Chapitre 3 : Cryptographie Symétrique moderne

1 Objectif

Pour un usage sécurisé de la cryptographie, il faut :

- Utiliser un algorithme de cryptage fiable : un adversaire qui connait l'algorithme ne doit pas être capable de déterminer la clé ou déchiffrer un texte chiffré.
- Un échange sécurisé de la clé de chiffrement.

Conclusion : l'algorithme de chiffrement ne doit pas être secret, seulement la clé doit l'être.

2 Diffusion et confusion

Pour combattre les attaques de type « cryptanalyse par statistique », Shannon a proposé en 1949 les deux méthodes suivantes :

- **Diffusion** : si on change une seule lettre du message en clair, plusieurs lettres chiffrées doivent changer, et réciproquement (si on change une seule lettre du message chiffré, plusieurs lettres du message en clair doivent changer).
- **Confusion** : si on change une seule lettre de la clé, tout le message chiffré doit changer.

3 Data Encryption Standard (DES)

3.1 Introduction

L'algorithme DES est définit en 1977 par National Institute of Standards and Technology (NIST). Il était l'algorithme de chiffrement symétrique (càd l'algorithme de chiffrement à clé sécrète) le plus utilisé au monde entre 1977 et 2001 (date d'apparition du nouveau standard AES).

Notations :

C (Ciphertext) : message chiffré
P (Plaintext) : message en clair
K (Key) : clé secrète
E (Encryption) : fonction de chiffrement
D (Decryption) : fonction de déchiffrement

Chiffrement : $\mathbf{C = E_K (P)}$
Déchiffrement : $\mathbf{P = D_K (C)}$

3.2 Paramètres de DES

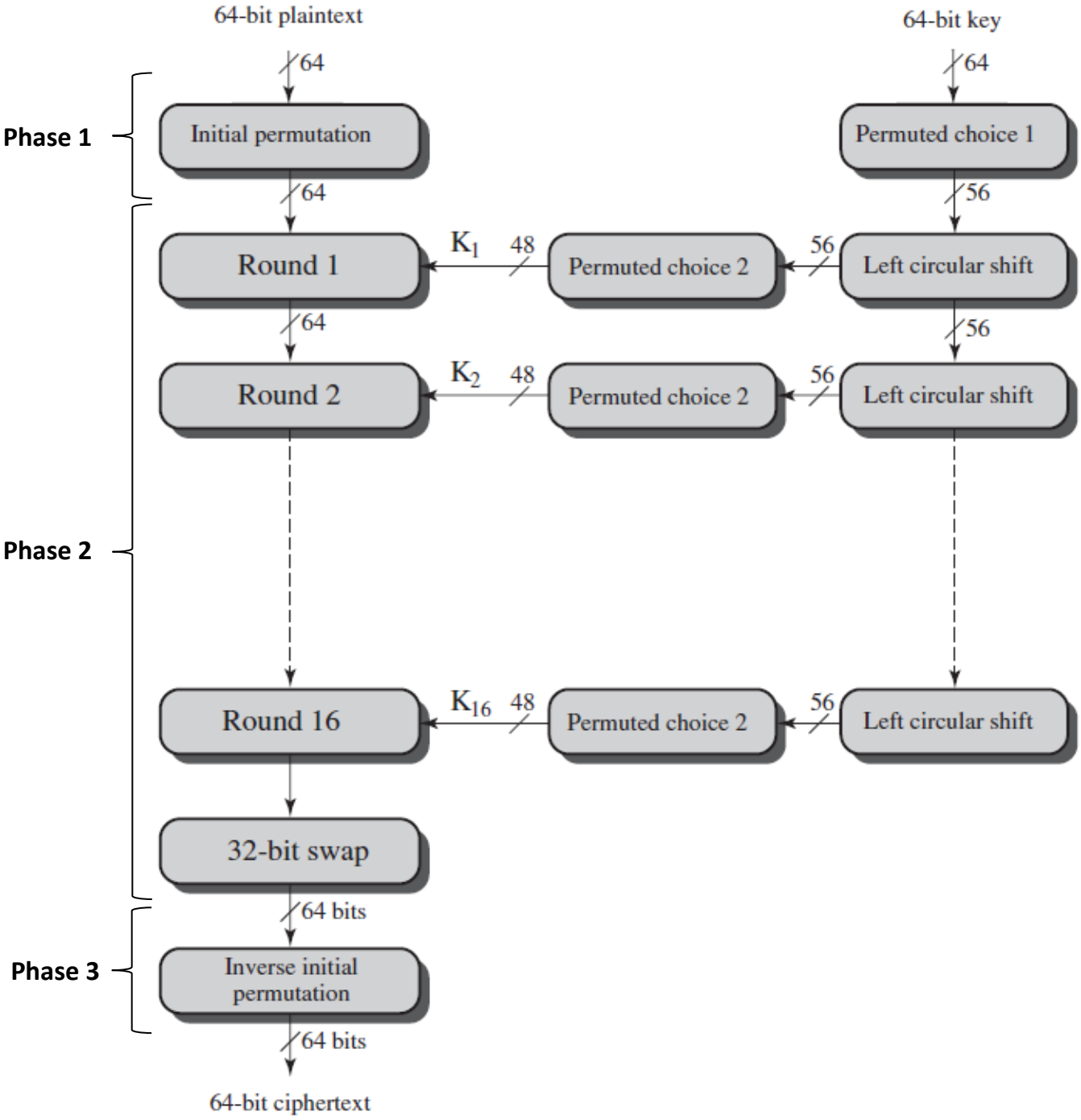
- **Chiffrement par bloc** : DES utilise un chiffrement par bloc de 64 bits. Un message à chiffrer est découpé en plusieurs blocs. On ajoute des bits de bourrage au dernier bloc pour atteindre la taille 64 bits.
- **Taille de la clé secrète** : Augmenter la taille de la clé améliore la sécurité (comment ? ....) mais réduit la vitesse de chiffrement/déchiffrement. DES utilise une clé de 56 bits.
  Remarque : en pratique, la fonction de chiffrement/déchiffrement demande une clé de 64 bits (8 octets) mais 7 bits de chaque octet sont utilisés et le $8^{ème}$ bit est ignoré (7bits x 8octets = 56 bits).
- **Nombre de rounds** : un bloc est chiffré à travers plusieurs rounds. La sécurité offerte par un seul round est insuffisante. Ainsi, augmenter le nombre de rounds permet d'améliorer la sécurité. DES chiffre chaque bloc d'un message à travers 16 rounds. Chaque round utilise une clé intermédiaire de 48 bits générée à partir de la clé sécrète.

3.3 Chiffrement DES

Le chiffrement d'un bloc se fait en 3 phases :

- **Phase 1** : le bloc passe par une permutation initiale (**IP**: Initial Permutation) qui réarrange les bits.
- **Phase 2** : cette phase est composée par 16 rounds. Chaque round réalise la substitution et la permutation des bits en appliquant une clé intermédiaire « $\mathbf{K_i}$ » de 48 bits. Après round 16, les deux moitiés du bloc sont permutées (**32-bit swap**).
- **Phase 3** : Appliquer une « permutation des bits » qui est l'inverse de la permutation initiale (IP^{-1}).

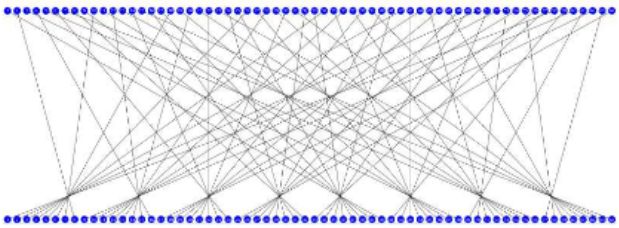
Phases de chiffrement de DES :



3.3.1 Phase 1 : Initial Permutation (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

64 bits d'entrée



64 bits de sortie

Tableau de permutation initiale : les 64 bits de sortie en fonction du numéro des bits d'entrée

Le bit N° 1 de sortie provient du bit N° 58 d'entrée. Le bit N° 21 de sortie provient du bit N° 30 d'entrée. Le bit N° 5 de sortie provient de quel bit d'entrée ? Le bit N° 64 de sortie provient de quel bit d'entrée ?

Exemple

On considère la fonction de permutation simplifiée qui produit les bits de sortie suivants :

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

a) Quel est le nombre de bits de sortie ?

b) Soit les bits d'entrée suivant « 1111 1111 0000 0000 ». Donner les bits de sortie.

c) Donner la table de permutation inverse.

d) Vérifier que l'application de la permutation inverse sur les bits de sortie de la fonction de permutation (question b) permet d'obtenir les bits d'entrée initiaux.

e) Donner la première ligne de la table de permutation initiale inverse (IP^{-1}) de DES.

3.3.2 Phase 2 : 16 rounds

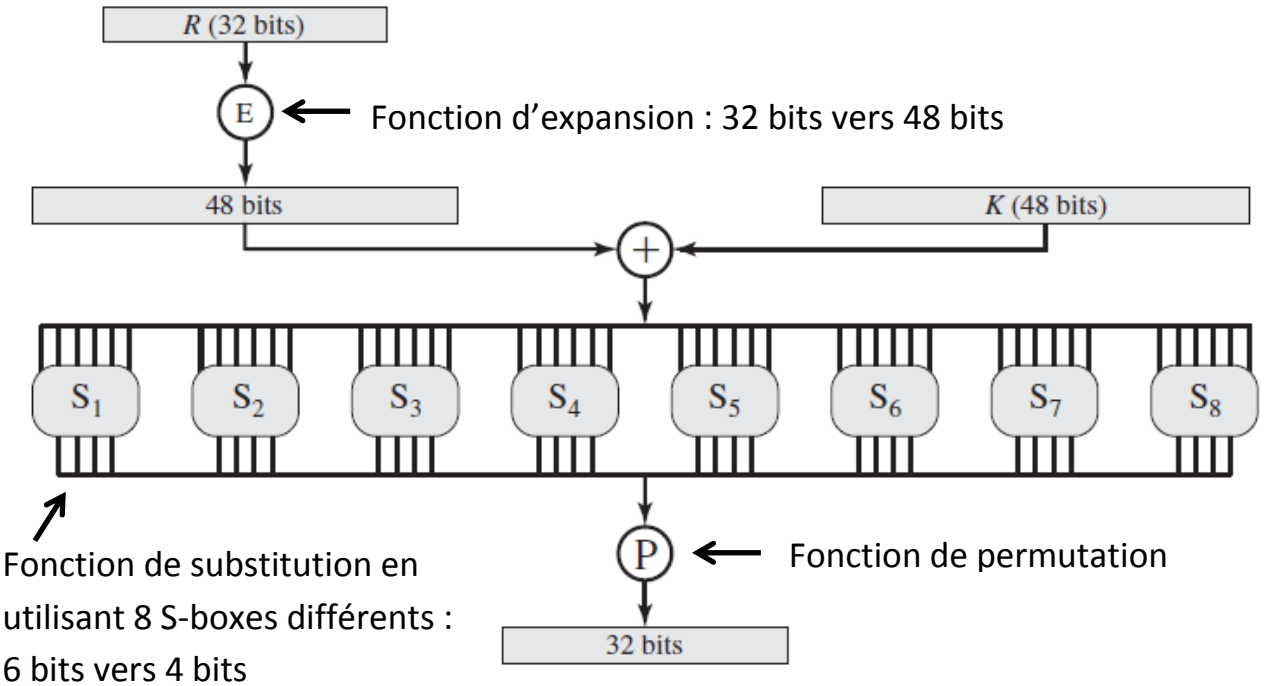
Un bloc de 64 bits : deux moitiés de 32 bits (L) et (R)

Chaque round réalise les opérations suivantes :

$L_i = R_{i-1}$

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

La fonction F :



Fonction d'expansion de la fonction F :

« R » passe par une fonction d'expansion (32 vers 48 bits) qui duplique 16 bits de « R » selon le tableau suivant :

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	28	29
28	29	30	31	32	1

Fonction de substitution de la fonction F :

La fonction de substitution est réalisée avec 8 S-boxes différents. Un S-Boxe reçoit 6 bits en entrée et donne 4 bits en sortie. Une substitution est réalisée comme suit :

- Le premier et dernier bits en entrée à un S-Boxe permettent de sélectionner la ligne du S-Boxe
- Les 4 bits du milieu permettent de sélectionner la colonne du S-Boxe
- La valeur décimale représente les 4 bits de sortie du S-boxe

Exemple : entrée 011001, quelle est la sortie de S1 et de S2 ?

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_1

00	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
01	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
10	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
11	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_2

Entrée 011001, Sortie de S1 : .....

Entrée 011001, Sortie de S2 : .....

Fonction de permutation de la fonction F :

La sortie des 8 S-boxes est 32 bits. Ces 32 bits sont permutés selon le tableau suivant :

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

3.3.3 Génération des clés intermédiaires

DES reçoit une clé initiale de 64 bits (seulement 56 bits seront utilisés : le 8eme bit de chaque octet est ignoré). Cette clé est utilisée pour générer 16 clés intermédiaires de 48 bits. La clé initiale passe par une première permutation des 56 bits (**PC1 : Permuted Choice 1**). Pour chaque round, une clé intermédiaire « K_i » est générée en combinant les opérations suivantes :

- Un décalage circulaire (des bits) à gauche
- Une 2ème permutation (**PC2 : Permuted Choice 2**) qui permet de sélectionner 48 bits parmi 56 bits

Remarque : grâce au décalage on obtient une clé différente à chaque round

La clé initiale passe d'abord par **PC1** (tableau suivant) qui produit une clé de 56 bits organisée en 2 moitiés : C_0 et D_0

C_0	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
D_0	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

PC1

Au round i, les deux moitiés C_{i-1} et D_{i-1} passent séparément par une rotation circulaire à gauche de 1 ou 2 bits (voir tableau suivant). Le résultat C_i et D_i passent ensuite par **PC2** pour obtenir K_i.

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

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

PC2

3.4 Fiabilité de DES

Avec une clé de 56 bits, on a 2^{56} clés possibles (environ 7,2 x 10^{16} clés). Une attaque par force brute doit parcourir en moyenne la moitié des clés.

Le temps moyen d'une attaque par force brute en fonction de la vitesse de chiffrement d'une machine :

- 10^6 chiffrements par seconde :

- 10^9 chiffrements par seconde :

- 10^{13} chiffrements par seconde :

En 1977, DES était suffisamment robuste. Mais de nos jours, il est possible de le casser par une attaque de type force brute. Pour améliorer la sécurité de DES, plusieurs alternatives ont été proposées telles que Double DES, Triple DES et AES.

4 Double DES

Double DES applique DES deux fois sur un message en utilisant 2 clé différentes, K1 et K2.

Chiffrement : $C = E_{K2} (E_{K1} (P))$

Déchiffrement : $P = D_{K1} (D_{K2} (C))$

L'objectif de Double DES est d'augmenter le nombre de clés possibles pour combattre l'attaque par force brute. Avec 2 clés de 56 bits, on pense avoir 2^{56} x 2^{56} = 2^{112} clés possibles. Si c'était vrai, « Double DES » serait suffisamment fiable de nos jours. Mais cet algorithme est vulnérable à l'attaque « rencontre au milieu » (en anglais Meet-in-the-middle).

L'attaque « rencontre au milieu » : On suppose qu'un attaquant ignore les clés K1 et K2 mais détient un message en clair (P) et le message chiffré correspondant (C). On obtient E_{K1} (P) = D_{K2} (C). L'attaque se déroule comme suit :

- On chiffre P avec toutes les valeurs possibles de K1 (2^{56} clés) et on mémorise les résultats dans une base de données.
- Pour chaque valeur possible de K2 (2^{56} clés), on déchiffre C et on compare le résultat avec tous les résultats mémorisés de chiffrement de P : si on a une égalité ➔ K1 et K2 trouvées.

➔ Avec l'attaque « rencontre au milieu », le nombre maximal de clés à essayer est 2^{56} + 2^{56} = 2^{57} clés et non pas 2^{112} clés.

Exercice :

L'algorithme « Triple DES » consiste à chiffrer un message trois fois en utilisant DES et 3 clés différentes dans l'ordre suivant : K1, K2 et K3.

1) Exprimer le message chiffré C en fonction de E, le message en clair P et les 3 clés.

2) Exprimer P en fonction de D, C et les 3 clés.

3) Quel est le nombre de clés possibles sans utiliser l'attaque « rencontre au milieu » ?

4) Que devient ce nombre avec l'attaque « rencontre au milieu » ?

5 Advanced Encryption Standard (AES)

5.1 Introduction

En 1997, NIST lance un concours pour la standardisation d'un nouvel algorithme de chiffrement symétrique qui remplace DES. 15 algorithmes candidats ont été proposés pour ce concours. Les 5 finalistes sont MARS, RC6, RIJNDAEL, SERPENT et TWOFISH. Parmi ces finalistes, l'algorithme RIJNDAEL est sélectionné pour être le nouveau standard AES. Ainsi, AES est définit en 2001.

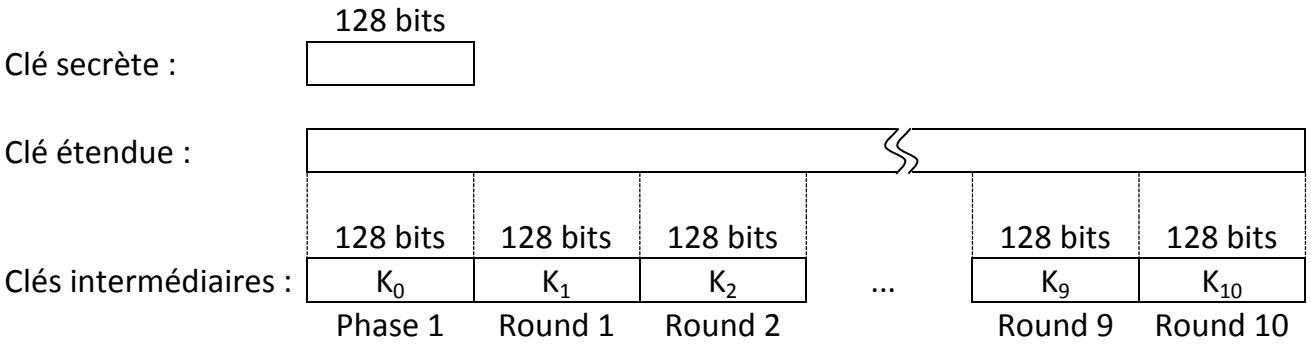
5.2 Paramètres de AES

- **Chiffrement par bloc** : taille du bloc est 128 bits
- **Taille de la clé** : AES utilise des clés avec 3 tailles différentes (128 bits, 192 bits et 256 bits). L'algorithme s'appelle AES-128, AES-192 ou AES-256 selon la taille de la clé.
- **Nombre de rounds** : le chiffrement se déroule en 2 phases. La première phase réalise une transformation initiale du bloc. La deuxième phase est composée de plusieurs rounds. Le nombre de rounds dépond de la taille de la clé :
    - Clé de 128 bits : 10 rounds
    - Clé de 192 bits : 12 rounds
    - Clé de 256 bits : 14 rounds

Le processus de chiffrement/déchiffrement utilise 3 types de clés :

- Clé secrète
- Clé étendue : générée à partir de la clé secrète
- Clés intermédiaires : ce sont des portions de 128 bits de la clé étendue. La phase de transformation initiale ainsi que chaque round utilisent des clés intermédiaires différentes.

Exemple : clés de l'algorithme AES-128



Paramètres de AES :

	AES-128	AES-192	AES-256
Taille de la clé secrète	128 bits	192 bits	256 bits
Taille du bloc	128 bits	128 bits	128 bits
Nombre de rounds	10	12	14
Taille de la clé intermédiaire	128 bits	128 bits	128 bits
Taille de la clé étendue	(10 + 1) x 128 = 1408 bits	(12 + 1) x 128 = 1664 bits	(14 + 1) x 128 = 1920 bits

6 Modes de chiffrement par bloc

Le chiffrement par bloc transforme un bloc de message en clair en un bloc chiffré de même taille. Si la taille du message en clair dépasse la taille d'un bloc, on peut utiliser l'un des 5 modes de chiffrement suivant pour chiffrer la totalité du message

- Electronic Codebook (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR)

Parmi ces 5 modes, on s'intéresse particulièrement à ECB et CBC.

6.1 Electronic Codebook (ECB)

ECB est le mode le plus simple. Il découpe un message en clair en plusieurs blocs, et chiffre chaque bloc indépendamment des autres. Tous les blocs sont chiffrés avec la même clé K. Lorsque la taille de la dernière portion du message est inférieure à la taille d'un bloc, il est nécessaire d'ajouter des données de bourrage. L'ensemble des blocs chiffrés représente le message chiffré.

Pour déchiffrer un message chiffré, on déchiffre chaque bloc indépendamment des autres blocs et en utilisant la même clé K.

Chiffrement ECB :

Déchiffrement ECB :

Inconvénients du mode ECB :

- Deux blocs en clair identiques produisent des blocs chiffrés identiques
- Si on change une seule lettre d'un message en clair, seulement le bloc chiffré contenant la lettre sera modifié (les autres blocs chiffrés restent inchangés).

6.2 Cipher Block Chaining (CBC)

Pour une meilleure sécurité du chiffrement, deux blocs en clair identiques doivent produire des blocs chiffrés différents. Cela est possible en utilisant le mode CBC. Ce mode fonctionne comme suit :

- **Chiffrement** : On applique la fonction XOR sur le bloc en clair à chiffrer et le bloc chiffré précédent. Le résultat de la fonction XOR représente le bloc à chiffrer avec la clé K. Pour chiffrer le premier bloc d'un message, on utilise un bloc spécial qui s'appelle **Vecteur d'Initialisation** (**IV**).
- **Déchiffrement** : On déchiffre le premier bloc et on applique la fonction XOR sur le premier bloc déchiffré et le IV (même IV utilisé en chiffrement). Le résultat obtenu est le premier bloc en clair. On applique la fonction XOR sur chaque bloc déchiffré et le bloc chiffré précédent pour obtenir les blocs en clair.

Chiffrement CBC :

Déchiffrement CBC :

Remarque : L'émetteur et le récepteur doivent échanger la clé K, le IV et le message chiffré. Seulement la clé doit être échangée sur un canal sécurisé. Pour une meilleure sécurité, il faut utiliser un IV différent avec chaque nouveau message.