



Data Security: Protocols

Digital Signature (Tk.7.fejezet)

A digitális aláírás protokollok feladatai:

1. az aláírás generálása (az X üzenetet küldő A fél végzi):

$$A \rightarrow B: X, D_A(X)$$

2. az aláírás ellenőrzése (B címzett által)

$$B: X = E_A(D_A(X)) ?$$

3. hitelességgel kapcsolatos vitás kérdések harmadik személy előtti (pl. bíróság által) tisztázása:

- pl. aláíró fél próbál tagadni (pl. kedvezőtlenülé vált időközben a tartalom)
- a címzett fél módosította az aláírt üzenetet



Data Security: Protocols

Digital Signature

A digitális aláírás és a valódi kézjegy közös tulajdonságai:

1. Az aláírás *hiteles*: amikor B ellenőrzi az aláírást A publikus kulcsával, meggyőződik, hogy azt csak A küldhette
2. Az aláírás *nem hamisítható*: csak A ismeri a saját titkos kulcsát
3. Az aláírás *nem újrahasználatos* (nem átemelhető másik dokumentumhoz): az aláírás a dokumentum függvénye is
4. Az *aláírt dokumentum nem módosítható*: ha módosítják a dokumentumot, az eredeti aláírás nem illeszkedik, s ez detektálható
5. Az aláírás *letagadhatatlan*: B vagy egy harmadik fél A közreműködése nélkül ellenőrizni képes az aláírást



Data Security: Protocols

Digital Signature

1 blokk méretű üzenetek:

Formátumozott rövid üzenet: $D_A(X)$

Nem formátumozott rövid üzenet: $X, D_A(X)$

> 1 blokk méretű üzenetek:

$[X, D_A(H(X))]$,

H: **kriptográfiai hash függvény**

- egyirányúság
- ütközésmentesség

Egyirányú (OWHF): egy y hash érték (lenyomat) ismeretében nehéz feladat olyan X' üzenetet (ősképet) kiszámítani, amelyre $h=H(X')$

Ütközésmentes (CRHF): nehéz feladat olyan X, X' ($X \neq X'$) üzenetpárt megadni, amelyeknek azonos a lenyomata, azaz amelyekre $H(X)=H(X') \rightarrow D_A(H(X))= D_A(H(X'))$

Data Security: Protocols

Birthday paradox 1 (Tk.4.2.fejezet)

$$p_r = \left(1 - \frac{1}{m}\right)\left(1 - \frac{2}{m}\right) \dots \left(1 - \frac{r-1}{m}\right) = \prod_{i=1}^{r-1} \left(1 - \frac{i}{m}\right)$$

$$1 - x \approx e^{-x} \quad \left(e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} \dots \right)$$

$$\prod_{i=1}^{r-1} \left(1 - \frac{i}{m}\right) \approx \prod_{i=1}^{r-1} e^{-\frac{i}{m}} = e^{-\frac{r(r-1)}{2m}}$$

$$1 - p_r \approx 1 - \exp(-r^2 / (2m))$$

Pl.

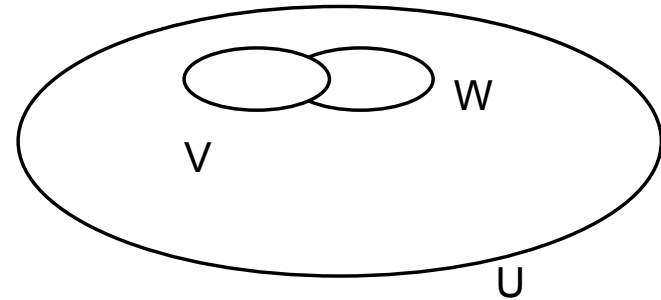
$$m = 365$$

$$r = m^{1/2} \approx 19, \quad p_r \approx 1 - \exp(-0.5) \approx 0.4.$$

Data Security: Protocols

Birthday paradox 2

$$P(V \cap W \neq \emptyset) = 1 - \frac{\binom{m}{2r} (2r)!}{\left(\binom{m}{r} r!\right)^2} \approx 1 - \exp(-3r^2 / m)$$



Pl.

$$r = m^{1/2}, p_r \approx 1 - \exp(-3) \approx 0.95$$



Data Security: Protocols

Digital Signature

Születésnapi paradoxon alapú támadás:

n bites hash lenyomat. Támadó előállít $2^{n/2}$ ártatlan és ugyanennyi csalásra felhasználható dokumentumot.

Születésnapi paradoxon: nagy a valószínűsége annak, hogy a két halmazban lesz legalább egy pár azonos hash értékkel

Támadó a pár ártatlan tagjára kér aláírást, s ezzel már lesz aláírása a csaló párjára is.

Védekezés: aláírás előtt az aláíró eszköz az aláírandó dokumentumhoz illeszt egy véletlenül sorsolt bináris sorozatot. A támadó predikálni nem képes a véletlenített dokumentumot, így a fenti támadást sikerrel nem tudja elvégezni.

$[(X,R), DA(H(X,R))]$



Data Security: Protocols

Digital Signature

"A" szeretne "B"-nek elküldeni egy blokk méretű formátumozott X dokumentumot, úgy hogy az aláírva és rejtjelezve. Két megoldáson gondolkozik:

a.) $A \rightarrow B: E_B(D_A(X))$

b.) $A \rightarrow B: D_A(E_B(X))$

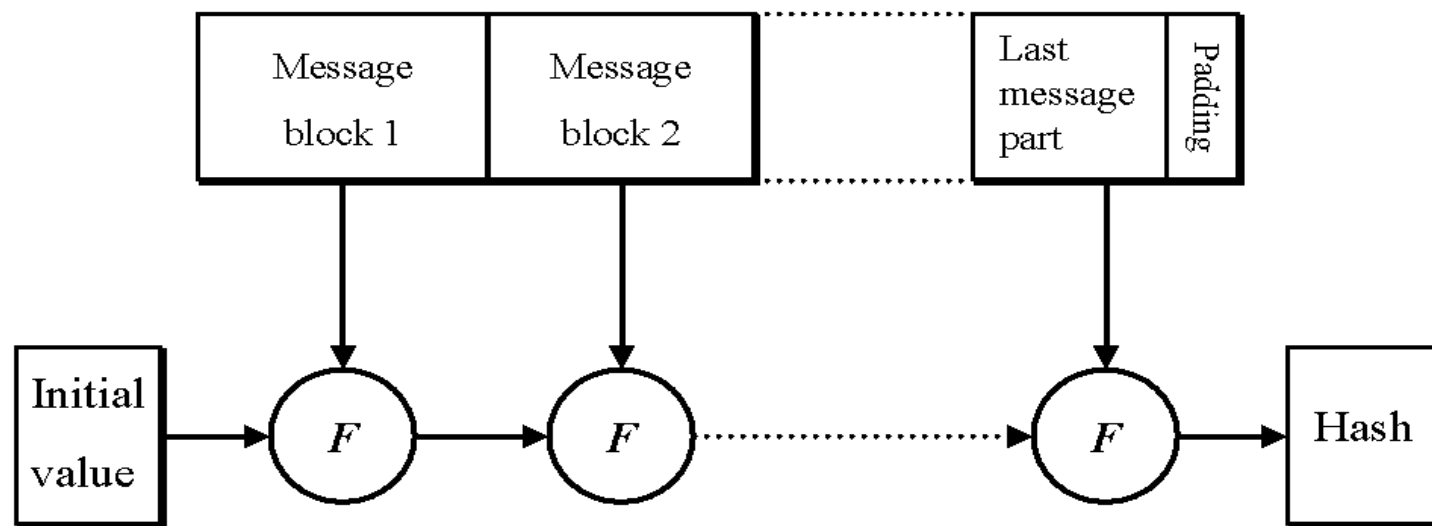
ahol publikus kulcsú technológiát alkalmazunk. Melyik megoldást válassza?

b.) esetén egy támadó lecserélheti A aláírását a sajátjára, így a saját nevében továbbíthatja egy általa nem látott, de sejtett tartalmú X üzenetet.

Data Security: Protocols

Hash functions (Tk.4.1.fejezet)

Az iterált kriptográfiai hash függvény



$X=[X_1, X_2, \dots, X_r]$, azaz X_i üzenetblokkok r hosszúságú sorozata.

$F: \{0,1\}^{2n} \rightarrow \{0,1\}^n$ iterációs (kompressziós) függvény

$H_i = F(H_{i-1}, X_i)$, $i=1,2,\dots,r$, $H_0 = \text{inic}$.

$h=H_r$: lenyomat, hash érték

H_0 : publikus inicializáló érték



Data Security: Protocols

Hash functions

Tulajdonságai:

egyirányú hash függvény (1.őskép-ellenálló, OWHF)

adott h ----> X nehéz feladat

2.őskép-ellenálló:

adott h ($=\text{Hash}(\cdot, X)$) ----> $X' \neq X$, $h = \text{Hash}(\cdot, X')$ nehéz feladat

ütközés-ellenálló (CRHF):

$X' \neq X$, $\text{Hash}(\cdot, X') = \text{Hash}(\cdot, X)$ nehéz feladat

pszeudo- vagy free-start:

a támadó, pl. egy ütközés előállításához, szabadon választhatja meg az iteráció H_0 kezdőértéket.



Data Security: Protocols

Hash functions

Damgard-Merkle padding (ütközésmentesség kiterjesztése)

Tegyük fel hogy van egy F kompressziós függvényünk ütközés-ellenálló tulajdonsággal. Ha az $X=[X_1, X_2, \dots, X_{r-1}]$ üzenetet kiegészítjük egy X_r blokkal, amely tartalmazza az X üzenet bithosszát (nulla bitekkel kiegészítve egész blokkhosszra), az F kompressziós függvényre épülő iterációs hash függvény CRHF lesz.

Bizonyítás: Indirekt:

1. eset:

$X=[X_1, X_2, \dots, X_r]$, $X'=[X'_1, X'_2, \dots, X'_r]$ párra azonos output:
iteráció outputtól visszafelé görgetése a visszafelé legelső eltérő üzenetblokkig,
ellentmondásra jutunk az iterációs fv. feltételezett ütközés ellenálló tulajdonságával.

2.eset:

$X=[X_1, X_2, \dots, X_{r_1}]$, $X'=[X'_1, X'_2, \dots, X'_{r_2}]$, $r_1 \neq r_2$ párra azonos output:
az utolsó blokkok különböznek, s már itt ellentmondásra jutunk az iterációs fv. feltételezett ütközés ellenálló tulajdonságával.



Data Security: Protocols

Hash functions (Tk.4.2-8. feladatok)

Egy H iterációs hash függvényt támadunk.

a.) Nehéz feladat-e $H(m, H_0) = H(m^*, H_0^*)$, $m \neq m^*$ pseudo ütközést előállítani?

b.) Mi a válasz, ha DM paddinget is alkalmaz a támadott tömörítő eljárás?

a.) Nem nehéz.

$H(H_0, [m_1, m_2]) = f(m_2, f(H_0, m_1)) = H(H_0^*, m_2)$, ahol $H_0^* = f(H_0, m_1)$

b.) A támadás nem működik MD padding mellett, mivel

$H(H_0, [m_1, m_2, \text{hossz}_1]) \neq H(H_0^*, [m_2, \text{hossz}_2])$.



Data Security: Protocols

Hash functions

n bites hash lenyomatot képező r -iterációs iteratív hash függvényből $n'=2n$ bites lenyomatot képzőt szeretnénk előállítani olyan módon, hogy a két utolsó iteráció eredményét konkatenáljuk azaz $H_{r-1}||H_r$ lenyomatot használunk, H_r helyett. Megfelelő-e ez a fajta konstrukciós ötlet?

Nem.

A kapott hash függvény ellen $2^{n/2}$ -nél kisebb számításigényű születésnap ütközéses támadás “végrehajtható”:

Elegendő csak magára az n bites H_{r-1} -re végrehajtani ezt a támadást, hiszen, ha $m=[M_1, M_2, \dots, M_{r-1}]$, $m'=[M'_1, M'_2, \dots, M'_{r-1}]$ üzenetpárra ütközés áll elő H_{r-1} -re, akkor egy fixen tartott H_r -rel meghosszabbítva m és m' üzeneteket, előáll az ütközés H_r -re, s így $H_{r-1}||H_r$ -re is. A nevezett támadás $2^{n/2}$ számításigényű maradt, tehát a dimenziónövekedés nem hatékony.



Data Security: Protocols

Party authentication (Tk.9.1-3. feladatok)

Tegyük fel, hogy egy A űrjármű leszálláshoz készülődik egy távoli bolygó B űrállomásán, s ehhez először azonosítania kell magát. A feltételek a következők:

1. B ismeri A jelszavát, ezen kívül más közös titkuk nincs.
2. A mod2 összeadásnál bonyolultabb műveletet nem tud végezni.
3. A lesugárzott jeleket egy, az űrállomás környéki C támadó is lehallgathatja, mivel nem lehet jól koncentrálni a sugárzást. Ugyanakkor a bolygón levő űrállomás képes úgy jeleket továbbítani, hogy azok a bolygó felszínén nem vehetők. Javasoljon egy kétlépéses protokollt az azonosításra!

Legyen pw a jelszó. B választ egy jelszó bitméretű r véletlen számot:

(1) $B \rightarrow A : r$

(2) $A \rightarrow B : pw+r \pmod{2}$

Data Security: Protocols

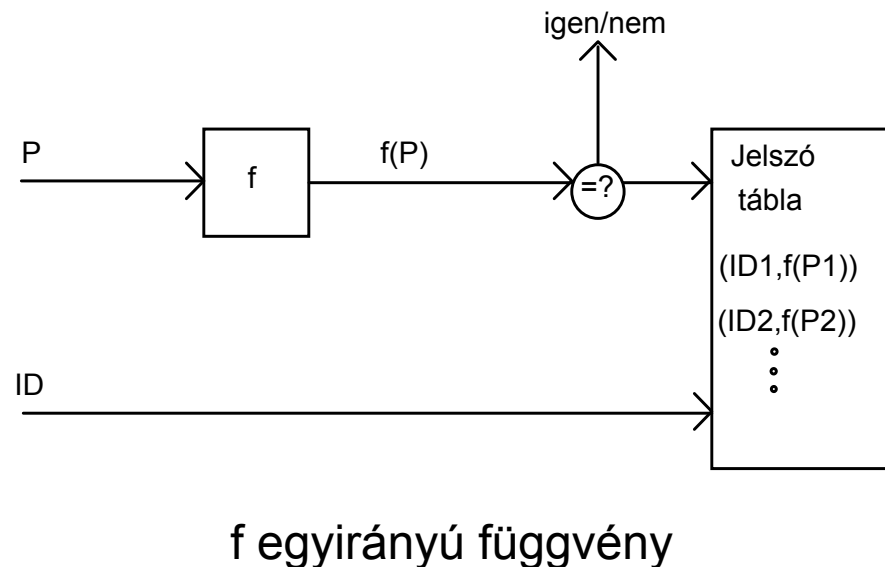
Party authentication (Tk.9.1-2.fejezetek)

Username:

Password:

A jelszavas rendszerek szokásos problémái:

- 1.) a nem megfelelő **jelszóválasztás**,
- 2.) a jelszó nyílt alakban történő **továbbítása** rendszerbe jutás pontjától (pl. terminál klaviatúra) az ellenőrzés pontjáig (pl. gazdagép),
- 3.) a jelszavak nem eléggé védett **tárolása** felhasználó oldalán, ellenőrzés oldalán (jelszófile)





Data Security: Protocols

Party authentication

Kihívás és válaszvárás (challenge and response)

1. A→B: r1
2. B→A: $y1=f(P2,r1)$
3. A: $y1=f(P2,r1)$?
4. B→A: r2
5. A→B: $y2=f(P1,r2)$
6. B: $y2=f(P1,r2)$?

- mindkét legális fél birtokában van partnere jelszavának (P1,P2)
- jelszavak nem kerülnek nyíltan átvitelre:
 - 'kihívás és válaszvárás' módszere (challenge and response)
 - véletlen elemek szerepe
- kommunikáló páronként kell egy-egy jelszó-párt egyeztetni
- a szótár alapú támadás veszélye nem csökkent

Data Security: Protocols

Party authentication

Egyszer használatos jelszó

Ini1.	A:	r generálása
Ini2.	A→B:	$ID_A, n, y=f^n(r)$
1.	A→B:	$P1=f^{n-1}(r)$
1.1	B:	$y=f(P1) ?$
2.	A→B:	$P2=f^{n-2}(r)$
2.1	B:	$y=f^2(P2) ?$
...		
i.	A→B:	$Pi=f^{n-i}(r)$
i.1	B:	$y=f^i(P2) ?$

Biztos lehet B abban, hogy A-val áll szemben? (hitelesítés)



Data Security: Protocols

Party authentication

Partnerhitelesítés nyilvános kulcsú rejtjelező függvények felhasználásával:

'kihívás és válaszvárás' típusú partnerazonosítási protokoll

B azonosítja magát A felé

1. B→A: R
2. A→B: $y = D_A(R)$
3. B: $R = E_A(y)$?

A protokoll biztonságosnak tűnik, de nem az:

Egy C támadó dekódoltat egy lehallgatott $y = E_A(x) = x^e \text{ mod } n$ rejtjelezett blokkot!

Védekezés: A ellenőrzi a 2.lépés eredményét, mielőtt továbbítaná B felé.



Data Security: Protocols

Party authentication

Finomított támadás (vak aláírás):

1. C választ egy R véletlen természetes számot, ahol $r < n$ és $(r, n) = 1$
2. C a következő előkészítő számításokat végzi el:

$$v = r^e \pmod n$$

$$w = vy \pmod n \quad (y = E_A(x) = x^e \pmod n)$$

$$t = r^{-1} \pmod n \quad (r = v^d \pmod n)$$

3. C megszemélyesíti B-t:

1'. $C(B) \rightarrow A$: w

2'. $A \rightarrow C(B)$: $u = D_A(w) = w^d \pmod n$

3'. $C(B)$: $tu = r^{-1}w^d = r^{-1}v^d y^d = v^{-d} v^d y^d = y^d = x \pmod n$,

ahol $r = v^d \pmod n$

Data Security: Protocols

Party authentication

Fiat-Shamir protokoll:

Kulcskiosztó központ (B):

p, q primek véletlen választása

$n = pq$ modulus

"A" ügyfél rendszerbe lépése (kulcsokat kap a központban):

u titkos kulcs (véletlen szám)

$v = u^2 \pmod{n}$ nyilvános kulcs

A protokoll alapeleme a következő négy lépés:

1. A \rightarrow B: $z = R^2 \pmod{n}$ ($0 < R < n$ véletlen szám)
2. B \rightarrow A: b (b véletlen bit)
3. A \rightarrow B: R ,ha $b=0$
 $w = R \cdot u \pmod{n}$,ha $b=1$
4. B: $z = R^2 \pmod{n}$? ,ha $b=0$
 $w^2 = z \cdot v \pmod{n}$? ,ha $b=1$

Data Security: Protocols

Party authentication

Hogyan próbálhatja egy C támadó megszemélyesíteni A felet?

1.) C végrehajtja az 1. lépést, megfigyeli a 2. lépésbeli b bitet.

$b=0$: C sikeres

$b=1$: C nehéz feladatot kap: gyököt vonjon $z \cdot v$ szorzatból modulo n
→ támadás sikervalószínűsége = $1/2$

1. A → B:	$z = R^2 \pmod{n}$	($0 < R < n$ véletlen szám)
2. B → A:	b	(b véletlen bit)
3. A → B:	R	,ha $b=0$
	$w = R \cdot u \pmod{n}$,ha $b=1$
4. B:	$z = R^2 \pmod{n} ?$,ha $b=0$
	$w^2 = z \cdot v \pmod{n} ?$,ha $b=1$

Data Security: Protocols

Party authentication

2.) C megpróbálja előrejelezni a 2. lépésben átküldésre kerülő b bitet.

C sejtése $b=0$:

1. $C \rightarrow B: z = R^2 \pmod{n}$

3. $C \rightarrow B: w = R \pmod{n}$

C sejtése $b=1$:

1. $C \rightarrow B: z = R^2 \cdot v^{-1} \pmod{n}$

3. $C \rightarrow B: w = R \pmod{n}$ (zv négyzetgyöke mod n)

b véletlen \rightarrow támadás sikervalószínűsége = $1/2$

1. $A \rightarrow B:$	$z = R^2 \pmod{n}$	$(0 < R < n$ véletlen szám)
2. $B \rightarrow A:$	b	(b véletlen bit)
3. $A \rightarrow B:$	R	,ha $b=0$
	$w = R \cdot u \pmod{n}$,ha $b=1$
4. $B:$	$z = R^2 \pmod{n} ?$,ha $b=0$
	$w^2 = z \cdot v \pmod{n} ?$,ha $b=1$



Data Security: Protocols

Party authentication

Támadás sikervalószínűségének csökkentése:

t -szer megismételjük a protokoll alapelemet
támadás sikervalószínűsége= 2^{-t}

A módszer további finomítása:

k db kulcs pár: $(v_1, v_2, \dots, v_k; u_1, u_2, \dots, u_k)$ $v_i = u_i^2 \pmod{n}$.

2. B \rightarrow A: b_1, b_2, \dots, b_k

3. A \rightarrow B: $w = R \cdot u_1^{b_1} u_2^{b_2} \dots u_k^{b_k} \pmod{n}$

4. B: $w^2 = z \cdot v_1^{b_1} v_2^{b_2} \dots v_k^{b_k} \pmod{n}$

támadás sikervalószínűsége= 2^{-tk}



Data Security: Protocols

Party authentication

Alkalmazás: intelligens kártyás azonosító rendszer

A: kártya és a tulajdonosa , B: azonosító terminál

Biztonsági cél: *hiteles terminálok biztonságosan azonosíthassák a kártyát (tulajdonosát), de csalási célú terminálok semmilyen használható titokhoz ne jussanak.* (terminálok üzemeltetője és a kártya kiállítója nem ugyanaz: pl. kártya egy pénzhelyettesítő POS (Point Of Sale) terminál felé)

$v=f(l,c)$,

f : kriptográfiai hash függvény:

v méret-szűkítése

különböző felhasználóknak különböző v nyilvános kulcs

l : ügyfél nyilvános azonosító([név,számlaszám,kártyaszám]),

c : (kicsi) nyilvános érték (választása: v kvadr. maradék legyen)

u : titkos kulcs a kártyán ki nem olvasható módon kerül elhelyezésre

A kártya lehetővé teszi a nyilvános adatelemek (l,c) ellenőrzését (olvasását)



Data Security: Protocols

Party authentication

Miért nem közvetlenül aknázzuk ki a mod n , $n=pq$ gyökvonás nehézségét?

1. $B \rightarrow A$: $R^2 \pmod{n}$
2. $A \rightarrow B$: $R \pmod{n}$

(Tf., hogy "A" ismeri p és q primeket is.)

A félnek gyököt kellene vonnia, s ennek a számításigénye jóval meghaladja a Fiat-Shamir protokoll számításigényét, amely ügyesen kikerüli a gyökvonás feladatát.