

Távközlési informatika

Kriptográfia

Dr. Beinschróth József

Fogalmak, alapelvek

- A biztonság összetevőinek egy része kriptográfián alapul – de a kriptográfia önmagában nem oldja meg a biztonság problémáját
- Fogalmak
 - Kriptográfia (cryptography)
 - Titkosítás – titkosítás: titkosító eljárások kifejlesztése és alkalmazása
 - Kriptoanalízis (criptoanalysis)
 - A titkosítás megfejtése
 - Kriptológia (criptology)
 - Kriptográfia + kriptoanalízis
 - Kulcs
 - Relatív rövid karaktersorozat, a hosszúsága kritikus
 - Titkos!?
- Alapelvek
 - A titkosítási algoritmusok publikusak!!! A titkosság kizárólag a kulcsokban rejlik.
 - Az üzenetek kell, hogy valamennyi redundanciát tartalmazzanak, de a túl sok redundancia egyszerűsíti a megfejtést.
 - A titkosított üzenetek ismételt elküldésének problémáját a titkosítás nem oldja meg, erre valamilyen külön módszer kell

A titkosítási modell

-klasszikus, szimmetrikus kulcsú-

A titkosításnak
adattárolás esetén
is jelentősége lehet

Aktív támadó:

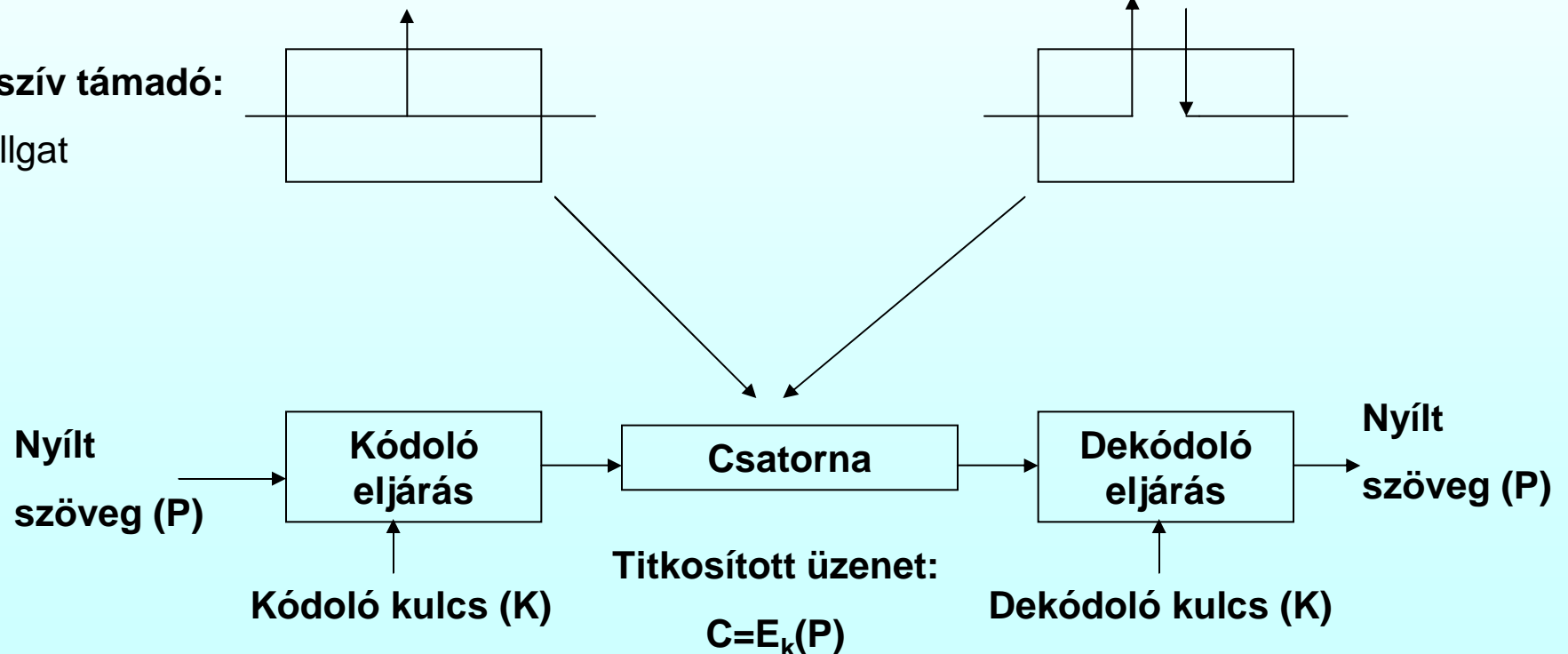
Helyettesít

Küldést kezdeményez

Ismételten elküld

Passzív támadó:

lehallgat



Kódolás: $C=E_k(P)$, dekódolás után visszacapjuk az eredeti nyílt szöveget: $P=D_k(E_k(P))$

Problémakörök

- Kódfejtői problémakörök
 - Titkosított szöveg alapú probléma (cipertext only):
 - Számos titkosított szöveg van, de nincs egyetlen nyílt szöveg sem
 - Ismert nyílt szöveg alapú probléma (known plaintext):
 - Rendelkezésre áll néhány nyílt szöveg és azok titkosított párja
 - Választott szöveg típusú probléma (chosen plaintext):
 - Lehetőség van egy tetszőlegesen választott szöveg titkosított párjának előállítására
- Elvárások
 - Az üzenet tartalmának elrejtése - rejtjelezés
 - A tartalom sértetlensége - integritás
 - Az üzenet letagadhatatlansága - bizonyíték
 - A forrás igazolhatósága – szerzői jogok?

Klasszikus titkosítási módszerek

- Helyettesítő kódolás
- Keverő kódolás
- Egyszer használatos bitminta (one time pad)

Ezek a módszerek tradicionálisak, a mai modern módszerek ezek kombinációit használják, ugyanakkor bonyolult és szövevényes algoritmusokon alapulnak abból a célból, hogy a kódfejtő munkáját megnehezítsék.

Helyettesítő kódolás

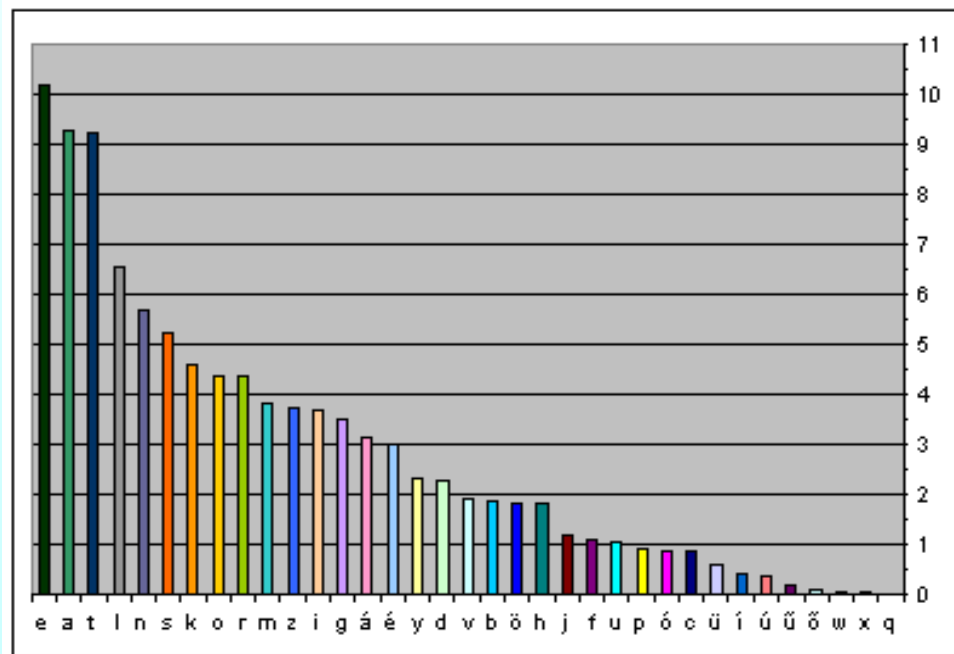
- Minden betű vagy betűcsoport egy másik betűvel vagy betűcsoporttal helyettesítődik
 - Pl. minden betű helyett a 3-mal utána következő betűt írjuk le
 - Például, ha a nyílt szöveg betűi:
 - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 - Akkor a titkos szöveg betűi (5 karakteres eltolás):
 - V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
 - Itt összesen 26 kulcs lehet! Feltörés: próbálgatás
 - Jobb módszer: Minden karakterhez egy másik karaktert rendelünk hozzá egy táblázat alapján
 - Ekkor összesen $26! = 4 \times 10^{26}$ lehetséges kulcs lesz (ennyiféle módon lehet a második sort felírni) Feltörés próbálgatással: 1 millió kulcs/s sebességgel 10^{13} évig tart
- A helyettesítő kódolás általános feltörési módszere: felhasználjuk a természetes nyelvek statisztikai jellemzőit
 - Előfordulási gyakoriság, betűkettősök, stb.
 - Az egyes betűk előfordulási gyakoriságából azonnal kiderül, hogy milyen nyelvű a szöveg

Helyettesítő kódolás

- Betűk előfordulásának relatív gyakorisága magyar, szóköz nélküli szövegben (10000 szavas újságcikk)

A	9,35	I	4,39	R	4,22
Á	3,72	J	1,21	S	6,57
B	1,72	K	5,35	T	7,87
C	0,6	L	6,3	U	1,29
D	1,71	M	3,92	Ü	0,93
E	9,71	N	5,47	V	1,81
É	3,87	O	4,47	W	0
F	0,88	Ö	2,14	X	0,01
G	3,55	P	1,04	Y	2,21
H	1,23	Q	0	Z	4,46

- 11 magyar nyelvű regény és novella, mintegy 4 500 000 karaktere alapján



Kettős betűk?

Jellegzetes betűsorozatok?

Keverő kódolás

- Nem történik helyettesítés, de a karakterek sorrendje megváltozik – a betűk előfordulásának relatív gyakorisága nem változik → nagy valószínűséggel keverő kódolás (ez a feltörést segíti)
- Pl.: oszlop alapú keverő
 - A nyílt üzenetet sorokba rendezve írjuk le és függőlegesen olvassuk ki, ez utóbbi adja a titkosított üzenetet
 - A feltöréshez meg kell találni az oszlopok számát és sorrendjét
 - Feltörés: valószínűen előforduló szavakat, kettős betűket stb. keresünk.

Egyszer használatos bitminta

- A kulcs egy véletlen bitsorozat, legalább olyan hosszú, mint az üzenet
- Képezzük a kódolandó üzenetnek megfelelő bitsorozat és a kulcs XOR kapcsolatát, ez a titkosított üzenet
- Feltörhetetlen, mivel a titkosított üzenet nem hordoz információt!
- Hátrány: rendkívül érzékeny az elveszett vagy közbeékelődött karakterre
- A kulcsot biztonságos csatornán kell továbbítani, de ezzel az erővel akár magát az üzenetet is továbbíthatjuk a biztonságos csatornán!
- A kulcs hossza felső korlát az üzenetek méretére

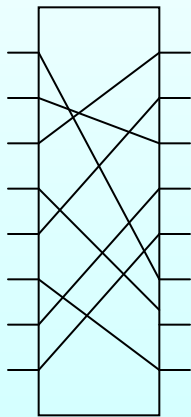
Szimmetrikus kulcsú algoritmusok

- Titkos kulcs

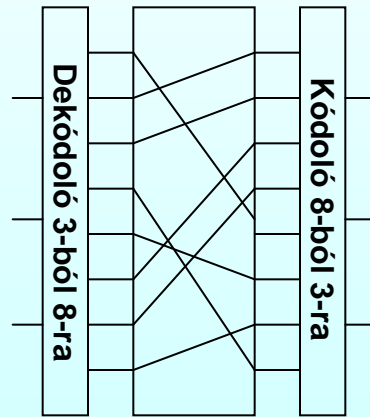
- Ugyanaz a kulcs használatos a kódoláshoz és a visszafejtéshez
- Bonyolult matematika, a kulcs titkos, a kulcs ismeretében mind a kódolás, mind a visszafejtés viszonylag egyszerű, kulcs hiányában a visszafejtés nagyon nehéz. (a kódolási algoritmus publikus: bitek felcserélése és bitminták más bitmintákkal való helyettesítése.)
- Probléma : a kulcs, ill. az abszolút biztonságos csatornán történő továbbítása.
- Példák: LUCIFER, Blowfish, DES, Triple DES, DESX, GDES, RDES, IDEA, RC4, RC5, Rijndael, Twofish, Serpent

Szimmetrikus kulcsú algoritmusok

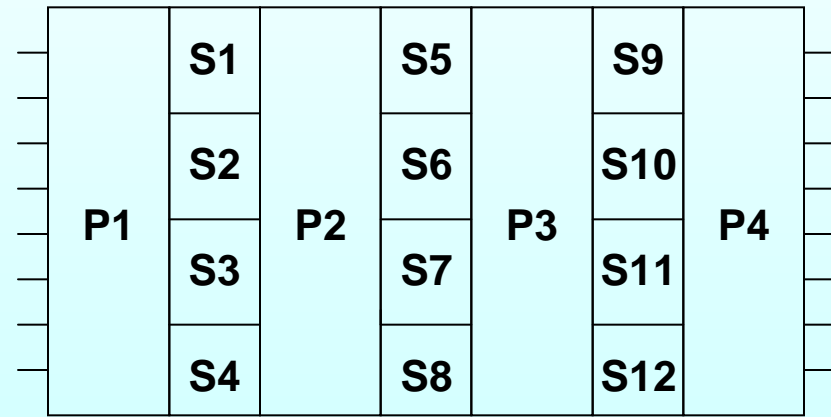
- Példa: blokk kódolók:
 - Egyszerűen hardveresen megvalósíthatók, gyorsak



P doboz
(permutation)



S doboz
(substitution)



Szorozattitkosító

Gyakorlat:

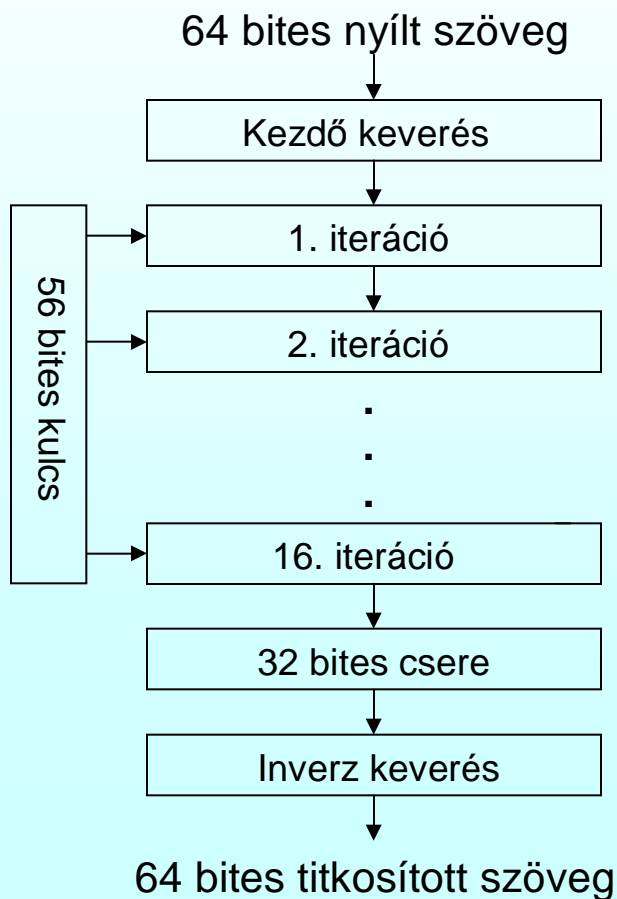
64-256 bemenet

18 fokozat

**A DES és változatai tipikusan
szorozattitkosítók**

Szimmetrikus kulcsú algoritmusok

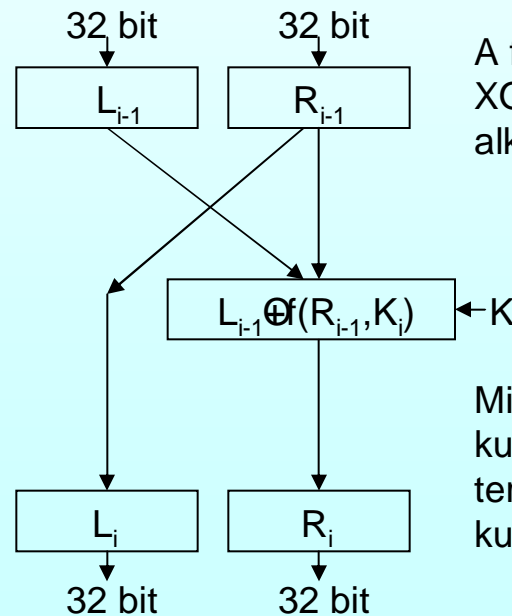
- DES (Data Encryption Standard)



A kezdő és inverz keverés egymásnak inverzei, mindkettő kulcsfüggetlen.

A 32 bites csere során a baloldali 32 bites blokk felcserélődik a jobboldalival.

Iteráció:



A függvény többlépéses keverés, XOR kapcsolat, S doboz, P doboz alkalmazásával valósul meg

Minden iterációs lépésben különböző kulcsot használunk, amelyek természetesen az eredeti kulcsból származnak.

Szimmetrikus kulcsú algoritmusok

- Gyenge pont: a kulcs ill. a kulcs eljuttatása a partnerhez
 - Szükséges egy abszolút biztonságos csatorna (diplomáciai futár, katonaság)
 - Shannon-féle alapséma: a rejtjelezés által védett kommunikációs csatornát ki kell egészíteni egy „abszolút biztonságos csatornával, amelyen a kulcstovábbítás történik
 - Abszolút biztonságos csatorna: nem a technológia, hanem szabályok, eljárási utasítások betartása alapján abszolút biztonságos – az emberi tényező megjelenik: tévesztések, fegyelem betartása stb.
 - Történelem: az USA NSA (Nemzetbiztonsági hivatal) központilag készítette és osztotta ki a kormányhivatalok számára szükséges kulcsokat - kémkedési botrány lett belőle: Roland W. Pelton NSA ügynök kiszolgáltatta a kulcsokat

Szimmetrikus kulcsú algoritmusok

Elterjedt, erős szimmetrikus kulcsú titkosítási algoritmusok:

Kód	Szerző	Kulcshossz
DES (Data Encryption Standard)*	IBM	56 bit
Rijndael	Daemen és Rijmen	128-256 bit
Serpent	Anderson, Biham, Knudsen	128-256 bit
Triple DES	IBM	168 bit
Twofish	Bruce Schneider	128-256 bit

***A DES ma már klasszikus, nem túl erős változat**

Aszimmetrikus kriptográfia

- Privát és publikus kulcsok
 - A szimmetrikus kriptográfiában alkalmazottól különböző eljárás
 - Az eljárásban a **kódoláshoz (E)** és a **dekódoláshoz (D)** tartozó kulcsok különbözőek – $D(E(P))=P$ (A kódoláshoz és a dekódoláshoz különböző kulcsot használunk.)
 - A D-ben és E-ben alkalmazott kulcsok között matematikai összefüggés van, de D kulcsának előállítása E kulcsából rendkívül nehéz.
 - Nagyon nagy számok prímtényezős felbontása nagyon nehéz feladat - például E kulcsa egy nagyon nagy szám (>100 számjegy), D kulcsa ennek prímtényezői egymás után írva
 - Másik példa nehéz feladatra: $a^x=b \bmod n$, $x=?$ a hatványozás inverze modulo n (pl. $2^x=7 \bmod 13$ megoldás: $x=11$)
 - D kulcsának ill. P-nek előállítása E kulcsából ill. $E(P)$ -ből nem lehetséges, azaz a választott nyílt szöveg típusú támadással szemben az eljárás ellenálló. ($E(P)$ halad a nyílt csatornán!. Bárki hozzáférhet.)
 - Ennek megfelelően E kulcsát nem kell titokban tartani! (publikus kulcs)
 - D kulcsa azonban titkos! (privát kulcs)

Aszimmetrikus kriptográfia

- Nyilvános (publikus) kulcsok

- A kulcsok továbbításához nincs szükség abszolút biztonságos csatornára – a publikus kulcs akár a tulajdonos honlapján szerepelhet! Többszereplős titkosított üzenetküldés lehetséges.
- Minden résztvevő két kulccsal rendelkezik, egy publikus (nyilvános) és egy privát (titkos) kulccsal, a két kulcs között bonyolult matematikai összefüggés van.
- Minden résztvevő ismeri minden résztvevő publikus kulcsát.
- A privát kulcsát minden résztvevő titokban tartja és senkinek nem küldi el.
- Egy egy résztvevő publikus és privát kulcsa között összefüggés van: a publikus kulccsal kódolt üzenet a privát kulccsal fejthető vissza. A publikus kulcs ismeretében sem a privát kulcs, sem a kódolt üzenetből a kódolatlan nem állítható vissza gyakorlatilag.

Aszimmetrikus kriptográfia

- Példák: RSA, RC1-9, PGP
- Hosszabb kulccsal nő a biztonság (>1024 bit)
- Hosszú üzenetek továbbítása problematikus (lassú algoritmusok). - A kódolt dokumentum mérete legalább kétszeresére nő!!
- A szimmetrikus kulcsok továbbításához viszont praktikusán használható eljárás.

Aszimmetrikus kriptográfia

- **RSA (Rivest, Samir, Adleman)**

- Premisszák

- Válasszunk két nagy (1024bit) prímszámot (p, q)!
- Számítsuk ki $n = p \times q$ és $z = (p-1) \times (q-1)$ értékeket!
- Keressünk egy z -hez relatív prímet (d)!
- Keressünk egy olyan e -t, amelyre $e \times d = 1 \pmod{z}$

- Kódolás

- A nyílt szöveget k bit hosszúságú szegmensekre (P) osztjuk, úgy, hogy $2^k < n$, azaz $0 \leq P < n$
- Minden P -re meghatározzuk a $C = P^e$ -t, ez a titkosított kód

- Dekódolás

- Minden C -re meghatározzuk a $P = C^d = (P^e)^d$ ez a nyílt szöveg, fennáll ugyanis, hogy, ha $0 \leq P < n$, akkor a kódoló és dekódoló függvények egymás inverzei

- Kulcsok

- Publikus: e, n
- Privát: d, n

d e -ből és n -ből gyakorlatilag

nem számítható!!!

Digitális aláírás

2001. óta létezik törvény
az elektronikus
aláírásról

- Elvárt feltételek

- (Nem a kézírás elektronikus vizsgálatáról van szó!)
- Igazolja, hogy az üzenet a feladótól és nem mástól származik, ill., hogy az üzenetet nem változtatták meg illetéktelenül
- Az üzenetküldő később ne tagadhassa le az üzenetet ill. a tartalmát (nonrepudation)
- A címzett önmaga ne állíthassa elő vagy ne változtathassa meg az üzenetet

- Alapelv

- Ha az aláírás műveletét kizárólag egy kulcs alkalmazásával lehet elvégezni és a kulcs az aláíró kizárólagos birtokában van (titkos), akkor az aláírás műveletét csak az aláíró végezhetette el

- Kétféle megközelítés

- Szimmetrikus kulcsú aláírások
- Nyilvános kulcsú aláírások

Digitális aláírás

- Szimmetrikus kulcsú aláírások
 - Létezik egy központi hitelességvizsgáló/tanúsító szerv, a résztvevők ebben megbíznak
 - A résztvevők csak a központi szerv közvetítésével kommunikálnak a szimmetrikus kriptográfia segítségével
 - Üzenet a résztvevőtől a központi szerv felé a feladó titkos kódjával titkosítva:
 - A címzett azonosítója
 - Véletlen szám, minden üzenetben más (a gyors újraküldés ellen)
 - Time stamp (az újraküldés ellen)
 - Üzenet
 - Üzenet a központi szervtől a címzett felé a címzett titkos kódjával titkosítva:
 - Az eredeti üzenet
 - A feladó azonosítója
 - Time stamp
 - A központi szervezet iránti bizalom kritikus probléma!

Digitális aláírás

$$D_i(E_i(P))=P$$

E_A : A publikus kulcsa
 E_B : B publikus kulcsa
 D_A : A titkos kulcsa
 D_B : B titkos kulcsa

- Nyilvános kulcsú aláírások

- Kiindulás: a titkosítási algoritmus a $D(E(P))=P$ tulajdonság mellett rendelkezzen az $E(D(P))=P$ -vel is! (Az RSA pl. ilyen.)
- Mielőtt a feladó elküldi az üzenetet, saját titkos kulcsával titkosítja. A címzett ezt a lépést majd a küldő publikus kulcsával „semlegesíti”.
- A küld levelet B-nek: $E_B(D_A(P))$ kerül továbbításra, előállítás:
 - 1. **Saját titkos kulcsával** kódol
 - 2. A címzett, **B publikus kulcsával** kódol
- B kap levelet A-tól: Megkapja $E_B(D_A(P))$ -t. Ebből előállítja $D_B(E_B(D_A(P)))=D_A(P)$ -t és $E_A(D_A(P))=P$ -t
 - 1. **Saját titkos kulcsa** segítségével előállítja $D_A(P)$ -t
 - 2. $D_A(P)$ -t eltárolja, ezzel tudja bizonyítani, hogy a hozzá érkező üzenet D_A -val lett titkosítva, azaz A titkosította (írta alá) azt, ha E_A -val ebből előállítható P, akkor ez bizonyított
 - 3. **A publikus kulcsa** segítségével előállítja P-t
- De facto szabvány az RSA, a legtöbb termék ezen alapul, további ismert algoritmus a DSS (Digital Signature Standard)

Mindkettő rendelkezésre áll!

Mindkettő rendelkezésre áll!

Gyakran nem szükséges az
üzenet tartalmát titkosítani,
de hitelesítés szükséges

Digitális aláírás

- Üzenet pecsétek

- Fő összetevő: hash – egyirányú függvény
- Hash (kivonat) digitális ujjlenyomat, egy bitsorozat, amelyet ismert algoritmussal az üzenetből készítünk (hash algoritmus)
 - A hash hossza mindig azonos, azaz nem függ a dokumentum méretétől – belőle az eredeti dokumentum nem állítható elő (egyirányú)
 - Két dokumentumhoz tartozó hash soha nem lehet azonos (ütközésmentes)
 - Az eredeti szövegben egyetlen bit megváltoztatása a hash-ben jelentős változást okoz (lavina hatás)
- A hash-t aláírjuk és továbbítjuk (hogyan magát az üzenet is rejtjelezzük-e, az más kérdés – erőforrás probléma)
- A címzett megkapja a titkosított, aláírt hash-t ugyanakkor maga is elő tudja azt állítani.
- A feladó és a címzett ugyanazt a hash algoritmust használja
- A címzett ellenőrzi, hogy a kapott és a helyben előállított hash megegyezik-e

Digitális aláírás

- Eljárás aláíráskor:
 1. Kivonat készítése
 2. A kivonat aláírása
 3. Továbbítás
- Eljárás ellenőrzéskor:
 1. Kivonat előállítás
 2. A kivonat dekódolása
 3. A kapott és a dekódolt kivonat összehasonlítása
- Ismert eljárások
 - MD5 (Message Digest)
 - SHA-1 (Secure Hash Algorithm)
 - ...

CA (Certification Authority)

- **Probléma**

- Az aszimmetrikus kriptográfia publikus kulcsainak meghamisítása (elfogott kérésre hamis kulcs elküldés, lecserélt weblap stb.)

- **Megoldás**

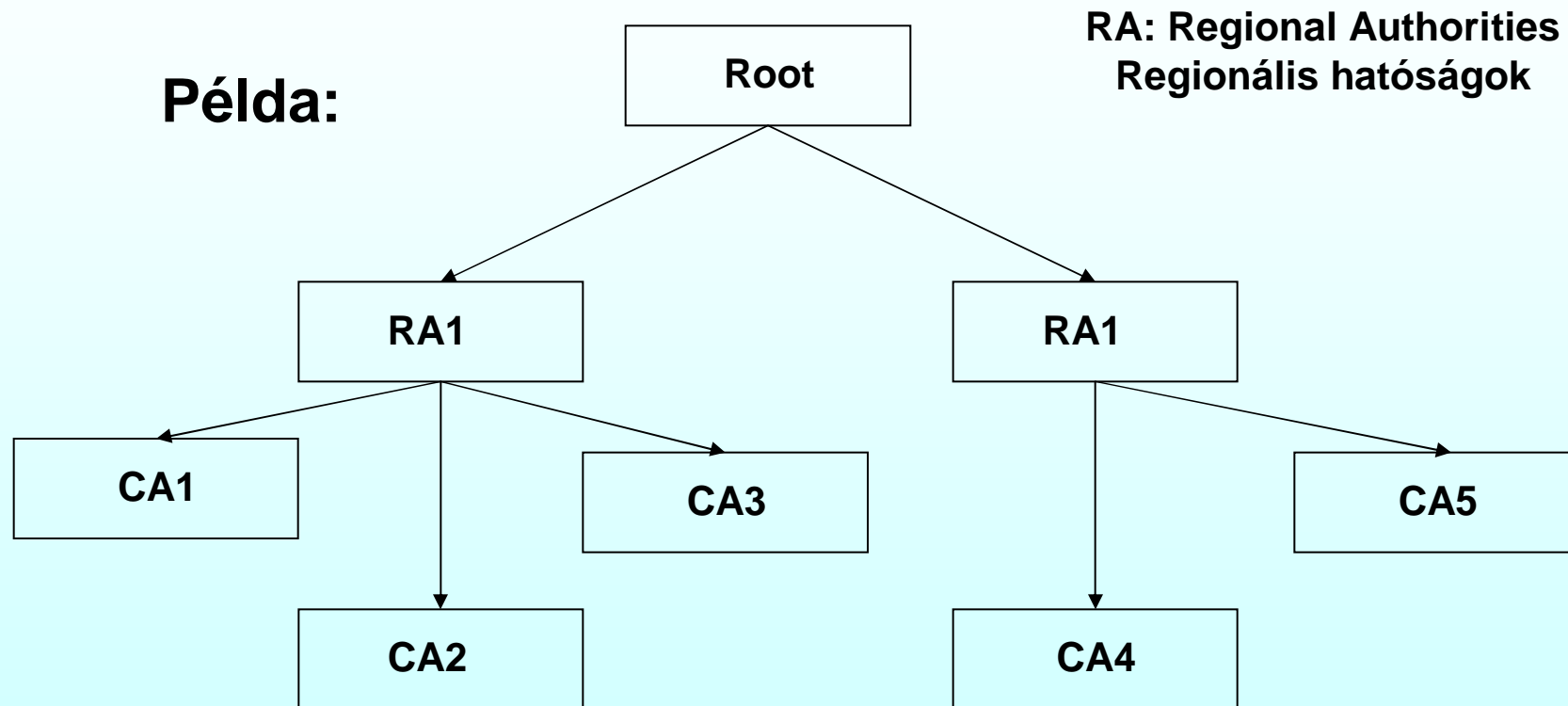
- A nyilvános kulcsok kezelése
- CA - tanúsító hatóság - tanúsításokat ad ki
 - Regisztráció
 - Pecséttel ellátott kulcsok kiosztása ill. hozzárendelése a felhasználó/cég nevéhez (A pecsét létrehozásában nemcsak a kulcs, hanem az adatok (név stb.) is szerepelnek!
 - A CA publikus kulcsát használva kapom meg a tanúsított cég nevét (azonosítóját) és a tanúsított cég publikus kulcsát
- Tanúsításokra vonatkozó ITU szabvány
 - X509

PKI (Public Key Infrastructure)

- **A gyakorlatban használt hitelesítő, tanúsító rendszer**
 - Egyetlen CA nem képes ellátni a feladatot
 - terhelés, bizalom, rendelkezésre állás stb.
 - PKI: Sok összetevő van, a PKI valamennyi összetevőt tartalmazza
 - Felhasználók, CA-k, tanúsítványok, tanúsítvány könyvtárak stb.
 - A PKI szervezetbe rendeli az összetevőket
 - Szabványokat tartalmaz a különféle dokumentumok és protokollok számára

PKI

Példa:



A root az RA-kat, az RA-k a CA-kat hitelesíti, a CA-k kiadják a tényleges tanúsítványokat
Több root is létezik (>100)

A root publikus kulcsát mindenki ismeri és elfogadja (pl. bele van építve a browserbe).
A kommunikációs partnerek megküldik egymásnak a az összes tanúsítványt amelyek alapján a rootig megtörténhet a hitelesítés (bizalmi lánc - chain of trust)

Dr. Beinschróth József

SSH – Secure Shell

- A „telnet biztonságos változata”, de használható ftp, rsh, rlogin, rcp stb. helyett is: bejelentkezhetünk egy távoli gépre és ott parancsokat adhatunk ki
 - Bejelentkezési lehetőség távoli gépre
 - A felhasználó a távoli gép eléréséhez teljes környezetet kap, a távoli gép szolgáltatásai elérhetővé válnak
 - A távoli gép parancsai a távoli gépen végrehajthatók
 - Fájl transzfer lehetőséget is biztosít
- Az alkalmazott hitelesítés az RSA-ra épül
- Az elterjedt op. rendszerekhez létezik SSH kliens és SSH szerver - Léteznek kereskedelmi és free változatok is:
 - Free: www.ssh.fi
- SSH szerver installálás
 - Az SSH szerver installálásakor automatikusan legyárt egy publikus és egy privát kulcsot

SSH – Secure Shell

- Kliens-szerver kapcsolat összetevői
 - Gép szintű hitelesítés
 - A kliens elkéri és megkapja a publikus kulcsot (megbízható forrásból vagy a szervertől)
 - A kliens egy véletlen üzenetet generál, titkosítja azt a publikus kulccsal, megtartja és elküldi a szervernek
 - A szerver a kapott üzenetet titkosítja a saját titkos kulcsával (előáll az eredeti üzenet), ezt visszaküldi a kliensnek
 - A kliens megállapítja, hogy valóban a megfelelő szerverrel lépett kapcsolatba
 - A folyamat szerepcsere után fordítva is lejátszódik, így a szerver is megállapítja, hogy a megfelelő klienssel lépett kapcsolatba
 - Felhasználói szintű hitelesítés
 - A felhasználó az azonosítóját és jelszavát titkosítva küldi meg a szervernek, így hitelesíti magát
 - Adatforgalom
 - Szimmetrikus kriptográfia: DES, DE3, IDEA stb. alapján

SSL – Secure Sockets Layer

- A web interaktívvá válásakor felmerült az igény a biztonságos kommunikáció megvalósítására
- Megoldás: az alkalmazási és szállítási réteg közé új réteg kerül: SSL (Secure Sockets Layer) – fogadja a browserből érkező kéréseket és átadja őket a TCP rétegnek
- Az SSL funkciói
 - Paraméterek egyeztetése
 - Hitelesítés
 - Titkosított átvitel
 - Adatintegritás (sértetlenség) biztosítása
- SSL fölött használt HTTP: HTTPS (Secure HTTP)
- Az SSL nemcsak a web alkalmazást képes támogatni!
- Az SSL eredetileg a Netscape tetméke volt, a belőle készült szabvány a TLS (Transport Layer Security)

SSL – Secure Sockets Layer

- Az SSL kapcsolat
 - Az összeköttetés kiépítése (A és B között)
 - A összeköttetés kiépítést kezdeményez (egyúttal elküldi a támogatott SSL verziószámot, a támogatott titkosítási algoritmusokat, ill. tömörítési eljárásokat és egy véletlen számot - R_A)
 - B kiválaszt egy titkosítási algoritmust és egy tömörítési eljárást, a válaszban elküldi ezeket A-nak, elküld ezen kívül egy véletlen számot is – R_B)
 - B elküldi továbbá saját tanúsított publikus kulcsát
 - A elküld B-nek egy 384 bites véletlen számot (előzetes főkulcs) B publikus kulcsával titkosítva (az adatok titkosítására használt tényleges viszonykulcs a R_A , R_B és az előzetes főkulcs felhasználásával áll elő – a tényleges viszonykulcsot ezután mindkét fél ki tudja számítani)
 - Ezután már csak a tényleges viszonykulcsot használják
 - (Ezután A titkosítottan küldött azonosítóval és jelszóval azonosítja magát – de ez nem része az SSL protokollnak)
 - Az adatforgalom biztosítása

Kerberos

- Viszonylag régi, de elterjedt hitelesítő protokoll
- Szimmetrikus kriptográfiára épül: DES alapú titkosítást használ
- Számos ingyenes és kereskedelmi termékben került felhasználásra (pl. MS Windows)
- Kerberos V5 verzió rfc 1510 (de facto szabvány)
- Az igényelt szolgáltatást igénybe vevő és nyújtó gépeken kívül szerepel még további két szerver kifejezetten a hitelesítést támogató funkciókkal
 - AS (Authentication server – hitelességvizsgáló szerver)
 - Adatbázisában minden a rendszerhez tartozó felhasználó nevének ill. a hozzájuk tartozó titkos kulcsokon kívül tárolja a TGS szerver titkos kulcsát is
 - TGS (Ticket-Granting Server) jegykiadó szerver
 - Adatbázisában minden a rendszerhez tartozó szerver nevét és a hozzájuk tartozó titkos kulcsokat tárolja

Kerberos

- A hitelesítés folyamata (A felhasználó el szeretné érni B szerver szolgáltatását)
 - A nyílt szöveggént elküldi nevét az AS-hez (A)
 - AS A titkos kulcsával kódolva (K_A) visszaküld A-hoz egy viszonykulcsot (K_S), ami később majd a B szerverrel folytatott kommunikációban lesz használatos és egy „jegyet”, ami a TGS szerver titkos kulcsával (K_{TGS}) kódolva tartalmazza A nevét és a viszony kulcsot (K_S) – $K_A(K_S, K_{TGS}(A, K_S))$, ebből A értelmezni tudja K_S -t ($K_{TGS}(A, K_S)$ -t nem tudja értelmezni, de tovább tudja küldeni)
 - A elküldi a TGS-hez az AS-tól kapott $K_{TGS}(A, K_S)$ -t, B szerver nevét és egy időbélyeget
 - A TGS szerver ebből vissza tudja kódolni A nevét és a viszonykulcsot: biztos lehet benne, hogy az üzenetet tényleg A küldte és az A-val folytatott további kommunikáció során használhatja a K_S viszonykulcsot

Kerberos

- TGS szerver K_s -sel titkosítva visszaküldi A-nak a B szerver nevét és egy másik viszonykulcsot (K_{AB}), amit majd A a B szerverrel folytatott kommunikáció során használhat, visszaküldi továbbá B titkos kulcsával titkosítottan A nevét és a viszonykulcsot (K_{AB}) - $K_s(B, K_{AB})$, $K_B(A, K_{AB})$
- A ebből elő tudja állítani K_{AB} -t, $K_B(A, K_{AB})$ -t nem tudja értelmezni, de tovább tudja küldeni)
- A elküldi a B-hez a TGS-től kapott $K_B(A, K_{AB})$ -t, B szerver nevét és egy időbélyeget
- B szerver ebből vissza tudja kódolni A nevét és a viszonykulcsot: biztos lehet benne, hogy az üzenetet tényleg A küldte és az A-val folytatott további kommunikáció során használhatja a K_{AB} viszonykulcsot
- (Az időbélyeg továbbítása azért lényeges, mert a jegyek az illetéktelen lehallgatások elleni védelem miatt viszonylag rövid élettartalmúak – feltételezzük, hogy a rendszerórák együtt járnak)

DNSsec – DNS security

- Nyilvános kulcsú kriptográfiai megoldás (rfc2535) nem teljeskörűen elterjedt
- Probléma: a DNS könnyen támadható közbeékelődéses támadással
 - Klasszikus közbeékelődéses támadás: A üzenetet küld B-nek, T elfogja az üzenetet és B nevében válaszol
 - Modernebb változat: a DNS szerver cache-be hamis adatokat íratunk be (a szerver 1 napig cache-el!) – DNS spoofing
 - Például
 - A típusú rekord helyesen:
 - www.bmf.hu valódi_ip_cím
 - Meghamisított változat
 - www.bmf.hu támadó_ip_címe

DNSsec – DNS security

- Hogyan kerül be a hamis bejegyzés a DNS-be? (poisoned cache)
 - Amikor egy DNS szerver kérdésre válaszol egy másik DNS szerver, akkor a válaszra vonatkozóan semmiféle hitelesítés nem történik
 - A DNS szerver az elsőként beérkező választ fogadja el, az ezután érkező(ke)t idejétmúlt kérésre vonatkozó ill. kéretlen válasznak tekinti és eldobja
- Megoldás
 - A DNS szerverek minden elküldött információt aláírnak a privát kulcsukkal
 - Így az adatok származási helye és hitelessége bizonyított
 - (Az üzenetek nem titkosítottak: a DNS beli információ publikus)
- A DNSsec bevezet új rekord típusokat, pl. KEY típusú rekord (publikus kulcs, felhasznált kriptográfiai algoritmus stb.)

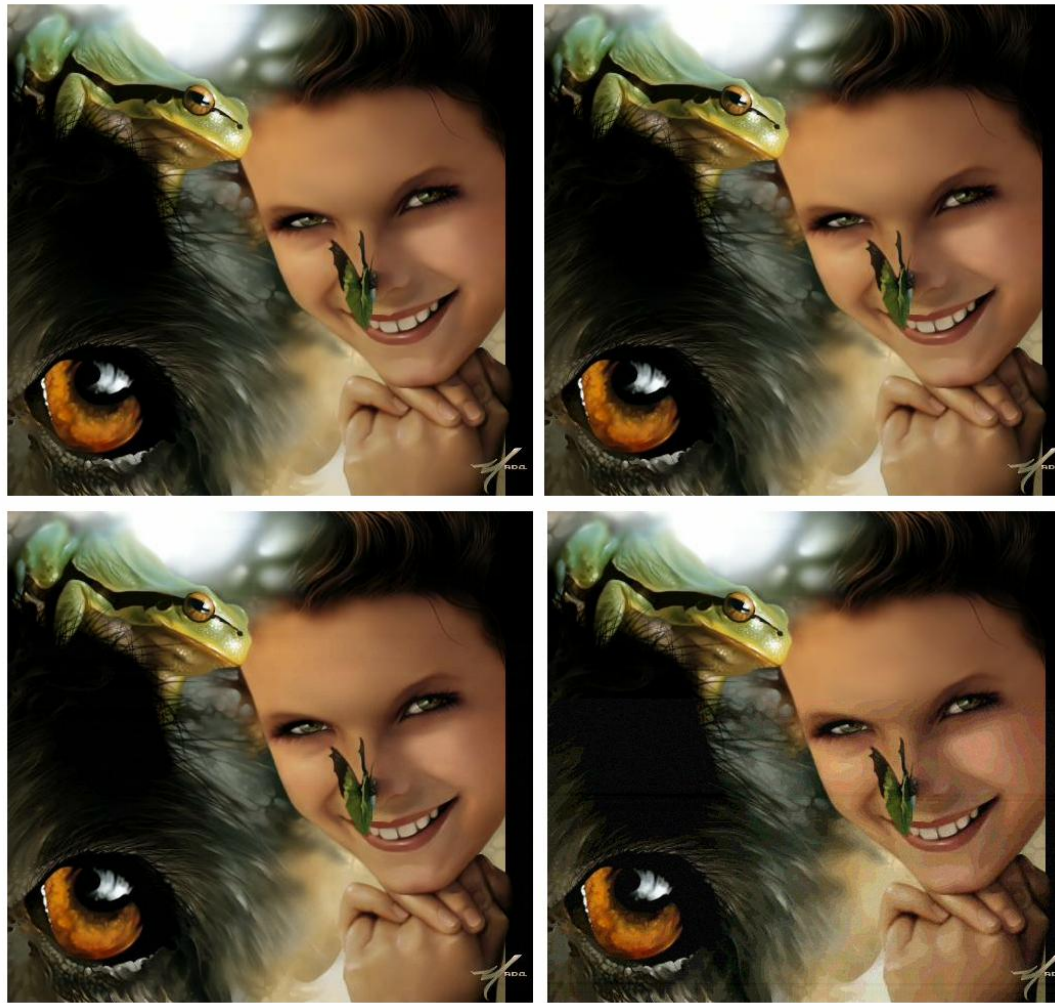
Szteganográfia - üzenetelrejtés

- A kommunikáció is rejtve marad
 - Adatok elrejtése fájlokban (tipikusan hang vagy kép fájlban)
 - Segítségével az elrejtendő fájlt elrejthetünk egy képben vagy egy akár MP3 formátumú zeneszámban úgy, hogy a képen vagy a zenén érzékszerveink segítségével semmilyen változást nem érzékelünk, az elrejtéshez használt kulcs ismerete nélkül pedig sehogyan sem mutatható ki, hogy az adott kép- vagy zenefájl önmagán kívül mást is tartalmaz
 - Tipikusan az egyes képpontok színét meghatározó bájtok legkisebb súlyozású bitjeit használják fel, kihasználva az emberi érzékszervek korlátait
 - Az elrejtett információt előzetesen kriptográfiai eljárással kódolják, így az elrejtés ténye észrevehetetlenné válik
 - Egy fájl általában csak egy nála lényegesen nagyobb másik fájlban rejthető el nyomtalanul
 - A kulcs ismeretlénben azonban az eredeti fájl visszanyerhető
 - Előnye az egyszerű titkosítással szemben az, hogy a támadó nem is tudja, hogy lenne mit feltörnie

Szteganográfia - üzenetelrejtés

- Más felhasználás: elektronikus vízjel
 - Az elektronikus úton terjesztett publikációk védelme, az elektronikus vízjel- vagy ujlenyomat (szerzői jogvédelem)
 - Egy képbe egyedi azonosítót lehet tenni, a képen ez nem látszik, de a kulcs segítségével kinyerhető belőle
 - Ehhez hasonlóan minden audio CD sávba is copyright információt lehetne rejteni
 - Az azonosító minden másolatban benne lesz, onnan gyakorlatilag eltávolíthatatlan
 - Ha copyright jelzésként nem szöveget vagy számot helyezünk el, hanem képet, emblémát vagy hangot (tehát olyan üzeneteket, melyek egyébként is redundánsak így „sok mindent kibírnak”), valószínű, hogy még különböző szűrések, „belerajzolások” után is megmarad a készítő eredeti(hez hasonló) azonosítója
 - Ha a hamisító saját „védjegyet” tesz a „művére”, az nem fogja az eredetit felülírni, hanem mindkettő megmarad (főként ha különböző módon, különböző helyekre teszik azonosítójukat).

Szteganográfia - üzenetelrejtés



Forrás: Dr. Tóth Mihály

Dr. Beinschróth József