**République Algérienne Démocratique et Populaire**
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**
**Université Mohammed Seddik Benyahia - Jijel**

**Faculté des Sciences Exactes et Informatique**
**Département d'Informatique**

---

## Polycopié

# Cryptographie

Niveau : 1ère année Master
Option : Réseaux et sécurité

Préparé par

**Dr. Ismahane SOUICI**

souici.ismahane@univ-jijel.dz
souici.ismahane@yahoo.fr

**2023**

# Avant-propos

Ce polycopié, destiné aux étudiants en formation de Master en réseaux et sécurité, présente le contenu de cours du module "Cryptographie" de la première année master (M1) conformément aux programmes d'enseignement du canevas proposé.

A travers ce support de cours, les étudiants pourront acquérir les notions de base en sécurité de données notamment en cryptographie et ainsi, approfondir leurs connaissances sur les scénarios de sécurisation de données et leurs caractéristiques. Afin de concrétiser et de rendre pratique les différentes notions abordées, une série d'exercices est proposée en annexe.

Ce support de cours est organisé en six chapitres comme suit :

Le premier chapitre initie les étudiants dans le domaine de cryptographie en présentant les principes fondateurs et la terminologie du domaine ainsi qu'une classification des méthodes de cryptages. Le deuxième chapitre expose brièvement la première classe dite classique ou encore à usage restreint car elle décrit la période avant les ordinateurs ; alors que le troisième chapitre expose les principes de base de la cryptographie moderne. Il explicite les principes, l'application, les avantages et les limites des deux principales catégories de méthodes cryptographiques modernes symétriques et asymétriques. Les trois derniers chapitres sont consacrés, respectivement, à la présentation d'exemples de crytposystèmes symétriques, asymétriques et hybrides.

# Table des matières

# Liste des figures

# *Chapitre I*
# *Principes fondateurs de la cryptographie*

## 1. Introduction

Depuis toujours, l´être humain a cherché à conserver certaines informations ou données secrètes, à défaut, à en restreindre l´accès à certaines personnes. C'est pourquoi, et dès l'antiquité, les peuples employèrent des codes secrets dans certains de leurs textes : les archéologues en ont découvert dans des hiéroglyphes égyptiens. De même, les Hébreux dissimulaient parfois leurs écrits en inversant l'alphabet, c'est-à-dire en employant la dernière lettre de l'alphabet à la place de la première, l'avant-dernière lettre à la place de la deuxième, et ainsi de suite. Sur le champ de bataille, les Spartes communiquaient souvent avec leurs généraux par le biais de messages écrits sur un ruban de parchemin enroulé en spirale sur un bâton de diamètre défini, appelé scytale. Une fois le ruban déroulé, on ne pouvait lire le message qu'en enroulant le ruban autour d'une règle identique. Jules César se servit également de codes secrets pour correspondre avec ses hommes, et laissa même son nom à un chiffre particulier.

Jusqu'au début du XXème siècle, la cryptographie a gardé une importance mineure, et les méthodes utilisées étaient bien souvent rudimentaires. Mais lors de la seconde guerre mondiale, et avec l´apparition de technologies de communication évoluées, telles que la radio, a rendu nécessaire la mise au point de mécanismes de cryptage empêchant l´interception des signaux par l´ennemi. Il était devenu indispensable de chiffrer les données transmises par les ondes (Enigma).

Avec l'avènement des réseaux, et tout particulièrement Internet, la cryptographie prend maintenant une nouvelle dimension, économique cette fois. C'est en effet toute la sécurité du commerce électronique qui dépend maintenant de l'inviolabilité des codes cryptés. Ainsi la cryptographie s'élargit du domaine confidentiel de la protection des gros serveurs (universités, entreprises, état) à la consommation de masse par les particuliers (commerce électronique, confidentialité des mails,...). À l'inverse, la cryptanalyse (craquage des codes cryptés) change elle aussi d'acteurs et d'objet. Jusqu'alors arme militaire et jeu de quelques génies travaillant pour la célébrité, elle devient une véritable arme de vol à grande échelle (détournement de codes de carte bleue, de fonds,...) et de guerre économique (vol de secrets industriels ou commerciaux).

## 2. Principes fondateurs de la cryptographie

### 2.1. Notion de cryptologie

La cryptographie compte parmi les différents systèmes d'écriture permettant de modifier de façon volontaire les caractères d'un message. Donc, ce procédé protège une communication qui devient lisible uniquement par l'expéditeur et par le destinataire auquel le message est adressé.

La *cryptographie* appartient à la **cryptologie** du grec *kruptos* « secret, caché » et *logos* « discours », qui est la science de l'écriture secrète englobant des pratiques concurrentes à savoir la **cryptographie**, le **déchiffrement** et la **cryptanalyse**. La première pratique qui est la cryptographie (du grec *kruptos* et *graphein*) est : « la discipline incluant les principes, les moyens et les méthodes de transformation des données, dans le but de masquer leur contenu, empêcher leur modification ou leur utilisation illégale, ainsi que les opérations inverses, pour rendre le document à nouveau intelligible ». Le déchiffrement est le processus permettant de transformer le message chiffré en message clair. Quand à la cryptanalyse, qui est un terme créé par le cryptologue américain *William Friedman* en 1920 (du grec *kruptos* et *analusis* « résolution, dissolution »), est l'art de décoder un message chiffré en mêlant une intéressante combinaison de raisonnement analytique, d'application d'outils mathématiques, de découverte de redondances, de patience, de détermination, et de chance.

Les deux disciplines de cryptographie et de cryptanalyse s'alimentent l'une l'autre. On ne peut pas évaluer la sécurité d'un mécanisme sans le soumettre à des attaques qui, à leur tour, conduisent à

des critères de conception pour rendre les procédés plus sûrs. Ces derniers seront à nouveau passés au crible du cryptanalyste…

### 2.2. Terminologie

La cryptologie, et par conséquent la cryptographie, est essentiellement basée sur l'arithmétique. Il s'agit dans le cas d'un texte de transformer les lettres qui composent le message en une succession de chiffres, puis ensuite de faire des calculs sur ces chiffres pour :

- ✓ d'une part les modifier de telle façon à les rendre incompréhensibles. Le résultat de cette modification (le message chiffré) est appelé **cryptogramme** (en anglais *ciphertext*) par opposition au message initial, appelé **message en clair** (en anglais *plaintext*) ;
- ✓ faire en sorte que le destinataire saura les déchiffrer.

Le chiffrement se fait généralement à l'aide d'une *clef de chiffrement*, le déchiffrement nécessite quant à lui une *clef de déchiffrement*. On distingue généralement deux types de clefs :

• **Les clés symétriques :** il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de *chiffrement symétrique* ou de *chiffrement à clé secrète*.
• **Les clés asymétriques :** il s'agit de clés utilisées dans le cas du *chiffrement asymétrique* (aussi appelé *chiffrement à clé publique*). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

On appelle *décryptement* (*décryptage*) le fait d'essayer de *déchiffrer illégitimement* le message (que la clé de déchiffrement soit connue ou non de l'*attaquant*). Lorsque la clef de déchiffrement n'est pas connue de l'attaquant on parle alors de **cryptanalyse** ou **cryptoanalyse** (on entend souvent aussi le terme plus familier de *cassage*).

Le processus cryptographique peut être récapitulé par la figure ci-dessous :

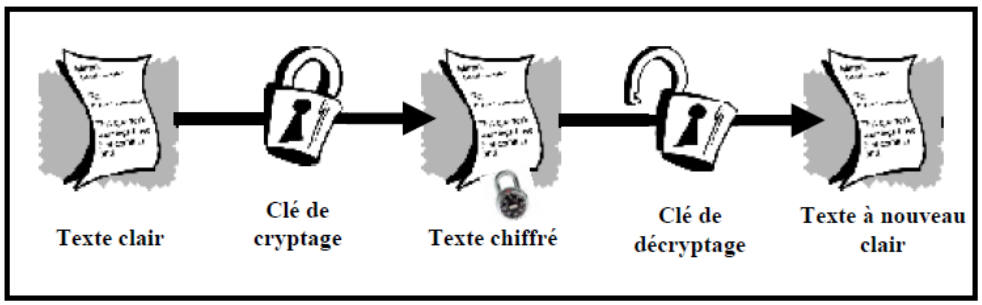



**Figure 1.1.** *Processus cryptographique.*

### 2.3. Les grandes menaces et les fonctionnalités offertes par la cryptographie

#### 2.3.1. Les grandes menaces

De façon générale, les grands types de menaces que peut subir un message lors de son échange peuvent être récapitulés à travers les points suivants :

**a. Les attaques passives :** Avec ce type d'attaque et lors d'une communication élaborée entre deux personnes souvent nommées *Alice* et *Bob*, *Oscar* qui est l'opposant ou l'attaquant, se contente d'écouter le message tout en essayant de menacer sa confidentialité. Dans ce cas, il se peut qu'une information secrète parvienne également à une personne autre que son destinataire légal.

**b. Les attaques actives :** Ici, *Oscar* peut menacer l'intégrité, qui sera présentée dans la section suivante. Ainsi, ces informations vont parvenir d'une personne autre que leur véritable auteur. Et comme exemple d'attaques actives, on peut citer :

- ✓ l'usurpation d'identité (de l'émetteur ou du récepteur) ;
- ✓ l'altération / modification du contenu des messages ;

- ✓ la destruction de messages/ le retardement de la transmission ;
- ✓ la répétition de messages (jusqu'à engorgement) ;
- ✓ la répudiation de message : l'émetteur nie avoir envoyé le message.

**c. La cryptanalyse :** Elle permet d'étudier la sécurité des procédés de chiffrement utilisés en cryptographie. Ainsi, elle désigne habituellement les techniques qui permettent d'extraire de l'information sur des secrets en observant uniquement les données publiques d'un cryptosystème. Les deux types de secrets sont le message clair et la clé. Ce qui compte avant tout dans une cryptanalyse, c'est de gagner de l'information sur le message clair. Ceci dit, il va de soi que gagner de l'information sur la clé de chiffrement privé permettant de déchiffrer tous les messages, ce qui résout définitivement le problème. Et suivant les données qu'elle nécessite, on distingue habituellement quatre méthodes de cryptanalyse :

- **attaque sur texte chiffré seul (ciphertext-only) :** le cryptanalyste possédant des exemplaires chiffrés des messages, essaye de faire des hypothèses sur les messages originaux qu'il ne possède pas en vue de retrouver la clé de déchiffrement. Dans ce cas, la cryptanalyse sera très difficile à cause du manque d'informations à disposition.
- **attaque à texte clair connu (known-plaintext attack) :** le cryptanalyste essaye de retrouver la clé de déchiffrement à partir de messages ou de parties de messages en clair possédés et de leurs versions chiffrées correspondantes.
- **attaque à texte clair choisi (chosen-plaintext attack) :** Consiste à retrouver la clé de déchiffrement à partir de messages en clair, et en ayant la possibilité de générer les versions chiffrées de ces messages avec un algorithme considérer comme une boîte noire.
- **attaque à texte chiffré choisi (chosen-ciphertext attack) :** le cryptanalyste possède des messages chiffrés et demande la version en clair de certains de ces messages pour mener l'attaque (retrouver la clé de déchiffrement).

### 2.3.2. Les fonctions de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages clairs aux yeux de certains utilisateurs pour assurer leur *fiabilité* et *confidentialité* surtout s'ils feront l'objet des communications via Internet. Désormais, les fonctions de la cryptographie se sont étendues pour englober de nouvelles fonctions en plus de la fiabilité et la confidentialité, il s'agit de garantir l'*intégrité* et l'*authenticité* des données échangées. Ceux-ci sont les fonctions principales de la cryptographie. Elle a d'autres fonctions, dites secondaires, qui sont : l'*horodatage*, le *témoignage*, l'*accusé de réception* et la *révocation*.

**a. La confidentialité :** Permet de protéger le contenu des informations sauvegardées ou transmises sur un réseau. Seules les personnes autorisées doivent pouvoir accéder aux informations ainsi protégées. Le *chiffrement de l'information* permet de résoudre le problème de la confidentialité : une personne souhaitant transmettre un message lui applique au préalable une fonction dite de chiffrement, et transmet le résultat au destinataire. Ce dernier retrouve le message original en utilisant une fonction de déchiffrement suivant le modèle de la cryptographie utilisé. Dans le modèle de la cryptographie à clé secrète les deux parties partagent la même clé de chiffrement et de déchiffrement, qui doit être gardée secrète. Les deux personnes jouent ainsi un rôle symétrique, tandis que, dans le modèle de la cryptographie à clé publique, le chiffrement est publique et le déchiffrement est confidentiel. Pour envoyer un message chiffré, on applique une fonction de chiffrement utilisant la clé publique du destinataire. Ce dernier est le seul qui peut retrouver le message original à l'aide de sa clé privée. Les deux clés sont liées mathématiquement, mais il doit être impossible dans la pratique de retrouver la clé privée à partir de la clé publique (plus de précisions, ainsi que quelques exemples de méthodes sur les deux modes de chiffrement, symétrique et asymétrique, seront donnés en avant respectivement dans les sections 3.3.2 et 4).

**b. L'intégrité :** C'est la capacité à reconnaître qu'une information a été altérée**,** soit de manière accidentelle ou intentionnelle.

**c. L'authentification :** Consiste à assurer l'identité d'un utilisateur, c'est à dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il doit être.

On distingue deux types d'authentification :

• **Authentification d'un tiers :** C'est l'action qui consiste à prouver son identité. Ce service est généralement rendu par l'utilisateur d'un « échange d'authentification » qui implique un certain dialogue entre les tiers communicants.
• **Authentification de l'origine des données :** Elle sert à prouver que les données reçues ont bien été émises par l'émetteur déclaré. Dans ce cas, l'authentification désigne souvent la combinaison de deux services : authentification et intégrité.

**d. La non-répudiation :** La non-répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction.

Ces fonctionnalités représentent des solutions aux problèmes causés par les menaces citées précédemment, ainsi :

- ✓ Pour assurer la confidentialité, on utilise un algorithme de chiffrement.
- ✓ Contre l'usurpation d'identité, on utilise des algorithmes d'authentification.
- ✓ Pour empêcher l'altération de données, on utilise des algorithmes de contrôle d'intégrité.
- ✓ Contre la répudiation, des algorithmes de signatures ont été proposés.

Les besoins de sécurisation et de confidentialité s'imposent à divers degrés dans différentes applications. Citons à titre d'exemple :

- ✓ Confidentialité des transactions bancaires,
- ✓ Protection de secrets industriels ou commerciaux,
- ✓ Protection des secrets médicaux,
- ✓ Protection des systèmes informatiques contre les intrusions,
- ✓ Protection de la confidentialité des communications dans le cadre d'une association d'un parti politique, d'un syndicat…
- ✓ Protection de la vie privée,
- ✓ Jeux
- ✓ Etc…

## 3. Algorithmes cryptographiques

Comme nous l'avons déjà mentionné, le but de la cryptographie est de permettre à deux personnes de s'échanger des informations en toute sécurité à travers un canal peu sûr, qui peut être une ligne téléphonique ou tout autre réseau de communication. L'information que l'on souhaite transmettre et que l'on appelle texte clair, sera donc chiffrée par un procédé de chiffrement et en utilisant une clé prédéterminée. Le destinataire est le seul qui peut retrouver l'information originale suite à une opération de déchiffrement de l'information chiffrée en utilisant une clé de déchiffrement sans laquelle ce procédé est impossible.

Le processus de chiffrement ou de déchiffrement utilise une fonction mathématique. C'est l'**algorithme cryptographique** ou encor appelé **chiffre**. La sécurité des données chiffrées est entièrement dépendante de deux choses : la force de l'algorithme cryptographique et le secret de la clé. Un algorithme cryptographique, plus toutes les clés possibles et tous les protocoles qui le font fonctionner constituent un **cryptosystème.**

### 3.1. Le principe de kerckhoffs

Longtemps, la sécurité d'un système cryptographique a reposé sur le secret qui l'entoure (cas du chiffre de César, du code ADFVGX utilisé par les Allemands durant la première guerre mondiale, …). Cette idée s'est ensuite abandonnée du fait qu'un tel secret peut toujours être révélé par un espion, sinon une étude approfondie finira par percer son fonctionnement. C'est par exemple ce qu'ont réussi les Polonais en reconstituant les organes d'une machine Enigma et plus récemment,

l'algorithme de chiffrement du GSM qui n'a jamais été officiellement révélé, on le trouve en détails sur le web. C'est pourquoi un système cryptographique doit dépendre d'un paramètre aisément modifiable : sa clé.

Le premier à avoir formalisé ce principe est le hollandais *Auguste Kerckhoffs* en écrivant en janvier 1883 dans le « Journal des sciences militaires » un article intitulé « La cryptographie militaire », où il disait :

*« Il faut bien distinguer entre un système d'écriture chiffrée, imaginé pour un échange momentané de lettres entre quelques personnes isolées, et une méthode de cryptographie destinée à régler pour un temps illimité la correspondance des différents chefs d'armée entre eux. Ceux-ci, en effet, ne peuvent, à leur gré et à un moment donné, modifier leurs conventions ; de plus, ils ne doivent jamais garder sur eux aucun objet ou écrit qui soit de nature à éclairer l'ennemi sur le sens des dépêches secrètes qui pourraient tomber entre ses mains.*
*Un grand nombre de combinaisons ingénieuses peuvent répondre au but qu'on veut atteindre dans le premier cas ; dans le second, il faut un système remplissant certaines conditions exceptionnelles, conditions que je résumerai sous les six chefs suivants :*

***1)*** *le système doit être matériellement, sinon mathématiquement, indéchiffrable.*
***2)*** *Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénients tomber entre les mains de l'ennemi.*
***3)*** *La clé doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants.*
***4)*** *Il faut qu'il soit applicable à la correspondance télégraphique.*
***5)*** *Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes.*
***6)*** *Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer. »*

Les points 2 et 3 sont les axiomes fondamentaux de la cryptographie suivant lesquels l'attaquant possède tous les détails de l'algorithme sans pouvoir rien faire puisqu'il lui manque la clé spécifique pour le chiffrement. Cela mène vers la certitude suivante : si on ne sait pas casser un algorithme même en sachant comment il fonctionne, on ne sait certainement pas le casser sans cette connaissance. Donc, un chiffre basé uniquement sur le secret de l'algorithme n'a aucun intérêt, car un jour ou l'autre ce secret sera découvert ou volé.

Par exemple, même si on connaît le mode d'emploi du carré de Vigenère (voir la section 3.3.1.1.b), on ne pourra quand même pas, ou difficilement, décrypter un message si on ne connaît pas la clef. Par contre, le chiffre Atbash, qui est une méthode de substitution alphabétique inversée sans utilisation de clé, repose entièrement sur la manière de chiffrer. Actuellement, on va encore plus loin où le mécanisme de chiffrement est publié afin que les cryptanalystes puissent l'étudier. D'ailleurs, on suppose toujours, en cryptanalyse académique, que le système de chiffrement est connu.

L'interprétation de ce principe par *Bruce Schneier* a portée à l'« élégance » dans le cassage d'un cryptosystème. Traduit de l'anglais : « *Le principe de Kerckhoffs s'applique au-delà des chiffres et des codes, c'est-à-dire aux systèmes de sécurité en général : tout secret est en fait un point de cassure possible. Par conséquent, le secret est une cause première de fragilité, donc cela même peut amener un système à un effondrement catastrophique. À l'inverse, l'ouverture amène la ductilité.* ».
Ici par fragilité, *Bruce Schneier* accentue sur le fait de garder comme secret une information peu coûteuse à remplacer en cas où le secret sera divulgué. Par exemple, si la sécurité d'un cryptosystème implémenté sur du matériel informatique et des logiciels géographiquement distants et largement dispersés, dépend de garder cette distribution secrète, alors sa divulgation demanderait de grands efforts en terme de développement, de tests et de distribution de nouveaux algorithmes.

Mais si le secret de l'algorithme était tout simplement une clé, sa divulgation entraîne moins de problèmes puisqu'il suffit d'en générer une nouvelle et de la distribuer. En bref, moins on a de secrets, moins on doit faire de maintenance.

### 3.2. Description formelle d'un algorithme cryptographique

D'une manière formelle, un cryptosystème est un quintuplet (*P, C, K, E, D*) satisfaisant les points suivants :

**1)** *P* est un ensemble fini de blocs de textes clairs possibles.
**2)** *C* est un ensemble fini de blocs de textes chiffrés possibles.
**3)** *K* est un ensemble fini de clefs possibles.
**4)** Pour tout $k \in K$, il y a une règle de chiffrement $e_k \in E$ et une règle de déchiffrement correspondante $d_k \in D$. Chaque $e_k : P \rightarrow C$ et $d_k : C \rightarrow P$ sont des fonctions telles que $d_k(e_k(x)) = x$ pour tout texte clair $x \in P$.

La principale propriété est la quatrième. Elle précise que si un texte clair x est chiffré en utilisant $e_k$, et si le texte chiffré y obtenu est ensuite déchiffré en utilisant $d_k$, on retrouve le texte clair x original.

Alice et Bob peuvent employer le protocole suivant pour utiliser un cryptosystème spécifique. Tout d'abord, ils choisissent une clé quelconque $k \in K$. Cette opération est effectuée lorsqu'ils se rencontrent en un même endroit loin d'être observés par Oscar, ou bien à travers un canal de communication sûr. Supposant qu'ensuite, Alice souhaite communiquer un message à Bob par un canal peu sûr, ce message étant une chaîne :

$$x = x_1 x_2 \ldots x_n$$

avec : $n \in Z$, $n \geq 1$, $x_i \in P$ et $1 \leq i \leq n$.

Chaque bloc $x_i$ est chiffré en utilisant la règle de chiffrement $e_k$ spécifiée par la clé k choisie. Ainsi, Alice calcule $y_i = e_k(x_i)$, $1 \leq i \leq n$, et la chaîne chiffrée obtenue sera :

$$y = y_1 y_2 \ldots y_n$$

Cette chaîne est envoyée dans le canal et une fois reçue par Bob, il la déchiffre en utilisant la fonction de déchiffrement $d_k$ pour récupérer le texte clair original $x_1 x_2 \ldots x_n$. Le procédé de communication est illustré sur la figure I.2.

Il est évident que chaque fonction de chiffrement $e_k$ doit être injective (c'est à dire ne pas chiffrer deux blocs différents en deux valeurs égales), sinon, le procédé de déchiffrement ne pourrait être fait sans ambiguïté. Plus précisément, si :

$$y = e_k(x_1) = e_k(x_2)$$

Avec : $x_1 \neq x_2$, Bob n'as aucun moyen de savoir si y doit être déchiffré en $x_1$ ou en $x_2$.

Les principes fondamentaux d'un algorithme de cryptographie sont basés sur deux notions essentielles, énoncées par Shannon en 1949 :

- **Confusion :** Sert à cacher la relation entre le clair et le chiffré. Donc, elle vise à rendre le texte aussi peu lisible que possible. Ceci peut se faire par une substitution méthodique de symboles, ou par un algorithme de codage aussi complexe que l'on veut. Comme ça aucune propriété statistique ne peut être déduite du message chiffré.
- **Diffusion :** Sert à cacher la redondance dans le message et à diffuser sur tout le chiffré l'influence du changement d'un bit de clef ou d'un bit du clair. Donc, elle vise à rendre chaque élément d'information du texte chiffré dépendant d'un nombre aussi grand que possible d'éléments d'information du texte clair. Ceci rend la découverte de l'algorithme, ou de la clé de cet algorithme, en principe plus difficile. Ainsi, toute modification du message en clair se traduit par une modification complète du chiffré.

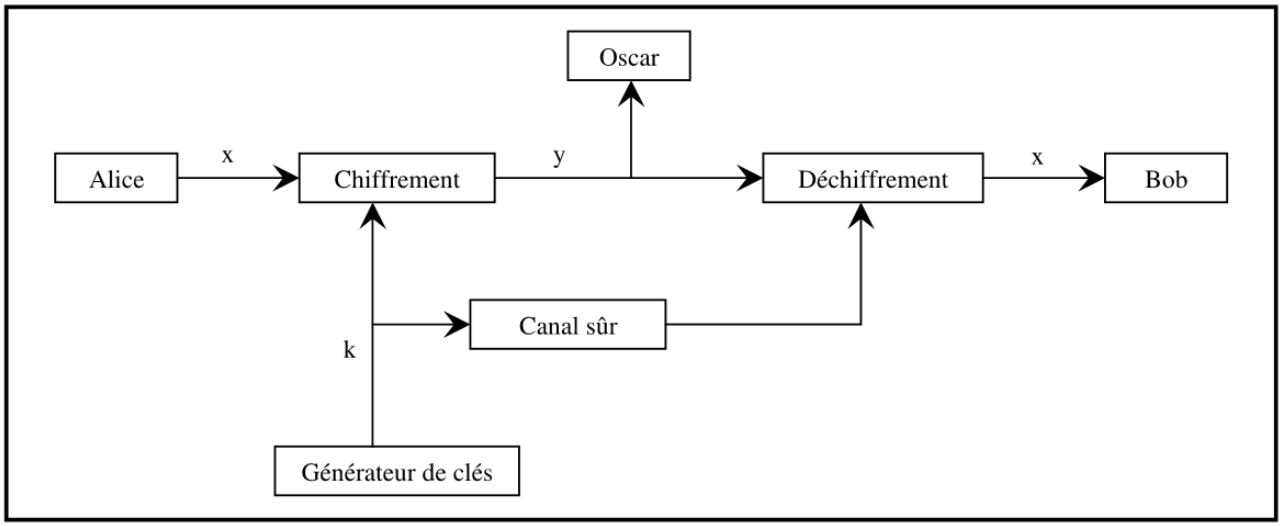



**Figure 1.2.** *Le procédé de communication.*

### 3.3. Classes de la cryptographie

Le schéma suivant présente les différentes classes de la cryptographie :

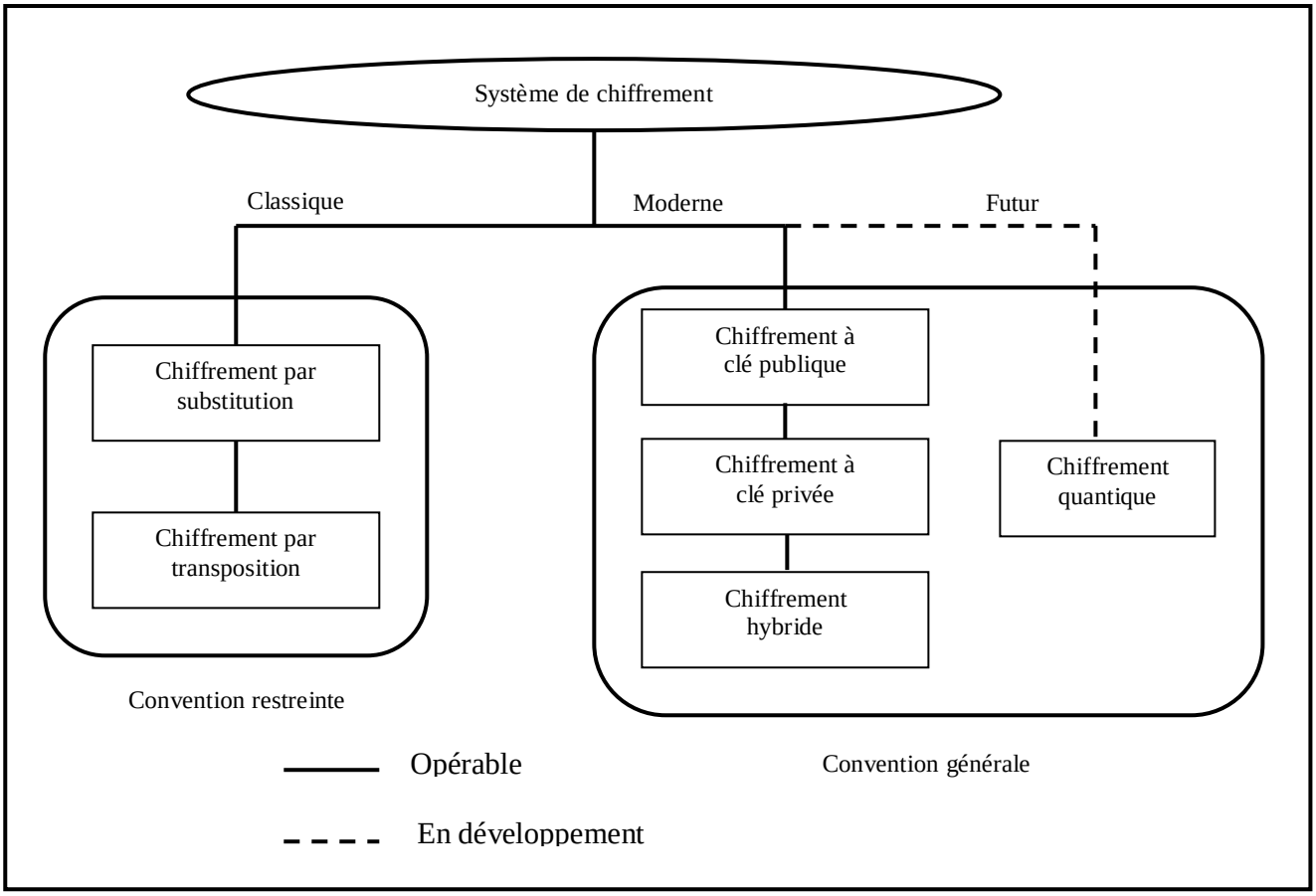



**Figure 1.3.** *Les classes de la cryptographie*.

# *Chapitre II*
# *La cryptographie classique*

## 1. Introduction

La cryptographie classique décrit la période avant les ordinateurs durant laquelle, les principaux outils utilisés consistent à remplacer des caractères par d'autres et les transposer dans des ordres différents tout en gardant secrètes les procédures de chiffrement ou de déchiffrement. Sans cela le système est complètement inefficace, puisque n'importe qui peut déchiffrer le message codé. On appelle généralement cette classe de méthodes : le chiffrement à **usage restreint**.

## 2. Catégories de la cryptographie classique

### 2.1. Cryptographie par substitution

Dans ce mode de cryptage, les lettres du message en clair sont remplacées par d'autres lettres, des chiffres ou d'autres symboles. Selon la façon de substituer, on distingue la substitution mono-alphabétique, la substitution homophonique et la substitution poly-alphabétique.

**a. Substitution mono-alphabétique :** C'est le plus simple des codages à réaliser. Il s'agit de remplacer chaque lettre par une lettre différente, ou même un autre symbole. Plus formellement :

$$f : A_M \rightarrow A_C$$
$$c_i = f(m_i) = m_i + k \pmod{|A_M|}$$

Où :

$A_M$ et $A_C$ sont respectivement l'ensemble d'alphabets du message en clair et l'ensemble d'alphabets du message chiffré.

$M = m_0\, m_1 \ldots m_{n-1}$ tel que : $\forall i, m_i \in A_M$ .

$C = c_0\, c_1 \ldots c_{n-1}$ tel que : $\forall i, c_i \in A_C$ .

La plus ancienne des méthodes s'inscrivant sous ce mode de chiffrement est le **chiffre de césar** utilisé par l'armée romaine (1ier siècle avant JC). Il consiste à décaler les lettres de l'alphabet d'un nombre n. Par exemple, pour n=3, A sera remplacé par D, B par E, C par F...Son principe très simple à mettre en œuvre facilite sa cryptanalyse du fait que, le nombre de façon de chiffrer un message reste très faible, puisqu'il est égal au nombre de lettres de l'alphabet, c'est à dire que l'on a seulement 26 façons. Malgré ça, cette même simplicité a conduit les officiers sudistes à le réemployer durant la guerre de Sécession. L'armée russe a fait de même en 1915. Une autre attaque possible contre ce système est la cryptanalyse fréquentielle qui se base sur le fait que les lettres les plus fréquentes dans le texte en clair restent les plus fréquentes dans le texte chiffré. Donc, ce système ne cache pas les fréquences d'apparition des caractères ce qui constitue une faiblesse importante permettant aux techniques statistiques d'associer une lettre probable aux lettres les plus fréquentes, et en appliquant une technique sémantique récursive, les algorithmes à base de substitutions mono-alphabétiques sont facilement cassés par les spécialistes.

Il est à noter que le code de César a été utilisé sur des forums Internet sous le nom de ROT13 (rotation de 13 lettres où A-->N...). Le ROT13 n'a pas pour but de rendre du texte confidentiel, mais plutôt d'empêcher la lecture involontaire. Son utilisation est simple : il suffit de re-chiffrer un texte, codé en ROT13, une deuxième fois pour obtenir le texte en clair.

Dans cette catégorie, on peut citer aussi : les alphabets désordonnés, le chiffre affine, ...

Dans ce même mode de chiffrement, et lorsqu'une même lettre sera substituée par plusieurs lettres qui seront bien déterminées à l'avance ; par exemple, 'A' peut correspondre à 5, 13, 25 ou 56; 'B' à 7, 19, 31, ou 42, …; cette façon particulière de substitution mono-alphabétique est appelée **substitution homophonique**. Ce procédé est plus sûr que le précédant (substitution mono-alphabétique), mais aussi craqué par les cryptanalystes ou par des espions expérimentés.

**b. Substitution poly-alphabétique :**

Aussi appelée à alphabets multiples. Elle a été inventée par Trithemius en 1518 et cryptanalyser par Kasiski en 1863. Avec cette méthode, une même lettre peut être remplacée par plusieurs symboles pris aléatoirement. Cela est garantie grâce à une clé $k = k_0\ k_1 \ldots k_{j-1}$ qui définie j fonctions distinctes $fk_1, fk_2, \ldots, fk_{j-1}$ définies comme suit :

$$\forall i : 0 \le i < n \qquad f_{k_l} : A_M \longrightarrow A_C \qquad \forall l : 0 \le l < j$$

$$c_i = f_{k_l \bmod j}(m_i) = m_i + k_{i \bmod j} (\text{mod} |A_M|$$

Avec $A_M$, $A_C$, M et C auront la même signification que ceux utilisés lors de la présentation de la M C substitution mono-alphabétique.

L'exemple le plus fameux de chiffre poly-alphabétique est sans doute le **chiffre de Vigenère**, qui a résisté aux cryptanalystes pendant trois siècles. Ce chiffre a été présenté en 1586 par le diplomate français Blaise de Vigenère lors de la publication de son œuvre : « Traité des chiffres ou Secrètes manières d'écrire », tout en s'appuyant sur les bases de Bellaso, Alberti, Porta et Trithème. Il s'agit en réalité d'une amélioration du chiffre de César, puisqu'il exploite la même méthode, mais en changeant le décalage de lettre en lettre. De plus, il utilise les 26 alphabets écrits en carré, et en décalant d'une lettre à chaque fois (voir Figure 2.1).

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| I | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| J | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| K | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| L | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| M | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| N | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| O | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| P | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| Q | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| R | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| S | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| T | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| U | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| V | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| W | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| X | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| Y | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| Z | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

**Figure 2.1.** *Le carré de Vigenère.*

Pour coder un message en utilisant ce chiffre, on choisit une clé de longueur arbitraire et on la répète selon la longueur du message à coder. On l'écrit ensuite au dessus du message à coder lettre par lettre. La première lettre du message chiffré sera la lettre située à l'intersection de la ligne correspondant à la première lettre de la clé avec la colonne correspondant à la première lettre du texte à chiffrer (texte clair), et on continu ainsi jusqu'à terminer le chiffrement de tout le texte. Pour déchiffrer, il suffit de faire la même opération en sens inverse, c'est à dire que sur la ligne de la lettre de la clé on recherche la lettre du message codé, la véritable lettre se trouve alors au sommet de la colonne correspondante.

Son point fort, c'est qu'il offre des modes de codage et de décodage faciles à appliquer, et son plus grand intérêt est que la même lettre sera codée de différentes manières en pénalisant ainsi toute tentative de cryptanalyse fréquentielle à condition que la longueur du message à chiffrer ne soit pas bien plus longue que celle de la clé. Dans le cas contraire, il sera possible de repérer la longueur de la clé dans le message. Ainsi si, par exemple, la longueur de la clé est de 3, alors la première lettre du message est codée avec la première lettre de la clé, la deuxième avec la deuxième, la troisième avec la troisième, et on revient, la quatrième avec la première,... Dans ce cas, on peut déterminer les caractères de la clé un par un suite à une analyse de fréquences d'apparition. La solution consiste donc, à considérer une clé se rapprochant le plus possible de la longueur du message, bien que même cela peu augmenter les possibilités de commettre des erreurs vue que le travail devient difficile ; et par conséquence, le message devient indéchiffrable.

Beaucoup d'autres méthodes s'inscrivent sous ce mode de chiffrement. L'une qui a attiré l'attention des militaires allemands pendant la seconde guerre mondiale, est la machine Enigma, qui est une sorte de machine à écrire très complexe qui pouvait avoir plus de 10000000000000000 clés possibles.

### 2.2. Cryptographie par transposition

Ici, c'est l'ordre des éléments d'une information qui est modifié (caractères d'une phrase, pixels d'une image...), ce qui permet de mieux cacher le message. Plusieurs types de transposition existent.

**a. Transposition simple par colonnes :** On écrit le message horizontalement dans une matrice prédéfinie, et pour retrouver le texte chiffré, on lit la grille verticalement. Le procédé inverse représente le procédé de déchiffrement. La figure ci-dessous résume ce principe.

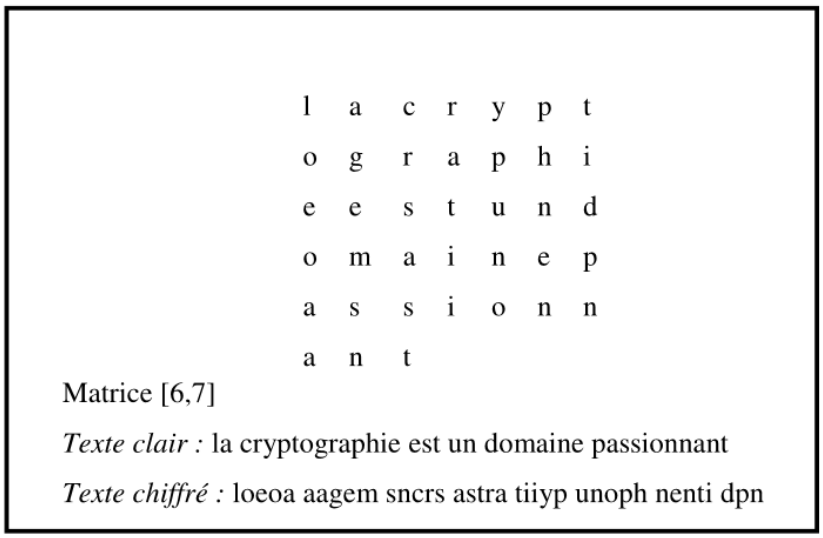



**Figure 2.2.** *Transposition simple par colonne.*

**b. Transposition complexe par colonnes :** Un mot clé secret constitué uniquement de caractères différents est utilisé pour construire une séquence de chiffres représentant les ordres d'apparition dans l'alphabet des différentes lettres composant ce mot. Le chiffrement se fait en écrivant d'abord le message par lignes dans un rectangle, comme le montre la figure 2.3, puis on lit le texte par colonnes en suivant l'ordre déterminé par la séquence.

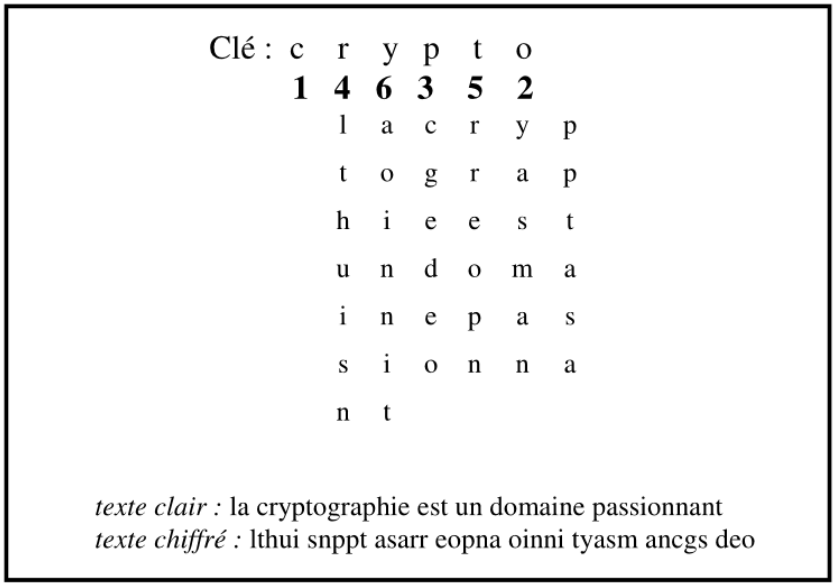
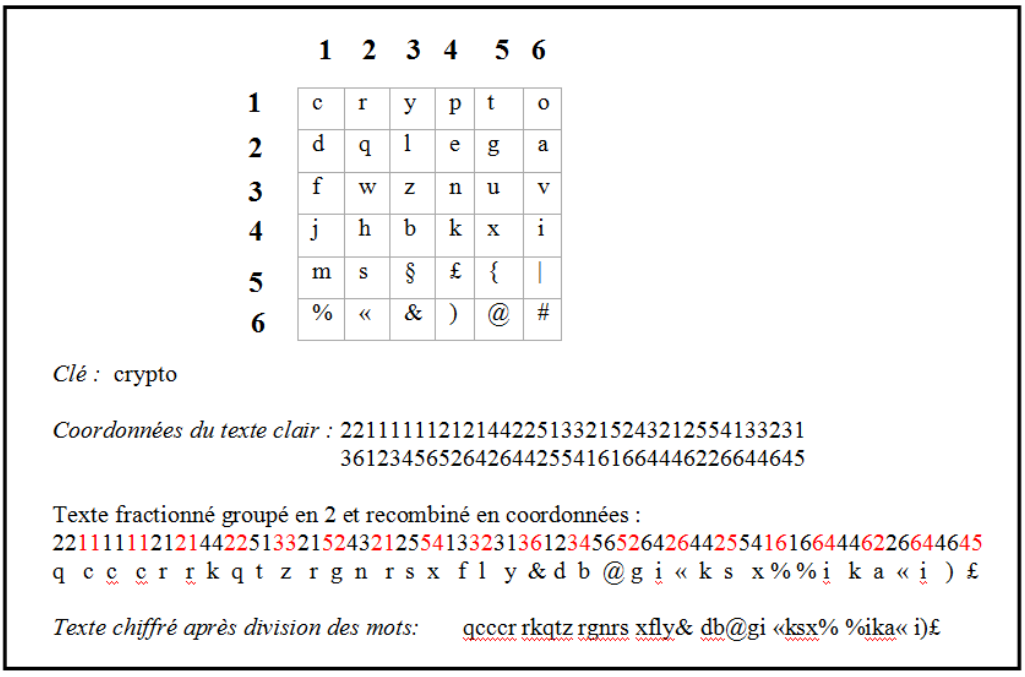



**Figure 2.3.** *La transposition complexe par colonnes.*

**c. Transposition par carré polybique :** Un mot clé secret est utilisé pour construire un alphabet dans un tableau, permettant d'extraire les coordonnées des lignes et des colonnes correspondant aux lettres du texte à chiffrer. Ainsi, chaque lettre du texte en clair est représentée par deux chiffres écrits verticalement sur deux lignes. L'étape qui suit, consiste à concaténer les deux lignes obtenues précédemment pour obtenir une seule ligne de chiffres, puis à recombiner ces chiffres deux par deux. Ces nouvelles combinaisons de chiffres représentent les coordonnées de lignes et de colonnes du texte chiffré.

Sur le même exemple de texte clair pris pour illustrer les façons précédentes de chiffrer (la cryptographie est un domaine passionnant), une illustration de ce mode de chiffrer est présentée dans la figure suivante.



| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **1** | c | r | y | p | t | o |
| **2** | d | q | l | e | g | a |
| **3** | f | w | z | n | u | v |
| **4** | j | h | b | k | x | i |
| **5** | m | s | § | £ | { | \| |
| **6** | % | « | & | ) | @ | # |

*Clé :* crypto

*Coordonnées du texte clair :* 2211111121214422513321524321255413323136123456526426442554161664446226644645
3612345652642644255416166444622664464 5

Texte fractionné groupé en 2 et recombiné en coordonnées :
22111111212144225133215243212554133231361234565264264425541616644462266446 45
q c c r r k q t z r g n r s x f l y & d b @ g i « k s x % % i k a « i ) £

*Texte chiffré après division des mots:* qcccr rkqtz rgnrs xfly& db@gi «ksx% %ika« i)£

**Figure 2.4.** *Transposition par carré polybique.*

Après la description de ces deux modes de cryptage (substitution et transposition), il est clair que les transpositions sont un peu plus sûres que les substitutions, mais elles ne fonctionnent que sur des messages à chiffrer d'une longueur limitée, en plus qu'elles sont plus gourmandes en mémoire ; ce qui limite leur utilisation dans les algorithmes.

# *Chapitre III*
# *La cryptographie moderne : Principes de base*

## 1. Introduction

Avec le développement des ordinateurs, les techniques de cryptographie ont clairement évolué, mettant à la touche ainsi les méthodes de cryptage manuel. Malgré ça, les procédés de substitution et de transposition restent toujours d'actualité mais en manipulant, cette fois-ci, des séquences de bits du fait que les ordinateurs ne manipulent que des données numériques ce qui rend les techniques de chiffrement actuelles plus sûres, voir même incassables pour certaines techniques, ou du moins prendraient des millions d'années avec la puissance actuelle des meilleurs supercalculateurs. D'autre part, il fait que maintenant les algorithmes ne sont plus cachés, mais au contraire sont connus de tous et leur sécurité est liée seulement aux clés utilisées.

La cryptographie moderne se scinde en deux parties nettement différenciées :

- ✓ La **cryptographie à clef secrète**, ou encore appelée **symétrique**;
- ✓ et la **cryptographie à clef publique**, dite également **asymétrique**.

La première, qui est la cryptographie symétrique, est la plus ancienne, et on peut la faire remonter à l'Égypte de l'an 2000 avant. J.C; la seconde, qui est la cryptographie asymétrique, remonte à l'article de W. Diffie et M. Hellman, « New directions in cryptography » daté de 1976.

## 2. La cryptographie symétrique

### 2.1. Principe

Les algorithmes de chiffrement à clé secrète (ou symétriques ou encore dits conventionnels) sont ceux pour lesquels l'émetteur et le destinataire partagent une même clé secrète, autrement dit, les clefs de chiffrement et de déchiffrement sont identiques. L'emploi d'un algorithme à clé secrète lors d'une communication nécessite donc l'échange préalable d'un secret entre les deux protagonistes à travers un canal sécurisé ou au moyen d'autres techniques cryptographiques.

Un paramètre essentiel pour la sécurité d'un système à clé secrète est la taille de l'espace des clefs. En effet, il est toujours possible de mener sur un algorithme de chiffrement, une attaque dite exhaustive pour retrouver la clé. Cette attaque consiste simplement à énumérer toutes les clefs possibles du système et à essayer d'utiliser chacune d'entre elles pour décrypter un message chiffré. Si l'espace des clefs correspond à l'ensemble des mots de k bits, le nombre de tentatives d'attaque k-1 exhaustive en vue de décrypter le message chiffré est égal à 2. Donc, pour pénaliser une telle attaque, il faut que l'espace des clés soit suffisamment grand. À titre d'exemple et en janvier 1998, une telle attaque a été réalisée contre le cryptosystème DES utilisant une clef secrète de 56 bits, en 39 jours sur 10000 Pentium en parallèle, puis en 56 heures en juillet 1998 à l'aide d'une machine dédiée comportant 1500 composants DES.

Il existe d'autres types d'attaques sur les systèmes de chiffrement à clé secrète dont la plupart consistent à exploiter certaines structures particulières de l'algorithme ou certaines caractéristiques statistiques dans la distribution des couples de textes clairs-chiffrés. Les plus connues sont la cryptanalyse différentielle, inventée par les Israéliens Biham et Shamir en 1991, et la cryptanalyse linéaire dont le principe a été initialement développé par Gilbert, Chassé et Tardy-Corfdir sur le chiffrement FEAL-8 puis appliqué au DES par le Japonais Matsui.

De façon générale, on considère qu'un chiffrement à clé secrète présente une bonne sécurité s'il n'existe pas d'attaque dont la complexité soit inférieure à celle de la recherche exhaustive. Et d'après les constatations empiriques qui sont la seule mesure de sécurité de ces systèmes, ils sont difficiles à cryptanalyser. Donc, la recherche en cryptographie symétrique se caractérise naturellement par

l'enchaînement de phases de défense et d'attaque - on s'endort cryptographe et se réveille cryptanalyste, selon le bon mot de Marc Girault.

Les algorithmes symétriques sont de deux types :

- ✓ Les algorithmes de **chiffrement en continu**, qui agissent sur le texte en clair un bit à la fois. Ce mode de chiffrement est encore appelé **chiffrement en flux** (*Stream cipher* en anglais) ;
- ✓ Les algorithmes de **chiffrement par blocs** (*Bloc cipher* en anglais), qui opèrent sur le texte en clair par groupes de bits appelés blocs.

Comme algorithmes de chiffrement à clé secrète on peut citer : DES, 3DES, AES, Blowfish, RC (Rivest Cipher), CSC (CS Cipher), IDEA, etc.

### 2.2. Chiffrement par blocs

Un procédé de chiffrement par blocs est un cryptosystème à clef secrète avec lequel on chiffre des messages M qui sont des blocs de n bits (où n est un entier fixé tel que 128, 256, ... par exemple). Un tel cryptosystème produit des messages chiffrés C qui sont aussi des blocs de n bits. La clef secrète K quant à elle est un bloc de *l* bits. Il est parfois possible que $l \neq n$.

En conclusion les messages clairs M et chiffrés C sont des éléments de $Z_2^l$ et les clefs secrètes sont des éléments de $Z_2$. (Rappelons que $Z_2 = \{0, 1\}$).

Le principe de fonctionnement d'un procédé de chiffrement par blocs **itéré** est d'appliquer plusieurs fois (c-à-d **d'itérer**) une même fonction sur le message clair afin d'obtenir le chiffré. De façon schématique, si E est une certaine fonction et M le message clair, l'idée est de calculer successivement :

$C_1 := E(M);$
$C_2 := E(C_1);$
: : :
$C_r := E(C_{r-1}).$

Dans un tel cryptosystème, *r* est le **nombre de tours** ou **nombre de rondes** et E s'appelle la **fonction de tour** ou **fonction de ronde**. Enfin $C_r$ est le message chiffré correspondant à M. En pratique, la fonction de tour E dépend d'un autre paramètre *k* de telle sorte que l'on calcule successivement :

$C_1 := E(M; k_1);$
$C_2 := E(C_1; k_2);$
: : :
$C_r := E(C_{r-1}; k_r).$

$k_i$ s'appelle la sous-clef de la $i^{ème}$ ronde.

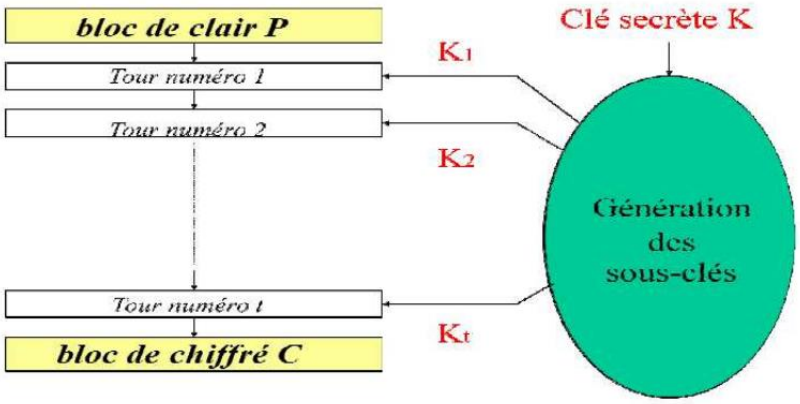



**Figure 3.1.** *Chiffrement par blocs.*

En règle générale, chaque **sous-clef** de tour $k_i$ est produite à partir de la clef secrète K, laquelle est i également appelée **clef principale**. La procédure permettant de générer les sous- clefs à partir de K s'appelle **un algorithme de dérivation de sous-clef**.

Une bonne sécurité est définie par une clé assez longue. Les clés très longues sont plus coûteuses en travail à cause notamment de leur génération, de leur transmission, de leur espace mémoire et de la difficulté de s'en rappeler (mots de passe). 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 modes de chiffrements par blocs**

Que ce soit pour DES ou des cryptosystèmes symétriques plus récents comme IDEA ou AES ou pour des cryptosystèmes asymétriques comme RSA ou ElGamal les clés sont de longueur fixée. Les messages eux peuvent avoir une longueur arbitraire. Pour adapter la taille du message à celle de la clef on décompose le message par blocs de taille fixe correspondant aux tailles des clés que l'on chiffre ensuite un à un et que l'on envoie successivement. Pour cela quatre modes essentiels de chiffrement par blocs sont possibles : ECB, CBC, CFB et OFB.

**a- Le mode ECB, Electronic Code Book**

Le mode ECB, Electronic Code Book, est le mode le plus simple. Le message, M, est découpé en blocs, $m_i$ , $i \geq 1$, et chaque bloc est crypté séparément par $c_i = E(m_i)$.

où $E = E_k$ dépend de la clé secrète k et $c_i$ est le bloc crypté correspondant. On procède donc suivant le schéma suivant :

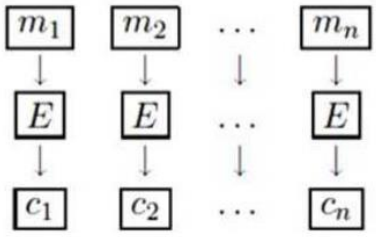



**Figure 3.2.** *Le mode ECB.*

On transmet :

$$c_1 \| c_2 \| \dots \| c_n$$

**b- Le mode CBC, Chiper Block Chaining**

Le mode CBC a été introduit pour qu'un bloc ne soit pas codé de la même manière s'il apparaît dans deux messages différents ou s'il apparaît deux fois dans un message.

Le message M est découpé en blocs $(m_i)_{i \geq 1}$, et chaque bloc est crypté de la manière suivante. On commence par choisir un bloc initial $c_0$. Chaque bloc clair $m_i$ est d'abord modifié en faisant un XOR de ce bloc avec le bloc crypté précédent, $c_{i-1}$, puis on crypte le résultat obtenu par XORisation avec la clé.

$$c_1 = E_k(m_1 \oplus c_0)$$
$$c_2 = E_k(m_2 \oplus c_1)$$
$$\vdots$$
$$c_i = E_k(m_i \oplus c_{i-1})$$
$$\vdots$$

Le déchiffrement nécessite de connaitre la fonction inverse de la fonction de codage $D_k = E_k^{-1}$ pour décrypter :

$$m_i = c_{i-1} \oplus D_k(c_i)$$

Ce système de transmission par blocs a une bonne sécurité et n'affeblit pas le cryptosystème, mais il nécessite de connaitre la fonction inverse $D_k$ de $E_k$.

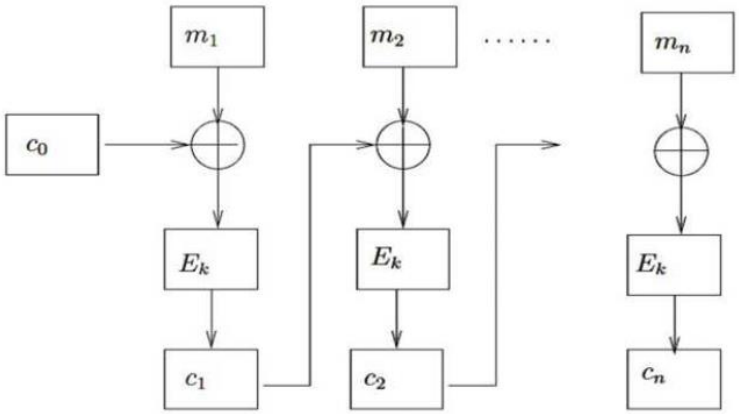



**Figure 3.3.** *Le mode CBC.*

**c- Le mode CFB, Cipher FeedBack**

Le mode CFB a été introduit pour ne pas avoir à calculer la fonction inverse, $D_k$ , de la fonction de chiffrage $E_k$.

Le principe est le même que celui du mode CBC. Le message M est découpé en blocs $(m_i)_{i \geq 1}$, et chaque bloc est crypté de la manière suivante. On commence par choisir un bloc initial $m_0$, choisit suivant les mêmes principes que le bloc $c_0$ en mode CBC.

Chaque bloc clair $m_i$ est XORé avec le crypté du bloc de sortie précédent, $c_{i-1}$, suivant le schéma :

$$c_1 = m_1 \oplus E_k(c_0)$$
$$c_2 = m_2 \oplus E_k(c_1)$$
$$\vdots$$
$$c_i = m_i \oplus E_k(c_{i-1})$$
$$\vdots$$

Ce mode est moins sûr que le CBC et est utilisé par exemple pour les cryptages réseaux. L'intérêt est que le déchiffrement ne nécessite pas de calculer $D_k$, en effet :

$$m_i = c_i \oplus E_k(c_{i-1})$$

(se souvenir que l'on calcule modulo 2 sans retenue, bit à bit) d'où un gain de temps.

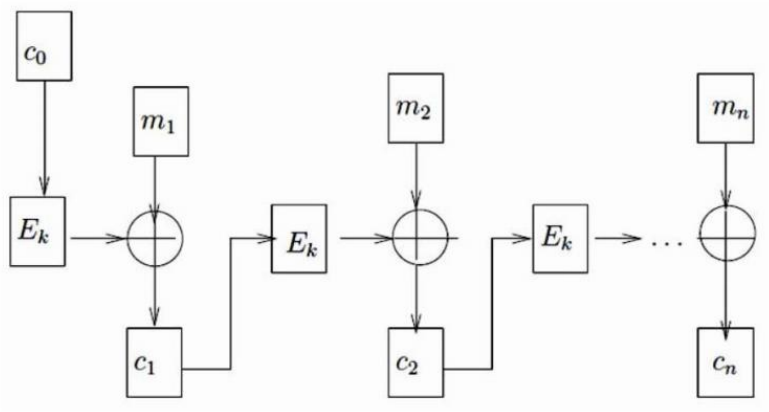



**Figure 3.4.** *Le mode CFB.*

### d- Le mode OFB, Output FeedBack

Le mode OFB est une variante de CFB qui permet d'avoir un cryptage et un décryptage totalement symétrique :

$$z_i = E_k(z_{i-1})\,;$$
$$c_i = m_i \oplus z_i$$

On transmet le message :

$$c_0 \,||\, c_1 \,||\, \ldots \,||\, c_n$$

Ce mode est utilisé par exemple pour le cryptage satellites et se déchiffre par :

$$z_i = E_k(z_{i-1})\,;$$
$$m_i = c_i \oplus z_i$$

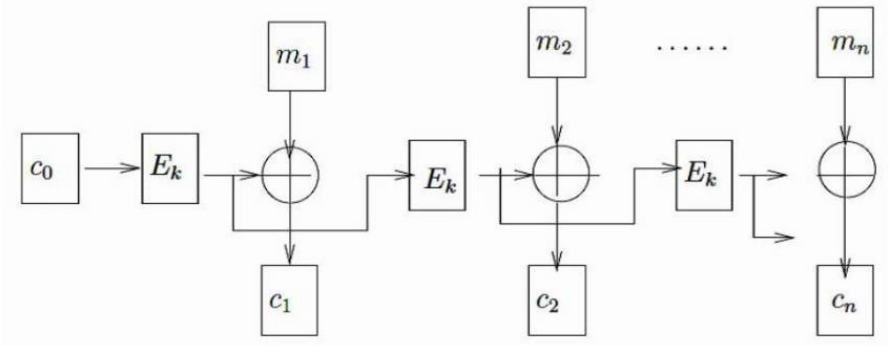



**Figure 3.5.** *Le mode OFB.*

Sa sécurité est équivalente à celle du mode CFB.

### e- Le mode CTR, Counter-mode encryption

Le mode CTR est lui aussi totalement symétrique, mais en outre facilement parallélisable. Il utilise pour le chiffrement un compteur de valeur initiale T.

$$c_i = m_i \oplus E_k(T + i)$$

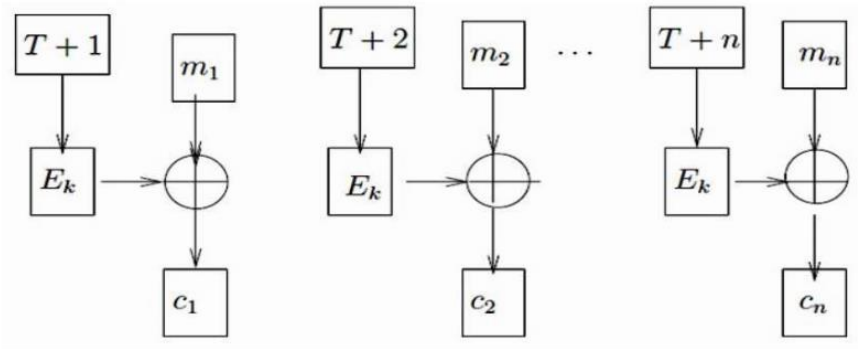



**Figure 3.6.** *Le mode CTR.*

Le déchiffrage est identique, et sa sûreté est équivalente à celle du mode CFB.

### 2.3. Schéma de Feistel

Un grand nombre de cryptosystèmes à clés secrètes ont une structure dit en **réseau de Feistel**. L'intérêt de ceux-ci est de pouvoir à partir de n'importe quelle fonction f construire une fonction de chiffrement. Insistons sur le fait que f est quelconque et donc n'est pas nécessairement bijective. Dans une construction de Feistel, le bloc d'entrée d'un round est séparé en deux parties.

- ✓ La fonction de chiffrement est appliquée sur la première partie du bloc,
- ✓ et l'opération binaire OU-Exclusif est appliquée sur la partie sortante de la fonction et la deuxième partie.
- ✓ Ensuite, les deux parties sont permutées.

Chiffrement :

$L1 = R0$

$R1 = L0 \oplus f(R0)$

Le déchiffrement est trivial :

$R0 = L1$

$L0 = R1 \oplus f(R0)$

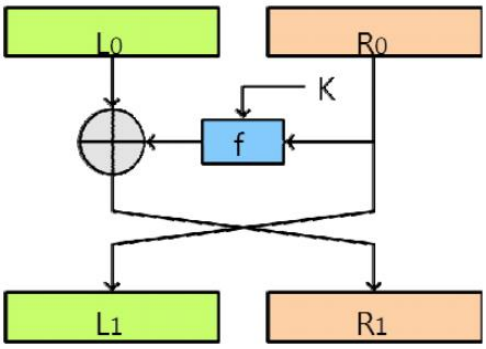



**Figure 3.7.** *Le schéma de Feistel.*

Le schéma de Feistel permet d'obtenir une bijection sur 2n bits, à partir d'une fonction non- bijective sur n bits. Un tour isolé donne une bijection simpliste, la complexité croît de façon exponentielle avec le nombre de tours.

## 3. La cryptographie asymétrique

### 3.1. Historique

Le concept de cryptographie à clé publique, qui est un autre nom de la cryptographie asymétrique, a été présenté pour la première fois par Whitfield Diffie et Martin Hellman dans leur article écrit pour le National Computer Conference en 1976, puis publié quelques mois plus tard dans New Directions in Cryptography sans pouvoir donner un exemple d'un système à clé publique. Il fallut attendre 1978 où la version académique du premier cryptosystème à clé publique a fait l'objet d'un article intitulé : « A Method for Obtaining Digital Signatures and Public-key Cryptosystems » écrit par Ronald Rivest, Adi Shamir, et Leonard Adleman. C'est le cryptosystème RSA dont l'appellation est tirée des trois noms de ses auteurs.

En réalité, James Ellis, qui travaillait au service du chiffre britannique (GCHQ, Government Communications Headquarters), avait eu cette idée un peu avant. En 1973, C.C. Cocks décrivit (pour le même service du chiffre) ce qu'on a appelé l'algorithme RSA. Enfin, en 1974, M. J. Williamson invente un protocole d'échange de clé très proche de celui de Diffie et de Hellman. Ces trouvailles n'ont été rendues publiques qu'en 1997 par le GCHQ.

### 3.2. Principe

La cryptographie à clé publique évite le partage d'un secret entre les deux interlocuteurs puisque, chaque utilisateur dispose d'un couple de clés : une clé publique qu'il met en général à disposition de tous dans un annuaire, et une clef secrète connue de lui seul. Ces deux clés, en plus d'être distinctes, elles ne peuvent se déduire l'une de l'autre. Alors, pour envoyer un message confidentiel à Bob, Alice chiffre le message clair à l'aide de la clé publique de Bob. Ce dernier, à l'aide de la clé secrète correspondante, sera le seul en mesure de déchiffrer le message reçu.

La notion primordiale sur laquelle repose le chiffrement à clé publique est celle de fonction à sens unique avec trappe. Sachons qu'une fonction est appelée à sens unique si elle est aisément calculée, mais difficile à inverser, ou plus exactement, infaisable en un temps réalisable avec une puissance de calcul raisonnable. Et une telle fonction sera dite à trappe, si le calcul de l'inverse devient facile dès que l'on possède une information supplémentaire qui est la trappe. Donc, la construction d'un système de chiffrement à clé publique à partir d'une telle fonction, sera une chose très simple où la procédure de chiffrement consiste simplement à appliquer la fonction au message clair. L'utilité d'utilisation d'une telle fonction, difficile à inverser si on ne connaît pas la trappe, dans le domaine de la cryptographie, réside dans le fait de rendre difficile la détermination du message en clair à partir du message chiffré sans connaître la clé secrète de déchiffrement. Cependant, la définition de ces fonctions particulières n'est pas aussi facile puisqu'elle s'appuie généralement sur des problèmes mathématiques réputés difficiles tel que le problème de la factorisation de grands nombres entiers. Mais, si quelqu'un trouve un jour le moyen de simplifier la résolution de l'un de ces problèmes, l'algorithme correspondant, c'est à dire construit autour de ce problème, finira par être abandonné.

Les systèmes asymétriques les plus connus sont : RSA, ElGamal, Rabin, McEliece, Courbes elliptiques (ECC), etc.

### 3.3. Applications

- **Transmission sécurisée de la clé symétrique**

Pour résoudre le problème d'échange de la clé secrète utilisée lors d'un chiffrement symétrique, le chiffrement asymétrique a été envisagé comme solution. Cette dernière consiste à chiffrer la clé secrète en utilisant un mécanisme de chiffrement asymétrique assurant, ainsi, un partage sécurisé de cette clé et évitant son interception par une personne tierce non autorisée. Donc, le chiffrement asymétrique intervient dans la seule phase d'échange de la clé symétrique, qui sera utilisée par la suite pour le reste de l'échange d'informations.

- **Mécanismes d'identification**

Le problème qui se pose avec le mode de chiffrement asymétrique est celui dit d'identification. Il est dû au fait que la clé publique est distribuée à toutes les personnes, ainsi, lors de la réception puis du déchiffrement d'un message chiffré, on a aucun moyen de vérifier avec certitude son origine. Afin de résoudre ce problème, on utilise des mécanismes d'identification permettant de garantir la provenance des informations chiffrées.

### 4. La cryptographie hybride

La cryptographie hybride consiste, comme son nom l'indique, en une association des deux techniques de cryptage précédentes où on code tout d'abord les données avec une clé privée dite clé de session, ensuite cette clé est cryptée à l'aide d'une clé publique classique. Dans cette politique de cryptage le choix de crypter la clé d'une manière publique au lieu de crypter les messages est du au fait que, la clé est souvent de petite taille par rapport aux données à chiffrer, donc, elle consomme beaucoup moins de temps lors de son chiffrement par rapport aux données. C'est pourquoi ces dernières sont chiffrées de manière symétrique. Ensuite, il ne reste qu'à transmettre le package contenant les données cryptées avec une clé privée, cryptée de son tour avec une clé publique. Une fois le package sera reçu, le récepteur procède inversement. Tout d'abord, il déchiffre la clé chiffrée à l'aide de sa clé privée pour obtenir la clé de session, qui sera utilisée, par la suite, pour déchiffrer les données chiffrées. Ainsi, les performances seront améliorées en associant la rapidité des systèmes de chiffrement symétriques et la bonne sécurisation des systèmes de chiffrement asymétriques.

Les logiciels comme PGP et GnuPG reposant sur ce concept permettent de combiner les avantages des deux systèmes.

### 5. Analogies

Les deux modes de cryptage moderne, qui sont le cryptage symétrique et le cryptage asymétrique, peuvent être comparés à des moyens physiques d'échange de messages confidentiels.

- ✓ Un système à clé secrète correspond à un coffre-fort, puisqu'avec ce dernier, et si Alice veut communiquer un message à Bob en le déposant dans ce coffre, ils doivent tout d'abord partager une information secrète qui est la combinaison indispensable pour toute opération de dépôt ou de récupération de documents. Donc, cette information doit être connue par Alice et Bob seulement.
- ✓ Du coté des systèmes à clé publique, ils correspondent à une boite aux lettres. Dans ce cas, toute personne souhaitant transmettre un document confidentiel à Bob n'a qu'à le mettre dans sa boite aux lettres qui ferme à clé. Seul Bob possédant cette clé peut ouvrir la boite et lire les documents qui lui sont destinés.

### 6. Comparaison entre les cryptosystèmes symétriques et asymétriques

Le tableau ci-dessous présente une comparaison entre les systèmes de chiffrement symétriques et les systèmes de chiffrement asymétriques, en énumérant les principaux avantages et inconvénients de chaque mode de cryptage.

| Méthode | Exemples | Avantages | Inconvénients |
|---|---|---|---|
| **À clefs Secrètes** | DES, AES | ▪ Rapidité de calcul en général (dépend de la taille de la clé).<br>▪ Adaptée au cryptage de flux de données. | ▪ Moins sécurisé (DES).<br>▪ Problème de communication de clefs entre émetteur et récepteur.<br>▪ Une clé pour chacun des correspondants. |
| **À clefs Publiques** | RSA, ElGamal | ▪ Très sécurisée à cause de l'utilisation de deux clés distinctes, l'une ne permettant pas de retrouver l'autre.<br>▪ Permet la signature électronique. | ▪ Lente.<br>▪ Problèmes de gestion de clefs publiques. |

## 7. Clé publique ou clé secrète, un compromis

La question qui se pose à ce niveau est : Dans quels cas on utilise le chiffrement symétrique ? Et dans quels autres cas le chiffrement asymétrique est conseillé ?

En effet, et d'après la table comparative présentée juste avant, il est clair que les systèmes de chiffrement à clé publique sont très lents par rapport aux systèmes de chiffrement à clé privée. Alors que, l'algorithme de chiffrement ne doit pas être le facteur limitant à notre époque où la vitesse de transmission de l'information constitue un enjeu crucial.

De plus, et à partir des descriptions des systèmes de chiffrement présentées précédemment, on arrive à constater que la taille des clés nécessaire en cryptographie à clé publique pour assurer une sécurité satisfaisante est plus grande que la taille des clés en cryptographie à clé secrète. En fait, la notion et l'importance de la taille de clé pour assurer la sécurité ne sont légitimes que dans le cas de la clé secrète, puisque ces systèmes reposent sur l'hypothèse que les seules attaques possibles sont les attaques exhaustives. Mais, dans le cas de la clé publique, la taille de clé n'a de pertinence que lorsqu'on considère le même système. Donc, le fait de dire que RSA de 512 bits est bien moins sûr qu'un AES de 128 bits, n'a aucune signification. Cependant, la seule mesure légitime pour évaluer un cryptosystème à clé publique est la complexité de la meilleure attaque connue.

# *Chapitre IV*
# *Cryptosystèmes symétriques*

Dans ce chapitre, nous allons présenter, plus ou moins en détails, les plus fameux des algorithmes de chiffrement s'inscrivant sous le mode de chiffrement à clé privée ou encore dit symétrique.

## 1. Masque jetable (one-time pad)

Cet algorithme, inventé en 1917 et connu aussi sous le nom de chiffre de Vernam, est un algorithme de chiffrement prouvé inconditionnellement sûr. Dans ce système, la clé possède la même taille que le texte à chiffrer et est appelée masque jetable. « Masque », car cette clef est combinée par ou exclusif avec le texte en clair pour obtenir le texte chiffré ; « jetable », car une clef ne doit servir qu'une seule fois.

La sécurité de ce système repose sur la génération complètement aléatoire de la clé, ce qui représente le grand avantage de ce système. Par conséquence, si le cryptanalyste ne possède aucune information sur laquelle son attaque va appuyer, tous les masques seront équiprobables. En effet, si M est le message à chiffrer, C le message chiffré correspondant et K le masque jetable, nous avons :

$$C = M \oplus K$$

Supposons que le cryptanalyste connaisse C ; alors :

$$\forall M' , \ \exists K' : C = M' \oplus K'$$

Donc, c'est : $K' = M' \oplus C$ $\Rightarrow$ tous les messages en clair sont équiprobables

$\Rightarrow$ Impossible de savoir quel est le bon texte en clair sans connaître la clé.

Malgré cela, ce système est limité à des applications extrêmes et ne peut être utilisé pour chiffrer des flux importants de données à cause de la taille de la clé nécessitant des générateurs aléatoires pour sa création.

## 2. DES

Le 15 mai 1973, le National Bureau of Standards des Etats-Unis lança un appel d'offre de système cryptographique dans le Federal Register, qui est un journal officiel américain. Cet appel déboucha sur le **Standard de chiffrement de données DES** qui est devenu le système cryptographique le plus utilisé dans le monde. IBM développa initialement DES comme modification d'un système antérieur appelé **LUCIFER**. DES, ou encore appelé **DEA (Data Encryption Algorithm)**, fut publié dans le Federal Register le 17 mars 1975. Après un nombre considérable de débats publics, on adopta DES comme standard pour des applications non classifiées le 15 janvier 1977. Depuis son adoption, DES a été réévalué par le National Bureau of Standards tous les cinq ans, approximativement. La plus récente révision date de janvier 1994 où il a été renouvelé jusqu'en 1998. Il est prévu que le standard s'arrête à cette date.

### 2.1. Description

Une description complète de DES est donnée dans le Federal Information Processing Standard Publication (FIPS) № 46 du 15 janvier 1977. C'est le cryptosystème qui a été le plus utilisé, de plus, il a bien résisté aux efforts des cryptanalystes pendant 25 ans.

Ce cryptosystème est un système de chiffrement *par blocs*. Cela signifie, comme nous l'avons déjà présenté dans le chapitre précédent, qu'il ne chiffre pas les données caractère par caractère, mais

il découpe le texte clair en blocs de 64 bits, dans le cas de ce cryptosystème. Ces blocs sont chiffrés séparément, puis concaténés. Ainsi, les données en entrée de cet algorithme seront des blocs de 64 bits du texte clair, et les données en sortie seront aussi des blocs de 64 bits de texte chiffré. C'est seulement la courte longueur de la clé, utilisée lors du chiffrement qui est de 56 bits, qui ne lui permet pas, aujourd'hui, d'assurer un bon niveau de sécurité, bien qu'elle a été largement suffisante au moment de sa conception.

L'algorithme est relativement simple puisqu'il combine des permutations et des substitutions. On donne tout d'abord une description générale de ce système qui se déroule en trois étapes :

**1)** Etant donné un bloc de texte clair *x*. Une chaîne de bits $x_0$ est construite en changeant l'ordre des bits de *x* suivant une *permutation initiale* IP fixée. On écrit :

$$x_0 = IP(x) = L_0R_0$$

Où $L_0$ contient les 32 premiers bits de la chaîne $x_0$ et $R_0$ contient les 32 bits restants.

**2)** 16 itérations (ou 16 tours) d'une certaine fonction sont effectuées. Chaque tour suit le même schéma qui consiste à prendre en entrée 32 bits : $L_{i-1}$ et $R_{i-1}$ du tour précédent et de produire de nouveau 32 bits $L_i$ et $R_i$ de la manière suivante :

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

**3)** Après le dernier tour, les moitiés gauche et droite ($L_{16}$ et $R_{16}$) sont échangées puis le texte sera permuté bit à bit par $IP^{-1}$ pour obtenir le bloc de texte chiffré *y*. Plus formellement, *y* s'obtient comme suit :

$$y = IP^{-1}(R_{16}L_{16}).$$

Cette description peut être résumée par le schéma général illustré par la figure suivante (figure 4.1), où on a seulement représenté quelques-unes des 16 étapes.

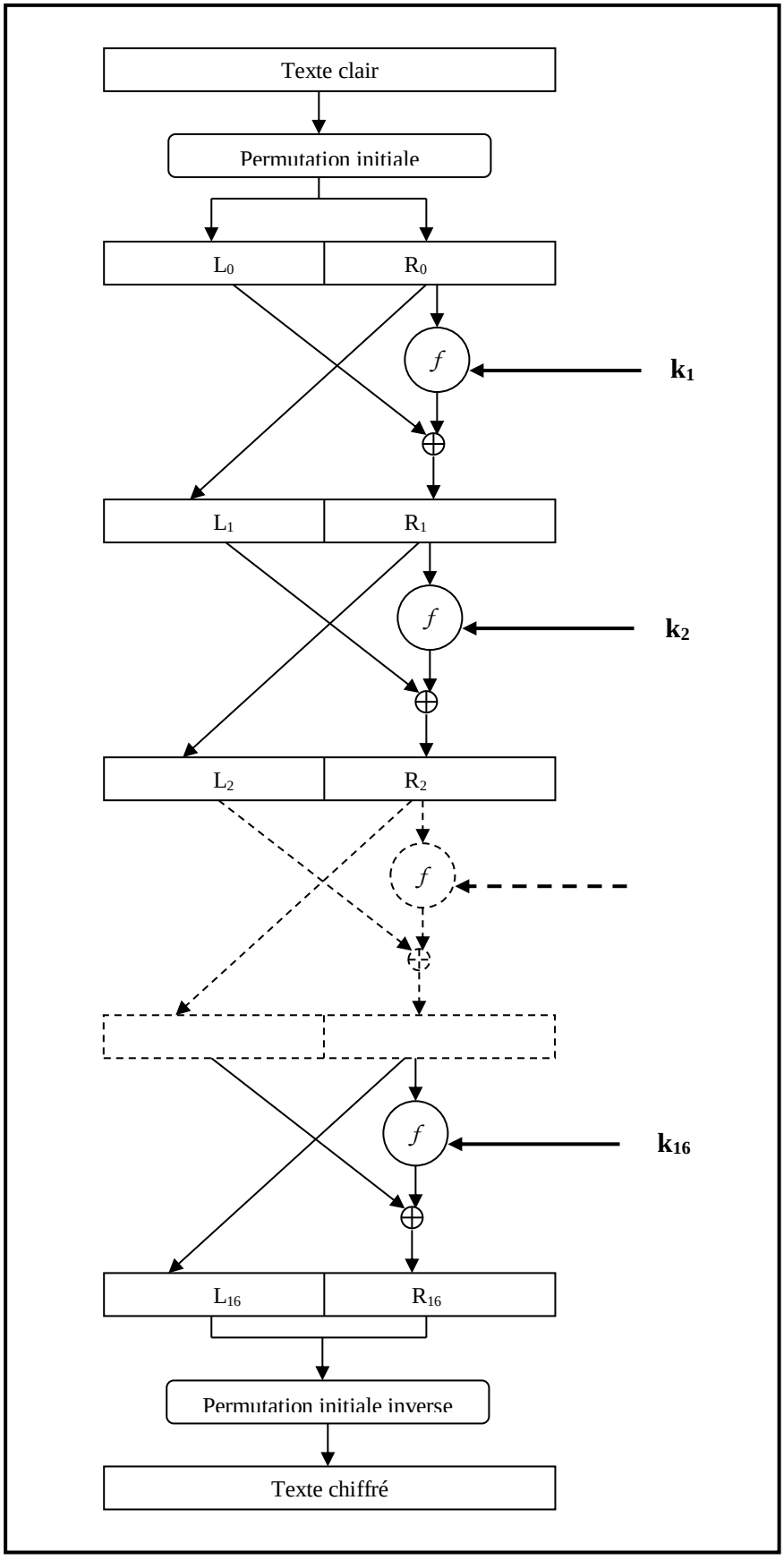



**Figure 4.1.** *Schéma général de DES.*

Nous présentons maintenant, plus ou moins en détail, chacune des étapes du système.

**a. Permutation initiale et permutation initiale inverse**

Dans un premier temps, chaque bit d'un bloc est soumis à la permutation initiale, pouvant être représentée par la matrice de permutation initiale (notée IP) donnée par la figure 2. Cette matrice de permutation indique, en parcourant la matrice de gauche à droite puis de haut en bas, que le 58ème bit du bloc de texte de 64 bits se retrouve en première position, le 50ème en seconde position et ainsi de suite.

A la fin des itérations, les deux blocs $L_{16}$ et $R_{16}$ sont "recollés", puis soumis à la permutation initiale inverse. Le résultat en sortie est un texte codé de 64 bits !

Au fait, lors du chiffrement d'un bloc DES, ces deux opérations qui sont l'application d'une permutation initiale et de son inverse à la fin, n'ont aucun rôle dans la sécurité de l'algorithme.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

**Permutation initiale**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

**Permutation initiale inverse**

**Figure 4.2.** *La permutation initiale et son inverse.*

**b. Division en blocs de 32 bits**

Une fois la permutation initiale réalisée, le bloc de 64 bits est scindé en deux blocs de 32 bits, désignées par $L_0$ et $R_0$

**c. Rondes (Itérations)**

À chacune des 16 itérations, on calcul deux nouveaux groupes de 32 bits $L_i$ et $R_i$ en fonction des deux groupes, notés $L_{i-1}$ et $R_{i-1}$, de l'itération précédente. Pour cela, on utilise une clé intermédiaire $k_i$ de 48 bits, obtenue par diversification à partir de la clé k de 56 bits comme le montre la figure 4.3.

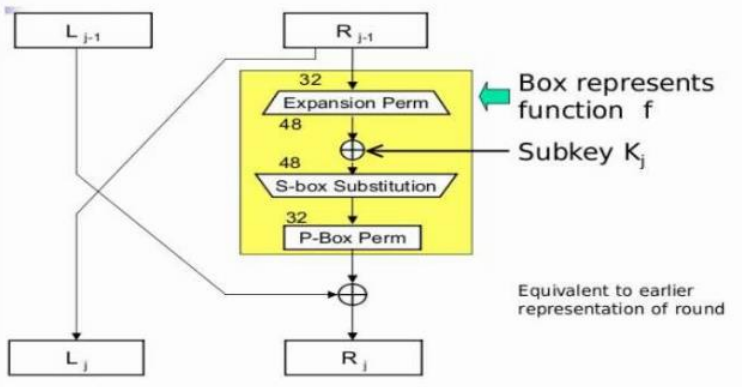



**Figure 4.3.** *Représentation détaillée d'une ronde.*

D'après la figure 4.3, on constate que la fonction f utilise deux arguments ayant des tailles différentes : $R_{i-1}$ de 32 bits et $k_i$ de 48 bits. Ainsi, $R_{i-1}$ sera expansé en 48 bits en redoublant, au hasard, 16 bits parmi les 32 bits initiaux. Les points suivants détaillent les différentes opérations nécessaires.

- **Fonction d'expansion**

Les 32 bits du bloc droit R sont étendus à 48 bits grâce à une table (matrice) appelé table d'expansion donnée par la figure 4.5 (notée E), dans laquelle les 48 bits sont mélangés et 16 d'entre eux sont dupliqués comme le montre la figure 4.4.

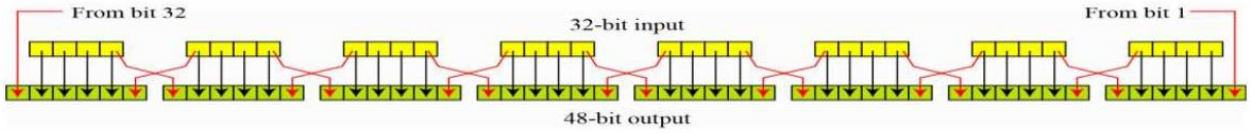



**Figure 4.4.** *Fonction d'expansion.*

| E | | | | | |
|---|---|---|---|---|---|
| **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** |

**Figure 4.5.** *Matrice d'expansion.*

- **OU exclusif avec la clé**

La matrice résultante de 48 bits est appelée E[R]. L'algorithme DES procède ensuite à un OU exclusif entre la première clé $K_j$ et E[R]. Le résultat de ce OU exclusif est une matrice de 48 bits que nous appellerons R par commodité (il ne s'agit pas du R de départ !).

- **Fonction de substitution**

R est ensuite divisée en 8 blocs de 6 bits. Chacun de ces blocs passe par des **fonctions de sélection** (appelées parfois **boîtes de substitution** ou **fonctions de compression**), notées généralement $\mathbf{S_i}$.

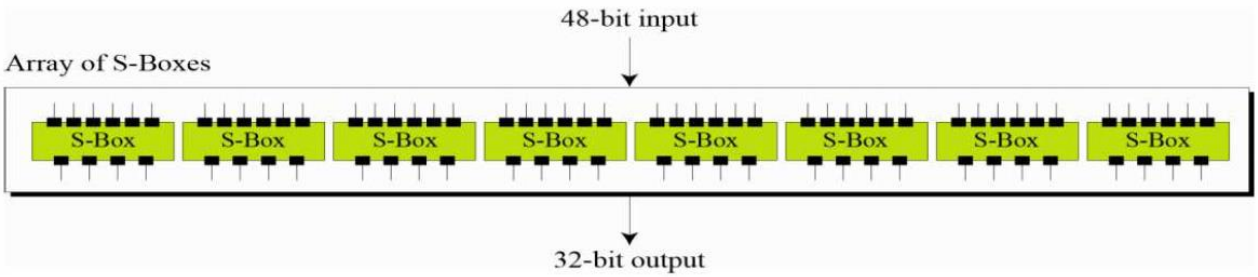



**Figure 4.6.** *Ensemble des huit S-Boxes.*

Les bits 1 et 6 de chaque bloc de 6 bits détermine (en binaire) la ligne de la fonction de sélection, les autres bits (respectivement 2, 3, 4 et 5) déterminent la colonne. Grâce à cette information, la fonction de sélection "sélectionne" une valeur codée sur 4 bits.

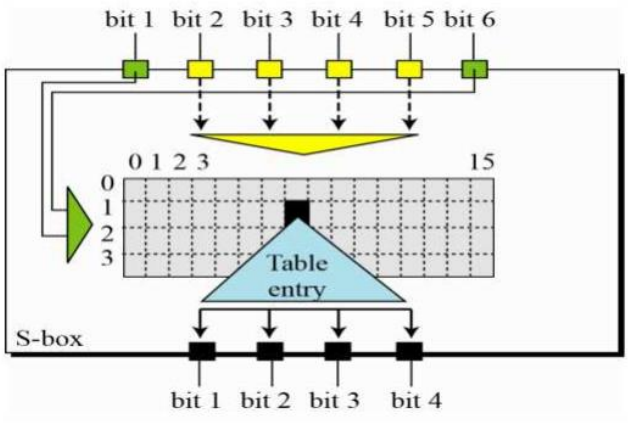



**Figure 4.7.** *Fonctionnement d'un S-Boxe.*

Pour mieux comprendre le fonctionnement d'un S-Boxe, nous allons le dérouler sur un exemple. Soit le bloc 101110. Les bits 1 et 6 donnent 10, c'est-à-dire 2 en binaire. Les bits 2, 3, 4 et 5 donnent 0111, soit 7 en binaire. Le résultat de la fonction de sélection est donc la valeur situé à la ligne n°2, dans la colonne n°7 de S qui est une matrice de 4 par 16, illustrée à travers la figure 4.8. Il s'agit de la 1 valeur 11, soit en binaire 1011.

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

$S_1$

**Figure 4.8.** *Le S-boxe $S_1$.*

Chacun des 8 blocs de 6 bits est passé dans la fonction de sélection correspondante, ce qui donne en sortie 8 valeurs de 4 bits chacune. Voici les sept autres fonctions de sélection :

| $S_2$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 15 | 1 | 8 | 14 | 6 | 11 | 3 | 4 | 9 | 7 | 2 | 13 | 12 | 0 | 5 | 10 |
| **1** | 3 | 13 | 4 | 7 | 15 | 2 | 8 | 14 | 12 | 0 | 1 | 10 | 6 | 9 | 11 | 5 |
| **2** | 0 | 14 | 7 | 11 | 10 | 4 | 13 | 1 | 5 | 8 | 12 | 6 | 9 | 3 | 2 | 15 |
| **3** | 13 | 8 | 10 | 1 | 3 | 15 | 4 | 2 | 11 | 6 | 7 | 12 | 0 | 5 | 14 | 9 |

| $S_3$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 10 | 0 | 9 | 14 | 6 | 3 | 15 | 5 | 1 | 13 | 12 | 7 | 11 | 4 | 2 | 8 |
| **1** | 13 | 7 | 0 | 9 | 3 | 4 | 6 | 10 | 2 | 8 | 5 | 14 | 12 | 11 | 15 | 1 |
| **2** | 13 | 6 | 4 | 9 | 8 | 15 | 3 | 0 | 11 | 1 | 2 | 12 | 5 | 10 | 14 | 7 |
| **3** | 1 | 10 | 13 | 0 | 6 | 9 | 8 | 7 | 4 | 15 | 14 | 3 | 11 | 5 | 2 | 12 |

| $S_4$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7 | 13 | 14 | 3 | 0 | 6 | 9 | 10 | 1 | 2 | 8 | 5 | 11 | 12 | 4 | 15 |
| **1** | 13 | 8 | 11 | 5 | 6 | 15 | 0 | 3 | 4 | 7 | 2 | 12 | 1 | 10 | 14 | 9 |
| **2** | 10 | 6 | 9 | 0 | 12 | 11 | 7 | 13 | 15 | 1 | 3 | 14 | 5 | 2 | 8 | 4 |
| **3** | 3 | 15 | 0 | 6 | 10 | 1 | 13 | 8 | 9 | 4 | 5 | 11 | 12 | 7 | 2 | 14 |

| $S_5$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2 | 12 | 4 | 1 | 7 | 10 | 11 | 6 | 8 | 5 | 3 | 15 | 13 | 0 | 14 | 9 |
| **1** | 14 | 11 | 2 | 12 | 4 | 7 | 13 | 1 | 5 | 0 | 15 | 10 | 3 | 9 | 8 | 6 |
| **2** | 4 | 2 | 1 | 11 | 10 | 13 | 7 | 8 | 15 | 9 | 12 | 5 | 6 | 3 | 0 | 14 |
| **3** | 11 | 8 | 12 | 7 | 1 | 14 | 2 | 13 | 6 | 15 | 0 | 9 | 10 | 4 | 5 | 3 |

| $S_6$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 12 | 1 | 10 | 15 | 9 | 2 | 6 | 8 | 0 | 13 | 3 | 4 | 14 | 7 | 5 | 11 |
| **1** | 10 | 15 | 4 | 2 | 7 | 12 | 9 | 5 | 6 | 1 | 13 | 14 | 0 | 11 | 3 | 8 |
| **2** | 9 | 14 | 15 | 5 | 2 | 8 | 12 | 3 | 7 | 0 | 4 | 10 | 1 | 13 | 11 | 6 |
| **3** | 4 | 3 | 2 | 12 | 9 | 5 | 15 | 10 | 11 | 14 | 1 | 7 | 6 | 0 | 8 | 13 |

| $S_7$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 4 | 11 | 2 | 14 | 15 | 0 | 8 | 13 | 3 | 12 | 9 | 7 | 5 | 10 | 6 | 1 |
| **1** | 13 | 0 | 11 | 7 | 4 | 9 | 1 | 10 | 14 | 3 | 5 | 12 | 2 | 15 | 8 | 6 |
| **2** | 1 | 4 | 11 | 13 | 12 | 3 | 7 | 14 | 10 | 15 | 6 | 8 | 0 | 5 | 9 | 2 |
| **3** | 6 | 11 | 13 | 8 | 1 | 4 | 10 | 7 | 9 | 5 | 0 | 15 | 14 | 2 | 3 | 12 |

| $S_8$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 13 | 2 | 8 | 4 | 6 | 15 | 11 | 1 | 10 | 9 | 3 | 14 | 5 | 0 | 12 | 7 |
| **1** | 1 | 15 | 13 | 8 | 10 | 3 | 7 | 4 | 12 | 5 | 6 | 11 | 0 | 14 | 9 | 2 |
| **1** | 7 | 11 | 4 | 1 | 9 | 12 | 14 | 2 | 0 | 6 | 10 | 13 | 15 | 3 | 5 | 8 |
| **1** | 2 | 1 | 14 | 7 | 4 | 10 | 8 | 13 | 15 | 12 | 9 | 0 | 3 | 5 | 6 | 11 |

**Figure 4.9.** *Les S-boxes $S_2$, $S_3$, $S_4$, $S_5$, $S_6$, $S_7$, $S_8$.*

Chaque bloc de 6 bits est ainsi substitué en un bloc de 4 bits. Ces bits sont regroupés pour former un bloc de 32 bits.

- **Permutation**

Le bloc de 32 bits obtenu est enfin soumis à une permutation P dont voici la matrice ou table :

| P | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 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 |

**Figure 4.10.** *Table de permutation.*

Cette succession d'opérations constituant la fonction f, peut être schématisée à travers la figure 4.11.

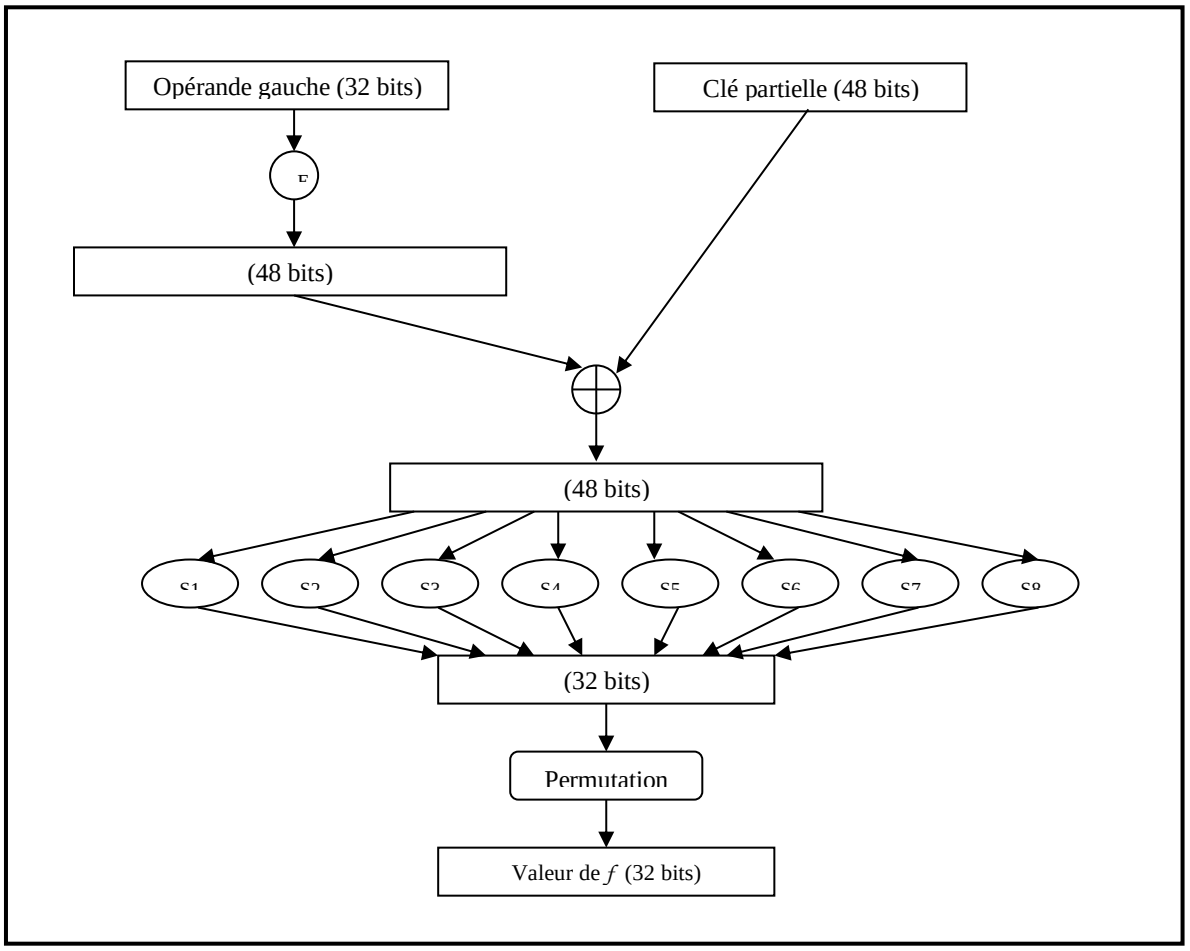



**Figure 4.11.** *Schéma de la fonction f.*

- **OU exclusif**

L'ensemble de ces résultats en sortie de P est soumis à un OU Exclusif avec le bloc L de départ (comme indiqué sur le premier schéma) pour donner les 32 bits de droite, tandis que le R initial donne les 32 bits de la partie gauche.

- **Itération**

L'ensemble des étapes précédentes (rondes) est réitéré 16 fois.

***Remarque :***

Le déchiffrement suit le même algorithme avec la même clef K. Seules les sous-clés sont appliquées dans le sens inverse.

### 2.2. Génération de clés

Etant donné que l'algorithme du DES présenté ci-dessus est public, toute la sécurité repose sur la complexité des clés de chiffrement. La clé de chiffrement est de longueur 64, dont 56 seulement serviront au chiffrement proprement dit.

Les sous-clés $K_1$, $K_2$, ... , $K_{16}$ , de 48 bits chacune, sont obtenues de la clé initiale K à l'aide de l'algorithme suivant :

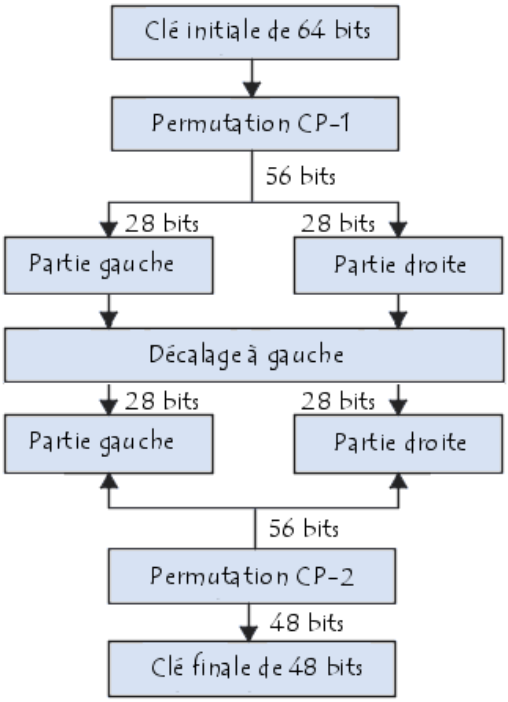



**Figure 4.12.** *Algorithme d'obtention d'une clé DES.*

Le détail de l'opération de génération de clés DES est le suivant :

**1°** Dans un premier temps les bits de parité de la clé sont éliminés afin d'obtenir une clé d'une longueur utile de 56 bits.

**2°** La première étape consiste en une permutation notée CP-1 dont la matrice est présentée ci- dessous :

**CP-1**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

**3°** La clé est ensuite divisée en deux blocs gauche et droit de 28 bits chacun. Ces deux blocs subissent ensuite une rotation à gauche.

| Round | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit shifts | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

**4°** Les 2 blocs de 28 bits sont ensuite regroupés en un bloc de 56 bits. Celui-ci passe par une permutation, notée CP-2, fournissant en sortie un bloc de 48 bits, représentant la clé $K_i$.

**CP-2**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 |

Des itérations de l'algorithme permettent de donner les 16 clés $K_1$ à $K_{16}$ utilisées dans DES.

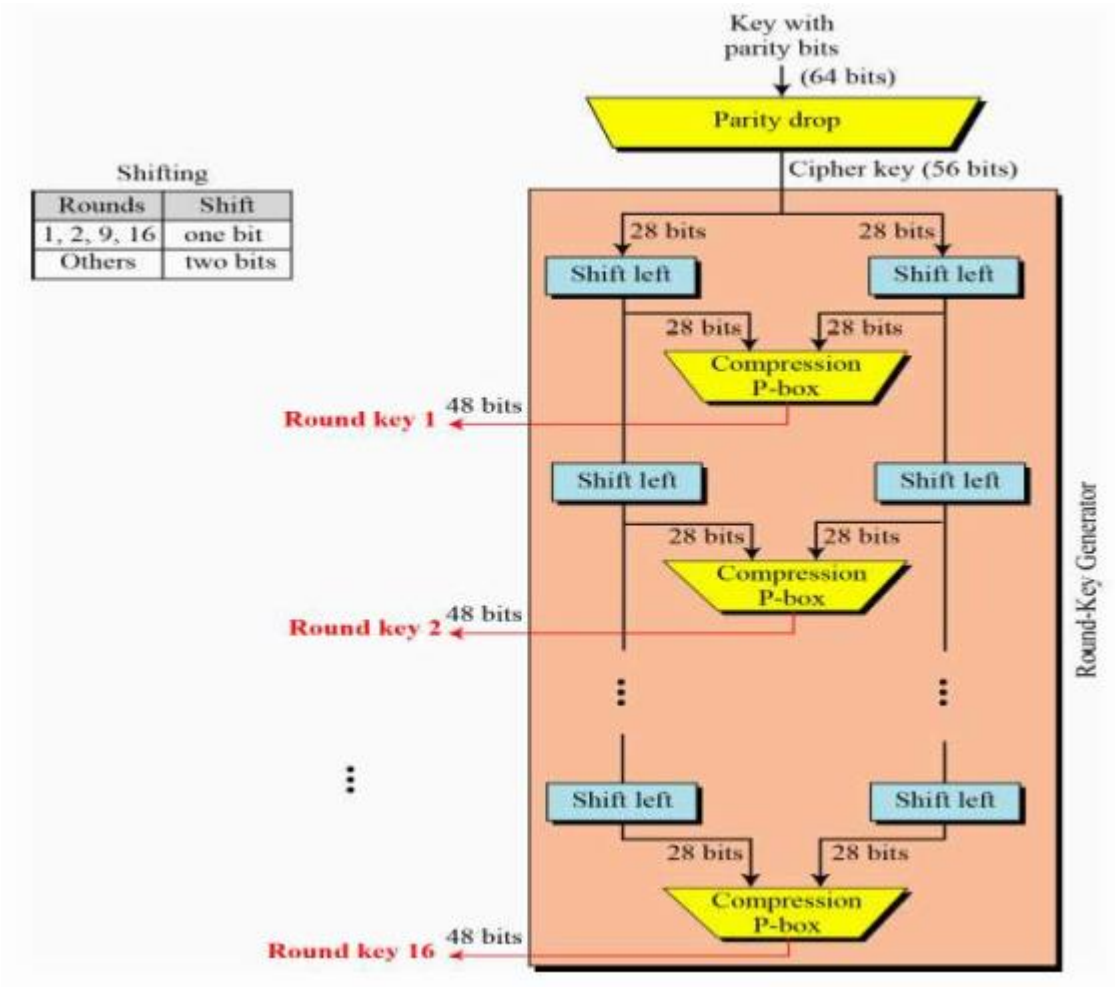



**Figure 4.13.** *Algorithme d'obtention des 16 clés DES.*

## 2.3. Cryptanalyse de DES

### a. Cryptanalyse différentielle

En 1990, Eli Biham et Adi Shamir, introduisent la méthode de ***cryptanalyse différentielle***. C'est grâce à cette méthode qu'ils ont pu trouver une attaque à texte clair efficace contre le DES. Cette attaque cherche des paires de texte en clair et des paires de texte chiffré, puis elle les analyse en comparant les différences notables entre ces deux paires.

Ainsi, un DES à 8 ou à 10 tours peut facilement être cassé, mais le DES complet à 16 tours est resté hors de portée de cette attaque.

### b. Cryptanalyse linéaire

La ***cryptanalyse linéaire*** a été introduite par H.Gilbert et M. Matsui dans le cas du DES. C'est une attaque à messages clairs connus, qui utilise de légers défauts statistiques des étages de substitutions, correspondant aux boites-S dans le cas de DES. Elle n'est utilisable que pour un DES restreint à quelques tours, mais le DES réel n'est pas menacé par cette attaque.

### c. Recherche exhaustive

Une autre attaque qui est celle par ***recherche exhaustive*** de la clé, est rendue envisageable vu l'accroissement de la puissance des ordinateurs, de leur nombre et des facilités de communication entre eux. Ainsi, les laboratoires RSA ont lancé en Janvier 1997 un défi consistant à décrypter par recherche exhaustive un message chiffré par DES pour démontrer que la taille des clés DES, qui est de 56 bits, a devenu insuffisante. Une équipe coordonnant des milliers d'ordinateurs du monde entier a abouti à la découverte de la bonne clé le 17 Juin 1997, après avoir exploré environ un quart de l'espace des clés. Bien qu'une telle recherche demande des moyens considérables, elle a montré que le DES n'offre plus aujourd'hui une grande sécurité.

## 3. Triple DES (3DES)

Pour pallier à l'insuffisance cryptographique observée du cryptosystème DES, due à la faible longueur de sa clé, il a été indispensable de chercher une solution rapide à cette situation. La première idée qui vienne à l'esprit est de combiner plusieurs chiffrements DES pour obtenir un système ayant une clé plus longue. Tout d'abord, une tentative consistant à combiner deux chiffrement DES a été essayée, mais il a vite paru qu'avec une attaque à message clair, dite « par le milieu », ce système sera remis en cause. Cette attaque s'appuie sur le message intermédiaire inconnu apparaissant entre les deux chiffrements DES successifs. Ainsi, elle construit la liste des messages intermédiaires possibles en chiffrant par DES un texte clair avec les $2^{56}$ clés possibles. En déchiffrant par DES le chiffré correspondant avec des clés différentes, on obtient une autre liste de messages intermédiaires possibles et le véritable message intermédiaire est dans l'intersection des deux listes. Le coût en mémoire de cette attaque est très important mais son coût en temps n'est pas significativement plus élevé que l'attaque exhaustive sur DES.

Ensuite, et en 1978, le **triple DES (3DES)** a été conçu par *Whitfield Diffie*, *Martin Hellman* et *Walt Tuchmann*. Il consiste à composer deux chiffrements DES de même clé séparée par un déchiffrement DES avec une autre clé. Donc, ce principe peut être formulé comme suit :

$$\textbf{Triple-DES}_{k1,k2} = \textbf{DES}_{k1} \textbf{ O } \textbf{DES}^{-1}_{k2} \textbf{ O } \textbf{DES}_{k1}$$

Le déchiffrement de son tour est formulé par :

$$\textbf{Triple-DES}^{-1}_{k1,k2} = \textbf{DES}^{-1}_{k1} \textbf{ O } \textbf{DES}_{k2} \textbf{ O } \textbf{DES}^{-1}_{k1}$$

Cette méthode de chiffrement reste hors portée de l'attaque exhaustive vu la taille de la clé 3DES qui est composée de deux clés DES et donc composée de 112 bits. Une autre variante à trois clés DES différentes peut être conçue. D'une façon plus formelle, son principe peut être donné par la formule suivante :

$$\textbf{Triple-DES}_{k1,k2,k3} = \textbf{DES}_{k1} \textbf{ O } \textbf{DES}_{k2} \textbf{ O } \textbf{DES}_{k3}$$

Malgré cela, cette variante reste aussi fragile à une attaque de coût en 2 s'appuyant sur l'un des deux messages intermédiaires.

## 4. AES

DES a très longtemps profité du soutien politique des USA. Par exemple, *Robert S. Litt* (Principal Associate Deputy Attorney General), a assuré le 17 mars 1998, que le FBI n'avait aucune possibilité technologique et financière de décoder un message codé avec un algorithme symétrique dont la clef

secrète a une longueur égale à 56 bits. Et pour compléter sa démonstration, il a déclaré aussi que 14000 PC Pentium durant 4 mois seraient nécessaires pour réaliser cela. C'est d'ailleurs, ce que *Louis J. Freeh* (Directeur du FBI) et *William P. Crowell* (Deputy Director de la NSA) ont déclaré de leur tour. Malgré cela, pas mal d'attaques contre le DES ont été développées. Ce qui fait que, conscient des risques concernant DES, le NIST (National Institute of Standards and Technology) a demandé à la communauté cryptographique de réfléchir au successeur : **AES**.

**AES** est le sigle d'**Advanced Encryption Standard** (en français, standard de chiffrement avancé). C'est l'algorithme **Rijndael**, du nom de leurs concepteurs Belges Joan Daemen et Vincent Rijmen. Il a été retenu par le NIST en octobre 2000 pour être l'algorithme AES, le nouveau standard de chiffrement pour les organisations du gouvernement des Etats-Unis, et ce, principalement pour des raisons de sécurité, performance, efficacité, facilité d'implémentation et flexibilité. De plus, son utilisation est très pratique car il consomme peu de mémoire.

Plus précisément, ce cryptosystème est issu d'un appel d'offre international lancé en janvier 1997 et ayant reçu 15 propositions en première ronde : LOKI97 (Australia, Australian Defence Force Academy, *L.Brown, J.Pieprzyk, J.Seberry*), RIJNDAEL (Belgique, *J. Daemen, V. Rijmen*), CAST-256 (Canada, Entrust Technologies), DEAL (Canada, Outerbridge, Knudsen), FROG (Costa Rica, TecApro International S.A), DFC (France, Centre National pour la Recherche Scientifique), MAGENTA (Allemagne, Deutsche Telekom AG), E2 (Japon, Nippon Telegraph and Telephone Corporation), CRYPTON (Corée, Future Systems), HPC (Etats-Unis, Université Arizona, *Rich Schroeppel*), MARS (Etats-Unis, IBM), RC6 (Etats-Unis, RSA Laboratories), SAFER+ (Etats-Unis, Cylink Corporation), TWOFISH (Etats-Unis, *B. Schneier, J.Kelsey, D.Whiting, D.Wagner, C.Hall, N.Ferguson*), SERPENT (Grande-Bretagne, Israël, Norvège, *R. Anderson, Eli Biham, L. Knudsen*).

Ces algorithmes ont été évalués par des experts, avec forum de discussion sur Internet, et organisation de conférences. Pour la même raison, 600 CD-ROM portant ces algorithmes ont été distribués dans plus de 50 pays en décembre 1998. Après la deuxième ronde qui a débuté le 22 Mars 1999 lors d'une conférence à Rome, Rijndael a été déclaré vainqueur par le NIST le 2 octobre 2000 sur les 5 candidats finalistes : MARS, RC6, Rijndael, Serpent et TwoFish. Toutefois, il est important de signaler que sur chacun des tests effectués, Rijndael n'est jamais sorti vainqueur. Il s'est distingué à chaque fois par des performances intéressantes et donc au final pour sa polyvalence. Les résultats des votes étaient comme suit :

- ✓ Rijndael : 86 votes,
- ✓ Serpent : 59 votes,
- ✓ Twofish : 31 votes,
- ✓ RC6 : 23 votes,
- ✓ MARS : 13 votes.

Malgré cela, la NSA a annoncé que tous ces finalistes pouvaient être considérés comme sûrs et qu'ils étaient suffisamment robustes pour chiffrer les données non-classifiées du gouvernement américain.

Donc, le terme d'AES remplace désormais celui de Rijndael sans que l'algorithme ne soit modifié. Mais en réalité, AES est un sous-ensemble de Rijndael, puisque ce dernier offre des tailles de blocs et de clefs qui sont des multiples de 32 compris entre 128 et 256 bits, tandis que AES travaille avec des blocs de 128 bits seulement. Cette longueur de blocs, d'un côté, et la longueur des clés utilisées avec ce même cryptosystème (AES), qui peut être de 128,192 ou de 256 bits, d'un autre côté, ont été jugées suffisantes en juin 2003 par le gouvernement américain, et ce, pour protéger des documents classifiés jusqu'au niveau « Secret », alors que le niveau « Top-secret » nécessite des clés de 192 ou 256 bits.

### 4.1. Description

Le message et la clé sont conservés sous forme de tables. Le nombre de colonnes dépend des tailles des textes et clés.

$$Nb = L_{bloc} / 32$$

$$Nk = L_{clef} / 32$$

Une colonne du tableau correspond à un mot de 32 bits. Ainsi, chaque petit bloc représente 8 bits, donc 1 octet. L'input et l'output sont donc gérés comme des séquences linéaires d'octets. Pour être tout à fait exact, l'algorithme AES n'est pas exactement celui de Rijndael dans la mesure où ce dernier supporte des tailles de blocs plus nombreux qu'AES. AES fixe la taille de blocs à 128 bits - représenté par $N_b = 4$. $N_b$ reflète le nombre de mots de 32 bits dans un bloc (c'est aussi le nombre de colonnes nécessaire pour une représentation matricielle). Chaque bloc contient $16 \times 8 = 128$ bits et est constitué de 16 octets, rangés dans un tableau $4 \times 4$ :

| | | | |
|---|---|---|---|
| $a_{0,0}$ | $a_{0,1}$ | $a_{0,2}$ | $a_{0,3}$ |
| $a_{1,0}$ | $a_{1,1}$ | $a_{1,2}$ | $a_{1,3}$ |
| $a_{2,0}$ | $a_{2,1}$ | $a_{2,2}$ | $a_{2,3}$ |
| $a_{3,0}$ | $a_{3,1}$ | $a_{3,2}$ | $a_{3,3}$ |

AES utilise des clés de 128, 192 ou 256 bits. La longueur de clé est caractérisée de façon similaire par $N_k = 4$, 6 ou 8.

Comme DES, AES exécute une séquence de rondes qui seront détaillés dans la suite. On note $N_r$ le nombre de rondes qui doivent être effectuées. Ce nombre dépend des valeurs de $N_b$ et de $N_k$. Les différentes configurations possibles sont détaillées dans le tableau suivant.

| | $N_k$ | $N_b$ | $N_r$ |
|---|---|---|---|
| AES-128 | 4 | 4 | 10 |
| AES-192 | 6 | 4 | 12 |
| AES-256 | 8 | 4 | 14 |

À chaque ronde, quatre transformations sont appliquées :

1. substitution d'octets dans le tableau d'état (SubBytes).
2. décalage de rangées dans le tableau d'état (ShiftRows).
3. déplacement de colonnes dans le tableau d'état sauf à la dernière ronde (MixColumns).
4. addition d'une "clef de ronde" qui varie à chaque ronde (AddRoundKey).

- **SubBytes**

L'étape SubBytes correspond à la seule transformation non-linéaire de l'algorithme. Dans cette étape, chaque élément de la matrice State est permuté selon une table de substitution inversible notée SBox.

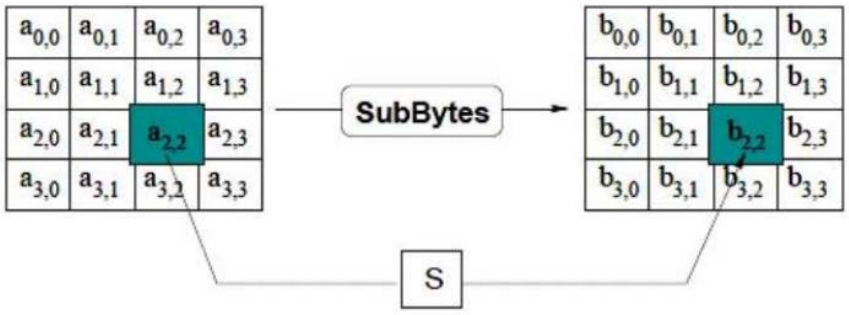



**Figure 4.14.** *SubBytes AES.*

Une seule S-Box est suffisante pour toute la phase de chiffrement.

| x \ y | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 52 | 09 | 6a | d5 | 30 | 36 | a5 | 38 | bf | 40 | a3 | 9e | 81 | f3 | d7 | fb |
| 1 | 7c | e3 | 39 | 82 | 9b | 2f | ff | 87 | 34 | 8e | 43 | 44 | c4 | de | e9 | cb |
| 2 | 54 | 7b | 94 | 32 | a6 | c2 | 23 | 3d | ee | 4c | 95 | 0b | 42 | fa | c3 | 4e |
| 3 | 08 | 2e | a1 | 66 | 28 | d9 | 24 | b2 | 76 | 5b | a2 | 49 | 6d | 8b | d1 | 25 |
| 4 | 72 | f8 | f6 | 64 | 86 | 68 | 98 | 16 | d4 | a4 | 5c | cc | 5d | 65 | b6 | 92 |
| 5 | 6c | 70 | 48 | 50 | fd | ed | b9 | da | 5e | 15 | 46 | 57 | a7 | 8d | 9d | 84 |
| 6 | 90 | d8 | ab | 00 | 8c | bc | d3 | 0a | f7 | e4 | 58 | 05 | b8 | b3 | 45 | 06 |
| 7 | d0 | 2c | 1e | 8f | ca | 3f | 0f | 02 | c1 | af | bd | 03 | 01 | 13 | 8a | 6b |
| 8 | 3a | 91 | 11 | 41 | 4f | 67 | dc | ea | 97 | f2 | cf | ce | f0 | b4 | e6 | 73 |
| 9 | 96 | ac | 74 | 22 | e7 | ad | 35 | 85 | e2 | f9 | 37 | e8 | 1c | 75 | df | 6e |
| a | 47 | f1 | 1a | 71 | 1d | 29 | c5 | 89 | 6f | b7 | 62 | 0e | aa | 18 | be | 1b |
| b | fc | 56 | 3e | 4b | c6 | d2 | 79 | 20 | 9a | db | c0 | fe | 78 | cd | 5a | f4 |
| c | 1f | dd | a8 | 33 | 88 | 07 | c7 | 31 | b1 | 12 | 10 | 59 | 27 | 80 | ec | 5f |
| d | 60 | 51 | 7f | a9 | 19 | b5 | 4a | 0d | 2d | e5 | 7a | 9f | 93 | c9 | 9c | ef |
| e | a0 | e0 | 3b | 4d | ae | 2a | f5 | b0 | c8 | eb | bb | 3c | 83 | 53 | 99 | 61 |
| f | 17 | 2b | 04 | 7e | ba | 77 | d6 | 26 | e1 | 69 | 14 | 63 | 55 | 21 | 0c | 7d |

**Figure 4.15.** *S-Box AES.*

- **ShiftRow**

Cette étape augmente la diffusion dans la ronde. L'opération shiftrow effectue une simple permutation circulaire à gauche par ligne de chacun des octets du bloc. La première ligne sera inchangée. Dans la seconde, chaque octet sera décalé d'un cran à gauche (le premier prenant la place du dernier). Dans la troisième, chaque octet sera décalé de 2 crans à gauche, et dans la quatrième, chaque octet sera décalé de 3 crans à gauche.

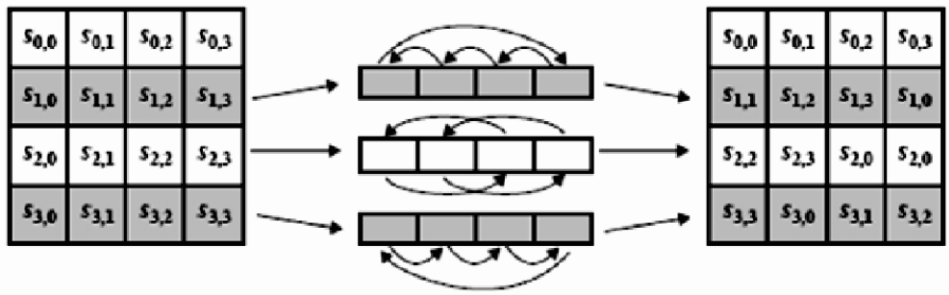



**Figure 4.16.** *ShiftRow AES.*

- **MixColumns**

Cette procédure effectue un "mélange" à l'intérieur de chaque colonne. Une différence sur 1 byte d'entrée se propage sur les 4 bytes de sortie. On a donc encore une étape de diffusion. La matrice utilisée est définie par Rijndael. Elle contiendra toujours ces valeurs.

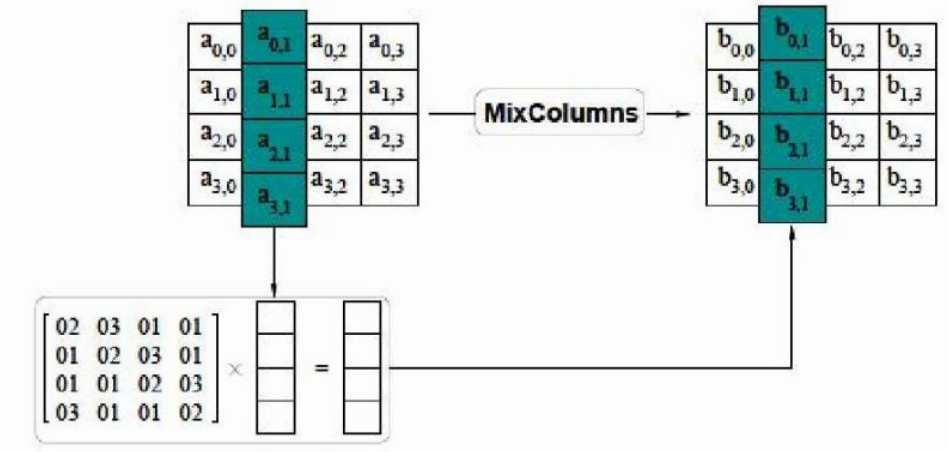


**Figure 4.17.** *MixColumns AES.*

L'opération inverse de MixColumns est notée InvMixColumns qui utilise la matrice suivante :

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}$$

- **AddRoundKey**

Lors de l'étape AddRoundKey, la matrice State est modifiée en l'additionnant avec une clé de ronde. Cette étape est illustrée dans la figure suivante :

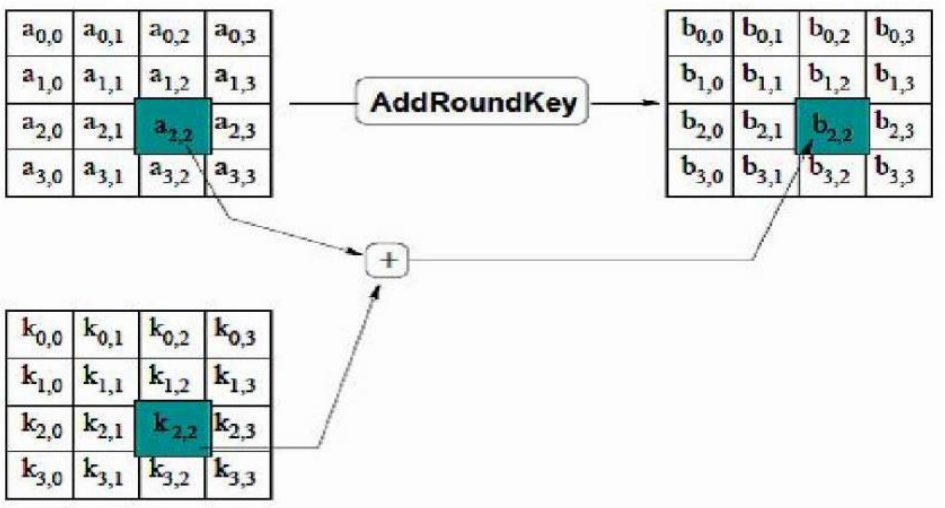



**Figure 4.18.** *AddRoundKey AES.*

Le chiffrement par AES suit le schéma donné par la figure 4.19 équivalent à celui de la figure 4.20. L'opération de déchiffrement, quant à elle, se réalise en inversant le schéma de chiffrement.

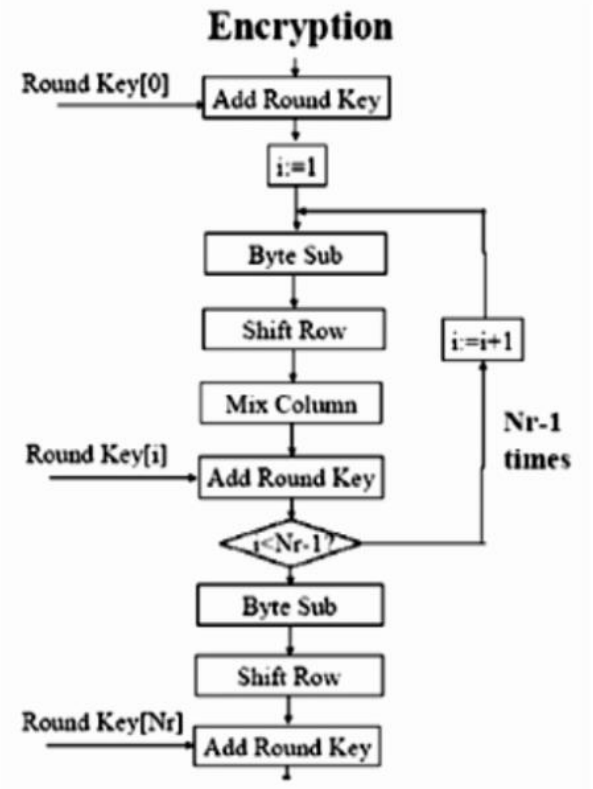



**Figure 4.19.** *Algorithme de chiffrement AES.*

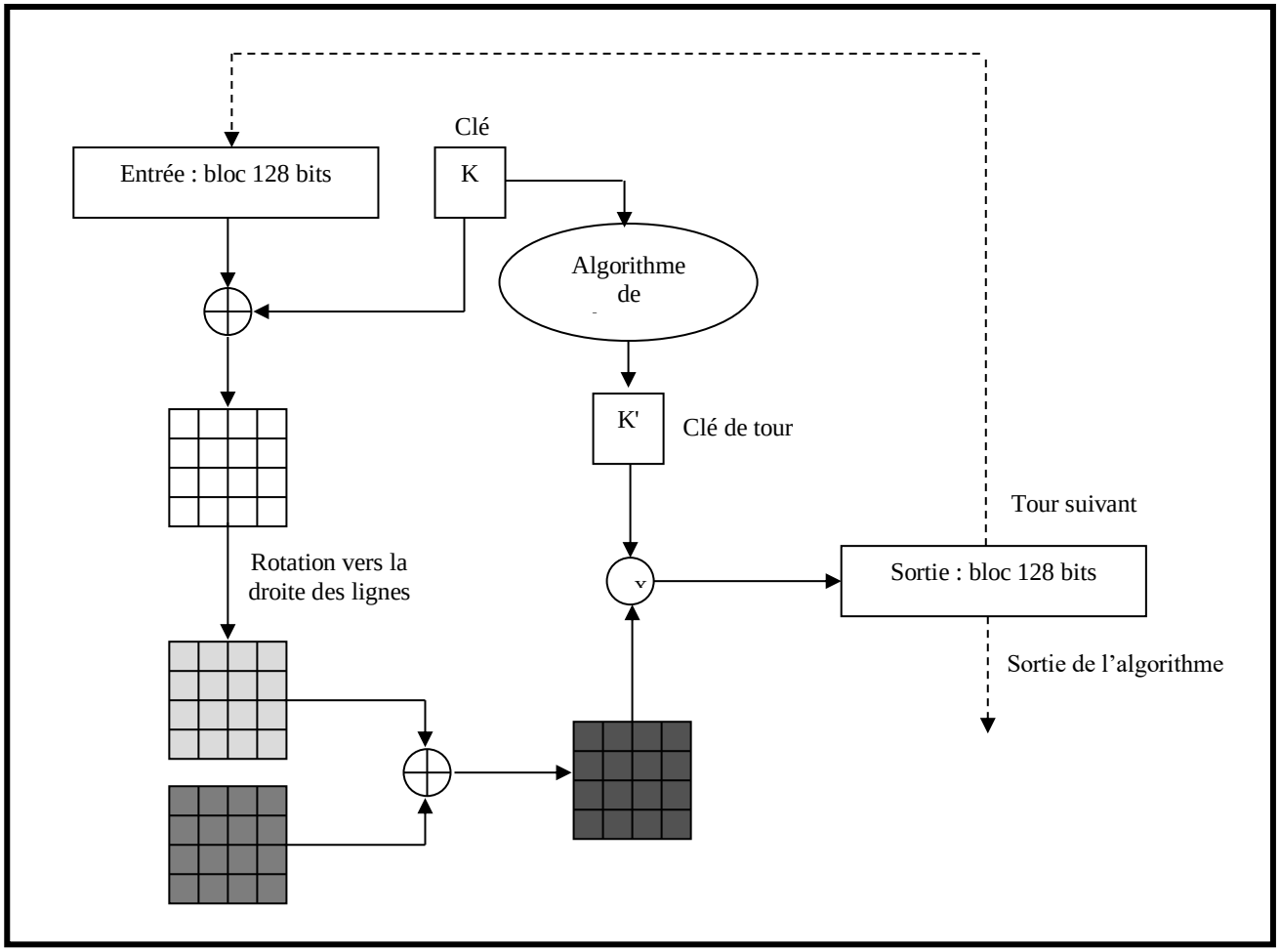



**Figure 4.20.** *Schéma de chiffrement AES.*

Remarque : Un algorithme de cadencement de clé calcule à partir de la clé K une suite de $N_r+1$ sous-clés de tour ($K_0$ ; … ;$K_{Nr}$ ).

### 4.2. Cryptanalyse de l'AES

L'AES n'a pour l'instant pas été cassé et la recherche exhaustive demeure la seule solution.

#### a. Attaques sur des versions simplifiées

Niels Ferguson et son équipe ont proposé en 2000 une attaque sur une version à 7 tours de l'AES 128 bits. Une attaque similaire casse un AES de 192 ou 256 bits contenant 8 tours. Un AES de 256 bits peut être cassé s'il est réduit à 9 tours avec une contrainte supplémentaire. En effet, cette dernière attaque repose sur le principe des « related-keys » (clés apparentées). Dans une telle attaque, la clé

demeure secrète mais l'attaquant peut spécifier des transformations sur la clé et chiffrer des textes à sa guise. Il peut donc légèrement modifier la clé et regarder comment la sortie de l'AES se comporte.

**b. Attaques sur la version complète**

Certains groupes ont affirmé avoir cassé l'AES complet, mais après vérification par la communauté scientifique, il s'avérait que toutes ces méthodes étaient erronées. Cependant, plusieurs chercheurs ont mis en évidence des possibilités d'attaques algébriques, notamment l'attaque XL et une version améliorée, la XSL. Ces attaques ont été le sujet de nombreuses discussions et leur efficacité n'a pas encore été pleinement démontrée, le XSL fait appel à une analyse heuristique dont la réussite n'est pas systématique. De plus, elles sont impraticables car le XSL demande au moins $2^{87}$ opérations voire $2^{100}$ dans certains cas. Le principe est d'établir les équations (quadratiques/booléennes) qui lient les entrées aux sorties et de résoudre ce système qui ne comporte pas moins de 8000 inconnues et 1600 équations pour 128 bits. La solution de ce système reste pour l'instant impossible à déterminer. En l'absence d'une preuve formelle sur l'efficacité d'attaques similaires au XSL, l'AES est donc considéré comme sûr.

**5. Blowfish**

Blowfish a été conçu par *Bruce Schneier* en 1993 comme étant une alternative aux algorithmes existants, en étant rapide et gratuit. Ce cryptosystème est sensiblement plus rapide que le DES. La grandeur de ses blocs est de 64 bits et il peut prendre une longueur de clé variant entre 32 bits et 448 bits. Ainsi, et depuis sa conception, il a été grandement analysé et est aujourd'hui considéré comme étant un algorithme de chiffrement robuste, mais il n'est pas breveté. Ainsi, son utilisation est libre et gratuite.

**6. Serpent**

Serpent, inventé par *Ross Anderson*, Eli Biham et Lars Knudsen, est un cryptosystème symétrique chiffrant des blocs de 128 bits. Il a été développé en vue d'être un Advanced Encryption Standard. Et bien que le choix du NIST pour AES s'est porté sur Rijndael, mais ça n'empêche de signaler que Serpent et Rijndael sont similaires, et que la principale différence entre eux, est que Rijndael est plus rapide mais Serpent est plus sûr. De plus, aucune attaque connue n'a réussie à casser cet algorithme.

**7. Twofish**

Twofish est un algorithme de chiffrement symétrique par bloc inventé et analysé par *Bruce Schneier, Niels Ferguson, John Kelsey, Doug Whiting, David Wagner* et *Chris Hall* au sein du Counterpane Labs, pour participer au concours AES, où, il a été l'un des cinq finalistes du concours sans pour autant être sélectionné pour le standard. Ce cryptosystème est conçu pour être très sûr et très flexible, en chiffrant des blocs de 128 bits avec une clé de 128, 192 ou 256 bits, et en reprenant quelques concepts présents dans le Blowfish du même auteur. Cependant, Twofish est légèrement plus lent que Rijndael mais plus rapide que les autres finalistes de l'AES.

En 2005, Counterpane Labs a passé un long temps en évaluant Twofish, sans pouvoir trouver d'attaques possibles sur la version complète de Twofish, qui semble être plus sûre que la version initialement annoncée durant le concours AES. Ainsi, la recherche exhaustive reste le seul moyen pour le casser. Malgré ça, il reste relativement peu utilisé.

**8. MARS**

MARS est un algorithme de chiffrement symétrique par blocs créé par IBM comme algorithme pour le standard AES. *Don Coppersmith* était l'un des concepteurs de cet algorithme, qui prend en charge des blocs de 128 bits et des clés de dimensions variables entre 128 et 448 bits par incréments de 32 bits. Cet algorithme est unique, car il associe toutes les techniques de cryptage connues dans un seul produit. Ainsi, il utilise deux algorithmes séparés, de façon que si une partie de MARS est cassée, le reste des chiffres restera sécurisé et les données seront sauvegardées. De plus, MARS offre une meilleure sécurité que le triple DES et il est plus rapide que le DES.

## 9. RC6

RC6 est un algorithme de chiffrement par bloc publié en 1998, et conçu au sein de la société RSA Security par *Ron Rivest, Matt Robshaw, Ray Sidney* et *Yiqun Lisa Yin* dans le cadre du concours AES, où il parvient à atteindre la finale aux cotés de quatre autres systèmes de chiffrement. Il est basé sur un bloc de 128 bits et supporte des clés de 128, 192 et 256 bits.

## 10. Conclusion

A l'heure actuelle l'utilisation du DES est simplement déconseillée à cause de la grande puissance de calcul assurée par les ordinateurs les plus récents. Toutefois le triple DES, permet d'apporter un niveau de sécurité acceptable et de résister aux attaques les plus classiques. Le choix de l'AES reste néanmoins le meilleur choix, dans l'attente d'un remplaçant ou d'une méthode d'attaque efficace qui va le remettre en cause.

# *Chapitre V*
# *Cryptosystèmes asymétriques*

Plusieurs systèmes à clé publique ont été proposés. Leur sécurité repose sur divers problèmes calculatoire. Les plus connus sont les suivant :

## 1. RSA

En cryptographie à clé publique, les trois lettres **RSA** sont certainement les plus célèbres. Ce cryptosystème tire son nom des noms de ses trois inventeurs : *R. Rivest, A. Shamir, et L. Adleman*. Ce système, inventé en 1977, est le premier protocole de cryptographie à clé publique. Il a été breveté par le MIT en 1983 aux États-Unis d'Amérique.

### 1.1. Description

Ce chiffrement est fondé sur la difficulté de factoriser un nombre qui est le produit de deux grands nombres premiers. Il utilise l'arithmétique de $Z_n$, qui est un anneau pour tout entier $n$ supérieur à 2, et où $n$ est le produit de deux nombres premiers impairs distincts $p$ et $q$. Pour un tel $n$, on a :

$$\varphi(n) = (p-1)(q-1)$$

La description formelle du système est donnée dans la figure 5.1.

Soit $n = pq$ où : $p$ et $q$ sont premiers, et soit $P = C = Z_n$.

On définit :

$K = \{(n, p, q, a, b) : n = pq, p \text{ et } q \text{ premiers} ; ab \equiv 1 \bmod \emptyset(n)\}$

Pour k = $(n, p, q, a, b)$, on définit :

$$\mathbf{e_k(x) = x^b \bmod n}$$

$$\mathbf{d_k(y) = y^a \bmod n}$$

$(x, y) \in Z_n$. Les valeurs $n$ et $b$ sont publiques, et les valeurs $p$, $q$ et $a$ sont secrètes.

**Figure 5.1.** *Le chiffrement RSA.*

La sécurité de RSA est basée sur l'hypothèse que la fonction $e_k(x) = x^b \bmod n$ est a sens unique, ce qui rend impossible à Oscar de décrypter un texte chiffré. Mais, Bob et à l'aide de la trappe qu'il garde secrète, qui est la factorisation $n = pq$, est le seul qui est en mesure d'accomplir l'opération de déchiffrement. Donc, il peut calculer $\varphi(n) = (p-1)(q-1)$ et calculer l'exposant de déchiffrement $a$ en utilisant l'algorithme d'Euclide étendu.

Il y a beaucoup de problèmes autour du chiffrement RSA à résoudre c'est pourquoi, il faut expliquer comment procéder efficacement au chiffrement et au déchiffrement. Ainsi, et pour mettre en œuvre le chiffrement RSA, Bob suit les étapes indiquées ci-dessous :

**1)** Bob engendre deux grands nombres premiers $p$ et $q$.
**2)** Bob calcul $n = pq$ et $\varphi(n) = (p-1)(q-1)$.
**3)** Bob choisis un $b$ aléatoire ($1< b< \varphi(n)$), tel que : pgcd($b$, $\varphi(n)$) = 1.
**4)** Bob calcul $a = b^{-1}$ en utilisant l'algorithme d'Euclide. ($ab \equiv 1 \bmod \varphi(n)$)
**5)** Bob publie $b$ et $n$ dans un répertoire.

Une attaque évidente à ce système consiste à tenter de factoriser $n$, alors que, l'intérêt du système RSA repose sur le fait, qu'à l'heure actuelle, il est pratiquement impossible de retrouver dans un temps raisonnable $p$ et $q$ à partir de $n$ si celui-ci est très grand. Donc, Bob est le seul qui peut calculer $a$ dans un temps court, sans que cela nécessite la transmission des entiers $p$ et $q$, ce qui empêche leur piratage. Mais, si une méthode de factorisation rapide sera développée, ce système serait aussitôt périmé. Ainsi, la sécurité de RSA semble satisfaisante bien qu'il ne soit pas prouvé mathématiquement qu'on ne puisse pas le casser. En effet, en augmentant constamment la taille des clés, ce système reste très fiable si ses utilisateurs suivent les conseils des spécialistes, qui peuvent porter sur la taille des clés, la forme des nombres employés ou sur les méthodes d'implémentation. De même, le bon choix de $p$ et $q$ est aussi, un point crucial assurant la bonne sécurisation de ce cryptosystème.

- **Choix de p et q :**

Le choix de $p$ et $q$ affecte grandement le niveau de sécurité de RSA. Pour cela, il faut évidemment se prémunir contre les algorithmes de factorisation dont la complexité dépend essentiellement de la taille du plus petit facteur premier de $n$, donc si possible choisir $p$ et $q$ de même taille. Il ne faut cependant pas les choisir trop proches, car alors une attaque exhaustive est possible : on suppose $q=p+k$ avec $k$ un petit entier, et comme on connaît $n = pq$ , on est ramené à résoudre une équation du second degré pour chaque valeur de $k$.
De plus, on impose usuellement que $p$ et $q$ soient des nombres premiers forts. Sachant, qu'un nombre premier $p$ est dit fort lorsque :

a) p - 1 a un grand facteur premier r,
b) p + 1 a un grand facteur premier,
c) r - 1 a un grand facteur premier.

La condition (a) permet de se prémunir contre la factorisation de p par l'algorithme P-1 de Pollard, la condition (b) permet de se prémunir contre la factorisation de p par l'algorithme P+1, attribué à Williams. Enfin, la condition (c) permet de se prémunir contre les attaques cycliques.

### 1.2. Cryptanalyse de RSA

Depuis son apparition, plusieurs attaques ont été découvertes contre le RSA. Et même si aucune de ces attaques n'est réellement destructive, elles démontrent toutefois qu'il faut implémenter RSA avec beaucoup de précautions. Elles illustrent également la difficulté à définir des notions de sécurité convenables, en chiffrement asymétrique.

*Phong Nguyen* avait groupé les attaques possibles en quatre grandes catégories :

- ✓ Attaques élémentaires,
- ✓ Attaques sur les implémentations de RSA,
- ✓ Attaques simples de RSA à petit exposant,
- ✓ Attaque par géométrie des nombres.

Dans ce qui suit, nous citons quelques exemples d'attaques tout en mentionnant la catégorie correspondante.

#### a. Recherche exhaustive : (1ère catégorie)

Comme la fonction de chiffrement RSA : $m \rightarrow m^b \bmod n$ est déterministe, c-à-d un message est toujours chiffré en le même chiffré, donc et si l'ensemble des messages possibles est connu et de

petite taille, il sera facile de décrypter par une recherche exhaustive. Pour éviter cette attaque, il est indispensable de randomiser les messages avant chiffrement. De ce fait, on aura une fonction de chiffrement probabiliste mieux que d'être déterministe. Cela revient à faire des transformations sur la fonction à sens unique à trappe avant d'être utilisée en chiffrement.

### b. Attaque par chronométrage : (2ème catégorie)

Considérons une carte à puce contenant une clef privée RSA. Normalement, un attaquant mettant la main sur la carte ne peut arriver à déterminer cette clef privée, en raison de protections physiques. *Kocher* a néanmoins démontré qu'en mesurant précisément le temps pris par la carte pour effectuer un déchiffrement RSA, un attaquant pouvait rapidement déterminer l'exposant privé *a* pour les implémentations usuelles du RSA.

## 2. Chiffrement d'ElGamal

L'algorithme **ElGamal**, créé par *Taher Elgamal*, est un algorithme de cryptographie asymétrique basé sur les logarithmes discrets. Cet algorithme est utilisé par de récentes versions de PGP, d'autres systèmes de chiffrement et même par le DSA (Digital Signature Algorithm), qui est un algorithme de signature numérique standardisé par le NIST aux États Unis. Ainsi, il peut être utilisé pour le chiffrement et la signature électronique, rappelons que la signature est l'ensemble des mécanismes permettant d'assurer au destinataire que le message envoyé a bien été rédigé par l'émetteur légal. De plus, et contrairement à RSA, cet algorithme n'a jamais été sous la protection d'un brevet.

### 2.1. Problème du logarithme discret

Le problème du logarithme discret est décrit dans le corps fini $Z_p$, où $p$ est un nombre premier, sachons que le groupe $Z_P^*$ est cyclique, et que ses générateurs sont appelés racines primitives modulo $p$. La figure suivante résume le problème du logarithme discret.

**Instance du problème :** I = ($p$, α, ß) où $p$ est premier, $\alpha \in Z_p$ est primitif et $ß \in Z_p^*$.

**Question :** Trouver l'unique $a$, $0 \leq a \leq p-2$ tel que :

$$\alpha^a \equiv ß \pmod{p}$$

On note cet entier $\log_\alpha ß$

**Figure 5.2.** *Problème du logarithme discret dans $Z_p$.*

Le problème du logarithme discret dans $Z_p$, faisant l'objet de nombreuses études, est réputé difficile. Sachons qu'aucun algorithme polynomial n'a été défini pour le résoudre. Cependant, et pour éviter les attaques connues, $p$ doit être convenablement choisi, et $p$–1 doit avoir un grand facteur premier.

L'utilité de ce problème en cryptographie provient du fait que calculer des logarithmes discrets est certainement difficile, tandis que calculer l'opération inverse d'exponentiation peut se faire efficacement avec l'algorithme d'exponentiation modulaire. En d'autres termes, l'exponentiation modulo $p$ est une fonction à sens unique pour des nombres premiers $p$ convenables.

### 2.2. Description

La figure suivante présente, d'une manière formelle, l'algorithme ElGamal.

Soit $p$ un nombre premier tel que le problème du logarithme discret dans $Z_p$ soit difficile, et soit $\alpha \in Z_p^*$ un élément primitif. Soit $P = Z_p^*$. $C = Z_p^* \times Z_p^*$ et

$$K = \{(p, \alpha, a, ß) : ß \equiv \alpha^a \pmod{p}\}$$

Les valeurs $p$, $\alpha$ et ß sont publiques, et $a$ est secret.

Pour $k = (p, \alpha, a, ß)$, et pour un $\acute{k} \in Z_{p-1}$ aléatoire (secret), la fonction de chiffrement est définit par :

$$\mathbf{e_k(x, \acute{k}) = (y_1, y_2)}$$

Où

$$\mathbf{y_1 = \alpha^{\acute{k}} \bmod p}$$

Et

$$\mathbf{y_2 = x\, ß^{\acute{k}} \bmod p}$$

Pour $y_1, y_2 \in Z_p^*$, la fonction de déchiffrement est définit comme suit :

$$\mathbf{d_k(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p}$$

**Figure 5.3.** *Chiffrement d'ElGamal.*

Informellement, le fonctionnement du chiffrement ElGamal peut être décrit par la suite des points suivants :

- ✓ Le texte clair est masqué par la multiplication par $ß^k$, en produisant $y_2$ ;
- ✓ la valeur $\alpha^k$ est également transmise en tant que partie du texte chiffré ;
- ✓ Bob, qui connaît l'exposant secret $a$, peut calculer $ß^k$ à partir de $\alpha^k$. Il peut alors « enlever le masque » en divisant $y_2$ par $ß^k$ et obtenir le texte clair x.

D'après la description, soit formelle ou informelle, de cet algorithme, il est clair que le chiffrement d'ElGamal est non déterministe, ou encore dit probabiliste, du moment où, l'opération de chiffrement dépend du texte clair, x, et d'une valeur aléatoire, k, choisie par Alice. Donc, plusieurs textes chiffrés peuvent correspondre à un même texte clair.

### 2.3. Cryptanalyse d'ElGamal

Une attaque possible à ce cryptosystème est celle dite ***man in the middle***. Son principe dans le cas d'ElGamal fonctionnant avec le mode *Diffie-hellman* est illustré sur la figure 5.4.

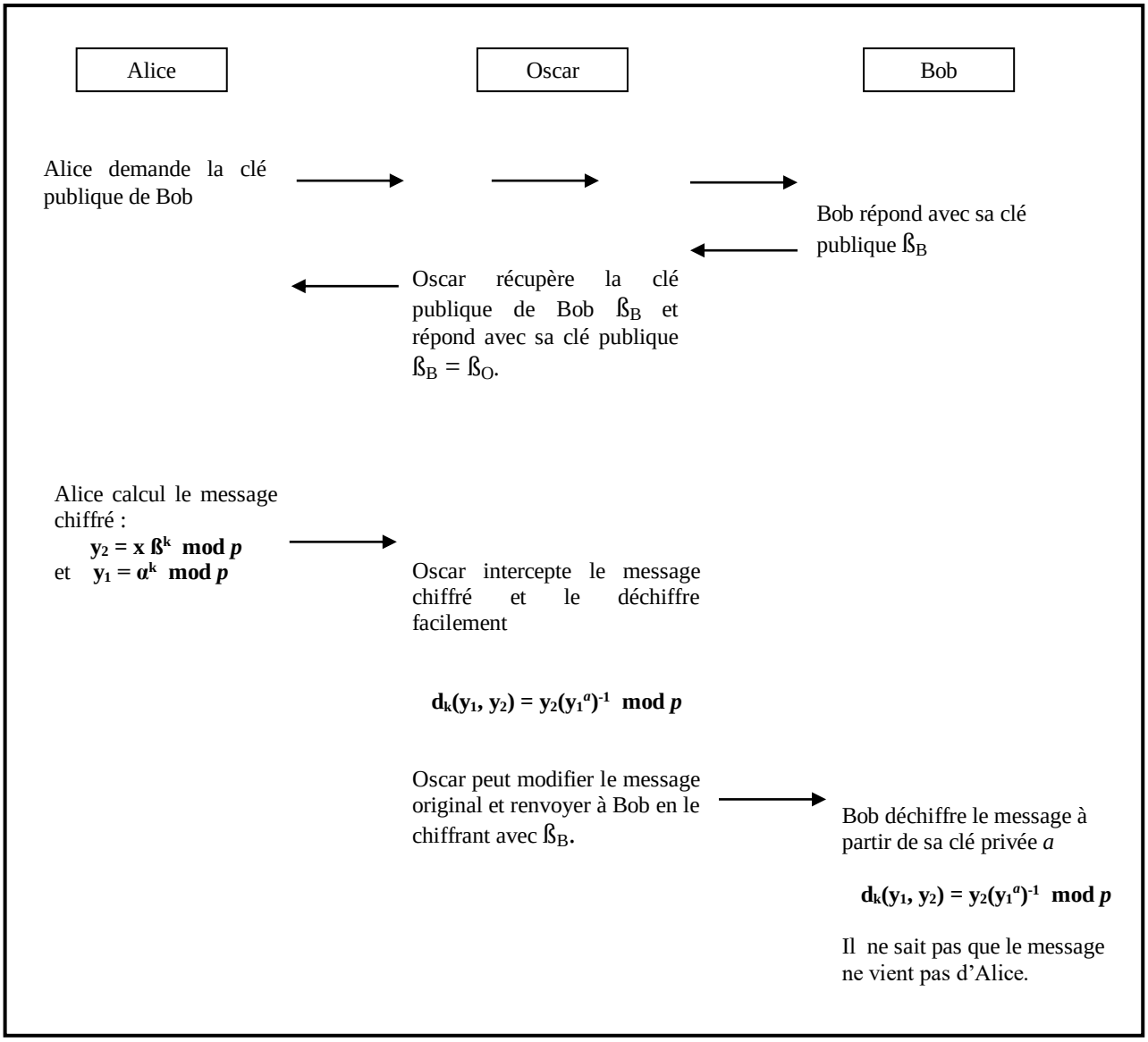



**Figure 5.4.** *Principe de l'attaque « man in the middle ».*

Pour contrer l'attaque, il faut être sûr de la provenance de la clef $ß_B$. Ainsi, deux solutions sont envisageables. La première consiste à signer le message ; la deuxième solution propose d'avoir un recours à un organisme tiers pour certifier la clé.

## 3. Conclusion

Les calculs faits en 1995 ont ouvris un vaste horizon devant le chiffre RSA, du fait que le cassage des clés de 130 chiffres, utilisées à l'époque, nécessite 150 ans. Alors que va-t-on dire avec les clés utilisées aujourd'hui, qui comportent plus de 300 chiffres et qui sont donc plusieurs milliards de fois supérieures ? Donc, la méthode est officiellement sûre si l'on respecte certaines contraintes de longueur de clés et d'usage. Toutefois, personne depuis 2500 ans n'a trouvé de solution rapide au problème de la factorisation, alors il est tout à fait clair, que seule une véritable révolution mathématique ou informatique serait capable de remettre en cause ce cryptosystème.

De même, casser l'algorithme ElGamal est dans la plupart des cas au moins aussi difficile que de calculer le logarithme discret. Cependant, il est possible qu'il existe des moyens de casser l'algorithme sans résoudre le problème du logarithme discret. Et pour assurer une bonne sécurisation de cet algorithme, Zimmermann en 2005 a recommandé l'utilisation des clés d'au

moins 1024 bits. Signalons aussi que le désavantage d'ElGamal par rapport à RSA réside dans le fait que le message chiffré est deux fois plus gros que le clair.

# *Chapitre VI*
# *Cryptosystèmes hybrides*

## 1. Cryptographie hybride - Rappel
### 1.1. Cryptographie asymétrique vs cryptographie symétrique

La cryptographie asymétrique offre des garanties au niveau de la confidentialité, de l'authenticité, et de l'intégrité, mais c'est un système de chiffrement double, donc lent. Le système symétrique est plus rapide, mais nous n'avons pas toujours l'occasion de transmettre la clé secrète par un moyen de transmission sécurisé.

Une cryptographie hybride combine les avantages des deux systèmes :

- La rapidité d'un système symétrique grâce à une clé secrète (One Time Session Key) valide le temps du transfert de l'information, ou le temps d'une session.
- La possibilité de transmettre la clé secrète par une cryptographie asymétrique.

### 1.2. Principe

La cryptographie hybride, dont les étapes de fonctionnement sont illustrées par la figure 6.1, procède de la manière suivante :

- Une clé aléatoire, appelée clé de session, est générée. Elle représente la clé secrète qui sera utilisée par un algorithme symétrique (3DES, IDEA, AES, …) pour chiffrer le message.
- Ensuite, la clé de session sera chiffrée grâce à la clé publique du destinataire. C'est ici qu'intervient la cryptographie asymétrique (RSA ou ElGamal). Comme la clé est courte, ce chiffrement prend peu de temps. Chiffrer l'ensemble du message avec un algorithme asymétrique serait bien plus coûteux, c'est pourquoi on préfère passer par un algorithme symétrique.
- Le message chiffré avec l'algorithme symétrique accompagné de la clé chiffrée correspondante seront envoyés de manière sécurisée vers le destinataire. Ce dernier déchiffre la clé symétrique avec sa clé privée et via un déchiffrement symétrique, il retrouve le message.

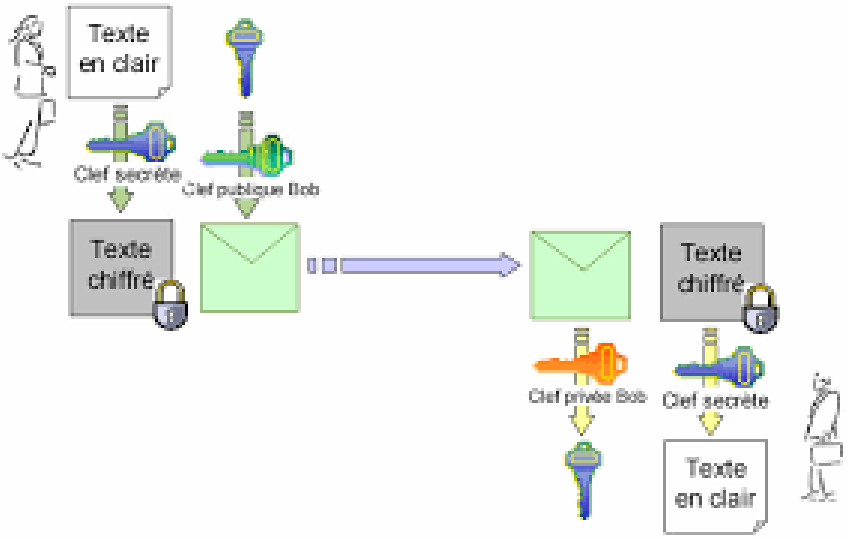



**Figure 6.1.** *Cryptographie hybride.*

## 2. Cryptosystèmes hybrides

### 2.1. PGP

*Philip Zimmermann*, qui est un mathématicien passionné par l'informatique, et en croyant à la philosophie qui dise que tout individu a droit à la confidentialité, notamment les organisations des droits de l'homme dans des pays soumis à la dictature, a commencé à travailler en 1984 sur un système cryptographique aussi sûr mais plus souple que le RSA. Ainsi, il a développé le **PGP** (**Pretty Good Privacy**) en 1991, puis il l'a mis à disposition gratuitement sur Internet sans se préoccuper des détails juridiques qui concernent son utilisation de RSA sans l'accord de son propriétaire, ou de son vendeur, ViaCrypt. Après quelques négociations commerciales et trois ans de menaces judiciaires par le gouvernement américain, PGP est à nouveau accessible depuis 1993, mais à 150 $ cette fois-ci et vendu par... ViaCrypt.

Maintenant, les principes et formats de messages utilisés par PGP ont été normalisés à l'IETF sous le nom **OpenPGP**.

#### 2.1.1. Principe

Lorsqu'un utilisateur chiffre un texte avec le système de chiffrement hybride PGP combinant des fonctionnalités de la cryptographie à clé publique et de la cryptographie symétrique, les données sont d'abord compressées. Cela à pour objectif de réduire le temps de transmission de ces données, et d'économiser l'espace disque et, surtout, le renforcement de la sécurité cryptographique du moment où, les cryptanalystes exploitent les modèles trouvés dans le texte en clair pour casser le chiffrement ; alors que la compression réduit ces modèles dans le texte en clair. Par conséquent, la résistance à la cryptanalyse sera, considérablement, améliorée.

Ensuite, l'opération de chiffrement se fait principalement en deux étapes, qu'on résume comme suit:

- ✓ PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ;
- ✓ PGP crypte la clé secrète IDEA et la transmet au moyen de la clé RSA publique du destinataire.

***Remarque :***

L'**IDEA** (**International Data Encryption Algorithm**), qui est un cryptosystème symétrique inventé en 1992, effectue des opérations du même genre que celles vues avec l'algorithme DES, et il manipule des blocs de 64 bits avec une clé de 128 bits.

En combinant les deux modes de cryptage symétrique et asymétrique, cette méthode de chiffrement profite des avantages de ces deux modes, à savoir la simplicité et la facilité d'utilisation du cryptage asymétrique et la rapidité de calcul du cryptage symétrique. De plus, le cryptage asymétrique résout le problème de la distribution des clés. Ainsi, les performances seront améliorées sans que la sécurité soit affectée.

#### 2.1.2. Cryptanalyse

Depuis 1978, la recherche universitaire civile a intensément attaqué la cryptographie à clé publique, sans pour autant réussir à la remettre en cause. Mais cela ne fournisse aucune garantie totale sur la sécurité de cette manière de chiffrer, car une attaque menée par le gouvernement, par exemple, qui ne se manque pas de ressources très développées ou même en utilisant quelconques nouvelles percées mathématiques classées top-secret, peut tenir à bout ces cryptosystèmes conventionnels utilisés dans PGP.

Tout de même, l'optimisme semble justifié. Les algorithmes de clé publique, les algorithmes de contraction de message, et les chiffres par blocs utilisés dans PGP ont été conçus par les meilleurs cryptographes du monde. Les chiffres de PGP ont subi des analyses de sécurité approfondies et des examens méticuleux de la part des meilleurs cryptographes dans le monde non

classé top secret. De plus, et même si les chiffres par blocs utilisés dans PGP possède quelques faiblesses, la compression du texte clair utilisée avant le chiffrement réduit de façon considérable ces faiblesses.

### 2.2. GPG

**GPG** (**Gnu Privacy Guard**) est dans le principe un clone de PGP, ou plus exactement une implémentation de l'OpenPGP, mais n'utilise aucun code de PGP. Donc, c'est l'équivalent libre de PGP. Il est entièrement écrit par des développeurs bénévoles, et est complètement libre, sous licence GPL (General Public License). Ainsi, il est remis à jour continuellement, aussi bien au niveau des fonctionnalités, qu'au niveau des éventuels problèmes d'implémentation.

D'un autre coté, de nombreux gouvernements ont restreint, aujourd'hui, l'usage du PGP, qui a été largement diffusé par son développeur, en pensant qu'un cryptage trop fiable ferait le jeu des terroristes et des trafiquants. Toutefois, il y'a pas mal d'applications qui utilisent encore ce système de chiffrement, telles que les paiements en ligne qui se font grâce au procédé SSL fonctionnant selon le principe du PGP.

De sa part, GPG, qui est un cryptosystème utilisant des algorithmes de chiffrement à clé publiques (DSA, RSA et ElGamal), est largement utilisé dans les communications par messagerie, c-à-d pour chiffrer des mails, ainsi que pour signer des données en garantissant, ainsi, leurs authenticité, intégrité et confidentialité.

#### 2.2.1. Historique

Bénéficiant d'un financement important de la part du ministère fédéral de l'Économie d'Allemagne, le projet est initié à la fin des années 1990 par Werner Koch dans le but de remplacer la suite PGP de logiciels cryptographiques (plus précisément, de cryptographie asymétrique) par une alternative en logiciel libre.

- 7 septembre 1999 : La première version stable est publiée.
- décembre 2013 : une première campagne de financement participatif s'est lancée, dans le but de créer un site web plus attractif, améliorer la documentation et sortir la version 2.1 de GnuPG. En 24 heures, 90 % de l'objectif (24 000 €) ont été atteints.
- Depuis 2015, la Core Infrastructure Initiative assiste le projet à hauteur de plus de 50 000 € par an.

#### 2.2.2. Caractéristiques

Comme pour tous les procédés de chiffrement asymétrique, le principal inconvénient de GnuPG est que la clé privée doit être enregistrée quelque part. Si c'est sur une clé USB que l'on garde avec soi, les risques de perte, de vol ou de copie existent. Si elle on la garde sur le disque dur d'un ordinateur, on est alors exposé aux risques classiques du piratage. Toutefois et pour limiter les risques, utiliser une phrase (ou mot) de passe, optionnelle peut protéger la clé privée.

Depuis novembre 2014, GnuPG est maintenu dans trois branches :

- la branche classique : portable mais ancienne, dont la dernière version est la 1.4.2x ;
- la branche stable : offre plus de fonctionnalités (par exemple le support des certificats X.509), dont la dernière version est la 2.0.3x ;
- la branche moderne : présente de nouvelles fonctionnalités (par exemple le support de la cryptographie sur les courbes elliptiques), dont la dernière version est la 2.2.x.

Depuis sa version 2.0, GnuPG peut être installé sur une carte à puce dont le code PIN peut etre utilisé pour protéger la clé privée. Cela améliore sensiblement la confidentialité.

### 3. Recommandation de longueur de clés

Depuis des décennies, les algorithmes de chiffrement ont sans cesse été découverts, améliorés et … cassés ! Le tableau, ci-dessous, présente les recommandations de longueur de clé.

| Date | Robustesse de sécurité | Algorithmes symétriques | Module de factorisation | Logarithme discret |
|---|---|---|---|---|
| Legacy | 80 | 2TDEA | 1024 | 160 |
| 2019 – 2030 | 112 | (3TDEA)<br>AES-128 | 2048 | 224 |
| 2019 – 2030<br>& au-delà | 128 | AES-128 | 3072 | 256 |
| 2019 – 2030<br>& au-delà | 192 | AES-192 | 7680 | 384 |
| 2019 – 2030<br>& au-delà | 256 | AES-256 | 15360 | 512 |

TDEA : Triple Data Encryption Algorithm

Toutes les tailles de clé sont fournies en bits. Ce sont les tailles minimales pour la sécurité.

Des algorithmes et des longueurs de clé pour une force de sécurité de 80 bits peuvent être utilisés en raison de leur utilisation dans des anciennes applications (c'est-à-dire qu'ils peuvent être utilisés pour traiter des données protégées par cryptographie). Ils ne doivent pas être utilisés pour appliquer une protection cryptographique (par exemple, le cryptage).

# Bibliographie

A. Kerckhoff, la cryptographie militaire, Journal des Sciences Militaires, vol IX, pp 5-38, janvier 1883.

B. Schneier, Cryptographie appliquée, Ed. Vuibert, 2001.

Ch. Paar and J. Pelzl , Understanding Cryptography: A Textbook for Students and Practitioners, Springer, 2009.

D. Coppersmith, The Data Encryption Standard (DES) and its strength against attacks, IBM Journal of Research and Development, 38(3), pp 243 – 250, 1994.

D. Giry and J. Quisquater, Bluekrypt-cryptographic key length recommendation, resource online: http://www. keylength. Com, 2020.

D.R. Stinson, Cryptography: Theory and Practice, Chapman and Hall/CRC, 3rd edition, 2005.

G. Dubertet, Initiation à la cryptographie : Cours et exercices corrigés, Ed. Vuibert, 2018.

G.Paul and S.Maitre, RC4 Stream cipher and its variants (Discrete Mathematics and its applications), CRC Press, 2011.

J-Ph. Aumasson, Serious Cryptography: A Practical Introduction to Modern Encryption, No Starch Press, 2017.

J. Daemen and V.Rijmen, The Design of Rijndael: AES — The Advanced Encryption Standard, Springer, 2002.

N. Koblitz, A course in Number Theory and Cryptography (second edition), Ed. Springer, 1994.

S. Garfinkel, PGP : Pretty Good Privacy, O'Reilly & Associates, 1995.

S. Singh, The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography, Anchor, 2000.

W.Diffie and M.Hellman, New directions in cryptography, IEEE Transactions on information theory, IT-22(6), 1976.

# Annexe

# Exercices

***Exercice n° 1 :***

Un chiffrement par substitution permute les caractères de l'alphabet. Dans un chiffrement par transposition, les symboles du message demeurent inchangés, mais leur ordre est permuté par une permutation des positions d'indice. A la différence de chiffrements par substitution, les chiffrements par transposition sont des chiffrements par blocs.
Considérons le message en clair m = INFORMATIQUE.

**1)** Trouver le chiffrement de m avec un chiffrement par transposition utilisant la clé k = [3,2,1,4].
**2)** Soit C un message chiffré

C = AEUFQ RUTRI QSUTX MEANE YNNAU DEESX OAUTX NUTAX

a- Trouvez le message en clair M avec la clé K = 5x8, lecture des colonnes 2-1-8-4-3-7-5-6
b- Chiffrez le message trouvé M avec transposition complexe par colonnes avec la clé K = ECRITURE et selon l'ordre alphabétique des caractères de la clé.

***Exercice n° 2 :***

On considère un diagramme de Feistel à deux rondes sur des chaînes de 8 bits avec deux fonctions $f_1$ et $f_2$. On pose :

$$f_1(a) := a \oplus 1011 \quad \text{et} \quad f_2(a) := \bar{a} \oplus 0101$$

pour toute chaîne *a* de 4 bits.

**1)** Calculer l'image de la chaîne 11010011 par ce diagramme.
**2)** Déterminer une chaîne de 8 bits dont l'image par le diagramme est elle-même.

***Exercice n° 3 :***

On considère une fonction de chiffrement par bloc de longueur 2 pour des clefs de longueur 2 donnée par :

$$\begin{array}{rcl} E_k : \{0,1\}^2 & \to & \{0,1\}^2 \\ (m_1, m_2) & \mapsto & S_1((m_1 \oplus k_1, m_2 \oplus k_2)) \end{array}$$

où la fonction S1 est décrite ci-dessous.

| $X$ | $[0,0]$ | $[1,0]$ | $[0,1]$ | $[1,1]$ |
|---|---|---|---|---|
| $S_1(X)$ | $[1,1]$ | $[1,0]$ | $[0,0]$ | $[0,1]$ |

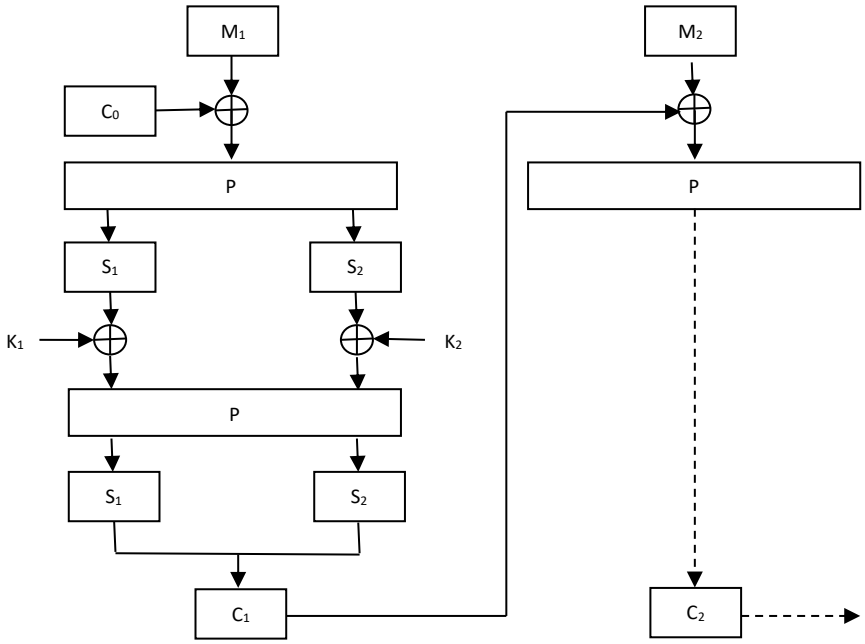
**1)** Chiffrer le message M = [0,1,1,1,0,1] avec la clef K = [1,0] et en utilisant :
- **(a)** le mode ECB
- **(b)** le mode OFB

**2)** Déchiffrer le message C = [0,1,1,1,0,1] dans le cas où il a été chiffré avec la clef K = [1,1] et en utilisant :
- **(a)** le mode CBC,

**(b)** le mode CFB.

***Exercice n° 4 :***

On considère le cryptosystème illustré par la figure suivante :




| X | [0,0] | [1,0] | [0,1] | [1,1] |
|---|---|---|---|---|
| $S_1$ | [1,1] | [1,0] | [0,0] | [0,1] |
| $S_2$ | [1,0] | [0,1] | [1,1] | [0,0] |

avec : $K = [k_1,k_2,k_3,k_4]$, $K_1 = [k_1 \oplus k_2, k_3 \oplus k_4]$, $K_2 = [k_1 \oplus k_3, k_2 \oplus k_4]$
et $P(1) = 3, P(2) = 4, P(3) = 2, P(4) = 1$.

**1)** Chiffrez le message M = [1,1,1,1] avec K = [1,0,1,1] et $C_0$ = [0,0,1,1]
**2)** Déchiffrez le message C = [0,1,0,1] chiffré avec la même clef.
**3)** En s'inspirant des modes opératoires, dégager les avantages et les limites de ce schéma.

***Exercice n° 5 :***

En adoptant le nouveau paramétrage DES suivant, chiffrer le message :

M = 1101001010101100.

- Taille de blocs = 8 bits,
- $IP = \begin{bmatrix} 8 & 6 & 4 & 2 \\ 7 & 5 & 3 & 1 \end{bmatrix}$ et $IP^{-1} = \begin{bmatrix} 6 & 2 & 8 & 4 \\ 5 & 1 & 7 & 3 \end{bmatrix}$,
- pour la fonction de substitution, utilisez la S1,
- $P = \begin{bmatrix} 3 & 1 & 2 & 4 \end{bmatrix}$,
- soit la suite de bits 11100101. De droite à gauche et en négligeant la première et l'avant dernière étapes dans l'algorithme d'obtention d'une clé DES, déduisez la clé formée du nombre de bits nécessaires.
- le schéma modifié sera itéré 2 fois.

***Exercice n° 6 :***

On considère le système RSA avec p = 19 et q = 23.

**1)** Calculer $n$ et $\varphi(n)$.
**2)** Déterminer l'exposant $a$ associé à b=9, b=14, puis b=5.
**3)** Calculer le chiffré associé au message m=42 quand b=5.
**4)** Déchiffrer le message c=264 avec la clé $a$ correspondante à b=5.

***Exercice n° 7 :***

**1)** Chiffrer le message "25" avec la clé publique RSA (b=13,n=77). Le calcul peut être facilement fait en remarquant que $25^4 \bmod 77 = 4$.
**2)** Sachons que n= $11 \times 7 = 77$, calculer la clé privée associée à la clé publique (b=13,n=77). Le calcul peut se faire facilement en remarquant que : $13 \times 37 = 481$ et $8 \times 60 = 480$.
**3)** Déchiffrer le message obtenu à la question (1) afin de trouver le message clair.

***Exercice n° 8 :***

Soit à chiffrer le message X = 6882326879666683 en utilisant l'algorithme RSA avec la clé publique (79, 3337). Pour pouvoir le déchiffrer, on procède à une factorisation de n en deux nombres de m chiffres.

**1)** Quelle est la valeur maximale de m ?
**2)** Le bloc codé 6325 peut-il résulter d'un codage avec la clé publique ? Même question avec la clé privée.

***Exercice n° 9 :***

**1)** Donner deux raisons distinctes pourquoi l'entier n = 7517137650 ne serait pas un bon modulo pour l'algorithme RSA.
**2)** Alice et Bob communiquent entre eux grâce à un cryptage RSA. Alice construit ses clefs publiques et secrètes ainsi : Elle choisit comme clef secrète (25, 133), ce qui lui permet d'utiliser le bon vieux code ASCII, puis en déduit sa clef publique (13, 133) qu'elle transmet en clair à Bob.
**2.1)** Vérifier que ces clefs sont cohérentes.
**2.2)** À l'aide de sa clef secrète, Alice envoie la lettre «I» à Bob (code ASCII : 73 en décimal).
- **a-** Que reçoit Bob ? (valeur en décimal)
- **b-** Faites le calcul de Bob pour retrouver le message d'Alice.
- **c-** Comment Eve pourrait-elle trouver la clef secrète d'Alice ?

**3)** Sachant que le même message *m* a été envoyé à 3 personnes différentes A1, A2, A3 et que ces trois personnes utilisent les clés publiques RSA (3, N1), (3, N2), (3 ,N3) respectivement. Expliquer comment un attaquant peut retrouver *m* à partir des chiffrés (*c1*, *c2*, *c3*) de *m* envoyés respectivement à A1, A2, A3 et des clés publiques.