

# Data Security: Secret key

Kulcsfolyamatos rejtjelezést tekintünk, azaz a kulcsbiteket mod 2 hozzáadjuk a nyílt szöveg bitekhez.

A kulcsot első 5 bitjéből periódikus ismétléssel nyerjük, az első 5 bitet jelölje  $k_1, \dots, k_5$ .

A nyílt szöveg egymás utáni bitjei,  $x_1, x_2, \dots$  képződési szabálya,  $x_i + x_{i+1} = x_{i+2}$ ,  $i=1, 4, 7, \dots$

Az első 15 megfigyelt rejtett szöveg bit a következő: 010101111100001.

Fejtsük meg a kulcsot!

# Data Security: Secret key

$$x_3 = x_1 + x_2 = y_1 + y_2 + k_1 + k_2 = k_1 + k_2 + 1$$
$$x_3 = k_3 \quad (y_3 = x_3 + k_3 = 0)$$

010 101 111 100 001

$$x_6 = x_4 + x_5 = k_4 + k_5 + 1$$
$$x_6 = k_1 + 1 \quad (\text{kulcsperiódus}=5)$$

$$x_9 = x_7 + x_8 = k_2 + k_3$$
$$x_9 = k_4 + 1$$

$$x_{12} = x_{10} + x_{11} = k_5 + k_1 + 1$$
$$x_{12} = k_2$$

$$x_{15} = x_{13} + x_{14} = k_3 + k_4$$
$$x_{15} = k_5 + 1$$

alapján

$$k_1 + k_2 + k_3 = 1$$

$$k_1 + k_4 + k_5 = 0$$

$$k_2 + k_3 + k_4 = 1$$

$$k_1 + k_2 + k_5 = 1$$

$$k_3 + k_4 + k_5 = 1$$

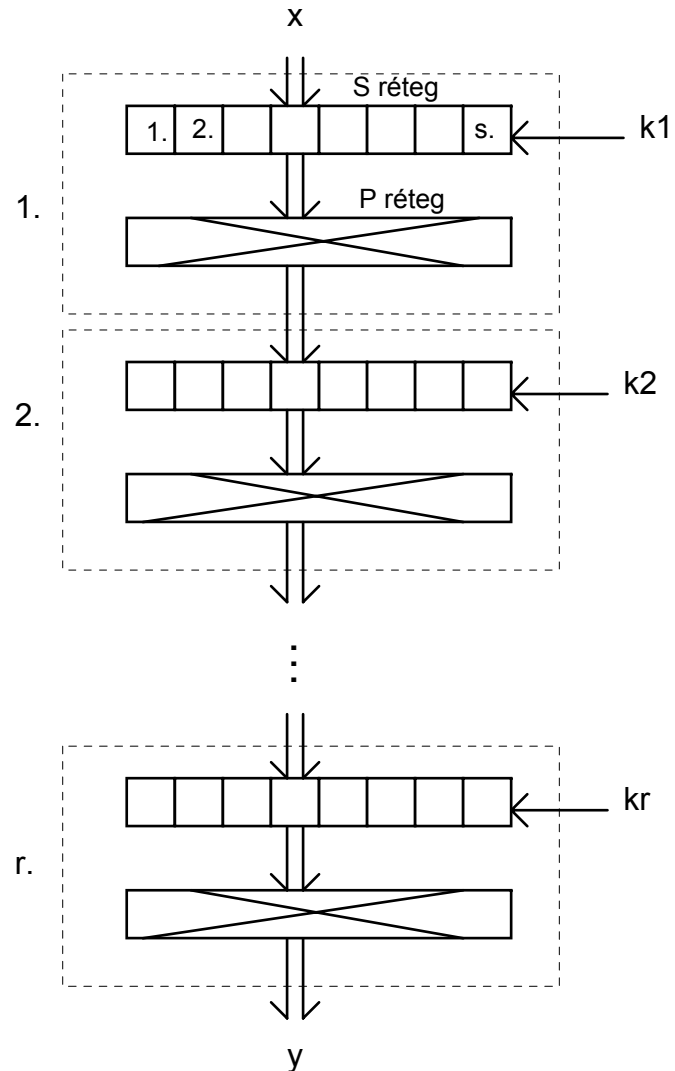
→  $K = (1, 0, 0, 1, 0)$ .

# Data Security: Secret key

## SPC (Substitution Permutation Cipher)

**Shannon-i elv:** *Erős invertálható transzformáció előállítható egyszerű, könnyen analizálható és implementálható, de önmagában gyenge transzformációk sokszori egymás utáni alkalmazásával.*

Példa: szimmetrikus kulcsú rejtjelezők (pl. DES, IDEA, AES)



# Data Security: Secret key

## SPC

### SPC tervezési kritériumok

- Invertálhatóság
- Balansz tulajdonság
- Teljesség
- Nemlinearitás
- Lavinahatás
- Lineáris dimenzió
- Differenciális egyenletesség
- ...

*Boole-függvény:*  $f : \{0,1\}^m \rightarrow \{0,1\}$

*S-box:*  $f : \{0,1\}^m \rightarrow \{0,1\}^n \quad 1 < n \leq m$

$f(x) = [f_1(x), f_2(x), \dots, f_n(x)] \quad f_i : \{0,1\}^m \rightarrow \{0,1\}$

# Data Security: Secret key

## SPC

### Invertálhatóság

$F = F_1 F_2 \dots F_r$  ,  $F_i$  az  $i$ -edik rétegbeli transzformáció

$$F^{-1} = F_r^{-1} F_{r-1}^{-1} \dots F_1^{-1}$$

### Balansz tulajdonság

*A transzformáció nem torzítsa el egy egyenletes eloszlású bemenet gyakoriság-statisztikáját.*

$$f : \{0,1\}^m \rightarrow \{0,1\}^n, \quad m \geq n$$

$$\#\{x \in \{0,1\}^m : f(x) = y\} = 2^{m-n} \quad \forall y \in \{0,1\}^n$$

### Példák

1.  $f(x) = \mathbf{A} \cdot x + b$      $\mathbf{A}$   $n \times m$ -es bináris mátrix, rang= $n$      $x \in \{0,1\}^m$ ,  $b \in \{0,1\}^n$

2.  $f : \{0,1\}^n \rightarrow \{0,1\}^n$     invertálható

# Data Security: Secret key

## SPC

### Nemlinearitás

Boole-függvények távolsága:

$$d(f, g) = \#\{x \in \{0,1\}^m : f(x) \neq g(x)\} = w(f + g) \quad f, g : \{0,1\}^m \rightarrow \{0,1\}$$

Lineáris Boole-függvény:

$$L_{u,v}(x) = u \cdot x + v \quad u, x \in \{0,1\}^m \quad v \in \{0,1\}$$

### Boole-függvény nemlinearitása

$$N(f) = \min_{u \in \{0,1\}^m, v \in \{0,1\}} d(f, L_{u,v})$$

### S-box nemlinearitása

$$N(f) = \min_{w \in \{0,1\}^n, w \neq 0} N(w \cdot f) \quad f : \{0,1\}^m \rightarrow \{0,1\}^n$$

# Data Security: Secret key

## SPC

Legyen  $n=2$ , továbbá  $f_1, f_2: V_2^n \rightarrow V_2^1$

jelölje az első illetve második output bitre vonatkozó transzformáció-komponenst

Ha  $f_1$  nemlinearitása  $N_1$ , mekkora  $f$  nemlinearitása, ha

1.)  $f_2 = f_1 \oplus 1$

2.)  $f_2 = 1$

1.) 0, mivel az outputok  $w=\{1,1\}$  súlyú lineáris kombinációja (bináris összege) konstans 1

2.) 0, mivel az outputok  $w=\{0,1\}$  súlyú lineáris kombinációja (a 2. kimenet) konstans 1.

# Data Security: Secret key

Lavinahatás kritérium

$$\frac{1}{2^m} \sum_{x \in \{0,1\}^m} w(f(x) + f(x + e_m^{(i)})) = \frac{n}{2} \quad 1 \leq i \leq m \quad f : \{0,1\}^m \rightarrow \{0,1\}^n$$

Szigorú lavinahatás kritérium

$$\frac{1}{2^m} \sum_{x \in \{0,1\}^m} (f(x) + f(x + e_m^{(i)})) = \left(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2}\right) \quad 1 \leq i \leq m \quad f : \{0,1\}^m \rightarrow \{0,1\}^n$$

# Data Security: Secret key

Differenciális egyenletesség

$$DDT_f(a,b) = |\{x \in \{0,1\}^m : f(x) + f(x+a) = b\}|$$

$$a \in \{0,1\}^m, b \in \{0,1\}^n$$

$$DDT_f(0,b) = 2^m \delta(b).$$

Példa:

Ha  $f=ux+v$  lineáris, akkor  $DDT_f(a,b) = 2^m$ , ha  $b=ua$ , egyébként 0

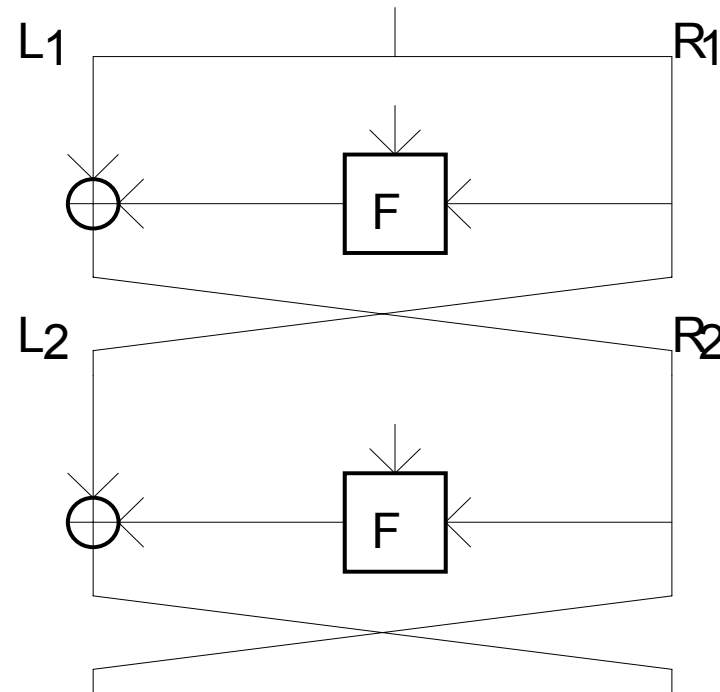
# Data Security: Secret key

## SPC

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i + F(R_i, K_i)$$

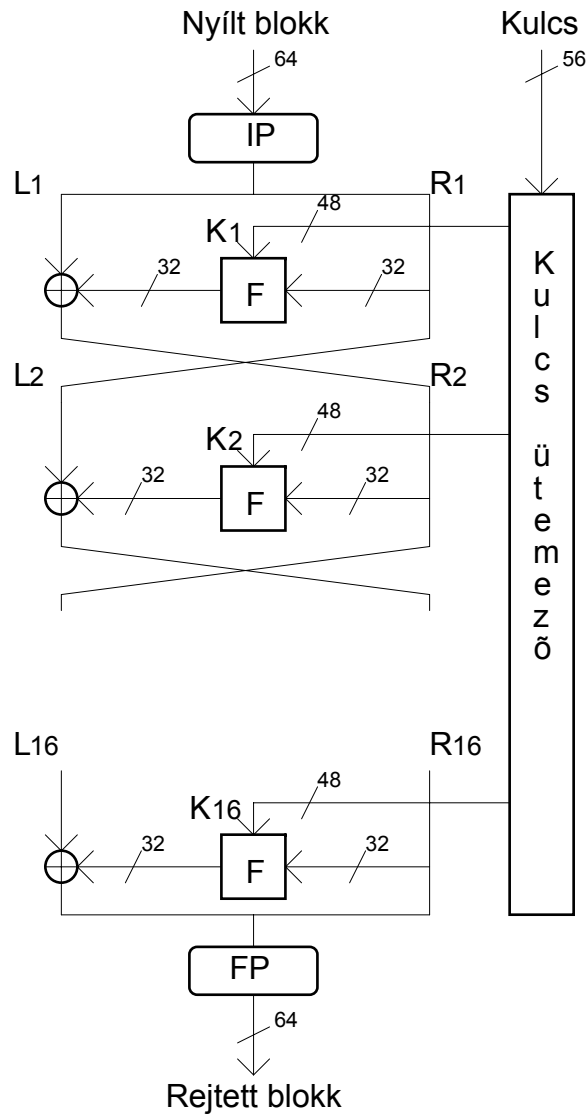
Invertálható, függetlenül attól, hogy  $F$   
invertálható, vagy sem!

$$L_j = R_{i+1} + F(L_{i+1}, K_i)$$
$$R_j = L_{i+1}$$



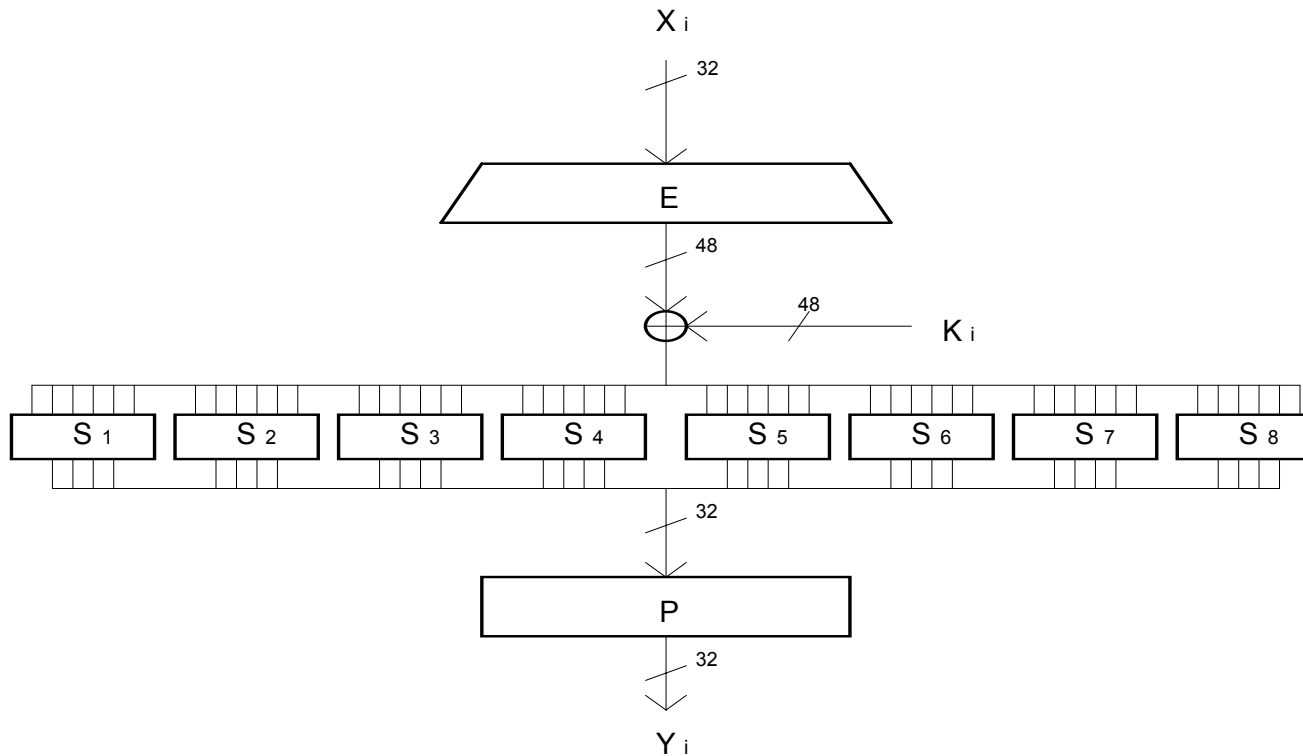
# Data Security: Secret key

## SPC



# Data Security: Secret key

## SPC



S1 box

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	5	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

# Data Security: Secret key

## SPC

### S-doboz tervezési kritériumok: DES

- Minden S-doboz bemenete 6, kimenete 4 bites legyen. (DES egyetlen chip-be integrálásához)
- Egyetlen S-doboz egyetlen kimeneti bitje se legyen közel a bemeneti bitek valamely lineáris függvényéhez. (Tehát a **neilinearitás** legyen nagy.)
- Ha rögzítjük a két szélső bit értékét, és csak a bemenet középső négy bitjét változtatjuk folyamatosan, akkor a kimeneten minden 4 bites vektor pontosan egyszer jelenjen meg. (Azaz az S-dobozban található 4 darab 4 bitet 4 bitbe helyettesítő tábla mindegyike legyen **balansz**. Ekkor persze maga az S-doboz is balansz, vagyis minden 4 bites kimeneti vektor pontosan négyszer jelenik meg, ha a bemeneten minden lehetséges értéket végigpörgetünk.)
- Ha az S-doboz bemenetén egyetlen bitet megváltoztatunk, akkor a kimeneten legalább két bit értéke változzon meg. (**lavinahatás**)
- Ha az S-doboz bemenetén a két középső bitet megváltoztatjuk, akkor a kimeneten legalább két bit értéke változzon meg.
- Ha két bemeneti vektor első két bitje különböző, utolsó két bitje azonos, akkor a megfelelő kimeneti vektorok nem lehetnek azonosak.
- Tetszőleges, nem nulla bemeneti differencia esetén, az adott differenciával rendelkező 32 bemeneti vektor pár közül legfeljebb nyolchoz tartozhat azonos kimeneti differencia. (nagy **differenciális egyenletesség**)

# Data Security: Secret key

## **P-doboz tervezési kritériumok: DES**

- A P-doboz legyen olyan, hogy minden S-doboz négy kimeneti bitje közül kettőt a következő réteg S-dobozainak középső bitjeihez, kettőt pedig szélső (táblázat választó) bitekhez továbbítson.
- Minden S-doboz négy kimeneti bitje a következő rétegben hat különböző S-dobozra legyen hatással.
- Ha egy S-doboz valamely kimeneti bitje egy másik S-doboz valamely középső bitjéhez van vezetve, akkor ez utóbbi S-doboz egyetlen kimenete sem lehet az előző S-doboz középső bemeneteihez vezetve.

# Data Security: Secret key

Tegyük fel, hogy DES rejtjelezést használunk 64 bites üzenetblokkon és belül a blokk végén 4 bites hibadetekciós ellenőrzőösszeget alkalmazunk.

Egy támadó már megismerte a kulcs első két bitjét, ezután kimerítő kulcskereséses támadást végez. 13 rejtjeles blokk megfigyelése elegendő-e a támadó számára a gyakorlatilag egyértelmű kulcsazonosításhoz?

(Tekintsük gyakorlatilag egyértelműnek a kulcsazonosítást, ha végül csak néhány kulcs közül kell a támadónak választania!)

Igen:

Annak a valószínűsége, hogy téves kulccsal helyes paritásúra dekódolunk egy rejtett szöveg blokkot,  $2^{-4}$ .

Annak a valószínűsége, hogy 13 rejtjeles blokk mindegyikét helyes paritásúra dekódoljuk téves kulcs mellett  $(2^{-4})^{13} = 2^{-52}$ .

Mivel a kulcstér mérete az előzetesen kiszivárgott két kulcsbit miatt már csak  $2^{54}$ , a nem kiszűrt téves kulcsok átlagos száma a kulcstér teljes végigkeresése után  $2^{54} \cdot 2^{-52} = 2^2 = 4$  lenne.

# Data Security: Secret key

## Birthday paradox

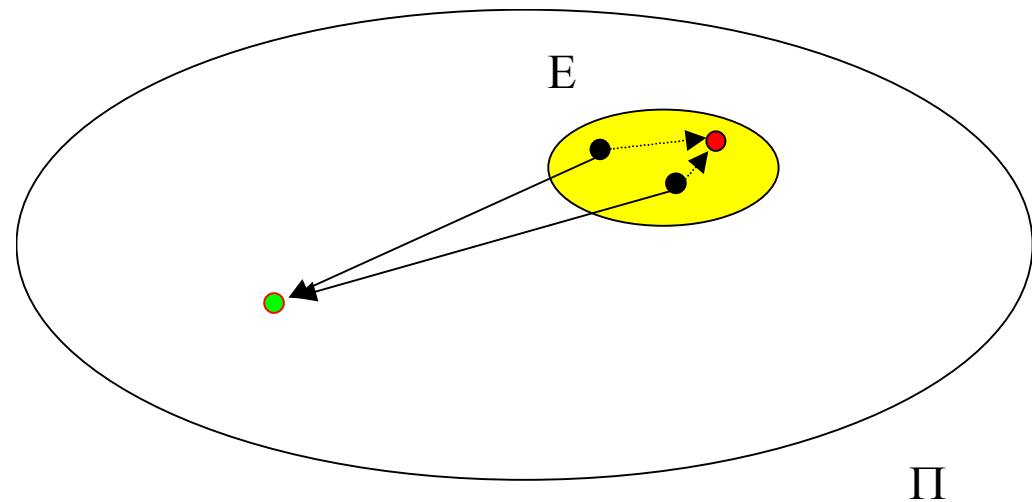
Algebrailag zárt blokk rejtjelező

$$E = \{E_k; E_k : X \rightarrow Y, k \in K\}$$

$$T = \{E, *\}.$$

$T$  zárt, ha  $EE = E$ ,

$$E_{k_1} \cdot E_{k_2} = E_{k_3}, k_i \in K$$



**Tétel** : A  $T$  zárt algebrai struktúra csoport.

# Data Security: Secret key

## Birthday paradox

### Születésnap paradoxon-1

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

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

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

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

Pl.

$$m = 365$$

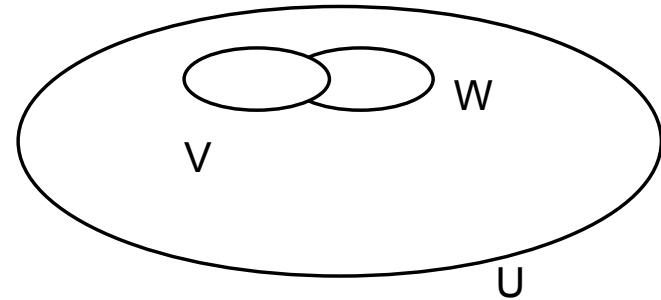
$$r = m^{1/2} \approx 19, \quad p_r \approx 1 - \exp(-0.5) \approx 0.4.$$

# Data Security: Secret key

## Birthday paradox

### Születésnapi paradoxon-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: Secret key

## Birthday paradox

Középen találkozás támadás zárt struktúrájú rejtjelező ellen

Támadó ismerete: ismert nyílt szövegű támadás

$$Q = \{(x_1, y_1), (x_2, y_2), \dots, (x_s, y_s)\},$$

$$y_i = E_k(x_i), \quad k \text{ az ismeretlen kulcs}$$

$$U = E$$

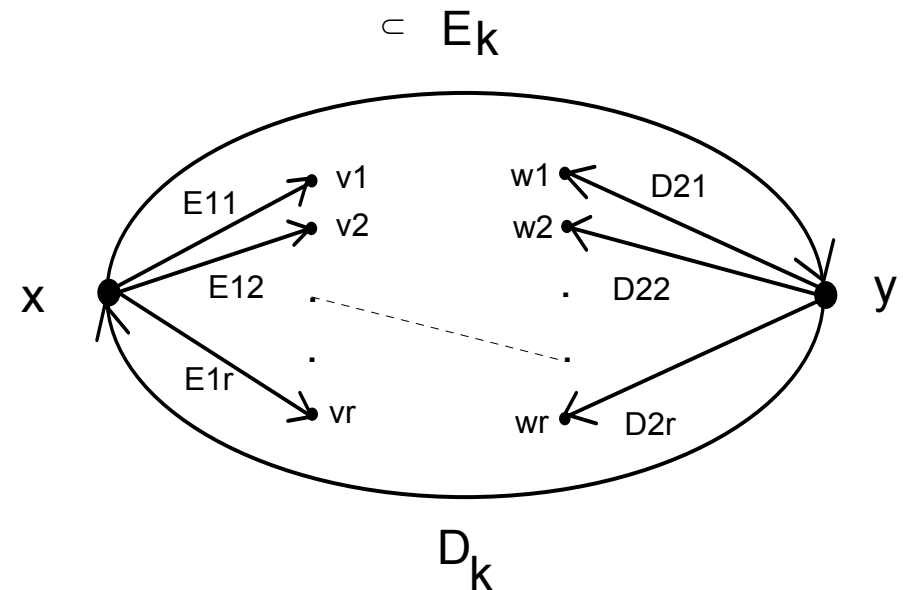
$$V = \{E_{11}, E_{12}, \dots, E_{1r}\}$$

$$W = \{D_{21}E_k, D_{22}E_k, \dots, D_{2r}E_k\},$$

ahol  $V, W \subset U$ .

$$\text{Ha } V \cap W \neq \emptyset, \text{ pl. } E_{1j} = D_{2i}E_k \rightarrow E_k = E_{2i}E_{1j}$$

$$r \approx |E|^{1/2} !$$



# Data Security: Secret key

## Birthday paradox

Két DES transzformációt egymás után használunk:

$$y = E_{k_2}(E_{k_1}(x))$$

ahol  $k_1, k_2$  két véletlen titkos kulcs.

Azt reméljük, hogy ezzel a DES 56 bites kulsméretének megfelelő kulcstér kimerítő keresés  $2^{56}$  nagyságrendű számításigényét  $2^{112}$  nagyságrendűre tudjuk emelni.

a.) Igazunk van-e? Milyen módszerrel támadna a támadó helyében, milyen adatok alapján.

b.) Mekkora becsli a támadás komplexitását (számításigény, tárkapacitás). (Számításigény a kódoló/dekódoló transzformációk számában. Tárkapacitás tárolandó blokkok számában.).

a.) Nem.

Középen találkozás támadás, ismert nyílt-rejtett szöveg párok alapján.

b.)  $2 \cdot 2^{56}$  a számításigény, illetve tárkapacitás nagyságrendje.

# Data Security: Secret key

## Birthday paradox

A számításokat gyorsítandó  $k=128$  kulcsbitünket nem egy biztonságos 128 bit kulcsméretű  $E^*$  blokk-rejtjelezőhöz használjuk, hanem két félre osztjuk  $k$  kulcsot, és

$$y = E^{**}_k(x) = E_{k1}(x \oplus k2)$$

rejtjelezést hajtunk végre, ahol  $E$  egy biztonságos 64 bites kulcsméretű blokk-rejtjelező,  $k1$  és  $k2$  64 bites felei a  $k$  kulcsnak,  $x$  egy 64 bites üzenetblokk.

A támadó megfigyelhet  $(x,y)$  nyílt-rejtett szöveg párokat. Az  $E^*$  és  $E$  rejtjelező csak kimerítő kulcskereséssel támadhatók.

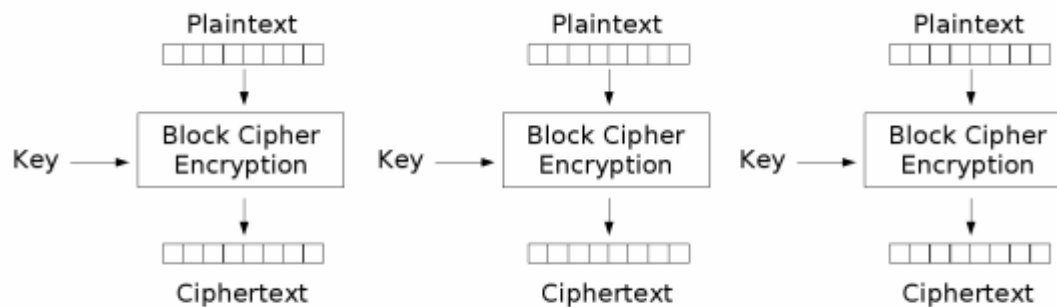
- Vesztettünk-e a támadhatóság okán vagy sem, hogy  $E^*$  helyett  $E^{**}$  rejtjelezést alkalmazzuk?
- Hasonlítsa össze, kulcskereső támadás számításigényét a két esetben!

a.) Igen, sokat veszítettünk, mivel  $E^{**}$  két blokk rejtjelező kaszkádja, így középen találkozás támadással támadható.

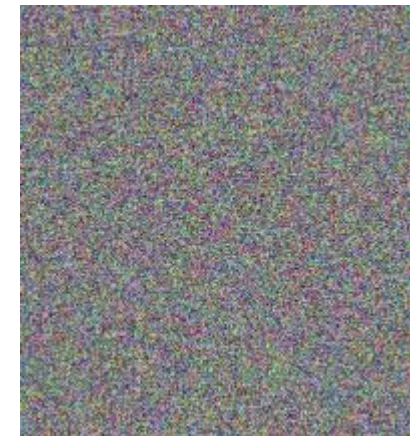
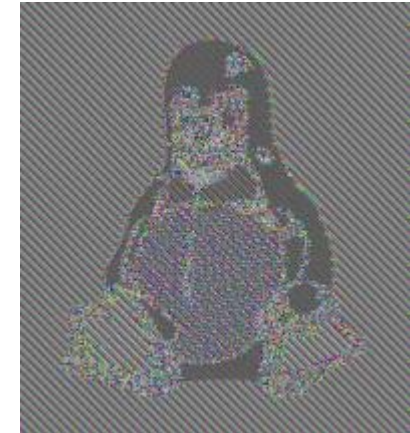
b.) Számításigény mérleg:  $(E^*) \cong 2^{128}$ ,  $(E^{**}) \cong 2 \cdot 2^{64}$

# Data Security: Secret key

## ECB



Electronic Codebook (ECB) mode encryption



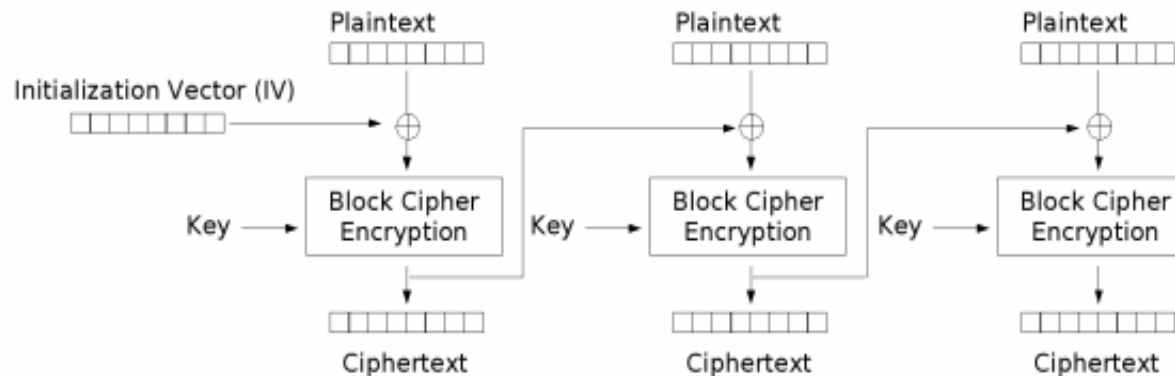
biztonság:

- a nyílt szöveg mintáit nem rejti megfelelően
- a blokkrejtjelező bemenete nem randomizált
- szótár (kódkönyv) alapú támadás lehetséges
- a rejtjeles blokkok felcserélhetők, törölhetők, helyettesíthetők



# Data Security: Secret key

## CBC



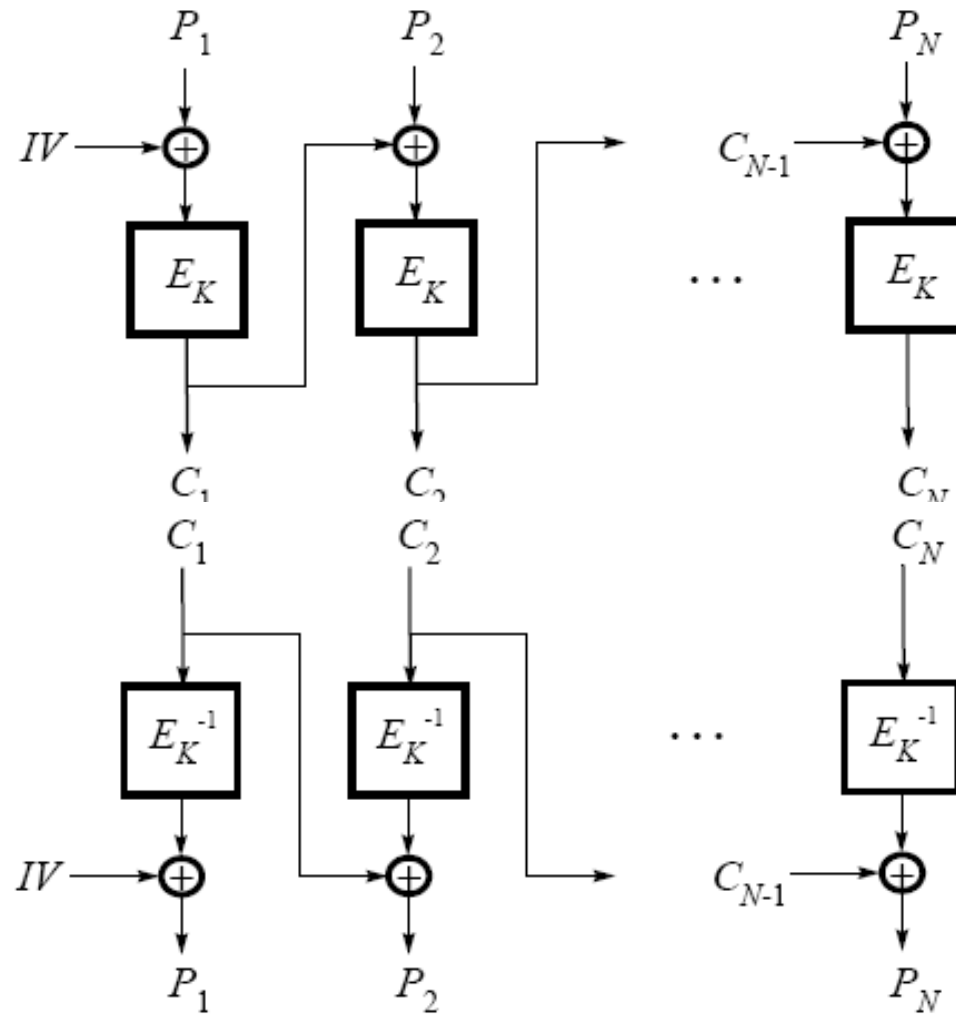
Cipher Block Chaining (CBC) mode encryption

biztonság:

- + a nyílt szöveg mintáit rejtí az előző rejtjeles blokkal való XOR-olás
- + a blokkrejtjelező bemenetét randomizálja az előző rejtjeles blokkal való XOR-olás
- + azonos nyílt szövegeket különböző IV-vel rejtjelezve különböző rejtjeles szövegeket kapunk
- + korlátozott mértékben detektálni lehet a rejtjeles blokkok felcserélését, törlését, helyettesítését
- kívág-és-beszúr támadások lehetségesek
- azonos rejtjeles blokkokhoz tartozó nyílt blokkok XOR összegét felfedi

# Data Security: Secret key

CBC



# Data Security: Secret key CBC

128 bites nyílt szöveg blokkok sorozatát AES rejtjelezővel CBC módban rejtjelezzük:  
Mennyi blokkot kell rejtjelezni ahhoz, hogy >0.5 valószínűséggel előforduljon két azonos rejtett szöveg blokk?

128 bites rejtett szöveg blokkok összes száma  $m=2^{128}$ . CBC módban a rejtett szöveg blokkokat modellezhetjük véletlenül választottaknak függetlenül a nyílt szöveg tulajdonságoktól. Így a születésnapi paradoxon alapján  $p \sim 1 - \exp(-r^2/2m)$  összefüggésből,  $p=0.5$  esetén  $1.17 \cdot 2^{64}$  eredmény adódik.

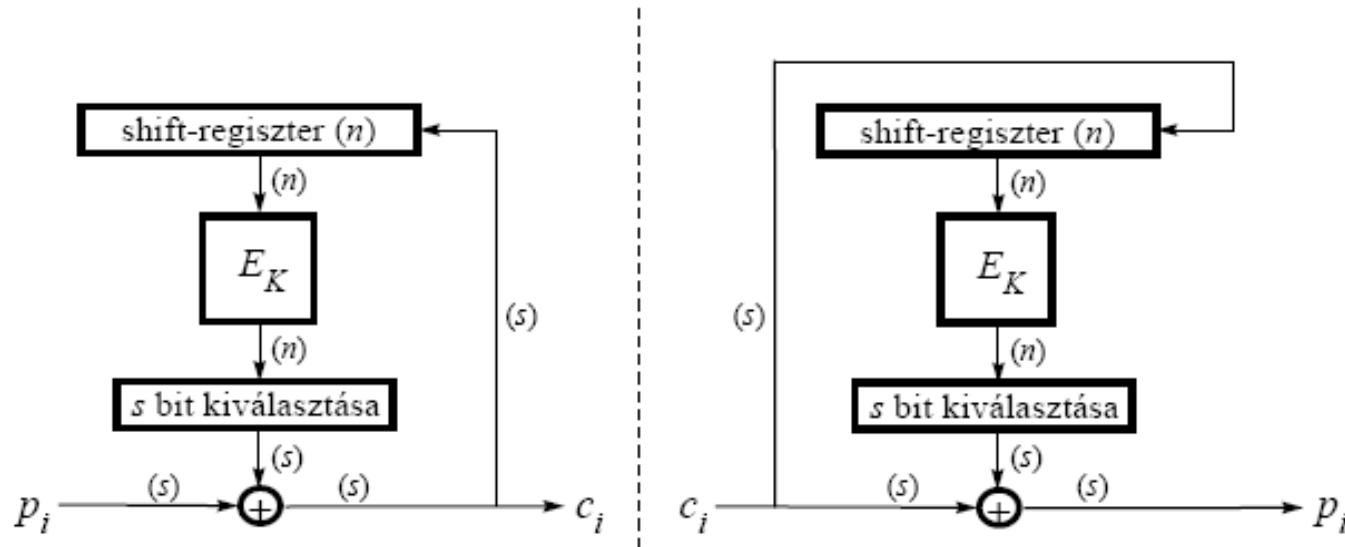
Ha két azonos rejtett szöveg blokkot detektáltunk, mit tudunk mondani a hozzájuk tartozó nyílt szöveg blokkról?

Meg tudjuk határozni a két nyílt szöveg differenciáját!

$$x_k \oplus y_{k-1} = x_i \oplus y_{i-1} \quad \rightarrow \quad x_k \oplus x_i = y_{k-1} \oplus y_{i-1}$$

# Data Security: Secret key

## CFB

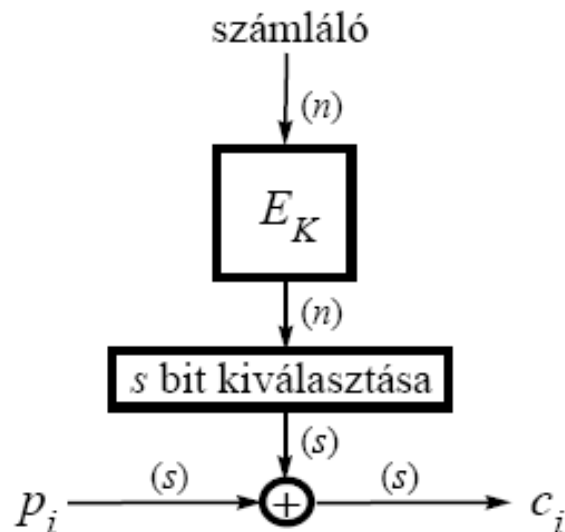
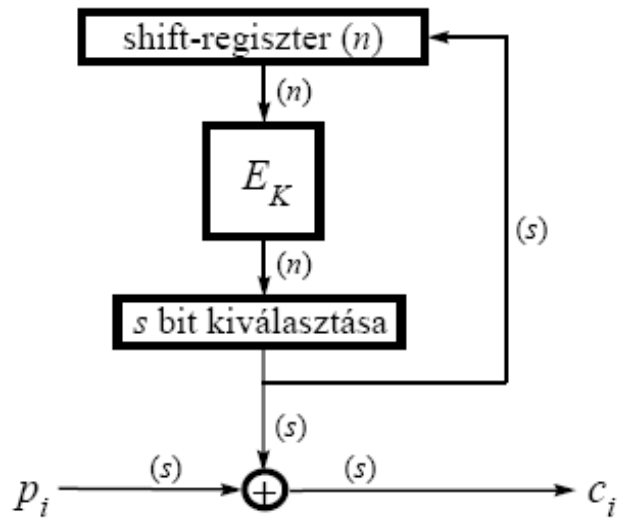


biztonság:

- + a nyílt szöveg mintáit rejti
- + a blokkrejtjelező bemente véletlen
- + azonos nyílt szövegeket különböző IV-vel rejtjelezve különböző rejtjeles szövegeket kapunk
- + korlátozott mértékben detektálni lehet a rejtjeles karakterek felcserélését, törlését, helyettesítését
- utolsó karakter bitjei manipulálhatók

# Data Security: Secret key

## OFB , CTR



biztonság:

- + a nyílt szöveg mintáit rejti
- + azonos nyílt szövegeket különböző IV-vel rejtjelezve különböző rejtjeles szövegeket kapunk
- különböző nyílt szövegeket azonos IV-vel rejtjelezve a nyílt szövegek megfejthetőek
- +/- korlátozott mértékben detektálni lehet a rejtjeles karakterek felcserélését, törlését, helyettesítését, de nem olyan mértékben, mint CFB mód esetén (a korlátozott hibaterjedés miatt)
- OFB módban  $n$ -nél kevesebb bites visszacsatolás esetén a generátor periódushossza jelentősen csökken
- + CTR módban a generátor periódushossza a számláló méretétől függ
- a visszaállított nyílt karakterek bitjei manipulálhatók

# Data Security: Secret key

## OFB

Véletlen bithibázású csatornán rejtjelezetten továbbítjuk az üzenetünket CBC blokk rejtjelező módban. A véletlen hibázás ellen hibajavító kódolást alkalmazunk. Végezzük a hibajavító kódolást a rejtjelezést megelőzően:

forrás → hibajavító kódolás → rejtjelezés,

rejtjelfejtés → hibajavító dekódolás → nyelő.

a.) Helyesen járunk-e el a fenti módon a hibák javításával kapcsolatosan?

b.) Mi a válasz a kérdésre, ha CBC mód helyett OFB módban rejtjelezünk?

a.) Nem.

A CBC mód hibaterjedés tulajdonsága szerint egy véletlen hiba esetén, hibázás utáni első blokk bitjeinek átlagosan fele hibás lesz, s még a rákövetkező blokk egy bitje. Ezt a nagymértékű meghibásodást csak igen költséges, komplex javító kóddal tudnánk eliminálni. A helyes megoldás a rejtjelezés utáni hibajavító kódolás alkalmazása.

b.) Igen.

Nincs hibaterjedés a kulcsfolyamatos típusú rejtjelezés mód miatt. Ez esetben alkalmazhatjuk a hibajavítást a rejtjelezést megelőzően.

# Data Security: Secret key

## Block cipher modes

Ha egy csatorna  $10^{-9}$  bithibaarányal működik, akkor hogyan alakul a bithibaarány rejtjelezett esetben?

- a.) 128 bites kódolás ECB rejtjelező módban
- b.) 128 bites kódolás CBC rejtjelező módban
- c.) 64 bites kódolás CFB byte alapú folyamrejtjelezésnél
- d.) 64 bites kódolás OFB byte alapú folyamrejtjelezésnél

a.)	b.)	c.)	d.)
64 E-9	65 E-9	33 E-9	E-9

# Data Security: Secret key

## Block cipher modes

Javasolható-e RSA blokk kódolás alkalmazása

1.) ECB módban?

2.) OFB módban?

1.) Igen, de csak korlátozottan.

Kulcs küldésre alkalmazható, csak véletlen, illetve nagy információtartalmú üzenet kódolható így. Nyílt szöveg alapú próbálgatás ellen nem véd.

2.) Sohasem alkalmazható.

Az OFB módban az RSA mindkét oldalon kódoló üzemmódban működne, azaz a nyilvános kulcs kellene a dekódoláshoz is.

# Data Security: Secret key

Melyiket blokk rejtjelező módot **nem** tanácsolná a következő alkalmazási feltételek esetén és miért?

- 1.) fennáll a kezdővektor (IV) átírásának veszélye
- 2.) bitkieséses szinkronhibás csatornán továbbítás
- 3.) nyílt szöveg 3 különböző értéket vehet csak fel

- 1.) CBC: IV átírással első üzenetblokk támadható
- 2.) OFB: szinkroncsúszás esetén a kulcsfolyam elcsúszik és véletlen bitfolyamot dekódolunk  
vagy CBC: egy bit elvesztése esetén a blokkhatárok az üzenet végéig elcsúsznak
- 3.) eredeti formában egyiket sem; üzenetteret randomizálással növelni kell