Chapitre 3 : Structure des chiffrements symétriques

9 juin 2023

Sommaire

Introduction

1. Chiffrement par flux
2. Chiffrement par blocs
   - Structure basique des chiffrements par blocs
   - La structure Feistel et le chiffrement DES
   - Le schéma substitution permutation (SPN) et l'AES
3. Mode d'opérations
4. Implémentation : Mini-Projet
5. Conclusion

Definition

Les schémas de chiffrement à clé symétrique, aussi appelé codes secrets, sont assez souvent classés comme étant

1. un chiffrement par flot dans lequel le chiffrement s'effectue un caractère (ou un bit) à la fois, ou
2. un chiffrement par bloc dans lequel le chiffrement s'effectue sur un bloc de caractères (ou bits).

- **Le chiffrement par bloc** est utilisé (en général) lorsque toutes les données sont disponibles avant le commencement du processus de chiffrement : chiffrement d'un email (**GnuPG**), chiffrement d'un fichier dans une machine, archivage des données à long terme.
- **Le chiffrement par flot** est utilisé (en général) pour le chiffrement d'une communication à temps réel : chiffrement de la communication entre un utilisateur et un serveur (**http/SSL**), chiffrement de la communication entre deux parties utilisant un téléphone, ...

Chiffrement par flux (généralisé)

Le principe basique derrière le chiffrement par flot est assez simple.

Definition ((One-time pad))

Soit z_t, pour $t \geq 0$, une clé donnée par une séquence binaire aléatoire

- l'expéditeur veut envoyer une séquence de message binaire m_t.
- la séquence chiffré de bits se calcule par $c_t = m_t \oplus z_t$
- le destinataire connaissant z_t, calcule $m_t = c_t \oplus z_t$

Il reste incassable sous l'hypothèse qu'une clé est utilisée une seule fois par chiffrement. Le problème principal avec le one-time pad est donc

- la séquence de clé est aussi longue que le message et
- pour chaque chiffrement nous avons besoin d'une nouvelle clé aléatoire.

Ce qui crée de sérieux problèmes de gestion et de distribution de clés. **Une remède** est d'utiliser un **PSG** (Pseudorandom Sequence Generator).

Principe des algorithmes de chiffrement à flux

Dans la cryptographie à chiffrement par flot

- une séquence pseudo-aléatoire de bits de longueur égale à la taille du message est générée par un *générateur de séquence pseudo-aléatoire (PSG : pseudorandom sequence generator)*.
- Cette séquence est ensuite additionnée (modulo 2) bit par bit au message et la séquence qui en résulte sera transmise.
- le déchiffrement s'effectue en générant d'abord la même séquence pseudo-aléatoire, puis on additionne (modulo 2) bit par bit le chiffré avec cette séquence.

Définition

Une graine ou germe aléatoire est un nombre ou une séquence utilisée pour l'initialisation d'un générateur pseudo-aléatoire.

La graine pour le générateur de bit pseudo-aléatoire, dans un chiffrement par flot, représente la clé secrete.

Definition

Un PSG est un algorithme déterministe commençant par une chaine raisonnablement courte de bits (appelée graine) et l'étend en une chaine binaire trés longue utilisée en tant que clé (suite chiffrante). La graine est la clé secréte (de taille par exemple 80 ou 128 bits) partagée entre les deux parties communicantes (l'expéditeur et le destinataire).

Sécurité

Un PSG est sécurisé

- si l'attanquant dispose d'un segment des bits générés, alors il lui est difficile de prédire le prochain bit,
- autrement dit, il doit être calculatoirement difficile de distinguer la suite pseudo-aléatoire générée à une suite aléatoire.

La plus part des chiffrements à flot utilise un vecteur d'initialisation (initialization vector (**IV**)). Ce vecteur n'est pas tenu secret.

- Le PSG se lance à partir de la paire (clé, IV) représentant la graine.
- La même clé peut être utilisée avec des IVs différents.

La contraine sur l'usage de ce protocole est que la paire (clé, IV) ne doit pas être répétée.

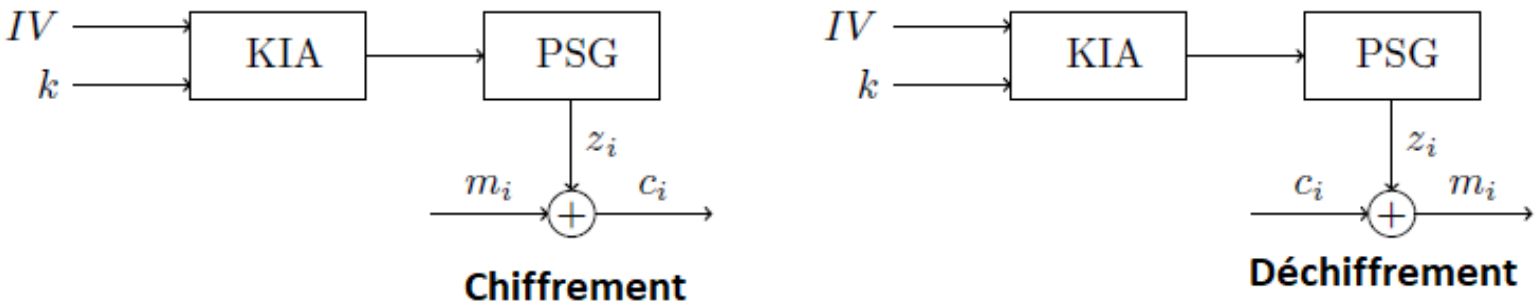
Definition (Hypothéses de sécurité)

Il est supposé que l'adversaire connait tout excepté la clé secrète. Il s'agit du principe de Kerckhoff's :

1. Les algorithmes dans le chiffrement à flot sont publiques.
2. L'unique information secrète dans le système est la clé partagée.
3. Un attaquant peut intercepter des communications (textes chiffrés).

Les deux phases dans un chiffrement à flot

1. la phase d'initialisation de clé, pour laquelle l'algorithme est appelé **KIA** (Key Initialization Algorithm),
2. la phase d'exécution du PSG.



Le vecteur d'initialisation (IV) est une information publique, et k, est une clé de chiffrement partagée. L'objectif du KIA est de mêler les bits de clé avec le IV afin d'obtenir une fonction non linèaire complexe de k et IV. Le KIA s'exécute une seule fois pour chaque session de chiffrement.

Chiffrement par blocs (généralisé)

Dans le chiffrement par blocs, les bits de message sont divisés par blocs et chaque bloc est séparément pris comme entrée d'un système de chiffrement (par exemple d'une permutation), en utilisant une même clé.

Definition

Un code par bloc consiste en une famille de fonctions de chiffrement

$$E_k : \{0,\ 1\}^n \longrightarrow \{0,\ 1\}^n$$

paramétrisées par une ℓ-bit clé k. Chaque fonction dans la famille est inversible. L'inverse de E_k est la fonction de déchiffrement D_k.

Si deux parties veulent communiquer de manière sécurisée :

1. Ils s'accordent d'abord sur une clé secrète $k \in \{0,\ 1\}^\ell$.
2. Pour transmettre $m \in \{0,\ 1\}^n$, une partie calcule $c = E_k(m)$.
3. Le destinataire calcule $m = D_k(c)$.

Méthode de chiffrement à blocs

Un code par bloc consiste en une paire d'opération de chiffrement et de déchiffrement E et D.

- Pour une clef fixée K, E associe $m = (m_0, \cdots, m_{n-1})$ à un texte chiffré $c = (c_0, \cdots, c_{n-1})$ de longueur n,
- D associe le texte chiffré au texte clair originel.

$$E : \quad \mathcal{K} \times \mathbf{F}_2^n \quad \longrightarrow \quad \mathbf{F}_2^n, \quad (D \circ E)(m) = m, \quad \mathbf{F}_2 = \{0, 1\}$$

L'opérateur de chiffrement E est constitué de plusieurs tours qui sont appliqués au texte clair l'un après l'autre.

- La clef secrète K est étendue, en utilisant un **key schedule** (un elagorithme de préparation de clés), en un ensemble de clés de tours $K_1, \cdots, K_r$.
- Chaque fonction de tour prend en entrée une clé de tour et la sortie du tour précédent.

Si le message clair est de taille plus grande que n on le découpe en des blocs, $B_0, B_1, \cdots$ de taille n. Puis pour chaque $i = 0, 1, 2, \cdots$, on applique le chiffrement. C'est une des méthodes opératoires sur les blocs.

Pour un texte claire M

- soit $M_0 = M$ et M_{i-1} l'entrée au $i-$ème tour.
- $M_r = C$ la sortie finale de $E_K(M)$.
- nous dénotons par R_i la $i-$ème fonction de tour, alors nous avons $M_i = R_i(K_i, M_{i-1})$.

Ceci réduit la tache de design d'un système de chiffrement par bloc à la tache de design de fonctions de tour et d'un algorithme de génération de clé. On note deux méthodes standards de design de fonctions de tours :

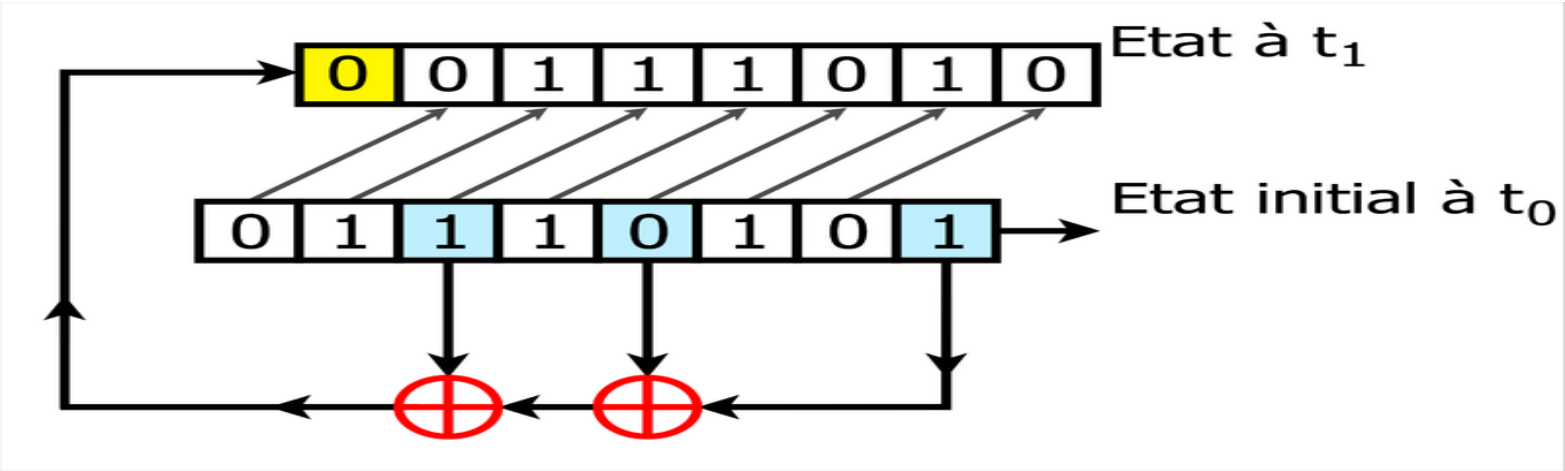
1. La structure de Feistel et
2. Le schéma de substitution permutation (SPN : Substitution-Permutation Network).

Nous considérons

- **DES** comme exemple d'un chiffrement par bloc basé sur la structure de Feistel.
- **AES Rijndael** comme exemple de chiffrement par bloc basé sur la structure SPN.

Registre à décalage avec rétroaction (Feedback shift register)

Un registre à décalage est un ensemble de bascules synchrones. À chaque cycle d'horloge, le nombre représenté par ces bascules est mis à jour. Le concept de décalage permet d'insérer une donnée dans le registre, ou la lire, bit par bit en série.



Si le bit entrant est le résultant d'un XOR avec certains bits du régistre alors on parle de régistre à **rétroaction** linéaire.

La structure Feistel

Définition

La structure de Feistel est un régistre à décalage avec rétroaction dont la fonction de rétroaction varie avec le temps. Autrement dit, la fonction de tour est une fonction de rétroaction variant avec le temps.

Notations

- Soit $\mathcal{F}_n$ l'ensemble des fonctions de $\{0,1\}^n$ dans $\{0,1\}^n$.
- Si L et R sont deux éléments de $\{0,1\}^n$, nous notons $L||R$ l'élément de $\{0,1\}^{2n}$ qui est la concaténation de L et R.

Définition

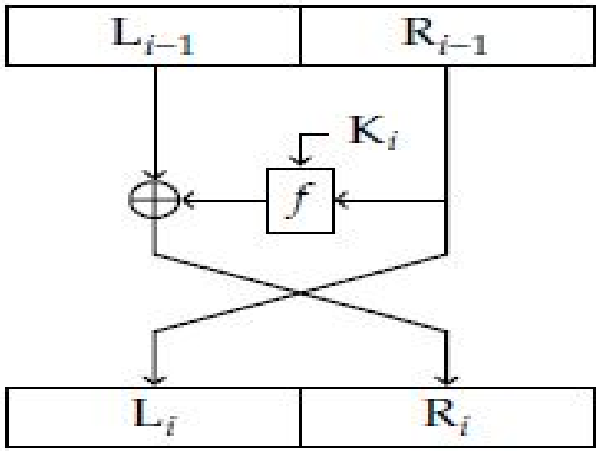
Soit $f \in \mathcal{F}_n$. La fonction de la structure de Feistel associée à f, notée, $\Psi(f)$, est la fonction de $\{0,1\}^{2n}$ dans $\{0,1\}^{2n}$ définie par

$$\forall (L,R) \in (\{0,1\}^n)^2, \Psi(f)(L||R) = S||T \iff S = R \text{ et } T = L \oplus f(R)$$

Les tours de la structure de Feistel

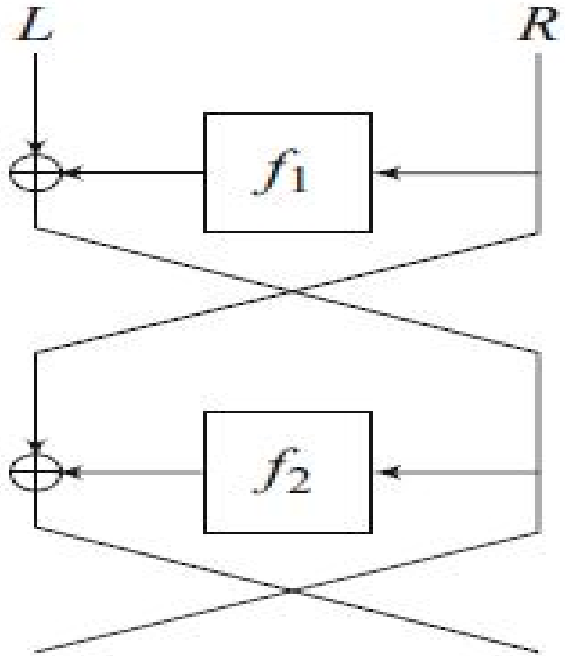
Cette construction a été introduite par Feistel dans les années 70.

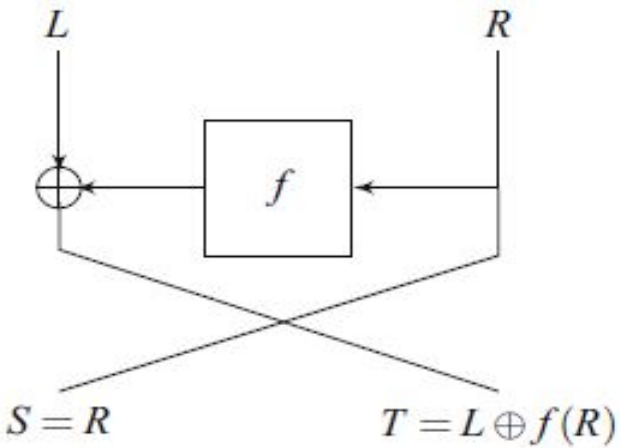
- Á chaque tour $i = 1, 2, \cdots, r$, on prend en entrée un bloc (L_{i-1}, R_{i-1}) et on le transforme en un bloc (L_i, R_i) en faisant intervenir la clef de tour K_i.
- On note f une fonction prenant en entrée et en sortie des blocs de n bits.

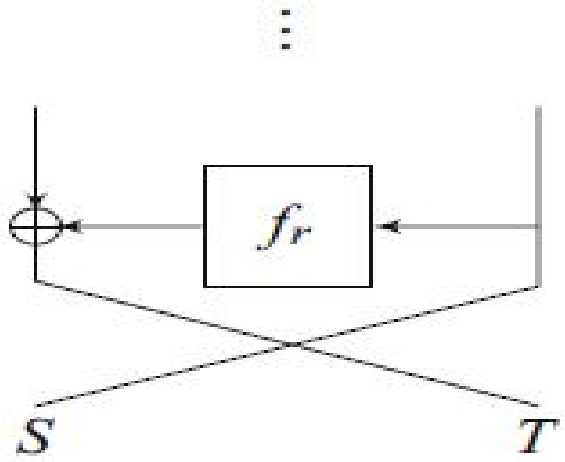


Au bout de r tours, le chiffré est $c = R_r||L_r = S||T$.

Les rounds dans la structure de Feistel

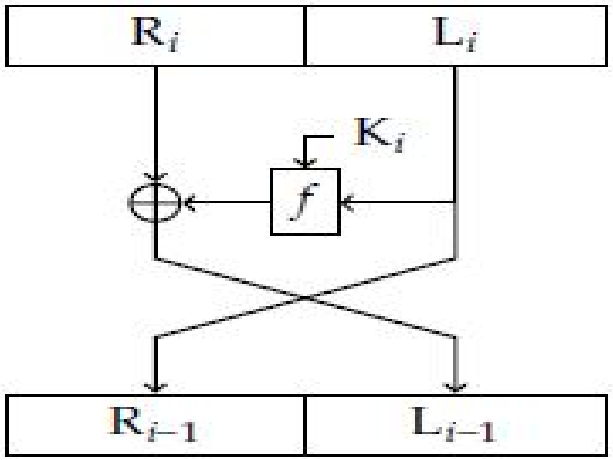






Inverse de la structure de Feistel

Ce schéma est inversible, (si l'on connaît les clefs de tours), que f soit une bijection ou non. En effet, on a $R_{i-1} = L_i$, et $L_{i-1} = R_i \oplus f(R_{i-1}, K_i)$



Déchiffrement sous la structure de Feistel

Le déchiffrement de $c = R_r||L_r$ se fait donc avec exactement le même procédé que le chiffrement en appliquant les clefs de tours dans l'ordre inverse. On retrouve bien, à la dernière étape $L_0||R_0$.

Exercice

Déterminer le nombre minimal de tours qu'il faut pour travailler avec la structure de Feistel. Justifier votre réponse.

DES : Data Encryption Standard

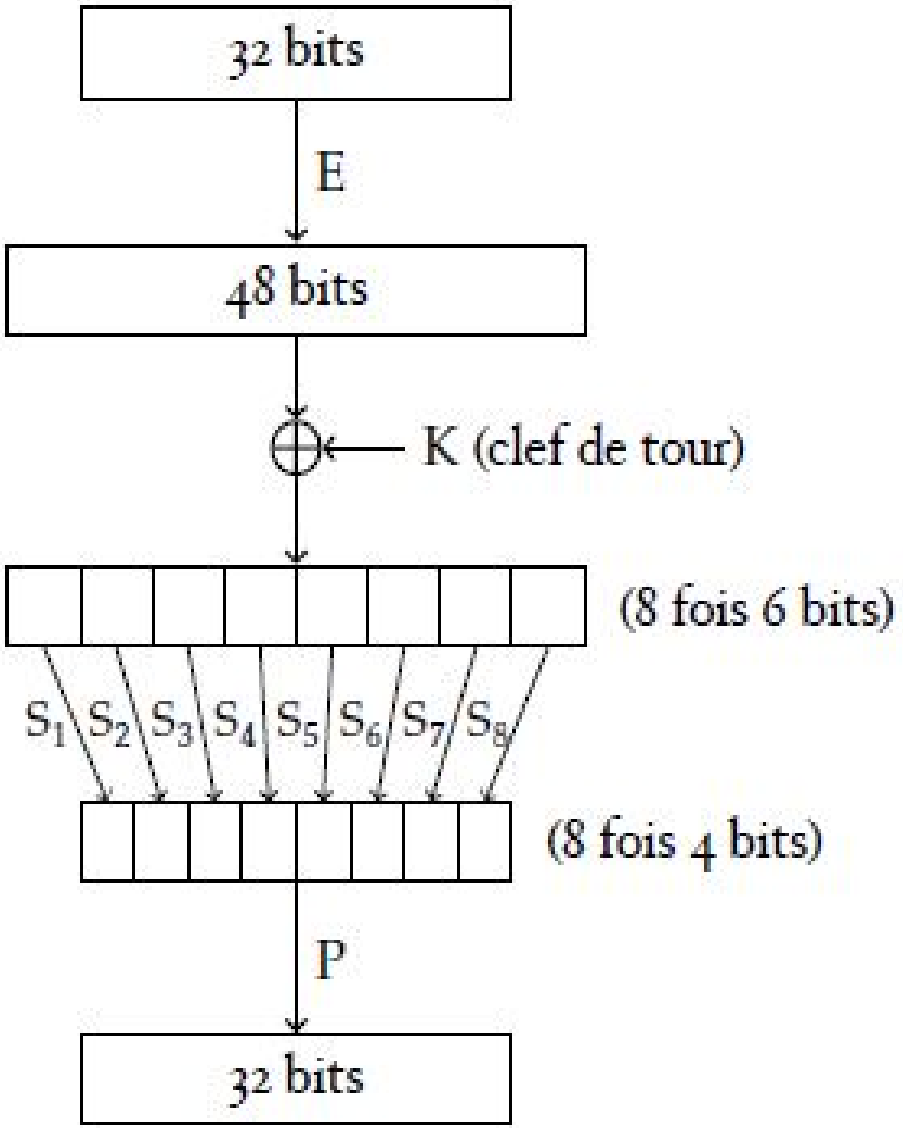
Le DES est l'un des exemples les plus connus d'un chiffrement utilisant un schéma de Feistel. Il s'agit d'un standard de chiffrement par bloc de 1977 à 2000.

- Les blocs de clair et chiffré sont de 64 bits,
- la clef secrète de 56 bits et les clefs de tour de 48 bits.
- On effectue 16 tours de schéma de Feistel avec des permutations initiale et finale.
- La fonction de tour opère sur des mots de 32 bits.

Le DES a été largement déployé dans des applications commerciales.

DES : Etude de la clef de tour

La clef de tour est la composition de plusieurs fonctions, suivant le schéma.



DES : Etude de la clef de tour

- La fonction d'expansion E est linéaire de $\mathbf{F}_2^{32} \longrightarrow \mathbf{F}_2^{48}$. On répète certains bits.

- La fonction P est une permutation des 32 bits.

Définition

Les deux fonctions, E et P apportent de la diffusion c'est-à-dire que si l'on change un bit en entrée, il y aura plusieurs bits changés en sortie.

La diffusion permet d'empêcher les attaques statistiques. Par **exemple** si un bit du clair est souvent égal à 1, cela ne doit pas se voir sur la distribution des chiffrés correspondants.

DES : Etude de la clef de tour

Définition

Les fonctions $S_1, \cdots, S_8$ sont appelées boîtes S ou S-box. Ce sont des fonctions booléennes vectorielles, ici de $\mathbf{F}_2^6 \to \mathbf{F}_2^4$, non linéaires. Elles permettent d'apporter de la confusion.

Le but de la confusion est de rendre complexe les relations entre bits de chiffré et bits de clef.

Historique : Les concepts de diffusion et confusion ont été introduits par Shannon en 1949.

Sécurité

En pratique la meilleure attaque reste la recherche exhaustive qui est maintenant largement faisable. En effet l'ensemble des clés possibles a un cardinal qui est égal à seulement 2^{56}.

Le Triple DES

Pour palier à la faiblesse du DES due à sa clef de 56 bits trop courte, on utilise (dans le monde bancaire) une variante utilisant 3 clefs DES $k_1,\ k_2,\ k_3$ appelée Triple DES.

Astuce

Cela consiste à composer trois fois le DES de la manière suivante : $c = k_3(k_2^{-1}(k_1(m)))$ autrement dit

$$c = Chiffrement_{k_3}(Dechiffrement_{k_2}(Chiffrement_{k_1}(m)))$$

avec trois options sur le choix des clefs

1. trois clefs distinctes, ce qui revient à 168 bits de clef ;
2. $k_1 = k_3$ et $k_1 \neq k_2$, ce qui revient à 112 bits de clef ;
3. $k_1 = k_2 = k_3$ une seule clef de 56 bits, on a un chiffrement classique du DES pour garantir la compatibilité. Ce qui explique pourquoi on alterne chiffrement et déchiffrement.

Inconvénient et sécurité du triple DES.

Inconvénient

Puisque 3-DES utilise trois fois la clé de DES alors ce système de chiffrement est trois fois plus lent que le DES.

Exercice : Cryptanalyse DES

L'attaque **meet in the middle** est une attaque cryptographique générique contre des schémas de chiffrement qui reposent sur l'exécution de plusieurs opérations de chiffrement en séquence (elle justifie la non utilisation du 2-DES en pratique).

1. Comment utiliser cette attaque contre le double DES ?
2. Peut t on l'utiliser contre le triple DES ?

Le schéma substitution permutation (SPN) et l'AES

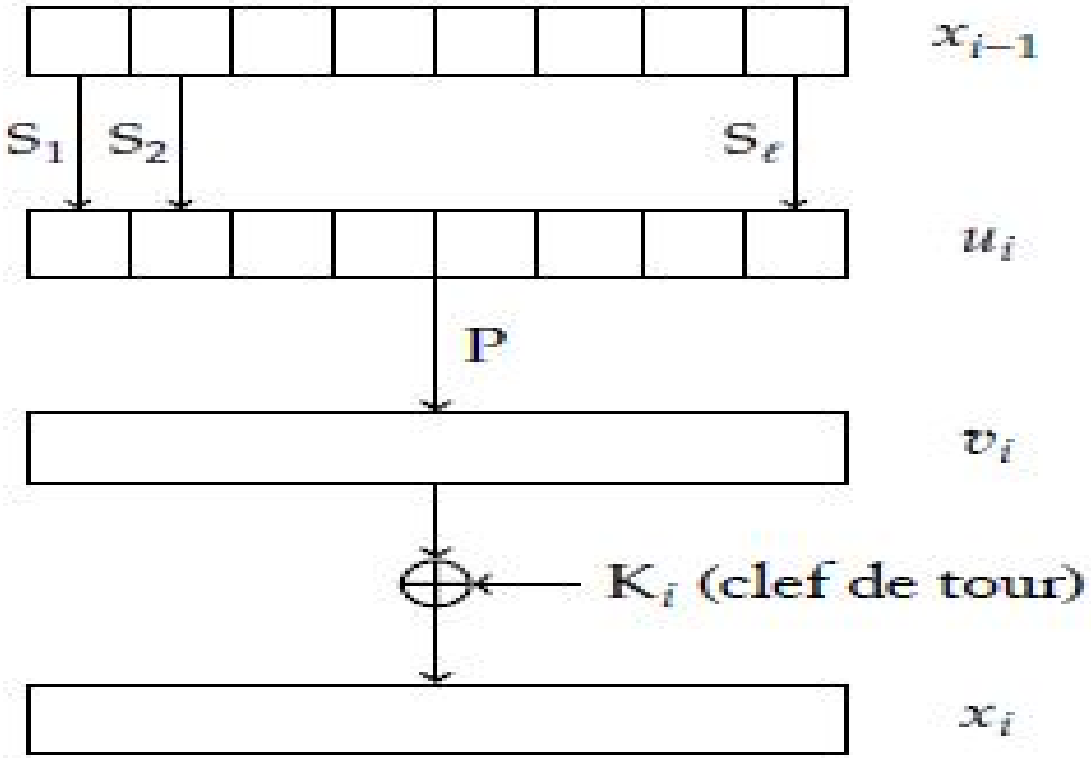
Substitution-Permutation Network (SPN)

C'est une autre construction itérative de chiffrement par bloc. On utilise maintenant une fonction de tour bijective en alternant les opérations de diffusion et de confusion.

- **Initialisation :** On effectue une étape initiale d'ajout bit à bit de la première clef de tour $x_0 = m + K_0$.
- **Chiffrement :** Pour $i = 1, \cdots, r$ on calcul $x_i = F_{K_i}(x_{i-1})$ pour obtenir un chiffré c_r.
- **Confusion :** La fonction f commence par l'application de ℓ boîtes S bijectives sur x_{i-1} découpé en ℓ sous blocs, pour donner un nouveau bloc u_i.
- **Diffusion :** On applique une permutation P sur les bits de u_i, on note v_i le résultat. Enfin, on ajoute la clef de tour : $x_i = v_i + K_i$.

Schéma SPN : illustration

Dans la structure SPN, chaque fonction de tour consiste en un petit nombre de couches successives. L'entière couche de substitution est elle même une application bijective.



L'étape initiale évite qu'un attaquant puisse calculer le début du chiffrement jusqu'à l'ajout de clef K_1.

Schéma SPN : Déchiffrement

Remarque

L'effet d'une couche de substitution est locale au sens où un bit de sortie, à une position particulière, dépend uniquement d'un petit nombre de bits d'entrée. Cet effet local est compensé par la couche de permutation qui permute ses bits d'entrées. La clef de tour est assez souvent incorporée au début ou à la fin de la fonction de tour.

Déchiffrement

Le déchiffrement se fait en "remontant" tout le chiffrement, toutes les opérations étant inversibles.

Exemple de structure SPN : l'AES

Exemple de structure SPN : l'AES

Le standard AES (Advance Encryption Standard), pour remplacé le DES, est issu d'un concours qui s'est déroulé de 1997 à 2001. Le vainqueur a été l'algorithme Rijndael conçu par Joan Daemen et Vincent Rijmen. Il existe des différences entre Rijndael et AES.

- Rijndael donne plusieurs choix de blocs et de taille de clé.
- AES adopte un unique sous ensemble sur les choix qui ont été proposés.

L'adoption du Rijndael en tant que chiffrement standard constitue un tounant décisif dans l'histoire de la cryptographie.

Choix des paramétres de AES

1. Il existe 3 longueurs de blocs permissibles : 128, 192, et 256 bits.
2. Il existe 3 longueurs de clef permissibles : 128, 192, et 256 bits.
3. Le nombre de tours est 10, 12, ou 14, et ceci est fonction de la longueur des clés.
4. Chaque tour est constitué de 3 fonctions, qui se dessinent en 4 couches comme suit :
   1. Une 8−bit permutation inverse (la transformation **sub-byte**),
   2. Une 32−bit transformation linèaire (l'opération **mix columns**),
   3. Une 128−bit permutation (l'opération **shift rows**) et
   4. Une addition d'une clé de tour.

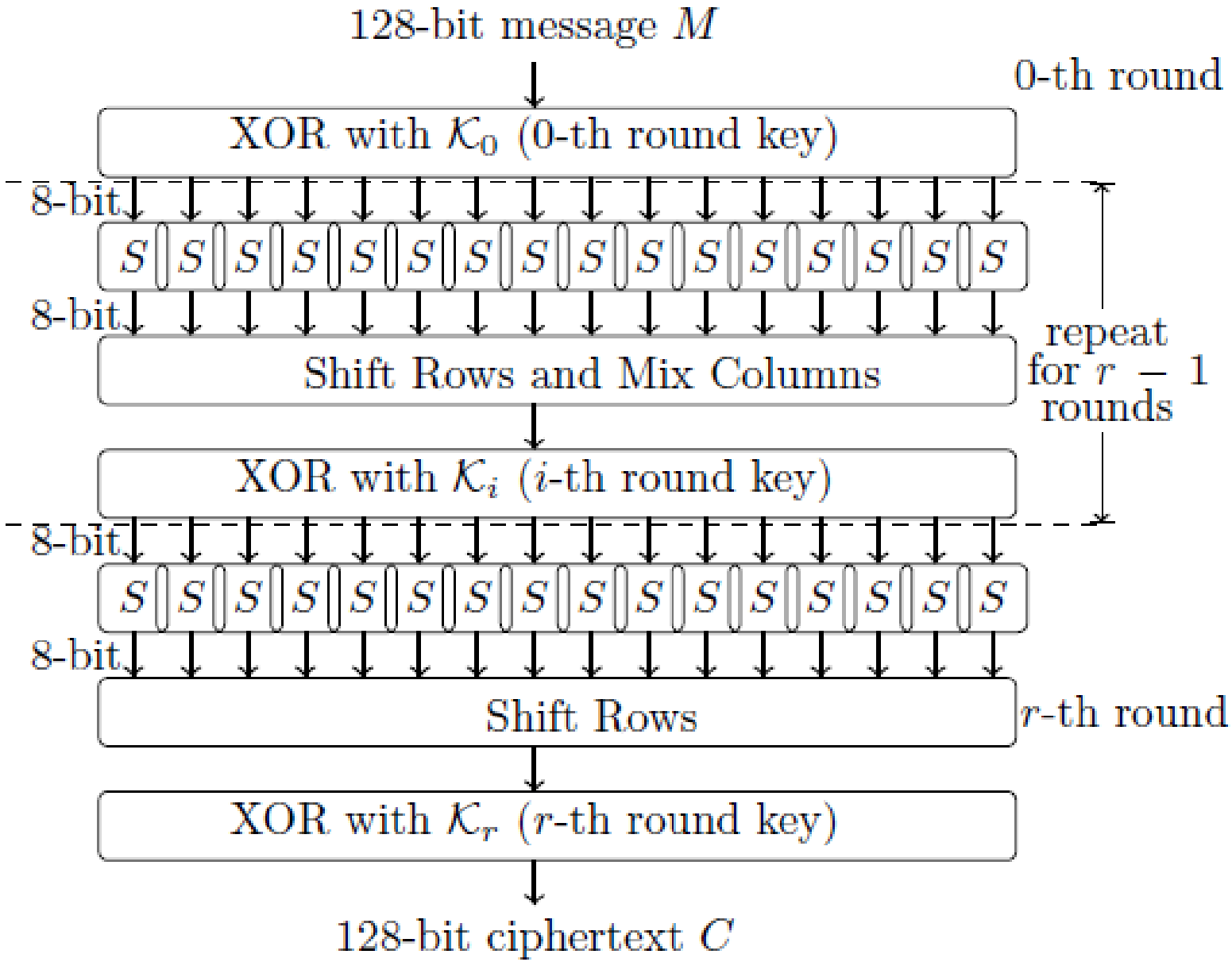
Description du fonctionnement de AES

Un texte clair m de 128 bits est l'état initial qui est représenté par des groupes de 4 par 4 octets.

1. Effectuer une opération **AddRoundKey** qui est un xor de la clé de tour avec l'état initial m.
2. Pour chacune des $r-1$ tours effectuer
   - une opération **SubByte** sur un état en utilisant un $S-$box ;
   - une opération **ShiftRows** sur l'état ;
   - une opération appelée **MixColumns** sur l'état ;
   - une opération **AddRoundKey**.
3. Pour le $r-$ème tour, effectuer un SubByte ; un ShiftRows et une opération AddRoundKey.
4. L'état final y est le texte chiffré.

Tout le déroulement de l'AES suit le schéma général d'un SPN, hormis le tour final qui n'inclut pas la fonction MixColumns. Pour la structure algébrique de ces différentes opérations voir la section implémentation.

Les opérations de tours pour AES Rijndael



Mode d'opérations

Méthodes d'utilisation des blocs

Pour le chiffrement des blocs d'un schéma symétrique on peut rencontrer quatre modes d'utilisation :

1. Le mode ECB (Electronic Code Book)
2. Le mode CBC (Cipher Block Chaining)
3. Le mode CFB (Cipher FeedBack)
4. Le mode OFB (Output FeedBack)

Le mode ECB

Un algorithme de chiffrement par bloc prend en entrée un message clair de n bits et donne en sortie un chiffré, qui est un autre bloc de bits (en général également n).

Définition

Les blocs sont chiffrés indépendamment les uns des autres. Ce mode est appelé mode ECB (Electronic Code Book).

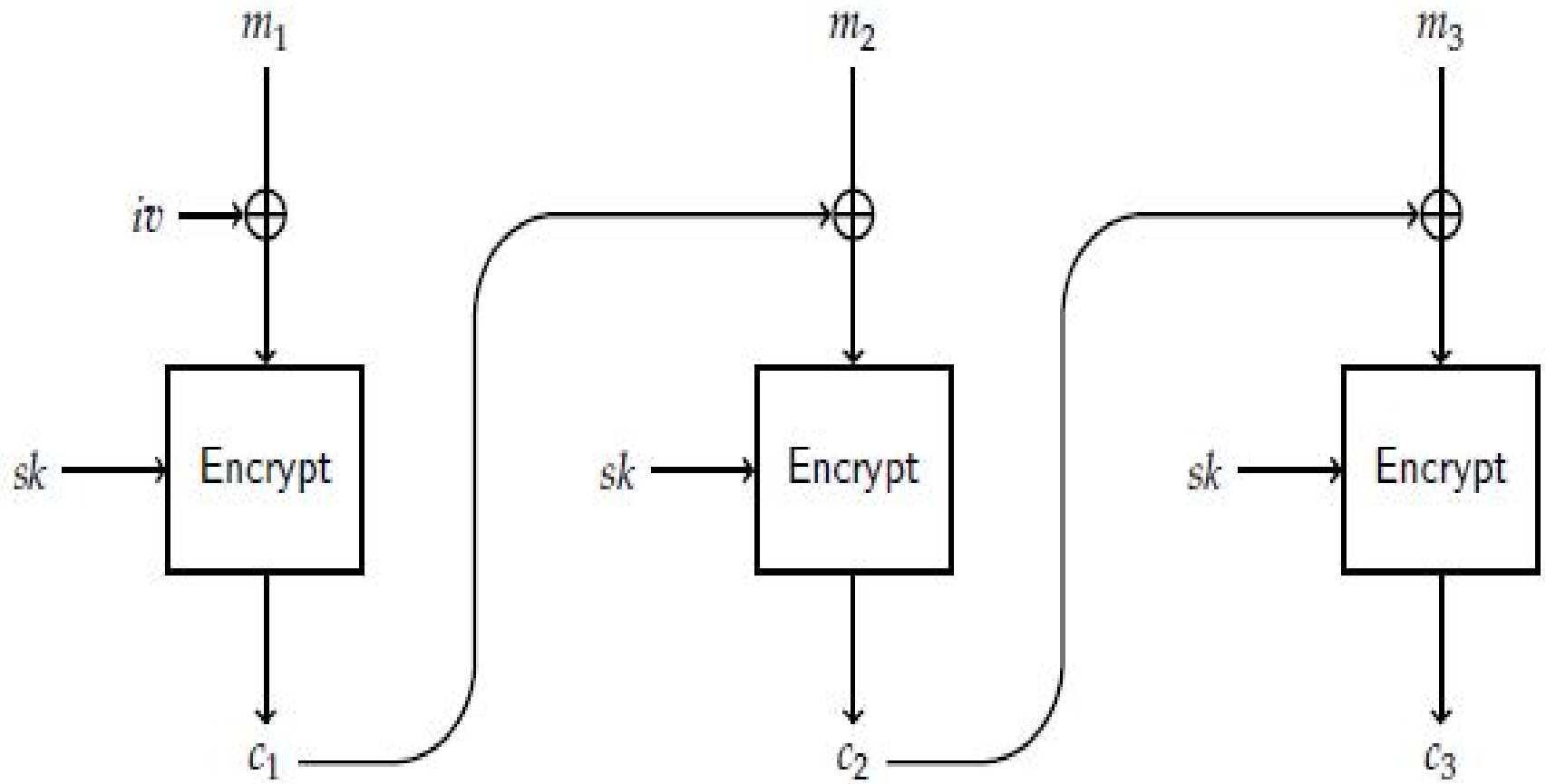
Ce mode ne cache pas les redondances éventuelles du texte clair. Par exemple si $m_i = m_j$ pour $i \neq j$ alors $c_i = c_j$.

Mode CBC

Afin de pallier aux faiblesses du mode ECB, on pourra utiliser le mode CBC (Cipher Block Chaining). Dans ce mode, avant de passer un bloc à la "moulinette", celui-ci est additionné modulo 2 au résultat du chiffrement du bloc précédent. Les blocs ne sont donc plus indépendants les un des autres. Le premier bloc n'ayant pas de précédent on lui en attribue un, le vecteur IV (Initialisation Vector). Pour $B_1,\ B_2, \cdots, B_i, \cdots B_n$ des blocs différents, le schéma obtenu est le suivant :

- Effectuer $IV \oplus B_1$, chiffrer le résultat. On obtient C_1.
- Effectuer $C_1 \oplus B_2$, chiffrer le résultat. On obtient C_2.
- .....
- Effectuer $C_{i-1} \oplus B_i$, chiffrer le résultat. On obtient C_i.
- .....

Mode CBC : Illustration



Pour déchiffrer on effectuera les opérations dans l'ordre inverse.

Le mode CBC

Le mode CBC présente deux inconvénients

1. il impose de déchiffrer la totalité du fichier avant de pouvoir accéder à une partie du fichier.

2. une erreur sur un bloc va se répercuter sur tous les suivants. Avec ce mode de chiffrement on sera très attentif au contrôle d'erreurs.

Les modes OFB et CFB

Dans les modes OFB et CFB, une séquence de clefs est engendrée pour agir sur le texte clair par un ou exclusif. Cela fonctionne comme les chiffrements en chaine. On peut utiliser ces modes sur des blocs de taille k.

- Les blocs de 1 sont utilisés pour chiffrer les données par bit.
- Les blocs (par exemple de 8) sont souvent utilisés pour chiffrer des données par octet.

Le mode OFB

OFB est en fait un chiffrement en chaine synchrone : la séquence de clef est produite par les chiffrements itérés d'un bloc initial IV de k bits (par exemple 64 bits).

- On définit $z_0 = IV$.
- On calcule la séquence de clefs $z_1 z_2 \cdots$ par la formule

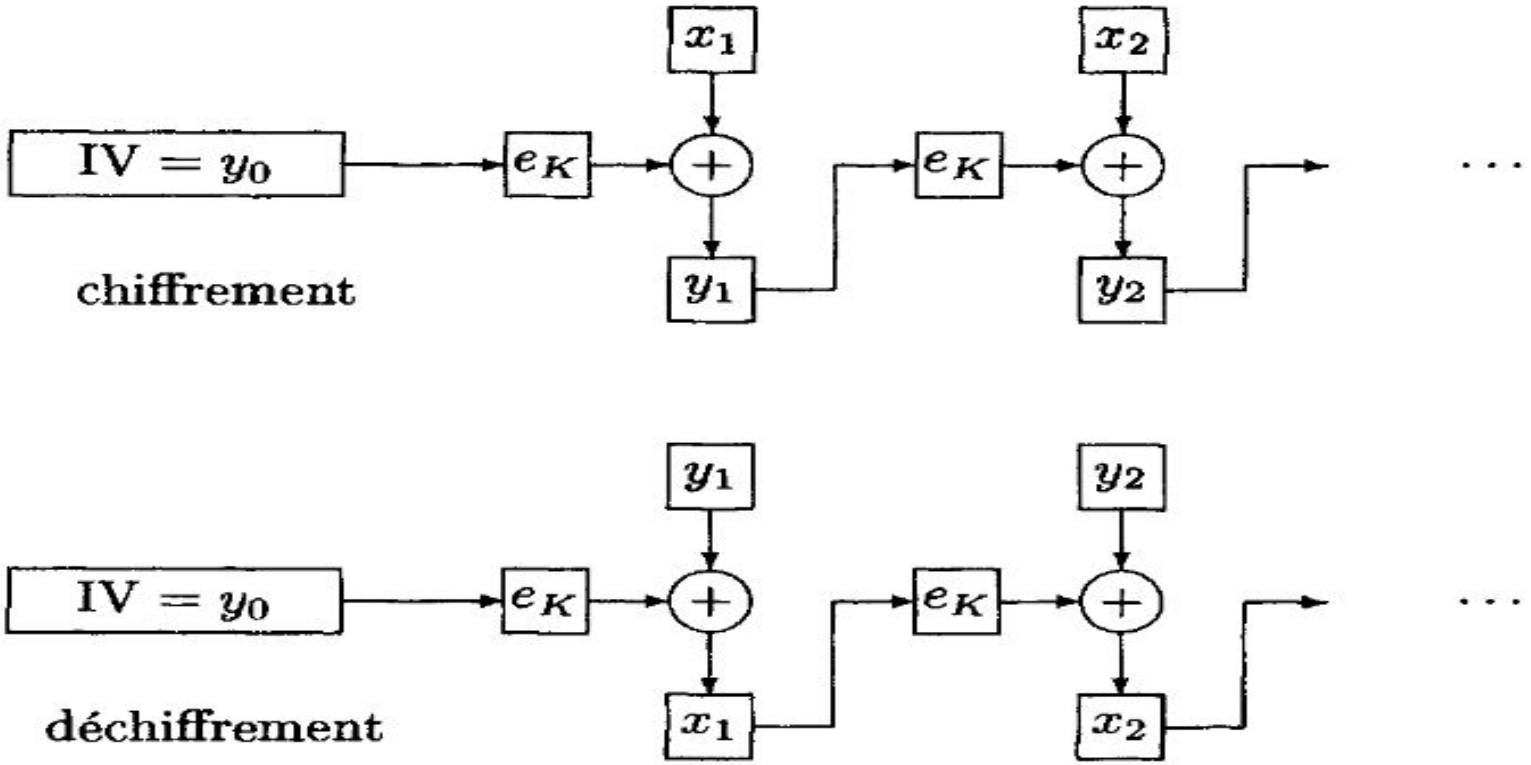
$$z_i = e_K(z_{i-1}), \ i \geq 1.$$

- La chaine de texte clair $x_1 x_2 \cdots$ est chiffré en calculant

$$y_i = x_i \oplus z_i, \ i \geq 1.$$

Le mode CFB

Dans le mode CFB, on commence avec $y_0 = IV$ (un bloc initial de k bits), et l'on produit la clef z_i en chiffrant le bloc de texte chiffré precedent, soit $z_i = e_K(y_{i-1}),\ i \geq 1$. Comme dans le mode OFB, on a $y_i = x_i \oplus z_i$.



Application pratique des modes opératoires

Les quatres modes d'utilisations ont des qualités et des défauts.

- Dans les modes ECB et OFB, si l'on change un bloc de texte clair x_i, cela ne perturbe que le bloc de texte chiffré y_i. Cela peut être une propriété indésirable. Ainsi, le mode OFB est souvent utilisé pour des transmissions par satellite.

- Dans les modes CBC et CFB, par contre, la modification d'un bloc de texte clair x_i perturbera le bloc de texte chiffré y_i et tous les suivants. Cela entraine que ces modes sont particuliérement adaptés au probléme de l'authentification. Plus précisément, ces modes sont utilisés dans les codes d'authentification de message.

En fin, les modes d'opérations nous permettent de définir des chiffrements par flot à partir des chiffrements par bloc.

Implémentation : Mini-Projet

Exercice pour Mini-Projet

Avant d'implémenter un modéle des chiffrements standards on doit étudier la structure algébrique de ces chiffrements. Pour les besoins de l'implémentation nous procéderons comme suit :

- Nous faisons un rappel sur les corps finis qui sont au coeur de la cryptographie.
- Nous donnons la structue algébrique d'une méthode de chiffrement choisie. Il s'agit du chiffrement AES.
- Nous donnons les différents opérateurs basiques de l'AES.
- Nous donnons les publications de Nist comme référence pour la structure algébrique des algorithmes de chiffrement standard à implémenter. Dans ce cadre voir FIPS 46-3 pour le DES et FIPS 197 pour l'AES.
- En fin nous donnons des indications pour l'implémentation.

Rappel sur les corps finis

Théorème (Existence et unicité des corps finis)

Pour tout nombre premier p et pour tout entier $r > 0$ il existe un corps finis (unique à isomorphisme près) de cardinal p^r.

Il distingue deux types de corps finis.

- Les corps finis premiers, $\mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$ où p est un entier premier.
- Les corps finis $\mathbf{F}_q$ où $q = p^r$, est tel que $r > 1$ et p un entier premier.

On définit un corps binaire, avec les notations suivantes

$$\mathbf{F}_2 = \mathbb{Z}/2\mathbb{Z} = \{0, 1\}.$$

Structure algébrique de AES

Rijndael utilise le corps fini $\mathbf{F}_{2^8}$ défini par le polynôme

$$p(x) = x^8 + x^4 + x^3 + x + 1.$$

Nous utilisons la représentation polynomial des éléments de $\mathbf{F}_{2^8}$. Donc nous pouvons identifier un élément de $\mathbf{F}_{2^8}$ par un vecteur de longueur 8 bits. Nous introduisons l'anneau des matrices suivant :

$$\mathcal{M}_4(\mathbf{F}_{2^8}) = \{X = (x_{ij})_{4\times 4} \mid x_{ij} \in \mathbf{F}_{2^8}\}.$$

Structure algébrique de AES

Pour la version 128 bits de l'AES Rijndael, un message M de taille 128 bits est considéré en 16 octets puis traduit en une matrice 4×4 :

$$M = (m_0, m_1, m_2, \cdots, m_{15})$$

où $m_i \in \mathbf{F}_{2^8}$, et l'état initial $M_0 \in \mathcal{M}_4(\mathbf{F}_{2^8})$ est donné par

$$M_0 = \begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix}.$$

Définition des opérateurs basiques de l'AES

La fonction de substitution consiste en une seule boîte S appliquée 16 fois, sur chaque octet de l'état.

SubByte

L'application SubByte est une application $S : \mathcal{M}_4(\mathbf{F}_{2^8}) \to \mathcal{M}_4(\mathbf{F}_{2^8})$.
Soient $c = (1, 1, 0, 0, 0, 1, 1, 0)$ et

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Pour $y = (y_0, y_1, \cdots, y_7) \in \mathbf{F}_{2^8}$, définir $T(y) = Ay^t + c^t$ et $\sigma(y) = y^{-1}$.
Pour $X = (x_{ij}) \in \mathcal{M}_4(\mathbf{F}_{2^8})$, S est défini comme $S(X) = (\mathcal{S}(x_{ij}))_{4\times 4}$ où $\mathcal{S}(x_{ij}) = (T \circ \sigma)(x_{ij})$.

Définition des opérateurs basiques de l'AES

La fonction de diffusion est la composée de deux fonctions linéaires, ShiftRows et MixColumns.

ShiftRow

La fonction ShiftRows consiste en une permutation circulaire des lignes de la matrice. La transformation ShiftRow R et son inverse sur un état X sont données comme suit

$$R(X) = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{11} & x_{12} & x_{13} & x_{10} \\ x_{22} & x_{23} & x_{20} & x_{21} \\ x_{33} & x_{30} & x_{31} & x_{32} \end{pmatrix}, \quad R^{-1}(X) = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{13} & x_{10} & x_{11} & x_{12} \\ x_{22} & x_{23} & x_{20} & x_{21} \\ x_{31} & x_{32} & x_{33} & x_{30} \end{pmatrix}$$

MixColumn

La fonction MixColumns est une multiplication par une matrice fixe inversible. Une transformation MixColumn L est une transformation linèaire sur $\mathbf{F}_{2^8}^4$. Soit α une racine de $p(x)$ dans $\mathbf{F}_{2^8}$. Pour un état donné X,

$$L(X) = LX.$$

La transformation linèaire L et son inverse sont données par

$$L = \begin{pmatrix} \alpha & 1+\alpha & 1 & 1 \\ 1 & \alpha & 1+\alpha & 1 \\ 1 & 1 & \alpha & 1+\alpha \\ 1+\alpha & 1 & 1 & \alpha \end{pmatrix}, \quad L^{-1} = \begin{pmatrix} \beta_0 & \beta_3 & \beta_2 & \beta_1 \\ \beta_1 & \beta_0 & \beta_3 & \beta_2 \\ \beta_2 & \beta_1 & \beta_0 & \beta_3 \\ \beta_3 & \beta_2 & \beta_1 & \beta_0 \end{pmatrix}$$

où $\beta_0 = \alpha^3 + \alpha^2 + \alpha$, $\beta_1 = \alpha^3 + 1$, $\beta_2 = \alpha^3 + \alpha^2 + 1$, et $\beta_3 = \alpha^3 + \alpha + 1$.

Ces deux dernières fonctions linéaires n'agissent que sur les octets, et pas sur les bits de l'état.

Chiffrement et déchiffrement sous AES

La composition d'opérateurs est définie comme suit :

$G(X) = (R \circ S)(X)$	$G^{-1}(X) = (S^{-1} \circ R^{-1})(X)$
$H(X) = (L \circ G)(X)$	$H^{-1}(X) = (G^{-1} \circ L^{-1})(X)$

Les clés des tours

Le nombre total de bits des clés de tour est égal à la taille des blocs que multiplie le nombre de tour plus 1. La version 128 bits utilise 10 tours. Alors on a besoin de 1408 bits. Ainsi, l'algorithme de génération de clés étend une clé de 128 bits à des clés de tours, au total de 1408 bits.

Chiffrement	Déchiffrement
$M_0 = M + \mathcal{K}_0,$	$C_0 = C + \mathcal{K}_{10},$
$M_i = H(M_{i-1}) + \mathcal{K}_i, 1 \leq i \leq 9$	$C_1 = G^{-1}(C_0) + \mathcal{K}_9$
$M_{10} = G(M_9) + \mathcal{K}_{10}$	$C_i = H^{-1}(C_{i-1}) + \mathcal{K}_{10-i}, 2 \leq i \leq 10$

Le texte chiffré est $C = M_{10}$ et le texte clair est $M = C_{10}$.

Indication pour l'implémentation

Reprendre le mini-projet développé en classe et suivre les points suivants.

- Considérer la dernière étape à la quelle on s'est arrêté.
- Essayer d'implémenter les nouveaux algorithmes traités entre temps.
- Vous pouvez utiliser des tailles réduites de clés, l'essentiel est que l'impémentation suit la logique de l'algorithme.
- Commenter clairement les différents bouts de codes.

Indication pour l'implémentation

En ce qui concerne la notation :

- La lisibilité du code compte.
- Le design des interfaces compte.
- Vous être libre de choisir vos méthodes de chiffrement dans le menu déroulant "Méthode" (voir énoncé mini-projet). Cependant la difficulté de l'algorithme choisi compte bien.
- Le fait que l'algorithme implémenté soit une standard compte aussi.
- Les propositions particulières comptent.

Bonne chance ! ! !

Conclusion

Les algorithmes symétriques ont un véritable avantage découlant notamment de leur rapidité (ils utilisent de petits entiers et des opérations rapides), et de leur facilité de conception. Cependant ils font face à plusieurs contraintes :

- Confidentialité de la clé secrète : probléme de partage de la clé à travers un canal sûr et problème de stockage de la clé.
- Durée de vie des clés assez courte.
- Certains services de sécurité ne sont pas pris en charge. Par exemple on ne peut déterminer qui entre les deux interlocuteurs légitimes, a chiffré un message. C'est le problème de la non-répudiation.
- Distribution des clés : si n personnes communiquent 2 à 2, il faut

$$C_n^2 = \frac{n(n-1)}{2}$$

clés. Ce chiffre devient trés difficile à gérer pour des n trés grands.

Ce qui nous pousse à explorer d'autres systèmes de chiffrement.