# KRIPTOGRÁFIAI PROTOKOLLOK

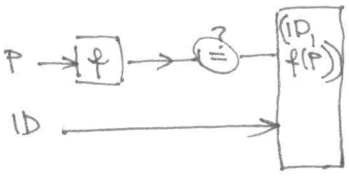
TITKOSÍTÓ ALGORITMUS = KÓDOLÓ/DEKÓDOLÓ TRF ⊕ KRIPTOGRÁFIAI PROTOKOLLOK

KRIPTOGRÁFIAI PROTOKOLLOK:
- kulcs védelme
- hitelesség biztosítása
- aktív támadásokkal szembeni védelem

Alapvető protokollok:
- partnerhitelesítés
- kulcskiosztás
- üzenetintegritás
- digitális aláírás
- titokmegosztás

## PARTNERHITELESÍTÉS

- Jelszavas partnerhitelesítés (PIN - kód)




azonosító: ID
jelszó: P
egyirányú fv: f

Szótár alapú támadások sikeresek lehetnek, ha az (ID, f(P)) párt olvassák el, még ha f egyirányú fv is, ha P nem elég véletlen.

- Egyszer használatos jelszó – változó jelszavas protokoll: A → B

Inicializálás: A: r szám generálása
A → B: $ID_A$, n, $y = f^{(n)}(r)$ elküldése

Kommunikáció:

1: A → B: $P_1 = f^{n-1}(r)$
B: $y \stackrel{?}{=} f(P_1)$ → komm.

2: A → B $P_2 = f^{n-2}(r)$
B: $y \stackrel{?}{=} f^2(P_2)$ → komm.

⋮

f nyilvános, r-t csak az A ismeri
fontos, hogy együtt lépjenek a hatványt/sorszámot.

- Partnerhitelesítés nyilvános kulcsú fvekkel

A-t szeretnénk azonosítani, $E_A$ nyílt, $D_A$ titkos

1.) B → A: r (véletlen)
2.) A → B: $y = D_A(r)$
3.) B: $E(y) \stackrel{?}{=} r$

Biztonságosabb:

1.) B → A: $r_2$ (véletlen)
2.) A → B: $D_A(r_1)$
A → B: $D_A(r_1 \oplus r_2) = z$
3.) B: $E_A(z) \stackrel{?}{=} r_1 \oplus r_2$

KULCSKIOSZTÁS

- Konvencionális algoritmussal, központtal

1.) $A \to Kp$: $ID_A, ID_B, r_1$

2.) $Kp \to A$: $E_{T,K_A}(r_1, ID_B, DK, E_{T,K_B}(DK, ID_A))$

3.) $A \to B$: $E_{T,K_B}(DK, ID_A)$

4.) $B \to A$: $E_{DK}(r_2)$

5.) $A \to B$: $E_{DK}(r_2 - 1)$

$T,K_A$, $T,K_B$ terminálkulcsok
$DK$ kapcsolatkulcs (régi, ismert DK-val lehet támadni)

– IDŐPECSÉT – szükség az órák szinkronizálására

1.) $A \to Kp$: $ID_A, ID_B$

2.) $Kp \to A$: $E_{TK_A}(T, L, DK, ID_B)$ $E_{TK_B}(T, L, DK, ID_A)$

3.) $A \to B$: $E_{DK}(T', ID_A)$, $E_{TK_B}(T, L, DK, ID_A)$

4.) $B \to A$: $E_{DK}(T'+1)$

- Nyilvános kulcsú algoritmussal

$C_A, C_B$ a $k_A^P$ és $k_B^P$ nyilvános kulcsok tanúsítványai
melyekben a bizpat $(ID_A, k_A^P)$ együttesre ad.
t bizpont nyilvános kulcsait V résztvevő ismeri

1.) $A \to B$: $ID_A, k_A^P, C_A$

$B \to A$: $ID_B, k_B^P, C_B$

$A \to B$: $E_B(r_1)$

$B \to A$: $E_A(r_2)$

$A, B$: $k = F(r_1, r_2)$

ÜZENETHITELESÍTÉS:

Cél: azon események detektálhatóvá tétele a vevőnél, melyek a kódolás és a vétel között megváltoztatták az üzenetet.

- Kriptográfiai ellenőrző összeg (MAC) (message authentication code)

Blokk-kódolás

$[x_1, x_2, \dots, x_r) \to (x_1, x_2, \dots, x_r, MAC(x_1, \dots, x_r))$

$MAC = g(y_r)$ m db bitet választ ki, annak az ell. összege.

ahol $y_i = E_k(x_i \oplus y_{i-1})$ $i = 1, \dots, r$, $y_0 = I$ inicializáló blokk

Ut vevő ismeri a MAC számítási módját, → ellenőrzik.

- Konvencionális rejtjelezéssel való hitelesítés

Szükséges a szöveg struktúráltsága v. struktúráltá tétele – Manipulation Detection Code (pl CRC)
cél, hogy a rejtjelezett üzenet struktúrája megtörjön manipulációs esetén
Hosszú üzenet esetén blokkok kihagyhatók lesznek:
→ Cipher Block Chaining – blokkláncolás

- Titkos kulcs nélküli üzenethitelesítés

  Egyirányú lenyomatkiszámító fv (= Hash-fv) segítségével MAC helyett csak a nyílt üzenet Hash-fves lenyomata szerepel ellenőrző összegként

  HASH-FÜGGVÉNY: olyan egyirányú leképezés, melynél nehéz feladat azonos hashképre (lenyomatra) vezető ősképeket találni.

  X, Hash(X) kerül átvitelre  Hash nyilvános

  kell telefonos (közvetlen) összeköttetés, hogy A felhívhassa B-t, hogy olvassa be a Hash érték kellően sok karakterét

- Digitális aláírás
  - aláírás generálása (küldő)
  - aláírás ellenőrzése (fogadó)
  - vitás kérdések tisztázása (3. fél)

A DIGITÁLIS ALÁÍRÁS TULAJDONSÁGAI
- könnyen generálható
- ne legyen áthelyezhető, hamisítható
- bárki ellenőrizhesse hitelességét

Nyilvános kulcsú titkosítással
- A akar hitelesíteni:

1.) A→B: $D_A(x)$ vagy $[x, D_A(x)]$

2.) B: $x = E_A(D_A(x))$ vagy $x \stackrel{?}{=} E_A(D_A(x))$

- csak A ismeri $D_A$-t → nem hamisítható
- $D_A(x)$ függvénye x-nek → nem átírható → x nem módosítható
- $E_A$ ismert → B-nek nincs szüksége A-ra egy 3. fél előtti bizonyításhoz

- Ha túl nagy az üzenet → $x, D_A(\text{Hash}(x))$ LENYOMATKÉSZÍTÉS

- IDŐPECSÉT
- IDŐPECSÉT + 3. SZEMÉLY: a 3. személy megbízható

1.) A → 3.Sz.: $U = S_A(I, S_A(x))$ I fejléc, $S_A$ aláírás

2.) 3.Sz: $V_A(U)$ $V_A$ validálás

3.) 3.Sz → A $W = S_3(T, I, S_A(x))$ $S_3$ aláírás

3.Sz → B $W = S_3(T, I, S_A(x))$

4.) A: $V_3(W)$ $V_3$ validálás

5.) B: $V_3(W), I, V_A(S_A(x))$

- Titok megosztása

  N részre osztva a titok

  K résztulajdonos tudja reprodukálni a titkot ($K<N$)

  — + lehetséges titkok halmaza $S$, elemeinek indexe $\{0,1,\dots,q-1\}$

  Válasszunk $N-1$ db véletlen elemet $S$-ből

  $r_1, r_2, \dots, r_{N-1}$ egyenletes eloszl.

  Legyen a megosztani kívánt titok $s$

  és

  $$r_N = S - (r_1 + r_2 + \dots + r_{N-1})$$

  Ha $N$ személy kapja meg $r_1, r_2, \dots, r_N$ elemeket

  • mind az $N$ szükséges a titokhoz

  — + Reed-Solomon-kód $(N,K)$ prím-ekkel $GF(q)$ felett

  $$\underline{r} = (r_0, r_1, \dots, r_{K-1}) \xrightarrow{G} \underline{c} = (c_0, c_1, c_2, \dots, c_{N-1})$$

  • $r_N$ jegyik eleme olyan, mint a fenti $r_N$

  • $c_i$-t osztjuk szét

  → $N-K$ törléses hiba javítható.

  (Polinomos repr.