Blokk rejtjelezők

PPKE, ITK
Csapodi Márton

Definíciók

n: nyílt és rejtjelezett szöveg blokkmérete, általában nincs expanzió

K: k bites kulcs, véletlenszerűen kiválasztva

a tényleges kulcsméret lehet <k

blokk rejtjelező: nyílt szöveg és kulcs leképezése rejtjelezett szövegre, mely minden kulcs esetén invertálható

rejtjelezés: $C=E_K(P)$

dekódolás: $P=D_K(C)$

Az összes lehetséges P->C bijekció száma: $2^n!$

valódi véletlen rejtjelező: minden lehetséges bijekciót megvalósít, óriási kulcsméretet eredményez, ha n nem "túl kicsi"

praktikusan: véletlennek tűnő rejtjelező (véletlen kulcs esetén, vagy a kulcs ismerete nélkül)

Támadások

A blokk rejtjelező a bizalmasságot védi

passzív támadás <-> aktív támadás
A passzív támadás fenyegeti a bizalmasságot

Feltételezés:

- minden rejtjelezett szöveg megismerhető
- csak a kulcs ismeretlen (Kerckhoff feltétel)

További osztályozás: mit ismer még a rejtjelezett szövegen kívül:

- ciphertext-only
- known-plaintext
- (adaptive) chosen-plaintext
- (adaptive) chosen-ciphertext

Támadások

tervezési elv: chosen-plaintext támadás ellen legyen védett

rejtjelező biztonsága: a leghatékonyabb támadás komplexitásával arányos

komplexitás szempontjai:

- data complexity
- storage complexity
- processing complexity

kimerítő keresés:

data/storage: max 2^n

processing: max 2^k

Jó rejtjelező: mindegyik komplexitás közel a határhoz

Kimerítő kulcskeresés

Ismert nyílt szöveg (known-plaintext) vagy redundáns nyílt szöveg esetén nagyságrendileg 2^{k-1} próbálkozással a kulcs meghatározható

Példa (redundáns nyílt szöveg):

8 ASCII karakterből + karakterenként paritás bitből álló blokk (64 bit)

k=56 (DES)

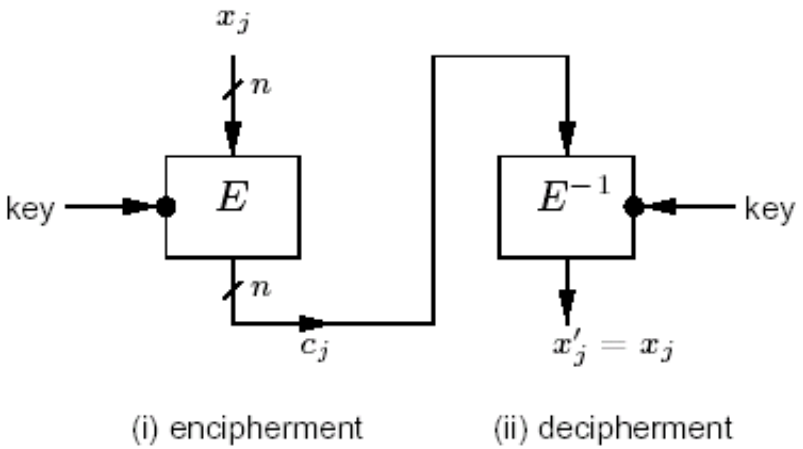
Rossz kulccsal is lehet korrekt paritású nyílt szöveget kapni, de további rejtjelszöveg blokkok dekódolása kiszűri a rossz kulcsot néhány blokk után

Működési módok / ECB

ECB: Electronic Codebook

jellemzők:

- azonos nyílt szöveg (és kulcs)
  -> azonos rejtjelezett szöveg
- blokkok függetlenek egymástól
- hiba esetén csak egy blokk hibás



korlátozásokkal alkalmazható

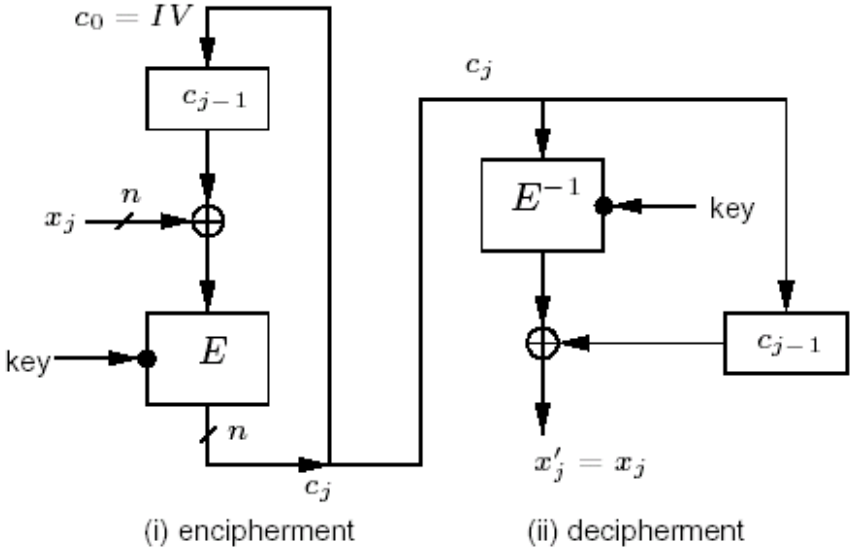
véletlen "padding" segít

Működési módok / CBC

CBC: Cipher-block Chaining

jellemzők:

- azonos nyílt szöveg (és kulcs)
  -> eltérő rejtjelezett szöveg
  véletlen IV-vel
- dekódoláshoz kell az előző rejtjelszöveg
- hiba esetén csak két dekódolt blokk hibás



IV lehet nyílt, de integritása szükséges (1. blokk biztonságos dekódolása miatt)

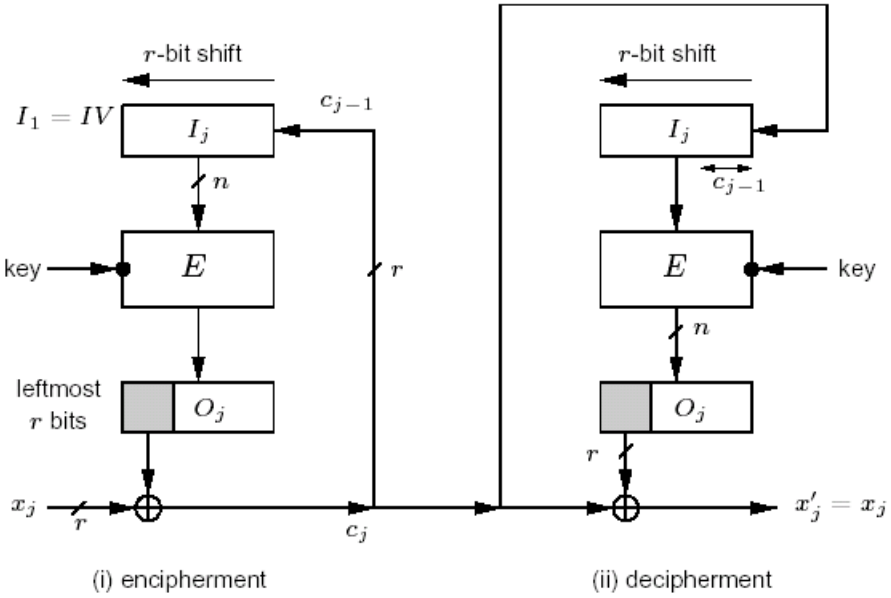
Működési módok / CFB

CFB: Cipher Feedback

jellemzők:

- azonos nyílt szöveg (és kulcs)
  -> eltérő rejtjelezett szöveg
  véletlen IV-vel
- dekódoláshoz kell
  az előző néhány rejtjelszöveg
- hiba esetén csak néhány dekódolt blokk hibás
- alacsonyabb hatásfok r<n miatt

IV lehet nyílt

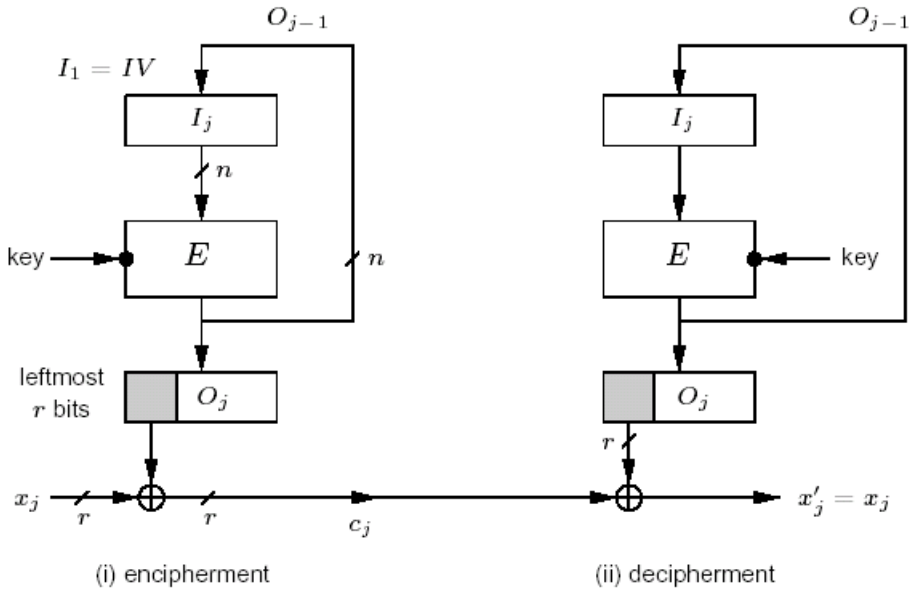


Működési módok / OFB

OFB: Output Feedback

jellemzők:

- azonos nyílt szöveg
  -> eltérő rejtjelezett szöveg
  véletlen IV-vel
- a dekódoló kulcsfolyam független
  a rejtjelszövegtől
- bithiba esetén csak egy dekódolt blokk hibás, de egy rejtjelszöveg bit kiesése esetén kiesik a szinkronból



IV lehet nyílt, de változtatni kell a kulcs újbóli használatakor

Visszacsatolás helyett lehet egy számláló is (CTR mode)

Működési módok

Megjegyzések:

1.

OFB, CTR: Szinkron adatfolyam rejtjelező kulcsgenerátora

CFB: Aszinkron adatfolyam rejtjelező

2.

CFB és OFB (CTR) módok esetén csak rejtjelezés (E(P)) számítása szükséges, erre érdemes optimalizálni (pl. AES)

Jellegzetes blokk rejtjelezők

szorzat rejtjelező (product cipher): helyettesítés, áthelyezés, lineáris leképezés, aritmetikai művelet, moduláris szorzás egymás utáni végrehajtása

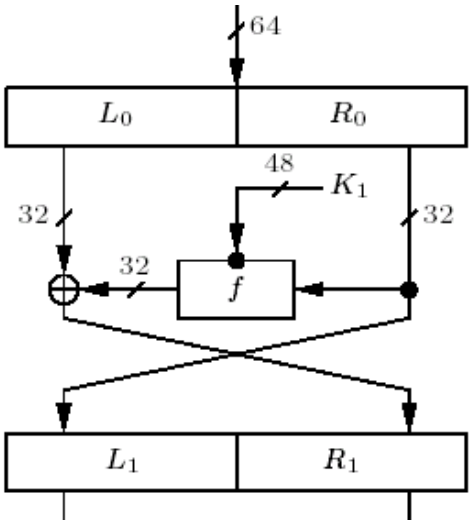
iterált rejtjelező: egy belső invertálható funkció ismétlése, minden lépésben a kulcsból generált alkulcs alkalmazásával

helyettesítéses-permutációs hálózat (substitution-permutation network): iterált szorzat rejtjelező egyszerű helyettesítésekből és permutációkból

Feistel rejtjelező: iterált rejtjelező, mely azáltal lesz invertálható, hogy minden lépésben tetszőleges f-re:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$



DES

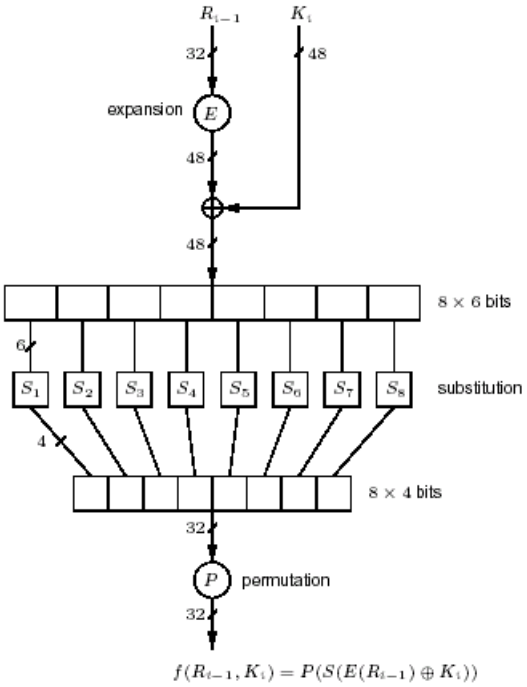
Feistel rejtjelező

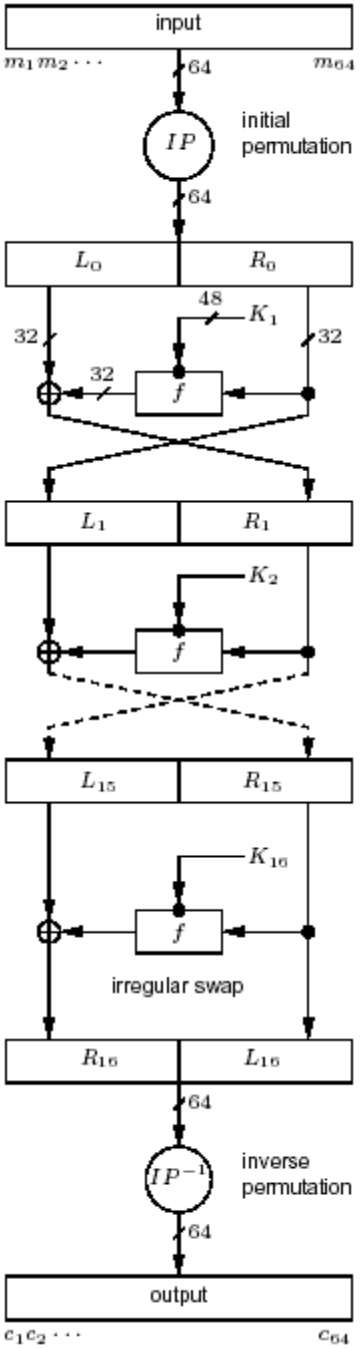
blokkméret: n=64

kulcsméret: k=64 bit, de a tényleges méret: 56 bit

8 paritás bit miatt

16 iteratív lépés





DES

Tulajdonságok:

minden rejtjelszöveg bit függ minden nyílt szöveg bittől: egy bit megváltoztatása minden rejtjel bitet 1/2 valószínűséggel változtat meg

nem jósolható, hogy egy rejtjelszöveg bit változtatása milyen hatással lesz a dekódolt szövegre

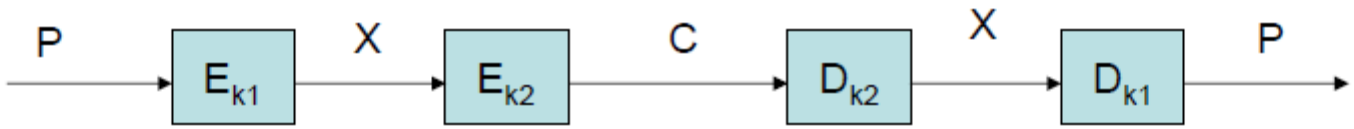
Nem csoporttulajdonságú: kompozícióra nem zárt, tehát két különböző kulccsal történő rejtjelezés egymás után nem váltható ki egy harmadik kulccsal történő rejtjelezésre

Emiatt van lehetőség többszörös rejtjelezésre (3DES):

- DES: nem biztonságos
- double DES: nincs értelme (meet-in-the-middle támadás, 2^{57} művelet)
- triple DES: kettő vagy három kulccsal (itt is van meet-in-the-middle támadás), de lassú

Double DES: Meet in The Middle Attack

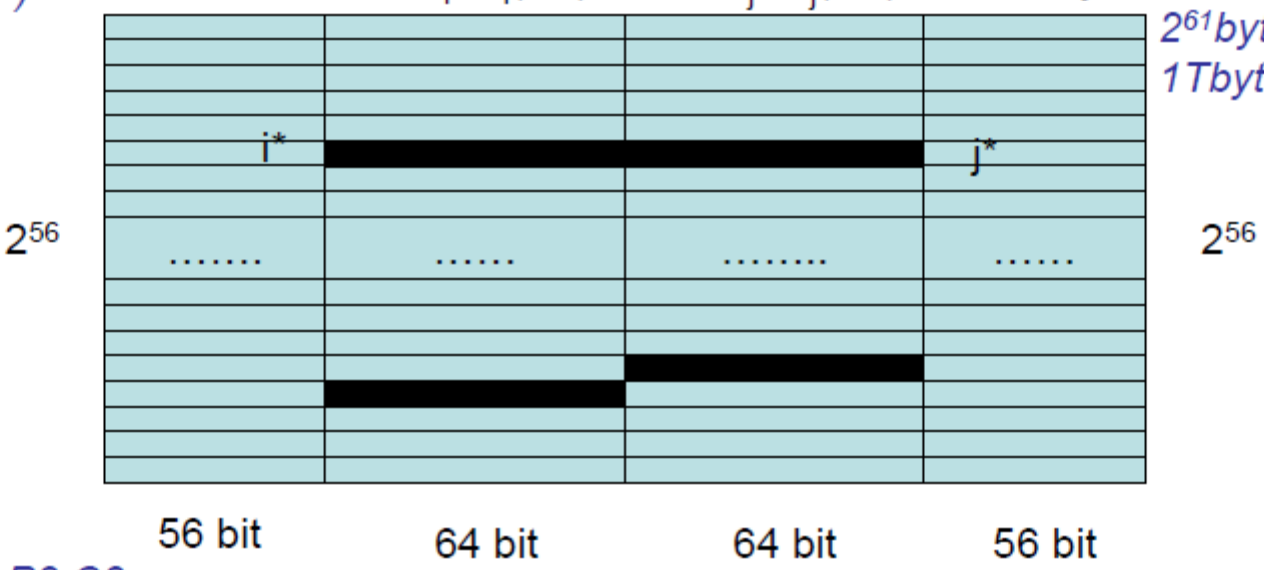
Known Plaintext Attack, two pairs: P1;C1 & P2;C2



$X=E_{k1}(P)$ $X=D_{k2}(C)$

First Pair: P1;C1
To find keys (i;j*)*
• Xi= Xj**

Total Memory
$256x2^{56}=2^{64}$bit
2^{61}byte~10^{18}byte
1Tbyte x 10^6run



Second Pair: P2;C2
To check keys (i;j*): P2=Di*[Dj*(C2)]; C2=Ej*[Ei*(P2)]; Average Run # = 2x 2^{56}/2= 2^{56}*