Cryptographie à clé secrète

La **cryptographie à clé secrète** utilise pour les échanges entre deux correspondants, une seule clé. Cette clé est utilisée à la fois pour le chiffrement du texte clair et pour le déchiffrement du texte chiffré. Pour cette raison un tel système est aussi appelé un **cryptosystème symmétrique**. La clé, appelée **clé secrète** ne doit être connue que par les deux interlocuteurs. En conséquence, il faut une clé différente par couple de correspondants. **L'échange de la clé secrète** est la principale difficulté de la mise en œuvre de ces systèmes.

La cryptographie ancienne, depuis le système de Jules Cesar, jusqu'à la machine enigma utilisée durant la dernière guerre mondiale de 1939-1945 est une cryptographie à clè secrète.

La cryptographie à clé secrète peut se classer en deux catégories :

1) **les systèmes de chiffrement par bloc**.
2) **les systèmes de chiffrement à flot**.

Un système de chiffrement par bloc utilise une transformation sur des blocs de texte clair de taille fixe, et renvoie des blocs de texte chiffré de taille fixe (en général de la taille du bloc d'entrée). On doit considérer que cette transformation est publique à l'exception de la clé secrète et bien entendu du texte clair auquel elle s'applique.

Un système de chiffrement à flot opère sur les symboles individuels du texte clair par une transformation dépendantt de la clé et de la position du symbole dans le flot de données.

1 Le chiffrement par bloc

1.1 Les principes généraux

1.1.1 Remarque préalable

Il faut considérer que les **systèmes de chiffrement par bloc** constituent les **atomes** (ou encore les **éléments primitifs**) qui constitueront des systèmes plus complexes, les **protocoles**. Ainsi il est important non seulement d'étudier ces atomes en eux mêmes, mais aussi leurs **modes d'utilisation**, c'est-à-dire la manière dont ils vont s'insérer dans un protocole. Il en sera de même des questions de **sécurité** : on devra étudier la sécurité de l'atome en lui même et la sécurité de son mode de fonctionnement à l'intérieur d'un protocole.

1.1.2 Description d'un chiffrement par bloc

Nous noterons $\mathcal{P} = \{0,1\}^{t_d}$ l'ensemble des **blocs de texte clair**, $\mathcal{C} = \{0,1\}^{t_d}$ l'ensemble des **blocs de texte chiffré** et $\mathcal{K} = \{0,1\}^{t_k}$ l'ensemble des **clés**. Un **système de chiffrement par bloc** est la donnée d'une **fonction de chiffrement** (qu'on doit considérer comme étant publique)

$$\mathcal{E} : \quad \mathcal{P} \times \mathcal{K} \to \mathcal{C}$$

qui à un bloc de texte clair $P \in \mathcal{P}$ de taille t_d et à une clé $K \in \mathcal{K}$ de taille t_k fasse correspondre un bloc de texte chiffré $C \in \mathcal{C}$ de taille t_d, et qui possède de plus la propriété suivante :

Pour chaque $K \in \mathcal{K}$ la fonction

$$\mathcal{E}_K : \quad \mathcal{P} \to \mathcal{C}$$

définie par

$$\mathcal{E}_K(P) = \mathcal{E}(P, K)$$

doit être bijective.

Ainsi on peut construire une **fonction de déchiffrement**

$$\mathcal{D} : \quad \mathcal{C} \times \mathcal{K} \to \mathcal{P}$$

telle que pour tout $K \in \mathcal{K}$, la fonction

$$\mathcal{D}_K : \quad \mathcal{C} \to \mathcal{P}$$

définie par

$$\mathcal{D}_K(C) = \mathcal{D}(C, K)$$

soit l'inverse de la fonction $\mathcal{E}_K$.

Bien entendu il est nécessaire de pouvoir calculer **facilement** les fonctions $\mathcal{E}$ et $\mathcal{D}$. Par contre il doit être **pratiquement impossible** de reconstituer la moindre information sur P ou K connaissant C. On cherchera aussi à rendre pratiquement impossible la reconstitution de la moindre information sur K connaissant un couple P, C ou même connaissant de nombreux couples P, C.

1.1.3 La diffusion et la confusion

Pour assurer la sécurité du système, C. Shannon parle de manière un peu informelle de deux principes généraux : la **confusion** et la **diffusion**.

- **La confusion** doit cacher les structures algébriques et statistiques.
- **La diffusion** doit permettre à chaque bit de texte clair d'avoir une influence sur une grande partie du texte chiffré. Ce qui signifie que la modification d'un bit du bloc d'entrée doit entraîner la modification de nombreux bits du bloc de sortie correspondant.

1.1.4 Iteration d'une fonction de tour

Dans les systèmes actuels, le chiffrement est obtenu en itérant une **fonction de tour** qui est **cryptographiquement faible**, c'est-à-dire qui ne constituerait pas à elle seule un système suffisamment robuste. Chaque itération est appelée un **tour** ou encore une **ronde**. Chaque tour prend en entrée la sortie du tour précédent (ou le bloc de texte

clair dans le cas du premier tour) et chiffre cette entrée grâce à la fonction de tour et à la clé de tour construite avec la clé K. Le nombre de tours sera noté N_r.
Ainsi on dispose d'une fonction

$$\mathcal{F} : \quad \{0,1\}^{t_d} \times \{0,1\}^{t_r} \to \{0,1\}^{t_d}$$

où t_r représente la taille de la cé de tour. Alors si P est le texte clair on calcule successivement

$$\begin{aligned} P_0 &= P, \\ P_1 &= \mathcal{F}(P_0, K_1), \\ \cdots &= \cdots\cdots, \\ P_{N_r} &= \mathcal{F}(P_{N_r-1}, K_{N_r}), \\ C &= P_{N_r}. \end{aligned}$$

Le calcul est parfois augmenté d'une phase de prétraitement (avant de commencer les itérations) et d'une phase de post-traitement (après la dernière itération).

1.1.5 Le schema de production des sous-clés

Nous venons de voir que chaque ronde utilise une clé de tour. Cette clé de tour est obtenue à partir de la clé secrète. Ainsi pour mener à bien le calcul itératif que nous venons de décrire, on doit donc associer à la clé secrète K, de taille t_k, N_r **sous-clés** (ou **clés de tour**) de taille t_r. Le procédé qui permet de calculer ces sous-clés est appelé **schéma de diversification de la clé** ou **schéma de production des sous-clés**.

1.1.6 Le modèle de Feistel

Le modèle de Feistel est un exemple de principe de construction de fonction de tour. Il est utilisé dans certains circuits de chiffremment et en particulier dans le DES. Dans le **modéle de Feistel** la taille t_d des données est paire (notons $t_d = 2n$) et chaque donnée intermédiaire P_i

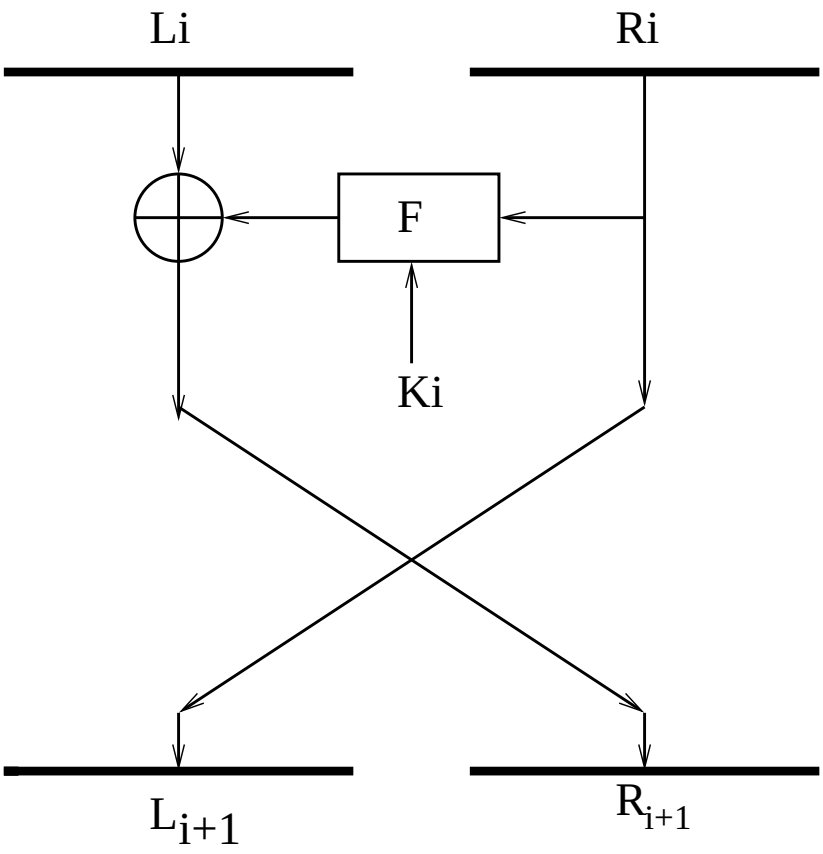

FIG. 1 – **Le modèle Feistel**

est découpée en sa partie droite R_i et sa partie gauche L_i, toutes deux de taille n. On calcule alors R_{i+1} et L_{i+1} de la façon suivante

$$L_{i+1} = R_i, \quad R_{i+1} = L_i \oplus f(R_i, K_i).$$

Dans ce modéle f est évidemment **non-linéaire** et donnée par ce qu'on appelle une **S-box**.

1.2 Les modes principaux d'utilisation

Supposons qu'on ait un texte clair qu'on découpe en plusieurs blocs d'entrée $P = P_1 P_2 \cdots P_k$. Comment va-t-on utiliser le système pour chiffrer ce texte clair ?

- **Electronic Codebook Mode : ECB**
  Le chiffrement. On part du texte clair $P = P_1 P_2 \cdots P_k$, où chaque bloc P_i a la taille exacte t_d de l'entrée du circuit de chiffrement. On

calcule alors pour $i = 1 \cdots k$ les textes chiffrés $C_i = \mathcal{E}_K(P_i)$. Le texte chiffré est $C = C_1 C_2 \cdots C_k$.

Le déchiffrement. On calcule pour $i = 1 \cdots k$ les $P_i = \mathcal{D}_K(C_i)$. On reconstitue ainsi $P = P_1 P_2 \cdots P_k$.

- **Cipher Feedback Mode : CFB**

  Nous aurons besoin d'introduire les notations suivantes : si B est un bloc de bits de taille n et si u est un entier tel que $0 \leq u \leq n$ alors $MSB_u(B)$ représente le bloc constitué par les u bits de B les plus à gauche (les u bits les plus significatifs). La notation $LSB_u(B)$ représente le bloc des u bits de B les plus à droite (les u bits les moins significatifs). Enfin, si B_1 et B_2 sont deux blocs de bits, $B_1||B_2$ représente le bloc formé par la concaténation (la juxtaposition) des deux blocs B_1 et B_2.

  Le chiffrement. Le texte clair est découpé en blocs P_i de taille s où $1 \leq s \leq t_d$. En plus du texte clair $P = P_1 P_2 \cdots P_k$, on abesoin d'un bloc initial de taille t_d tiré aléatoirement I_1. On calcule alors successivement

$$
\begin{aligned}
Z_1 = \mathcal{E}_K(I_1) \qquad & C_1 = P_1 \oplus MSB_s(Z_1), \\
& I_2 = LSB_{t_d - s}(I_1)||C_1, \\
Z_2 = \mathcal{E}_K(I_2) \qquad & C_2 = P_2 \oplus MSB_s(Z_2), \\
\cdots = \cdots\cdots \qquad & \cdots = \cdots\cdots, \\
& I_k = LSB_{t_d - s}(I_{k-1})||C_{k-1}, \\
Z_k = \mathcal{E}_K(I_k) \qquad & C_k = P_k \oplus MSB_s(Z_k).
\end{aligned}
$$

  Le message chiffré est alors $C = I_1 C_1 C_2 \cdots C_k$.

  Le déchiffrement. Il s'effectue en calculant successivement

$$
\begin{aligned}
Z_1 = \mathcal{E}_K(I_1) \qquad & P_1 = C_1 \oplus MSB_s(Z_1), \\
& I_2 = LSB_{t_d - s}(I_1)||C_1, \\
Z_2 = \mathcal{E}_K(I_2) \qquad & P_2 = C_2 \oplus MSB_s(Z_2), \\
\cdots = \cdots\cdots \qquad & \cdots = \cdots\cdots,
\end{aligned}
$$

$$I_k = LSB_{t_d-s}(I_{k-1})||C_{k-1},$$
$$Z_k = \mathcal{E}_K(I_k) \qquad P_k = C_k \oplus MSB_s(Z_k).$$

Remarque.- Si $s = t_d$ alors pour $j \geq 2$ on a $I_j = C_{j-1}$. Donc en posant $C_0 = I_1$ on obtient pour $j \geq 1$ $Z_j = \mathcal{E}_K(C_{j-1})$ et $C_j = P_1 \oplus Z_j$.

- **Cipher Block Chaining Mode : CBC**

  Le chiffrement. Le texte clair est découpé en blocs P_i de taille t_d. En plus du texte clair $P = P_1P_2 \cdots P_k$, on a besoin là encore d'un bloc initial tiré aléatoirement C_0. On calcule alors successivement

$$C_1 = \mathcal{E}_K(C_0 \oplus P_1),$$
$$C_2 = \mathcal{E}_K(C_1 \oplus P_2),$$
$$\cdots = \cdots\cdots,$$
$$C_k = \mathcal{E}_K(C_{k-1} \oplus P_k).$$

  Le message chiffré est alors $C = C_0C_1\mathbb{C}_2 \cdots C_k$.

  Le **déchiffrement**. Il s'effectue en calculant successivement

$$P_1 = \mathcal{D}_K(C_1) \oplus C_0,$$
$$P_2 = \mathcal{D}_K(C_2) \oplus C_1,$$
$$\cdots = \cdots\cdots,$$
$$P_k = \mathcal{D}_K(C_k) \oplus C_{k-1}.$$

- **Output Feedback Mode : OFB**

  Le chiffrement.Le texte clair est découpé en blocs P_i de taille t_d. À partir d'un bloc initial Z_0 tiré aléatoirement, on construit un **masque jetable par itération**, ainsi on a pour $i = 1 \cdots k$ $Z_i = \mathcal{E}_K(Z_{i-1})$, ce qui permet de calculer $C_i = P_i \oplus Z_i$. Le message chiffré est $Z_0C_1C_2 \cdots C_k$.

  Le **déchiffrement** s'effectue en calculant les Z_i successifs à partir de Z_0 comme on l'a fait pour le chiffrement puis en écrivant $P_i = C_i \oplus Z_i$.

- **Counter Mode : CTR**
  Le chiffrement. Le texte clair est découpé en blocs de taille t_d. Dans ce mode on dispose d'un compteur de taille t_d. On tire au sort une valeur initiale CTR_1 pour ce compteur. On calcule alors

$$Z_1 = \mathcal{E}_K(CTR_1) \qquad C_1 = P_1 \oplus Z_1.$$

  Puis on incrémente le compteur, c'est-à-dire qu'on calcule

$$CTR_2 = CTR_1 + 1 \mod 2^{t_d}.$$

  Plus généralement, on calcule successivement

$$CTR_i = CTR_{i-1} + 1 \mod 2^{t_d},$$

$$Z_i = \mathcal{E}_K(CTR_i),$$

$$C_i = P_i \oplus Z_i.$$

  Le texte chiffré transmis est constitué de la valeur CTR_1 du compteur suivi des blocs $C_1C_2 \cdots C_k$.
  Le déchiffrement. Il se fait en calculant succéssivement à partir de la valeur initiale du compteur

$$Z_i = \mathcal{E}_K(CTR_i),$$

$$P_i = C_i \oplus Z_i,$$

$$CTR_{i+1} = CTR_i + 1 \mod 2^{t_d}.$$

- **Remarques sur ces différents modes**
  Le mode **ECB** possède l'inconvénient suivant : si dans un texte clair, deux blocs sont identiques, ils seront chiffrés de la même façon, et le même motif apparaitra dans le texte chiffré. Il est donc nécessaire que les tailles des blocs soient suffisamment grandes pour éviter une attaque statistique sur la forme des motifs. Cependant ceci est un avantage si on veut faire un accés aléatoire aux blocs chiffrés (par exemple si on veut accéder à une partie bien précise d'un fichier

et éventuellement la modifier sans toucher au reste du fichier). Ce mode est considéré comme peu sûr.
Dans le mode **CFB**, on n'utilise pas la fonction $\mathcal{D}_K$. Ceci peut être un avantage si le système utilisé est plus rapide en chiffrement qu'en déchiffrement (cas du système AES). Le texte chiffré est un peu plus long que le texte clair à cause du bloc I_1 qui doit aussi être transmis. La modification d'un bloc a des répercussions sur d'autres blocs. On ne peut donc pas faire d'accés aléatoire.
Dans le mode **CBC** le texte chiffré est encore un peu plus long que le texte clair à cause du C_0. Là encore la modification d'un bloc a des répercussions sur d'autres blocs.
Dans le mode **OFB** le texte chiffré est encore un peu plus long que le texte clair à cause du Z_0. La fonction de déchiffrement $\mathcal{D}_K$ n'est pas utilisée. La modification d'un P_i ne modifie que le C_i correspondant. On peut donc faire des accés aléatoires. Cependant c'est moins commode que dans le cas du ECB puisqu'il faut calculer par itération tout le masque jusqu'au bloc concerné.
Le mode **CTR** est un mode très simple à implémenter. Il est très sûr et n'utilise pas la fonction de déchiffrement du circuit. Le texte chiffré, qui doit contenir la valeur initiale du compteur, est un peu plus long que le texte clair. On peut dans ce mode faire des accès aléatoires de manière plus commode que dans le mode OFB puisqu'il faut seulement chiffrer la valeur du compteur pour le bloc considéré. Ce mode est particulièrement intéressant.