Tema 3: Criptografia Simètrica

Carlos Borrego

Carlos.Borrego@uab.cat

Departament d'Enginyeria de la Informació i de les Comunicacions
Universitat Autònoma de Barcelona

Criptografia i Seguretat

Material adaptat de:
Material de classe de Criptografia i Seguretat

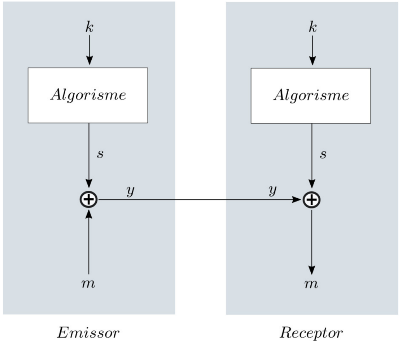
Dr. Guillermo Navarro

Universitat Autònoma de Barcelona

http://www.deic.uab.cat/

Contingut

Les xifres de flux



L'algorisme és determinista per tant la seqüència que en resulta no és completament aleatòria i a partir d'un cert moment es repeteix.

CSPRNG

Els generadors pseudoaleatoris criptogràficament segurs (CSPRNG) generen seqüències no predictibles.

En concret, per a que un PRNG sigui considerat un CSPRNG, cal que les seqüències que genera tinguin dues propietats (a partir de k bits de la seqüència generada s_{i+1}, s_{i+2}, ..., s_{i+k}):

- No existeix un algorisme en temps polinomial que pugui predir el següent bit de la seqüència, s_{i+k+1}, amb probabilitat major al 50%.
- No és computacionalment possible predir el bit anterior de la seqüència, s_i.

Tests d'aleatorietat del NIST (1)

El **test de freqüència de bits individuals** comprova que la proporció d'uns i zeros de la seqüència proporcionada és similar.

Per fer-ho, en primer lloc es transforma la seqüència binària d'entrada a una seqüència de 1 i -1:

Després, es calcula s_{obs}:

$$s_{obs} = \frac{\left| \sum_{i=1}^{n} x_i \right|}{\sqrt{n}}$$

Si la seqüència és aleatòria s_{obs} tendirà cap a 0, mentre que si hi ha massa zeros o massa uns en la seqüència, aleshores s_{obs} tendirà a ser major a zero.

Tests d'aleatorietat del NIST (2)

El **freqüència en un bloc** comprova que el número de 0/1 en un bloc de m bits sigui aproximadament m/2. Per fer-ho, es particiona la seqüència a avaluar en b = n / m blocs de m bits, descartant els bits sobrants.

$$\underbrace{\overbrace{s_1, s_2, \ldots, s_m}^{k=1}}_{m \text{ bits}} \; \underbrace{\overbrace{s_{m+1}, s_{m+2}, \ldots, s_{m+m}}^{k=2}}_{m \text{ bits}} \; \cdots \; \underbrace{\overbrace{s_{(b-1)m+1}, \ldots, s_{(b-1)m+m}}^{k=b}}_{m \text{ bits}}$$

Aleshores, per cada bloc k (amb k = 1, . . . , b), es calcula:

$$\pi_k = \frac{\sum_{j=1}^{m} s_{(k-1)m+j}}{m}$$

és a dir, es calcula la proporció d'uns que hi ha a cada bloc. Finalment, es calcula:

$$\chi^2_{obs} = 4m \sum_{k=1}^{b} (\pi_k - 1/2)^2$$

Tests d'aleatorietat del NIST (3)

El test de **ràfegues** comprova si el número de ràfegues tant d'uns com de zeros de la seqüència (N>100) s'assembla al que trobaríem en una seqüència aleatòria.
Definirem una ràfega com un conjunt de bits consecutius iguals, és a dir una ràfega de longitud k consta dels elements s_t , . . . , s_{t+k-1} , tals que
$s_t \neq s_t = s_{t+1} = ... = s_{t+k} \neq s_{t+k}$.
Per avaluar la prova de ràfegues, es calcula:

$$V_n(obs) = \left(\sum_{i=1}^{n-1} r(i) \right) + 1$$

on r(i) és la funció:

$$r(i) = \begin{cases} 0, & \text{si } s_i = s_{i+1} \\ 1, & \text{altrament} \end{cases}$$

Valors grans de V obs indiquen que les oscil·lacions de valors en la seqüència avaluada succeeixen ràpidament.

Addicionalment, aquest test té com a prerequisit que la seqüència passi el test de freqüència de bits individuals.

Contingut

Generadors congruencials

Els generadors congruencials es basen en equacions modulars recurrents del tipus:

$$x_n = (ax_{n-1} + b) \bmod m$$

Exemple: La funció *rand()* del sistema UNIX BSD utilitza el següent generador congruencial afí:

$$x_n = (1103515245x_{n-1} + 12345) \bmod 2^{31}$$

Watch:
Magic 'Nothing Up My Sleeve' Numbers - Computerphile
https://www.youtube.com/watch?v=oJWwaQm-Exs

Període

$$x_n = (ax_{n-1} + b) \bmod m$$

- Si m és primer i b = 0: període m - 1 si a és un element primitiu en m.
- Si m és una potència de 2 i b = 0: té un període com a màxim de m/4 (si a=3 o a=5 (mod 8)).
- Si b != 0: període m si i només si: mcd(m,b)=1, a-1 és divisible per tots els factors primers d'm i a-1 és divisible per 4 i m és divisible per 4.

Activitat

- Implementeu un Generador congruencial del tipus:

$$x_n = (ax_{n-1} + b) \bmod m$$

- Per un *a* un *b* donat i $m = 100$ calculeu el seu període.
- Per parelles envieu-vos missatges xifrats fent servir l'operació:

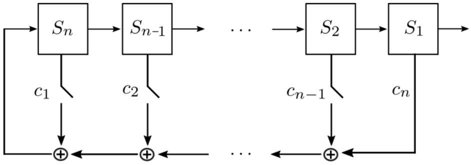
  $$C_i \;=\; M_i \;\; xor \;\; K_i$$

  sent C_i el missatge xifrat, M_i el missatge en clar i K_i els nombres obtinguts pel generador congruencial definit.
- Desxifreu els missatges xifrats fent servir:

  $$M_i \;=\; C_i \;\; xor \;\; K_i$$

LFSR, Linear Feedback Shift Register

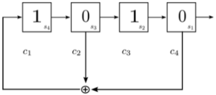
Un registre de desplaçament realimentat linealment (LFSR) de longitud n és un dispositiu físic o lògic format per n cel·les de memòria i n portes lògiques:



- Initial state: $\{S1, \ldots, S_n\}$
- Polinomi de conexions (*feedback polynomial*): $C(x) = 1 + c_1x^1 + c_2x^2 + \ldots + c_nx^n$

Exemple de l'LFSR

L'estat inicial és 1010, que correspon a l'impuls de rellotge t = 0.



El polinomi de connexions corresponent a l'LFSR:
$C(x) = 1 + 0x^1 + 1x^2 + 0x^3 + 1x^4 = 1 + x^2 + x^4$

Exemple de l'LFSR

Evolució de l'LFSR en els diferents instants de temps:

Impuls de rellotge (t)	s_4	s_3	s_2	s_1	**Sortida**
0	1	0	1	0	0
1	0	1	0	1	1
2	0	0	1	0	0
3	0	0	0	1	1
4	1	0	0	0	0
5	0	1	0	0	0
6	1	0	1	0	0
7	0	1	0	1	1
⋮		⋮			⋮

L'impuls de rellotge t=6 tornem a tenir l'estat inicial i, per tant, a partir d'aquí la seqüència es torna a repetir (període 6).

Polinomi de connexions

- **Factoritzable**:
  - seqüència depèn de l'estat inicial.
  - període sempre $< 2^n - 1$.
- **Irreductible (però no primitiu)**:
  - seqüència depèn de l'estat inicial (període de mida fixa).
  - període és un divisor de $2^n - 1$.
- **Primitiu**:
  - seqüència no depèn de l'estat inicial.
  - període de $2^n - 1$.

Polinomi primitiu

Per un LFSR de mida *n*

- Sempre grau *n*
- El nombre de polinomis primitius de grau *n* és:

$$\frac{\phi(2^n - 1)}{n}$$

Primitive Polynomial List:

`www.partow.net/programming/polynomials/index.html`

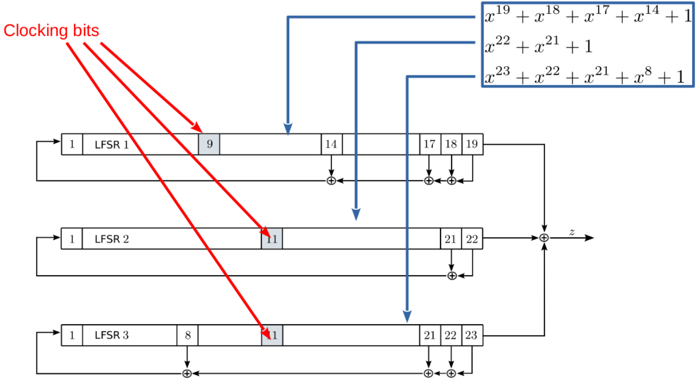
Contingut

A5

A5 - funcionament

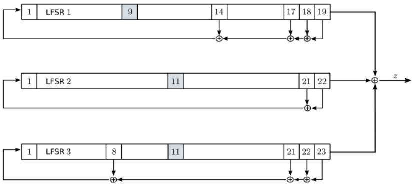
- Inicialització
- Generació seqüència de 228 bits
- Xifrar 228 bits

A5



Clocking bits
$x^{19} + x^{18} + x^{17} + x^{14} + 1$
$x^{22} + x^{21} + 1$
$x^{23} + x^{22} + x^{21} + x^{8} + 1$
1
LFSR 1
9
14
17
18
19
1
LFSR 2
11
21
22
z
1
LFSR 3
8
11
21
22
23

A5




```
LFSR1: 1011 1000 1101 1000 010
LFSR2: 1011 0110 1111 0100 0010 01
LFSR3: 1110 1111 1001 1100 1000 001
```

Time *t*:

- Out: 0 ⊕ 1 ⊕ 1 = **1**

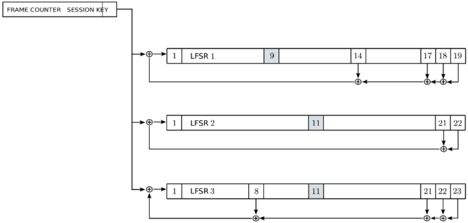
```
LFSR1: 1101 1100 0110 1100 001
LFSR2: 1101 1011 0111 1010 0001 00
LFSR3: 1110 1111 1001 1100 1000 001
```

Time *t + 1* :

- Shift: LFSR1, LFSR2
- Out: 1 ⊕ 0 ⊕ 1 = **0**

A5

A5 Iniciatlització



- Omplit tots els estats amb 0
- 64 rotacions
  - NO clocking
  - XOR session key
- 22 rotacions
  - NO clocking
  - XOR frame counter
- 100 rotacions
  - With clocking

- Session key 64 bits (secret)
- Frame counter 22 bits (public)

4

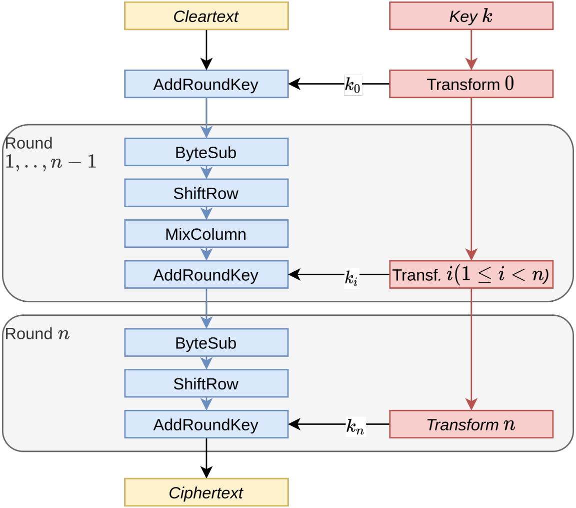
Contingut

Data representation

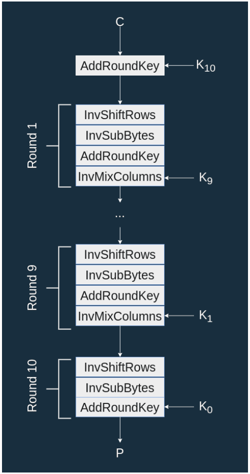
19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

- Cleartext block: $m = m_1 m_2 \ldots m_{128}$
- Grouped in 16 bytes (8 bits per byte) in a matrix representation

AES encryption process



AES decryption process



Sobre notació

- Si p és primer, $GF(p)$ es el cos finit $(\mathbb{Z}_p, +, \cdot)$
- $GF(p^m)$ es un cos amb elements polinomis de grau màxim $m - 1$ i coeficients a $GF(2)$. Notació equivalent: $(\mathbb{Z}_p[x]/m(x), +, \cdot)$ on el grau de $m(x)$ és m.

AES

En general AES opera en el grup $GF(2^8)$ o $(\mathbb{Z}_p[x]/m(x), \oplus, \otimes)$ on $m(x) = x^8 + x^4 + x^3 + x + 1$: polinoms de grau màxim 7 i coeficients binaris (a $\mathbb{Z}_2$ o $GF(2)$).

- E.g. $a(x) = a_7x^7 + \cdots + a_1x + a_0$, $a_i \in \mathbb{Z}_2$

1 byte $->$ un polinomi de $GF(2^8)$. P.e.

hexadecimal	$\rightarrow$	binary	$\rightarrow$	polinomi
63	$\rightarrow$	`0110 0011`	$\rightarrow$	$x^6 + x^5 + x + 1$

Amb les operacions:

- Suma "$\oplus$": suma polinomis a $\mathbb{Z}_2$ (o bitwise XOR)
- Multiplicació "$\otimes$": multiplicació de polinimis mòdul $x^8 + x^4 + x^3 + x + 1$

Operacions

- Suma $\oplus$: bitwise XOR. E.g

$$57 \oplus 83 = D4$$
$$01010111 \oplus 10000011 = 11010100$$
$$(x^6 + x^4 + x^2 + x + 1) \oplus (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2$$

- Producte $\otimes$: producte polinoms $\pmod{x^8 + x^4 + x^3 + x + 1}$

$$57 \otimes 83 = C1$$
$$01010111 \otimes 10000011 = 11000001$$

$$(x^6 + x^4 + x^2 + x + 1) \otimes (x^7 + x + 1) \pmod{x^8 + x^4 + x^3 + x + 1}$$
$$= x^{13} + x^{11} + x^9 + x^8 + x^7 + x^7 + x^5 + x^3 + x^2 + x + x^6 + x^4 + x^2 + x + 1 \pmod{x^8 + x^4 + x^3 + x + 1}$$
$$= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \pmod{x^8 + x^4 + x^3 + x + 1}$$
$$= x^7 + x^6 + 1$$

Representació de bloc de 128 bits a AES

Binari:

```
00000001 00000010 00000011 00000100
00000101 00000110 00000111 00001000
00001001 00001010 00001011 00001100
00001101 00001110 00001111 00010000
```

Hexadecimal: $\begin{pmatrix} 01 & 02 & 03 & 04 \\ 05 & 06 & 07 & 08 \\ 09 & 0a & 0b & 0c \\ 0d & 0e & 0f & 10 \end{pmatrix}$

Polinomial:

$$\begin{pmatrix} (1) & (x) & (x+1) & (x^2) \\ (x^2+1) & (x^2+x) & (x^2+x+1) & (x^3) \\ (x^3+1) & (x^3+x) & (x^3+x+1) & (x^3+x^2) \\ (x^3+x^2+1) & (x^3+x^2+x) & (x^3+x^2+x+1) & (x^4) \end{pmatrix}$$

MixColumns

$C = MixColumns(B)$

$$\begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{pmatrix} \rightarrow \text{MixColumns} \rightarrow \begin{pmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{pmatrix}$$

- Cada columna de B: vector que es multiplica per una matriu 4×4 constant.

Multiplicació

$$\left.\begin{array}{l}
\begin{pmatrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{pmatrix}\cdot\begin{pmatrix}b_{00}\\ b_{10}\\ b_{20}\\ b_{30}\end{pmatrix}=\begin{pmatrix}c_{00}\\ c_{10}\\ c_{20}\\ c_{30}\end{pmatrix}\\
\begin{pmatrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{pmatrix}\cdot\begin{pmatrix}b_{01}\\ b_{11}\\ b_{21}\\ b_{31}\end{pmatrix}=\begin{pmatrix}c_{01}\\ c_{11}\\ c_{21}\\ c_{31}\end{pmatrix}\\
\begin{pmatrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{pmatrix}\cdot\begin{pmatrix}b_{02}\\ b_{12}\\ b_{22}\\ b_{32}\end{pmatrix}=\begin{pmatrix}c_{02}\\ c_{12}\\ c_{22}\\ c_{32}\end{pmatrix}\\
\begin{pmatrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{pmatrix}\cdot\begin{pmatrix}b_{03}\\ b_{13}\\ b_{23}\\ b_{33}\end{pmatrix}=\begin{pmatrix}c_{03}\\ c_{13}\\ c_{23}\\ c_{33}\end{pmatrix}
\end{array}\right\}\longrightarrow\begin{pmatrix}c_{00} & c_{01} & c_{02} & c_{03}\\ c_{10} & c_{11} & c_{12} & c_{13}\\ c_{20} & c_{21} & c_{22} & c_{23}\\ c_{30} & c_{31} & c_{32} & c_{33}\end{pmatrix}$$

Exemple

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} b_{00} \\ b_{10} \\ b_{20} \\ b_{30} \end{pmatrix} = \begin{pmatrix} c_{00} \\ c_{10} \\ c_{20} \\ c_{30} \end{pmatrix}$$

$$(02 \otimes b_{00}) \oplus (03 \otimes b_{10}) \oplus (01 \otimes b_{20}) \oplus (01 \otimes b_{30}) = c_{00}$$

Recordeu, cada element és un polinomi!

Exemple

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} b_{00} \\ b_{10} \\ b_{20} \\ b_{30} \end{pmatrix} = \begin{pmatrix} c_{00} \\ c_{10} \\ c_{20} \\ c_{30} \end{pmatrix}$$

$$(02 \otimes b_{00}) \oplus (03 \otimes b_{10}) \oplus (01 \otimes b_{20}) \oplus (01 \otimes b_{30}) = c_{00}$$

Recordeu, cada element és un polinomi!

Exemple

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} b_{00} \\ b_{10} \\ b_{20} \\ b_{30} \end{pmatrix} = \begin{pmatrix} c_{00} \\ c_{10} \\ c_{20} \\ c_{30} \end{pmatrix}$$

$$(02 \otimes b_{00}) \oplus (03 \otimes b_{10}) \oplus (01 \otimes b_{20}) \oplus (01 \otimes b_{30}) = c_{00}$$

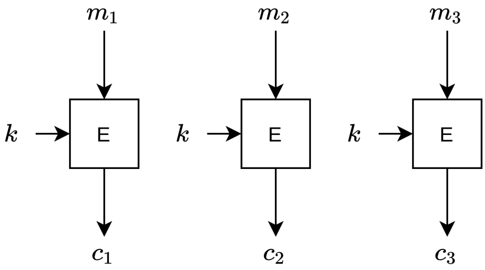
Recordeu, cada element és un polinomi!

Contingut

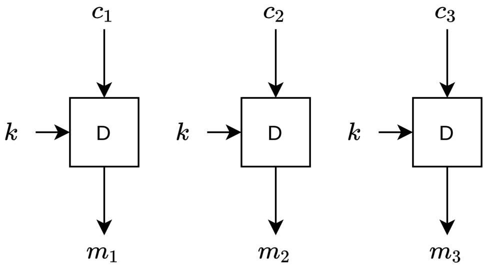
Notation

$MSB_n(x)$	n Most Significant Bits of x
$LSB_n(x)$	n Least Significant Bits of x
$x \mid y$	x concatenated with y
$E_k(x)$	encryption of x with key k
$D_k(x)$	decryption of x with key k
IV	Initialization Vector: generalment no cal que sigui secret, sí impredictible i únic per cada xifrat.

ECB,Electronic Code Book



$c_i = E_k(m_i)$



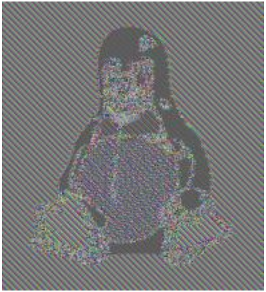
$m_i = D_k(c_i)$

ECB

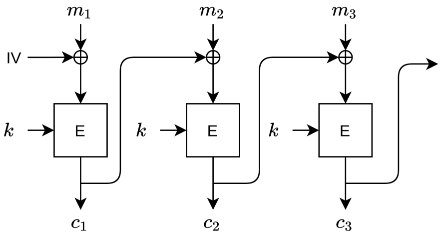
- Blocs m_i iguals $\Rightarrow$ blocs c_i iguals
  - – No amaga patrons de dades
- Blocs es xifren independentment
  - + paral·lelització
  - + accés aleatori
  - + errors en c_i no es propaguen
  - – no es detecten reordenacions, insercions, eliminacions

ECB exemple

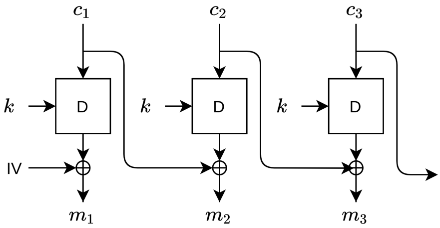
ECB exemple



CBC, Cipher Block Chaining



$$c_0 = IV,\ c_i = E_k(m_i \oplus c_{i-1})$$

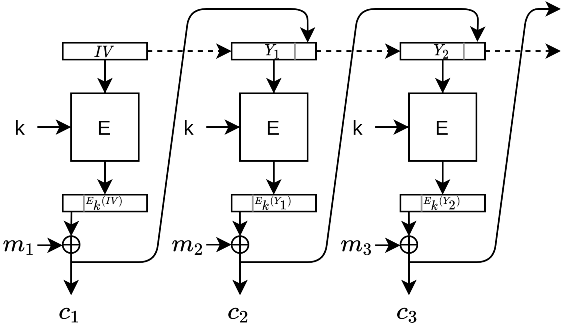


$$c_0 = IV,\ m_i = D_k(c_i) \oplus c_{i-1}$$

CBC

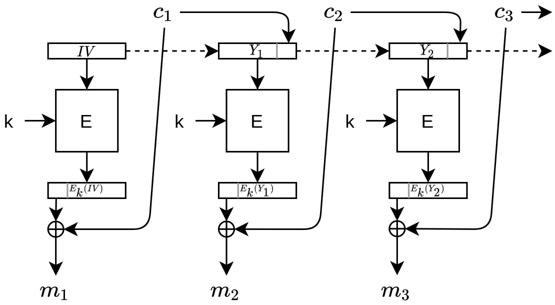
- $+$ oculta patrons de dades (mateix $m_i \not\Rightarrow$ mateix c_i)
- $-$ Xifrat no paral·lelitzable, però desxifrat sí.
- $\cdot$ error en c_i afecta a blocs més endavant (m_i, m_{i+1}).

CFB, Cipher Feedback



$$Y_0 = IV$$
$$Y_i = LSB_{b-n}(Y_{i-1}) \mid c_i$$
$$c_i = m_i \oplus MSB_n(E_k(Y_{i-1}))$$

CFB, Cipher Feedback

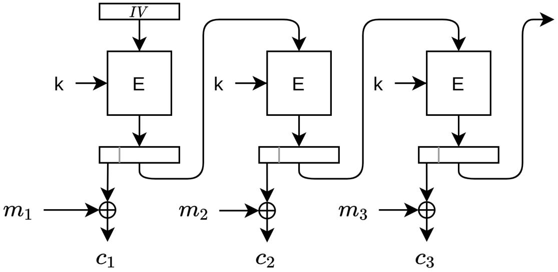


$$Y_0 = IV$$
$$Y_i = LSB_{b-n}(Y_{i-1}) \mid c_i$$
$$m_i = c_i \oplus MSB_n(E_k(Y_{i-1}))$$

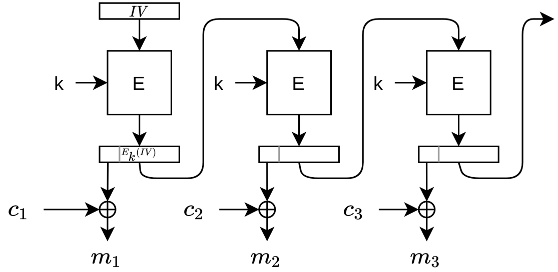
CFB

- Mida de bloc de text en clar $n \leq$ mida de bloc de xifrat b.
- Si $n = b \Rightarrow$ CBC.
- Es pot fer servir per convertir un criptosistema de bloc a un de flux (*CFB-1* o *1-bit CFB*).
- Xifrat no paral·lelitzable, però desxifrat sí.
- Error en c_i afecta a blocs més endavant.

OFB, Output Feedback



$$I_0 = IV$$
$$I_i = E_k(I_{i-1})$$
$$c_i = m_i \oplus MSB_n(I_i)$$

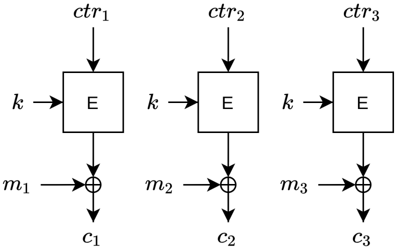


$$I_0 = IV$$
$$I_i = E_k(I_{i-1})$$
$$m_i = c_i \oplus MSB_n(I_i)$$

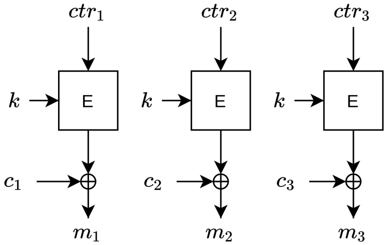
OFB

- Permet construir un xifrat en flux a partir d'un en bloc
- El keystream es genera de forma independent al missatge (cleartext o ciphertext).
- Errors en c_i no es propaguen (errors en IV afecten tot el xifrat/desxifrat).
- No paral·lelitzable (però es pot pre-generar el keystream).

CTR, Counter



$$ctr_i = IV \mid i$$
$$c_i = m_i \oplus E_k(ctr_i)$$



$$ctr_i = IV \mid i$$
$$m_i = c_i \oplus E_k(ctr_i)$$

CTR

- $\Rightarrow$ Xifrat de flux
- Paral·lelitzable (no requereix cap tipus de feedback)
- Errors en c_i no es propaguen.

Tema 3: Criptografia Simètrica

Carlos Borrego

Carlos.Borrego@uab.cat

Departament d'Enginyeria de la Informació i de les Comunicacions
Universitat Autònoma de Barcelona

Criptografia i Seguretat