Kriptográfia I

Szimmetrikus kulcsú titkosítás

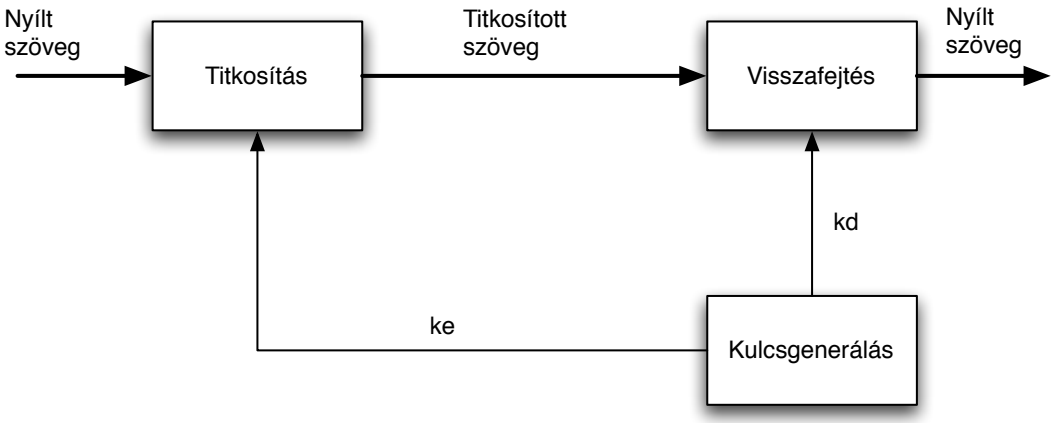
Kriptorendszerek

- Nyíltszöveg üzenettér: M
- Titkosított üzenettér: C
- Kulcs tér: K, K'
- Kulcsgeneráló algoritmus: $G:\mathbb{N}\mapsto K\times K'$
- Titkosító algoritmus: $E:M\times K\mapsto C$
- Visszafejtő algoritmus: $D:C\times K'\mapsto M$
- Titkosítás és visszafejtés:

$$c = E_{ke}(m) \qquad m = D_{kd}(c) \qquad D_{kd}(E_{ke}(c)) = m$$

- Szimmetrikus és aszimmetrikus kulcsú kriptorendszerek: $ke \in K,\ kd \in K',\quad ke = kd$ ill. $ke \neq kd$

Kriptorendszerek (folyt.)



Elvárások, tulajdonságok

- E,D algoritmusok nem tartalmaznak titkos részeket
- E egyenletesen teríti szét az üzeneteket a titkosított üzenettérben (esetleg véletlen elosztással) $\lim_{n\to\infty} \bigcup_n F^n R = \Omega$
- A megfelelő kulccsal E, D hatékonyan üzemeltethető
- A visszafejtés nehézségét a kulcs mérete (s) határozza meg, ideje p(s)-nél rosszabb

Helyettesítéses titkosítás

$E_k(m)$ Helyettesítő függvény $\pi : M \mapsto C$

$D_k(c)$ Fordított helyettesítő függvény $\pi^{-1} : C \mapsto M$

Például:

M=C=$\mathbb{Z}_{26}$

A=0, B=1,...Z=25

$$E_k(m) \quad \begin{pmatrix} 0 & 1 & 2 & \dots & 25 \\ 21 & 12 & 25 & \dots & 6 \end{pmatrix}$$

$$D_k(c) \quad \begin{pmatrix} 0 & 1 & 2 & \dots & 25 \\ 24 & 21 & 15 & & 2 \end{pmatrix}$$

Kulcstér mérete $26! = 4 * 10^{26}$

Gyenge titkosítás: term. nyelvek frekvencia analízésével megfejthető

Eltolásos titkosítás

$K = M = C$

$N := \# M$

$$\begin{cases} E_k(m) \leftarrow m + \mathrm{k} \pmod{\mathrm{N}} \\ D_k(c) \leftarrow c - k \pmod{\mathrm{N}} \end{cases}$$

Caesar titkosítás: k=3

Ha lnko(k,N)=1, akkor minden m<N-re
km (mod N) lefedi $\mathbb{Z}_N$-t

$$\begin{cases} E_k(m) \leftarrow \mathrm{km} \pmod{\mathrm{N}} \\ D_k(c) \leftarrow k^{-1}\mathrm{c} \pmod{\mathrm{N}} \end{cases} \qquad \begin{cases} E_k(m) \leftarrow \mathrm{k_1 m + k_2} \pmod{\mathrm{N}} \\ D_k(c) \leftarrow k_1^{-1}(\mathrm{c - k_2}) \pmod{\mathrm{N}} \end{cases}$$

Polialfabetikus titkosítás

- A nyílt szöveg elemek több titkosított üzenettér elemre képződnek le

- Vigenère titkosítás
  - m hosszúságú kulcs
  - nyílt szöveg felosztása m hosszúságú blokkokra
  - eltolásos titkosítás blokkonként (eltolások a kulcs alapján)

Vernam titkosítás

$m = b_1 b_2 ... b_n \in \{0,1\}^n$

$k = k_1 k_2 ... k_n \in_U \{0,1\}^n$

$c_i = b_i \oplus k_i$

Helyettesítéses titkosítás speciális esete $M = C = K = \{0,1\}^*$

Egyszer használt kulccsal információ-elméletileg erős titkosítás.

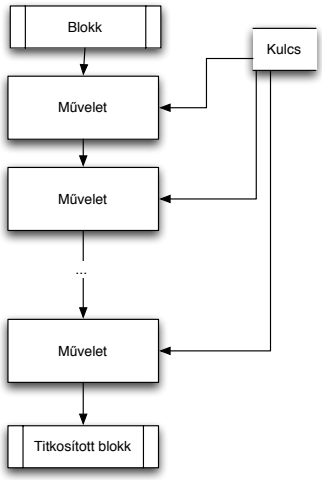
Transzpozíciós titkosítás

- Permutációs titkosítás
  - üzenet elemeinek átrendezése
  - kulcs: $\pi = (\pi(1), \pi(2), ...\pi(b))$
  - titkosás: $e_\pi(x_1, x_2, ...x_b) = (x_{\pi(1)}, x_{\pi(2)}, ...x_{\pi(b)})$
  - visszafejtés: $\pi^{-1}(\pi(i)) = i, i = 1,2,...b$

$$d_\pi(y_1, y_2, ...y_b) = (y_{\pi^{-1}(1)}, y_{\pi^{-1}(2)}, ...y_{\pi^{-1}(b)})$$

A Feistel architektúra

- Üzenet blokkok
- Kulcs (adott mérettel)
- Számítási menetek (rögzített szám)
  - alkulcsok
  - gyorsan elvégezhető műveletek



A DES

- Data Encryption Standard – United States' National Bureau of Standards 1977
- Blokk titkosítás $M = C = \{0,1\}^{64}$ $K = \{0,1\}^{56}$
- 3 lépéses működés
  - kezdeti permutáció

    $(L_0, R_0) \leftarrow IP(InputBlock)$
  - 16 menet $(i = 1,2,...16)$

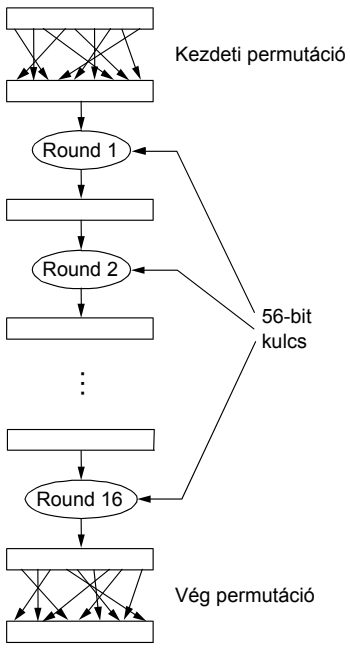
    $L_i \leftarrow R_{i-1}$

    $R_i \leftarrow L_{i-1} \oplus f(R_{i-1}, k_i)$

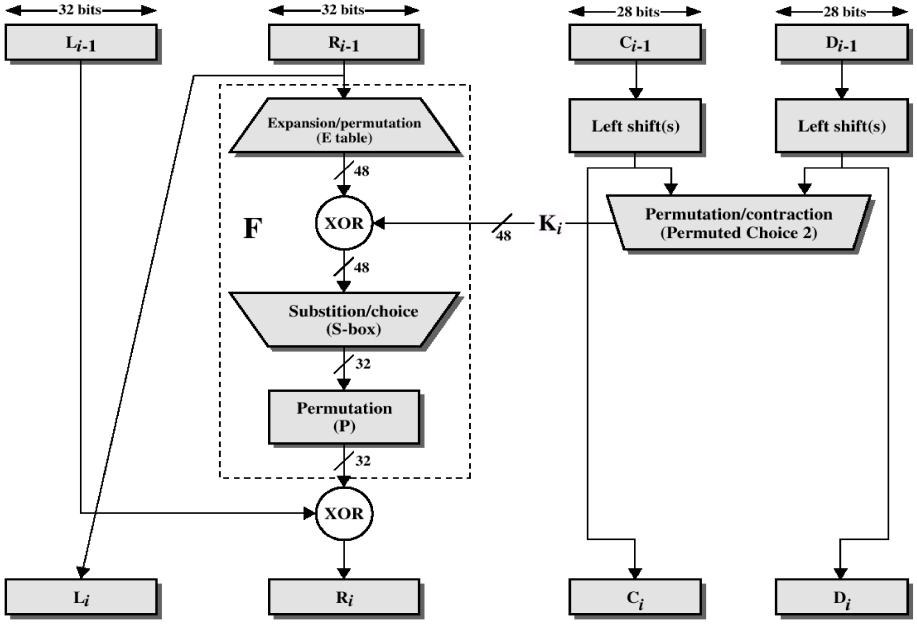
    k_i menet kulcs 48 bit
    f: S-box
  - végső permutáció

    $OutputBlock \leftarrow IP^{-1}(R_{16}, L_{16})$

A DES (folyt.)

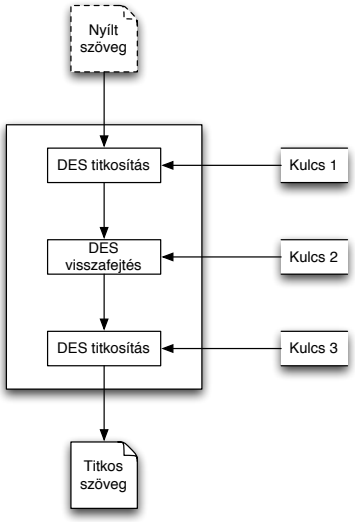


A DES egy lépése



A DES biztonsága

- Az S-box nem-lineáris működése fontos, a differenciális kriptoanalízis megelőzésére
- Rövid kulcshossz
  - brute force támadás
  - Deep Crack 1998 56 óra
- Triple DES séma



Az AES

- Advanced Encription Standard – NIST 2000
  - Rijndael algoritmus
  - Változtatható blokk és kulcsméret (128,192,256 bit)

A Rijndael titkosítás

- 128 bites blokkok

$$InputBlock = m_0 m_1 ... m_{15}$$

$$InputKey = k_0 k_1 ... k_{15}$$

$$InputBlock = \begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix}$$

- 10 menet

```
Round(State,RoundKey){
   SubBytes(State);
   ShiftRows(State);
   MixColumns(State);
   AddRoundKey(State,RoundKey);
}
```

- Invertálható transzformáció a visszafejtéshez

A Rijndael titkosítás (folyt.)

- Belső függvények véges test felett működnek RijndaelField(2^8)
- SubBytes $x \in (F_{2^8})^*$ $y = Ax^{-1} + b$
- ShiftRows
  - transzpozíció
- MixColumns
  - 4 oszlopra polialfabetikus helyettesítés
- AddRoundKey
  - bitenkénti XOR a kulccsal
  - key schedule séma szerint

Rijndael jellemzők

- Gyors implementáció lehetősége
  - véges test feletti műveletekhez logaritmus lookup táblázatok
  - timing analízis kizárása
- Származtatott jól használható hash függvények

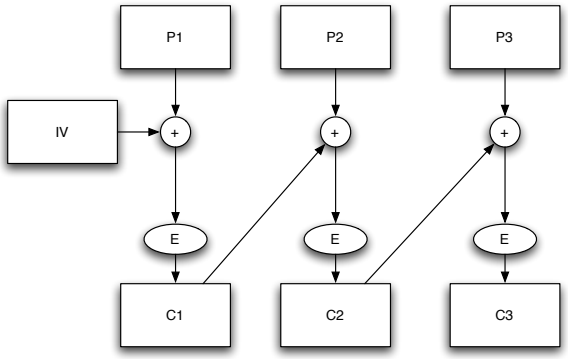
Blokk titkosítások biztonságának javítása

- Elektronikus kódkönyv mód (ECB)
  - üzenet szegmensek egymástól független kódolása
- Kódblokk láncolás mód (CBC)
  - blokk kódolása függ az előző blokkoktól
- Titkosított blokk visszacsatolás (CFB)
  - előző titkosított blokk visszacsatolása
- Kimenet visszacsatolás (OFB)
  - blokk titkosítás kimenetének visszacsatolása

CBC

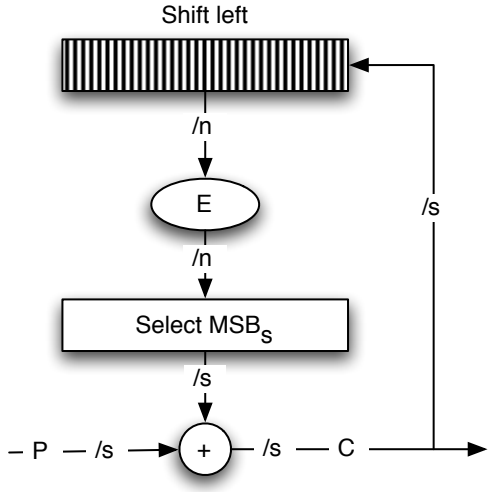
$$C_0 \leftarrow IV$$

$$C_i \leftarrow E(P_i \oplus C_{i-1})$$



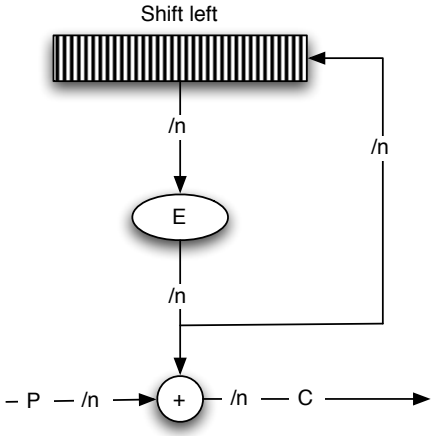
CFB

$$I_1 \leftarrow IV$$
$$I_i \leftarrow LSB_{n-s}(I_{i-1}) \| C_{i-1}$$
$$O_i \leftarrow E(I_i)$$
$$C_i \leftarrow P_i \oplus MSB_s(O_i)$$



OFB

$$I_1 \leftarrow IV$$
$$I_i \leftarrow O_{i-1}$$
$$O_i \leftarrow E(I_i)$$
$$C_i \leftarrow P_i \oplus O_i$$



Kulcscsatornák

- Hagyományos technikák
- Nyílvános kulcsú technikák