

Chiffrements symétriques

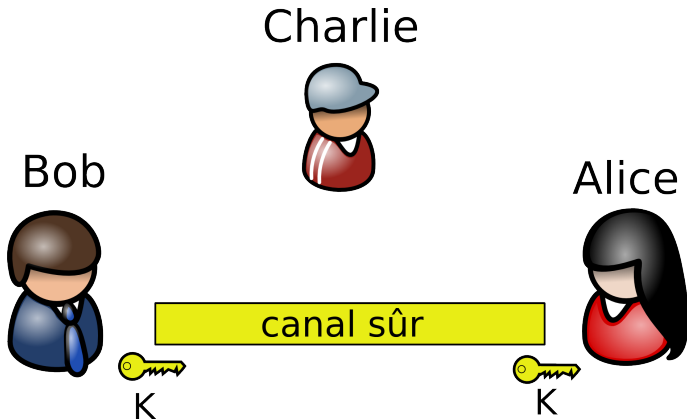
Laurent Fousse

22 septembre 2008

Outline

- 1 Systèmes de chiffrements symétriques modernes
 - Chiffrements par flot
 - Chiffrements par blocs
 - Modes opératoires

Principe de fonctionnement



Plan

- 1 Systèmes de chiffrements symétriques modernes
 - Chiffrements par flot
 - Chiffrements par blocs
 - Modes opératoires

Chiffrements par flots

- Un chiffrement par flot considère le message comme un **flux de caractères** (usuellement des bits ou des octets), par opposition au chiffrement par blocs.
- L'opération de chiffrement effectuée sur chaque caractère varie au cours du chiffrement : un état est nécessaire.

Les chiffrements par blocs chiffrent le message vu comme une suite de blocs avec une transformation fixe.

Chiffrement par flot synchrone

Définition

Le flux chiffrant est généré indépendamment du texte clair et du texte chiffré.

$$\sigma_{i+1} = f(\sigma_i, k)$$

$$z_i = g(\sigma_i, k)$$

$$c_i = h(z_i, m_i)$$

σ_i : état; f : fonction d'évolution; g : fonction de génération de flux chiffrant

z_i : flux chiffrant

m_i : texte clair; c_i : texte chiffré; h : fonction de sortie

Chiffrement par flot synchrone

- L'émetteur et le récepteur doivent être synchronisés (même clef, même état).
- Pas de propagation d'erreur.
- Des attaques actives sont possibles.

Chiffrement par flot additif binaire

Définition

Un chiffrement par flot additif binaire est un chiffrement par flot où les caractères sont des bits et h est la fonction de OU-exclusif (XOR).

Chiffrement par flot auto-synchronisant

Le flux chiffrant est généré comme fonction de la clef et d'un nombre fixé des chiffres précédents:

$$\sigma_j = (c_{j-t}, c_{j-t+1}, \dots, c_{j-1})$$

$$z_j = g(\sigma_j, k)$$

$$c_j = h(z_j, m_j)$$

Chiffrement par flot auto-synchronisant

- Auto-synchronisation: si des chiffres sont modifiés ou effacés, le déchiffrement reprendra correctement après un nombre borné d'erreurs (t au plus).
- Propriété de diffusion (chaque élément du texte clair influence tout le flux chiffré suivant).

Linear Feedback Shift Registers (LFSR)

- Un algorithme pour produire un flux de bits.
- Utilisé comme un générateur de nombre aléatoire dans le cadre des chiffrements par flot additifs.

Exemple d'utilisation: E0

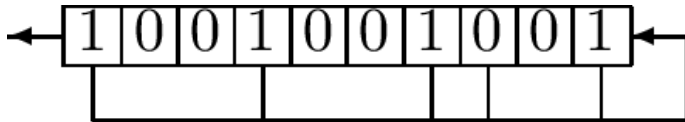
E0 est un chiffrement par flot utilisé pour les communications Bluetooth.

- Chiffrement par flot binaire additif
- 4 LFSR de tailles 25, 31, 33 and 39 bits (total: 128).
- 4 bits d'état interne.

Linear Feedback Shift Registers (Fibonacci setup)

- L registres numérotés $(0, 1, \dots, L - 1)$
- à chaque clock d'horloge, le contenu du registre 0 est copié dans la séquence de sortie.
- le contenu du registre i est déplacé dans le registre $i - 1$ pour $1 \leq i \leq L - 1$.
- le nouveau contenu du registre $L - 1$ est le bit de rétroaction, calculé comme l'addition (mod 2) du contenu précédent d'un certain ensemble fixé de registres.

Linear Feedback Shift Registers (Fibonacci setup)



$(s_0, s_1, \dots, s_{L-1})$ état initial

$$s_j = (c_1 s_{j-1} + c_2 s_{j-2} + \dots + c_L s_{j-L})$$

$$C(D) = 1 + c_1 D + c_2 D^2 + \dots + c_L D^L$$

LFSR: Exercices

- 1 Montrer que la sortie d'un LFSR est ultimement périodique.
- 2 Borner la période de la séquence de sortie d'un LFSR en fonction de son nombre de registres.
- 3 Calculer les séquences binaires possibles pour un LFSR dont le polynôme de rétroaction est $P(X) = X^4 + X^2 + X + 1$. Quelles sont leurs périodes ?
- 4 Mêmes questions avec $P(X) = X^4 + X^3 + X^2 + X + 1$ et $P(X) = X^4 + X^3 + 1$.

Linear Feedback Shift Registers

Theorem

La séquence de sortie d'un LFSR à L registres est périodique avec une période inférieure à $2^L - 1$ (et ce pour tout état initial).

Theorem

Tout séquence binaire périodique peut être générée par un LFSR.

Primitive polynomial

Definition

A polynomial $P(X)$ of degree $L > 0$ defined in \mathbb{F}_2 is primitive if it is irreducible and

$$\min \{ i > 0 \mid X^i = 1 \pmod{P} \} = 2^L - 1$$

Primitive polynomial: Exercises

Show that an irreducible polynomial over \mathbb{F}_2 :

- 1 always a non-zero constant coefficient,
- 2 always an odd number of non-zero terms,
- 3 has at least one monomial of odd degree.

Are the converse true?

Primitive polynomial: Exercises

Over \mathbb{F}_2 :

- 1 Is $X^2 + X + 1$ primitive?
- 2 Show that $X^6 + X^3 + 1$ is not primitive.
- 3 Show that $X^5 + X^3 + X^2 + X + 1$ is primitive. What about the other degree-5 polynomials?

Maximum period LFSR

Theorem

A L -stages LFSR has period $2^L - 1$ (the maximum possible) iff its connecting polynomial is primitive and the initial state is non-zero.

Attaques contre des LFSR

- À partir d'un certain nombre de bits du flot chiffrant, l'algorithme de Berlekamp-Massey retrouve le polynôme minimal $f(X)$ d'un LFSR qui le génère.
- En supposant que le polynôme f a un degré plus petit que L , il est nécessaire de connaître $2L$ bits de la séquence pour trouver f .

L'algorithme de Berlekamp-Massey a une complexité $\mathcal{O}(L \log L)$. Il est possible d'utiliser un algorithme plus simple dont la complexité est $\mathcal{O}(L^3)$.

Linear relations

Rappel:

$$s_j = (c_1 s_{j-1} + c_2 s_{j-2} + \dots + c_L s_{j-L})$$

Supposons $s_0, s_1, \dots, s_{2L-1}$ connus. On peut écrire le système suivant:

$$\begin{aligned} s_L &= (c_1 s_{L-1} + c_2 s_{L-2} + \dots + c_L s_0) \\ s_{L+1} &= (c_1 s_L + c_2 s_{L-1} + \dots + c_L s_1) \\ s_{L+2} &= (c_1 s_{L+1} + c_2 s_L + \dots + c_L s_2) \\ &\vdots \\ s_{2L-1} &= (c_1 s_{2L-2} + c_2 s_{2L-1} + \dots + c_L s_{L-1}) \end{aligned}$$

L équations indépendantes avec L inconnues, donc on peut utiliser la méthode de Gauss.

Plan

- 1 Systèmes de chiffrements symétriques modernes
 - Chiffrements par flot
 - Chiffrements par blocs
 - Modes opératoires

Description des chiffrements par blocs

- Ils traitent le message comme une **séquence de blocs** (généralement de 64 bits à 256 bits).
- Chaque bloc est traité séparément.

Description des chiffrements par blocs

Definition (Chiffrement par blocs)

Un chiffrement par blocs est une fonction

$E : \{0, 1\}^n \times K \rightarrow \{0, 1\}^n$ telle que pour chaque clef $k \in K$, la fonction $m \mapsto E(m, k)$ est une bijection de $\{0, 1\}^n$ dans lui-même, noté E_k . On écrit $D_k = E_k^{-1}$ la fonction de déchiffrement correspondante.

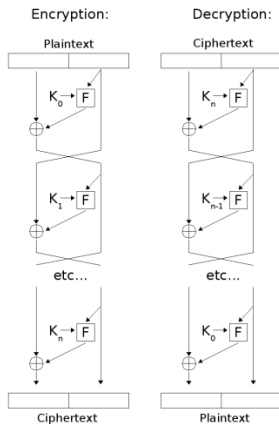
Exemples de chiffrements par blocs

Chiffrement	Taille de blocs	Taille de clef	Année	Inventeur
DES	64	56	1975	Coppersmith, Feistel, ...
IDEA	64	128	1991	Massey, Lai
Blowfish	64	0 to 448	1993	Schneier
AES	128	{ 128, 192, 256 }	1998	Daemen, Rijmen
Serpent	128	{ 128, 192, 256 }	1998	Anderson, Biham, Knudsen

Description de DES

- Approuvé en tant que standard de chiffrement aux États-Unis en 1976, après deux appels à candidature (en 1973 et 1974).
- «Amélioré» par la NSA en cours de processus (au niveau de la conception des S-boxes).
- Utilise des blocs de taille $n = 64$ bits.
- Utilise une clef de taille 56 bits.
- Basé sur le principe d'un réseau de Feistel.

Description des réseaux de Feistel



Feistel Cipher

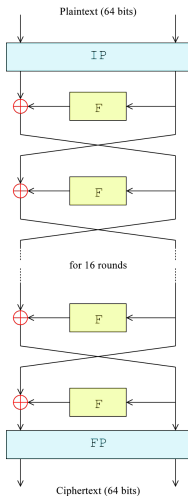
Description des réseaux de Feistel

- La taille de bloc doit être paire, on divise le bloc en deux:
 $m = (L_0, R_0)$.
- r tours.
- Chaque tour transforme $(L_{i-1}, R_{i-1}) \rightarrow_{K_i} (L_i, R_i)$ en

$$\begin{aligned}L_i &= R_{i-1} \\R_i &= L_{i-1} \oplus f(R_{i-1}, K_i)\end{aligned}$$

- La sortie est (R_r, L_r) .

Description de DES (schéma général)



Description de DES (détails)

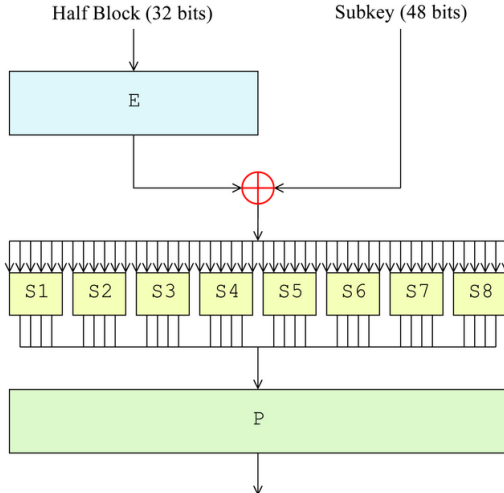
- Une permutation initiale (IP) sur les 64 bits du bloc.
- La permutation inverse est appliquée à la fin du chiffrement.
- 16 clefs de ronde K_1, \dots, K_{16} de 48 bits chacune sont déduites des 56 bits de la clef principale.
- Pour chaque ronde du réseau de Feistel, on utilise la fonction

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$$

où

- E est une expansion fixée de 32 bits vers 48 bits,
- S est composé de 8 applications fixées de 6 bits vers 4 bits, appelées S-boxes.
- P est une permutation fixée sur 32 bits.

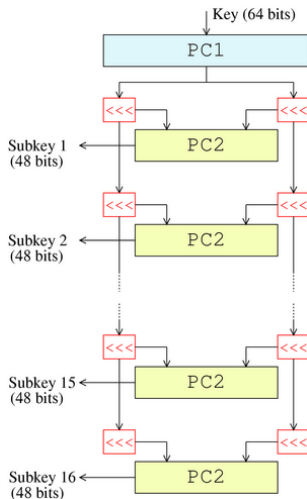
Description de DES (fonction f)



S-Boxes: S1

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

Description de DES (génération des sous-clefs)



Propriétés des chiffrements par blocs

Certaines propriétés attendues des chiffrements par blocs sont:

- Chaque bit du chiffré devrait dépendre de tous les bits de la clef et de tous les bits du texte clair.
- Modifier un bit du clair ou de la clef devrait modifier tous les bits du chiffré avec une probabilité $\frac{1}{2}$.
- Changer un bit du chiffré devrait produire un changement non prédictible dans le clair correspondant.

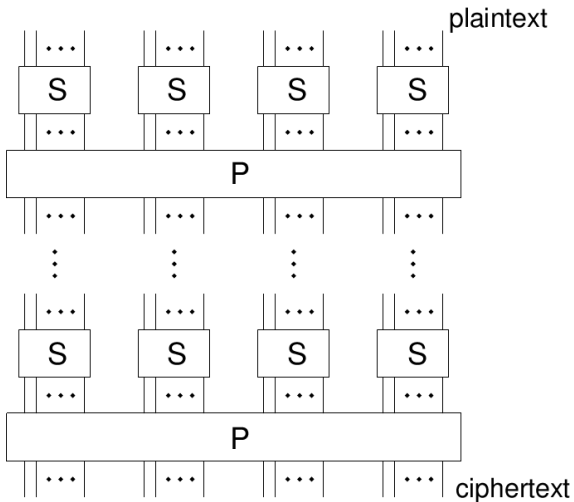
Anomalies de DES

- Il existe 4 *clefs faibles*: $E_k(E_k(x)) = x$.
- Il existe 6 paires de *clefs semi-faibles*: $E_{k_1}(E_{k_2}(x)) = x$.

AES

- Successeur de DES.
- Approuvé comme standard de chiffrement en 2002 suite à un appel à candidature datant de 1997.
- Basé sur le principe de réseau de substitution/permutation (SPN).

Description d'un SPN



Description de AES

- Une ronde initiale composée de `AddRoundKey`.
- Plusieurs rondes (10, 12 ou 14 suivant la taille de la clef) composées de:
 - 1 `SubBytes` — une substitution non-linéaire où chaque octet est remplacé par un autre d'après une table.
 - 2 `ShiftRows` — une étape de permutation où chaque ligne de l'état subit une rotation cyclique.
 - 3 `MixColumns` — une opération de mélange sur les colonnes.
 - 4 `AddRoundKey` — chaque octet de l'état est combiné avec la clef de ronde.
- Dernière ronde (sans `MixColumns`):
 - 1 `SubBytes`
 - 2 `ShiftRows`
 - 3 `AddRoundKey`

Description de AES

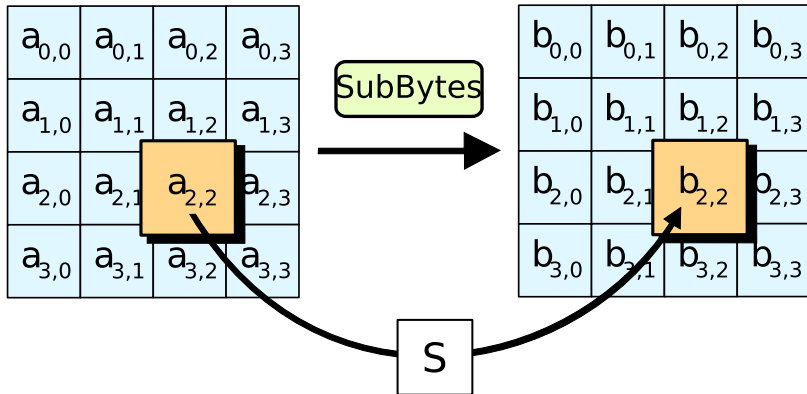
- L'état est un carré 4x4 de 16 octets.
- Chaque octet est vu comme un élément du corps $GF(2^8) = \mathbb{F}_2[X]/m(X)$ où $m(X) = X^8 + X^4 + X^3 + X + 1$.
- Exemple: l'octet 01100011 représente $X^6 + X^5 + X + 1$.

Calculs dans $GF(2^8)$

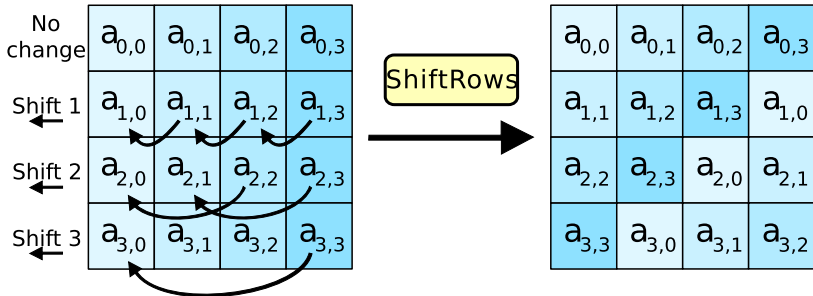
Exemples tirés du standard:

$$\begin{aligned}\{57\}\{83\} &= \{c1\} \\ \{57\}_{16} &= 5 \cdot 16 + 7 \\ &= 87 = \{01010111\}_2 \\ &= X^6 + X^4 + X^2 + X + 1 \\ \{83\}_{16} &= 8 \cdot 16 + 3 \\ &= 131 = \{10000011\}_2 \\ &= X^7 + X + 1\end{aligned}$$

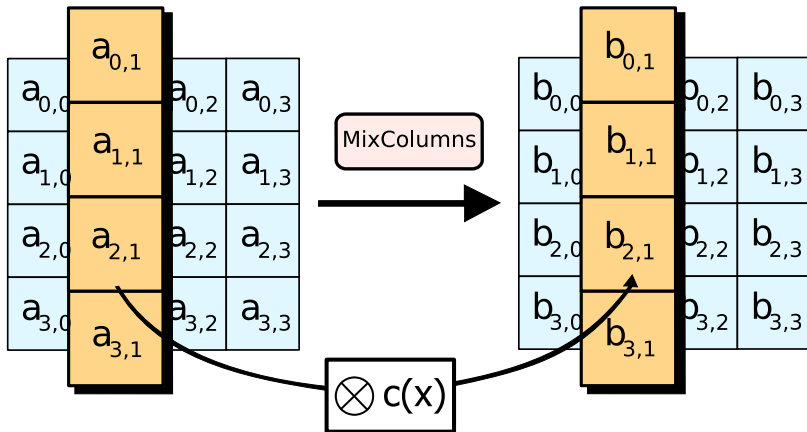
Description de AES



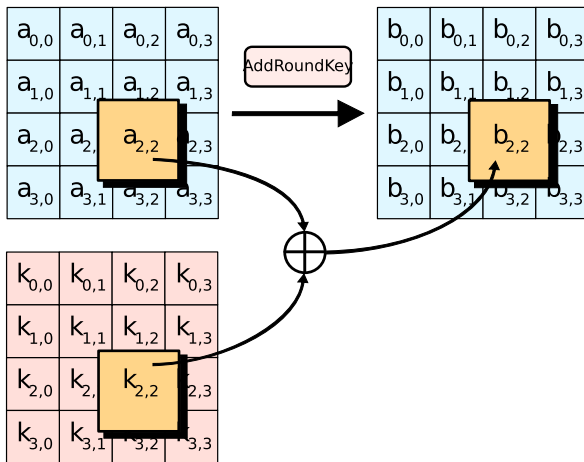
Description de AES



Description de AES



Description de AES



Description de AES (étape SubBytes)

La transformation `SubBytes` s'applique séparément sur chaque octet du carré:

- Prendre l'inverse multiplicatif $\text{mod } m$.
- Appliquer une transformation affine:

$$b'_i = b_i \oplus b_{i+4} \oplus b_{i+5} \oplus b_{i+6} \oplus b_{i+7} + c_i$$

où $c = \{01100011\}_2$.

Description de AES (étape SubBytes)

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Description de AES (étape `SubBytes`)

1 Est-ce que `SubBytes` est inversible?

Description de AES (étape MixColumns)

- Chaque colonne est vue comme un polynôme sur $GF(2^8)$.

$$a_i(t) = a_{0,i} + a_{1,i}t + a_{2,i}t^2 + a_{3,i}t^3$$

- On multiplie le polynôme par $c(t) = \{03\}t^3 + \{01\}t^2 + \{01\}t + \{02\}$ et on le réduit mod $t^4 + 1$.
- Le polynôme obtenu est ré-interprété comme une colonne dans le nouvel état.
- On applique cette opération pour toutes les colonnes.

Description de AES (étape `MixColumns`)

- Est-ce que `MixColumns` est inversible?

Remarques sur les chiffrements par blocs

- Une étape de substitution ajoute de la *confusion* au chiffrement. La confusion sert à rendre la relation entre la clef et le chiffré aussi complexe que possible.
Quelles opérations participent à la *confusion* dans DES et AES?
- Une étape de transposition (permutation) dans une ronde ajoute de la *diffusion* au chiffrement. Le but est d'étaler les propriétés de redondance dans le clair sur tout le chiffré.
Distinguer les opérations participant à la *diffusion* dans DES et AES.

Plan

- 1 Systèmes de chiffrements symétriques modernes
 - Chiffrements par flot
 - Chiffrements par blocs
 - Modes opératoires

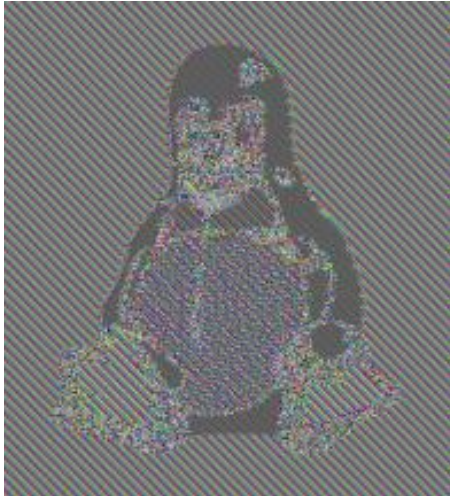
Modes opératoires pour les chiffrements par blocs

- Un chiffrement par blocs est une transformation d'un seul bloc de clair en un bloc de chiffré dépendante de la clef.
- Il est possible d'utiliser un chiffrement par blocs de différentes manières pour chiffrer un message constitué de plusieurs blocs m_0, m_1, \dots, m_l . On appelle cela un *mode opératoire*.
- Quelques modes opératoires:
 - ECB (Electronic codebook)
 - CBC (Cipher block chaining)
 - CFB (Cipher feedback)
 - OFB (Output feedback)
 - Counter mode.

Exemple de ECB (Texte clair)



Exemple de ECB (chiffré avec ECB)

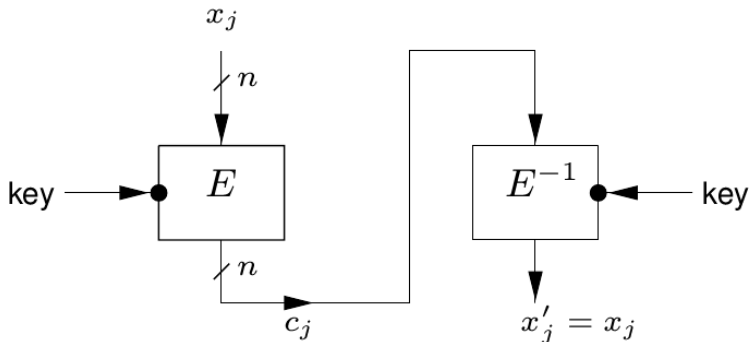


Exemple de ECB (chiffré avec les autres modes)



Electronic codebook

Electronic Codebook (ECB)

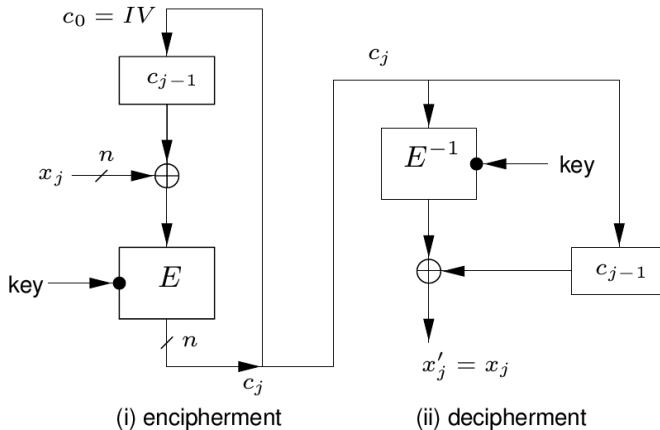


(i) encipherment

(ii) decipherment

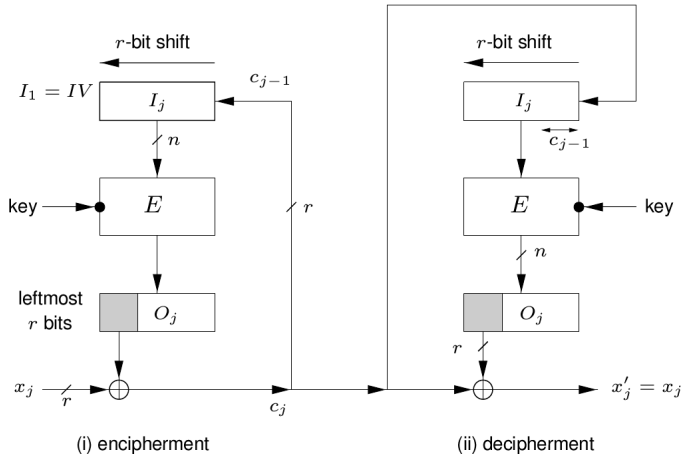
Cipher Block Chaining

Cipher-block Chaining (CBC)



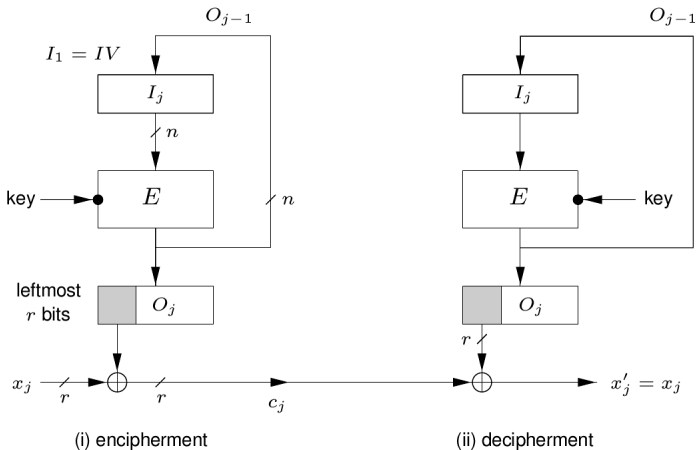
Cipher feedback

Cipher feedback (CFB), r -bit characters/ r -bit feedback



Output feedback

Output feedback (OFB), r -bit characters/ n -bit feedback



Propriétés des modes opératoires

ECB:

- Le même bloc est toujours chiffré en le même bloc.
- Les blocs sont chiffrés de façon indépendante: une permutation dans les blocs chiffrés correspond à la même permutation dans les blocs en clair.
- Pas de propagation d'erreur.

Propriétés des modes opératoires

CBC:

- Le chiffrement d'un bloc dépend du chiffrement de tous les blocs précédents.
- Il y a propagation d'erreur: une erreur dans un bloc chiffré affecte deux blocs déchiffrés.
- Les erreurs sont rattrapées après deux blocs faux.

Exercices (chiffrements par blocs et modes opératoires)

CBC on the english alphabet

On the alphabet $A = \{0, 1, \dots, 25\}$ we introduce the ciphering function $e : (k, x) \mapsto k \times x \pmod{26}$. A block is a letter in this case.

- 1 On which condition on k is e a proper cipher function?
What the space of possible keys?
- 2 Decrypt the following cryptogram, ciphered with key $k = 19$ and unknown IV.

FDDXL XTBQS VAWNH BXVWG JAYEH BSCCJ NKTWV
YMYOT BSPXL KG

Exercices (chiffrements par blocs et modes opératoires)

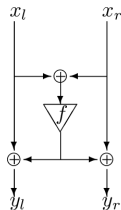
- 1 Montrer comment distinguer un réseau de Feistel à deux rondes d'une fonction quelconque: $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$.
- 2 Montrer des faiblesses dans un réseau de Feistel dont la fonction f est:

$$f(x) = B(A(x) \oplus k)$$

où k est la clef, A et B sont des fonctions linéaires.

Exercices (chiffrements par blocs)

Considérons une ronde du chiffrement par blocs IDEA:



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

$$\begin{cases} y_l = x_l \oplus f(x_l \oplus x_r) \\ y_r = x_r \oplus f(x_l \oplus x_r) \end{cases}$$

- 1 Montrer que cette ronde est inversible pour toute fonction f et trouver son inverse.
- 2 Cette ronde est équivalente à trois rondes d'un schéma de Feistel avec des fonctions f différentes et bien choisies pour chaque ronde. Trouver ce réseau de Feistel équivalent.
- 3 Comment peut-on distinguer cette fonction de chiffrement d'une fonction aléatoire?