjük az alapján, hogy mit is tesz. A passzív betolakodó az információs csatornát hallgatja le és így próbál információt szerezni. Ő elköveti a passzív támadást (passive attack), amelynek az a lényege, hogy észrevétlenül hallgatózik, nem változtatja meg a kommunikációt. Az a támadó, aki már a kommunikáció tartalmát is megváltoztatja, és nem tartja be a protokoll szabályait, aktív támadást (active attack) követ el. Előfordulhat az az eset is, hogy a támadó a protokollunk egyik szereplője. Ekkor a támadót csalónak (cheater) szokás nevezni. Itt szeretném megemlíteni a hackert és a crackert. Ez az a két megnevezés melyet mindig összekevernek. A hacker jól felkészült szakember, aki ismeri a biztonsági protokollokat, az operációs rendszereket és azok gyenge pontjait. Kihívást keres, és nem akar kárt okozni, behatolásaira felhívja a figyelmet. A hackerek egy része ebből él, mint biztonsági tanácsadó. A cracker hasonló a hackerhez, de a tudását a rombolásra irányítja. Fő feladatának tekinti az anarchia kialakítását és a rendszer megbénítását.

2.3. Elnevezések

Azt a szöveget, melyet a feladó el szeretne juttatni a fogadóhoz nyílt szövegnek (plaintext, cleartext) nevezzük. Az összes lehetséges nyílt szöveg alkotja a nyílt szövegek terét (plaintext space). Egy konkrét nyílt szövegből titkosítással (encryption, enciphering) áll elő a küldött szöveg. Ez a kriptoszöveg (ciphertext), amely a nyílt szöveg értelmét, vagy

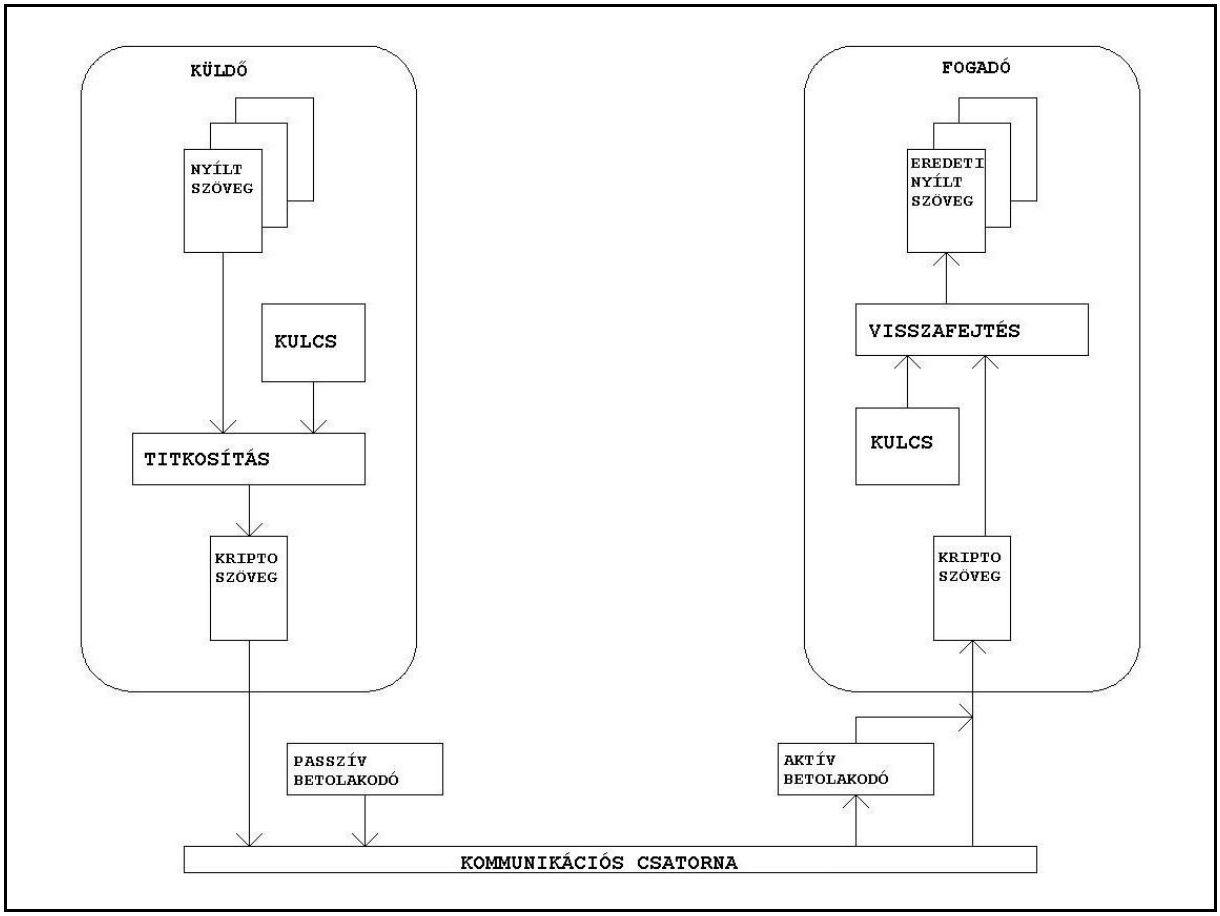
annak egy jellemző tulajdonságát elrejti. Az összes lehetséges kriptoszöveg halmaza alkotja a kriptoszövegek terét (ciphertext space).

A titkosításhoz egy algoritmust (cipher) használunk. Az algoritmus matematikai műveleteket használ, hogy a nyílt szöveget értelmezhetetlen adathalmazzá változtassa. Ez a folyamat a kódolás (encoding). Az algoritmusok különböznek abban, hogy bokk titkosítást (block cipher), vagy folyó titkosítást (stream cipher) használnak. Az első esetben rögzített hosszú részekre vágja a nyílt szöveget, és ezekre a részekre hajtja végre az algoritmus a lépéseit. A második esetben folyamatosan, bitenként alkalmazódik az algoritmus a kiinduló szövegre. A kódolt és így titkos szöveget egy kommunikációs csatornán juttatjuk el a feladónak. Ebből a kódolt szövegből elő kell állítani a nyílt szöveget, ezt a titkosítás inverz műveletével, a visszafejtéssel (decryption) tesszük meg. Az algoritmus feladata most az, hogy az értelmezhetetlen adathalmazból matematikai műveletek segítségével visszaállítsa (decoding) az eredeti nyílt szöveget.

2.4. Kulcsok

A megfelelő kriptoszöveg előállításához az algoritmuson kívül szükség van egy kulcsra (key) is. Alapvető elvárás egy titkosító módszerrel szemben, hogy az adott információ halmazból úgy készítsen egy másik információ halmazt, hogy az algoritmus ismeretében sem lehetséges a titkosítás és visszafejtés a megfelelő titkosító és visszafejtő kulcs nélkül. Ha a titkos szöveget nyílt szöveggé alakítjuk jogtalanul, azaz a megfelelő kulcs nélkül, akkor elkövetjük a feltörést. A lehetséges kulcsok halmazát kulcstérnek nevezzük. Az a fajta feltörési mód, amelyben az összes lehetséges kulcs

kipróbálásra kerül, a primitív próbálgatás (brute force attack), vagy más néven a nyers erő módszere. Azok az algoritmusok, melyek a titkosításhoz és a visszafejtéshez is ugyanazt a kulcsot használják, a szimmetrikus algoritmusok (symmetric algorithm). Ebből következik, hogy azok az algoritmusok, melyek más-más kulcsokat használnak a titkosításhoz és a visszafejtéshez, azok az aszimmetrikus (asymmetric) vagy nyilvános kulcsú (public key algorithm) algoritmusok. Ilyenkor a titkosítást a nyilvános kulccsal (public key), a visszafejtést a titkos kulccsal (private key) végezzük. Ennek a fejezetnek a lezárásaként tekintsük át még egyszer az alapfogalmakat egy kis ábra segítségével.



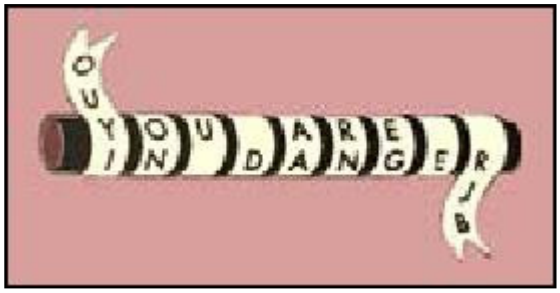
1. ábra – A kommunikáció modellje

3. Történelmi áttekintés

Minden, napjainkban jelentős tudományágnak, visszanyúlnak a gyökerei, már az ókori görögök korába. Nincs ezzel másképp a kriptográfia sem. Mi sem bizonyítja ezt jobban, mint maga tudományág neve. A görög kryptos szó annyit tesz: titok, a graphos szó pedig: írást jelent.

3.1. A spártaiak módszere

A spártaiak olyan titkosítást találtak fel és alkalmaztak melynek alapja egy rúd. Ez a módszer az üzenet betűinek átrendezésén alapszik. A kulcs maga a rúd, azaz annak az átmérője. A titkosításhoz szűkség volt egy vékony csík papiruszra, melyet egy rúdra tekertek, úgy hogy minden tekerés szorosan illeszkedjen egymáshoz. Ekkor egy olyan egységes felületet kaptak, amire ezek után fel tudták írni az üzenetet. Aztán letekerve a papiruszt a rúdról, egy értelmetlen szöveg keletkezett. Az üzenetet csak akkor lehetett ismét elolvasni, ha a megfelelő átmérőjű rúdra megint feltekerték.



2. ábra – Spártaiak rúdja

3.2. Caesar módszer

Egy másik híres titkosítás, egy nagy római hadvezérről Julius Caesarról kapta a nevét. Ez egy egyszerű helyettesítő üzenet elrejtési eljárás volt. A módszer lényege, hogy vesszük a kiinduló ABC-t, és ezt helyettesítjük egy másik ABC-vel. A betűk sorrendjét megtartjuk, azaz csak eltoljuk az ABC-t egy bizonyos hosszúsággal. Caesar ezt a hosszt 3-nak választotta, de működik az algoritmus tetszőleges k hosszúságú eltolással

is. A nyílt szövegünkben minden karaktert szisztematikusan kicserélünk az új ABC-ben neki megfelelő karakterre.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W

3. ábra – A kiinduló és az eltolt ABC

Fontos, hogy az így titkosított üzenetben a betűk helyzete, a szavakban elfoglalt pozíciójuk és az egymáshoz való viszonyuk nem változik meg. Ezért ez a titkosítás viszonylag egyszerűen, próbálgatással is könnyen feltörhető. Ez a kulcstér szűkösségéből és a rengeteg szabályosságból következik. Ha megfejtünk egyetlen egy betűt, akkor az ABC rendezettsége miatt az összes többi betűt is megkapjuk.

Javíthatunk az algoritmuson, ha a második ABC-t nem eltolással állítjuk elő, hanem összekeverjük a betűket. Azaz egy véletlen sorrendben írjuk fel az ABC betűit és így feleltetjük meg a betűket kölcsönösen egymásnak. Tanácsos továbbá kihagyni a szóközöket és az írásjeleket a nyílt szövegből, mivel ezek jelentős segítséget nyújtanak a támadónak. Gondoljunk bele, hogy a névelők kötött helye és formája – és ezek kriptoszövegbeli felismerése – mekkora segítség lehet. Az ABC két-három betűje könnyen megállapíthatóvá válik általuk. Mert még ebben a feljavított Caesar módszerben sem változik meg a betűk szavakban betöltött pozíciója és az egymáshoz viszonyított helyzetük sem. Azaz a kriptoszöveg bizonyos részeiből következtéseket vonhatunk le a nyílt üzenetre vonatkozólag. Az ilyen jellegű információk elfedésére találták ki a betűk betűcsoportokkal való helyettesítését. Ennek a módszernek viszont meg kell említeni azt a hátrányát, hogy a titkosított üzenet hossza

annyiszorosára nő, ahány betűvel helyettesítünk egy betűt. Ezeket a betűcsoportokat egy táblázatos formában tudjuk szemléletesen megjeleníteni. A megfejtésre és kódolásra egyazon táblázatot kell használni, tehát a táblázat a kódoló és dekódoló kulcs is.

	AA	AB	AC	BA	BB	BC	CA	CB	CC
A	f	u	p	z	t	o	e	r	c
B	v	b	i	d	x	h	k	n	--
C	s	j	m	a	l	q	y	g	--

4. ábra – Egy lehetséges három jelen alapuló csoportosítás.

A módszer gyengéje, hogy nem használjuk fel a kriptoszövegben az összes lehetséges jelkombinációt. Mivel nincs annyi betűnk az ABC-ben, így nem szükséges az összes kombináció. Ekkor ebből megállapítható, hogy egy betűt hány betűcsoport alkot. Ebből az információból meghatározhatók a betűk valószínűségi határai és így már lehetséges megpróbálkozni valamilyen betűeloszlást figyelembe vevő visszafejtési módszerrel. Az eddig bemutatott helyettesítő rendszerek gyengéje az, hogy minden betűt mindig ugyanazzal a másik jellel, betűvel, vagy betűcsoporttal helyettesítenek és ez kiváló támpont a feltöréssel próbálkozóknak.

3.3. Vigenere titkosítás

A következő titkosítási módszer egy francia diplomatáról, Blaise de Vigenere-ről kapta a nevét, aki az 1560-as években kidolgozott egy olyan módszert, amiben nem egy darab ABC-t használunk, hanem többet. Azaz a monoalphabetic substitution helyett polyalphabetic substitution valósul meg. Az ötlete az volt, hogy a betűket más és más betűkkel kell helyettesíteni, mondjuk egy kulcsszó függvényében. Két megvalósítási folyamata ismert, a numerikus módszerrel és a táblázatos módszerrel történő titkosítás.

A numerikus módszer esetén az ABC betűihez számokat rendelünk. Ezután kiválasztunk egy megfelelő kulcsszót, és a nyílt szöveg alá írjuk, egymás után amennyiszer szükséges. Ekkor a nyílt szöveg betűjének numerikus értékét és az alá kerülő kulcsszó betűjének az értékét összeadjuk. Az így kapott értékből visszaszármaztatjuk az ABC megfelelő betűjét. Így előáll egy olyan szöveg, melyben az azonos betűk más-más betűvel lesznek elfedve. Megszűnnek a nyelvre jellemző sajátosságok, és a betűeloszlás összekuszálódik. Így a gyakoriságelemzések használhatatlanná válnak. Megállapítható, hogy amilyen hosszú a kulcsszavunk annyiféle ABC-t használunk a titkosításhoz. A titkos üzenet megfejtése is hasonló szisztéma szerint zajlik, mint a titkosítás, csak az összeadás műveletét felváltja a kivonás. A kriptoszöveg alá felírjuk a kulcsszavunkat annyiszor ahányszor szükséges, majd egy-egy betű numerikus értékéből kivonjuk az alatta található kulcsszó betűjének numerikus értékét. Az így kapott számértékeket visszaalakítjuk betűvé a megfelelő ABC szerint. Természetes az itt említett kivonás és összeadás az ABC-hez rendelt számok moduló osztályában zajlik.

A táblázatos módszer esetén egy táblázat szolgáltatja a kulcsot mind a titkosításhoz, mind a visszafejtéshez. A táblázat kitöltése nagyon egyszerű. Az első sorába az ABC betűi kerülnek sorban. A második sorába az ABC betűi eggyel balra tolva. A harmadik sorában az ABC betűi kettővel balra tolva, és ez így folytatódik, míg tart az ABC. A megfelelő kulcsszó kiválasztását itt is meg kell tennünk. Akkor a titkosítás a következő. A kulcsszó aktuális betűjét megkeressük az első oszlopban, a titkosítandó betűt az első sorban. Az oszlop és a sor kereszteződésében lévő betű adja a megfelelő helyettesítést. A megfejtés a következő képpen valósul meg. Megkeressük a kulcsszó betűjét az első oszlopban.

Az így kijelölt sorban megkeressük, a titkosított szöveg aktuális betűjét. Ez a betű kijelöl egy oszlopot. Ennek az oszlopnak a tetején van a nyílt szöveg megfelelő betűje.

3.4. Kasiski algoritmus

A Vigenere titkosítást sokáig feltörhetetlennek hitték, de 1863-ban Friedrich Wilhelm Kasiski kitalált és publikált egy módszert, amellyel sikerült feltörnie a kódolt szöveget. Mint már említettem a használt ABC-k darabszámát a kulcsszó hossza határozza meg. Ha például egy öt betűs kulcsszót használunk, akkor minden betűnek ötféle alakja lehet. De ez a szabályszerűség igaz minden betűből alkotott szóra is. Ha egy szó sokszor fordul elő a nyílt szövegben, akkor nagy valószínűséggel egy vagy több rejtjeles alakja ismétlődni fog a kriptoszövegben. A kulcsszóhossz egész számú többszöröse által meghatározott távolságban lévő azonos nyílt szövegek, azonos kódszöveget adnak. Ezek az ismétlődések jó támpontot adnak a feltöréshez. A feltöréshez szükségünk lesz egy nyelvnek megfelelő referencia táblára. Ez a tábla a betűk gyakoriságának előfordulását tartalmazza a nyílt szövegben. Első lépésben meg kell próbálnunk az ismétlődések által meghatározni a kulcsszó hosszát. Ha több lehetséges hossz is van, akkor sorban ki kell majd próbálni mindre a következőket. Emeljük ki a szövegünkből a megállapított hossznak megfelelő karaktereket. Az így kapott füzérben nézzük meg a karakterek előfordulási gyakoriságát és vessük össze ezt a referencia táblánkkal. Ez alapján következtetéseket lehet levonni, hogy mely kódolt betűnek mely eredeti betű felelhet meg. És a kulcsszó első betűjére is fény derül. Ezután ismételjük a műveleteket, eggyel jobbra tolva a kiindulási pontunkat, az eredeti kriptoszövegben. Annyiszor ismétlünk amennyi a feltételezett kulcsszó hossza. Minden ismétlésben a kulcsszó egy újabb betűjét tudjuk meghatározni, és a kódolt betűknek is

megtaláljuk az eredeti szövegbeli párjukat. Ha a legvégén egy értelmes szöveget kapunk megfejtésnek, akkor kiinduláskor jól állapítottuk meg a kódszó hosszát. Ha többé-kevésbé értelmes a szöveg, akkor a kódszó hossza jó, csak valamely betűpozíciónál rosszul következtettünk a referenciatábla és az előfordulás gyakorisága alapján. Ha teljesen értelmetlen szöveget kapunk, akkor egy másik kódszó hosszal lehet elkezdeni az egész eljárást.

3.5. Enigma

Az időben ugorva eljutunk a második világháborúig, ahol megtaláljuk a gépi titkosítást és annak is a leghíresebb képviselőjét, az Enigmát. Ezt az eszközt a németek fejlesztették ki és használták az üzeneteik titkosítására. Működési alapelve a rotor-elv. Egy rotor bemenő oldalán érkező 'A' elektromos jelet mondjuk 'H' elektromos jellé alakít. Hogy mi is legyen a leképezés egy-egy ilyen tárcsán, azt be lehetett állítani. Több tárcsát egymásmellé téve (először 3db, majd később 5db) és sorba kötve őket tovább bonyolítható a helyettesítés. Ezután még tovább bonyolítva a dolgot ez első tárcsa minden egyes jel után ugrik egyet, a második tárcsa viszont csak akkor ugrik, ha ez első tárcsa már körbeért. A harmadik tárcsa meg csak akkor ugrik egyet, ha a második tárcsa is körbeért. Amit így megvalósított az Enigma, az egy egyábécés helyettesítés volt, de azt olyan bonyolult módon tette, hogy megfejthetetlennek hitték. A titkosítás szimmetrikus volt, és ugyanazon gép egyaránt alkalmas volt titkosításra és visszafejtésre. A géphez tartozott még egy 26 karakteres billentyűzet, egy kapcsolórendszer, és egy 26 lámpás kijelző. A kapcsolórendszer a kezdeti keverést biztosította, a lámpák a kódolt és visszafejtett betűket szolgáltatták felvillanásukkal. A gép a kapcsolásoknak köszönhetően soha nem képezett le egy betűt önmagára, és a

keverést biztosító kapcsolótábla sohasem cserélt meg két szomszédos betűt. Ezek a speciális tulajdonságok lehetőséget adtak a kódfejtőknek, hogy nagyon sok kombinációt kizárjanak. A feltörést nagyban segítette, hogy a lengyel titkosszolgálat néhány napig hozzájutott egy modellhez, amit lemásoltak. A franciák dokumentumokat szereztek egy német árulótól az enigmával kapcsolatban. A feltörést a lengyel kriptográfusok kezdték meg. Rájöttek arra, hogy az üzenetek első három betűje a tárcsák kezdőbetűit (kezdő pozícióit) azonosítja. Sőt ezt a három betűt a biztonság kedvéért még egyszer megismétlik az üzenet elején. Ezt a hat betűt a napi parancs szerint titkosították, de az üzeneteket már ezekkel a viszonykulcsként alkalmazott rotorbeállításokkal. Ezután építettek egy a forgótárcsák lehetséges sorrendjét (permutációját) kitaláló gépet, melyet Bombe névvel illettek. Amikor a három tárcsa mellé még két újabb társult, akkor a lengyeleknek elfogyott a rendelkezésre álló számítási kapacitásuk, ezért megosztották eredményeiket az angol kollégákkal. Az angoloknál Alain Turing továbbfejlesztette a gépet és a feltörés továbbra is sikeresen folyt. Így a szövetséges erők gond nélkül tudták olvasni a német kódolt üzeneteket. A számítógépek óriási fejlődésével azonban az Enigma rendszerű gépek elvesztették versenyképességüket. Végül az ilyen gépek használatának a célja már nem is titkosítás volt, csak az időnyerés, amit az a késleltetés jelentett, amíg az ellenség megfejti az üzenetet.

5.ábra – A 3 és 5 tárcsás enigma

3.6. Generációk

Ennek a fejezetnek a végén, a fejlődéseket sorba véve megállapíthatunk úgynevezett generációkat. Az első generációt a történelem kezdetétől számíthatjuk a XVI. századik. Ebben a generációban megállapítható, hogy elsősorban egyábécés helyettesítő módszereket használtak. A második generáció a XVI. századtól a XX. századig számolható, amikor a többábécés helyettesítő módszerek hódítottak teret. A harmadik generáció a XX. század elejétől eredeztethető. Ekkor hódítottak teret maguknak a technikai újítások. Lehetővé vált a mechanikus és elektromechanikus eszközök fejlesztése és használata. Ezek is több ábécén alapuló titkosítást használtak, de az ABC-k száma gigászi méretekig növekedett. A negyedik generáció a XX. század végétől napjainkig, a XXI. századig tart és még jelenleg is zajlik. Egyre több produkciós kódoló jelenik meg és ezzel egyetemben robbanásszerűen nő a használt kulcstér mérete. Óriási méreteket ölt az elektronika fejlődése és ezzel együtt a rendelkezésre álló számítási kapacitás. Megfelelő számítási kapacitással az algoritmusok feltörhetők ugyan, de a titkosító algoritmus architektúrájának megtartás mellett, a kulcs méretének növelésével a biztonsági szint a végtelenségig fokozható. És ezután következik a nem is olyan távoli jövő és az ötödik generáció. A kvantum elvű titkosítás, amely kvantum számítógépek segítségével megy majd végbe és a kvantumfizikát és annak törvényeit hívja segítségül, ennek megvalósítására.

4. Aszimmetrikus rendszerekről

A nyilvános kulcsú vagy aszimmetrikus kriptográfia alapelvét Whitfried Diffie és Martin Hellman [13] fogalmazták meg. Az első aszimmetrikus titkosítási algoritmust R. C. Merkle és Martin Hellman [14] publikálta és az a hátizsákprobléma egy speciális estén alapult. Erről Adi Shamir [15] megmutatta, hogy könnyen feltörhető. A Merkle–Hellman algoritmussal csaknem egy időben publikálta Ronald Rivest, Adi Shamir és Leonard Adleman [16] az RSA algoritmust, amelyet máig is biztonságosnak tartanak és széles körben elterjedt. De erről majd később az ötödik fejezetben lesz szó. Érdekességként megjegyezném, hogy bár nem bizonyították, de az angolok azt állítják, hogy néhány évvel ezek előtt 1970-ben kitaláltak egy hasonló módszert. Ezen felfedezést akkor katonai titokként kezelték, ezért nem publikálták.

Ezen rendszerek létjogosultságát az adta, hogy a szimmetrikus rendszereknél ugyanazt a kulcsot használjuk titkosításhoz és megfejtéshez is. Ez azt jelenti, hogy a kulcsot ismernie kell mindegyik kommunikációs félnek még a kommunikáció előtt. Azaz valahogy meg kell egymással osztaniuk a kulcsokat egy titkos csatornán. De ha van egy ilyen titkos csatorna, akkor meg minek az üzenetek titkosítása kérdezhetnénk. Mivel egy ilyen biztonságos csatorna csak nagyon rövid ideig létezik, ha egyáltalán létezik, a kulcsok cseréje nagyon körülményes, vagy pedig nem kellően biztonságos. Így marad a kulcsok egy nem biztonságos úton történő cseréje, vagy a személyes találkozás. A személyes találkozáskor a kulcsokat kiválasztják és kicserélik. Ez viszont nem éppen kellemes és nem is mindig kivitelezhető folyamat.

Ezzel szemben az aszimmetrikus rejtjelezés alapötlete az, hogy a titkosítás folyamatát elválasztja a megfejtéstől, és olyan algoritmust használ, ahol a titkosító kulcs nem azonos a megfejtéshez használt kulccsal. Tehát a nyilvános kulcsú rendszerekben a résztvevők két kulcsot birtokolnak. Az egyik a nyilvános kulcs (public key), a másik a titkos kulcs (private key). Ezek a kulcsok egymással szoros összefüggésben vannak, azaz a nyilvános kulccsal kódolt szöveget, a titkos kulccsal lehet megfejteni. Néhány rendszerben ez az eljárás megfordítható, de mindegyikben. Felfogható úgy az egész, mint egy láda zárja, amit bezárni a nyilvános kulccsal, kinyitni meg csak a titkos kulccsal lehet. A kulcsok neve is mutatja a funkciójukat. A nyilvános kulcsot nyugodtan lehet publikálni, hirdetni, és terjeszteni, azaz itt már nem probléma, ha nem megfelelően biztonságos csatornán lesz elküldve a kommunikációs partnereknek. Ugyanakkor biztosítani kell, hogy a nyilvános kulcs tulajdonosát azonosíthassuk. Lényegében ez a fő feladat. A titkos kulcsot viszont szigorúan őrizni kell, és titokban kell tartani, mert a nyilvános kulccsal rejtjelezett üzenetet csak az tudja elolvasni, akinek a titkos kulcs a birtokában van. Rajta kívül senki más nem férhet az üzenet tartalmához, még a feladó sem.

A titkos kommunikáció előtt ebben az aszimmetrikus rendszerben is szükség van előzetes megbeszélésre, mert a nyilvános kulcsot be kell szereznünk. De a szimmetrikus módszerekkel ellentétben eljutottunk odáig, hogy ezt a kulcsot nyugodtan küldhetjük nem kellően biztonságos csatornán is. A könnyebb kommunikáció érdekében és a kulcs tulajdonságainak köszönhetően lehetőség nyílik ezeket a nyilvános kulcsokat egy osztott adatbázisban tárolni, mint egy nagy nyilvános elektronikus telefonkönyv. Csak telefonszámok helyett az adott személy neve mellett annak a nyilvános kulcsa található. Ha

valakinek levelet szeretnénk írni, akkor csak kikeressük ebből az adatbázisból a neve mellett szereplő nyilvános kulcsát és kódoljuk az üzenetet. Ezután elküldjük a kódolt szöveget, és biztosak lehetünk benne, hogy ezt a privát levelet csak ő tudja majd elolvasni a titkos kulcsa segítségével. Ha valakivel gyakran kommunikálunk, és nem szeretnénk mindig az általában lassú szerverről a kulcsot lekérni és keresgetni, akkor lehetőség van arra, hogy helyi kis adatbázisba tároljuk le az ilyen fontos személyek nyilvános kulcsát. Ezt nevezik kulcskarikának (keyring). Ekkor figyelni kell arra, hogy a helyi adatbázisunkban mindig az aktuális információk legyenek fellelhetők, azaz szinkronban legyünk a nagy adatbázissal a kulcsok hitelességére és érvényességére vonatkozóan. Ezzel el is érkeztünk oda, hogy rendelkezésünkre áll egy nyilvános kulcsú infrastruktúra.

4.1. A digitális aláírás

Ha feltesszük, hogy a kulcsok szerepe felcserélhető az algoritmusokban, azaz a titkosító algoritmus elfogadja a megoldó kulcsot és a megfejtő algoritmus is működik a titkosító kulccsal, akkor lehetőség nyílik arra, hogy a címzett biztonsággal azonosítani tudja az üzenet küldőjét.

1) Küldés előtt a feladó a titkosító algoritmussal és saját titkos (megoldó) kulcsával kódolja az üzenetet. Ezzel olyan átalakítás megy végbe, amelyet csak ő képes elvégezni, mert a titkos kulcsot csak ő ismeri.
2) Ezután az így kapott üzenetet titkosítja a címzett nyilvános kulcsával, ezzel biztosítva, hogy a küldemény tartalmához csak a címzett férjen hozzá.
3) A fogadó az üzenet kézhezvétele után a saját titkos kulcsával és a megfejtő algoritmusával kibontja a csomagot.

4) A megkapott eredményt még nem tudja elolvasni, mert az első lépésben azt a feladó titkosította a saját titkos kulcsával. De mivel a kulcsok szerepe felcserélhető, így a feladó nyilvános kulcsával és megfejtő algoritmus segítségével visszaállítja a címzett az eredeti üzenetet.

A feladó biztos lehet benne, hogy csak a címzett képes elolvasni az üzenetet, mert a hármas lépéshez szükséges kulcsot csak a fogadó személy ismeri. A címzett bizonyosságot nyer és meggyőződik a feladó személyében, mert a négyes lépésben csak akkor tudja megfejteni az üzenetet, ha azt a feladó titkos kulcsával rejtjelezték. Ez annyit jelent, hogy a nyilvános kulccsal csak akkor nyerhető vissza az üzenet, ha azt a hozzá tartozó titkos kulccsal rejtjelezték. Ebben az értelemben használva a kulcsokat viszont nincs titkosító és megfejtő kulcs, csak nyilvános és titkos kulcs. Így elmondható, hogy a titkosítás során a nyilvános kulcsunkkal csomagolják be a nekünk szánt üzenetet, amit mi majd a titkos kulcsunkkal bontunk ki. Ha pedig szeretnénk aláírni az általunk küldött üzenetet, azaz hitelesen igazolni magunkat, akkor a titkos kulcsunkkal csomagoljuk be a küldeni szánt nyílt szöveget, amit majd a címzett az aláírás ellenőrzője a nyilvános kulcsunkkal bont ki. Egy jó (digitális) aláírásra a következő a tulajdonságok igazak.

- *Hiteles*: meggyőz arról, hogy az aláíró saját akaratából, megfontoltan írta alá a dokumentumot.
- *Hamisíthatatlan*: az aláírás egyértelműen bizonyítja, hogy az aláírótól és nem mástól származik az aláírás.
- *Nem használható fel újra*: az aláírás a dokumentum részét képezi, nem vihető át egy másik dokumentumra.

- *Az aláírt dokumentum megváltoztathatatlan*: az aláírás után nem lehet észrevétlenül változtatni a dokumentum szövegén, megőrződik annak konzisztenciája.
- *Az aláírás nem tagadható le*: az aláíró később hiába állítja, hogy az aláírás nem tőle származik, mert bizonyítható az aláíró személye.

Ez az aláírás akkor tekinthető szabályos digitális aláírásnak, ha egy időpecsét is társul hozzá. Azaz a dokumentumhoz a dátum és pontos idő automatikusan hozzáfűződik. Nagyon fontos, hogy az aláírásra használt kulcspár, vagy algoritmus, különböző legyen, a titkosításra használt kulcspártól, vagy algoritmustól. Ha azonos algoritmussal ugyanolyan kulcspárt használunk, akkor egy lehetséges támadó játszi könnyedséggel megszerezheti az üzenetet. A hagyományos aláíráshoz hasonlóan a digitális aláírás is a dokumentum hitelességét bizonyítja, és időpecséttel együtt, nemcsak a kiadójának személyét, hanem a kiadás időpontját is igazolja.

4.2. Hash függvények

Természetesen a gyakorlatban az előbb ismertetett néhány lépéses eljárást meg szokta előzni egy nagyon is fontos mozzanat. Egy újabb fontos segédeszközzel a hash függvénnyel és annak használatával egészül ki az eljárás. A hash algoritmuson olyan matematikai eljárást értünk, mely képes tetszőleges hosszúságú bájtsorozatból előállítani annak digitális ujjlenyomatát. A szemléletesség kedvéért az említett bájtsorozatot számítógépes fájlként képzeljük el (mint az esetek többségében valóban az is), az ujjlenyomatra pedig olyan számként gondolhatunk, melyet az eljárás a fájl minden egyes bitjének figyelembe vételével számít ki. A fájl mérete tetszőleges lehet, az ujjlenyomat hossza viszont mindig

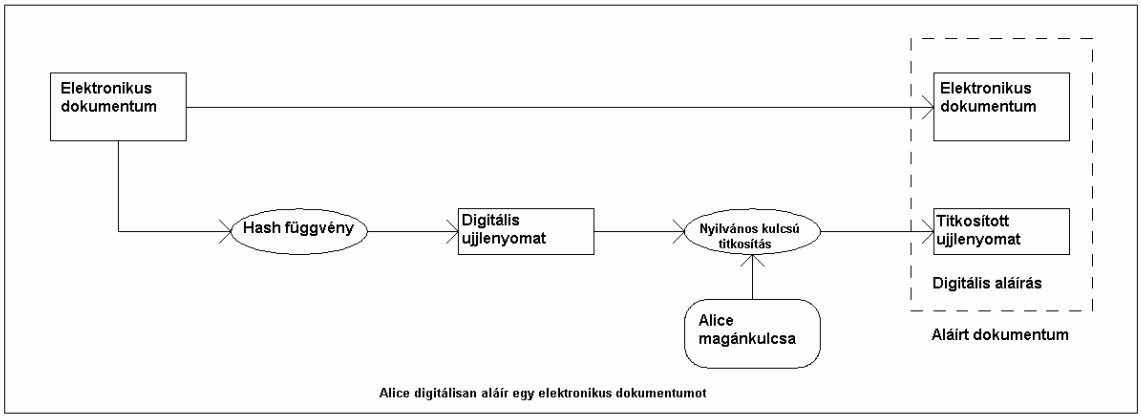
ugyanakkora, függetlenül attól, hogy az algoritmus mekkora fájlból kiindulva határozza meg. A jó hash függvényekre jellemzők a következő tulajdonságok:

- Különböző fájlokhoz különböző ujjlenyomatot készít. Két eltérő állománynak tehát nem lehet azonos ujjlenyomata. A hash függvény tehát ütközésmentes. A skatulyaelv miatt ütközésmentes hash függvény nem létezik, viszont itt gyakorlati ütközésmentességet követelünk meg, azaz ne lehessen generálni két különböző értelmes szöveget, amelyeknek a hash értéke megegyezik.
- Magából az ujjlenyomatból az eredeti adatfájl nem rekonstruálható. Azaz ez egy egyirányú függvény, melynek nem létezik inverze.
- Ha az eredeti fájl egyetlen bitje is megváltozik, akkor már más ujjlenyomat tartozik hozzá. Az ujjlenyomat így akár ellenőrző összegként is használható.

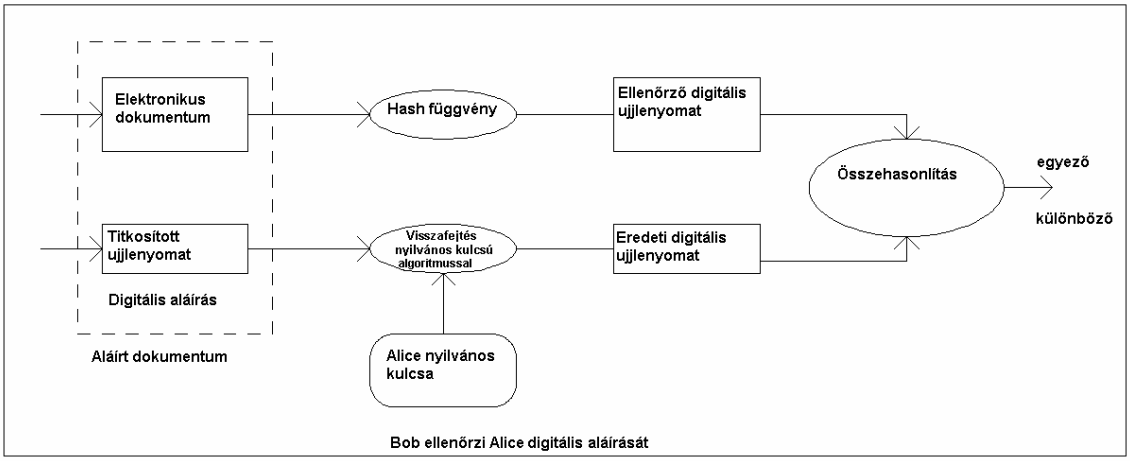
A hash függvényeket többek között azért alkalmazzák előszeretettel, mert az aszimmetrikus algoritmusok viszonylag lassúak, így nagy fájl vagy fájlok titkosítására nem célszerű őket használni. Viszont a függvénnyel előállított rövid értékekre már gyorsan és praktikusan alkalmazhatóak.

Természetesen sokféle hash algoritmus létezik, közülük azonban csak kettő terjedt el a gyakorlatban, az MD5 [22] és az SHA-1. Az MD5 128-bites, az SHA-1 [23] 160-bites ujjlenyomatot generál. Mindkét esetben tehát, a kiindulási fájl méretétől függetlenül, igen rövid ujjlenyomatok keletkeznek. A hash algoritmusok és a nyilvános kulcsú titkosítás alapfogalmaival már minden eszköz a rendelkezésünkre áll ahhoz, hogy jó digitális aláírást készíthessünk. A digitális aláírásnak többféle megközelítése

létezik, de közülük a legegyszerűbbnek és legelterjedtebbnek, az RSA algoritmuson alapuló megoldás számít. Amelyet már fentebb ismertettünk, azaz mikor a kulcsok szerepe felcserélhető. Lássuk akkor ábrákkal szemléltetve hogyan módosul a digitális aláírásunk elkészítésének folyamata a hash függvények használatával.



6. ábra – A digitális aláírás folyamata



7. ábra – A digitális aláírás ellenőrzése

Vannak azonban más elveken nyugvó digitális aláírási módszerek is. Közülük a legismertebb az ún. Digital Signature Algorithm (Digitális Aláírás Algoritmus), röviden DSA [27] vagy a DSS [27], amivel kriptográfiai alkalmazásokban is találkozhatunk. Megalkotása során fontos tervezési szempont volt, hogy felhasználását csak a digitális aláírások

előállítására korlátozzák, aszimmetrikus kulcsú titkosításra alkalmatlan legyen.

4.3. Hitelesség, érvényesség

Az előző fejezet legvégén szóba került az érvényesség és a hitelesség. A hitelesség az aszimmetrikus rendszerek „Achilles” pontja. Biztonsági szempontból óriási jelentősége van, ezen múlik a kommunikáció tényleges titkos volta. Ezért a szerver mikor elküldi nekünk a lekérdezett nyilvános kulcsot, kódolja azt saját titkos kulcsával, amit mi a szerverhez tartozó nyilvános kulccsal tudunk dekódolni. Ez a nyilvános kulcs általában valamilyen biztonságos úton került már hozzánk és teljes biztonsággal tudjuk, hogy ahhoz a szerverhez tartozik. Így garantálni tudja nekünk a szerver, hogy a lekérdezett kulcs ne változhasson meg, valóban az eredeti tulajdonosához tartozzon. Ezzel megelőzhetünk egy olyan támadási lehetőséget, hogy a támadó befurakodjon közénk és a szerver közé. Azaz kiadja magát a szervernek, és a saját nyilvános kulcsát küldje el nekünk. Ha ez megtörténne, akkor a támadó a kimenő leveleket elfogva, mindegyiket el tudná olvasni, hiszen birtokában van a nyilvános kulcshoz tartozó titkos kulcsnak. Sőt, hogy ez a turpiszság számunkra rejtve maradjon, az eredeti üzenetet titkosíthatja a címzetthez tartozó eredeti nyilvános kulccsal, majd az így kapott kriptoszöveget elküldheti a címzettnek a mi nevünkben. Így a kommunikáció teljes lenne, mivel a címzett megkapja a levelet, és mindent rendben talál. Mi idővel kaphatunk választ a levelünkre ezért nincs okunk kételkedni abban, hogy a kommunikáció bizalmasan zajlott le. Holott a támadó tud mindent, amit a mi üzenetünk tartalmazott. Mivel a betolakodó nem ismerheti a szerver titkos kulcsát ezt a problémát megoldottuk.

Most a szerver hitelességén túl vegyük figyelembe a kulcsok hitelességét is. Gondoljunk bele, mi történne akkor, ha korrumpálódna az említett adatbázis, és a név mellet már nem a hozzá tartozó eredeti kulcs állna. Azaz a támadó kicseréli a szerveren a nyilvános kulcsot a sajátjára. Ekkor a szerver aláírásával igazolná, hogy a kulcs hiteles ezzel is a hamis biztonság érzetét keltené bennünk. Az előbb említett folyamat ismét eljátszható lenne. Azaz elfogva a mi üzenetünket, képes lenne megfejteni a betolakodó azt a saját titkos kulcsával, és azután az eredeti nyilvános kulccsal rejtjelezve továbbküldheti a kriptoszöveget a mi nevünkben a címzett irányába. A kommunikáció ismét teljes lenne. Az ilyen és ehhez hasonló támadás kivédése érdekében rengeteg energiát kell fordítani a kulcsszerverek megfelelő védelmére, és ezek biztonságos üzembentartására.

Joggal merül fel a kérdés, hogy a szerver kulcsát vajon ki hitelesíti. Vagyis ki mondja meg nekünk, hogy a szerver által használt titkos és nyilvános kulcs ténylegesen a szerverhez tartozik-e. Aki a szerver kulcsát hitelesíti, annak a kulcsát ki hitelesíti, és így tovább. Több megoldási is létezik ennek a problémának a kezelésére. A legrégebben használatos modell egy hierarchikus fát képzel el (hierarchial trust). Ebben van legalább egy személy, vagy szervezet, akinek nem lesz hitelesített kulcsa, de őt ettől függetlenül mindenki elfogadja hiteles személynek. Ő a hierarchikus fa csúcsán helyezkedik el, neve: gyökérhitelesítő (root of certification tree). Mivel nincs senki, aki az ő kulcsát aláírhatná, így a nyilvános kulcsa csak közvetlen, direkt módon ellenőrizhető. Az olyan kulcshitelesítő szervereket, amelyek igazolják egy nyilvános kulcs és a tulajdonosának összetartozását hitelesítő szervezetnek, Certification Authory-nak (CA) nevezzük, míg a tulajdonos kulcs összetartozást igazoló és a CA által

kibocsátott igazolást hitelességi bizonyítványnak, Certificate-nek hívjuk. Az egyes szereplők a hierarchiában felettük állók igazolását elfogadják, függetlenül attól, hogy megbízhatónak, vagy megbízhatatlannak tartják, az igazolást kiállító személyt, vagy szervezetet. Azaz egy munkahelyi hierarchiában működő infrastruktúra esetén a főnök aláírását el kell fogadni a beosztottnak, függetlenül attól, hogy milyen embernek is tartja ő a főnökét. Kereszt és egyenrangú hitelesítések nem megengedettek. Az így kialakuló kapcsolatokban egy hitelességi bizonyítvány nyomon követése, a bizonyítványlánc feltárása alulról felfelé (bottom-to-up) történik. A modell nagy hátránya, ha egyszer kompromittálódik a gyökérhitelesítő (root of certification tree), akkor az egész rendszer összeomlik. A modell struktúrája miatt, egy lassan vagy alig változó, statikus környezetben használható jól.

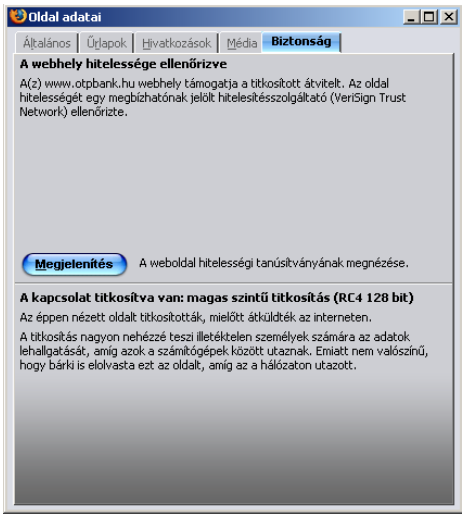
Egy másik a kulcsok hitelesítéséhez használatos módszer a bizalmi háló elvén működő módszer. Itt nincs alá és fölé rendelt viszony. Ennek köszönhetően, ez főként a privát szférában terjedt el, de napjainkban az üzleti megoldások is e felé haladnak. Ez a módszer ötvözi a közvetett és közvetlen módszerekkel történő igazolást. Megengedi a kereszthitelesítéseket és így egy pókháló szerű kapcsolatlánc jön létre. A modell kevésbé érzékeny egy CA kompromittálódására, köszönhetően a többszörös kapcsolatoknak. Nincs szükség root-CA tevékenységre, mert minden résztvevő maga dönti el, hogy kinek az igazolását fogadja el. Ha egy kulcs egy olyan személy vagy intézmény által van aláírva, akiben feltétlenül megbízunk, akkor elfogadjuk hitelesnek azt a kulcsot. A mi kulcsunkkal mi is hitelesíthetünk kulcsokat a kommunikációs partnereink felé, akik megbíznak a mi kulcsunkban és döntésünkben. Ha a partnereink meglátják a mi

aláírásunkat egy számukra ismeretlen személyhez tartozó nyilvános kulcson, és minket elfogadnak megbízható személynek, akkor el fogják ők is fogadni ezt az általunk „bemutatott” nyilvános kulcsot. Meghatározhatjuk, hogy hány megbízhatónak ismert személy aláírása szükséges egy nyilvános kulcson ahhoz, hogy mi azt hitelesnek fogadjuk el. Bárki hitelesíthet, azaz aláírhatja bárki kulcsát, de hogy kinek az aláírását fogadjuk el, az csak rajtunk múlik. Az elfogadott kulcsok, újabb kulcsokat hitelesíthetnek. Az így kialakuló hitelesítési lánc mélységéről is rendelkezhetünk egy ilyen modellben. A bizonyítványok eredetét és „családfáját” azonban sokkal nehezebb feltérképezni. A rengeteg kölcsönös és kereszthivatkozások miatt lehet, hogy végtelen ciklusba kerülünk. Nagy előnye viszont, hogy roppant rugalmas így egy gyorsan, dinamikusan változó környezetben is kiválóan alkalmazható.

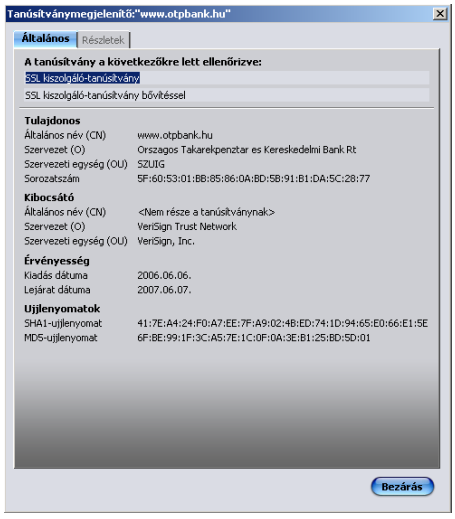
A hierarchikus és hálós hitelesítési modellek közül azonban mégis a hierarchikus felé mozdul el a gyakorlat. Ennek oka, hogy a hitelesítő tevékenység egyre fontosabb szerepet játszik az államigazgatási gyakorlatban. A nagy pénzintézetek is inkább megbíznak az általuk rendszeresen ellenőrzött CA-kban, mint egy kontrollálatlan hálós modellben. A törvényhozásban is egyszerűbb egy szabályokat jobban követő és strukturáltabb, szigorúbb felépítésű hierarchikus modellre törvényeket megalkotni. Így a valós világban, szinte kizárólagosan a hierarchikus modell található meg.

Itt említenék meg egy Magyarországon nemrég bevezetésre került szolgáltatást, az elektronikus közjegyzést. A magyar közjegyzőknek fel kellett készülniük erre az egész ország területén. Az összes közjegyzőnek rendelkezni kell minősített közjegyzői aláíró tanúsítvánnyal, amit minden ember hitelesnek fogad el az országban. [20, 21]

A másik roppant fontos kérdés a hitelesség mellett, a kulcsok érvényessége és az érvényességi idők korrekt kezelése. Minden a kulcsok kezelésével foglalkozó rendszernek lehetőséget kell biztosítani arra, hogy a saját kulcsait valaki érvénytelennek minősítse. Erre akkor kerülhet sor, ha a kulcsot például elveszítette a tulajdonosa vagy kulcsát eltulajdonították, esetleg feltörték, és erről értesül. Ám a kulcsot semmiképpen sem szabad törölni az adatbázisból, mert egy olyan dokumentum aláírás ellenőrzésekor, ami még az érvénytelenítés előtt keletkezett, szükség lehet rá. Ha csak a gyanúja is felmerül annak a ténynek, hogy a kulcsunk kompromittálódott, akkor azonnal érvénytelenítsük azt. Amíg ezt nem tesszük meg, addig az illetéktelen felhasználó minden gond nélkül elolvashatja az üzeneteinket, és aláírhat a mi nevünkben bármit. Ez a cselekedet, kimeríti az okirathamisítás fogalmát. Az előbb említett bizalmi hálós modellben lehetőségünk van megkérni minden aláírót, hogy vonja vissza az aláírását a mi kulcsunkról, ezzel is csökkentve azok körét, akik a bizalmi elv alapján felhasználhatják a mi kulcsunkat. Az érvénytelenítés a kiadott hitelességi bizonyítványok visszavonását (revoking of certification) is jelentheti, és emellett a kulcsot egy úgynevezett érvénytelenségi listára helyezi. A kulcsok nem csak kompromittálódásuk miatt kerülhetnek az érvénytelenségi listára. A kulcsokat általában határozott időre generáljuk, és ennek az időnek a letelte után a kulcs ugyanúgy visszavonásra kerül, mintha ellopták volna. Ezzel is kikényszerítve azt, hogy a felhasználó bizonyos időközönként cserélje le a kulcsait. Tehát egy érvénytelen hitelességi bizonyítvány nem jelenti feltétlenül azt, hogy a hozzá tartozó kulcs kompromittálódott. Ha használni szeretnénk egy nyilvános kulcsot titkosításra, vagy aláírás ellenőrzésre feltétlenül győződjünk meg arról, hogy nem hamis-e, nem járt-e le, és hogy nincs-e érvénytelenítve.



8. ábra – Az OTP bank biztonsági ellenőrzése



9. ábra – A kulcs érvényességének ellenőrzése

Ha nem lennénk még megelégedve, az így nyert adatokkal további információkat kaphatnánk a kulcsról, ha a részletek fület lenyitnánk (lásd 9. ábra).

4.4. Titkos kulcsok tárolása

Elérkeztünk végezetül oda, hogy a nyilvános kulcsok miatt már nem kell aggódnunk, de a titkos kulcsokról és azok tárolásáról még egy szó sem esett. Mivel a digitális aláírás napjainkban elfogadott forma ezért a privát kulcsok biztonságos tárolása talán még fontosabb, mint a nyilvánosaké. Egy bárki által használható gépen lehetőleg ne is használjuk, vagy járjunk el fokozott óvatossággal. A probléma tehát abból adódik, hogy a bonyolult sok-sok bites kulcsokat kénytelenek vagyunk elektronikus formában tárolni, nem tudjuk fejben tartani, és ehhez a kulcsfájlhoz esetleg mások is hozzáférhetnek, ha gépünket felügyelet nélkül hagyjuk, vagy megfelelő védelem nélkül csatlakoztatjuk a hálózatra. Kézenfekvő, hogy a kulcsfájlt inkább olyan hordozón tároljuk, mely csak addig van a gépben, ameddig feltétlenül szükséges, utána kivesszük, és magunkkal visszük. Magát a titkos kulcsot hordjuk magunknál ugyanúgy, mint az okiratainkat. Ezt az

intelligens kulcstároló eszközök segítségével valósíthatjuk meg, vagy ha nem rendelkezünk ilyen eszközökkel, akkor egy USB kulcs jó ötletnek tűnik. A kulcsot ekkor nekünk kell importálni a rendszerbe, exportálni a rendszerből, majd gondoskodni arról, hogy használat után még hírnyoma se maradjon a privát kulcsunknak. Ám itt fennáll az a veszély, hogy a támadónak akár csak néhány percre is sikerül megszerezni az eszközt, akkor a fájlt észrevétlenül lemásolhatja, hisz nem rendelkezik az USB kulcs hozzáférési védelemmel. Az említett intelligens tároló eszköz beszerzése azért a legjobb megoldás, mert a rajta tárolt adatokat akár az épsége árán is megőrzi. Egy lehetséges ilyen eszköz a Rainbow iKey 2032, vagy iKey 3000, amely megjelenését tekintve egy USB kulcsra hasonlít, de az adatokat titkosítva és védve tárolja.

10.ábra – Rainbow iKey3000

A titkos kulcsok tárolásának ideális megoldását az ún. intelligens kártyák (smart card) jelentik. Az intelligens kártyák bankkártyaszerű berendezések, melyek a kártya anyagába ágyazva miniatűr célszámítógépet (chip-et) tartalmaznak az alábbi tulajdonságokkal felruházva:

- Az intelligens kártya, pontosabban a rajta elhelyezett chip speciális kártyaolvasó készüléken keresztül kapcsolható a felhasználói munkaállomáshoz. Ezt követően a felhasználói gép és a kártyára integrált célszámítógép képes kommunikálni egymással.
- A kártya chip alkalmas adattárolásra, például kulcsfájl befogadásra. A tárolt adatokat fizikailag védi, tartalmához csak PIN kód megadását követően enged hozzáférést. Rögzített számú sikertelen PIN kód megadás

után (pl. 5) a kártya blokkolódik, több kódot már nem fogad el, a próbálgatásos támadások kivédése érdekében.

- A kártya chip-eket úgy tervezik, hogy hatékonyan ellenálljanak a legkifinomultabb fizikai támadásoknak is, információ tartalmuk – a gyártók szándékai szerint – még jól felszerelt laboratóriumokban sem kinyerhetők.
- A modernebb kártya chip-ek már nem csak adattárolásra, hanem alapvető kriptográfiai műveletek elvégzésére is képesek, mint például nyilvános és titkos kulcspár generálása, valamint nyilvános kulccsal titkosított üzenet visszafejtése és digitális aláírás készítése aszimmetrikus algoritmussal. Az ilyen típusú kártyák esetén már megoldható, hogy az aszimmetrikus kulcspár titkos magán komponense teljes életciklusa alatt soha ne hagyja el a kártya biztonságos környezetét. Ott generálódik és a visszafejtési és digitális aláírás készítési műveletekre is a kártya chip-en belül kerül sor. Valóban még ilyen kártya nem létezik, de intenzív kutatási és fejlesztési erőforrásokat fordítanak kifejlesztésére.

11. ábra – Master Chip Card

12. ábra – Visa Chip Card

Ezeket a kártyákat napjainkban főleg bankkártyaként alkalmazzák, mivel mágnescsíkos elődeitől eltérően nem másolhatók, és a kért tranzakciók esetükben még digitális aláírással is hitelesíthetők. Használatuk azonban az

informatikai biztonság más területein is gyorsan terjed. Számos kriptográfiai alkalmazás támaszkodik már rájuk és a jogintézményként bevezetésre kerülő digitális aláírás rendszerek sem képzelhető el nélkülük. Ezen kártyák hátrányaként meg kell említeni, hogy speciális olvasó berendezésekre van szükség az alkalmazásukhoz, ami így nem támogatja a mobil felhasználót és költséges beruházást is igényel.

4.5. Előnyök és hátrányok

Vegyük sorra, mit is nyerhetünk egy ilyen rendszer használata esetén, és hogy milyen negatívumokkal jár a nyilvános kulcsú infrastruktúra használata. Az előnyöknél és hátrányoknál viszonyítási alapként használjuk fel a lényegesen nagyobb múlttal rendelkező szimmetrikus algoritmusok családját és azok jellemző tulajdonságait. Így képet kaphatunk arról vajon megéri-e számunkra egy esetleges aszimmetrikus rendszer használata a szimmetrikus helyett.

Előnyök:

- A résztvevő feleknek két kulcsuk van, ezek feladata más és más. Ebből az egyiket nyilvánosságra lehet hozni, és szabadon lehet terjeszteni, míg a másikat titokban kell tartani, és szigorúan kell őrizni. A kulcsok tulajdonságai lehetővé teszik az egyszerű és gyors kulcscserét a kommunikálni kívánó partnerek között. Nem szükséges a személyes találkozás és a biztonságos csatorna ehhez, szemben a szimmetrikus rendszerekkel, ahol a kulcsok egyeztetése meglehetősen nagy bonyodalommal jár.
- A kulcsok szerepe nem kötődik a fizikai folyamatokhoz, mivel a titkosító és megfejtő folyamat csak logikailag különbözik. Nem lehet egyértelműen azt állítani, hogy a nyilvános kulcsot csak kódolásra, a titkos kulcsot meg csak

dekódolásra használhatjuk fel. Hiszen láttunk erre nagyon szép ellenpéldát, a digitális aláírásnál, ahol a szerepek felcserélődnek.

- Nagy létszámú résztvevő esetén sem jelenthet gondot a kulcsok megosztása. Szimmetrikus rendszerekkel ellentétben itt N résztvevő esetén, nem N*(N-1)/2 kulcsot kell kezelni, hanem csak N darabot. Tehát takarékosan bánunk a rendelkezésre álló szűkös erőforrással, a tárhellyel.
- A már előbb említett digitális aláírás teljes mértékben meg tud valósulni. Teljesít a rendszer minden olyan kritériumot, ami szükséges egy hiteles digitális aláírás elkészítéséhez. Ezt annak köszönhetjük, hogy a legtöbb aszimmetrikus rendszerben a titkosító és megfejtő folyamatok logikailag felcserélhetők. A titkos kulcs egyedisége valódi biztonságot garantál az aláírónak.
- A használt algoritmusok jól definiáltak, és megfelelően nehéz matematikai problémákon alapszanak. Így a bizalom az algoritmusok irányában töretlen. Ez azt eredményezi, hogy a kulcsok felhasználhatósági ideje jóval hosszabb lesz. Nem ritka, hogy egy-egy kulcspárt évekig használjanak.
- A használt kulcsok mérete sokkal hosszabb, mint a szimmetrikus rendszereknél. A javasolt kulcshossz 1024 és 2048 bit között mozog, szemben 128-256 bites szimmetrikus hosszokkal. Ezek a nagy és hosszú kulcsok kiszűrnek támadási formákat, és elriasztó hatásuk van. Lehetetlenné teszik például a próbálgatásos, úgynevezett brute-force támadásokat.

Hátrányok:

- A hosszú kulcsokból és a nehéz matematikai problémákból adódik a nagy számolási kapacitás, és az algoritmusok lassúsága. Nem képesek így bizonyos gyakorlati igényeket

kielégíteni. A rivális algoritmus családnál a számolások, akár több százszor is gyorsabban végbemennek. Ezért konkrét felhasználáskor, ötvözni szokták az aszimmetrikus és a szimmetrikus algoritmusokat, és egy úgynevezett hibrid rendszert használnak.

- Az óriási kulcsok nehézkessé teszik a kulcsmenedzsmentet. A nagyobb kulcsok tárolása, több helyet igényel, tehát amit megspórolunk a kulcsok csökkenő darabszámánál, azt itt felhasználjuk. A kulcsok szerverről történő lekérdezése, rendszerbe történő importálása, vagy rendszerből történő exportálása időtöbbletet jelent, és állandó hálózati hozzáférést kíván meg egy kulcsszerverhez.
- A kulcsok mérete és a velük való számolás, nagyszám aritmetikát igényel, aminek megvalósítása nehézkes és bonyolult feladat.
- A nyilvános kulcsú infrastruktúra működéséhez szükségünk van egy megbízható harmadik félre, aki garantálja és tanúsítja, hogy a felhasználóhoz valóban az a kulcs tartozik ami. Azaz a kulcs és tulajdonosa ténylegesen összetartoznak.
- Egyetlen algoritmusról sem bizonyították be a mai napig, hogy elméletileg feltörhetetlen, azaz hogy teljesen biztonságos lenne. Ezt eddig csak a One Time Pad (OTP) algoritmusok tudják. Az PKI-k alapjául szolgáló algoritmusok biztonságosságát az adja, hogy az inverz műveleteikhez a mai ismereteink szerint olyan operációkat kell végrehajtani, amelyek előállítása vagy időben, vagy társzükségletben lehetetlen.

A nyilvános kulcsú algoritmusok történelmi háttere meglehetősen kicsi. A gyakorlati és elméleti tapasztalatok

csak néhány évtizedes múlttal büszkélkedhetnek. Az 1970-es évektől származtathatjuk őket, amikor is előkerültek azon matematikai problémák, amik az algoritmusok alapját képezik. Ekkor jelentek meg olyan számítástechnikai eszközök is, melyek segítségével a kutatás megindulhatott. Így nem tudhatjuk, hogy az algoritmusok valóban kiállják-e majd az idő próbáját.

5. PKI

A nyilvános kulcsú infrastruktúra, hasonlóan az elektromos hálózathoz, egy háttérszolgáltatás. Ahogyan az elektromos hálózat lehetővé teszi a legkülönfélébb villamos berendezések (hűtőgép, mosógép, vasaló, TV, stb.) működtetését szerte a világon, azáltal, hogy szabványos módon otthonainkba szállítja a villamos energiát, úgy digitális környezetben a nyilvános kulcsú infrastruktúra szolgálja ki alkalmazásainkat. Ha be szeretnénk mutatkozni valakinek, vagy éppen mások kilétét szeretnénk ellenőrizni, vagy ha adatainkat titkosítani kívánjuk, esetleg digitális aláírással látjuk el, a PKI szolgál ki minket és az alkalmazásainkat. Kicsit konkrétabban fogalmazva a nyilvános kulcsú infrastruktúra az, amelyre rákapcsolódva a böngésző programunk ellenőrizni tudja az elektronikus áruház vagy bank webszerverét a pénzügyi tranzakciók lebonyolítását megelőzően. A PKI szolgáltatásra támaszkodik levelező programunk is, mikor elektronikus üzeneteink rejtjelezését és digitális aláírását végzi, és a szövegszerkesztőnk előtt ülve is ugyanezen szolgáltatást használjuk ki amikor, az éppen elkészült dokumentumot jogilag is elismert elektronikus aláírással szignáljuk.

A nyilvános kulcsú infrastruktúra a feladatát ügyfelei (hálózati eszközök, szerverek, valóságos személyek) azonosításán keresztül látja el. Ahogy egyes országok államai hivatalos dokumentumokat adnak (személyazonossági igazolvány, útlevél, stb.), bocsátanak ki polgárai azonosítására, úgy a nyilvános kulcsú infrastruktúra (illetve az azt működtető hitelesítő szervezet) digitális tanúsítványok kiállításával igazolja a hozzá fordulók kilétét. A kiállított tanúsítvány szabványos elektronikus dokumentum [lásd 5.3.], kezelésére a legkülönbözőbb gyártóktól származó legkülönfélébb eszközök és

számítógépes alkalmazások is képesek. A tanúsítvány alapján a kommunikáló partnerek, számítógépes alkalmazások, berendezések képesek egymás személyazonosságának megállapítására, egymás digitális aláírásának ellenőrzésére, valamint az átadott információk titkosítására és visszafejtésére. A nyilvános kulcsú infrastruktúra tehát képessé teszi a különböző forrásból származó számítógépes programokat, hogy együttműködjenek fejlett informatikai biztonsági funkciók ellátása során. A programokat természetesen fel kell készíteni az együttműködésre. Ahogyan a mosógép is csak fogadja az áramot az elektromos hálózatból, s a többi az ő dolga, úgy a PKI-ra felkészített alkalmazás is csak fogadja a digitális tanúsítványokat, azok kezelése, felhasználása már ugyanúgy az ő dolga, feladata.

5.1. Az infrastruktúrával szembeni elvárások

Tekintsük meg azt, hogy egy ilyen nyilvános kulcsú infrastruktúrával (Public Key Infrastructure – PKI) szemben milyen lényegi elvárások merülnek fel. Egy szervezet alkalmazottjai, illetve a szervezet információs rendszereit használó külső partnerek, polgárok elvárják, hogy a rendszerek és a bennük tárolt adatok, információk, megbízható módon, az év 365 napján biztonságosan a rendelkezésükre álljanak. Azaz egész évben, a hét 24 óráján át rendelkezésre kell állnia ennek az infrastruktúrának, minden felhasználó számára. Beleértve a rendszereket, hálózatokat, a felhasználók segítését, a folyamatos, katasztrófatűrő működést, és mindezt jelentős többlet beruházások nélkül. Ezt a jó PKI a digitális tanúsítványokkal, illetve más minősített elektronikus jogosítványokkal biztosítja, lehetővé téve az ellenőrzött belépést az információs rendszerekbe, a titkosított üzenetkezelést és adatátvitelt a szervezet számára lényeges alkalmazások esetén. A követelmények teljesítésére egy ilyen

szolgáltatást nyújtani kívánó cég/szervezet **minősített** on-line PKI szolgáltatást kell üzemeltessen, amely alkalmas a felhasználók azonosítására (authentication), a jogosultságaik szerinti tevékenységek engedélyezésére (authorization) és a digitális aláírások kulcsainak menedzselésére a legkülönbözőbb Internet/intranet alkalmazásokban. A szervezeti PKI célja az adatvagyon védelme a hitelesítés, a bizalmasság, a letagadhatatlanság, a teljesség és a rendelkezésre állás megteremtésével.

A megbízható PKI technológiába beletartozik a tanúsítvány kibocsátás és életciklus menedzsment, a különféle tanúsítvány típusok előállítása és működtetése, a megfelelő nyilvántartási tevékenységek, a rekordok tárolása, a címtárak integrálása és a kulcsok menedzselése a legkritikusabb üzemeltetési és terhelési viszonyok között is. Általános tévedés, amit sokan gondolnak, hogy a PKI valami tárgyiasítható dolog, pedig valójában a PKI egy lehetőség, amely a nyilvános kulcsok közzétételét, menedzselését és használatát könnyíti meg.

A PKI szolgáltatás azonban jogi és pénzügyi felelősséggel jár, és igen jelentősek a bizalmi szempontok a sikerességében. Az hogy mely cégek, felhasználók, milyen szolgáltatót választanak, nagyban függ a szolgáltató megbízhatóságától, a nevéhez kapcsolódó referenciáktól, és a szolgáltatóba fektetett általános bizalomtól. A szolgáltatónak biztosítania kell a teljes információtechnológiai hátteret beleértve az informatikai rendszereket, távközlést, adatbázisokat, de ezen túlmenően a telephely fizikai biztonságát, az "Internet-biztos" belső hálózatokat, a magas rendelkezésre állást biztosító redundáns rendszereket, a katasztrófatűrő képességet, a PKI szakterület szakértőit, életképes PKI joggyakorlatot, és financiálisan biztos PKI szavatosság védelmet.

A költségek és a felelősség megosztására integrált, hierarchikus rendszereket használnak elterjedten. A PKI tanúsító hatóság (CA) telepítésének első napjától a legmagasabb biztonsági követelmények szerint kell eljárni, amelyek szigorúbbak az általános titkosított adatátviteli és feldolgozó rendszerekre előírt követelményeknél. Ez akkor is igaz, ha az első telepítés idején nincsenek szenzitív, védendő állományok, azonban a későbbiekben bármikor szükséges lehet igénybevétele az igen értékes információk védelmére. Tekintve, hogy a CA szigorú biztonsági feltételeket támaszt, az épület és az üzemeltetés költségei sok vállalkozás számára elriasztóak.

A CA funkció külső kézbe adása nem feltétlen megoldás, mivel a vállalatok legtöbbször teljes irányítási, ellenőrzési jogot akarnak gyakorolni a nyilvános kulcsú infrastruktúrájuk felett. Mivel maguk akarják meghatározni, hogy ki kaphat tanúsítványt, mi legyen a tanúsítvány tartalma, hogyan és miképp vonható vissza a tanúsítvány és a CA mindennapi üzemeltetését is kézben szeretnék tartani. De van amikor nincs más megoldás, mint külső szolgáltatót bevonni a rendszerbe, aki magas színvonalon tudja szavatolni és nyújtani az említett szolgáltatásokat. Alapkövetelmény továbbá, a szabvány alapú átjárhatóság, azaz a nyílt együttműködés a hálózatok között. Alkalmasnak kell lennie a külső kapcsolatok kezelésére, mivel a PKI különböző szervezeteket, közösségeket szolgálhat ki egy zárt hálózaton belül, több privát hálózat között és azokon kívül az Interneten is. Ez a nyílt architektúra, amely a nemzetközi piac legjobb alkalmazásaira épül, biztosítja valamennyi régi és bármilyen új alkalmazással az együttműködést. Elkerülve a végfelhasználói rendszereken az alkalmazás specifikus szoftverek telepítését, továbbá

lehetőséget ad a szükséges együttműködésre a szervezeten kívüli rendszerekkel.

5.2. A PKI komponensei és feladatai

A legalapvetőbb építőköve a rendszernek a hitelesítő eszközök (CA, Certification Authority), továbbá szükséges még a regisztrációs egység (RA, Registration Authority) és a nyilvános tanúsítványtár (Directory). A komplex faladatok megoldásához a rendszert alkotó különböző komponensek precíz együttműködése szükséges. A teljes infrastruktúrának meg kell valósítania:

- Ügyfelek biztonságos azonosítását.
- Nyilvános kulcsú tanúsítványok kiadása azonosított ügyfelek részére.
- Lejárt tanúsítványok megújítása.
- A tanúsítványok publikálása nyilvános címtárban.
- Tanúsítványok visszavonása, a visszavont tanúsítványok listájának kiadása.

Hitelesítő szerezet (CA, Certification Authority)

A hitelesítő szervezet (tanúsító szevezet, eredeti angol nevéből származó, elterjedten használt rövidítéssel CA) a rendszer központi eleme.
Feladatai:

- ✓ Tanúsítványkérések fogadása.
- ✓ Egyes implementációkban a kulcspárok generálása.
- ✓ A beérkező tanúsítványkérések alapján megfelelő nyilvános kulcsú tanúsítványok kiállítása.

- ✓ A kiadott tanúsítványok közzététele (publikálása) a nyilvános tanúsítványtárban.
- ✓ Korábban kiadott tanúsítványok megújítása, szükség esetén új kulcspár előállítása.
- ✓ Kompromittálódott, régi vagy használhatatlanná vált kulcsokhoz tartozó tanúsítványok visszavonása.
- ✓ A visszavont tanúsítványok listájának közzététele (publikálása) a tanúsítványtárban, vagy egyéb, bárki számára hozzáférhető helyen.

A CA rendszerint egy jól védett számítógép, mely a kért elektronikus dokumentumokat összeállítja és digitálisan szignálja. Megfelelő fizikai és logikai védelem kiemelkedő jelentőségű, hiszen ha bárki illetéktelen személy a hitelesítő szervezet magánkulcsának birtokába jut, azonnal képessé válik hamis tanúsítványok (más szóval kulcsigazolások) kiadására a valódi CA nevében, ami az egész rendszer hitelességét romba dönti. Ezért a tanúsítvány kiadó eszköz rendszerint jól védett épületben, lezárt szobában tárolják, ahol csak kiválasztott személyek szűk köre férhet hozzá. Magánkulcsát külön hardver alapú biztonsági modul őrzi, s a berendezést bizonyos esetekben még a hálózatra sem kapcsolják rá. Ha mégis, akkor külön tűzfal mögött helyezkedik el, ami csak jól meghatározott végpontokról, szigorúan ellenőrzött kommunikációt hajlandó átengedni.

Regisztrációs egység (RA, Registration Authority)

A regisztrációs egység (RA) a rendszernek az az eleme, mely közvetlen kapcsolatot tart az ügyfelekkel. Az adminisztratív munka ezen a szinten jelenik meg.

Feladata:

- ✓ Az ügyfelek megbízható azonosítása.
- ✓ Az ügyfelek azonosítását követően a tanúsítvány kérés összeállítás és továbbítása a tanúsítványkiadó eszköz (CA) részére.
- ✓ Ügyfelek tanúsítvány visszavonási kéréseinek fogadása és továbbítása a CA-nak (bizonyos megvalósítások esetén).

Vállalati infrastruktúra esetén az RA funkciót megvalósíthatja a személyzeti osztály, amely az új belépőkkel kapcsolatos adminisztrációt amúgy is intézi, kereskedelmi szolgáltató esetén pedig a feladatát elláthatja az ügyfélszolgálati iroda. Az RA feladat ellátásához természetesen mindkét alkalommal tartoznia kell megfelelő számítógépnek, amely képes szabványos elektronikus tanúsítványkérő dokumentum összeállítására és továbbítására. Bizonyos estekben azonban az RA funkció nem igényli formális iroda fenntartását. A célnak egy egyszerű Web szerver is megfelel, ahová az ügyfelek közönséges böngésző programmal jelentkeznek be, s egyszerű kérdőív kitöltésével indítják el a tanúsítványkérésüket. [lásd 5.7.] Ugyanabban a nyilvános kulcsú infrastruktúrában több regisztrációs egység is helyet kaphat. Nagyvállalati rendszer esetén az egyedüli hitelesítő eszköz működhet például a központban, s egy-egy regisztrációs egység minden vidéki telephelyen.

Nyilvános tanúsítványtár (Directory)

Tanúsítványtáron, ebben az összefüggésben olyan adatbázist értünk, mely tartalmazza a tanúsító eszköz (CA) által kibocsátott tanúsítványokat (kulcsigazolásokat), a visszavont tanúsítványok listáját, és alkalmas arra, hogy bármely tanúsítvány állapotáról (érvényes, visszavont, felfüggesztett)

rövid idő alatt felvilágosítást adjon. Két legfontosabb felhasználási területe:

- ✓ Ha egy ügyfél részére titkos üzenetet szeretnénk küldeni, és nyilvános kulcsa még nincs a birtokunkban, a tanúsítványtárból könnyen hozzájuthatunk.
- ✓ Bizalmas üzenet küldését, vagy titkosított hálózati kommunikációt megelőzően egyszerű lekérdezéssel meggyőződhetünk róla, hogy a partner tanúsítványa érvényes-e még (nem vonták-e vissza idő közben).

A tanúsítványtár elérése rendszerint az LDAP (Lightweight Directory Access Protocol) [24, 25] protokoll használatával történik. A nyilvános kulcsú infrastruktúra szolgáltatásait valódi és közvetett ügyfelek veszik igénybe. Valódi ügyfélen azokat a személyeket, szervezeteket értjük melyek tanúsítványát az infrastruktúra CA-ja állította ki. Közvetett felhasználók pedig azok, akik ugyan nem rendelkeznek az infrastruktúra CA-jától kulcsigazolással (tanúsítványuk vagy mástól származik, vagy egyáltalán nincs nekik), viszont az infrastruktúrát üzemeltető szervezetet elfogadják megbízható hitelesítés szolgáltatóként, azaz elfogadják az általa kiállított kulcsigazolásokat.

5.3. Tanúsítványok, szabványok

Annyit emlegettük már őket, hogy lassan ideje, hogy pontosan definiáljam mit is értünk tanúsítvány alatt. Hogyan is épül fel és milyen irányelvek mentén születik meg. Mivel a nyilvános kulcsú infrastruktúra célja az informatika legkülönbözőbb területein felmerülő biztonsági igények kielégítése, feladatát csak akkor képes megfelelően ellátni, ha a vele kapcsolatba kerülő legkülönbözőbb gyártóktól származó eszközök is képesek az együttműködésre. Ez csakis

szabványok kidolgozásával és a szabványokhoz illeszkedő infrastruktúra komponensek, eszközök és alkalmazások fejlesztésével történhet. Mit is kell szabványosítan? – merül fel a kérdés. Néhány példa a teljesség igénye nélkül:

- A tanúsítványkérő elektronikus dokumentum formátuma, milyen információkat tartalmazzon és pontosan hogyan.
- A nyilvános kulcsú tanúsítvány formátuma milyen legyen, milyen információkat tartalmazzon és pontosan hogyan.
- A nyilvános tanúsítványtárhoz való hozzáférés szabályrendszere, protokollja.
- A visszavonási listák formátuma, milyen információkat tartalmazzon és pontosan hogyan.
- Kulcsok és tanúsítványok továbbításának formátuma.

A felhasznált szabályok különböző forrásokból származnak. Az RSA cég, mely úttörő szerepet játszott a nyilvános kulcsú kriptográfián alapuló technológiák terjesztésében, a szabvány kidolgozásában is az élen járt. Nevükhöz fűződik az ún. PKCS ajánlások (Public Key Cryptography Standard – Nyilvános Kulcsú Kriptográfiai Szabvány) megalkotása. Az ITU (International Telecommunication Unit) nemzetközi távközlési szervezet szintén több fontos dokumentummal járult hozzá a munkához. Közülük, külön ki kell emelni, az X.509 jelű ajánlást, mely a nyilvános kulcsú tanúsítványok formátumát határozza meg. Az Internet Engineering Task Force (IETF) pedig PKIX néven olyan szabványkészletet dolgozott ki, mely a nyilvános kulcsú infrastruktúrát illesztette az Internet hálózati technológiájához. A PKIX ajánlások közül külön ki kell emelni az RFC 2459 jelű dokumentumot, mely az X.509 szabványnak megfelelő nyilvános kulcsú tanúsítvány leírását tartalmazza.

Most pedig nézzük az ígért szabványos tanúsítványt, hogy is kell neki kinéznie, és milyen mezőket kell neki tartalmaznia. Mint említettük az X.509 jelű ajánlás az irányadó annak is 3.0 verziószámot viselő változata. Ez az ajánlás pontosan meghatározza, milyen mezőket kell és lehet a tanúsítványban szerepeltetni, s a kérdéses információkat hogyan kell a dokumentumban ábrázolni.

- Verzió (version): Annak az X.509 ajánlásnak a verziószáma, melyhez igazodva a tanúsítványt kiállították. A lefrissebb a hármas verzió (V3).
- Sorozatszám (serial number): A tanúsítvány egyedi sorszáma. Ugyanaz a tanúsító eszköz nem adhat ki két megegyező sorozatszámú kulcsigazolást.
- Aláírási algoritmus (signature algorithm): A digitális aláíró algoritmus megjelölése, mellyel a hitelesítő eszköz a tanúsítványt szignálta.
- Kiállító (issuer): A tanúsítványt kiállító hitelesítő szervezet *részletes neve*.
- Érvényesség kezdete (valid from): A tanúsítvány érvényességének kezdő dátuma.
- Érvényesség vége (valid to): A tanúsítvány lejárati dátuma.
- Tulajdonos (subject): A tanúsítvány tulajdonosának *részletes neve*.
- Nyilvános kulcs (public key): A tanúsítvány tulajdonosának nyilvános kulcsa.
- Alapvető megkötések (basic constraint): A kulcs felhasználhatóságára vonatkozó korlátozások. Például itt tüntethető fel, hogy a kulcs tulajdonosa egyszerű ügyfél,

vagy kibocsátó tanúsító szervezet. Egyszerű ügyfélkulccsal hiába állítunk ki tanúsítványt, az alkalmazások nem fogják a keletkezett dokumentumot hitelesnek elfogadni.

- Kulcshasználat (key usage): A tanúsítványnak ez a része rögzíti, hogy a kulcs pontosan mire használható fel (digitális aláírás, titkosítás, bejelentkezés, stb.). Egyszerre több cél is megjelölhető, de nem ajánlott.
- Kibővített használat (enhanced key usage): Itt további célok jelölhetők ki a kulcs használatára vonatkozóan.
- Rövid név (friendly name): A kulcs tulajdonosának rövid, egyszerű neve.
- Aláírás (signature): A tanúsítványt kibocsátó szervezet CA digitális aláírása.

Az X.509-es típusú kulcsigazolások értelmezéséhez még egy nagyon lényeges fogalmat kell tisztáznunk. Két helyen is szerepel dőlt betűvel a részletes név (Distinguished Name DN) fogalma. A részletes név összetett struktúra, képes leírni egy személy vagy szerepkör pontos helyét valamely szervezeti hierarchiában, lépésről lépésre adva meg melyik országban, melyik városban, mely szervezet melyik egységénél dolgozó személyről van szó.

Az alakotóelemei:

- Ország (Country – C): Az országnak a megjelölése, mely a részletes név tulajdonosának lakóhelye, tartózkodási helye.
- Állam vagy tartomány (State or Province – S): Az országon belül az állam, tartomány vagy megye neve.

- Helység (Locality – L): Város, helység, a részletes név tulajdonosának lakóhelye, tartózkodási helye.
- Szervezet (Organization – O): Szervezet, cég, ahol a név tulajdonosa dolgozik.
- Szervezeti egység (Organizational Unit – OU): Szűkebb szervezeti egység a cégen belül.
- Közönséges név (Common Name – CN): A tulajdonos neve.

Ha én, Farkas Róbert Debrecenben dolgozom a Debreceni Egyetemnél, az Informatikai Karon akkor a részletes nevem a következőképpen írható fel:

C=HU, S=Hajdú-Bihar, L=Debrecen, O=Debreceni Egyetem, OU=Informatikai Kar, CN=Farkas Róbert

Hitelesítő eszköz (CA) részletes nevében közönséges névként (CN) természetesen nem egy személy megjelölése, hanem a hitelesítő eszköz elnevezése szerepel. Például a Deutsche Telekom AG által működtetett tanúsító szervezet egyik legfelsőbb szintű tanúsító eszközének (root CA) részletes neve az alábbi:

C=DE, O=Deutsche Telekom AG, OU=T-TeleSec Trust Center, CN=Deutsche Telekom Root CA 1

Az X.509 típusú tanúsítvány kiállítóját és tulajdonosát részletes névvel kell azonosítani, de a név valamennyi lehetséges komponensét nem kötelező megadni. Legtöbbször a helység (locality) és az állam vagy taromány (state or province) mező marad el, amint ebben a fenti példában is történt. De természetesen a részletes név más összetevői is kimaradhatnak, magánszemélyek részére kiadott tanúsítványokban például a munkahely adatainak feltüntetése lehet szükségtelen. Az X.509-es tanúsítvány a felsoroltakon kívül még további

információkat is tartalmazhat, ezek listáját az RFC ajánlásban olvashatjuk. A lista saját kiterjesztésekkel is bővíthető.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2778 (0xada)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer:
            countryName               = HU
            localityName              = Budapest
            organizationName          = NetLock Halozatbiztonsagi Kft.
            organizationalUnitName    = Tanusitvanykiadok
            commonName                = NetLock Uzleti (Class B) Tanusitvanykiado
        Validity
            Not Before: Jan 19 15:35:56 2007 GMT
            Not After : Jan 19 15:35:56 2008 GMT
        Subject:
            countryName               = HU
            stateOrProvinceName       = Budapest
            localityName              = Budapest
            organizationName          = CIB Bank Zrt.
            organizationalUnitName    = E-banking Department
            commonName                = ibank.cib.hu
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:ca:c7:02:f6:46:37:7e:b8:d4:5a:d6:5a:ae:84:
                    ae:25:3f:76:b1:af:31:99:6d:d5:e3:a5:c4:74:d1:
                    2f:f6:f8:0b:32:a8:26:2f:ed:c8:ac:6e:6c:9b:84:
                    a3:e6:59:88:b2:0b:a6:bc:73:e2:eb:3d:8d:bc:6f:
                    5b:5e:f4:2b:a6:85:a2:0e:fc:2c:0d:15:f9:c9:82:
                    a8:5a:bb:46:11:b1:09:a9:65:7a:13:47:d3:dc:69:
                    58:e5:c7:1c:98:2c:eb:54:8b:75:48:b5:ad:04:89:
                    8e:53:fb:8c:b7:e0:1c:79:5f:67:27:f0:e6:4e:f9:
                    c0:56:d3:e6:15:3e:07:03:d7
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Key Encipherment
            X509v3 Basic Constraints: critical
                CA:FALSE
            Netscape Comment:
                FIGYELEM! Ezen tanusitvany a NetLock Kft. Altalanos Szolgaltatasi Feltetel
eiben leirt eljarasok alapjan keszult. A hitelesites folyamatat a NetLock Kft. termekfelel
osseg-biztositasa vedi. A tanusitvany elfogadasanak feltetele az eloirt ellenorzesi eljara
s megtetele. Az eljaras leirasa megtalalhato a NetLock Kft. Internet honlapjan a https://w
ww.netlock.net/docs cimen vagy kerheto az ellenorzes@netlock.net e-mail cimen. IMPORTANT!
The issuance and the use of this certificate is subject to the NetLock CPS available at ht
tps://www.netlock.net/docs or by e-mail at cps@netlock.net.
            Netscape Cert Type:
                SSL Server
            X509v3 CRL Distribution Points:
                URI:http://crl1.netlock.net/index.cgi?raw&crl=uzleti

    Signature Algorithm: sha1WithRSAEncryption
        67:25:90:76:c3:71:11:c0:da:08:d0:ac:96:57:ad:2c:33:0c:
        19:de:92:77:ba:8a:c5:f3:f1:72:59:44:3f:2a:86:20:65:a7:
        a7:57:01:f7:01:2f:92:0d:34:b2:d2:05:2b:b9:7a:88:24:56:
        58:1d:69:ff:cf:1f:33:c5:39:a3:08:e1:53:53:f3:f3:5f:ca:
        4a:d5:1f:83:68:48:b0:31:55:7d:6c:d4:dd:30:87:0f:fe:c3:
        e8:5f:fe:03:63:4a:44:eb:e1:d9:da:64:3e:c6:e6:cc:40:1d:
        de:97:86:61:93:02:fc:21:c9:82:ee:13:82:83:80:68:5d:65:
        e8:04

-----BEGIN CERTIFICATE-----
MIIFazCCBNSgAwIBAgICCtowDQYJKoZIhvcNAQEFBQAwgZkxCzAJBgNVBAYTAkhV
MREwDwYDVQQHEwhCdWRhcGVzdDEnMCUGA1UEChMeTmVOTG9jayBIYWxvemFOYml6
dG9uc2FnaSBLZnQuMRowGAYDVQQLExFUYW51c2lOdmFueWtpYWRvazEyMDAGA1UE
AxMpTmVOTG9jayBVemxldGkgKENsYXNzIEIpIFRhbnVzaXR2YW55a2lhZG8wHhcN
MDcwMTE5MTUzNTU2WhcNMDgwMTE5MTUzNTU2WjCBgTELMAkGA1UEBhMCSFUxETAP
BgNVBAgTCEJ1ZGFwZXNOMREwDwYDVQQHEwhCdWRhcGVzdDEWMBQGA1UEChMNQOlC
IEJhbmsgWnJOLjEdMBsGA1UECxMURS1iYW5raW5nIERlcGFydG1lbnQxFTATBgNV
BAMTDGliYW5rLmNpYi5odTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAyscC
9kY3frjUWtZaroSuJT92sa8xmW3V46XEdNEv9vgLMqgmL+3IrG5sm4Sj5lmIsgum
vHPi6zZNvG9bXvQrpoWiDvwsDRX5yYKoWrtGEbEJqWV6EOfT3GlY5cccmCzrVIt1
SLWtBImOU/uMt+AceV9nJ/DmTvnAVtPmFT4HA9cCAwEAAaOCAtYwggLSMA4GA1Ud
DwEB/wQEAwIFIDAMBgNVHRMBAf8EAjAAMIICWgYJYIZIAYb4QgENBIICSxaCAkdG
SUdzRUxFTSEgRXplbiBOYW51c2lOdmFueSBhIE5ldExvY2sgS2ZOLiBBbHRhbGFu
b3MgU3pvbGdhbHRhdGFzaSBGZWxOZXRlbGVpYmVuIGxlaXJOIGVsamFyYXNvayBh
bGFwamFuIGtlc3p1bHQuIEEgaG1OZWxlc2lOZXMgZm9seWFtYXRhdCBhIE5ldExv
Y2sgS2ZOLiBOZXJtZWtmZWxlbG9zc2VnLWJpenRvc2lOYXNhIHZlZGkuIEEgdGFu
dXNpdHZhbnkgZWxmb2dhZGFzYW5haybmZWx0ZXRlbGUgYXogZWxvaXJ0IGVsbGVu
b3J6ZXNpIGVsamFyYXMgbWVndGVOZWxlLiBBeiBlbGphcmFzIGxlaXJhc2EgbWVn
dGFsYWxoYXRvIGEgTmVOTG9jayBLZnQuIEludGVybmxhcGphbiBhIGh0
dHBzOi8vd3d3Lm5ldGxvY2submVOL2RvY3MgY2ltZW4gdmFneSBrZXJoZXRvIGF6
IGVsbGVub3J6ZXNAbmVObG9jay5uZXQgZS1tYWlsIGNpbWVuLiBJTVBPUlRBTlQh
IFRoZSBpc3N1YW5jZSBhbmQgdGhlIHVzZSBvZiBOaGlzIGN1cnRpZmljYXRlIGlz
IHN1YmplY3QgdG8gdGhlIE5ldExvY2sgQ1BTIGF2YWlsYWJsZSBhdCBodHRwczov
L3d3dy5uZXRsb2NrLm5ldC9kb2NzIG9yIGJ5IGUtbWFpbCBhdCBjcHNAbmVObG9j
ay5uZXQuMBEGCWCGSAGG+EIBAQQEAwIGQDBBBgNVHR8EOjA4MDagNKAyhjBodHRw
Oi8vY3JsMS5uZXRsb2NrLm5ldC9pbmRleC5jZ2k/cmF3JmNybD11emxldGkwDQYJ
KoZIhvcNAQEFBQADgYEAZyWQdsNxEcDaCNCslletLDMMGd6Sd7qKxfPxc1lEPyqG
IGWnp1cB9wEvkgOOstIFK716iCRWWB1p/88fM8U5owjhU1Pz81/KStUfg2hIsDFV
fWzU3TCHD/7D6F/+A2NKROvh2dpkPsbmzEAd3peGYZMC/CHJgu4TgoOAaF116AQ=
-----END CERTIFICATE-----
```

13. ábra – A NetLock Kft. Class B típusú tanúsítványa

Már évek óta, mikor a kliens személyi számítógépekre telepítjük a Mozilla Firefox, Netscape Navigátor, vagy a Microsoft Internet Explorer böngészőket, egyben telepítjük a WEB szerverek tanúsítványait is a nyilvános kulcsaikkal együtt. Ez biztosítja az SSL alapú kommunikációt vagy a Java és ActiveX szoftverek digitális aláírás ellenőrzését. A csoportmunkát támogató Lotus Notes szintén a nyilvános titkosítást alkalmazza. A nyilvános kulcsú infrastruktúra licencek tekintetében az RSA kapcsolatokkal rendelkező VeriSign-t és a Northern Telecom érdekeltségű Entrust-ot érdemes megemlíteni. A következő lépés az infrastruktúra fejlesztésében a szervezeten belüli kulcs menedzsment és név/címtár szolgáltatások használatának elősegítése. Ezeket például a Microsoft Windows 2000 natívan tartalmazza.

Érdemes megemlíteni, hogy első és egyedüli magyar szolgáltatóként a NetLock Kft-t. A NetLock fokozott biztonságú főtanúsítványai már alapértelmezetten megtalálhatók a Mozilla Suite 1.8, Firefox 1.5, Thunderbird 1.5, Internet Explorer, Outlook, Outlook Express, The Bat, Eudora, PGP Universal 2.0, Kmail, Konqueror, Kleopatra és Safari böngésző és levelező szoftverekben. A NetLock az első hitelesítés-szolgáltató Magyarországon, melynek tanúsítványai Linux környezetben is alkalmazhatók és a legtöbb böngészőben alapértelmezetten megtalálhatók.

5.4. A tanúsítványok életciklusa

Ahogy sorra veszzük a tanúsítványok különböző életciklusait, úgy tárul fel előttünk az infrastruktúra működése. Három jelentős állomást különböztethetünk meg. Az első a tanúsítvány megszületése. Ezután következik a leghosszabb szakasz, mikor használjuk a tanúsítványunkat. Majd mikor elérkezik az idő, akkor visszavonásra kerül a megfelelő

igazolásunk. Most vegyük sorra ez egyes korszakokat és azok lépéseit külön-külön.

A tanúsítvány kiadás folyamata

- Az ügyfél a megszokott módon nyilvános és magán komponensekből álló kulcspárt generál magának.
- A magán komponenst biztonságba helyezi, a nyilvános kulccsal pedig jelentkezik a regisztrációs egységnél (RA), s jelzi, milyen típusú tanúsítványra van szüksége.

Ezen első két lépést az élő gyakorlatban szintén a szolgáltató végzi el, ugyanis a felhasználók széles köre és nagy többsége nem képes ezt a „munkát” elvégezni. Vagy ha el is tudják végezni, akkor nem kellő biztonsággal és körültekintéssel járnak el.

- Az RA elvégzi a személyazonosság megállapítása céljából előírt műveleteket. Magánszemély esetén ez jelentheti személyi igazolvány, útlevél, jogosítvány, vagy egyéb dokumentumok ellenőrzését, intézmények esetén pedig kérheti a cégbejegyzés másolatát, igazolását, vagy egyéb hivatalos iratokat. A művelet minden mozzanatát dokumentálja az iratokról fénymásolatot készít és archiválja azokat.
- Az azonosítás végeztével az RA az ügyfél nyilvános kulcsából valamint a tanúsítvány kiadásához szükséges egyéb adatokból elektronikus tanúsítványkérő dokumentumot formáz, digitálisan aláírja azt, majd továbbítja a hitelesítő eszköznek.
- A CA átveszi a tanúsítványkérést és ellenőrzi a digitális aláírást. Ha rendben találja, akkor biztos lehet benne, hogy a dokumentum valóban az RA-tól származik, aki már elvégezte az ügyfél azonosítását.

- A CA kiállítja a kért kulcsigazolást és publikálja a nyilvános tanúsítványtárban. Opcionálisan elküldheti az RA-nak is, aki majd továbbítja az ügyfélnek.

A tanúsítvány használata

A tanúsítvány felhasználási módja rendszerint alkalmazásfüggő. Ám, hogy mégis érzékeltetni tudjam a folyamat függését a nyilvános kulcsú infrastruktúrától, egy elektronikus bevásárlás esetében mutatom be a műveltet. Alice könyvet szeretne venni egy Internetes áruházban. A műveletnek lesznek előfeltételei melyek a következők:

- A bevásárló hely a szolgáltatást webszerveren keresztül nyújtja. A szerver rendelkezik nyilvános és magánkulccsal, valamint a hozzá tartozó tanúsítvánnyal, melyet egy nyilvános kulcsú infrastruktúrát üzemeltető, megbízhatóságáról ismert tanúsítványkiadó szervezet (hitelesített szolgáltató) állított ki a számára.
- Alice-nek birtokában van annak a magánkulcsnak a nyilvános párja, mellyel a webszerver tanúsítványát aláírták. Ez a gyakorlat a tanúsító szervezet CA komponensének nyilvános kulcsát jelenti. Alice a kulcshoz biztonságos módon jutott hozzá, például a böngésző programjába eleve be volt építve.
- Most hogy a vásárlás előfeltételei adottak, nézzük magát a vásárlás lépéseit:
- Alice böngésző programja segítségével felveszi a kapcsolatot az elektronikus áruház webszerverével, kiválasztja a megvásárolni kívánt cikkeket, majd jelzi fizetési szándékát.

- A webszerver elküldi neki a nyilvános kulcsú tanúsítványát, mely tartalmazza a cég és a szerver nevét, valamint a szerver nyilvános kulcsát.
- Alice (a kibocsátó CA nyilvános kulcsával) ellenőrzi a tanúsítványt. Ha rendben találta, akkor bizonyos lehet benne, hogy a tanúsítványból származó kulcs valóban a bevásárló helyet üzemeltető céghez tartozik, mivel a hitelesített szolgáltató a dokumentum kiállításakor a szükséges ellenőrzéseket elvégezte.
- Alice kapcsolatba lép a tanúsítványt kiadó szervezet címtárával is, és meggyőződik róla, hogy a kapott tanúsítványt nem vonták-e vissza időközben.
- Alice ezután a kapott nyilvános kulcs felhasználásával titkosított csatornát alakít ki a szerverig, és a fizetés rendezése céljából elküldi bankkártyájának adatait.

Ha a tanúsítvány rendben van és még nem vonták vissza, Alice biztos lehet benne, hogy mindaz amit a tanúsítványból származó nyilvános kulccsal titkosít, csak a bevásárló hely üzemeltetője számára lesz olvasható.

A tanúsítvány visszavonása

A visszavonások tekintetében két lényeges esetet kell megkülönböztetnünk. Az első, mikor a tanúsítványt az érvényességi idejének lejárata előtt kell visszavonni. A második, mikor a tanúsítványt az érvényességi idejének lejárata miatt kell visszavonni. Nagyon fontos mozzanatbeli különbségek vannak a két visszavonás között. Fontos hogy az egyik esetben maga a felhasználó jelzi a visszavonás tényét a másik esetben pedig a CA kezdeményezi a visszavonást.

A nyilvános kulcsú tanúsítványok minden esetben jól meghatározott érvényességi idővel rendelkeznek. Bizonyos

esetekben azonban már az érvényességi idő lejártát megelőzően is szükség lehet a tanúsítvány visszavonására, azaz érvénytelenítésére. Ilyen eset lehet, ha a tanúsítványhoz tartozó magánkulcs hozzáférhetetlenné vált, elveszett, vagy rendszerhiba miatt megsérült, ekkor a címzett nem tudja értelmezni a tanúsítványban található nyilvános kulccsal kódolt üzeneteket. Esetleg illetéktelen kezekbe került a magánkulcs (vagy fennáll ennek gyanúja), így a tanúsítvány alapján titkosított üzeneteket más is tudja olvasni, illetve a jogos tulajdonos nevében digitális aláírásokat készíthet.

A visszavonás lépései:

- Az ügyfél arra a felismerésre jut, hogy nyilvános kulcsú tanúsítványát érvényteleníteni kell.
- Értesíti a regisztrációs egységet (RA).
- A regisztrációs egység ellenőrzi a bejelentő személyazonosságát, majd digitálisan aláírt üzenetben kezdeményezi a tanúsítvány visszavonását a tanúsító eszköznél (CA).
- A CA elvégzi a tanúsítvány visszavonásával kapcsolatos teendőket, és a művelet eredményét publikálja a tanúsítványtárban.

Ebben az esetben, ha újból szeretne tanúsítványt használni az ügyfél, egy teljesen új kulcspárra lesz szüksége. Továbbá az új kulcsokhoz tartozó teljesen új tanúsítványra, melyet a tanúsítványigénylő folyamatban ismertetett módon kell megint igényelnie.

Ha viszont a tanúsítványt csak azért kell visszavonni, mert az érvényességi ideje lejárt, akkor a szolgáltatón múlik, hogy milyen megoldást javasol az ügyfélnek. Ugyanis ha a tanúsítvánnyal és a hozzá tartozó kulcspárral semmi gond nem

volt, a szolgáltató dönthet úgy, hogy megújítja a kulcsokra vonatkozó tanúsítványt. Ez mind az ügyfél szempontjából mind a CA szempontjából kényelmesebb. Az ügyfél a saját jól bevált kulcspárját használhatja tovább, a CA-nak meg nem kell kulcsgenerálással foglalkoznia. Viszont a CA dönthet úgy is, hogy kikényszeríti az új kulcsok használatát. Ilyenkor a kulcsokat a CA mindig visszavonja. Ekkor a felhasználó eldöntheti kíván-e új kulcsokat és a kulcshoz tanúsítványt igényelni. Ha igen, akkor a tanúsítvány igénylési folyamat lépéseit ismét végre kell hajtania, de mostmár könnyített konstrukcióban.

A visszavonás lépései:

- A CA észreveszi, hogy az adatbázisában egy kulcspárnak hamarosan lejár az érvényessége. Ezt jelzi a regisztrációs egységnek (RA).
- Az RA digitálisan aláírt üzentben kiértesíti az ügyfelet, és felajánlja vagy egy új kulcs elkészítésének lehetőségét vagy a régi kulcspárjának megújítását. Ez a szolgáltatótól függ.
- Az ügyfél megkapja az üzenetet, ellenőrzi, hogy valóban az RA-tól érkezett-e, majd eldönti mit is szeretne.
- Ha nem szeretne új kulcsot, vagy nem szeretné meghosszabbítani a tanúsítványát nincs semmi tennivalója. A kulcsok érvényességi idejének lejáratakor a CA automatikusan elvégzi a visszavonással kapcsolatos teendőket, és a műveletek eredményét publikálja a tanúsítványtárban.
- Ha viszont szeretné meghosszabbítani a tanúsítványt, vagy igényli az új kulcsokat, akkor ő is egy digitálisan aláírt üzenetben értesíti erről a szándékáról regisztrációs egységet.

- A regisztrációs egység most még az érvényes digitális aláírás miatt játszi könnyedséggel tudja azonosítani a felhasználót. Így súlyos regisztrációs procedúráktól mentesül az ügyfél. Majd tovább küldi a CA-nak az igénylést egy digitálisan aláírt üzenetben.
- A CA elvégzi a megfelelő műveleteket. Azaz új kulcs generálása esetén elkészíti az új kulcspárt a régieket pedig visszavonja. Megújítás esetén pedig bejegyzi a régi kulcsok mellé az új lejárati időket.
- A CA értesíti az RA-t az művelet sikeres végrehajtásáról, vagy elküldi neki a frissen generált titkos kulcsot, amit majd mostantól az ügyfélnek használnia kell.
- A regisztrációs egység a megkapott titkos kulcsot az ügyfél régi nyilvános kulcsával titkosítja, majd egy digitálisan aláírt dokumentumban elküldi az igénylőnek. Megújítás esetén elegendő csak egy digitálisan aláírt tájékoztatást küldenie.
- Az ügyfél ellenőrzi az RA hitelességét, és ha mindent rendben talál, akkor a titkosított üzenetet a régi titkos kulcsával visszafejti. Így megkapja az új titkos kulcsát, amit biztos helyre ment. A régi titkos kulcsot meg kell semmisítenie. Kulcsmegújítás esetén az igénylő tudomásul veszi, hogy a CA meddig hosszabbította meg az érvényességet.

5.5. Visszavonási listák

A tanúsítványok érvényességi idejének szerepe egész pontosan az, hogy a CA csak ennek az időnek a lejáratáig vállalja a tanúsítványok menedzselését, az érkező esetleges visszavonási és felfüggesztési kérelmek kezelését. A tanúsítványok visszavonása hagyományos módon, ún. visszavonási

listák (Certificate Revocation List, CRL) kibocsátásával történik. Ilyenkor a hitelesítő eszköz összegyűjti a beadott visszavonási kérelmeket, és szabványos időközönként nyilvánosságra hozza ezek listáját. A lista tartalmazza a visszavont tanúsítványok azonosító adatait a CA digitális aláírásával ellátva. A listát az ügyfelek letöltik és munka közben ennek alapján döntik el, hogy egy adott tanúsítványt elfogadnak-e vagy sem. A visszavonási listák kezelése felhasználói oldalon meglehetősen nehézkes és megvan az a hátránya is, hogy az ügyfelek az egyes tanúsítványok visszavonásáról nem azonnal szereznek tudomást, csak később, mikor a következő lista publikálása esedékes lesz. A közbenső időben fennáll a veszélye, hogy lesznek, akik munkájuk során olyan tanúsítványt is elfogadnak, melyek hitelessége már nem garantált. Jobb megoldásnak tűnik, ha a visszavonási listák helyett (vagy mellett) a CA (tanúsító eszköz) a visszavonás tényét azonnal jelzi a tanúsítványtárban, az ügyfelek pedig, minden egyes alkalommal, mikor kulcsigazolást készülnek elfogadni, előbb a hálózaton keresztül a tanúsítvány adatbázishoz fordulnak a kulcsigazolás státuszának ellenőrzése végett. A megoldás gyors, és a visszavonás ténye azonnal a felhasználók számára is láthatóvá válik. Hátránya, hogy on-line hálózati kapcsolatot és nyilvános tanúsítványtárat (vagy megfelelő funkcionalitású adatbázist) igényel, míg a visszavonási lista akár egy közönséges webszerverről is letölthető napi-heti egyszeri Internet kapcsolattal, a lista kibocsátásának gyakorisága függvényében.

5.6. Jogi háttér és megfontolás

A hagyományos aláírásoknak komoly bizonyító erőt tulajdonítunk mind a jogi világban, mind a hétköznapokban. Aláírásunkkal ismerjük el a szabályok tudomásulvételét, eszközök átvételét, szerződések tartalmával való

egyetértésünket. A digitális világ térhódításával viszont ezek a dokumentumok, melyek korábban papíron keletkeztek, egyre gyakrabban elektronikus formában kerülnek elénk. Az így kapott anyagot természetesen kinyomtathatjuk és szükség esetén a megszokott módon szignálhatjuk is, habár ez nehézkesebbé teszi a folyamatot. A kommunikációs technológiák, számítógépes hálózatok elterjedésével azonban életünk olyan közeggel bővült, melyen a hagyományos hitelesítési megoldások gyakran már nem alkalmazhatók. Szükségessé vált a kézi aláírás elektronikus változatának létrehozása.

A jogilag elismert elektronikus aláírás megalkotása kettős feladatot jelentett. Egyrészt technikai megoldást kellett találni az aláírás létrehozásának problémájára, másrészt ki kellett dolgozni a szükséges jogszabályi hátteret, mely az új eszközt megfelelő garanciákkal ellátva illeszti be a mindennapi élet rendszerébe. A dolog technikai háttere volt az egyszerűbb, amit már az előző fejezetekben nagyjából vázoltam.

Az elektronikus aláírás az Európai Unió szintjén szabályozott kérdés. Az Európa Tanács 1999. december 13-án bocsátotta ki saját irányelveit, s a tagállamoknak 2001. július 19-ig kellett saját joganyagukat ezzel összhangba hozni. A magyar elektronikus aláírás törvény is a fenti irányelvnek megfelelően készült, s 2001. szeptember 1-én lépett hatályba. A magyar törvényben megkülönböztetik az elektronikus aláírást, a fokozott biztonságú elektronikus aláírást és a minősített elektronikus aláírást. Ezek definíciója a következő (2001. évi XXXV. törvény):

2§ 6. Elektronikus aláírás: elektronikusan aláírt elektronikus dokumentumhoz azonosítás céljából logikailag hozzárendelt vagy azzal elválaszthatatlanul összekapcsolt adat.

2§ 15. Fokozott biztonságú elektronikus aláírás: elektronikus aláírás, amely alkalmas az aláíró azonosítására, egyedülállóan az aláíróhoz köthető, olyan eszközökkel hozták létre, amelyek kizárólag az aláíró befolyása alatt állnak, és a dokumentum tartalmához olyan módon kapcsolódik, hogy minden – az aláírás elhelyezését követően a dokumentumon tett – módosítás érzékelhető.

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.

Az Uniós szabályozás technológia független, azaz nem teszi kötelezővé a nyilvános kulcsú kriptográfia, vagy bármely más rivális megoldás (pl. ujjlenyomat vagy retina mintázaton alapuló eljárás) alkalmazását. Olyan keretrendszert határoz meg, mely valamennyi lehetséges megoldást lefedi. Mindazonáltal a nyilvános kulcsú kriptográfia az, mely a lehetséges megoldások közül a legéletképesebbnek bizonyult, így a továbbiakban ezt tekintjük irányadónak.

Mind az európai mind a magyar jogszabályok következetesen elektronikus aláírásról beszélnek, mely tetszőleges elektronikus aláírási megoldást jelent. A digitális aláírás kifejezés viszont szigorúan a nyilvános kulcsú kriptográfiát használó megközelítésre vonatkozik. A jogalkotók elképzelése szerint a felhasználók az elektronikus aláírást megfelelő kulcsok használatával állítják elő. A kulcsok hitelességét Ún. hitelesítés szolgáltatók garantálják, melyek ellenőrzik a kulcstulajdonos személyazonosságát, és vizsgálatuk eredményéért tanúsítvány kiadásával vállalnak felelősséget. A szabályozás értelmében hitelesítési szolgáltatást bárki

indíthat, a tagállamok ezt a tevékenységet nem köthetik előzetes hatósági engedélyezéshez.

A hitelesítés szolgáltatók által kiadott tanúsítványok két nagy csoportba sorolhatók. Beszélhetünk nem minősített és minősített igazolásokról. Az első esetben a szolgáltató a tanúsítványok kiadása céljából kiépíti infrastruktúráját, kialakítja saját szervezetét és eljárásrendjét, melyek létrehozásakor támaszkodik meglévő tapasztalataira, lehetőségeire, szakértelmére, piaci igényekre és az iparág által legjobbnak tartott gyakorlatra. Az ilyen közegben kiadott tanúsítványokat nem minősített tanúsítványoknak nevezzük. Amint az a fentiekből is kitűnik, nem minősített tanúsítványok kiadásának infrastrukturális háttere és eljárásrendje jelentős mértékben függ a szolgáltató saját döntéseitől.

Gyökeresen más a helyzet a minősített tanúsítványok esetében. Minősített tanúsítványt csak minősített hitelesítés szolgáltató bocsáthat ki, melynek meg kell felelnie az államilag előírt követelményeknek. Az iparág által elfogadott, legbiztonságosabbnak ítélt gyakorlat alapján a tagállamok megszabhatják azoknak az elvárásoknak a listáját, melyeket minősített hitelesítés szolgáltatóknak ki kell elégíteniük. Ezek magukban foglalják:

- A háttér infrastruktúra felépítésével és működtetésével kapcsolatos követelményeket.
- A különböző technikai eszközök elvárt biztonsági szintjének meghatározását és ezek igazolásának módját.
- A szervezet által követendő eljárásrendet.

- Az ügyfelek személyazonosságának megállapításához szükséges követelményeket és a folyamat dokumentálásának módját.
- A szerver által nyújtandó minimális szolgáltatások felsorolását.

A hitelesített szolgáltatók önkéntes alapon eshetnek át a minősítési folyamaton, s ha a vizsgálat eredményeként megfelelnek az előírtaknak, minősített hitelesítés szolgáltatóként állami nyilvántartásba veszik őket. A minősített hitelesítés szolgáltatók jogosultak minősített tanúsítványok kiadására, az előírt ügymenetet követve.

Az egyes tagállamok kötelesek a minősített hitelesítés szolgáltatók felügyeletére, ellenőrzésre, annak garantálása végett, hogy a meghatározott biztonsági követelményeket valamennyi hitelesítés szolgáltató egyformán teljesíti. A műveletekhez szükséges tanúsítványok és egyéb elvárások alapján az EU irányelvek háromféle elektronikus aláírást különböztetnek meg:

- Egyszerű elektronikus aláírás.
- Fokozott biztonságú elektronikus aláírás.
- Minősített elektronikus aláírás.

Egyszerű elektronikus aláíráson olyan, akár mindenféle technológia biztonságot is nélkülöző eljárást értünk, melynek során az aláíró, a személyazonosságára utaló információkat helyez el egy elektronikus dokumentumban. Egyszerű elektronikus aláírás esetén az aláíró személye nem határozható meg egyértelműen, s a dokumentumban bekövetkezett változások sem mutathatók ki. Ilyen típusú aláírásra jó példa, mikor egy szövegszerkesztővel írt levél végére odagépeljük a nevünket.

Az uniós szabályozással összhangban, jogvita esetén, egyszerű elektronikus aláírás figyelembe vételét nem lehet megtagadni pusztán azért, mert a szignó elektronikus formában létezik. Azon túl viszont bizonyító ereje nyilván igen csekély.

A fokozott biztonságú elektronikus aláírásnak már az előzőeknél sokkal szigorúbb követelményeknek kell megfelelnie. Létrehozása olyan eszközön történik, mely teljes egészében az aláíró felügyelete alatt áll (pl. intelligens kártya). Alkalmas az aláíró személyazonosságának megállapítására. Segítségével a szöveg legcsekélyebb mértékű megváltozása is kimutatható, ha arra az aláírást követően került sor. A feltételek nyilvános kulcsú kriptográfiára és X.509-es tanúsítványokra támaszkodva könnyűszerrel kivitelezhetők.

Ha az ügyfél rendelkezik minősített tanúsítvánnyal, kulcsait intelligens kártyákon tárolja, s az aláírásokkal kapcsolatos műveleteket is a kártyán végzi el, akkor szignója megfelel a fokozott biztonságú aláírásokkal támasztott elvárásoknak. Jogi következményeket tekintve, ha a jogszabály valamilyen területen írásba foglalást ír elő, az ilyen típusú szignóval ellátott elektronikus dokumentum megfelel az előírásoknak.

Minősített elektronikus aláíráson olyan fokozott biztonságú elektronikus aláírást értünk, ami biztonságos aláíró eszközön készült (pl. intelligens kártya), és hitelesítése céljából minősített tanúsítványt bocsátottak ki. Az ilyen típusú szignóval ellátott elektronikus dokumentum teljes bizonyító erejű magánokiratnak minősül.

A biztonságos aláíró eszköz (BALE) egy pontosan meghatározott fogalom. Csak olyan intelligens kártya tekinthető ilyennek, mely rendelkezik az ezt igazoló

dokumentumokkal. Ez legtöbbször az SSCD (Secure Signature Creation Device) védelmi profilnak való megfelelést jelenti. A 15/2001-es MeH rendelet határozza meg, hogy egy eszköz hogyan kaphat ilyen igazolást. Magyarországon a hitelesítés szolgáltatók felügyeletét a Nemzeti Hírközlési Hatóság (korábban Hírközlési Főfelügyelet) látja el. Webhelyén megtalálható valamennyi, elektronikus aláírással kapcsolatos törvény és alacsonyabb szintű jogszabály a minősített és nem minősített hitelesítés szolgáltatók felsorolásával együtt. A magyar elektronikus aláírás törvény a 2001. évi XXXV: törvény. A minősített hitelesítés szolgáltatók működésével kapcsolatos elvárások szabályozás a 2/2002. (IV.26) MeHVM irányelv.

5.7. Tanúsítvány típusok

Ha igazán körültekintően akarunk szolgáltatót választani, akkor érdemes a cég szolgáltatási szabályzatát is áttanulmányozni. A szolgáltatási szabályzat olyan dokumentum, mely részletesen ismerteti a különböző típusú tanúsítványok kibocsátásának gyakorlatát, a cég által működtetett nyilvános kulcsú infrastruktúra jellemzőit, a tanúsítványok hitelességének garanciáit. Az itt leírtak alapján győződhetünk meg arról, hogy a választott szolgáltató valóban kiérdemli-e a bizalmunkat.

Szolgáltatási szabályzatát minden hitelesítés szolgáltató cég köteles nyilvánosságra hozni, ami a gyakorlatban a dokumentum Web-en való publikálását (a szolgáltató web-szerverén történő elhelyezését) jelenti. A regisztrált hazai szolgáltatók vonatkozó szabályzatai a hírközlési hatóság szerverén is megtalálhatók. A dokumentum pontos helyét gyakran a CA tanúsítványában is feltüntetik, a Bizonyítvány irányelv (Certificate Policies) mezőben. Ha elégedettek vagyunk a szolgáltatási szabályzattal, akkor következhet az igényeinknek

megfelelő tanúsítványtípus kiválasztása. A szolgáltatók különböző fajta kulcsigazolásokat kínálnak, melyeket rendszerint különböző kibocsátó eszközzel (CA-kal) hitelesítenek. Így nem meglepő, ha ugyanaz a szolgáltató több kibocsátó CA-val is rendelkezik. A különböző CA-k fizikailag ugyanazon a hardveren is elhelyezkedhetnek. Lényeg az, hogy különböző típusú tanúsítványok hitelesítése különböző aláíró kulccsal történjen. Ebben az esetben az egyes CA-k, az aláíró kulcsok alapján, logikailag különülnek el. Az egyes tanúsítványtípusok számos vonatkozásban térhetnek el egymástól:

- A tulajdonos azonosítása eltérő módon történik. Egyik tanúsítványtípus esetén, pl. a szolgáltató megköveteli az igénylő személyes megjelenését, s az azonosítást közjegyző végzi két különböző, fényképpel ellátott hivatalos dokumentum alapján. Másfajta kulcsigazolás kéréshez viszont elegendő lehet, ha az ügyfél iratainak csak fénymásolatát küldi be. A második esetben természetesen nagyobb a visszaélés veszélye, mely a kérdéses tanúsítványtípus megbízhatóságát csökkenti.
- A felelősségvállalás mértéke különbözik. A hitelesítés szolgáltatók az általuk kiadott tanúsítványok tartalmának valódiságáért rendszerint anyagi felelősséget is vállalnak. Ez azt jelenti, hogy ha egy ügyfél részére nem valódi személyazonosságának megfelelő tanúsítványt állítanak ki, és az ügyfél ezzel visszaél, a szolgáltató, meghatározott összeghatárig, kártérítési kötelezettséget vállal. A kártérítés maximális mértéke függ a kérdéses tanúsítványtípushoz tartozó személyazonosítási eljárás szigorúságától. Alaposabb ellenőrzés esetén a kötelezettségvállalás mértéke is magasabb.

- A ügyfél típusa is különbözhet. Beszélhetünk személyes, szervezeti vagy eszköz tanúsítványról. Személyes tanúsítvány tulajdonosa konkrét fizikai személy, szervezeti tanúsítvány egy céget vagy intézményt azonosíthat, eszköz tanúsítvány pedig valamely számítástechnikai berendezés (pl. web-szerver) használatába kerülhet.
- A tanúsítványok engedélyezett felhasználási területei is különbözhetnek. Az X.509-es kulcsigazolások tartalmazzák, hogy a hitelesítés szolgáltató pontosan milyen célból állította ki azokat. Készíthetők olyan tanúsítványok, melyek csak digitális aláírások ellenőrzésére alkalmasak, mások üzenettitkosításra vagy kulcs cserére, és lehetnek olyanok, melyek valamennyi területen alkalmazhatók.

Tekintsük át vázlatosan, és a teljesség igénye nélkül, a legrégebbi hazai hitelesítés szolgáltató, a Netlock kínálatát. Itt négyféle, kereskedelmi célra szánt tanúsítvány osztályt különböztethetünk meg, beszélhetünk QA, A, B és C osztályú tanúsítványokról.

- QA osztályú tanúsítvány: Minősített tanúsítványt jelent. Tulajdonosa természetes személy lehet, akinek azonosítása közjegyző segítségével történik. Használata nagy értékű pénzügyi tranzakciók, államigazgatási műveletek, jogi következményekkel járó kötelezettségvállalások esetén ajánlott a művelet hitelesítése céljából. A felelősségvállalás felső értékhatára 50 millió forint.
- A osztályú tanúsítvány: Tulajdonosa személy, szervezet, vagy hálózati szerver lehet, melynek azonosítása közjegyző bevonásával történik. Szerver esetén természetesen a tulajdonos cég vagy intézmény azonosítására kerül sor. Használata nagy értékű pénzügyi

tranzakciók, pénzmozgást eredményező üzenetváltások, valamint szerződéskötések hitelesítése esetén ajánlott. A felelősségvállalás értékhatára 5 millió forint.

- B osztályú tanúsítvány: Tulajdonosa személy, szervezet, vagy hálózati szerver lehet. A tulajdonos azonosítása továbbra is szigorú lépéseken keresztül valósul meg, de már közjegyző bevonása nélkül. Az ellenőrzés során a szolgáltató az ügyfél személyes megjelenését és az azonosításra alkalmas eredeti dokumentumok bemutatását követeli meg. A tanúsítvány használata elektronikus levelezéshez, közepes kockázatú tranzakciókhoz, online szolgáltatások igénybevételéhez és szoftverek forrásának ellenőrzéséhez ajánlott. A kötelezettségvállalás értékhatára 500 ezer forint.
- C osztályú tanúsítvány: Tulajdonosa személy, szervezet, vagy hálózati szerver lehet, melynek azonosítása korlátozott módon történik. A regisztráció dokumentumok másolatainak beküldése alapján valósul meg, személyes megjelenés nem szükséges. Használata elektronikus levelezéshez, kisebb kockázatú tranzakciók védelméhez ajánlott. A felelősségvállalás értékhatára 50 ezer forint.

A felsorolt tanúsítvány osztályok alapján az ügyfelek a következő tanúsítvány típusokat igényelhetik:

- Személyes aláíró tanúsítvány: Tulajdonosa természetes személy lehet, és csak digitális aláírás előállítására (ellenőrzésére) használható. Az ügyfél választásától függően az QA, A, B és C osztályok bármelyikébe tartozhat.
- Személyes titkosító tanúsítvány: Tulajdonosa természetes személy lehet, s kizárólag üzenetek, dokumentumok

titkosítására használható. Az ügyfél választásától függően az A, B és C osztályok bármelyikébe tartozhat.

- Munkatársi aláíró tanúsítvány: Hasonló a személyes aláíró tanúsítványhoz, de a kulcsigazolásban a tulajdonos munkahelyének neve is feltüntetésre kerül.
- Munkatársi titkosító tanúsítvány: Hasonló a személyes titkosító tanúsítványhoz, de a kulcsigazolásban a tulajdonos munkahelyének neve is feltüntetésre kerül.
- Szervezeti aláíró tanúsítvány: Jogi személy igényelheti, A, B és C osztályokba tartozhat.
- Szervezeti titkosító tanúsítvány: Jogi személy igényelheti, A, B és C osztályokba tartozhat.
- SSL tanúsítvány: Hálózati szervert üzemeltető, domain névvel rendelkező jogi vagy természetes személy igényelheti a szerverrel való biztonságos kommunikáció megvalósítása érdekében. A, B és C osztályba tartozhat.

Ha elektronikus levelezésünket titkosítással és elektronikus aláírással is védeni akarjuk, külön titkosító és aláíró tanúsítványra lesz szükségünk. Ez összhangban van azzal az elvvel, hogy egy kulcspárra (és a hozzá tartozó tanúsítványra) ne bízzunk össze nem tartozó feladatokat (titkosítás és aláírás), még akkor sem ez technikailag lehetséges volna, hiszen az ilyen típusú megoldás biztonsági rést vihet a rendszerbe. A szolgáltató kínál még egyéb SSL, VPN, WAP GATEWAY, IPSEC, stb. tanúsítványokat és valamennyi esetben teszt osztályú kulcsigazolásokat is, a technológia kipróbálása céljából.

6. Az RSA algoritmus

Ha valaki nyilvános kulcsú infrastruktúráról beszél, nem kerülheti ki azt, hogy az RSA-ról [16] beszéljen. A mai napig ez a legelterjedtebb és legnépszerűbb algoritmus. Egyaránt használható digitális aláírásra és titkos adatcserére. Az algoritmust a Massachusetts Institut of Technology professzorai Ronald Rivest, Adi Shamir és Len Adleman szabadalmazták 1978-ban. Ez egy olyan nyilvános kulcsú kriptográfiai rendszer (Public Key Cryptosystem), amelynek biztonsága a faktorizáció problémáján alapul. Azaz a nagy prímszámok szorzásán, és az így létrejövő még nagyobb szám prímtényezőkre bontásának nehézségén. Az RSA nevet a szerzők nevének kezdő betűiből kapta ez a méltán népszerű módszer. Ezt az algoritmust az USA-ban szabadalom védte, de ez 2000-ben lejárt. Így a mai napon már bárki, jogdíj nélkül készíthet RSA-án alapuló hardver és szoftver eszközt. Pontosan az ISO 9796-os szabványa írja le 1991-óta.

14. ábra – Az RSA alkotói

6.1. Az RSA kulcsok generálása

Első és legfontosabb része a kulcsok generálása. Itt dől el, hogy mennyire lesznek majd biztonságban az adataink. Minden a kommunikációban résztvevő személynek választania kell két megfelelően nagy (minimum 100 decimális jegyű) **p** és **q** prímszámot. Ezután kiszámítja mindenki az **n = p*q** modulust, és a **φ(n) = (p-1)*(q-1)** számot. Ezeknek a számoknak az ismeretében választani kell véletlenszerűen egy **e** számot is, melynek relatív prímnek kell lennie a (p-1)-hez és (q-1)-hez is. Tehát e-nek és (p-1)-nek a legnagyobb közös osztója 1, továbbá e-nek és (q-1)-nek a legnagyobb közös osztója is 1 lesz. Emiatt, és φ(n)-nek a számolási definíciója miatt, e-nek és φ(n)-nek a legnagyobb közös osztója is 1 lesz. Most meghatározza mindenki az e szám inverzét, de moduló φ(n) osztályban. Azaz keres mindenki egy olyan **d** számot, amelyre teljesül az **e*d ≡ 1 mod φ(n)** és **1 < d < φ(n)**. Ennek a lépéssorozatnak a legvégén megsemmisítjük p, q és φ(n) számokat. Ezekre a továbbiakban már nem lesz szükségünk.

Ez mind szép és jó, de hogy lesz ebből kulcs, és hogy is kell majd titkosítani és megfejteni. A résztvevők nyilvános kulcsát **KP = (n,e)** összetartozó számpár adja, ahol n a modulus lesz és e a kitevő. Ezt a számpárt kell mindenkivel megosztani, aki számunkra levelet szeretne küldeni rejtjelezett formában. A titkos kulcsunk a **KS = d** lesz, amelyet biztonságba kell helyezni, és őrizni kell.

6.2. Üzenetváltás RSA kódolással

A kódolt üzenet küldésének magyarázásához a szokásos kriptográfiai terminológiát használom. Azaz a küldő Alice, a címzett Bob lesz, aki pedig megpróbálja feltörni vagy lehallgatni az üzenetet, Mallory. Első lépésben Alice kikeresi Bob nyilvános kulcsát a nyilvános adatbázisból, vagy ha nem

létezik ilyen adatbázis, akkor megkéri Bobot, hogy küldje le neki a publikus kulcsát. Miután Alicenek megvan a $KP_B = (n_B, e_B)$ kulcs, blokkokra szabdalja az eredeti szövegét, úgy hogy az eredményül előálló blokkoknak megfelelő számok, ne legyenek nagyobbak, mint Bob modulusa. Ebben a lépésben következett egy fontos mozzanat, szöveg helyett már számokról van szó. Ez nem jelent mást, mint a felosztott szövegrészhez valamilyen szisztéma szerint számértékeket rendelünk. Hiszen az egész titkosítás nem más, mint matematikai műveletek elvégzése. Valljuk meg őszintén, hogy ez betűkkel nem kivitelezhető. Legyenek most ezek a keletkező blokkok M_i-vel jelölve. A titkosítást $\mathbf{C = E_B(M)}$ Alice minden egyes blokknak megfelelő számra elvégzi. Ami nem jelent mást, mint hogy kiszámolja $\mathbf{C_i = (M_i)^e \bmod n}$ értékeket. Ahol természetesen az $e = e_B$ és $n = n_B$, hiszen Alice Bobnak szánja az üzenetet. A moduláris hatványozás művelténél érdemes az intelligens hatványozás módszerét használni, mert jelentős mennyiségű idő spórolható vele. Alice a C_i blokkokat, helyesebben az azoknak megfelelő számértékeket, egymás mellé helyezve elkészíti a kriptoszöveget és elküldi Bobnak. Ez a kódolt üzenet halad át nyilvános kommunikációs csatornán.

Bob most megkapja a kriptoszöveget, és nyomban neki is lát a megfejtésnek. Mivel tudja, hogy az érkezett üzenet egy C_1, C_2, C_3,… számsorozat, melynek tagjaira igaz hogy $\mathbf{0 \leq C_i < n_B}$, ezért ilyen darabokra tördeli szét azt. Minden egyes C_i tagra külön el fogja végezni a visszafejtés $\mathbf{D_B(C) = D_B(E_B(M)) = M}$ műveletét. Ami azt jelenti, hogy $\mathbf{(C_i)^d \bmod n = M_i}$ kifejezéseket számítja ki, ahol $d = d_B$ és $n = n_B$. Azaz a saját titkos kulcsát és a hozzá tartozó modulust használja fel. Ezek után már csak a blokkok visszaállítása marad hátra. El kell végezni a blokkoknak megfelelő számértékek visszaállítását betűkké vagy betűcsoportokká. Majd sorrendben egymás mellé helyezve ezeket

a visszatranszformált blokkokat, már olvashatóvá is válnak Alice sorai.

Sikerült kódolni és dekódolni az adatainkat, de az RSA többre is képes, segítségével a dokumentumainkat alá is tudjuk írni. Ennek a folyamata eltérő az előbb vázolt rejtjelezéstől. Alicenak az első lépése, most egy megfelelő egyirányú függvény keresése lesz. Majd ezzel az egyirányú függvénnyel elkészíti az M üzenet **MD** kivonatát. A kivonatra egy nagyon fontos megszorítás érvényes, hogy nem lehet nagyobb az $\mathbf{n_A}$ modulusnál. Ezután kerül sor a $\mathbf{DS = D_A(MD)}$ aláírási műveletre, amely nem mást jelent, mint $\mathbf{DS = (MD)^d \bmod n}$ kifejezés kiszámítását. Ekkor természetesen $d = d_A$ és $n = n_A$, vagy másképpen mondva Alice a saját tikos kulcsát használja, a hozzá tartozó modulussal. A lényegi résszel meg is vagyunk.

Elküldjük az eredeti M üzenetet és DS-t Bobnak. Mivel az egyirányú függvény közös megegyezéssel lett kiválasztva, ezért azt Bob is ismeri. Így el tudja ő is készíteni, a megkapott M üzenetből az MD kivonatot, nevezzük most ezt MD_1-nek. Majd Bob kikeresi az adatbázisból, vagy megszerzi valamilyen úton, Alice nyilvános kulcsát $\mathbf{KP = (e_A, n_A)}$. Ekkor tudja ellenőrizni azt az aláírást, amelyet Alice küldött, DS megfejtésével. Ehhez el kell végeznie a $\mathbf{(DS)^e \bmod n = MD_2}$ művelet kiszámítását, ahol $e = e_A$ és $n = n_A$. Vagy egyszerűsített formában felírva, a következő képlet igaz $\mathbf{E_A(DS) = E_A(D_A(MD)) = MD_2}$.

Bobnak mostmár minden információ a rendelkezésére áll, hogy ténylegesen leellenőrizze az aláíró személyét. Összehasonlítja MD_1 és MD_2-őt. Ha megegyeznek, akkor az aláíró tényleg Alice volt és biztos, hogy azt a dokumentumot írta álá, amit megkapott Bob.

Az egyirányú függvény, csak akkor produkálja ugyanazt az üzenetkivonatot, ha a dokumentum teljes mértékben ugyanaz. Ha Mellory azzal próbálkozna, hogy kicserél az átküldött üzenetben akár egy betűt is, akkor más kivonatot kap Bob és az összehasonlításnál lebukik Mellory. Láthatóvá válik, hogy a dokumentumot valaki megpróbálta módosítani. Az ellenőrzést mindenki végre tudja hajtani, aki hozzáfér Alice nyilvános kulcsához. Így széles körben alkalmazható ez az aláírás.

Ahhoz, hogy belássuk az algoritmus tényleg működik, már csak egy apró lépés hiányzik. Be kell látnunk, hogy titkosítás és a visszafejtés kommutatív, azaz mindegy melyik műveletet végzi el Bob és melyiket Alice. A nagy kérdés, ami bizonyításra szorul, hogy **E(D(M)) = D(E(m)) = M** egyenlőség teljesül-e. Kibontva ezt az egyenlőséget az $(M_i)^{e*d} \bmod n = M_i$ egyenlőséghez jutunk el, amely bármely i esetén igaz éréket kell hogy szolgáltasson. Mivel $e*d \equiv 1 \bmod \varphi(n)$, ami az ismertetett Euler tételéből következik, így bizonyítást nyert a titkosítás és visszafejtés kommutativitása.

6.3. Támadás az RSA ellen

Közelítsük meg az RSA algoritmust, most Mellory szemszögéből. Vizsgáljuk végig milyen módszerekkel, lehetne megkísérelni a feltörést, és milyen figyelmetlenségeket lehetne kihasználni, az üzenetek zavartalan olvasásához. Az egyik legkönnyebb dolga akkor van a támadónak, ha Alice és Bob modulusa megegyezik. Ha $n_A = n_B$, akkor a nyílt szöveg blokkjai nagyon egyszerűen meghatározhatók.

Ettől talán csak akkor van egyszerűbb dolga Mellorynak, ha az exponenseket, d-ét és e-ét nagyon picire választja a generáló. Ebben az esetben a teljes nyílt szöveg megfejthető minimális számolási kapacitás mellett is.

Ha nem figyel oda Alice vagy Bob és p–q különbsége nem lesz kellően nagy, akkor egy négyzetszám teszteléssel rábukkanhat Mellory n-nek két prímfaktorára. Mondanom se kell, hogy ez szintén a nyílt szövegek teljes megismerését jelentené. További figyelmetlenség lehet az is, ha Alice és Bob rendszeresen küldi el ugyanazt a titkosított üzenetet, vagy több helyen azonos kódolt üzenetet használ. Jellemző lehet ez az elköszönésekre, vagy megszólításokra. Ekkor a kínai maradéktételt felhasználva, a támadó visszanyerheti az eredeti szöveget.

Ezeken felül egy lehetséges figyelmetlenség esetén előfordulhat, hogy Alice vagy Bob, ugyanazt a kulcsot használja rejtjelezésre és aláírásra is. Ekkor vannak olyan esetek, amikor Mellory vissza tudja nyerni a küldött üzeneteket.

Beszélni kell még egy bonyolultabb feltörési formáról is. Ez az iterációs támadás, aminek a lényege az a lépéssorozat, mely $\mathbf{C_k = (C_{k-1})^e \bmod n}$ iterációs lépéssel írható le. Ebben iterációban C_0 egy kriptoszöveg blokk, a k = 1,2,… és az egész iterálás addig tart, míg rá nem talál Mellory egy másik kriptoszöveg blokkra. Azaz $C_k = C_j$ esetén a támadó megtalálja a C_{k-1} nyílt szöveg blokkot.

Ezeket a támadási módokat megfelelő paraméterválasztással és kellő odafigyeléssel elkerülhetjük. Ekkor csak a faktorizáció marad, amelyre nem ismeretes jelen pillanatban olyan gyors algoritmus, amivel a feltörés a rendelkezésre álló számítási kapacitással, belátható időn belül megtehető.

Vegyük sorra akkor, hogy mire is kell figyelni a paraméterek megválasztásánál. A legfontosabb, hogy p és q számokat megfelelő nagyságrendűnek válasszuk. Ez azt jelenti,

hogy legalább 10^{100} legyen a méretük. A méret mellett fontos a két választott szám különbségének nagysága is. Ennek méretéről megoszlanak a vélemények, de legalább 10^{95} nagyságú legyen. A (p±1) és a (q±1) számoknak csak nagy prímosztói legyenek. Az e kitevőnket úgy válasszuk meg, hogy multiplikatív rendje φ(p), φ(q) modulusokra nézve megfelelően nagy legyen. Ez azt jelenti, hogy ne létezzenek olyan k és t kisszámok, hogy $e^k \equiv 1 \mod \varphi(p)$ és $e^t \equiv 1 \mod \varphi(q)$ kongruencia teljesül. Végezetül még egy apró kitétel az e paraméter választására. Az e > log(n) egyenlőtlenségnek teljesülnie kell.

Ami nekünk most logikus, egyszerű kis matematikai alkalmazást jelent, az a megalkotóknak több éves kutatómunkát és milliónyi zsákutcát jelentett. Ha az RSA paramétereit jól választjuk meg, akkor a kriptoanalízis különféle módszereit alkalmazó feltörési kísérletek eredménytelenek maradnak. Eddig sem bizonyítani, sem cáfolni nem tudták az RSA biztonságát, de több mint húsz éve folyó intenzív feltörési kísérletek sorozata azt sugallja, hogy az algoritmus jól működik és biztonságos.

7. A program bemutatása

A diplomamunkám szerves részét képezi, egy kis alkalmazás kifejlesztése, amely a gyakorlatban igyekszik bemutatni, a nyilvános kulcsú infrastruktúra működését. Természetesen nem beszélhetünk professzionális rendszerről, amely megfelel minden biztonsági kritériumnak, és ami teljes funkcionalitással működik. A cél nem is az volt, hogy megvalósításra kerüljön egy professzionális hitelesítés szolgáltatást nyújtó alkalmazás, hanem hogy a program sematizált módon bemutassa azt, amiről az elméleti témakidolgozás szól.

7.1. Szükséges környezet, technológia

A program JAVA nyelven készült, így futtatásához nélkülözhetetlen egy JVM (Java Virtual Machine) a kliens számítógépen. Javasolt J2SE Runtime Environment (JRE) 1.5, vagy magasabb verzió beszerzése, ami tartalmazza a szükséges JVM-et. Az alkalmazás az alkalmazott technológia miatt elméletileg, platformfüggetlen. A használt fejlesztői környezet JBoss Eclipse v3.2 volt. Szükségünk lesz a CA üzemeltetéséhez egy alkalmazás szerverre. Ehhez javasolt JBoss Aplication Server 4.0.4-es verziójának telepítése, vagy ettől frissebb és újabb kiadás üzembeállítása. Az alkalmazás szerver mögé kerüljön egy Oracle adatbázisrendszer és konfiguráljuk is be, hogy a 1521-es alapértelmezett (default) porton szolgáljon ki. Ehhez az OracleXE-t tudom ajánlani. A szerver az adatbázist HIBERNATE-es technológiával éri majd el és így kommunikál vele.

A szükséges technikai eszközökben közös az a tulajdonság, hogy mindegyik kivétel nélkül ingyenesen beszerezhető, és jogtisztán használható.

7.2. Alkalmazás indítása, inicializálás

Az alkalmazáshoz az adatbázist alapállapotba kell hoznunk. Ehhez le kell futtatnunk a mellékelt SQL szkripteket, meghatározott sorrendben. Így kialakul a megfelelő adatbázis szerkezet és inicializálódik a CA adataival az adatbázis.

1. sequence.sql
2. pkiuser.sql
3. pkikeys.sql
4. pkicert.sql

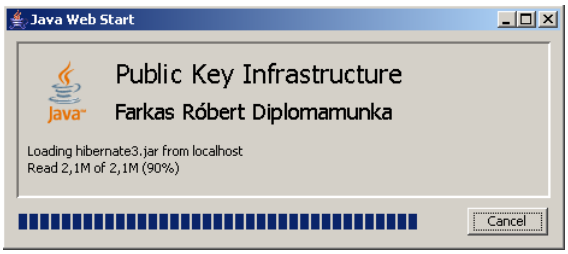
Mivel RMI hívásokkal fog kommunikálni a kliens a szerverrel, ezért el kell indítani a szerver oldalon egy rmid.exe és egy rmiregistry.exe nevű fájt, a megfelelő paraméterekkel. Ezeket a fájlokat a Java/Jre/bin könyvtárban találjuk meg. Ezek elindításához javasolnám, hogy Eclipse alatt nyissuk meg az egész Projectet, és az ExternalTools menüponton belül keressük ki a fájlokat. Itt már rendelkezésre állnak a futáshoz a szükséges paraméterek. Ezek megfelelő beállításával nem kell törődnünk. Ezután a JBoss alkalmazásszerverünk server könyvtárában meg kell keresnünk a deploy könyvtárat. Ide be kell másolnunk a Pki.ear és a Pki.war fájlokat. Mikor elindítjuk a szervert, fel fogja ismerni ezeket a fájlokat és beregisztrálja magának az alkalmazásunkat. Mostmár csak meg kell nyitnunk egy webböngésző programot. Ide kell beírnunk a következőt:

http://host:8080/Pki/

A host természetesen az a host, ahová telepítettük a JBoss alkalmazás szerverünket, és ahová felkonfiguráltuk az alkalmazásunkat. Vigyáznunk kell a kis és nagybetűkre, mivel a JBoss erre érzékeny (Case sensitive). Innentől kezdve mindent elintéz nekünk a JAVA Web Start. Ekkor feldobódik egy ablak, amelyben kiválaszthatjuk, hogy menteni, vagy megnyitni

szeretnénk-e az alkalmazást. Ezután ha kiválasztottuk a megnyitást, akkor láthatjuk, ahogy a kliens oldali eszközök települnek a gépünkre, majd ezután megkérdezi tőlünk a rendszer, hogy engedélyezzük-e a futtatást. Azaz megbízunk-e a program készítőjében és engedélyezzük-e, hogy az alkalmazás elinduljon. Ha ez az első futtatás, akkor még arra is rákérdez, hogy akarunk-e parancsikont az asztalra helyezni.

15. ábra – Web Start 1



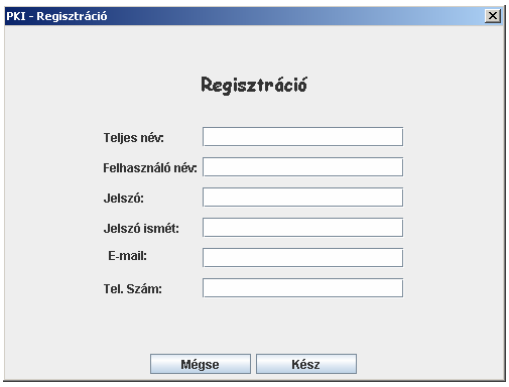
16. ábra – Web Start 2

7.3. Regisztráció, belépés

Miután elindítjuk a programot egy bejelentkezési képernyő fog megjelenni. Ez a képernyő szolgált az azonosításra. A szolgáltató el tudja dönteni, hogy jogosultak vagyunk-e igénybe venni a szolgáltatást, azaz regisztrált felhasználók vagyunk-e már.

Ha szeretnénk igénybe venni a valós életben egy ilyen szolgáltatást, akkor a hitelesítés szolgáltatónál jeleznünk kell ezt az igényünket, és minden esetben igazolnunk kell magunkat. Jelen esetben erre szolgál a **Regisztráció** gomb. A gomb lenyomása után ki kell töltenünk a megfelelő űrlapot és azt a **Kész** gomb megnyomásával, feldolgozásra kell továbbítanunk a szolgáltatóhoz. A feldolgozás sematizált az

alkalmazásban. Egy távoli metódus hívás, amely ellenőrzi, hogy szerepel-e már ilyen nevű felhasználó a rendszerben. Ha nem, akkor a szolgáltató már nem kér semmilyen további azonosítást (pl. személyi igazolvány fénymásolat, útlevél, stb.), hanem egyből nyugtázza a felhasználót.



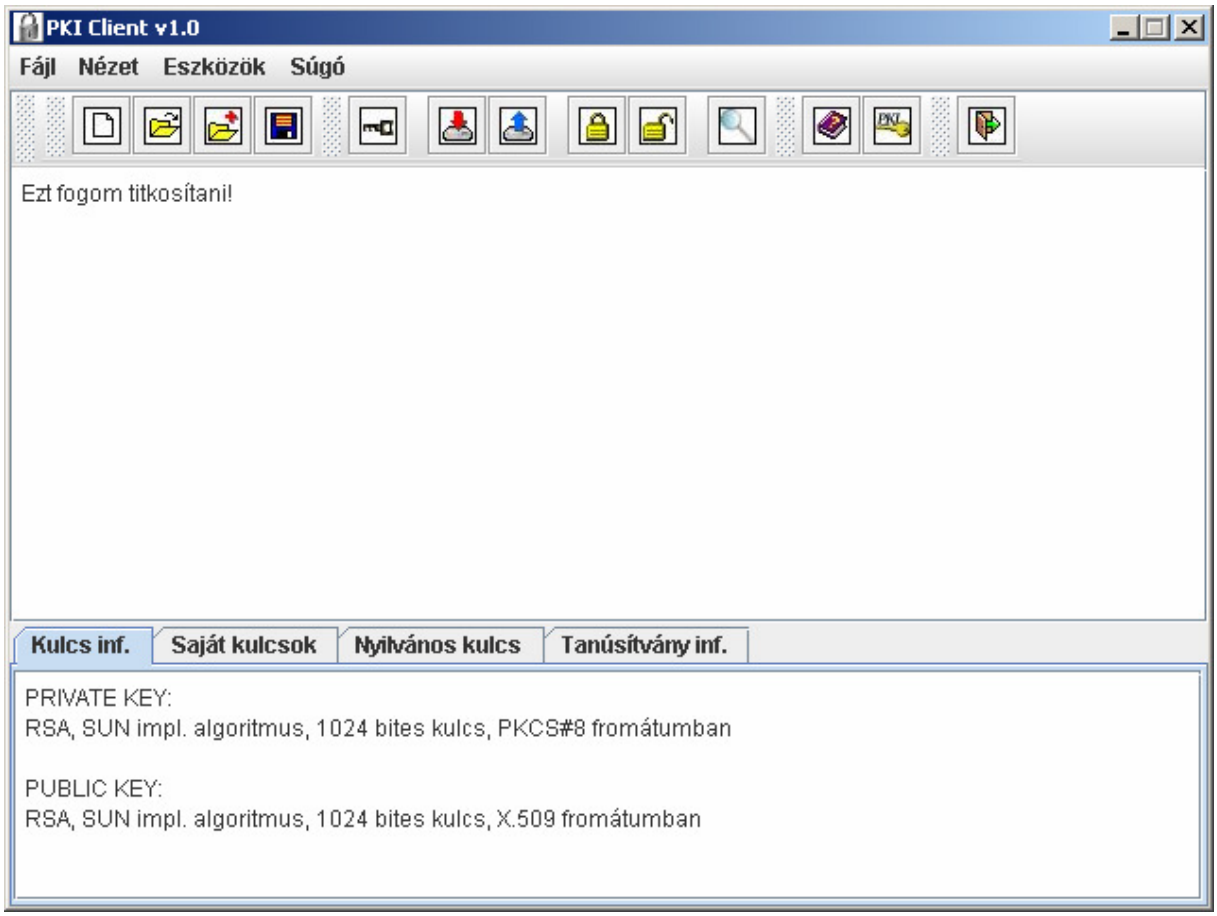
17. ábra – Bejelentkezési képernyő **18. ábra** – Regisztrációs képernyő

Ezután már igénybe vehetjük a szolgáltatást. A bejelentkezési képernyőn bekérésre kerül egy felhasználó név és a hozzátartozó jelszó. Miután rányomunk a **Rendben** gombra egy távoli metódus hívás zajlik le, mely a szerver oldalon indít el egy lekérdezést. A szerver az adatbázisban megkeresi a felhasználót és a hozzátartozó jelszót és összehasonlítja a kapottal. Ha a felhasználó nem szerepel az adatbázisban, vagy ha hibás a megadott jelszó, azt a szerver jelzi a kliens felé. A kliens ekkor egy erre megfelelő hibadobozt jelenít meg. Ha sikeresen megtörténik az azonosítás, akkor továbbléphetünk a konkrét alkalmazás használatára.

Ha pedig úgy döntenénk, hogy nem szeretnénk igénybe venni a szolgáltatást, a **Mégsem** gombra nyomva bezárhatjuk a programot.

7.4. Az alkalmazási felület bemutatása

Miután beléptünk az alkalmazásba vegyük azt használatba. Szemünk elé tárul egy menüsor, amely a Fájl, Nézet, Eszközök, Súgó füleket tartalmazza. A menüket lenyitogatva találunk menüpontokat, melyek mellett rendre megtalálhatók az őket reprezentáló ikonok is. A menüsor alatt egy színes eszköztár helyezkedik el, amely a menük tartalmainak gyorsított elérésére szolgál. Az eszköztár elemei tetszőlegesen ki és bekapcsolhatók. Ezt követően találjuk meg a szövegbeviteli mezőt, ahová majd a titkosítani kívánt szövegünket gépelhetjük be. Vagy ebbe a szövegbeviteli mezőbe nyithatjuk meg és láthatjuk a kapott, titkosított üzeneteinket. Legalul információs bokszokat találunk, amelyek a különböző kulcsokról szolgáltatnak fontos és hasznos információkat.



19. ábra – Kliens képernyő

A felület több szempontból is felhasználóbarát. Hosszasan elidőzve az egér mutatójával az egyes elemek fölött rövid szöveges üzenetet kapunk. Ebben az üzenetben tájékoztatást kapunk az ikon, gomb, menüelem funkcióját illetően és hogy mit is takar az valójában. Az ikonválasztást is igyekeztem olyan módon megoldani, hogy tükrözze a mögötte megbúvó műveleteket.

7.5. Fájl menü eszközei

A fájl menühöz kapcsolva találjuk meg a más rendszerekből is jól ismert funkcionalitásokat és azok ikonjait. Itt lelhető fel minden olyan eszköz, amelyek kezelni tudják a meglévő, vagy éppen készülő szöveges, vagy titkosított fájljainkat.

- Az „Új lap” menüelem egy teljesen új, üres szövegbeviteli mezőt hoz létre és bocsát rendelkezésünkre.
- A „Megnyitás” menüelem a már meglévő TXT fájlok között enged tallózni és kiválasztani egyet. Majd ennek tartalmát egy tiszta üres szövegbeviteli mezőn jeleníti meg.
- A „Hozzáfűzés” menüelem hasonló a megnyitáshoz, de a már begépelt szövegünk végéhez illeszti a kitallózott TXT fájl tartalmát.
- A „Mentés” menüelem elmenti a háttértárra a szövegbeviteli mező tartalmát, a korábban megadott névvel és TXT kiterjesztéssel. Ha nem adtunk meg még nevet, akkor első alkalommal azt is bekéri.
- A „Mentés mint…” menüelem funkcionalitását tekintve teljesen megegyezik a mentéssel, de itt mindig kötelező megadni milyen néven kívánjuk menteni és hová a szövegünket.

- A „Kilépés” menüelem lezárja az alkalmazásunkat. És kiléptet a teljes rendszerből.

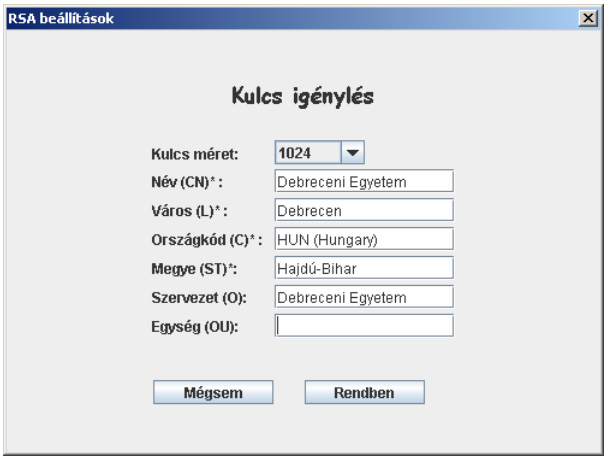
7.6. Nézet menü

Ennek a menünek a használata roppant egyszerű. Itt csak úgynevezett jelölőnégyzetek (checkboxok) találhatók. Alapállapotban mindegyik be van jelölve. Ha kikapcsoljuk valamely jelölést, akkor a hozzá tartozó eszköztárat kapcsoljuk ki. Azaz a megfelelő funkciókhoz kapcsolódó gyors elérést segítő ikonok eltűnnek az ikonsorból. Újbóli bejelölésre viszont ismét láthatóvá, elérhetővé vállnak.

7.7. Eszközök menü

Ez a menü szolgáltatja a program motorját. Ezen eszközök segítségével tudjuk kezelni különböző kulcsainkat, és ezek által vehetjük igénybe a PKI által nyújtott szolgáltatásokat.

- A „Kulcs igénylés” menüpont az első és egyben legfontosabb pont. Ha teljesen friss felhasználók vagyunk, akkor mielőtt használni tudnánk az alkalmazást, igényelnünk kell egy kulcspárt saját magunknak. Ezt itt erre a menüpontra kattintva tehetjük meg. Ha már igényeltünk korábban kulcsot, akkor a szerver figyelmeztet, hogy már van érvényes kulcsunk, amit meg is tekinthetünk a „Saját kulcsok” nevű tájékoztató mezőben. A kulcsokról bővebb információt a „Kulcs inf.” nevű fülön kapunk. A kulcsigénylés azért fontos, mert ekkor



20.ábra – Kulcs és tanúsítvány adatok

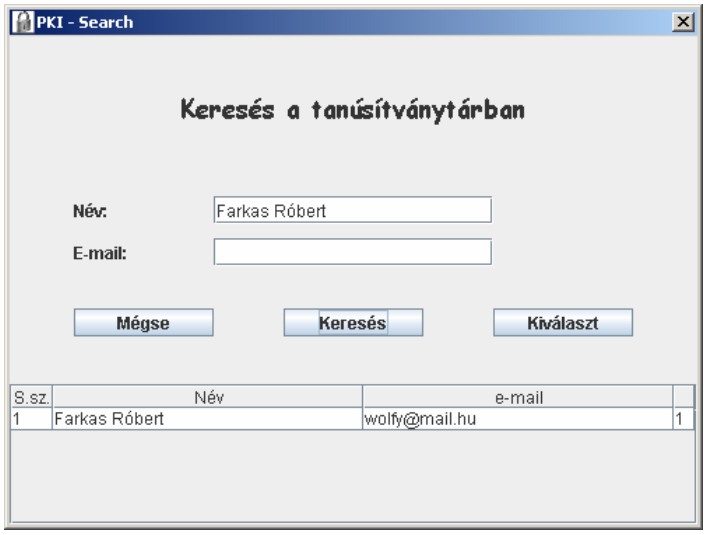
egy távoli metódushívással a szerveren legenerálódnak a kulcsaink teljes biztonságban. A kulcsigénylésnél meg kell adnunk azokat az adatokat is, amelyek majd szerepelni fognak a tanúsítványunkban. Ezek alapján a szerver kiállít a nyilvános kulcsunkhoz egy sematizált tanúsítványt is, amely bekerül a tanúsítványtárba. Ezután, ha valaki később szeretne titkosított üzenetet küldeni nekünk, akkor már megtalálja a nyilvános kulcsunkat. Ezzel tud majd titkosítani oly módon, hogy csak mi legyünk képesek azt visszafejteni.

- A következő két menüelem, ikon nem aktív. Java nyelv bizonyos korlátai miatt, az idő rövidsége miatt és mert nem ez volt a fő cél, már nem került implementálásra. Ez a két eszköz lett volna, mely felkészítené az alkalmazást különböző szolgáltatók kulcsainak kezelésére. De jelezni kívántam, hogy esetleges továbbfejlesztési lehetőségnek ez egy nagyon jó kiindulás lehet. A „Tanúsítvány importálás” menüelem segítségével egy más szolgáltató által előállított kulcsot és a hozzá tartozó tanúsítványt importálni lehetne a programba. A másik „Tanúsítvány exportálás” menüelem pedig arra hivatott, hogy az alkalmazásból kulcsot és hozzá tartozó tanúsítványt lehessen exportálni.

- A „Titkosítás” menüelem a begépelt, vagy megnyitott szöveget fogja titkosítani a „Nyilvános kulcs” mezőben található kulccsal. Itt annak a személynek a nyilvános kulcsa látható, akinek szeretnénk küldeni az üzenetet. Ezt a kulcsot előzetesen ki kell keresni a tanúsítványtárból. Ha ezt nem tettük volna meg, és úgy nyomunk rá erre a menüpontra, akkor felajánl az alkalmazás egy keresést a tanúsítványtárban.

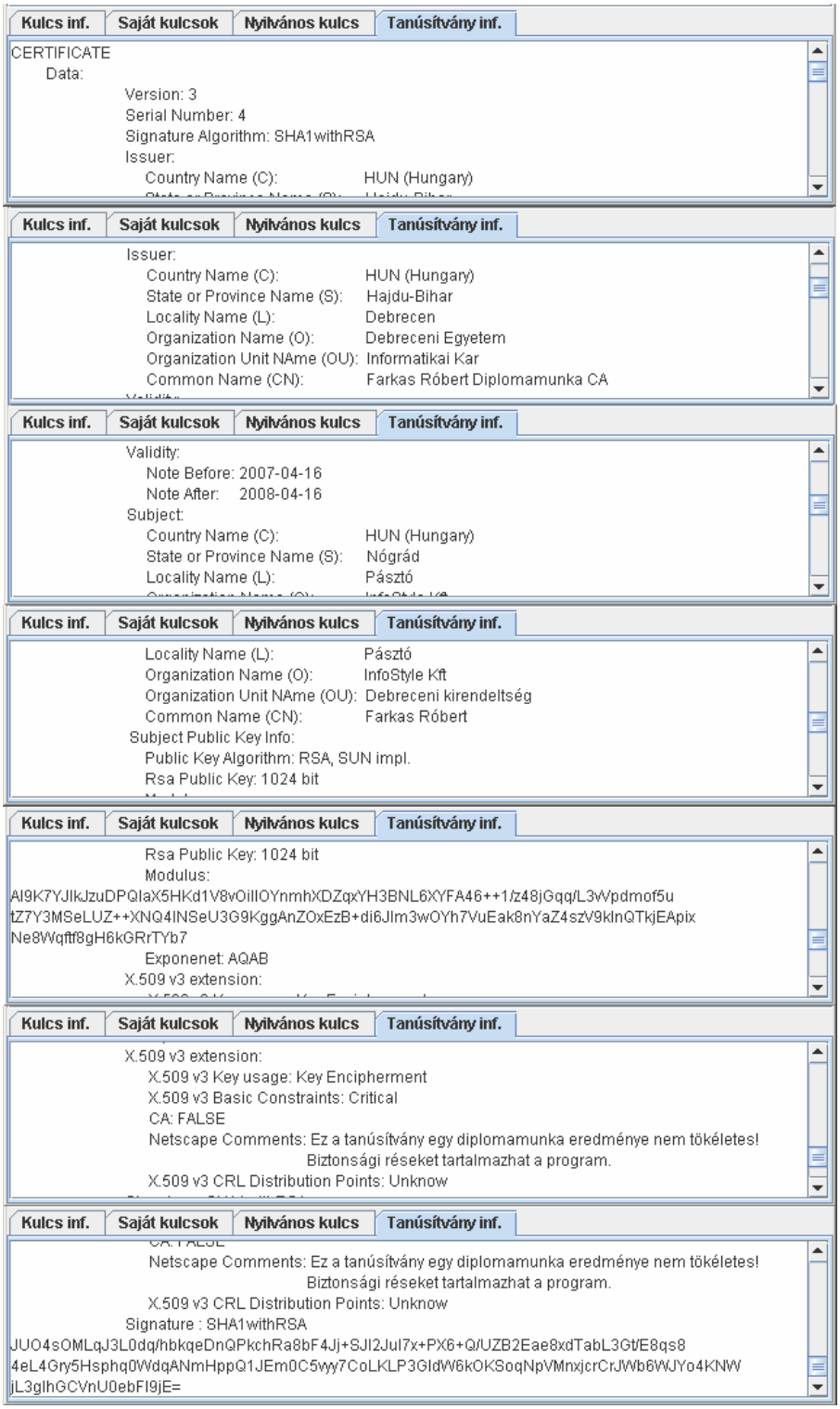
- A „Visszafejtés” menüelem a kapott, és a beviteli szövegmezőben megtalálható titkos üzenetet fogja visszafejteni. Ehhez minden esetben a saját titkos kulcsunkat fogja használni. Ekkor előfordulhat, hogy a szöveg legvégére néhány „□” szimbólum kerül beillesztésre. Ez azért következik be, mert a titkosítás blokkosítása előtt végbemegy egy kiegészítés (padding). Mindig blokkméretre egészül ki az utolsó blokk is, ha nem volt elegendő karakter. Ezután, ezekről a karakterekről már nem tudom visszafejtésnél eldönteni, hogy értékes karakter-e vagy, kiegészítő karakter. Így amennyi karakterrel kiegészült az utolsó blokk annyi kiskocka jelenik meg a visszafejtett szövegünk után. Ezek száma természetesen függ a blokkmérettől.

- A „Keresés a tanúsítványtárban” menüelem segítségével tudunk rákeresni valakire. A keresésnél megadhatunk e-mail címet vagy teljes nevet, esetleg mindkettőt. A keresési feltételnek megfelelő találatok egy táblázatban jelennek meg. A keresés szintén egy távoli metódushívással zajlik, így biztosítva van, hogy az adatbázishoz csak a szerver szabályozott módon férjen hozzá. A táblázat elemeiből választhatjuk ki azt, akit keresünk, és akinek a nyilvános kulcsára szükségünk van. A kereséssel egyúttal letöltődnek a kulcshoz tartozó tanúsítvány adatok0



21. ábra – Keresés ablak

is, amik a „Tanúsítvány inf.” fülecskén lesznek elérhetők. És a biztonság kedvéért digitálisan aláírásra kerül a szerver által.



22. ábra – A program által kiadott tanúsítvány

7.8. Súgó menü

A súgó menüpont hivatott arra, hogy elmagyarázza a program működését, útmutatást és segítséget adjon az alkalmazás rendeltetésszerű használatához. Továbbá információk nyerhetők magáról az alkalmazásról, és annak készítőiről.

- A „Tartalom” menüelemre kattintva kaphatunk rövid ismertetőt az egyes funkcionalitásokról. Ez az alapértelmezettnek beállított webböngészőnkben nyílik meg. Itt ismerkedhetünk meg röviden a program működésével, menüpontoknak megfelelően fejezetekbe szervezve a jobb áttekintés kedvéért. Elmagyarázva azt, hogy az egyes menüpontok alatt elhelyezkedő menüelemek mögött milyen művelet húzódik meg. Ha kiválasztunk egyet, és arra rányomunk, akkor mi fog történni. Mire számítson a felhasználó.

- A „Névjegy” menüpont alatt található az alkalmazás névjegye, amelyből információk nyerhetők a program keletkezéséről és készítőiről. Továbbá nagyban megtekinthető a program logó képe is.

7.9. Összegzés, fejlesztési lehetőségek

Összességében elmondható, hogy a kifejlesztett terméknek vannak még nem is kis hiányosságai, de mint említettem ez egy sematizált alkalmazás. Diplomamunkám célja nem egy professzionális rendszer készítése volt. Az eddig elkészült verzió viszont már tudja azokat a dolgokat, amit egy hitelesítés szolgáltatótól elvárhatunk:

- Rendelkezik regisztrációs eljárással.
- Megfelelően biztonságosan generál nekünk egy kulcspárt a szolgáltató.

- Kiállítja a nyilvános kulcshoz tartozó tanúsítványt, és azt publikálja is.
- A tanúsítványtárban biztosítja a keresés lehetőségét.
- Biztosítja a kulcsok használatát, tehát lehetséges azok használata titkosításra és visszafejtésre.

A hiányosságok közül csak néhányat említenék meg. Ezek szerintem a legégetőbb problémák és ezek teszik legsebezhetőbbé az alkalmazást. Törekedtem arra, hogy mindegyik hiányosság kezelésére tegyek javaslatot is.

A program talán egyik legnagyobb gyengesége, hogy a szerver és a kliens között a kommunikáció nem titkosítva zajlik. Ezért a kommunikációs vonal lehallgatása esetén nem biztosítja az átküldött adatok sértetlenségét és nem véd az adatlopás ellen. Ezt legegyszerűbben úgy lehetne megoldani, hogy miden kommunikációt a már meglévő titkosító eljárással (RSA), kódoljuk. A szerver nyilvános és titkos kulcsai adottak, mely az adatbázis inicializálásakor létrejön. Így ezeket használva a probléma könnyen áthidalható lenne.

Egy fontos és szintén nagy hiányossága az alkalmazásnak, hogy a kulcsunkat nem tudjuk visszavonni. Bár a lehetőséget erre megteremtettem, mert az adatbázisban létezik egy logikai mező, amely ennek a jelzésére szolgál. Csak le kellene programozni az ide illő részeket, és kezelni kellene ennek a figyelését az alkalmazásunkban.

A felhasználók kezelésénél is lehetne finomítani a rendszeren. A regisztrációnál ellenőrizni kellene valamilyen formában a bekért adatok helyességét, és lehetőséget kellene biztosítani a felhasználónak arra, hogy törölje magát a rendszerből. Persze ez a törlés, mint minden nagy alkalmazásban szigorúan csak logikai lehetne. Ennek a

lehetősége is adott, mert az adatbázisban már erre is létezik megfelelő logikai mező. Az adatok ellenőrzésére a JAVA biztosít megfelelő eszközrendszert a java.util.regex csomagban. Ebben a csomagban reguláris kifejezések kezelésére van lehetőség, amelyeket nagyon jól lehet majd utána mintaillleszteni.

Az alkalmazást lehetne webes kommunikációra képessé tenni. Így nem kellene leveleinket TXT fájlokba menteni, és ezután ezt a TXT fájlt egy levelező programmal továbbküldeni. Hanem egyből a felületről elektronikus levél formában elküldhetnénk a címzettnek a titkosított üzenetet. Ehhez már meglehetősen nagy átalakításra, és rengeteg programozói munkára lenne még szükség.

8. Összefoglalás

Diplomamunkám témája a Nyilvános Kulcsú Infrastruktúrák bemutatása. Megközelítését az alapoknál kezdtem és törekedtem olyan szakszavak és kifejezések elmagyarázására, amely nélkül nem lehetséges a kriptológiának a megismerése. Leírásra került, hogy milyen hosszú fejlődési utakon keresztül jutott el a szakma oda, ahol most tart. Ezáltal időben jól elhatárolhatóvá váltak az egyes korszakok, és az azokra jellemző titkosítások, technikák.

Elemzem és bemutatom a nyilvános kulcsú infrastruktúra hátterében megbúvó aszimmetrikus kriptorendszert. Helyenként összehasonlítva a konkurens szimmetrikus rendszerrel. Továbbá kiemelésre kerültek olyan pontok amelyek, ez előnyére vagy a hátrányára vállnak. Ezzel is segítve egy lehetséges felhasználót abban, hogy el tudja dönteni vajon érdemes-e neki, ezen infrastruktúrát választania, használnia.

Bemutatom azt is, hogy a szolgáltatók milyen rendszerben és milyen módon üzemeltetik a rendszerüket. Milyen szigorú törvényi és minőségi elvárásoknak kell eleget tenniük ahhoz, hogy a **minősített** nevet kiérdemelhessék. Ismertetésre került, hogy a szolgáltatók milyen szabvány szerint állítanak ki tanúsítványokat, és ezeknek milyen mezőket kell tartalmazni és a mezőknek mi is a szerepük. Továbbá, hogy milyen mezőket érdemes alaposan megvizsgálni ahhoz, hogy kijelenthessük: megbízunk a másik félben és elfogadjuk őt hiteles személynek.

Írok a kulcsok és tanúsítványok tárolásának módjáról, azokról az eszközökről, melyek egyáltalán a tárolást lehetővé teszik. Megfogalmazom a kulcsok használatában rejlő kockázatokat, és azokat a teendőket, amelyek egy kulcs beszerzése, tárolása során felmerülhetnek.

Foglalkozom a legnépszerűbb aszimmetrikus algoritmus, az RSA bemutatásával és elemzésével. Kitérek a működése mögött meghúzódó, megalapozott matematikai háttérre. Feldolgozásra került, hogy az algoritmust milyen esetekben, milyen módszerekkel lehet megpróbálni feltörni. Rámutatok arra is, hogy ezeket hogyan lehet elkerülni.

Egyes fejezetekből kiderül, hogy Magyarország teljes mértékben felkészült a nyilvános kulcsú infrastruktúrák kezelésére. Mind a törvényeink, mind a szolgáltatást nyújtó hitelesítés szolgáltatóink megfelelő minőségűek. Ennek bizonyítéka, hogy a NetLock Kft. főtanúsítványa mára bekerült a legnépszerűbb böngészők és levelezők tanúsítványtárába.

Általánosságban azonban elmondható, hogy csak nagyon kevesen foglalkoznak azzal, hogy a mindennapi kommunikációjukat, dokumentumaikat, adataikat megfelelően védjék. Pedig látható, hogy milyen jól elgondolt és viszonylag kis fáradtságot igénylő modellek és szolgáltatások állnak napjainkban már a rendelkezésünkre. Észrevétlenül de legtöbb esetben már használjuk is ezeket az alkalmazásokat, csak nem tudjuk, nem értjük mit és mire kellene figyelni.

Igyekeztem ezért bemutatni, hogy a felhasználónak milyen óvintézkedéseket kell tennie ahhoz, hogy egy ilyen infrastruktúrát szabályszerűen és jól tudjon használni. Próbáltam szemléltetni az infrastruktúrában rejlő lehetőségeket és azt, hogy az élet mely területein milyen módon hódít teret magának.

9. Köszönetnyilvánítás

Szeretném megköszönni a témavezetőmnek Prof. Pethő Attilának a Debreceni Egyetem dékánjának, hogy a drága idejét nem sajnálva mindig készséggel a rendelkezésemre állt. Észrevételeivel, iránymutatásaival, irodalom ajánlásaival a megfelelő irányba terelte kusza gondolataimat. Az Ő hathatós közreműködésével lett tartalmas és logikailag teljes a munkám.

Megköszönöm Farkas Norbertnek, Hegedüs Zsoltnak és a többi munkatársamnak (InfoStyle Kft.) a program elkészítésében, felépítésében nyújtott tanácsokat és iránymutatásokat.

Köszönettel:

10. Irodalomjegyzék

01. **Prof. Pethő Attila:**
Kriptográfia1, DE egyetemi jegyzet

02. **Prof. Pethő Attila:**
Kriptográfia2, DE egyetemi jegyzet

03. **Dr. Csirmaz László:**
Kriptográfiai protokollok, DE egyetemi jegyzet

04. **Dr. Lakatos Piroska:**
Algebrai kódelmélet, DE egyetemi jegyzet

05. **Ködmön József:**
Kriptográfia, Az informatikai biztonság alapjai, ComputerBooks Budapest, 2002

06. **Virasztó Tamás:**
Titkosítás és adatrejtés, Biztonságos kommunikáció és algoritmikus adatvédelem NetAcademia Budapest, 2004

07. **Gerencsér András:**
Digitális aláírás szolgáltatás architektúrája, BKÁE egyetemi jegyzet

08. **Nagy Sándor:**
Útlevél a digitális világhoz,
ComputerBooks Budapest, 2005

09. **Dr. Nyékyné Gaizler Judit (szerk.) et al.:**
JAVA 2 útikalauz programozóknak 1.3, ELTE TTK Hallgatói Alapítvány Budapest, 2001

10. **Johannes A. Buchmann:**
Introduction to Cryptography, Chapter 14, Public-Key Infrastructures, Springer 2000

11. **John Linn:**
Trust Models and Management in Public Key Infrastructures, RSA Laboratories

12. **NetAcademia – tudástár:**
Digitális aláírás

13. **Whitfried Diffie and Martin Hellman:**
New direction in cryptography, IEEE Trans. on Information Theory, 22 (1976), 644-654.

14. **R. C. Merkle and Martin Hellman:**
Hiding information and signatures in trapdoor knapsacks, IEEE Trans. on Information Theory, 24 (1978), 525-530.

15. **Adi Shamir:**
A polynomial-time algorithm for breaking the basic Merkel-Hellman cryptosystem, IEEE Trans on Information Theory, 30 (1984) 699-704.

16. **Ronald Rivest, Adi Shamir, Leonard Adleman:**
A method for obtaining digital signature and public-key cryptosystems, Communications of the ACM, 21 (1978), 120-126.

17. **Buttyány Levente:**
Kriptográfia és alkalmazásai, Budapest, Typotex 2004

18. 2001. évi XXXV. törvény az elektronikus aláírásról

19. 2004. évi LV. törvény az elektronikus aláírásról szóló 2001. évi XXXV. törvény módosításáról

20. 1991. évi XLI törvény a közjegyzőkről, egységes szerkezetben végrehajtásról szóló 13/1991 (XI.26.) IM rendelettel, 7A§, 21/A§

21. 2004. évi CXL. törvény a közigazgatási eljárás és szolgáltatás általános szabályairól, 72§, Elektronikus ügyintézéssel kapcsolatos szabályok fejezet

22. RFC-1321: The MD5 Message-Digest Algorithm, http://www.faqs.org/rfcs/rfc1321.html

23. RFC-3174: US Secure Hash Algorithm (SHA1), http://www.faqs.org/rfcs/rfc3174.html

24. RFC-2251: Lightweight Directory Access Protocol v3, http://www.faqs.org/rfcs/rfc2251.html

25. RFC 2559: Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2

26. RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile, http://www.faqs.org/rfcs/rfc2459.html

27. U.S. Federal Information Processing Standard, Digital Signature Standard, FIPS 186 http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf

28. NetLock Kft., http://www.netlock.net

29. http://www.rsasecurity.com

30. http://www.rsasecurity.com/rsalabs/pkcs/index.html

31. http://www.di-mgt.com.au/rsa_alg.html

32. Nemzeti Hírközlési Hatóság, http://www.nhh.hu, http://www.hif.hu