Cryptographie symétrique

Prof. Ismail Akharraz

Université Sidi Mohamed Ben Abdellah
Faculté Polydisciplinaire de Taza, Maroc

Sommaire

1. Introduction.

- Il y a deux types de cryptographie :
  - La cryptographie symétrique (cryptosystèmes symétriques)
  - La cryptographie asymétrique (cryptosystèmes asymétriques)
- La cryptographie symétrique est :
  - La plus ancienne : ... , Cesar, hill, ...
  - La plus utilisée actuellemnt : grande rapidité
  - Utilise la même clé pour chiffrer et déchiffrer.
- Fonctionne suivant deux procédés différents : cryptage par blocs et cryptage par flôt(en continu ou en stream).

2. Chiffrement par flot

1. Un cryptage en continu effectué bit-à-bit.
2. On crypte bit-à-bit sans attendre la réception complète des données à crypter.
3. Le chiffre One-Time Pad est le seul chiffre qui est théoriquement inviolable.
4. RC4 est l'algorithme le plus utilisé aujourd'hui pour chiffrer les flux.
5. Le chiffrement en continu est surtout utilisé dans les communications en life(sur internet).
6. Il est caractérisé par :
   - une utilisation réduite de la mémoire.
   - peu de propagation d'erreur.
   - pas ou peu d'algorithmes standard

2. Chiffrement par flot

2.1. One-Time Pad.

- Masque jettable (aussi chiffre de Vernam).
- Chiffre poly-alphabétique
- La clé est chaine aléatoire de même longueur que le message d'origine utilisée une sule fois.
- Le chiffre consiste à xorer la clé avec le message.
- Dificile à pratiquer:
  - Une clé pour chaque méssage.
  - Transmission des clés.
  - L'aléatoire
- Théoriquement incassable (Claude Shannon en 1949).
- Histoire : Téléphone rouge entre Washington et Moscow depuis 1963.

2. Chiffrement par flot

2.1. One-Time Pad.

Exemple.

Chiffrer SALUT	
Cesar	One-Pad Time
- 26 décalage possible - 26 chiffre possible pour le mot SALUT - Facile à essayer toutes les possiblités	- La clé est de longueur 5 - Le choix est aléatoire (au hasard) - Chaque lettre a 26 possibilités - 26 x 26 x 26 x 26 x 26 possibilités - 11881376 possibilités

2. Chiffrement par flot

2.2. RC4 (Rivest Cipher 4).

- Algorithme de chiffrement symétrique et rapide créé par Ronald Rivest en 1987.
- RC4 fonctionne de la façon suivante :
  1. Choix d'une clé de longueur entre 1 et 256 octets(En pratique, 5 ou 13 octets).
  2. Créer deux tableaux S et T de taille 256 chacun pour contenir des octets.
  3. Initialiser S avec les nombres de 0 à 256.
  4. Remplir T avec la clef en la répétant autant de fois que nécessaire.
  5. Effectuer des opérations aléatoires des éléments de S en fonction de ceux de T.
  6. Ré-effectuer ces opérations aléatoires sur S pour obtenir la clé finale.
  7. Effectuer un XOR entre la clé et le message à chiffrer.

2. Chiffrement par flot

2.2. RC4 (Rivest Cipher 4).

Remarques.

1. RC4 a été tenu secret jusqu'à 1994 où un algorithme compatible a été révelé par rétro-engineering.
2. RC4 est utilisé dans les transactions chiffrées sur Internet (Intégré dans Secure Socket Layer : SSL).
3. Le OU exclusif n'est rien d'autre qu'une substitution polyalphabétique.
4. RC4 Considéré comme moyennement sûr.

3. Chiffrement par blocs.

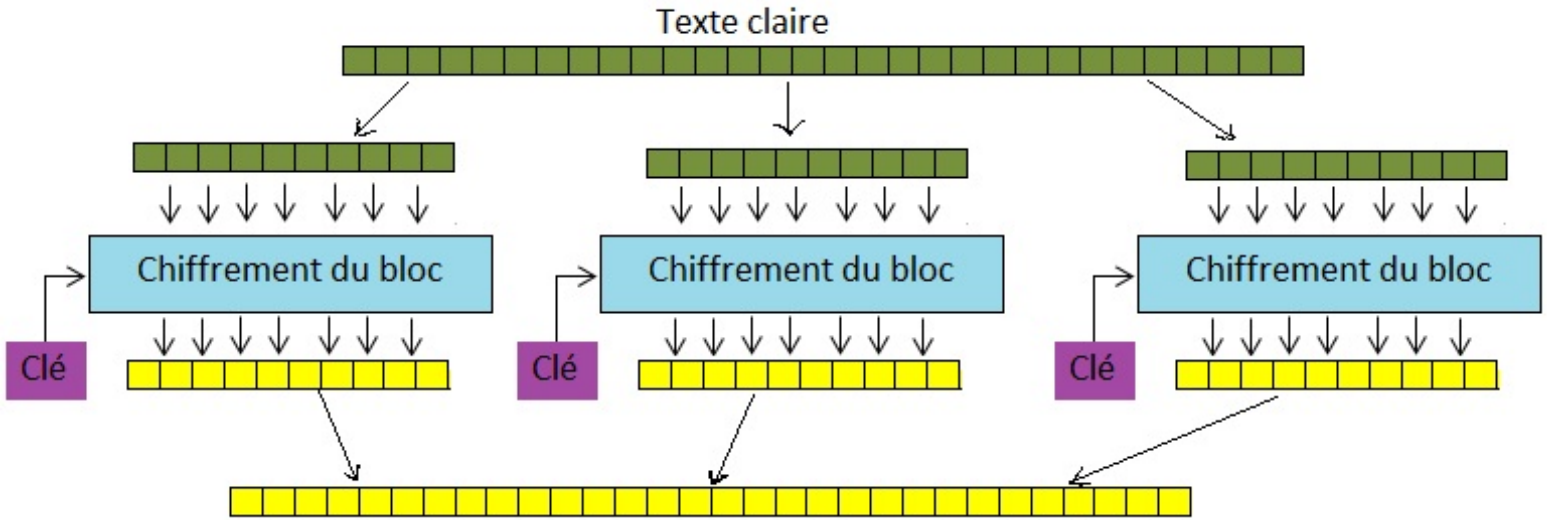
1. Le cryptage en blocs (block-cipher) est très utilisé et permet une meilleure sécurité.
2. Ils s'appliquent à des blocs de données et non à des flux de bits.
3. Ces blocs sont habituellement de 64.
4. La taille de la clé varie suivant l'algorithme et suivant le niveau de sécurité requis :
   - Un cryptage utilisant une clé longue de 40 bits est aisément cassable.
   - Un cryptage de 56 bits est moyen puisque cassable mais nécessitant pas mal de moyens et un temps considérable pour le casser.
   - Un cryptage de 128 bits est plus fort à l'heure actuelle.

3. Chiffrement par blocs.

1. Les algorithmes symétrique par bloc les plus connus : DES, AES, Skipjack ...
2. Ces algorithmes utilisent l'un des modes de chiffrement de blocs suivants :
   - Electronic Code Book (ECB),
   - Cipher Block Chaining (CBC),
   - Cipher Feed Back (CFB),
   - Output Feed Back (OFB).

4. Modes de chiffrement par blocs.

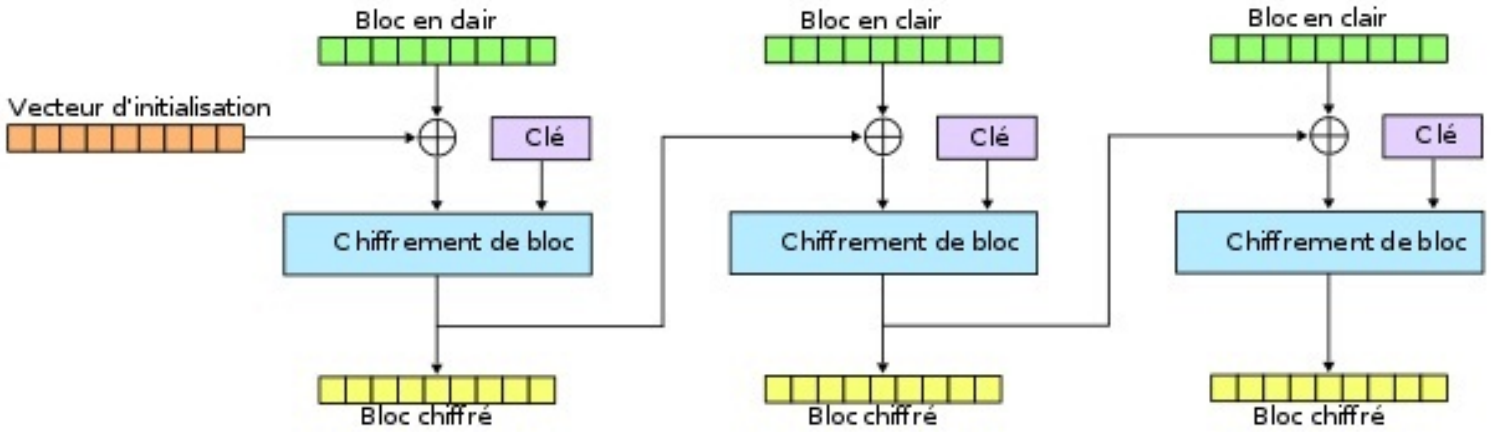
4.1. Electronic Code Book (ECB)



1. Il crypte chaque bloc indépendamment des autres;
2. Cela permet entre de crypter suivant un ordre aléatoire;
3. Mais! ce mode est très vulnérable aux attaques.
4. Mais! pour une clé de 128 bits ou plus, ces attaques ne sont pas pratiquable de nos jours.
5. Sensible à l'inversion ou la duplication de blocs sans que le destinataire s'en aperçoive.

4. Modes de chiffrement par blocs.

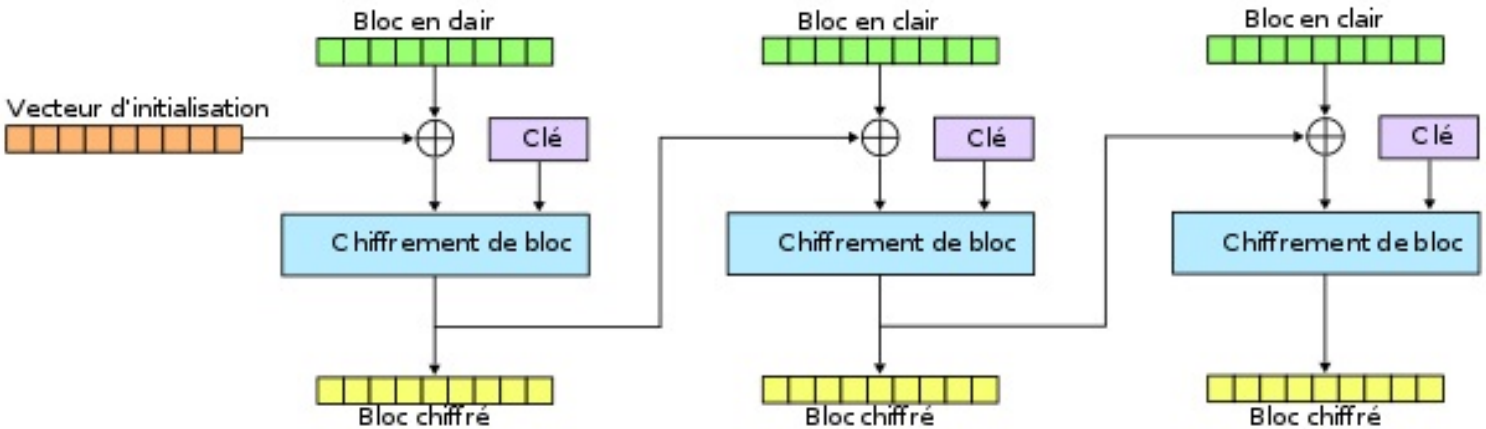
4.2. Cipher Block Chaining (CBC)



1. C'est le mode le plus courant.
2. Le vecteur d'Initialisation change à chaque session, et doit être transmis au destinataire.
3. Il n'est pas nécessaire de le chiffrer avant de l'envoyer : il peut être connu de l'adversaire.
4. Il peut constituer une faille sérieuse s'il est mal choisi.
5. CBC ne peut pas être parallélisé : le bloc courant nécessite que le précédent soit chiffré.

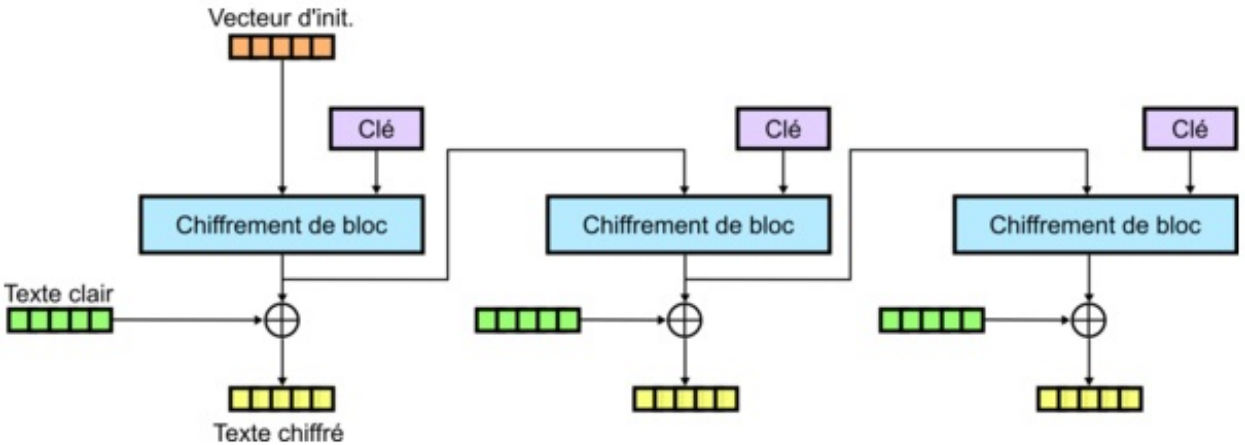
4. Modes de chiffrement par blocs.

4.3. Cipher FeedBack (CFB)



4. Modes de chiffrement par blocs.

4.4. Output Feed Back (OFB)



1. Il est possible de le pré-calculer en chiffrant successivement le vecteur d'initialisation.
2. Il n'est donc sûr que si la fonction de chiffrement alliée à la clé forme une bonne suite pseudo-aléatoire.

5. Schéma général des cryptosystèmes symétriques.

(1) Coder l'information source en binaire. On obtient une chaîne de caractères composée de 0 et de 1.

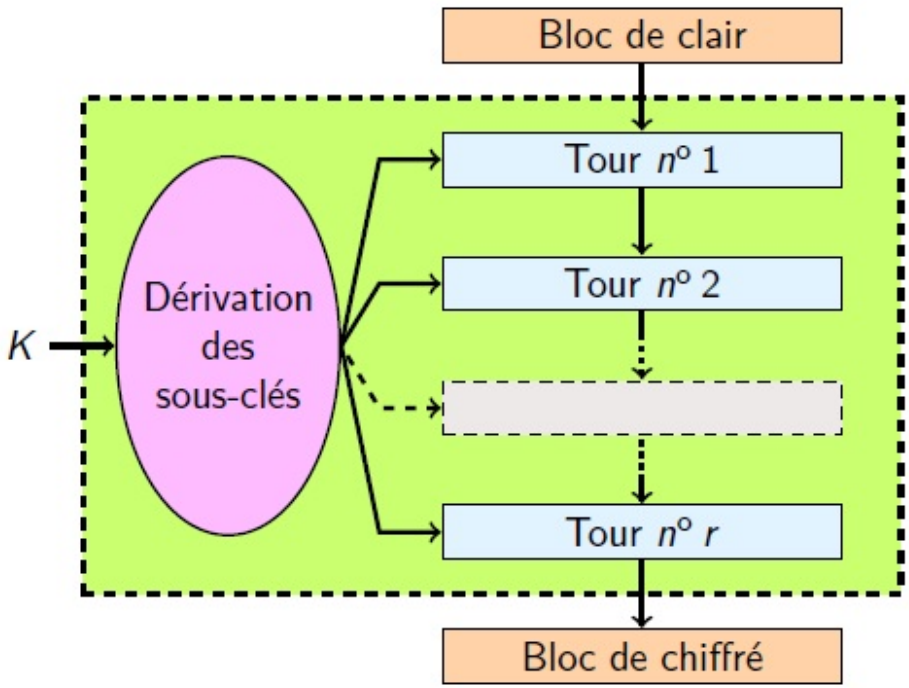
(2) Découper cette chaîne en blocs de longueur donnée (par exemple 64 bits ou 128 bits ou 256 bits).

(3) Appliquer un mode de chiffrement de blocs.

(4) Recommencer un certain nombre de fois l'étape précédente, on appelle cela une ronde.

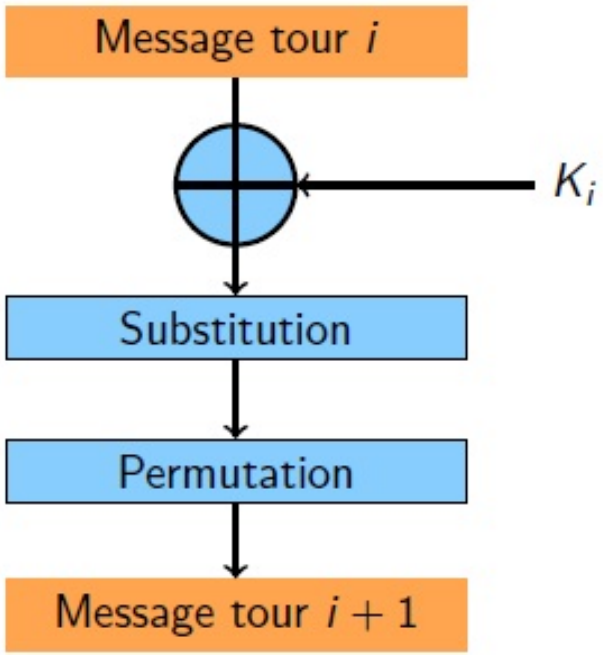
(5) Passer au bloc suivant et retourner à l'étape 3 jusqu'à ce que tous les blocs soient chiffrés.

5. Schéma général des cryptosystèmes symétriques.



6. Chiffrement d'un tour.

6.1. Par substitutions-permutations.



Décomposition d'un bloc :

- On crée une sous clé pour chaque tour.
- Couche de substitution.
- Couche de permutation.

6. Chiffrement d'un tour.

6.2. Par le schéma de Feistel.

- Feistel (IBM, 1973) : permet de construire facilement des algorithmes de chiffrement par Bloc.

Brique de base

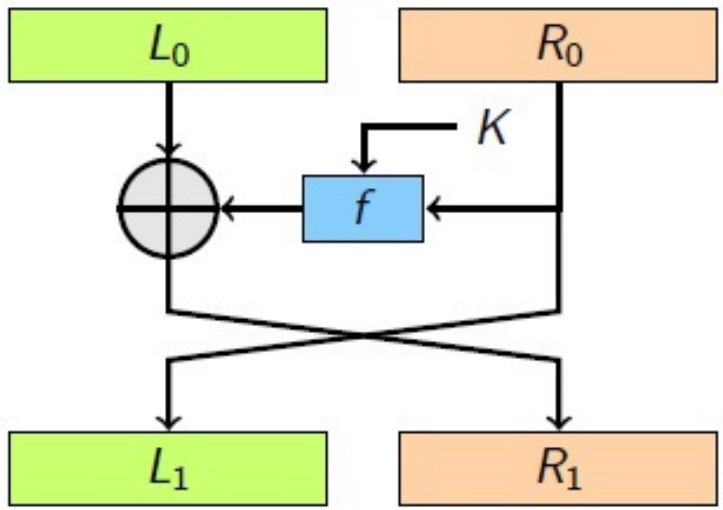
Obtenir une bijection sur $2n$ bits, à partir d'une fonction non-bijective sur n bits

Chiffrement:

$$L_1 = R_0$$
$$R_1 = L_0 \oplus f(R_0)$$

Le déchiffrement est trivial :

$$R_0 = L_1$$
$$L_0 = R_1 \oplus f(R_0)$$



La fonction f est appelée la *fonction de confusion*

- La fonction de chiffrement et la fonction de déchiffrement sont identiques. Ainsi la fonction n'a pas à être inversible, c'est la structure qui l'est.

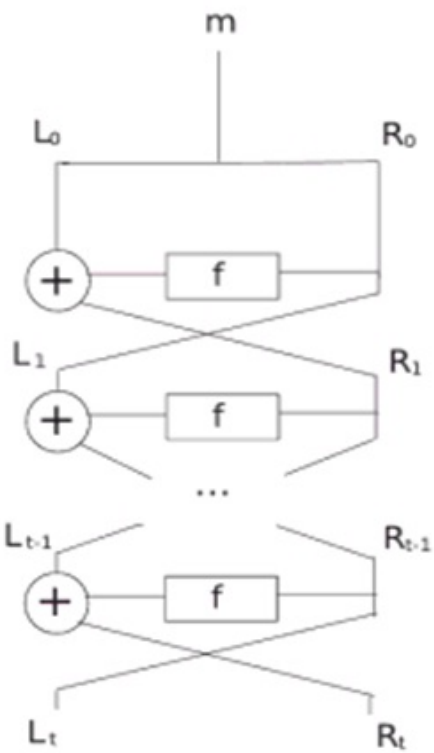
6. Chiffrement d'un tour.

- Feistel est un chiffrement itératif de t tours .
- A partir d'un bloc de clair de 2n bits (L_0, R_0) , on aura en sortie un bloc chiffré (L_t, R_t) de taille 2n.
- A chaque tour, le schéma transforme (L_{i-1}, R_{i-1}) en (L_i, R_i) tels que:
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$

$k_1, k_2, \ldots K_t$: sous clés dérivées de la clé secréte K.

- Pour tout couples (L_i, R_i),on peut trouver le couple (L_{i-1}, R_{i-1}), par les opérations :

$R_{i-1} = L_i$, et $L_{i-1} = R_i \oplus f(L_i, k_i)$.



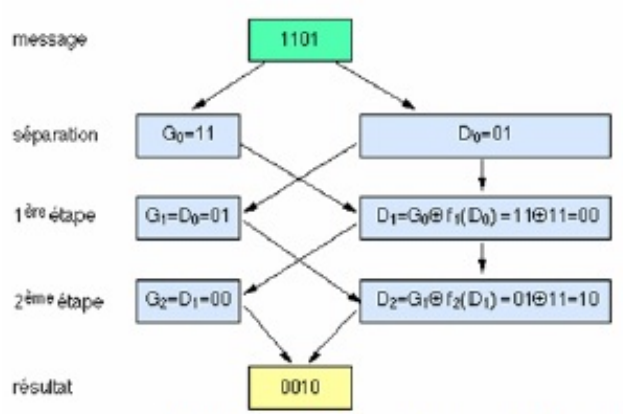
6. Chiffrement d'un tour.

Exemple

On se base sur une table de correspondance pour déterminer le résultat du chiffrement d'un bloc après passage dans une structure de Feistel.

entrée	f_1	sortie	entrée	f_2	sortie
00	→	01	00	→	11
01	→	11	01	→	00
10	→	10	10	→	00
11	→	01	11	→	01

Table de correspondance de fonctions



Exécution d'un schéma de Feistel

message	résultat
0000	0100
0001	1100
0010	1010
0011	0111
0100	0011
0101	1001
0110	1111
0111	0000
1000	1101
1001	0101
1010	0001
1011	1110
1100	1000
1101	0010
1110	0110
1111	1011