Cours: Sécurité Informatique

2022-2023

Chapitre 02 : Initiation à la cryptographie

02 - Principes des crypto-systèmes symétriques : Algorithmes DES, AES

02 - Principes des crypto-systèmes symétriques : Algorithmes DES, AES

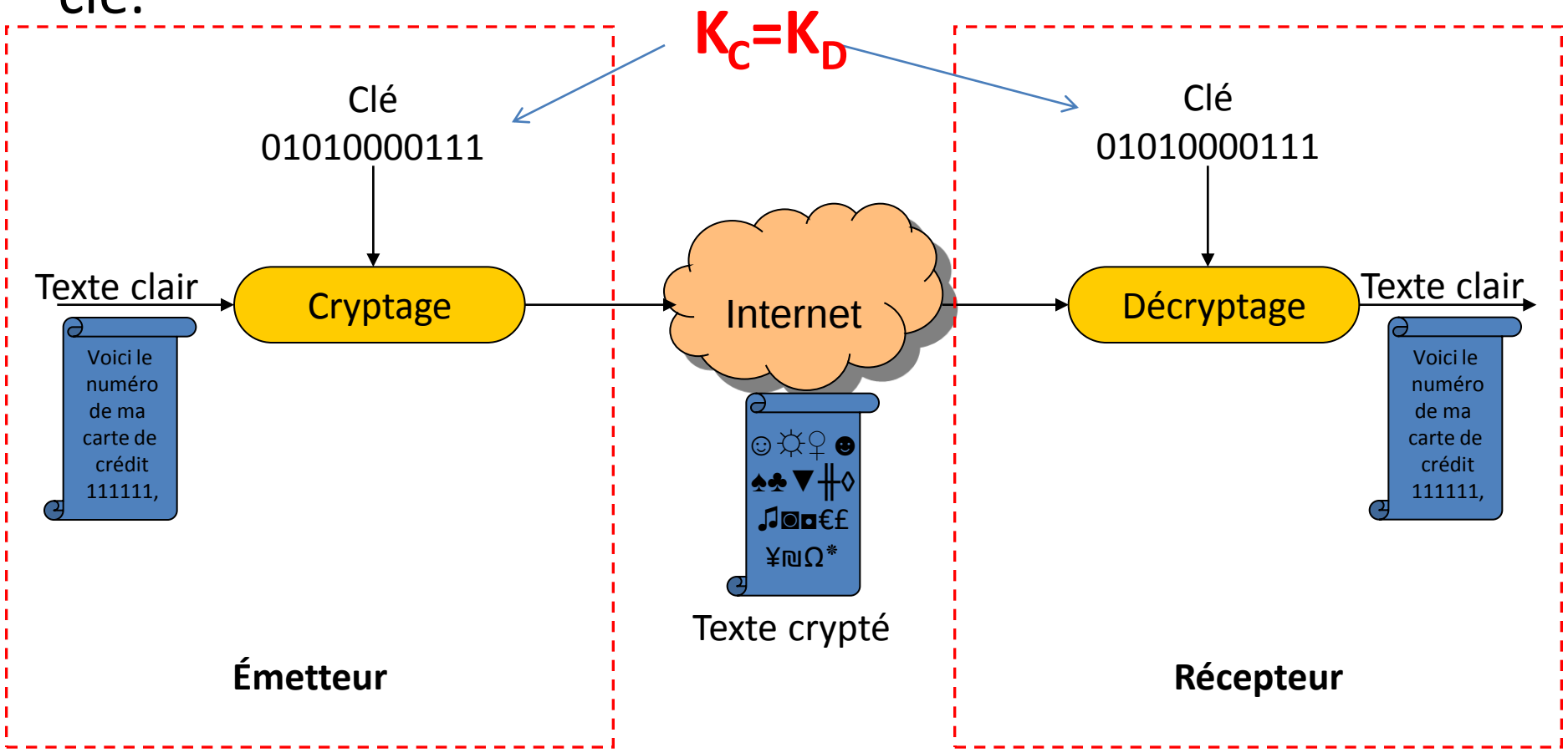
Introduction

- Les systèmes de cryptage à **clé privée**, appelé aussi système de cryptage **symétrique**, sont utilisés depuis déjà plusieurs siècles. C'est l'approche la plus authentique du chiffrement de donnée et mathématiquement la moins problématique.

- Autres termes anglais utilisés : Single-key, one-key, private-key, conventional encryption.

Introduction

- L'échange de la clé doit se faire sur un canal sécurisé.
- La sécurité repose totalement sur la confidentialité de la clé.



Introduction

- Il y a deux catégories de systèmes à clé privée: les chiffrements **par blocs** et les chiffrements **par flux.**
- Les deux catégories **différentes** selon la **manière de traiter les données** en claire à crypter, soit par blocs soit bit par bit comme un flux de données.
- Chacune des deux approches possède ces **avantages** et ces **inconvénients**, et ces **circonstances** d'applications selon le besoin.
- Plusieurs algorithmes symétriques ont été proposés dans les deux catégories selon des principes différents.

Terminologie

Les S-Boxes

- Les S-boxes (*substitution-box*), composantes des systèmes cryptographiques, sont des **tables de substitution** (ou boîte de substitution, fonction de substitution).
- Une table de substitution prend en général une variable de *m* bits en entrée et produit une sortie de *n* bits, les entrées et les sorties **n'ont pas forcément la même taille**.
- Elles peuvent avoir **plus d'entrées** que de **sorties**, ou plus de **sorties** que **d'entrées**. Les S-Boxes permettent de casser la **linéarité** de la structure de chiffrement (principe de confusion).
- Les tables sont souvent définies à l'avance, mais il arrive parfois qu'elles soient générées par l'algorithme (par exemple dans **Blowfish**).

Terminologie

Les S-Boxes

Exemple :

- Voici une S-Box (S_5) tirée de l'algorithme **DES**. La sortie de 4 bits est obtenue à partir de l'entrée de 6 bits.
- On **divise** ces 6 bits en **deux parties** : les deux bits aux **extrémités** et les quatre bits restants (**au centre**).
- Les deux bits indiquent la **ligne** et les bits centraux donnent la **colonne** correspondante.

S_5		4 bits au centre de l'entrée															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Bits externes	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	**0000**	1110	1001
	01	**1110**	**1011**	**0010**	**1100**	**0100**	**0111**	**1101**	**0001**	**0101**	**0000**	**1111**	**1100**	**0011**	**1001**	**1000**	**0110**
	10	0100	0010	0001	1011	1100	1101	0111	1000	1111	1001	1100	0101	0110	**0011**	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1100	**0100**	0101	0011

Terminologie

Les S-Boxes

- Par exemple, avec une entrée "011011", on divise en "**0** 1101 **1**". Ce qui donne pour la ligne "01" et pour la colonne "1101". La sortie de la table est alors "1001".
- Les valeurs présentes dans les S-Box doivent être choisies de manière à éviter les **attaques**.
- Dans le cas de **DES**, il a été prouvé que les tables avaient été conçues de manière à résister à la **cryptanalyse différentielle** (technique qui ne sera publiée que bien des années plus tard).

Terminologie

Les P-Boxes

- **P-Box** (*permutation-box*) est une **table de permutation** employée dans les algorithmes de chiffrement. Elle indique comment **échanger** les éléments d'une structure.
- Une P-Box contribue à la « **diffusion** » en mélangeant les données et en améliorant **l'effet avalanche**.
- Une P-Box peut se présenter sous plusieurs formes, il s'agit en général d'un **tableau** à une dimension comme [1,8,5,3,4,6,7,2].
- Ce tableau signifie que le premier élément reste en place, que la deuxième sortie prend la valeur de la huitième entrée, que la troisième sortie prend la valeur de la cinquième, etc.

Terminologie

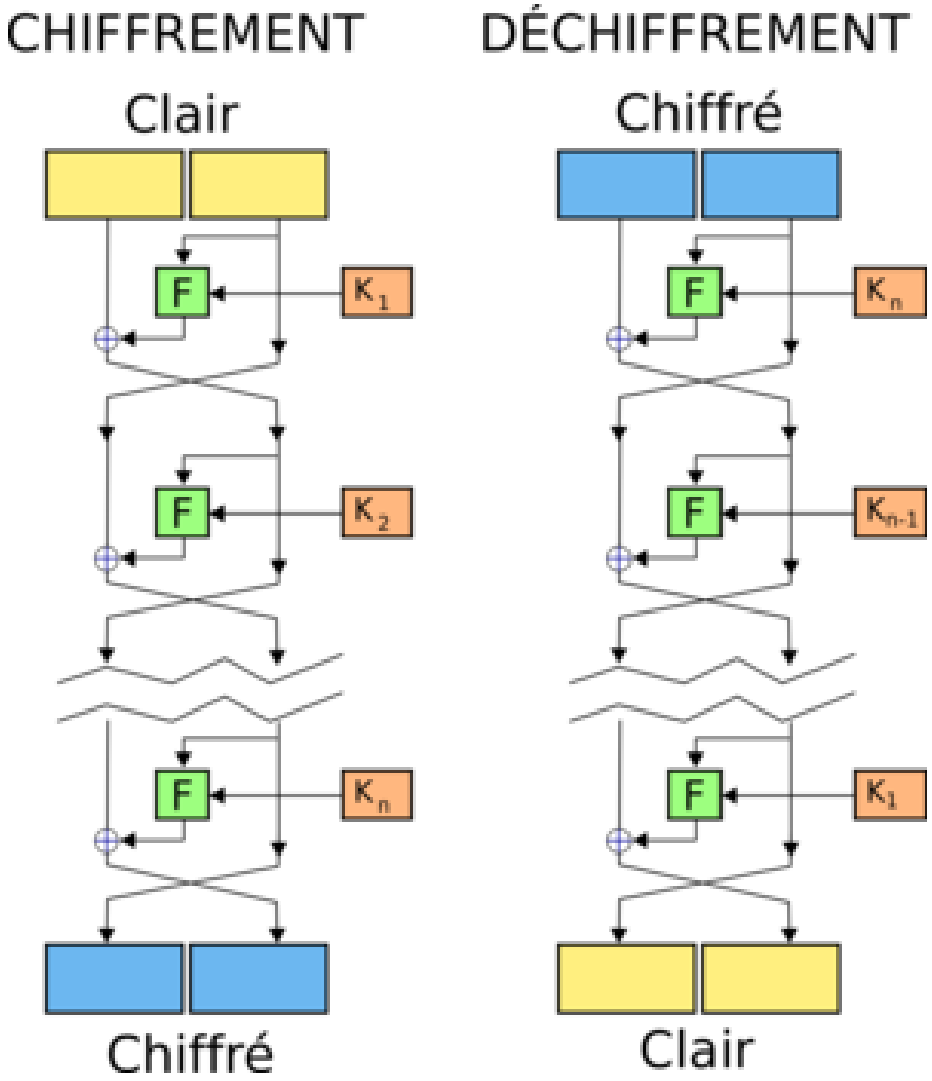
Réseaux de Feistel

- Un **réseau de Feistel** est une construction utilisée dans les algorithmes de **chiffrement par bloc**, nommée d'après le cryptologue d'IBM, **Horst Feistel**. Elle a été utilisée pour la première fois dans **Lucifer** et **DES**.
- Un réseau de Feistel repose sur des principes simples, dont des **permutations**, des **substitutions**, des **échanges** de blocs de données et une **fonction** prenant en entrée une clé intermédiaire à chaque étage.
- Cette structure offre plusieurs avantages, le chiffrement et le déchiffrement ont une architecture **similaire, voire identique,** dans certains cas. L'implémentation **matérielle** est aussi **plus facile** avec un tel système.

Terminologie

Réseaux de Feistel

- Un réseau de Feistel est subdivisé en plusieurs *tours* ou étages.
- Le réseau traite les données en deux parties de taille identique.
- À chaque tour, les deux blocs sont échangés puis un des blocs est combiné avec une version transformée de l'autre bloc.



Terminologie

Réseaux de Feistel

- Un tour opère comme suite :

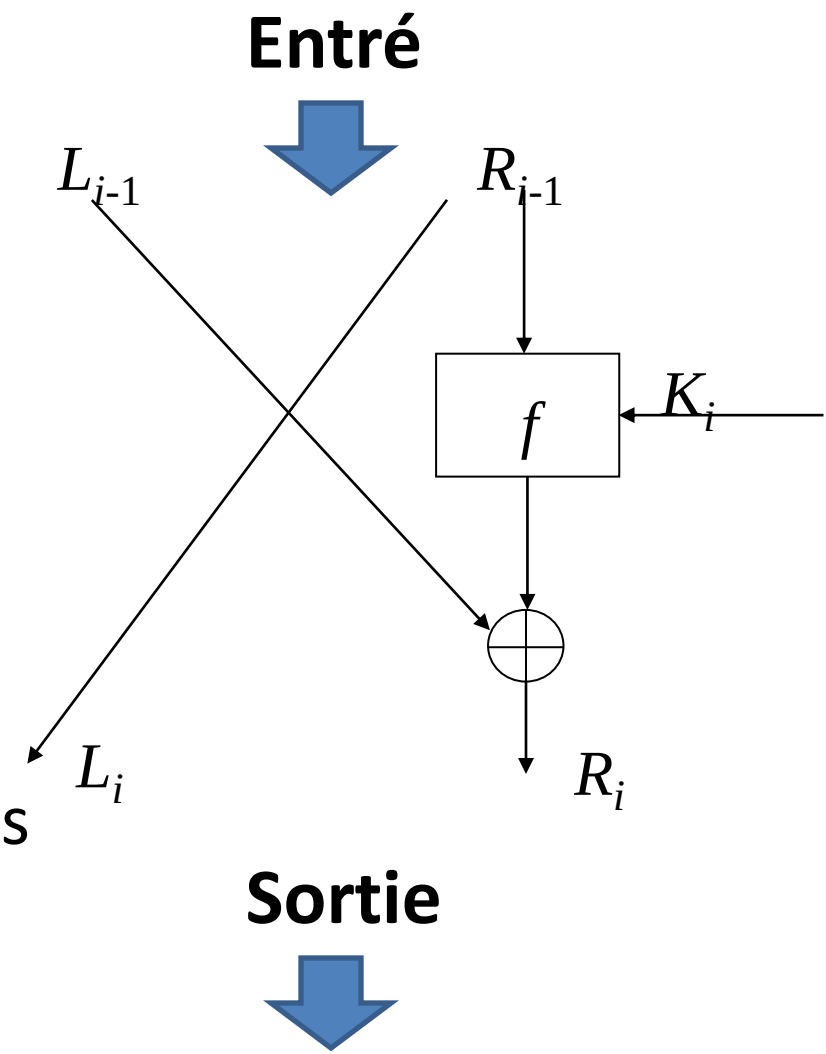
$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \otimes f(R_{i-1}, K_i)$$

- Le cryptage se fait selon la séquence:

$L_0R_0 \rightarrow L_1R_1 \rightarrow \ldots\ldots\ldots\ldots\ldots \rightarrow L_nR_n$

- Pour le Décryptage on inverse

$L_nR_n \rightarrow R_0L_0$,et on applique les clés dans l'ordre inverse.



Terminologie

Réseaux de Feistel

- La taille de chaque Bloc L_i et R_i est généralement de 32 bits dans le standard (32+32=64 bits)
- La taille de chaque clé K_i et de 48 bits, les clés K_i sont **extraites** de la clé **initiale** K de taille 64 bits.
- Un grand nombre d'algorithmes utilise des réseaux de Feistel, avec des variantes. Voici une liste non exhaustive :
  - DES
  - Blowfish
  - Twofish
  - SEED
  - RC5
  - OAEP

Chiffrement par Blocs

- Dans ce type de chiffrement, il y a une **séparation** du texte clair en blocs d'une **longueur fixe**, et un algorithme chiffre un bloc **à la fois**.
- La taille des blocs a un impact sur la **sécurité** et sur la **complexité** : les blocs de grandes dimensions sont plus sécuritaires, mais sont plus lourds à implémenter.
- Les chiffrements par blocs sont aussi utilisés dans des systèmes à clé publique.

Chiffrement par Blocs

Exemples d'algorithmes :

- Dans ce qui suit, on expose les principaux algorithmes de chiffrements par blocs reconnus comme standards. Certains de ces algorithmes ne sont plus utilisés (DES), mais certains d'autres sont toujours considérés comme cryptographiquement sûre :

    1) DES
    2) Triple-DES
    3) IDEA
    4) BlowFish
    5) AES

Chiffrement par Blocs : Le DES

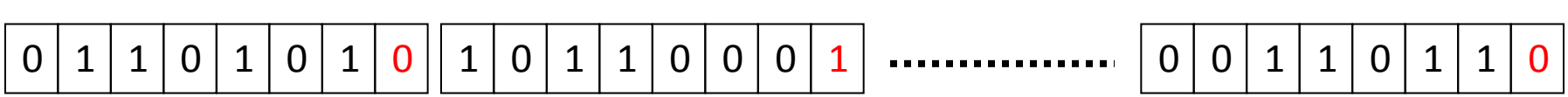
Principe du DES

- DES est un algorithme de chiffrement par bloc.
- L'information à chiffrer est découpée en blocs de tailles identiques (sur 64 bits).
- Les blocs sont ensuite chiffrés les uns après les autres en utilisant la même clé.
- La clé de DES et sur 64 bit, et ainsi la taille des blocs de données et des blocs codés.
- Lors du déchiffrement, les blocs sont rassemblés de nouveau en utilisant le principe inverse avec la même clé.

Chiffrement par Blocs : Le DES

Principe du DES

- En vérité **seulement 56 bits** sont utilisés. Un bit de chaque octet est utilisé pour le **contrôle de la parité** de la clé :



- Le principe de fonctionnement de DES est basé sur le mécanisme des Réseaux de Feistel.
- La version standard du DES utilise un réseau de Feistel de 16 rounds pour le chiffrement et le déchiffrement.

Chiffrement par Blocs : Le DES

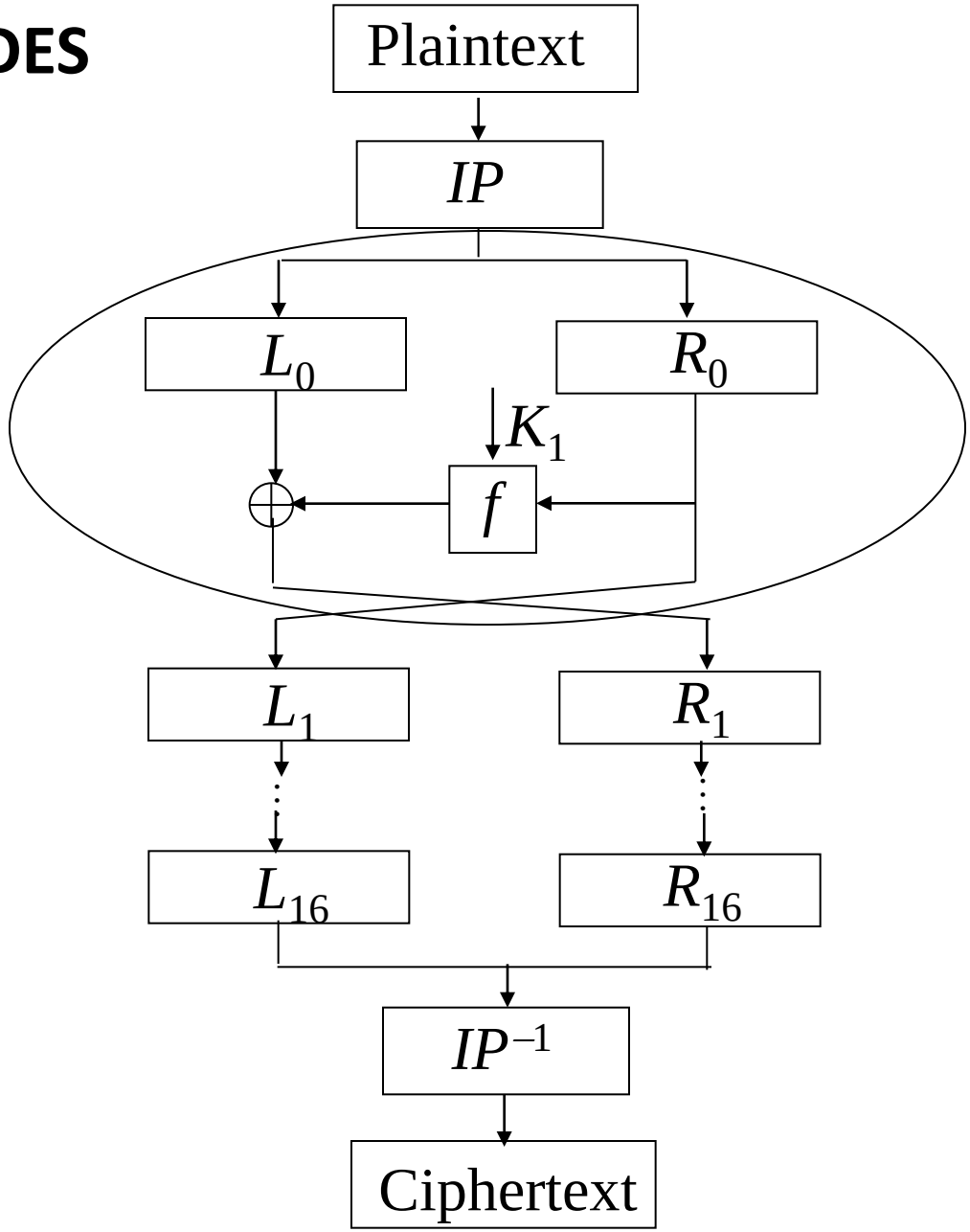
Principe du DES

- L'algorithme DES agit en trois étapes :

1) Les 64 bits *m* en entrée sont permutés selon une permutation fixe pour obtenir m_0=IP*(m),* m_0 est ensuite divisé en deux parties L et R de taille 32 bits chacun tel que $m_0=L_0R_0$
2) Pour i=1..16 faire : $L_i=R_{i-1}$ et $R_i= L_{i-1}\otimes f(R_{i-1},K_i)$
3) Inverser le résultat final $m_{16}=L_{16}R_{16}$ pour obtenir $R_{16}L_{16}$, ensuite la permutation initiale est appliquée dans le sens inverse pour obtenir $C=IP^{-1}(R_{16}L_{16})$

Chiffrement par Blocs : Le DES

Principe du DES



Chiffrement par Blocs : Le DES

La permutation initiale : IP

(58 50 42 34 26 18 10 2 60 52 44 36 28 20 12 4 62
54 46 38 30 22 14 6 64 56 48 40 32 24 16 8 57 49
41 33 25 17 9 1 59 51 43 35 27 19 11 3 61 53 45
37 29 21 13 5 63 55 47 39 31 23 15 7)

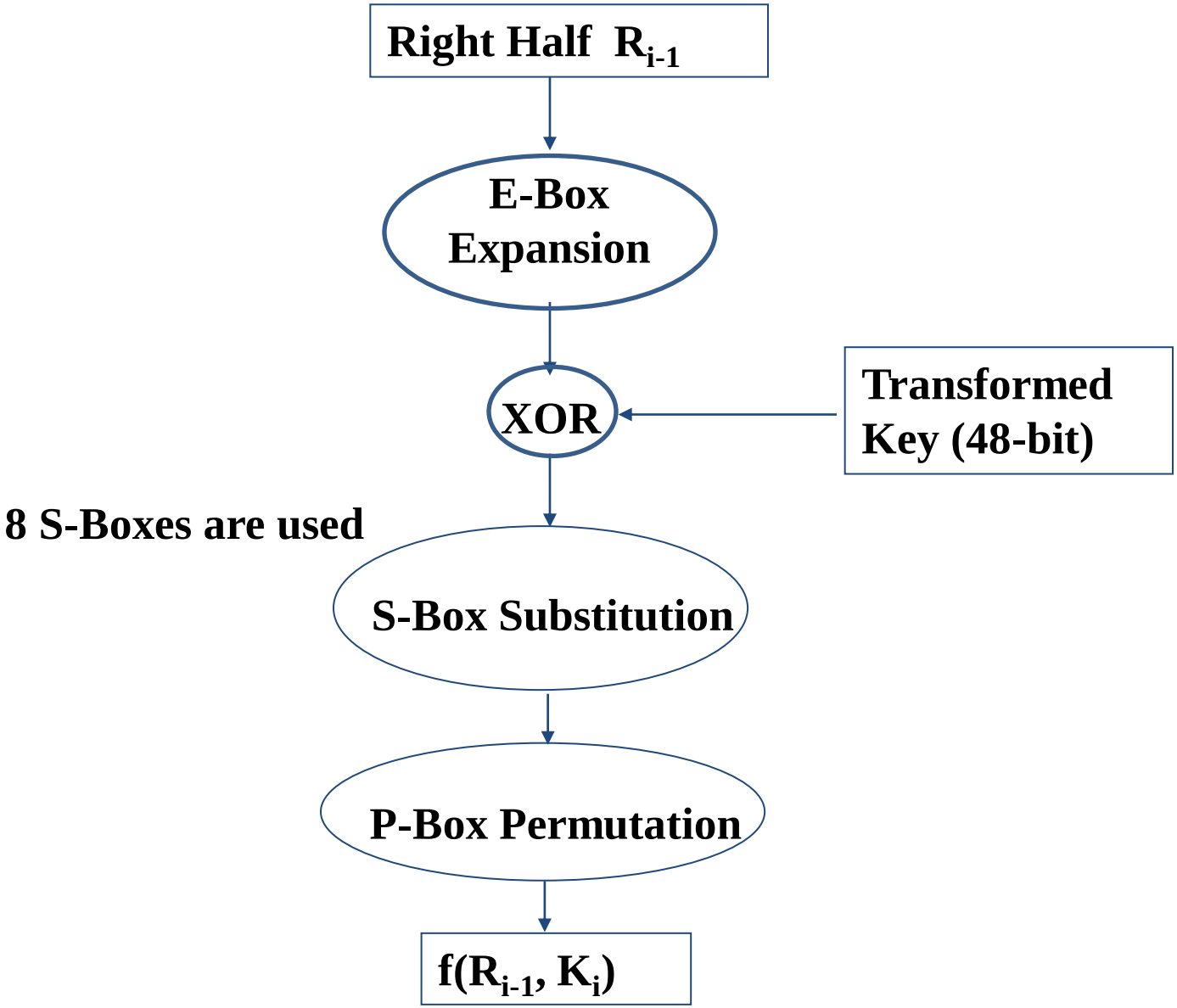
- C'est une **permutation fixe**, choisie au début par la NSA (pour des raisons de conception électronique)

La permutation inverse correspondante : IP^{-1}

(40 8 48 16 56 24 64 32 39 7 47 15 55 23 63 31 38
6 46 14 54 22 62 30 37 5 45 13 53 21 61 29 36
4 44 12 52 20 60 28 35 3 43 11 51 19 59 27 34
2 42 10 50 18 58 26 33 1 41 9 49 17 57 25)

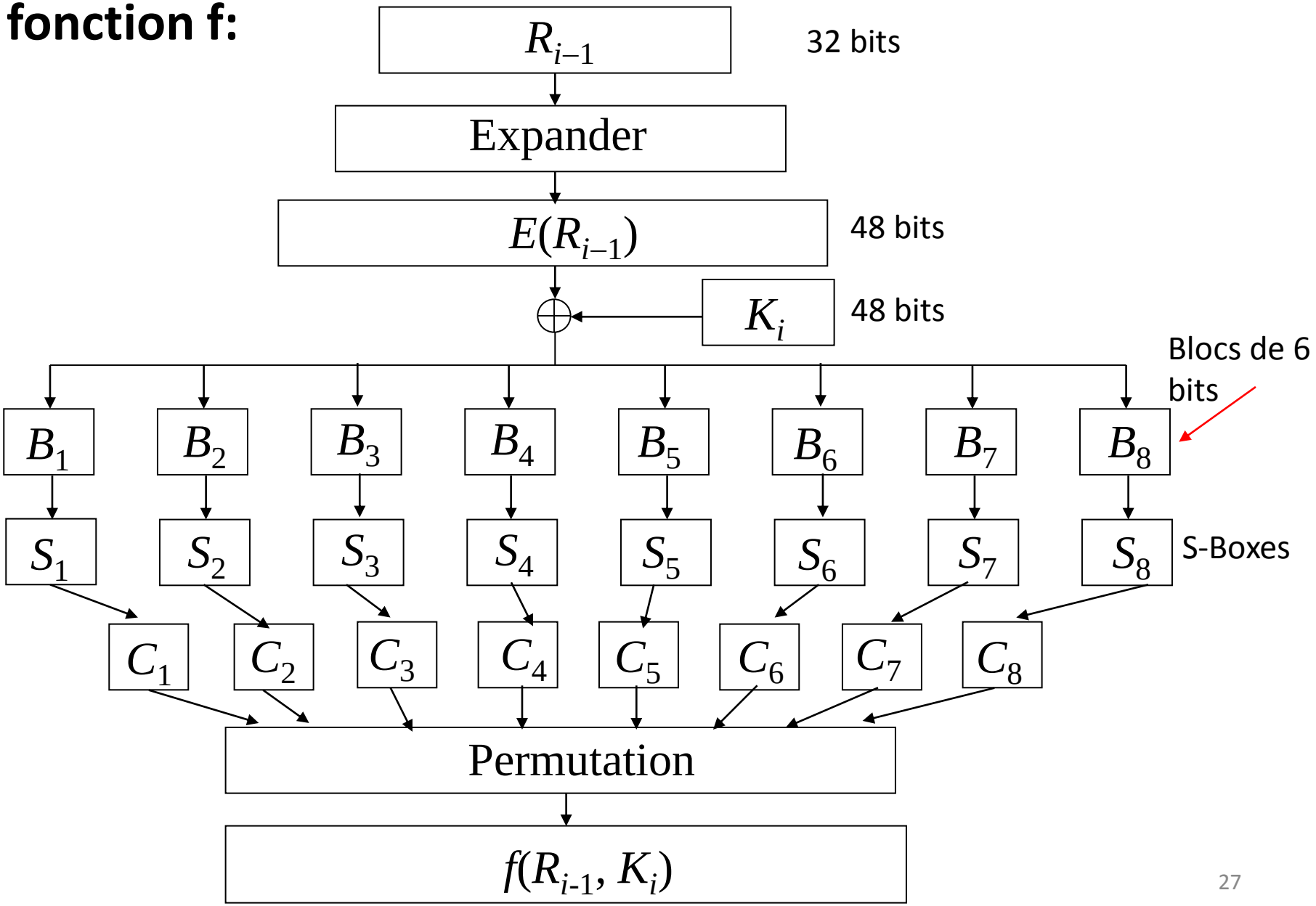
Chiffrement par Blocs : Le DES

La fonction f:



Chiffrement par Blocs : Le DES

La fonction f:



Chiffrement par Blocs : Le DES

La fonction f:

La fonction *f* procède comme suite :

1) La chaîne R_{i-1} (32 bits) en entrée est expansée en $E(R_{i-1})$ (48 bits) en utilisant la table d'expansion suivante :

(32 1 2 3 4 5 4 5 6 7 8 9 8 9 10 11 12 13 12 13 14
15 16 17 16 17 18 19 20 21 20 21 22 23 24 25 24 25 26 27
29 28 29 30 31 32)

2) Calculer $E(R_{i-1}) \otimes K_i$ est écrire le résultat sous la forme $B_1B_2...B_8$ tel que chaque B_i est un groupe de 6 bits.

3) Transformer chaque B_i en utilise la S-Box S_i , la ligne d'une S-Box est spécifiée par les bits b_1b_6 de B_i et la colonne par $b_2b_3b_4b_5$. Le résultat est 8 chaînes de 4 bits chacun $C_1C_2....C_8$.

- Les 8 S-Boxes sont fixes (prédéfini).

Chiffrement par Blocs : Le DES

- Les S-Boxes utilisées sont:

S-box 1

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

S-box 2

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

S-box 3

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

Chiffrement par Blocs : Le DES

- Les S-Boxes utilisées sont:

S-box 4

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

S-box 5

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

S-box 6

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

Chiffrement par Blocs : Le DES

S-box 7

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

S-box 8

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

4. La chaîne résultante $C_1C_2…C_8$ est permutée à la fin en utilisant la permutation fixe suivante :

(16 7 20 21 29 12 28 17 1 15 23 26 5 18 31 10 2 8 24 14 32 27 3 9 19 13 30 6 22 11 4 25)

- Le résultat est une chaîne de 32 bits qui donne $f(R_{i-1}, K_i)$.

Chiffrement par Blocs : Le DES

La gestion des clés

- La clé initiale du système DES et sur 64 bits, seulement 56 bits sont utilisés (**bits de parité**). Les 56 bits sont d'abord permutés en utilisant la table suivante :

(57 49 41 33 25 17 9 1 58 50 42 34 26 18 10 2 59 51 43 35 27
19 11 3 60 52 44 36 63 55 47 39 31 23 15 7 62 54 46 38 30
22 14 6 61 53 45 37 29 21 13 5 28 20 12 4)

- Le résultat obtenu est ensuite divisé en deux parties C_0 et D_0 chacune sur 28 bits.
- Pour i=1..16 (chaque itération) on calcule $C_i=LS_i(C_{i-1})$ et $D_i=LS_i(D_{i-1})$,tel que LS_i est un shift à gauche par une ou deux positions, selon la table suivante :

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Chiffrement par Blocs : Le DES

La gestion des clés

- C_i et D_i sont de nouveau combinés en 56 bits.
- Ensuite, 48 bits seulement sont sélectionnés de la chaîne résultante selon la table suivante :

(14 17 11 24 1 5 3 28 15 6 21 10 23 19 12 4 26 8 16 7
27 20 13 2 41 52 31 37 47 55 30 40 51 45 33 48 44 49 39
56 34 53 46 42 50 36 29 32)

- Le résultat et une clé Ki de 48 bits dans chaque itération i.
- Chaque clé K_i est utilisée dans un tour i. Les clés sont toutes une version transformée de la clé initiale sur 56 bits.

Chiffrement par Blocs : Le DES

Clés faibles de DES:

- Une clé K est faible si $E_K(E_K(x)) = x$.
- Une paire K_1 et K_2 de clés et mi-faible si : $E_{K1}(E_{K2}(x)) = x$.
- Les clés faibles ont la particularité de générer des sous clés identiques par paire :

  $$(k_1=k_{16}, k_2=k_{15}, \ldots\ldots, k_8=k_9),$$

  ceci facilite la cryptanalyse.
- DES possède 4 clés faibles (et 6 paires mi-faibles):

  0101 0101 0101 0101
  1F1F 1F1F 0E0E 0E0E
  E0E0 E0E0 F1F1 F1F1
  FEFE FEFE FEFE FEFE

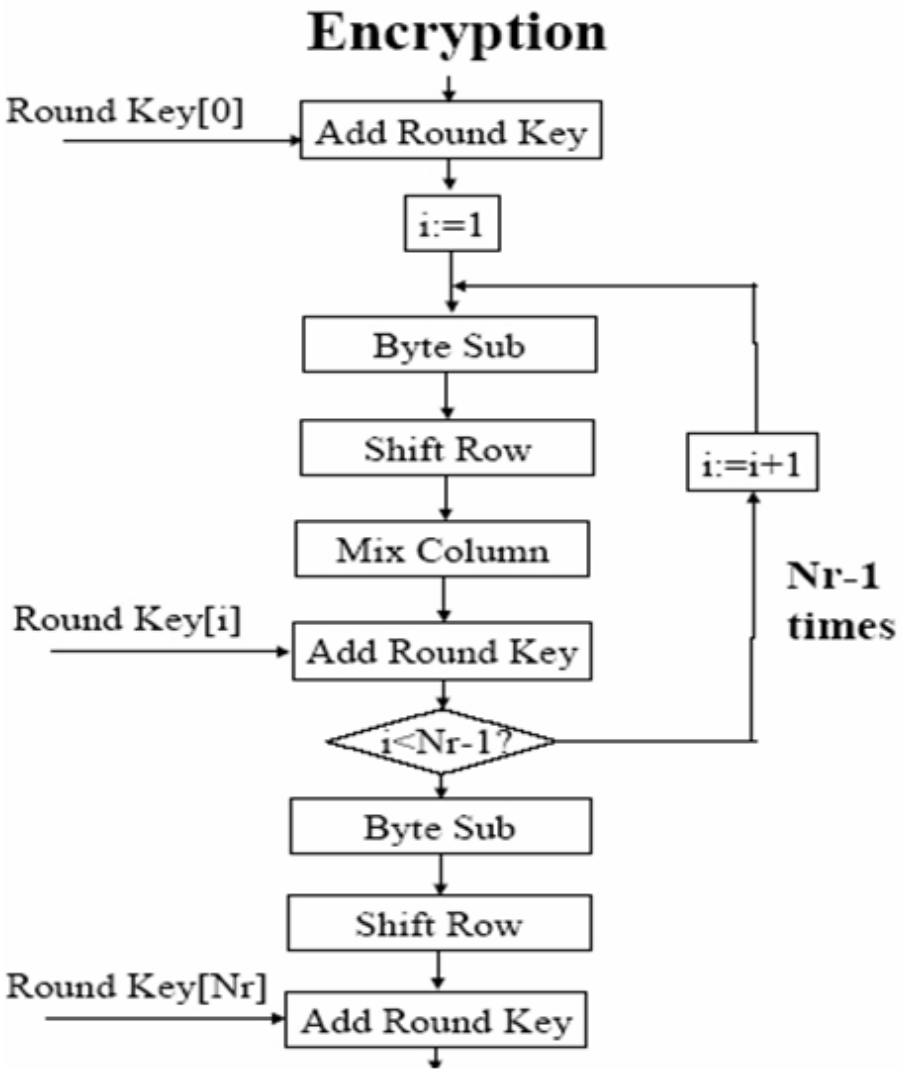
Chiffrement par Blocs : AES

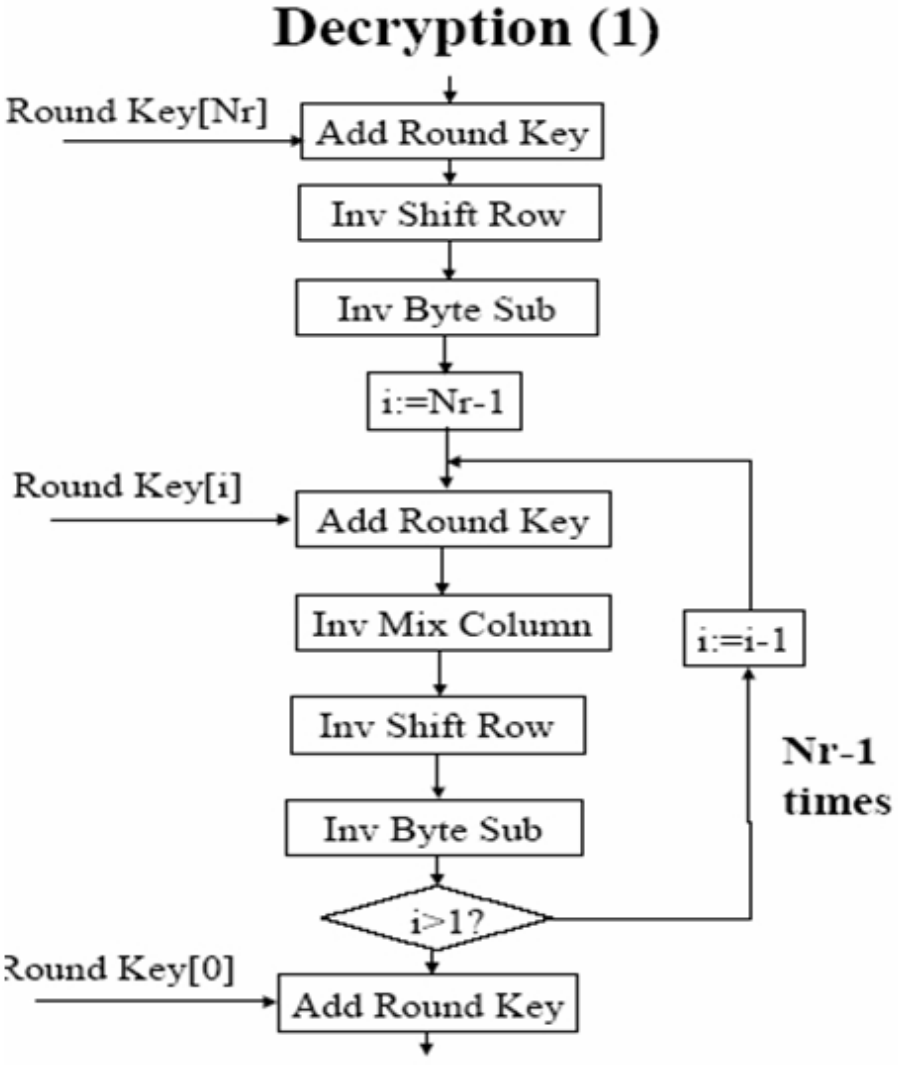
- ***Advanced Encryption Standard*** ou **AES** (standard de chiffrement avancé), aussi connu sous le nom de **Rijndael**, est un algorithme de chiffrement symétrique, choisi en octobre 2000 par le NIST pour être le nouveau standard de chiffrement pour les organisations du gouvernement des États-Unis.
- L'algorithme prend en entrée un bloc de 128 (AES), 192 ou 256 bits (Rijndael), (N_b = 4, 6 ou 8).

  128 bits = 16 octets = 4 mots de 32 bits
- La clé fait 128, 192 ou 256 bits. Donc la taille de la clé est caractérisée par N_k = 4, 6 ou 8.
- Comme DES, AES exécute une séquence de rondes. On note N_r le nombre de rondes qui doivent être effectuées. Ce nombre dépend des valeurs de N_b et de N_k.
- Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours.

Chiffrement par Blocs : AES

- Les algorithmes de chiffrement et de déchiffrement d'AES sont donnés par la figure suivante :

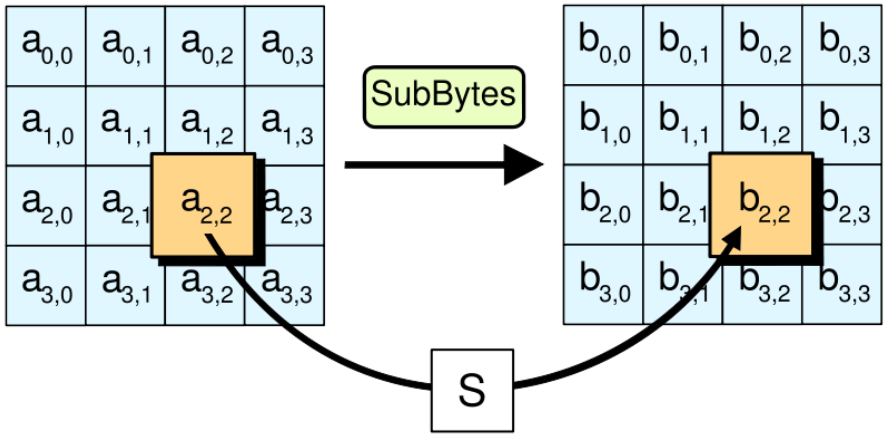




Chiffrement par Blocs : AES

- Les 16 octets en entrée sont placés dans une matrice de $4xN_b$ éléments.
- Le chiffrement AES consiste en une addition initiale de clé, notée AddRoundKey, suivie par N_r-1 rondes, chacune constituée de quatre étapes :

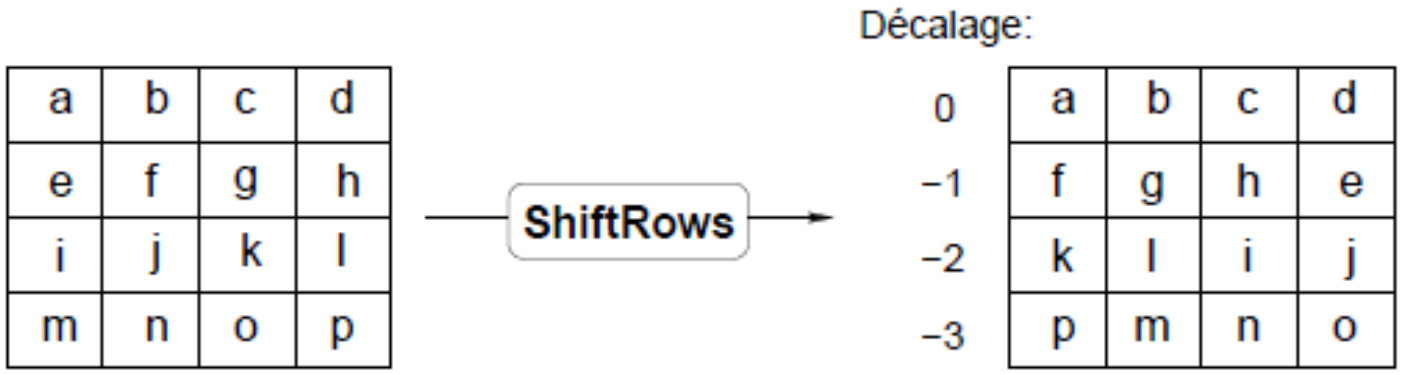
1) **SubBytes :** il s'agit d'une substitution non-linéaire lors de laquelle chaque octet est remplacé par un autre octet choisi dans une table particulière une S-Box.



Chiffrement par Blocs : AES

2) **ShiftRows:** est une étape de transposition où chaque élément de la matrice est décalé cycliquement à gauche d'un certain nombre de colonnes.

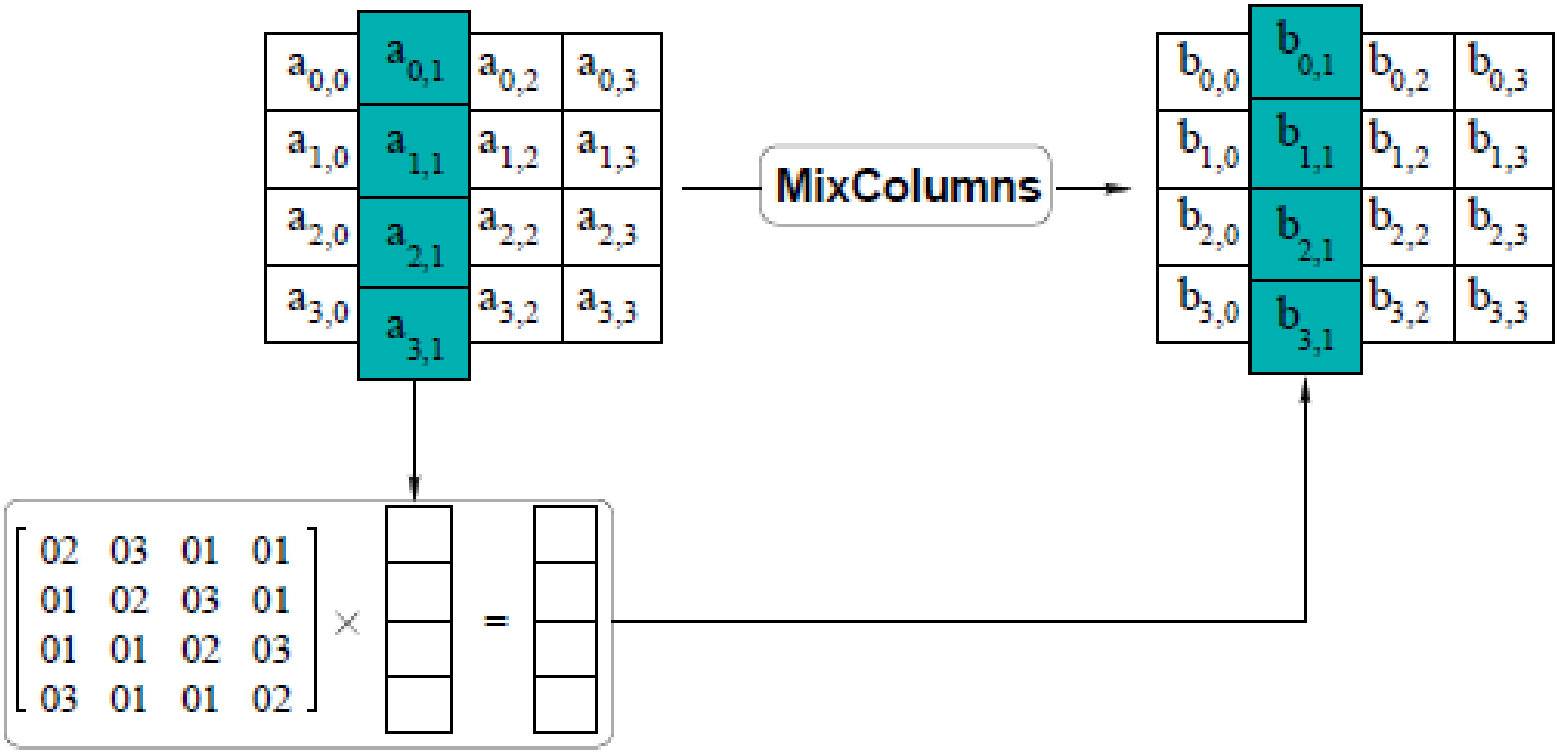
- Chaque ligne est décalée par C_i position:



Opération Shiftrows dans AES

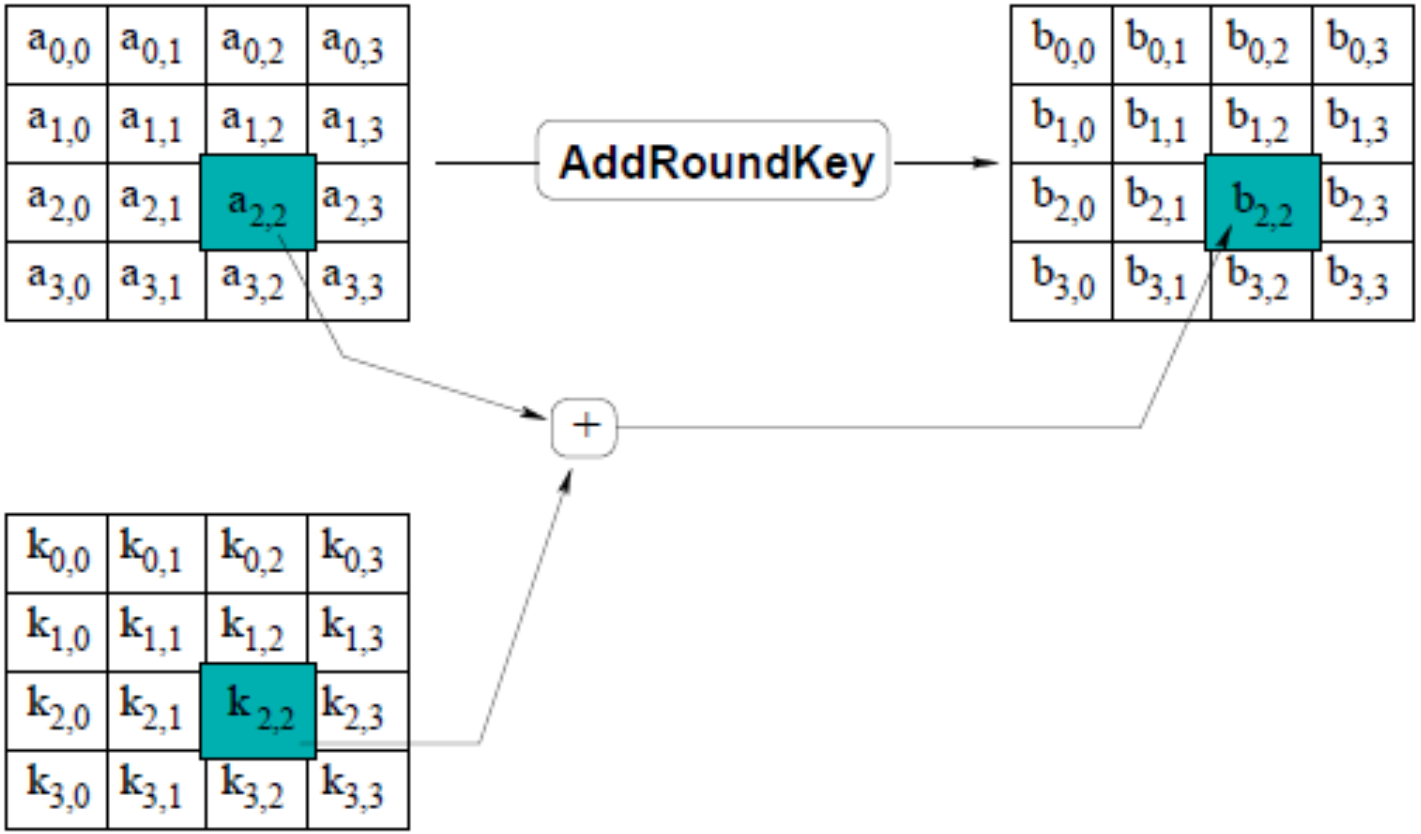
Chiffrement par Blocs : AES

3) **MixColumns** effectue un produit matriciel en opérant sur chaque colonne (vu alors comme un vecteur) de la matrice.



Chiffrement par Blocs : AES

4) **AddRoundKey :** qui combine par addition chaque octet avec l'octet correspondant dans une clé de ronde obtenue par diversification de la clé de chiffrement.



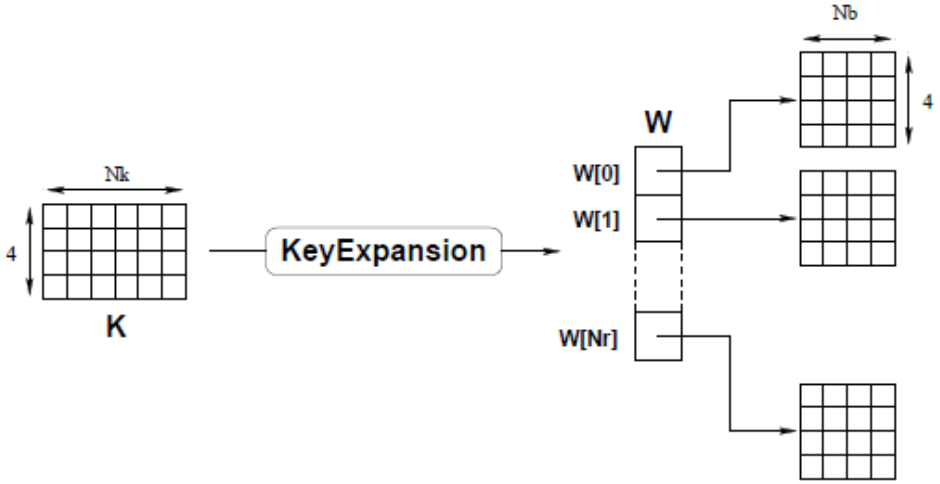
Chiffrement par Blocs : AES

- Enfin, une ronde finale **FinalRound** est appliquée (elle correspond à une ronde dans laquelle l'étape **MixColumns** est omise).

```
AES_Encrypt(State, K) {
  KeyExpansion(K, RoundKeys);
/* Addition initiale */
AddRoundKey(State, RoundKeys[0]);
/* Les Nr-1 rondes */
for (r=1; i<Nr; r++) { SubBytes(State);
                       ShiftRows(State);
                       MixColumns(State);
                       AddRoundKey(State, RoundKeys[r]); }
 /* FinalRound */
SubBytes(State);
ShiftRows(State);
 AddRoundKey(State, RoundKeys[Nr]);
}
```

Chiffrement par Blocs : AES

- La clé initiale K de taille $4*N_k$ octet est diversifier en une clé étendue de $4*N_b*(N_r+1)$ octet, donc N_r+1 ensemble de clé chacun de taille $4*N_b$.



- Une procédure récursive spéciale est utilisée pour la génération des nouvelles clés. Une Rotation est effectuée avec une permutation (S-Box prédéfinit) pour générer chaque ensemble.

Chiffrement par Blocs : AES

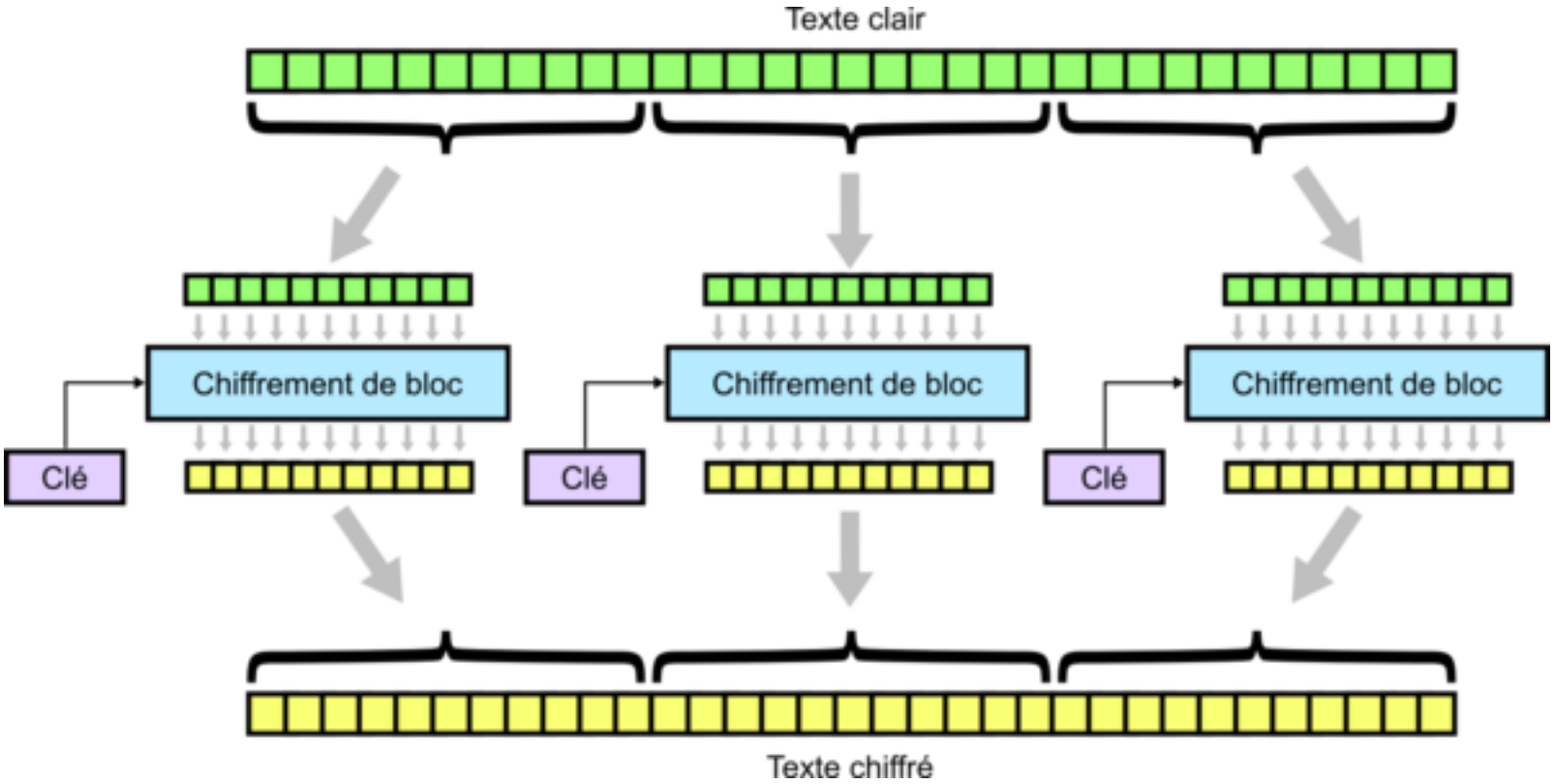
- L'AES n'a pour l'instant pas été cassé et la recherche exhaustive demeure la seule solution. AES a été conçu de telle manière à rendre des méthodes classiques comme la cryptanalyse linéaire ou différentielle très difficiles.
- Performance:
  1) Portable sur un grand nombre de processeurs.
  2) Algorithme suffisamment parallèle pour exploiter les architectures multiprocesseurs.
  3) Débit de cryptage jusqu'à 1Gb/s (cryptage matériel).

Modes d'opération

- Il y a différentes méthodes d'utiliser le **chiffrement par blocs**, elles sont appelées « **modes d'opération** » (*modes of operation*). Quatre modes sont définis dans ANSI X3.106-1983. Les quatre standards sont :
  - **Electronic Code Book** (ECB),
  - **Cipher Block Chaining** (CBC),
  - **Cipher FeedBack** (CFB)
  - **Output FeedBack** (OFB).
- Le chiffrement par blocs DES par exemple s'utilise avec ces modes d'opération. À noter qu'il y a d'autres modes d'opération existants.

Mode opératoire ECB

- Il s'agit du mode le plus simple. Le message à chiffrer est subdivisé en plusieurs blocs qui sont chiffrés séparément les uns après les autres.



Mode opératoire ECB

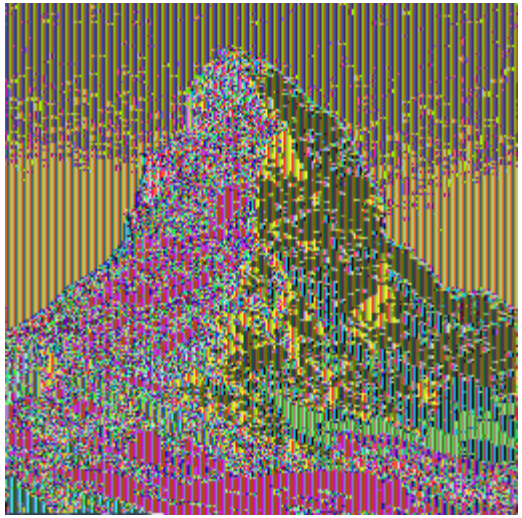
- Le **gros défaut** de cette méthode est que deux blocs avec le **même contenu** seront chiffrés de la **même manière**, on peut donc tirer des informations à partir du texte chiffré en cherchant les **séquences identiques**.

- Ce mode est pour ces raisons **fortement déconseillé** dans toute application cryptographique. Le seul avantage qu'il peut procurer est un **accès rapide** à une **zone quelconque** du texte chiffré et la possibilité de déchiffrer une partie seulement des données.

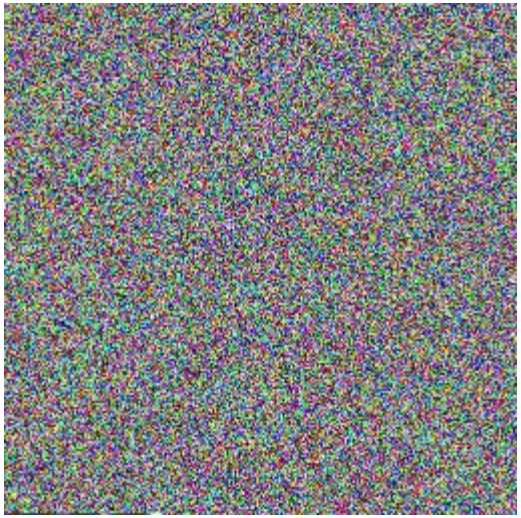
Mode opératoire ECB

- La vulnérabilité est encore plus flagrante sur une image. Les images sont constituées de nombreuses redondances qui font que des blocs sont chiffrés de la même manière en mode ECB. Ci-dessous, le chiffrement en ECB est réalisé sur des blocs de 4 pixels. On distingue très nettement les formes du Cervin ainsi que les séparations entre les blocs.

Originale



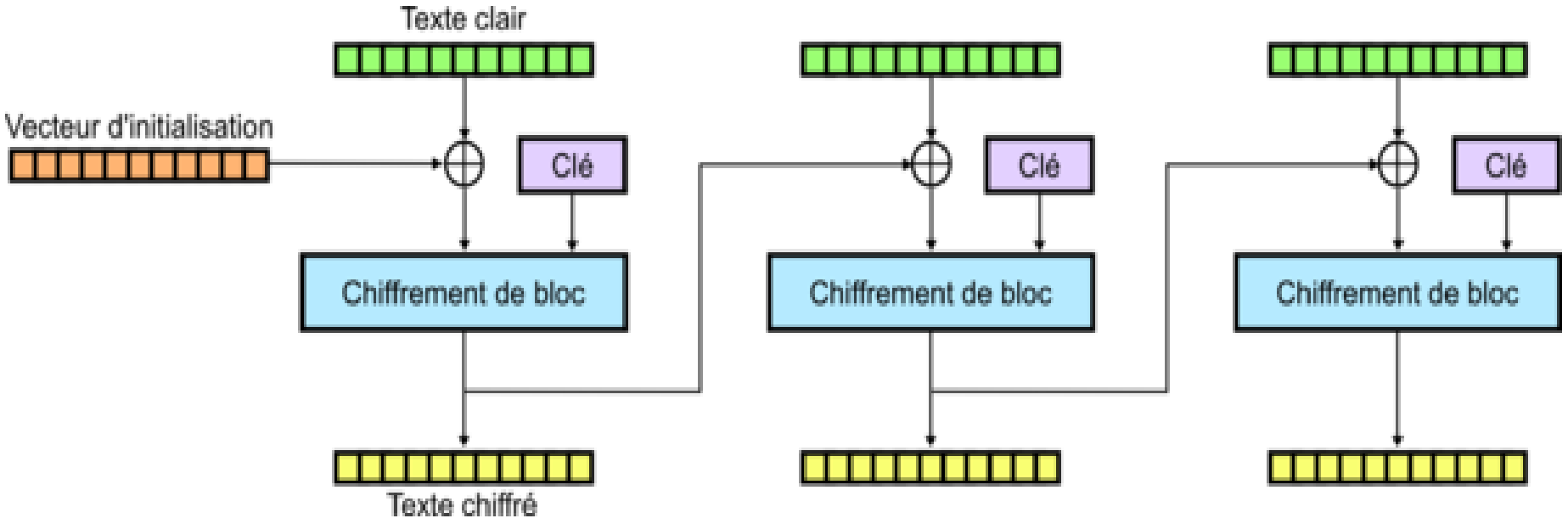
avec ECB



Autre mode

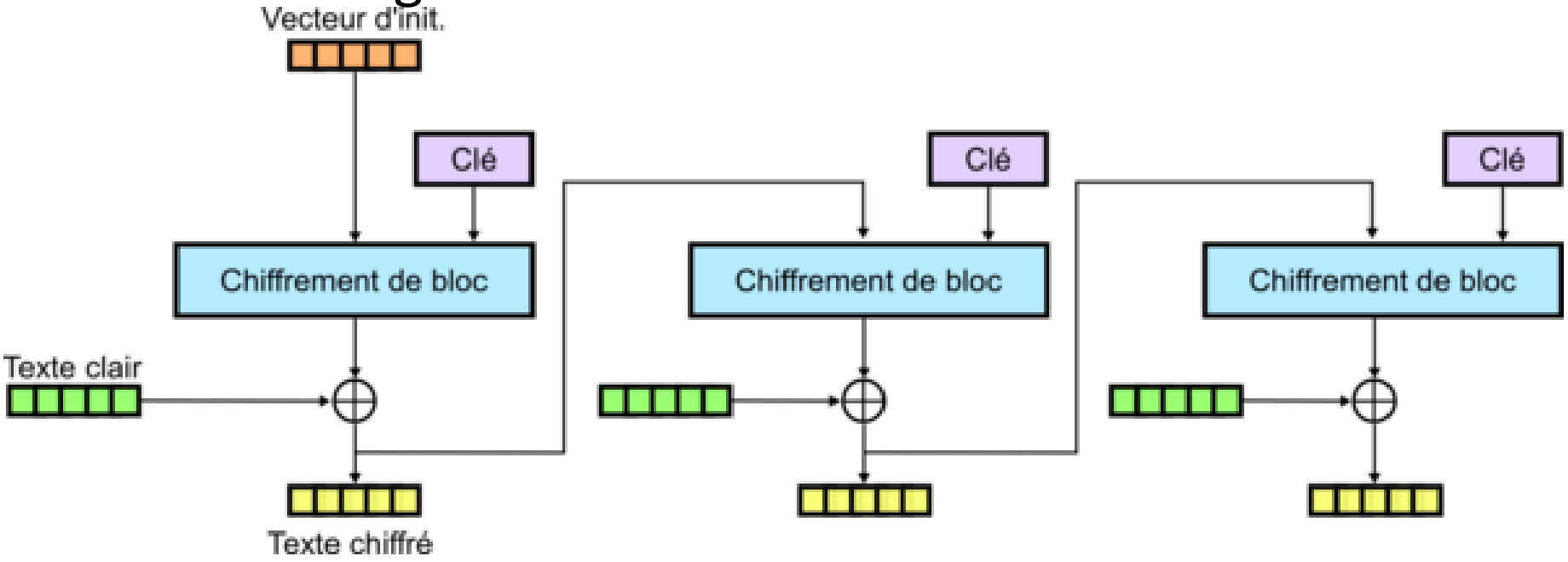
Mode opératoire CBC (Enchainement des blocs)

- Dans ce mode, on applique sur chaque bloc un « OU exclusif » avec le chiffrement du bloc précédent avant qu'il soit lui-même chiffré. De plus, afin de rendre chaque message unique, un **vecteur d'initialisation** (IV) est utilisé.



Mode opératoire CFB (Chiffrement à rétroaction)

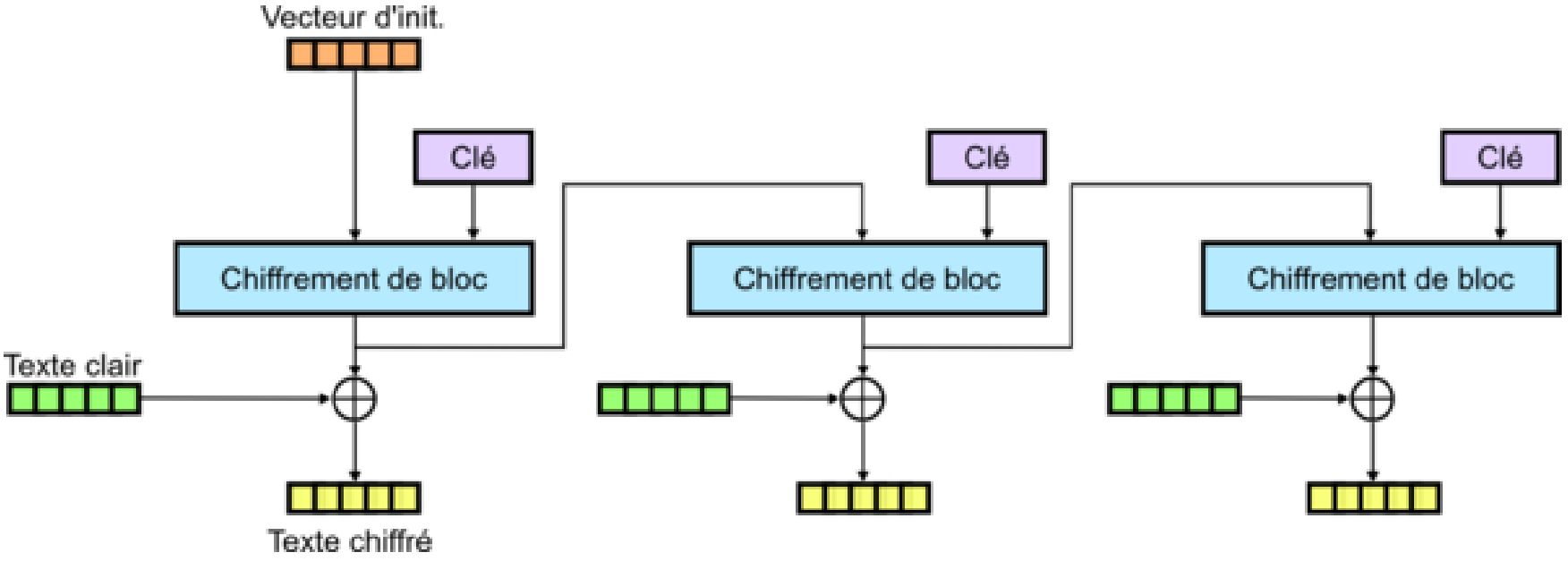
- Ce mode et le suivant agissent comme un chiffrement par flux. Ils génèrent un flux de clés qui est ensuite appliqué au document original.



- Dans ce mode, le flux de clé est obtenu en chiffrant le précédent bloc chiffré. C'est le mode le plus utilisé.

Mode opératoire OFB (à rétroaction de sortie)

- Dans ce mode, le flux de clé est obtenu en chiffrant le précédent flux de clé.



- C'est un mode de chiffrement de flux qui possède les mêmes avantages que CFB. De plus, il est possible de le précalculer en chiffrant successivement le vecteur d'initialisation.

Conclusion

- Les algorithmes de chiffrement à clé secrète ne permettent pas de garantir la **non-répudiation**, et il posent le problème de transmission de la clé partagée.
- Quand il y'a beaucoup d'utilisateurs => problèmes de gestion de clé, problème de renouvellement des clés…

  => Solution : cryptographie à **clé publique** !
- Leurs avantages par rapport aux algorithmes à clé publique : **très rapide** (des dizaines de fois !), **facile** à implémenté et requirent **moins de ressources** (implémentation Hard).
- Certains systèmes utilisent une **combinaison** des deux : **PGP = IDEA + RSA**.