Université d'Avignon et des Pays de Vaucluse
CERI, M1-M2 Alternance, 2009-2010

Sécurité des Systèmes d'Information et des Réseaux : Cryptographie

Khalil IBRAHIMI

Laboratoire d'Informatique d'Avignon (LIA)

Contact : khalil.ibrahimi@univ-avignon.fr

Plan

Références

1- Titre : *Cryptographie appliquée : protocoles, algorithmes, et code source en C,*
Auteur : Bruce Schneier.

2- Titre : *Handbook of Applied Cryptography,*
Auteurs : A. Menezes, P. van Oorschot and S. Vanstone.
http://www.cacr.math.uwaterloo.ca/hac/index.html

Chapitre 1 : Concepts généraux et définitions

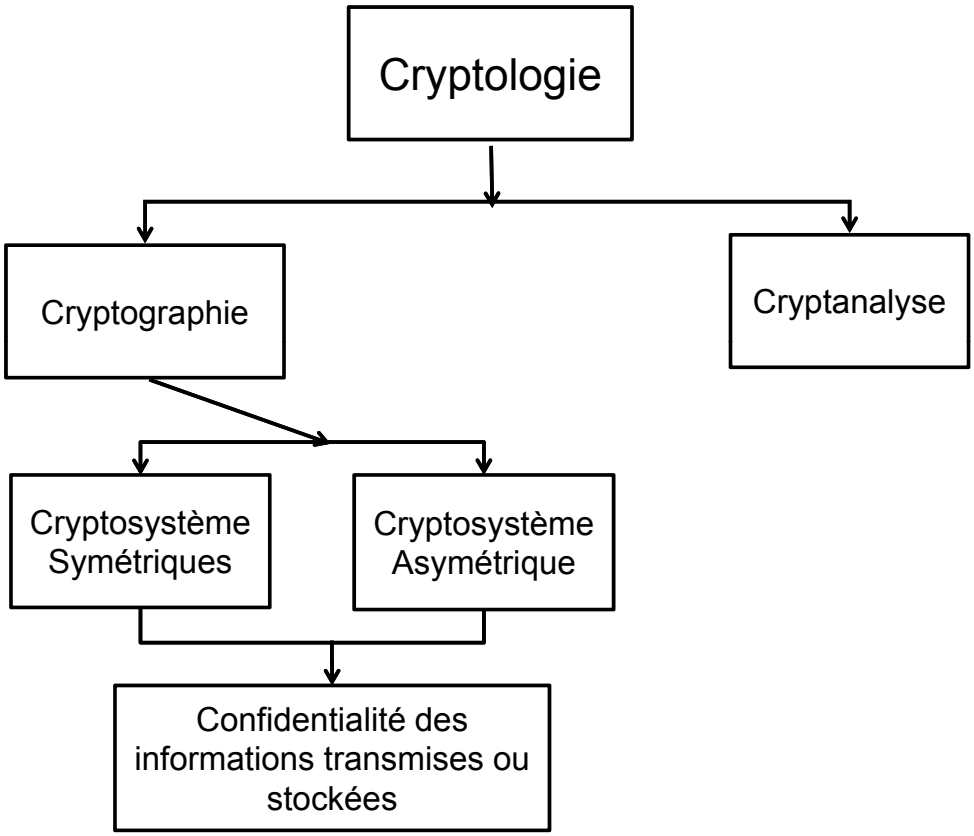


Schéma général de la cryptologie

Quelques applications civiles :

- Protocoles cryptographiques sécurisés tels que :
  - Protocole d'IBM de gestion de clés secrètes,
  - Kerberos,
  - SSL,
  - IPsec, …
- Système de paiement sécurisé
- Carte à puce et transactions bancaires
- Échange confidentiel de données médicales, juridiques, …
- Réseaux téléphoniques (Téléphonie fixe et mobile, fax, …)

A quoi sert la cryptographie?

- Confidentialité des informations stockées ou transmises.
  - Seuls les utilisateurs *autorisés* peuvent accéder à l'information.
- Intégrité des informations stockées ou transmises.
  - Seuls les utilisateurs *autorisés* peuvent modifier l'information.
- Authentification des utilisateurs.
  - L'utilisateurs est-il ou non *autorisé* ?
- Rendre une information incompréhensible.

Concepts généraux et définitions

- Objectif principal d'un cryptosystème est de chiffrer un message en clair en un message chiffré appelé cryptogramme. Ce message ne doit pas être déchiffré par les cryptanalystes, décrypteurs, hackeurs, …
- Il doit être uniquement déchiffré par le destinataire légitime. Un cryptosystème est caractérisé par cinq composants :
  - Un espace **M** = {M1, M2, …} des messages en clair;
  - Un espace **C** = {C1, C2, …} des messages chiffrés;
  - Un espace **K** = {K1, K2, …} des clés;

Concepts généraux et définitions

– Un algorithme de chiffrement paramétré par une clé K dans **K**

$$\begin{aligned} \mathbf{E}_k : \mathbf{M} &\longmapsto \mathbf{C} \\ m &\longmapsto c = \mathbf{E}_k(m) \end{aligned}$$

– Un algorithme de déchiffrement paramétré par une clé K' dans **K**

$$\begin{aligned} \mathbf{D}_{k'} : \mathbf{C} &\longmapsto \mathbf{M} \\ c &\longmapsto m = \mathbf{D}_{k'}(c) = \mathbf{D}_{k'}(\mathbf{E}_k(m)) \end{aligned}$$

Concepts généraux et définitions

Cryptosystème :

– Message en clair (**M**) + algorithme de chiffrement (**E**) avec la clé **K**; **ou bien**
– Algorithme de déchiffrement (**D**) + Message chiffré (**C**) + toutes les clés possibles **K** et **K'**.

Un cryptosystème moderne doit satisfaire les conditions suivantes :

- Les algorithmes de chiffrements (E) / déchiffrement (D) doivent être opérationnels pour tout clé K, stables ils sont publiquement connus (grand public, …);
- La sécurité de système doit reposer uniquement sur le secret des clés K et K'.

=> Ce qui donne naissance aux cryptosystèmes à usage général.

Cryptosystèmes à usage générale

Les cryptosystèmes modernes sont conçus en tenant compte des conditions précédentes. Ils forment deux classes de cryptosystèmes : **Symétrique et Asymétrique**.

- Les cryptosystèmes symétriques sont synonymes de systèmes cryptographiques à clés secrètes ou privées. Une même clé **K** est utilisée pour le chiffrement et déchiffrement c'est-à-dire **K = K'**.
- Généralement un seul algorithme est utilisé pour le chiffrement et le déchiffrement (**D = E**).
- Le secret de la clé de chiffrement / déchiffrement est donc partagé entre émetteur **A** et récepteur **B**.

Cryptosystèmes à usage générale

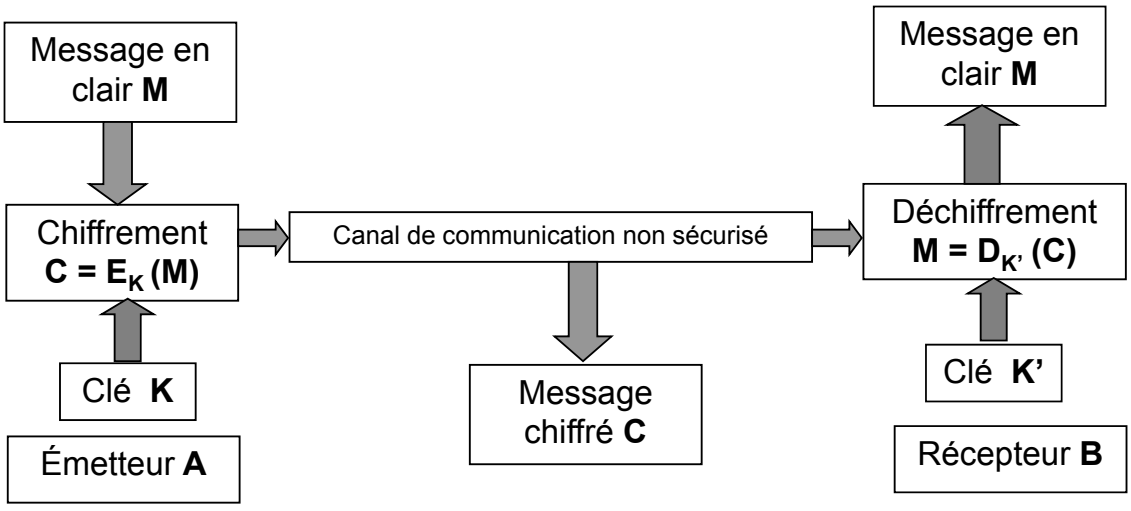


Fig 1 : **Synoptique générale d'un cryptosystèmes à usage générale**

Cryptosystèmes à usage générale

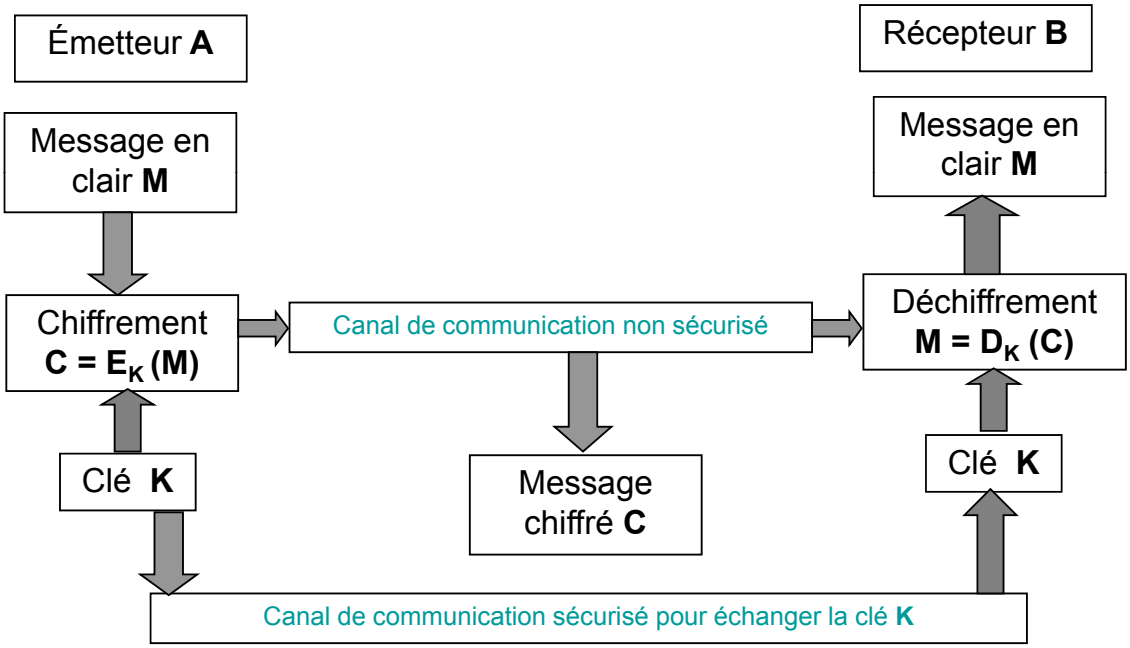


Fig 2 : **Synoptique générale d'un cryptosystèmes à clé symétrique**

Cryptosystèmes à clé symétrique

- Avantages
  - Rapidité de chiffrement / déchiffrement,
  - Confidentialité locale par un seul utilisateur (protection de fichiers dans une machine),
  - Champs d'application très vaste (Banques, communications téléphoniques),
  - Longueur de la clé relativement petite codée entre 40 bits et 256 bits.
- Inconvénients
  - Si la clé est compromise (volée, piratée, …) le système n'est plus fiable et donc plus de confidentialité,
  - Dans un réseau de **N** correspondants, il faut distribuer **N (N-1)/2** clés par des canaux sûr et donc problème de distributions des clés par des voies sécurisées.

Cryptosystèmes à clé symétrique

- Outils
  - Substitution,
  - Transposition,
  - Ou Exlusif,
  - Décalage logique,
  - Combinaison des fonctions ci-dessus.

- Exemples de réalisation
  - DES,
  - IDEA,
  - MARS,
  - 3DES, …

Cryptosystèmes à usage générale

- Le concept de cryptosystèmes asymétriques a été introduit, en 1976, par Delfie Hellman. L'idée de base est repose essentiellement sur deux algorithmes $\mathbf{E}_k$ différent de $\mathbf{D}_{k'}$ pour **K** différent de **K**' (voir la figure 3 ci-après).

- Ce cryptosystème met en jeu deux clés, une pour le chiffrement (**K**) et une autre pour le déchiffrement (**K**'). De ce fait, chaque utilisateur doit posséder une paire de clé (**K**, **K**'). La clé **k** est appelée publique **Kp :** elle peut être rendue publique dans un annuaire, une base de donnée. L'autre clé **K**' est appelée privée **Kpr** doit être garder secrète et n'est connue que de son propriétaire.

- La clé **Kp** est utilisée pour le chiffrement du message en clair M.

- La clé **Kpr** est utilisée pour le déchiffrement du message chiffré C.

Cryptosystèmes à usage générale

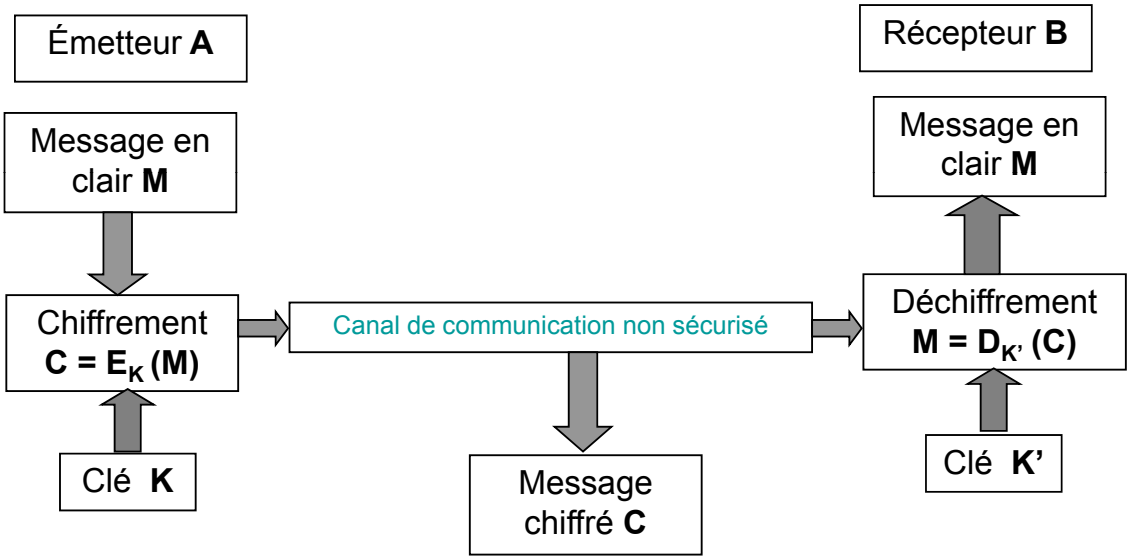


Fig 3 : **Synoptique générale d'un cryptosystèmes à clé asymétrique**

Aspects techniques de la cryptographie asymétrique

- Différente possibilités d'utilisation de **Kp** et **Kpr**

1. Confidentialité (voir la figure 4)
   - Il s'agit de garantir le secret de l'information transmise ou archivée.
   - Soit **A** un utilisateur avec la paire de clé suivante : **Kp^A** la clé publique, **Kpr^A** la clé privée,
   - Soit **B** un utilisateur avec la paire de clé suivante : **Kp^B** la clé publique, **Kpr^B** la clé privée,
   - **A** est l'émetteur du message chiffré et **B** est le récepteur de ce message.

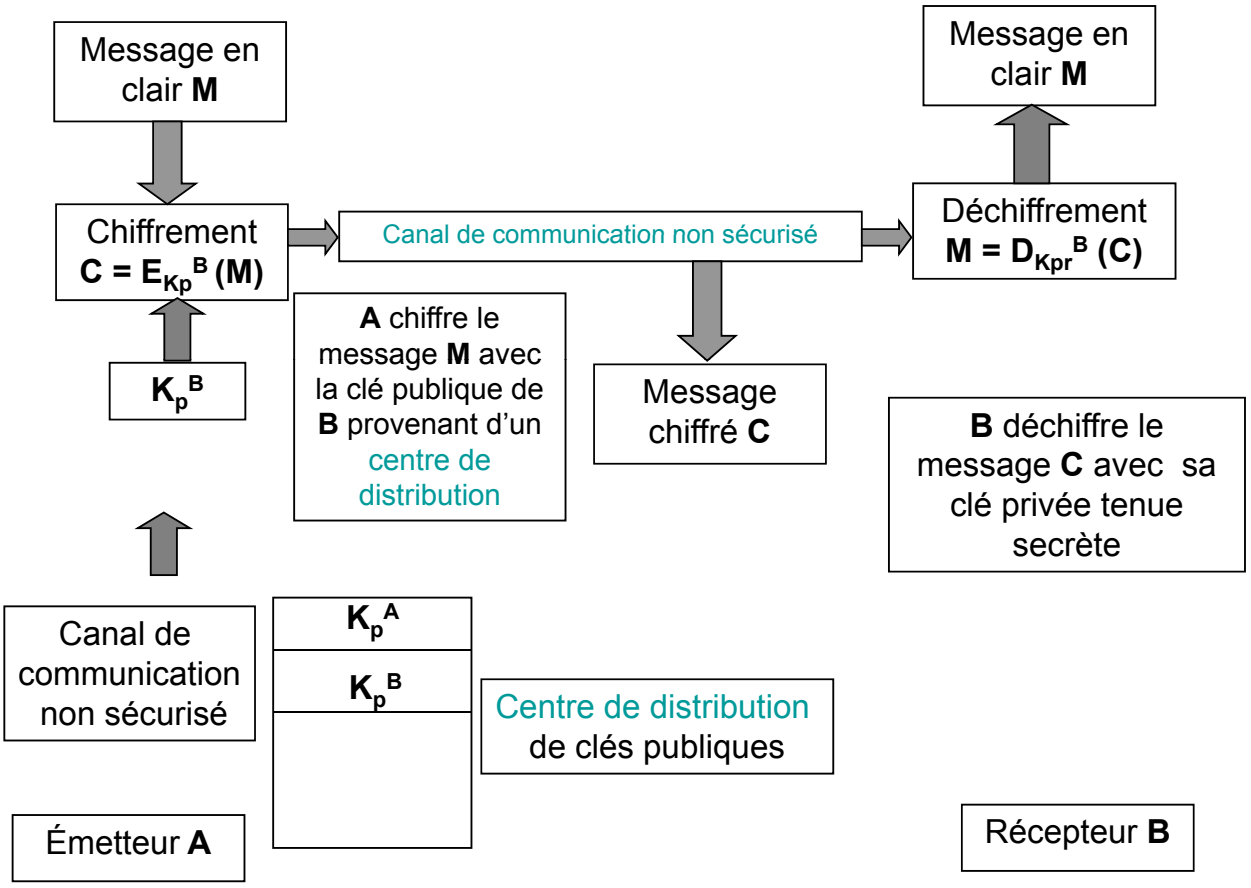


Fig 4 : **Synoptique du mécanisme de confidentialité**

Aspects techniques de la cryptographie asymétrique

- Dans ce schéma, le chiffrement / déchiffrement est réalisé uniquement en utilisant la paire de clé de **B** (K_p^B, K_{pr}^B).
- **A** chiffre le message **M** avec la clé publique **B** ➔ **C** = Ek_p^B (**M**).
- **B** déchiffre le message **c** avec sa clé privée ➔ **M** = Dk_{pr}^B (**C**).
- L'inconvénient majeur est que l'auteur du message n'est pas forcément identifié et peut être anonyme (que les clés publique qui sont connus).

Aspects techniques de la cryptographie asymétrique

2 Authentification (voir la figure 5)

- Il s'agit de garantir l'origine d'une information (le courrier électronique, un bon de commande transmis en ligne),
- **A** chiffre le message **M** avec sa clé privée tenue secrète

  $(\mathbf{K_{pr}^A})$ $(\mathbf{C} = \mathbf{Ek_{pr}^A(M)}$,
- **B** déchiffre le message **c** avec la clé publique de **A** $(\mathbf{k_p^A})$

  $\mathbf{M} = \mathbf{Dk_p^A(C)} = \mathbf{Dk_p^A[Ek_{pr}^A(M)]}$,
- Dans ce schéma, le chiffrement / déchiffrement s'effectue en utilisant uniquement la pair de clé de **A** $(\mathbf{k_p^A}, \mathbf{k_{pr}^A})$.

Aspects techniques de la cryptographie asymétrique

- **B identifie A** en utilisant la clé publique de **A** pour déchiffrer le message **C** ce qui donne **l'authentification** de **A.**
- Pas de **confidentialité** puisque la clé publique de **A** est connue et publique (n'importe qui peut déchiffrer le message **C**).

3 Intégrité

– Il s'agit de garantir la confidentialité et l'authenticité,

– **A** chiffre le message avec sa clé privée $(\mathbf{k_{pr}^A})$, puis le résultat sera

Aspects techniques de la cryptographie asymétrique

ensuite chiffré par la clé publique de **B** (k_{pr}^{B}),

1. $C = E_{kpr}^{A}(M)$
2. $C' = E_{kp}^{B}(C) = E_{kp}^{B}[E_{kpr}^{A}(M)]$

– **B** déchiffre le message **c**' avec sa clé privée (k_{pr}^{B}), puis le résultat sera ensuite déchiffré par la clé publique de **A** (k_{pr}^{A}),

1. $D_{kpr}^{B}(C') = C$
2. $E_{kp}^{A}(C) = E_{kp}^{A}[D_{kpr}^{B}(C')]$
   $= E_{kp}^{A}[D_{kpr}^{B}(E_{kp}^{B}[E_{kpr}^{A}(M)])] = M$

- **Hypothèse fondamentale :**
  On ne doit pas pouvoir trouver la clé privée à partir de la clé publique (connue par tous le monde et circule sur des canaux non sécurisés).

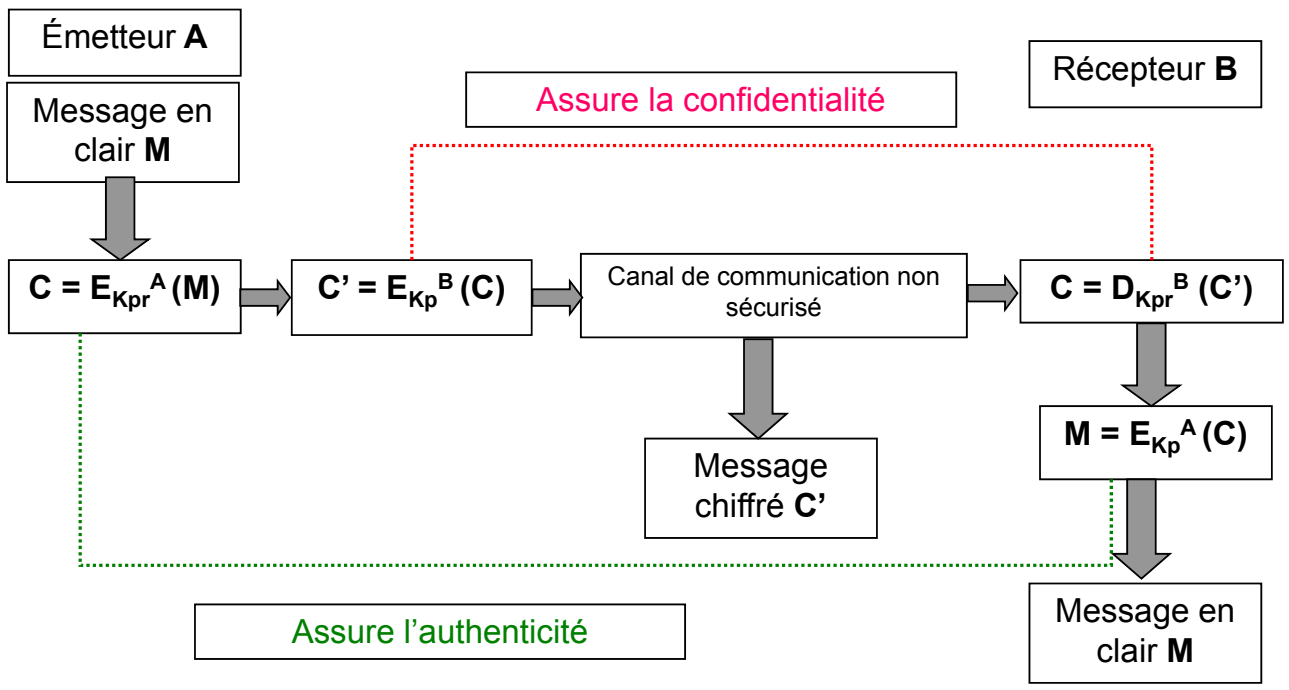


Fig 5 : **Schéma du mécanisme de confidentialité / authentification**

Cryptosystèmes à clé asymétrique

- Avantages
  - Échange de clé publique sur un canal non sécurisé (pas besoin d'un canal sécurisé),
  - Possibilité de création de base de données des clés publiques,
  - Authentification du message grâce à la signature numérique,
  - Nombre de clés croît linéairement avec le nombre d'utilisateurs ➔ N utilisateurs ➔ N pairs de clés.
- Inconvénients
  - Temps de calcul mathématique et espace mémoire relativement important, donc moins rapides que les cryptosystèmes symétriques,
  - Sensible aux attaques à message clairs choisis (clé publique),
  - Validité des clés publiques ➔ problème de certifications des clés publiques ➔ confiance à une tiers personne ➔ organisme de distribution des clés,
  - Longueur des clés très grandes : nombre premiers aléatoires avec une certaine probabilité codée entre 1024 et 4096 bits.

Cryptosystèmes à clé asymétrique

- Outils
  - Théorie des grands nombres premiers,
  - Fonctions d'exponentiation modulaires.
- Exemples de réalisation
  - RSA, Diffie-Hellman,
  - El Gamal, …
- Cryptosystème hybrides

  L'idée de base consiste à utiliser un cryptosystème asymétrique pour échanger les clés d'un cryptosystème symétrique. Ce système combine donc tous les avantages des deux cryptosystèmes symétriques et asymétriques.

Cryptosystèmes hybride

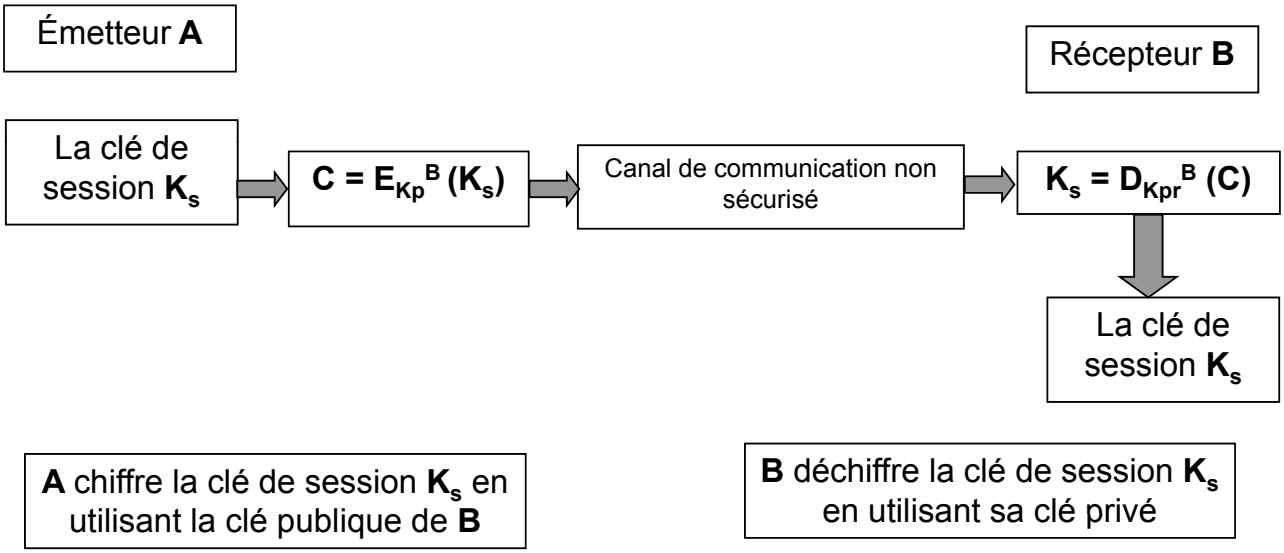


Fig 6 : **Schéma d'un cryptosystème hybride : échange de clé de session (cryptosystème asymétrique)**

Cryptosystèmes hybride

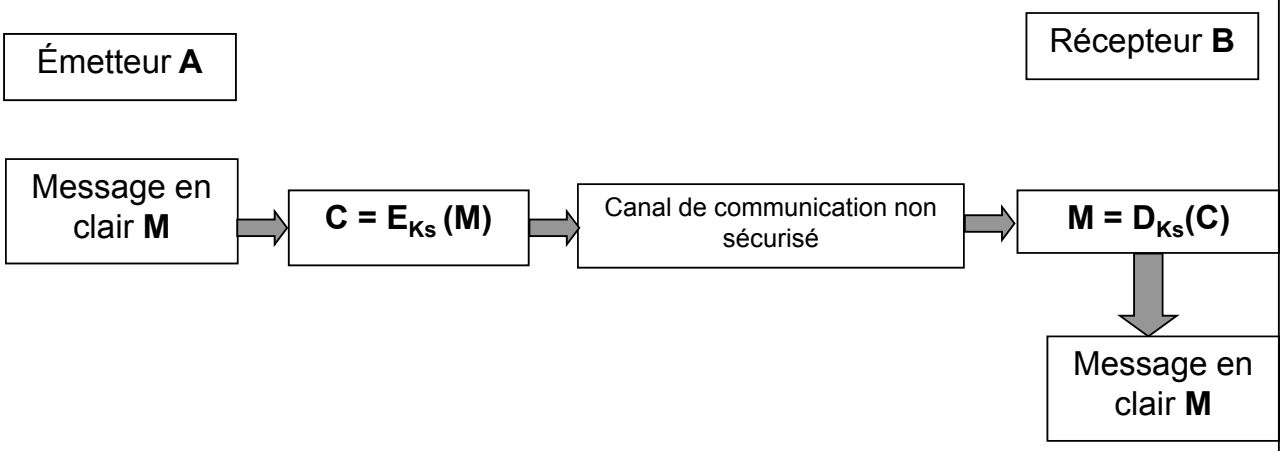


Fig 7 : **Schéma d'un cryptosystème hybride : communication (cryptosystème symétrique)**

Exemple de réalisation :

PGP (Pretty Good Privacy)

Chapitre 2 : Aspects techniques de la cryptographie symétrique

Cryptographie classique

- Substitution

Une substitution est un remplacement d'alphabet. A chaque caractère ou groupe de caractères du message en clair on substitue un autre caractère ou groupe de caractères. Ce procédé doit augmenter la confusion de façon à compliquer la liaison entre le message en clair. La substitution inverse redonne le message en claire.

1 Substitution monoalphabétique

Soient A = {a0, a1, …, an-1} les caractères alphabétiques ordonnés dans le message en clair et C = {f(a0), f(a1), …, f(an-1)} les caractères alphabétiques utilisés dans le message chiffré. La fonction f est la fonction de chiffrement. Soit M = {m0, m1, …, mr-1} le message en clair où mi sont les caractères du message M de A.
La clé et l'algorithme de chiffrement sont combinés et représentés par

$$C = Ek(M) = f(m0)\ f(m1)\ \ldots f(mr-1)$$

Cryptographie classique

Les substitutions monoalphabétiques à décaler de K positions modulo le nombre de caractère de A. La forme explicite de la fonction f est donnée par :

$$f(a) = (a+K) \bmod n$$

Où

- a: caractère à chiffrer
- K: nombre de décalage, clé de chiffrement
- n: dimension de A.

Exemple :

A : ABCDEFGHIJKLMNOPQRSTUVWXYZ (les caractères alphabétiques majuscule du français standard)

M : RENAISSANCE

n: 26

K :3 (chiffrement de Jule Cesar)

Cryptographie classique

Exercice :

Trouvez l'ensemble de l'alphabets de substitution C?

Chiffrez le message en clair M?

Déchiffrez le message chiffré C = Ek(M)?

Vérifiez que la redondance dans le message en clair apparaît dans le message chiffré et aussi si l'ensemble C est choisie aléatoirement?

Cryptographie classique

2. Substitution polyalphabétique

Dans la substitution monoalphabétique la fréquence de distribution des caractères du message en clair est préservée dans le message chiffré (redondance ce qui donne la possibilité de l'analyse statistique). Pour remédier ce problème, la substitution polyalphabétique consiste à utiliser plusieurs alphabets de substitution. Son principe est basé sur une clé (un mot, une phrase, …) et d'une matrice d'alphabets de substitution C1, …, Cn avec la fonction de chiffrement associée à Ci :

$$f_i(A) = C_i \text{ pour } i = 1, \ldots, n$$

Cryptographie classique

En pratique le nombre d'alphabets utilisées dans le chiffrement est égale au nombre utilisés dans la clé. De ce fait, la clé de chiffrement doit donc être constituée de d caractère, à chaque caractère de la clé on attribut un alphabet de substitution Ci. Donc, cette clé sera répétée autant de fois que nécessaire pour chiffrer le message M.

M = m1, …, md, md+1, …, m2d, …,
Clé = k1, …, kd, k1, …, kd, …,
Ek(M) = f1(m1), …, fd(md), f1(md+1), …, fd(m2d), …,

Les fonctions fi peuvent être à décalage :
$f_i(m_i) = (m_i + k_i) \bmod n$,
Où
Ki (i=1, …, d) clé de chiffrement, est la valeur de décalage de la iéme alphabet de substitution par rapport à l'alphabet standard A.

- Exemple : Chiffrement de Vigènere à décalage (1863)

La matrice de l'alphabet est obtenue en choisissant la valeur de Ki égale à la kiéme position du caractère dans l'alphabet A. Cette matrice propose 26 alphabets de substitution :

Clé Utilisée \ Lettre en clair	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	Lettre chiffrée
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	

Exemple en utilisant la matrice de l'alphabets de substitution

K = K1 K2 K3 K4 (d=4, 4 alphabets de substitution parmi 26)
K1 = 1 (mod 26)
K2 = 0 (mod 26)
K3 = 13 (mod 26)
K4 = 3 (mod 26)

La clé est donc égale K = BAND

Message en clair

M = INFO RMAT IQUE S

K = BAND BAND BAND B
Message chiffré

C = Ek(M) = JNSR SMNW JQHH T

3 Substitution par polygrammes

Cette technique permet de chiffrer les carctères par blocs de deux caractères (bigramme) ou trois caractères (trigramme), …. L'algorithme le plus connu est play fair Wheastone (1854) qui a été
utilisé par les anglais durant la première guerre mondiale.

Dans cet algorithme, basé sur la substitution bigramme, on utilise une matrice alphabétique carré (5x5) de 25 caractères dont les caractères dépendent du choix d'une clé K.

a- Préparation du message à chiffrer

1- Elimination des signes, ponctuation et on groupe les lettres par groupe de deux

2- Si une paire se compose de deux lettres identiques la 2ème lettre sera remplacée par X.

3- Le caractère J est traité comme I.

b- Construction de la matrice carré

Le mot clé, choisi, est inscrit horizontalement sans répéter aucun caractère puis les caractères restants de l'alphabet sont écrits en respectant leur ordre alphabétique et en traitant I et J comme IJ.

c- Règle à suivre pour chiffrer le bigramme mi mj d'un message M

1- Si mi et mj occupent la même ligne ➔ ci et cj sont les caractères de substitution situés respectivement à droite de mi et mj.

2- Si mi et mj sont dans la même colonne ➔ ci et cj sont les caractères de substitution situés respectivement en dessous de mi et mj.

3- Si mi et mj, se trouvent dans des lignes et colonnes différentes ➔ ci (ou cj) sera le caractère qui se trouve à l'intersection de la ligne mi (ou mj) et de la colonne de mj (ou mi).

d-Exemple :

Avec la clé K = SPART, trouvez le message chiffré C du message en clair M = INFORMATIQUES

3. Masque jetable (one time Pad)

Pour éviter les attaques statistiques, cette technique consiste à utiliser une clé, appelé masque, constituée de K caractères tries au hasard et de même longueur que le message en clair. Le chiffrement est effectué par :

Ci = f(mi) = (mi + Ki) mod n

Où

mi : le caractère à chiffrer

Ki : caractère de la clé (masque) généré aléatoirement pour chiffrer mi

n : dimension de l'ensemble de l'alphabets A.

Cette fonction est identique au système de vigénere avec une génération aléatoire des caractères de la clé dont la taille est celle du message en clair.

Exemple :

n = 26

K = TBFRGFARFMIKL

M = MASQUEJETABLE

Calculer le message chiffré C = Ek(M)?

Cryptographie classique

- # Transposition

Le chiffrement par transposition est basé sur les permutations des caractères du message en clair. De ce fait, les caractères sont toujours mais dans un Autre ordre pour augmenter les diffusions dans le message en clair. Ce procédé est fondé essentiellement sur des matrices d'ordres nxp.

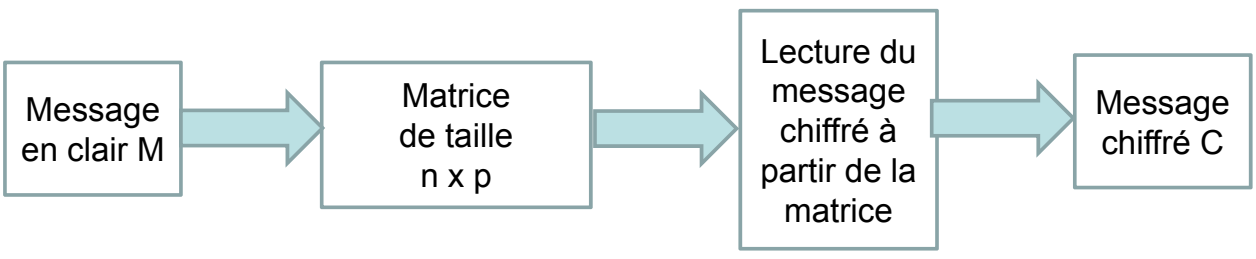


Fig 6. Principe de la transposition

1. Transposition simple par colonnes

Dans la matrice, le message en clair est écrit horizontalement et le message chiffré est obtenu en lisant la matrice verticalement. De ce fait, l'ordre de la matrice représente la clé de chiffrement K.

Exemple :
K = 4 x 4
M = INFORMATIQUES
Donc nous avons une matrice de 4 lignes et 4 colonnes.
X est pour compléter la matrice.

1	2	3	4
I	N	F	O
R	M	A	T
I	Q	U	E
S	X	X	X

D'où le message chiffré est

C = IRISNMQXFAUXOTEX

Le déchiffrement s'effectue en écrivant le message C en colonnes. Le message en clair est obtenu en lisant horizontalement la matrice. La transposition par colonne peut être compliquée si on lit les colonnes dans autre ordre. La clé K sera représentée par l'ordre de la matrice et le séquencement de la lecture des colonnes.
Exemple :
K = 4 x 4, lecture des colonnes 2-4-1-3
M = INFORMATIQUES

1	2	3	4
I	N	F	O
R	M	A	T
I	Q	U	E
S	X	X	X

D'où le message chiffré est

C = NMQXOTEXIRISFAUX

1. Transposition complexe par colonnes

Dans la transposition complexe par colonnes, le nombre de colonnes (p) de la matrice est fixé par le nombre de caractères d'une clé K (tous les caractères sont différents les uns aux autres), le nombre de ligne (n) dépendra de la longueur du message en clair. Le classement dans l'ordre alphabétique des caractères de la clé permet de fixer le séquencement de la lecture des colonnes de la matrice.

Exemple :
K = GATS, p = 4
M = INFORMATIQUES

G	A	T	S
I	N	F	O
R	M	A	T
I	Q	U	E
S	X	X	X

Classement par ordre alphabétique des caractères de K

A	G	S	T
N	I	O	F
M	R	T	A
Q	I	E	U
X	S	X	X

Message chiffré est
C = NMQX IRIS OTEX FAUX

Cryptographie moderne

La cryptographie symétrique classique traitaient des cryptosystèmes basé sur des caractères. Les différents algorithmes remplaçaient ou transposaient les caractères. Les meilleurs systèmes combinaient les deux opérations plusieurs fois.
Actuellement la cryptographie symétrique moderne utilise la même philosophie. La différence est que les algorithmes associés manipulent des bits au lieu des caractères. Donc, il y a un passage d'un alphabet de 26 caractères à un alphabet de 2 caractères (base 2).
La plus part des algorithmes combinent également des substitutions avec des transpositions en plus des deux fonctions logiques : « ou-exclusif» et décalage logique.

Cryptographie moderne

1. Ou Exclusif simple

La fonction logique « ou-exclusif » ou « XOR » est une opération classique sur les bits, est notée '^' en langage C et (+) en mathématique binaire.
La table de vérité ou a et b sont codés sur un bit est

Entrée a	Entrée b	Sortie S
0	0	0
0	1	1
1	0	1
1	1	0

Table de vérité

La fonction canonique est

$$S = \bar{a}b + \bar{b}a = a \bigoplus b$$

Considérons un bloc M du message en clair et une clé K de chiffrement. Ils sont codés à l'intérieur d'une machine informatique en code ASCII, et le chiffrement se réalise en effectuant un XOR entre les différents bits de M et C :

$$C = M \bigoplus K$$

$$ci = mi \bigoplus ki$$

ci : l'état logique du ième bit du message chiffré,
mi : l'état logique du ième bit du message en clair,
Ki : l'état logique du ième bit de la clé K.

Le déchiffrement bit à bit est

$$mi = ci \bigoplus ki = mi \bigoplus ki \bigoplus ki = mi \bigoplus 0 = mi$$

Le message déchiffré est

$$M = C \bigoplus K$$

2. Masque jetable (one time Pad)

L'idée du masque jetable peut être facilement étendue au chiffrement de données binaires. Cette méthode représente une généralisation de XOR simple. On utilise donc un masque composé de bits au lieu des caractères et au lieu de l'addition, on utilise le XOR. Pour déchiffrer, on applique au message chiffré le XOR avec le même masque jetable. On rappel que la taille du masque jetable est égale à celle du message en clair, et que le masque doit être généré aléatoirement et utilisé une seul fois.
L'inconvénient majeur du masque jetable est la synchronisation entre l'expéditeur et le destinateur sur le canal de transmission. Si le destinataire est décalé de quelques bits (bits perdu durant la transmission) et donc le message déchiffré n'aura aucun sens. Malgré cela, les masques jetables ont encore leur utilité essentiellement pour des canaux de communications ultra-secrets et à faible débit. Le chiffrement est effectué par l'opération : $C = M \bigoplus K$
M est le message en clair; K est la clé de chiffrement et C est le message chiffré.

Modes cryptographiques

Il existe deux modes de base de fonctionnement des algorithmes symétriques ou à clé secrète : les algorithmes de chiffrement par blocs et les algorithmes de chiffrement en continu. Les algorithmes de chiffrement par blocs manipulent des blocs de texte en clair et de texte chiffré. Les algorithmes de chiffrement en continu manipulent des flux de texte en clair et de texte chiffré bit par bit ou octet par octet.

- Modes de chiffrement par blocs
  - Carnet de codage électronique (ECB : *Electronic Code Blook*)

    Chaque bloc de n bits du message en clair est chiffré indépendamment des autres blocs du message en clair. Le même bloc sera donc toujours chiffré en un même bloc de même taille en utilisant la même clé.

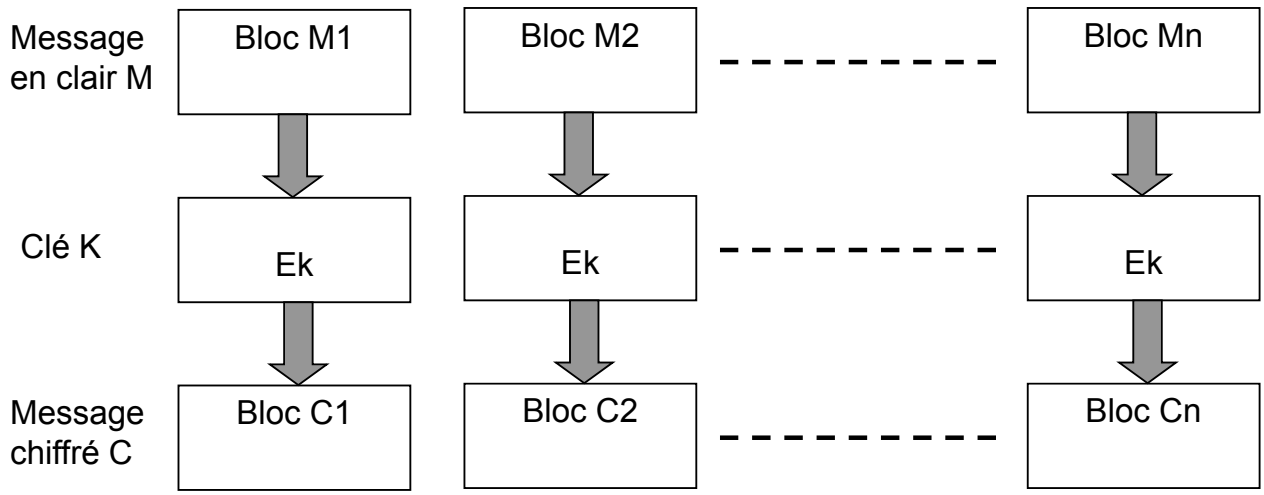


Fig 7 : **Mode ECB**

Le chiffrement par ECB est : Ci = Ek(Mi) et le déchiffrement est Mi = Dk(Ci).

Modes cryptographiques

- Modes de chiffrement par blocs
  - Chaînage de blocs (CBC : Cipher *Block Chaining*)

  Dans le mode CBC, le chiffrement s'effectue par un mécanisme de chaînage. Ainsi, le résultat du chiffrement du bloc précédent est combiné, par un ou exclusif, avec le bloc courant du message en clair. De ce fait, le résultat de chiffrement de ce bloc dépendra non seulement du bloc en clair qui l'a engendré mais aussi du bloc du message en clair qui le précède. Le chiffrement du dernier bloc dépendra du résultat de chiffrement de tous les blocs qui le précédent.

  Un message en clair M est décomposé en blocs de même taille n que celle de la clé K en nombre de bits (n=64 bits (8caractères), n=128 (16 caractères), …). Aussi le bloc chiffré Ci est de même taille n.
  Pour chiffré le premier bloc M1 un vecteur d'initialisation VI est utilisé comme bloc précédent et sa valeur peut être aléatoire.

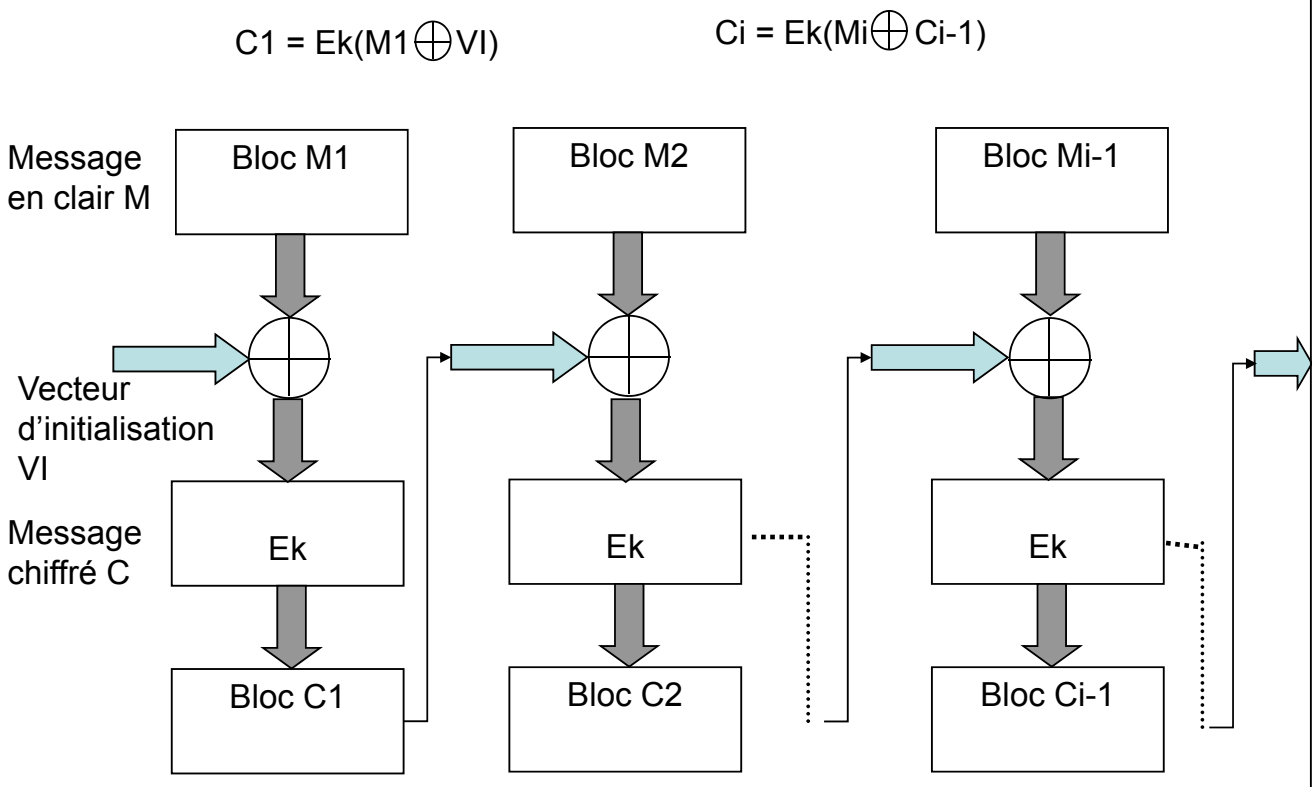


Fig 8 : **Mode CBC : Chiffrement**

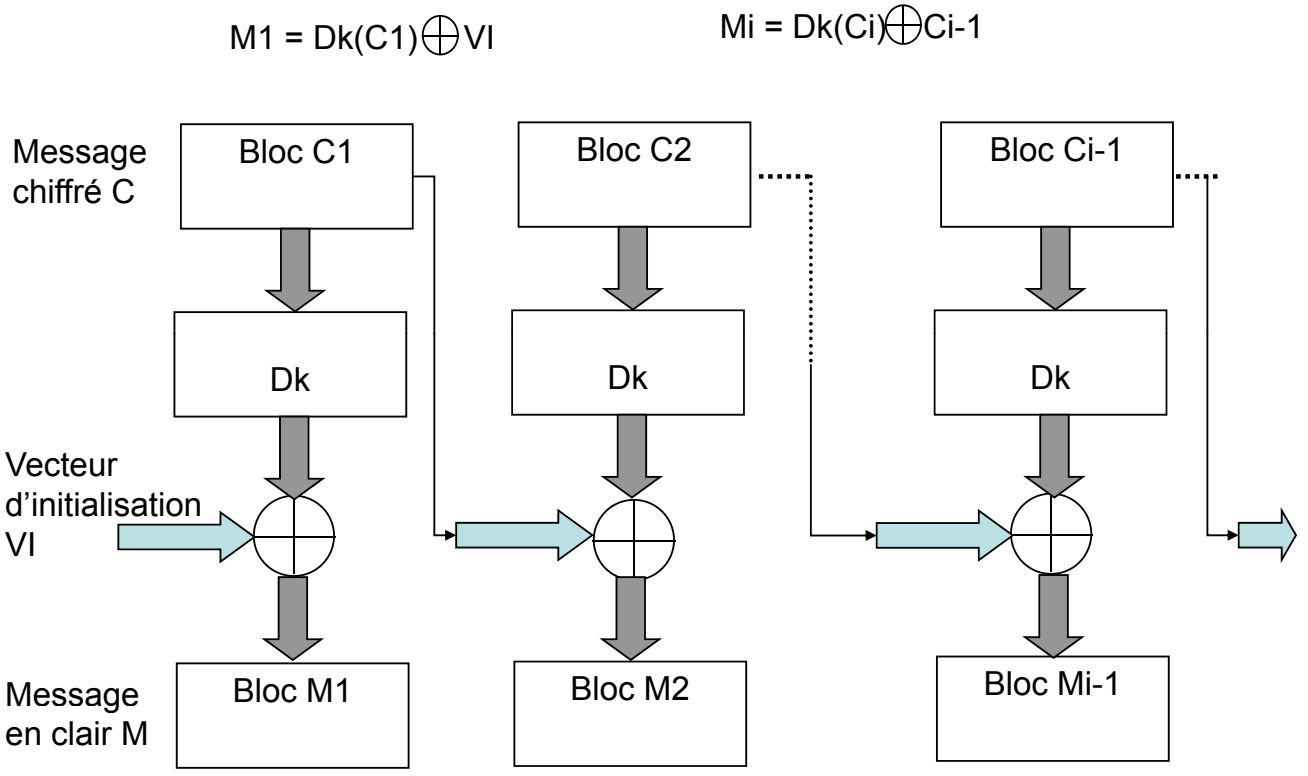


Fig 9 : **Mode CBC : Déchiffrement**

Modes cryptographiques

- Modes de chiffrement en continu
  - Chiffrement à rétroaction (CFB : *Cipher Feed Back*)

En mode CBC, les données peuvent être chiffrées par unités de m bits plus petite ou égale à la taille d'un bloc à chiffrer de n bits.
La figure suivante montre un exemple de chiffrement en mode CFBà 8 bits appliqué à un algorithme de chiffrement par bloc de 64 bits (n = 64 et m = 8). Un algorithme de chiffrement par bloc en mode CFB manipule une file d'attente de la taille d'un bloc d'entrée. Initialement, la file est initialisée par un bloc quelconque de même taille que le bloc à chiffrer : vecteur d'initialisation VI. Ensuite, la file est chiffrée par un algorithme de chiffrement par bloc Ek et les 8 bits les plus à gauche du résultat de chiffrement sont combinés par ou exclusif avec le premier caractère de 8 bits du message en clair pour former les 8 premiers bits du message chiffré.

Modes cryptographiques

Les 8 bits sont placés dans les 8 bits les plus à droite de la file d'attente et les autres bits sont décalés de 8 positions vers la gauche. Les 8 bits les plus à gauche sont ignorés. Les autres caractères sont chiffrés de la même manière jusqu'à la fin du bloc de 64 bits.

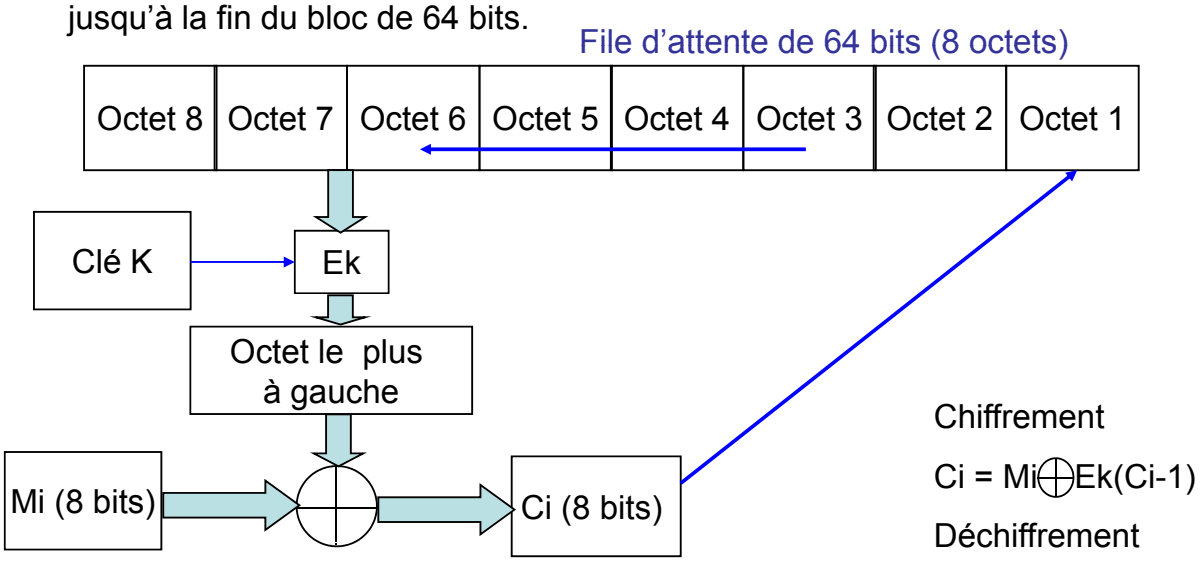


Fig 8 : **Mode CFB**

Chiffrement

$Ci = Mi \oplus Ek(Ci-1)$

Déchiffrement

$Mi = Ci \oplus Ek(Ci-1)$

Modes cryptographiques

– Mode de rétroaction de sortie (OFB : *Output-Feed Back*)

Ce mode est similaire au mode CFB, sauf que n bits chiffrés sont rangés dans les 8 positions les plus à droite de la file d'attente. Le résultat de chiffrement peut être transmis et ne contribue pas au chiffrement du caractère suivant.

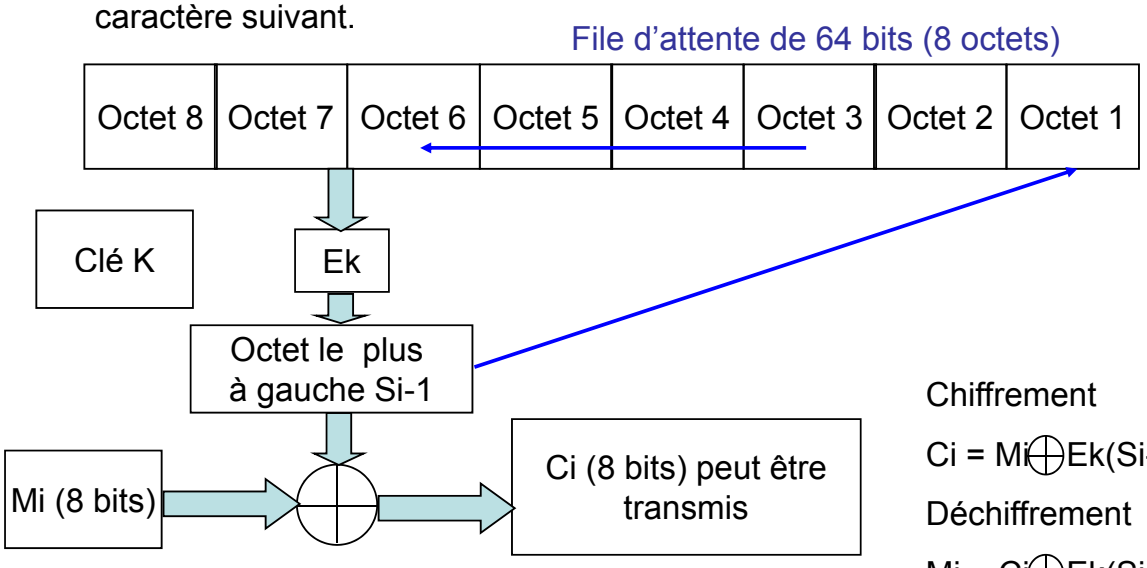


Fig 10 : **Mode CFB**

Chiffrement

$Ci = Mi \oplus Ek(Si-1)$

Déchiffrement

$Mi = Ci \oplus Ek(Si-1)$

Le standard des algorithmes de chiffrement de données : le DES (Data Encryption Standard)

Historique : Dès le début des années **1960** la technologie des circuits intégrés permet de travailler à des circuits combinatoires complexes permettant d'automatiser:

la méthode de **substitution** et la méthode de **transposition**. Un chercheur Des laboratoires d'IBM, Feistal Horst, propose l'idée d'un algorithme de chiffrement très fiable dont les composantes simples permettent le codage très facilement sur un circuit électronique. Le projet Est retenu et sera développé sous le nom de code LUCIFER.

En **1972**, à la recherche de l'algorithme le plus sûr possible, le NBS lance un appel d'offre à travers un cahier de charge. Vu les traits de ressemblance avec le projet LUCIFER, IBM apporte quelques modifications afin de satisfaire la norme proposée.

Le standard des algorithmes de chiffrement de données : le DES

1977 : L'algorithme de chiffrement conçu par IBM est retenu par le NBS (National Bureau of Standards) sous le nom de DES.
1978 : L'ANSI valide à son tour cet algorithme sous le nom du DEA (Data Encryption Algorithm).

Principe : L'algorithme DES est un cryptosystème de chiffrement symétrique par blocs (mode ECB). Le principe de base du DES est fondé essentiellement sur la transposition binaire ou permutation pour insérer le mécanisme de **diffusion** et sur la substitution binaire pour intégrer le mécanisme de **confusion**.

Dans un premier temps, le message en clair est découpé en plusieurs blocs de 64 bits (8 caractères). Chaque bloc est chiffré par une clé de 64 bits. Le bloc chiffré a la même taille que le bloc initial du message en clair de 64 bits.

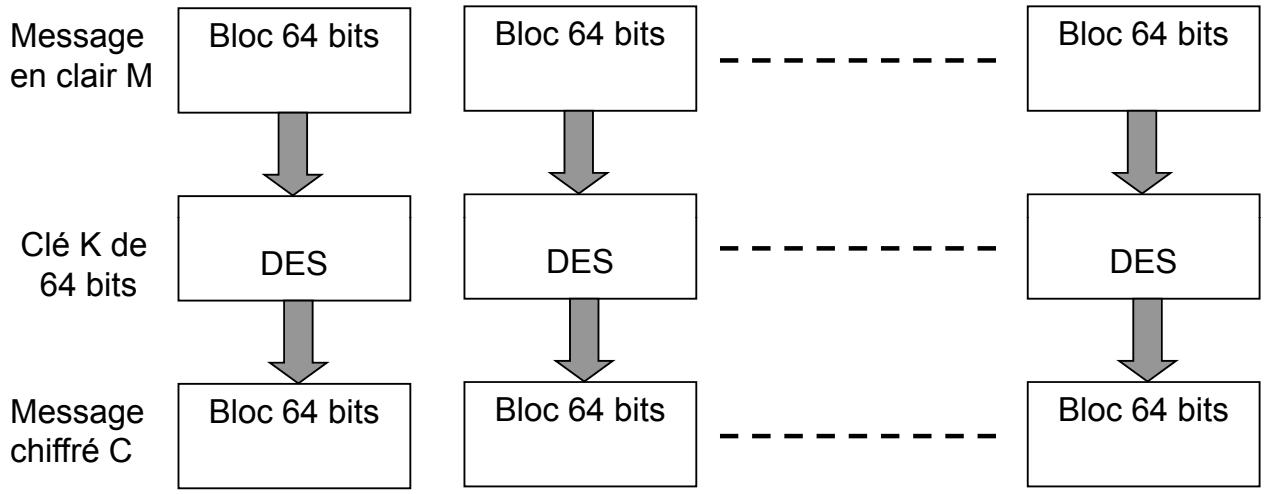


Figure 11 : **Principe du DES en mode ECB**

Le standard des algorithmes de chiffrement de données : le DES

Chiffrement : Pour chiffrer un bloc de 64 bits, la fonction principale du DES est fondue sur un processus itératif composé de 16 itérations. De ce fait, à partir de la clé de chiffrement on génère 16 sous clés Ki (i=1, 2, …,16). A chaque itération correspond une sous clé Ki et chaque itération est composée de 5 étapes comme le montre la figure 12.

Le message en clair de 64 bits constitue donc l'entrée de la première itération.

Bloc T

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	12	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Indices des bits d'un bloc T de 8 octets du message en clair M.

Le standard des algorithmes de chiffrement de données : le DES

Avant la première étape d'une itération i, ce bloc va subir une permutation initiale comme suit. Cette permutation consiste à déplacer les 4 colonnes paires vers les 4 premières lignes tout en inversant les indices des bits d'une colonne. Ensuite, les 4 colonnes impaires restants sont déplacés vers les 4 lignes en inversant également l'ordre des bits. Voir les tableaux dans la page suivante.

Le nouveau bloc généré par cette permutation est divisé en deux parties de 32 bits chacune. La partie droite notée R(i-1), constituée uniquement des bits pairs de la permutation. La partie gauche notée L(i-1) est constituée par les indices des bits impairs restants.

Le standard des algorithmes de chiffrement de données : le DES

Permutation initiale IP

IP:

	0	1	2	3	4	5	6	7
1	58	50	42	34	26	18	10	2
9	60	52	44	36	28	20	12	4
17	62	54	46	38	30	22	14	6
25	64	56	48	40	32	24	16	8
33	57	49	41	33	25	17	9	1
41	59	51	43	35	27	19	11	3
49	61	53	45	37	29	21	13	5
57	63	55	47	39	31	23	15	7

Indices des bits d'un bloc T permuté par IP.

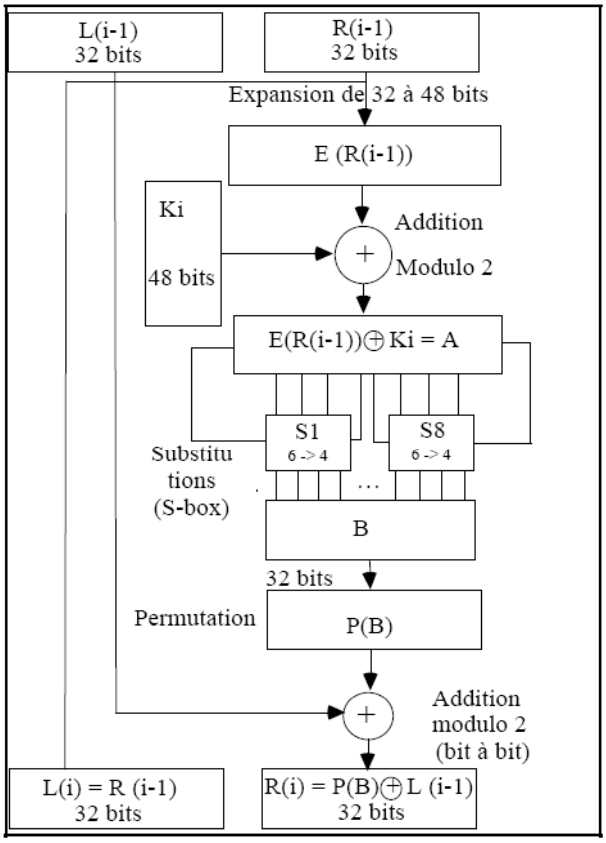


Figure 12 : Principe de fonctionnement d'une itération

Partie droite R(i-1)

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

Partie gauche L(i-1)

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

Après cette division, les 5 étapes de l'itération i sont:

Étape 1: Les 32 bits de R(i-1) entrent dans une table de permutation expansive noté E(Ri-1). Elle change l'ordre des bits et répète certains bits tout en réalisant une extension de 32 à 48 bits. Cette présentation représente de la manière suivante :

Le standard des algorithmes de chiffrement de données : le DES

La table correspondante E(Ri-1) de 48 bits est :

Bloc E(Ri-1)

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

Le standard des algorithmes de chiffrement de données : le DES

Étape 2: E(Ri-1) a une taille de 48 bits, elle sera transformée par un XOR avec la sous clé Ki générée pour l'itération i à partir de la clé K et elle doit avoir également une même taille de 48 bits.

Etape 3: Le résultat de l'étape 2 est de 48 bits, il sera divisé en 8 sous blocs de 6 bits chacun. Chaque sous bloc subira une substitution binaire compressive qui génère un bloc de 4 bits. Ainsi, cette substitution génèrera 8 blocs de 4 bits chacun et donc un bloc de 32 bits. Ceci permet de couper définitivement tout rapport entre le bloc du message en clair et le bloc chiffré correspondant. De ce fait, il y a 8 tables de substitutions. Voir la figure 13.

Etape 4: Les bits du bloc B vont être ensuite permutés selon la table P Suivante : Les bits changent de position, aucun bit n'est utilisé deux fois et aucun bit n'est ignoré.

Une table Sj (j=1,…,8) est constituée de 4 lignes, 16 colonnes. L'élément général de cette table est codé sur 4 bits.

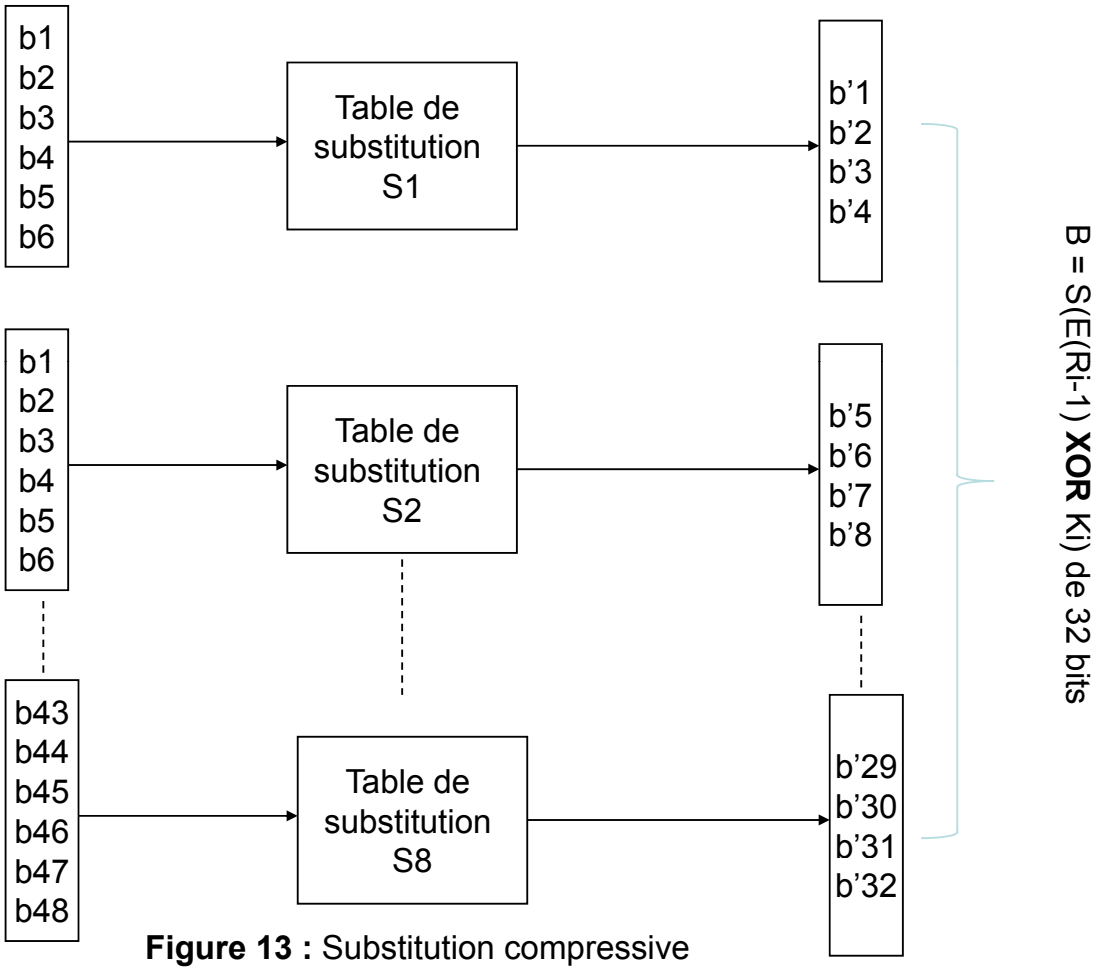


Figure 13 : Substitution compressive

Le standard des algorithmes de chiffrement de données : le DES

- S-Box1

	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

Mécanisme:

Substitution pour le bloc b1b2b3b4b5b6. Les deux bits faible et fort sélectionnent le numéro de la ligne b6b1 =X, les 4 bits intermédiaires le numéro de la colonne b2b3b4b5 =Y. La position (X,Y) permet de substituer le bloc b6b5b4b3b2b1 par un autre bloc de 4 bits comme intersection de la ligne X et colonne Y.

Exemple : G = 101110. Les premiers et derniers bits donnent 10, c'est-ŕ-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 ŕ la ligne n°2=X, dans la colonne n°7=Y. Il s'agit de la valeur 11=(X,Y), soit en binaire 1011.

Le standard des algorithmes de chiffrement de données : le DES

Les tables de substitutions :

S-Box1:

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

S-Box2:

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

S-Box3:

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

Le standard des algorithmes de chiffrement de données : le DES

S-Box4:

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

S-Box5:

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

S-Box6:

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

Le standard des algorithmes de chiffrement de données : le DES

```
S-Box7:
    4  11 2  14 15 0 8  13 3  12 9 7  5  10 6 1
    13 0  11 7  4  9 1  10 14 3  5 12 2  15 8 6
    1  4  11 13 12 3 7  14 10 15 6 8  0  5  9 2
    6  11 13 8  1  4 10 7  9  5  0 15 14 2  3 12
S-Box8:
    13 2  8  4 6  15 11 1  10 9  3  14 5  0  12 7
    1  15 13 8 10 3  7  4  12 5  6  11 0  14 9  2
    7  11 4  1 9  12 14 2  0  6  10 13 15 3  5  8
    2  1  14 7 4  10 8  13 15 12 9  0  3  5  6  11
```

Le standard des algorithmes de chiffrement de données : le DES

Etape 5: Dans cette étape l'opération suivante est effectuée :
P(B) **XOR** L(i-1) et elle va constituer la nouvelle partie droite de l'itération suivante Ri. Et la nouvelle partie gauche de l'itération suivante sera la partie droite avant le chiffrement du bloc Li = Ri-1.
Ainsi, s'achève une itération Ri = P(B) XOR Li et Li = Ri-1 constituent les parties droite et gauche de l'itération suivante. A la fin de la 16ème itération on obtient le bloc L16, R16 de 64 bits qui subira une permutation finale (inverse de la permutation initiale et constitue le bloc chiffré T du message en clair M.).

Le standard des algorithmes de chiffrement de données : le DES

Le **bloc chiffré** est

	0	1	2	3	4	5	6	7
1	40	8	48	16	56	24	64	32
9	39	7	47	15	55	23	63	31
17	38	6	46	14	54	22	62	30
25	37	5	45	13	53	21	61	29
33	36	4	44	12	52	20	60	28
41	35	3	43	11	51	19	59	27
49	34	2	42	10	50	18	58	26
57	33	1	41	9	49	17	57	25

La figure 14 suivante présente le synoptique générale pour chiffrer un bloc T du message M.

Déchiffrement : Procéder exactement de la même manière en appliquant les clés en sens inverse (de k16 à k1).

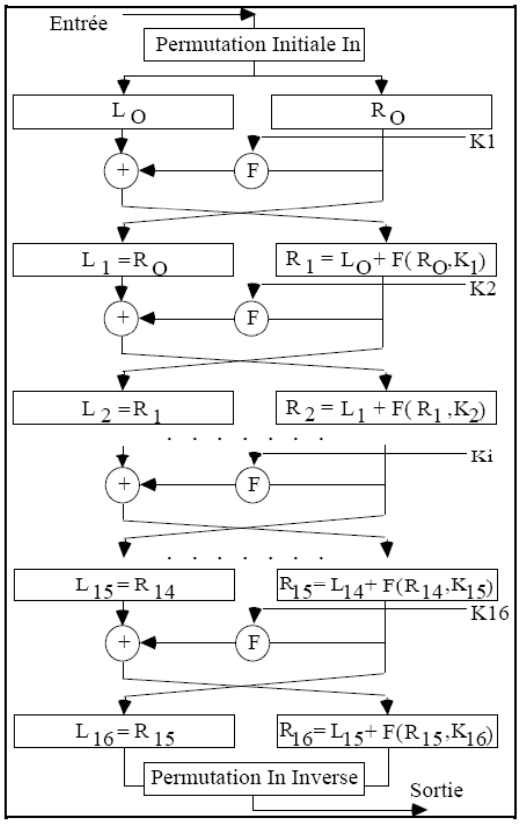


Figure 14 : Architecture générale du DES

Domaine d'utilisation:

Sur l'implémentation matérielle, le DES est actuellement capable de chiffrer et de déchiffrer jusqu'à 1 Gegabits /seconde. Il est donc éligible pour chiffrer et déchiffrer sans surcoût d'échange sur un réseau ou sur un bus. Ila été utilisé dans tous les domaines désirant une sécurité acceptable : Banque, industrie et également il a été envisagé pour les communications téléphoniques et pour les signaux vidéo de haute définition.

Niveau de sécurité du DES :

La qualité d'un cryptosystème est évaluée par le temps nécessaire et le coût du matériel utilisé pour trouver le message en clair à partir d'un message chiffré. Depuis sa naissance, le DES a été soumis à toutes les techniques imaginables de cryptoanalyse (linéaire, différentielle, …). Régulièrement, à l'occasion des conférences internationales sur le chiffrement (CRYPTO'xx, EUROCRYPT, AUSTCRYPT …), des experts ont cherché les faiblesses de ce code qui, rappelons le, est actuellement le plus répandu sur le marché. Les résultats de ces recherches ont mis à jour quelques faiblesses à savoir : la taille de la clé, le nombre d'itérations et le schéma de conception des tables-S.

Taille de la clé :

Les experts ont estimé que la taille de la clé utilisée est très faible. Leurs arguments tournaient principalement autour de la possibilité d'une attaque exhaustive. Le DES code des messages en clair grâce à une clé de taille 56 bits. Donc, le nombre des clés possibles est de $2^{56}=7,2*10^{16}$ clés. En imaginant un ordinateur capable de tester la validité d'une clé par microseconde (attaque directe), et en supposant que la bonne clé est la dernière testée, le temps de calcul approcherai 2258 années. Mais ces suppositions sont toutes prises par excès. Certains mathématiciens estiment que pour que le code soit inviolable la clé doit avoir la taille de 300 bits. Au début IBM a proposé une clé de 128 bits mais la NSA l'a réduit à 64 bits. La raison d'une telle réduction n'a pas été rendue publique.

.

Tables-S (S-Box) :

L'existence de ces tables-S permettent d'imposer au message chiffré un caractère non linéaire des bits entre eux. Classifiés par la NSA, les critères de réalisation des fonctions S(i) n'ont jamais été rendus publics. Aussi est-il difficile de se prononcer sur le caractère fortement non linéaire de leur comportement. De plus, à travers plusieurs articles écrits dans les parutions régulières des conférences, deux mathématiciens (Diffie & Hellman) ont mis en évidence la structure quasi-linéaire de ces fameuses fonctions. Cette considération pourrait réduire considérablement le temps de cryptoanalyse.

Nombre d'itération :

La sécurité du DES avec 16 itérations est grande et résiste à l'heure actuelle à toutes les attaques effectuées avec des moyens financiers et temporels raisonnables (i.e. moins de 10 millions de dollars et moins d'un mois). Plusieurs versions du DES avec 3, 4, 6,… itérations ont été touts cassées.

Modes opératoires : Le DES peut être utilisé avec les 4 modes cryptographiques : ECB, CBC, OFB et CFB. Le mode le plus utilisé, à cause de simplicité est ECB et il ne protège pas contre la présence des blocs redondants. Le mode CBC est occasionnellement utilisé.

Conclusion:

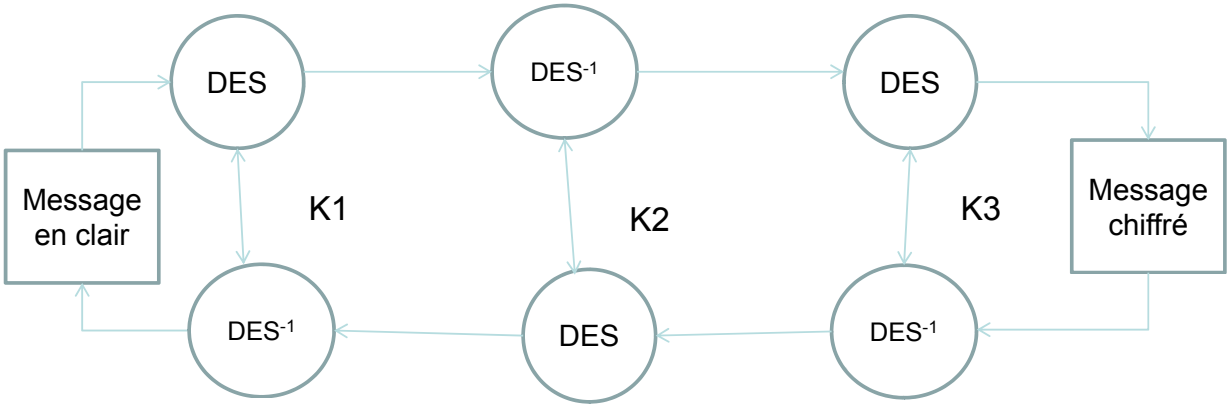
Ancien standard ayant tenu plus de 20 ans.
Excellentes performances en vitesse de chiffrement : 1 G bit / s en matériel et 1 M bits / s en logiciel.
Niveau de sécurité très raisonnable pour les applications civiles.
Il est probablement peu sûr pour un attaquant de gros moyens mais très suffisant pour les applications habituelles.

Variantes du DES :

Un grand nombre d'algorithmes ont vu le jour ces dernières années pour renforcer la sécurité et d'essayer de surmonter les inconvénients du DES (taille des clés, tables-S et le nombre d'itération).

Triple DES

Les mécanismes de chiffrement et déchiffrement sont réalisés par trois utilisations successives du DES en impliquant trois clés différentes : K1, K2 et K3 comme suit:



Le standard des algorithmes de chiffrement de données : AES (Advenced Encryption Standard)

Historique :

Le 2 janvier 1997, le NIST a lancé un processus de remplacement de DES : Advanced Encryption Standard, (AES). Un appel d'offre a été lancé le 12 Septembre 1997. Il y était requis qu'AES possède une longueur de bloc égale à 128 bits et qu'il supporte des longueurs de clef égales à 128, 192 et 256 bits. De plus, Il était nécessaire qu'AES soit libre de droits dans le monde entier.

Les soumissions devaient être rendues le 15 juin 1998. Des vingt-et-un systèmes cryptographiques soumis, quinze remplissaient tous les critères nécessaires et ont été acceptés en tant que candidats AES. Le NIST a présenté les quinze candidats lors de la *First AES Candidate Conference le 20/08/1998*. Une *second AES Candidate Conference s'est tenue en mars 1999. En août 1999, cinq candidats ont été choisis par le NIST comme finalistes : MARS, RC6, Rijndael, Serpent et Twofish. En avril 2000, une Third AES Candidate Conference a eu lieu. Le 2 octobre 2000, Rijndael a été choisis comme standard avancé de chiffrement.*

Le standard des algorithmes de chiffrement de données : AES (Advenced Encryption Standard)

Critères principaux d'évaluation d'AES:

1. Sécurité :
   La sécurité des algorithmes proposés était absolument essentielle et n'importe quel algorithme présentant des faiblesses de sécurité aurait été éliminé.
2. Coût
   Le coût se rapporte à l'efficacité en termes de calculs (vitesse et besoin en mémoire) sur divers types de plates-formes, que ce soit en logiciel, en matériel ou sur une carte à puce.
3. Caractéristiques de l'algorithme et de son implémentation.
   Elles incluent la flexibilité et la simplicité de l'algorithme.

Le standard des algorithmes de chiffrement de données : AES (Advenced Encryption Standard)

Description d'AES:

AES est un algorithme de chiffrement symétrique itéré; le nombre d'étages ou de ronde, qu'on le note Ne, dépend de la longueur de la clef. Ne = 10 si la clé a une longueur de 128 bits; Ne=12 dans le cas d'une clé de longueur 192 bits et Ne =14 si la clé comporte 256 bits.

La figure 15 montre succinctement le déroulement du chiffrement. Chaque ronde/étage est composé comme suit :

1. BYTE_SUB (Byte Substitution) est une fonction non-linéaire opérant indépendamment sur chaque bloc à partir d'un table dite de substitution.
2. SHIFT_ROW est une fonction opérant des décalages (typiquement elle prend l'entrée en 4 morceaux de 4 octets et opère des décalages vers la gauche de 0, 1, 2 et 3 octets pour les morceaux 1, 2, 3 et 4 respectivement).
3. MIX_COL est une fonction qui transforme chaque octet d'entrée en une combinaison linéaire d'octets d'entrée et qui peut être exprimée mathématiquement par un produit matriciel sur le corps de Galois (2^8).
4. Le + entouré d'un cercle désigne l'opération de OU exclusif (XOR). Ki est la ième sous-clé calculée par un algorithme à partir de la clé principale K.

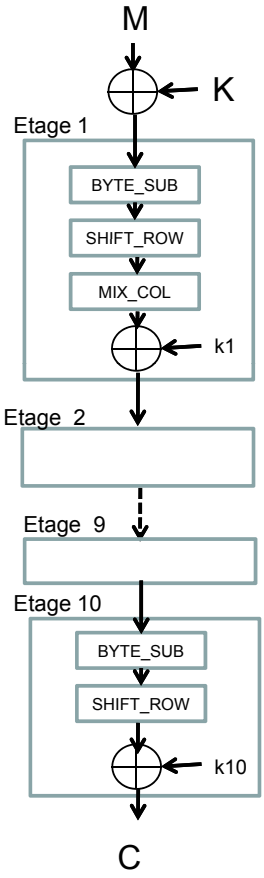


Figure 15 : Chiffrement selon l'algorithme AES.

Chapitre 3 : Aspects techniques de la cryptographie asymétrique

Le concept de la cryptographie asymétrique a été introduit pour la première fois par Diffie Hellman en 1976.

Utilise l'exponentielle modulaire, le problème d'échanges des clés des algorithmes symétriques.

L'idée neuve dans le domaine était d'utiliser deux clés différentes pour chiffrer et déchiffrer un message. Une des clés sera publiée et l'autre doit être maintenu secrète.

Cryptosystèmes asymétriques ou à clé publique.

A la même époque une nouvelle approche à clé publique a été proposée par Rivest, Shamir et Aldeman appelée RSA.

Les algorithmes RSA, ElGamal et Rabin seront à la fois pour le chiffrement et les signatures numériques.

Utilise également l'exponentielle modulaire.

Chiffrement et déchiffrement, l'échange des clés symétriques et l'authentification des messages.

Les algorithmes à clé publique sont conçus pour résister aux attaques à texte en clair choisi; leur sécurité dépend à la fois de la difficulté de déduire la clé privée à partir de la clé publique et la difficulté de déduire le texte en clair à partir du texte chiffré.

Inconvénient majeur de ces algorithmes

La vitesse de chiffrement et de déchiffrement sont nettement plus faible que pour les algorithmes de la cryptographie symétrique. Habituellement sont trop lent pour le chiffrement de données en masse.

Avantage

Chiffrement et le déchiffrement

Authentification des messages (sorte de signature numérique).

Chiffrer et échanger des clés de session des algorithmes symétrique dans un réseau non sécurisée.

I- RSA

Le principe de base de cet algorithme est fondé sur des fonctions appartenant à la famille des fonctions trappes ou pièges. Elles sont appelées des fonction à sens unique à brèche secrète : facile à calculer dans un sens et difficile voir impossible à calculer dans le sens inverse sans la connaissance de la brèche secrète.

I- 1. Principe

L'utilisation du RSA consiste à générer tout d'abord des clés publiques et privées et ensuite effectuer le chiffrement, le déchiffrement des messages ou l'authentification (signature numérique des messages).

Etape 1 : Génération des clés

Pour générer les clés publique et privée d'un utilisateur, cette étape est basée sur une multiplication de deux grands nombres premiers faciles à calculer et sur la difficulté de factoriser le résultat de cette multiplication. Sous cette condition, cette étape consiste à calculer trois nombres **e**, **d** et **n** permettant de définir :

Une clé publique Kp = (e , n)
Une clé privée Kpr = (d , n)

Etape2 : Chiffrement et déchiffrement

Le chiffrement/ déchiffrement des messages est basée sur l'exponentielle modulaire de base **a** et d'exposant **b** modulo **n** :

$$\mathbf{a^b \bmod n}$$

Elle est facile à calculer dans un sens et elle est difficile à inverser (le logarithme discret)

Chiffrement

$$C = E_{kp}(M) = M^e \bmod n \qquad \text{avec } K_p = (e, n)$$

Déchiffrement

$$M = D\,k_{pr}(C) = C^d \bmod n \qquad \text{avec } K_{pr} = (d, n)$$

Pour que les deux mécanismes soient réversibles il faut que

$$(M^e \bmod n)^d \bmod n = (M^{ed} \bmod n) \bmod n = M$$

Ainsi, les clés publiques et privées sont liées par l'équation suivante :

$ed \bmod n = 1 \rightarrow$ calcul de **l'inversion modulaire**

Si on fixe **e** on calcul **d** ou inversement

II- Outils mathématiques

La cryptographie asymétrique utilise essentiellement la théorie des nombres et plus particulièrement les outils suivants :

1. Génération de très grands nombres premiers,
2. Arithmétique modulaire de base n,
3. Exponentielle modulaire de base n,
4. Inversion modulaire de base n,
5. Fonction d'Euler et théorème d'Euler.

1. Génération de très grands nombres premiers

Un nombre premier est un entier dont les seuls facteurs soit 1 et lui-même. Deux nombres entiers sont premiers entre eux si leurs PGCD = 1. L'algorithme le plus utilisé pour tester la primalité d'un entier de grande valeur est celui de **Rabin-Miller**.

2. Arithmétique modulaire de base n

$b = a \bmod n$ $\quad a \equiv b \pmod n$ $\quad$ a est congrue à b modulo n

$a = kn + b$

a, b, k et n sont des entiers. Si a et b>0 et b<n alors b est le reste de la division de a par n appelé résidu de a modulo n et représente une réduction modulaire. L'ensemble des résidus est **{ 0 , 1 , …. , n-1 }.**

L'arithmétique modulaire a les mêmes propriétés que l'arithmétique classique : commutativité, associativité et distributivité.

3. Exponentielle modulaire de base n

a^x mod n

Il existe plusieurs techniques qui permettent de calculer l'exponentielle modulaire sans engendrer des résultats démesurés. Le meilleur algorithme est celui de **Montgomery**.

4. Inversion modulaire de base n

Le calcul de l'inverse modulaire est un problème difficile à résoudre par rapport à l'inversion classique **(ax = 1, x=1/a → a^{-1})**

ax mod n =1 avec a et x Є [0 , n-1]

Cette équation a une solution unique a^{-1} si :

a et n sont premiers entre eux PGCD(a , n)=1, sinon pas de solution.

Si n est premier, chaque entier de l'ensemble des résidus [0 ,n-1] est premier par rapport à n et chacun a un inverse dans l'ensemble des résidus calculé par :

$x = a^{-1}$ mod n

appelé l'inverse multiplicatif modulo n de a.
L'algorithme d' Euclide étendu ou le théorème d'Euler permettent de calculer x.

5. Fonction d'Euler

Le théorème d'Euler permet de calculer l'inverse modulaire d'un entier modulo n en utilisant l'ensemble restreint des résidus modulo noté Φ(n).

Si n est premier alors Φ(n) est le cardinal de l'ensemble des résidus
{0, 1, 2, …, n-1}-{0} : ensemble restreint des résidus
Φ(n) = n -1 représente le cardinal de l'ensemble d'entiers premier avec n.
Si n = p x q avec p et q nombre premiers alors

$$\Phi(n) = \Phi(p)\,\Phi(q) = (p - 1)(q - 1)$$

La fonction Φ(n) est appelée fonction **totient d'Euler**.

Théorème d'Euler

Si PGCD (a ,n) = 1 alors **$a^{\Phi(n)}$ mod n =1**

La résolution de ax mod n =1 est

1. a $a^{\Phi(n)-1}$ mod n =1
2. a^{-1} = $a^{\Phi(n)-1}$ mod n
3. x = a^{-1} mod n
4. x = $a^{\Phi(n)-1}$ mod n

Par la généralisation du théorème d'Euler, l'inversion de ax mod n = b est calculée par :

x = b $a^{\Phi(n)-1}$ mod n

Factorisation

La factorisation d'un nombre consiste à chercher tous les facteurs premiers qui le compose (252 = 41* 61*101).

III- Génération des clés publique et privée d'un correspondant A

1- Calcul des nombres nA, eA et dA

1. Trouver deux entiers premiers pA et qA dont les valeurs sont extrêmement grandes, valeurs usuelles sont codées sur 2048 et 4096 bits. Puis calculer

   nA =pA qA.

1. Calculer la fonction d'Euler

   ΦA (nA) = (pA -1)*(qA-1)

1. Calculer les nombres eA et dA.

Condition pour avoir le chiffrement par (eA , nA) et le déchiffrement par (dA , nA)

$(M^{eA} \bmod nA)^{dA} \bmod nA = M^{eAdA} \bmod nA$

Si eAdA mod nA = 1 → M mod nA = M

Par conséquent, M mod nA = M impose la décomposition du message à chiffrer ou à déchiffrer sous la forme de blocs **dont la taille est inférieur à celle de nA.**

2- Calcul du nombre eA

Pour que **eA dA mod Φ (nA) = 1** possède une solution unique il faut que le **PGCD (eA , Φ (nA)) = 1**

De ce fait, eA doit appartenir à l'ensemble restreint des résidus Φ (nA) de nA.

Choisir eA dans [2, nA - 1].
(eA , nA) représente la clé publique K^A_p à publier.

3- Calcul du nombre dA

L'inversion dA de eA peut être calculé par le théorème d'Euler :

$e_A^{\Phi(nA)} \bmod nA = 1$

$dA = e_A^{-1} = e_A^{\Phi(nA)-1} \bmod nA$

$dA = e_A^{-1} = e_A^{\Phi(nA)-1} \bmod nA.$

Remarque :
e^{-1} peut être également calculé par l'algorithme d'Eculide étendu
$dA = e_A^{-1} \bmod \Phi(nA)$

dA Є [2 ,nA – 1]
(dA , nA) représente la clé privée K^A_{pr} à garder secrète

Important **: Les nombres pA, qA et Φ (nA) doivent être détruits.**

Ainsi, le chiffrement d'un message M est effectué par $C=M^{eA} \bmod nA$ et le déchiffrement du message C par $M = C^{dA} \bmod nA$.

III. Exemples

Exemple 1
Chiffrement / déchiffrement du message M=8 reçu par A

Génération des clés publique et privée de A
choix de deux nombres premiers pA =3 et qA= 5 → nA = pAqA =15
Φ (nA) = (pA-1)(qA-1) = 8
Calcul de eA et dA.

Choix de eA dans [2 , nA-1] premier avec nA par exemple eA =11.

$$K^A_p = (11 , 15)$$

$$dA = 11^{\Phi(nA)-1} \bmod 15 = 3$$

$$K^Apr = (3 , 15)$$

Chiffrement de M

Un correspondant B chiffre M = 8 avec la clé publique de A

$$C = 8^{11} \bmod 15 = 2$$

Déchiffrement de C

A reçoit C et le déchiffre avec sa clé privée

$$M = 2^3 \bmod 15 = 8$$

Exemple 2

Chiffrement / déchiffrement du message M « couleurs » reçu par A

Génération des clés publique et privée de A

1. Choix de deux nombres premiers pA = 53 et qA = 17 → nA = pAqA = 901
2. Φ (nA) = (pA - 1) (qA - 1) = 832
3. Calcul de eA et dA

4. Choix de eA dans [2 , nA-1] premier avec nA par exemple eA =83

$$K^Ap = (83 , 901)$$

Ainsi, $dA = 83^{\Phi(nA)-1} \bmod 901 = 411$

$$K^Apr = (411 , 901)$$

Chiffrement de M

B chiffre le message M avec la clé publique de A : **(83, 901)**

Le message M = m1 m2 m3 m4 m5 m6 m7 m8 = c o u l e u r s est codé en ASCII par 099 111 108 101 11 7 114 115.

Le message M est découpé sous la forme de blocs dont le nombre de chiffre est inférieur ou égale à celui de n → **3 chiffres**

Blocs mi	Chiffrement	Bloc ci
m1	**099^{83} mod 901**	**789**
m2	111^{83} mod 901	797
m3	117^{83} mod 901	196
m4	108^{83} mod 901	233
m5	101^{83} mod 901	475
m6	117^{83} mod 901	196
m7	114^{83} mod 901	198
m8	115^{83} mod 901	378

C = c1c2c3c4c5c6c7c8 = 789 797 196 233 475 196 198 378.

Déchiffrement de C

A reçoit C et le déchiffre avec sa clé privée :

Blocs ci	Chiffrement	Bloc mi
c1	**789^{411} mod 901**	**099**
c2	797^{411} mod 901	111
c3	196^{411} mod 901	117
c4	233^{411} mod 901	108
c5	475^{411} mod 901	101
c6	196^{411} mod 901	117
c7	198^{411} mod 901	114
c8	378^{411} mod 901	115

M = 099 111 117 108 101 117 114 115 = c o u l e u r s
On peut remarquer que l'utilisation de l'élévation à la puissance puis l'utilisation du modulo permet de changer la base des registres choix et d'introduire une discontinuité.

Exemple
m7 = 114 et m8 = 115 valeurs peu différents
c7 = 198 et c8 =378 valeurs très différents

Authentification des messages

L'authentification d'un message peut être considéré comme une signature numérique. En effet, A signe son message M en utilisant sa clé privée : K^A_{pr}

$$\mathbf{C = D_{K^A_{pr}}(M) = M^{dA} \bmod nA}$$

B déchiffre C en utilisant la clé publique de A : K^A_p

$$\mathbf{M = E\ KAp\ (C) = MeAdA \bmod nA}$$

L'inconvénient majeur est que M peut être déchiffrer par n'importe qui puisque eA et nA sont publiques.

Chiffrement/déchiffrement et authentification des messages entre A et B

Scénario 1 : A chiffre le message M avec sa clé privée puis chiffre le résultat avec la clé publique de B

$$C = Ek^Bp(Dk^Apr(M)) = (M^{dAeB} \bmod nA) \bmod nB$$

B déchiffre le message C en utilisant sa clé privée, ensuite avec la clé publique de A :

$$M = Ek^Ap(Dk^Bpr(C)) = (C^{dBeA} \bmod nB) \bmod nA$$

Scénario 2 : A chiffre le message M avec la clé la clé publique de B et le résultat avec sa clé privée

$$C = Dk^Apr(Ek^Bp(M)) = (M^{dAeB} \bmod nB) \bmod nA$$

B déchiffre le message C en utilisant la clé publique de A, ensuite avec sa clé privée :

$$M = Dk^Bpr(Dk^Ap(C)) = (C^{eAdB} \bmod nA) \bmod nB$$

Pour avoir un mécanisme réversible il faut que :

$nA < nB$ pour le scénario 1
$nB < nA$ pour le scénario 2.

Attaque de RSA :
Les nombres n et e sont publiques, de ce fait il y a une possibilité de chercher p et q puis Φ(n) en se basant sur la factorisation de n en deux nombres premiers. Si on arrive l'algorithme sera cassé car le calcul de e et de d devient très facile.

Sécurité de RSA :
La sécurité de RSA réside au niveau de la longueur des nombres p et q. Ces deux nombres ont été codés sur 512 bits, 640 bits et 1024 bits. Actuellement sont codés sur 2048 et 4096 bits.

Performances :
Le RSA est moins rapide que le DES à cause des calculs des puissances modulo. Dans les réalisations matérielles le RSA est environ 1000 fois plus lente que le DES. Au niveau logiciel, le chiffrement de RSA est bien plus rapide si vous choisissez bien la valeur de e (les plus courants sont 3, 17 et 65537).

Conclusion :
Le RSA est le plus populaire parmi les algorithmes à clé publique. Vu la complexité de calculs, le RSA est souvent utilisé au début des communications pour échanger des clés de session (faible quantité d'information à communiquer) des algorithmes symétriques qui seront utilisés pour effectuer des communications en différé ou en temps réel.

Algorithme de Diffie et Hellman :
Diffie et Hellman ont été les fondateurs de la cryptographie à clés publiques. Leur algorithme a permis de résoudre le problème d'échange des clés de la cryptographie à clé symétrique. Son principe est basé sur l'exponentielle modulaire. La fonction utilisée est

$$g^m \bmod n,$$

Où g et n sont deux grands entiers premiers entre eux et sont publiques, m est un entier aléatoire gardé secret.

Le protocole d'échange des clés secrètes se déroule comme suit :

1. Au début deux correspondants A et B se mettent d'accord sur n et g,
2. A choisit un grand nombre entier aléatoire X gardé secret et calcul : $g^X \bmod n$ et envoie le résultat à B,
3. B choisit un grand nombre entier aléatoire Y gardé secret et calcul : $g^Y \bmod n$ et envoie le résultat à A,
4. A reçoit $g^Y \bmod n$, puis calcul $(g^Y \bmod n)^X \bmod n = g^{YX} \bmod n$,
5. B reçoit $g^X \bmod n$, puis calcul $(g^X \bmod n)^Y \bmod n = g^{XY} \bmod n$,

Ainsi, la valeur $K = g^{YX} \bmod n$ reçue par les deux parties devient leur secret partagé et par conséquent, elle peut être utilisé comme clé de session pour communiquer en utilisant un algorithme à clé symétrique.

K^A p = g^X mod n est la clé publique de A et sa clé privée est K^A pr = X.
K^B p = g^Y mod n est la clé publique de B et sa clé privée est K^B pr = Y.
Pour trouver les clés privées X et Y un attaquant doit résoudre l'inverse de g^X mod n ou g^Y mod n. Cette résolution reste un problème très redoutable en mathématique.

Algorithme d'ElGamal

La version de base de cet algorithme, publiée en 1985, a été utilisée pour effectuer des signatures numériques. Elle tire également sa sécurité sur la difficulté de calculer les logarithmes discrets. Une autre version modifiée a été proposée pour le chiffrement et le déchiffrement des messages.

3.3.1 Signature numérique des messages

Pour générer une paire de clé, un correspondant doit choisir un nombre premier n et deux nombre aléatoire g et x < n. Ensuite, il doit calculer :

$y = g^x \bmod n$

Ce qui permet de générer :

Une clé publique Kp =(y, g, n)
Une clé privée kpr = x

La signature d'un message M nécessite le choix d'un nombre aléatoire k tel que k et n-1 soient premiers entre eux. Cette signature sera composée de deux parties. La première partie de la signature est exprimée par :

$a = g^k \bmod n$

La deuxième partie de la signature est obtenue en résolvant l'équation ci-dessous avec l'algorithme d'Euclide étendu:

M = (xa + kb) mod (n-1)

La signature de M sera donc représentée par a et b

Pour vérifie une signature, un correspondant B reçoit M, aA et bA. Il utilisera la clé publique de A : $\mathbf{K_p}$ (yA, gA, nA) pour confirmer l'égalité suivante :

$y^{aA}\ a^{bA} \bmod nA = g^{M}_{A} \bmod nA$

Exemple :

1- Clé publique et privée de A

- nA = 11
- gA = 2 < nA
- xA < nA , xA = 8
- $yA = gA^{xA} \bmod nA = 2^8 \bmod 11 = 3$

K^A_p = (3, 2, 11) **K^A_{pr} = 8**

2- signature de M=5

- Choix de kA = 9, nA – 1 = 10, PGCD(9,10) = 1
- Calcul de $aA = g^{kA} \bmod nA = 2^9 \bmod 11 = 6$
- Résolution de $M = y^{aA}_A \, a^{bA}_A \bmod nA = g^M_A \bmod nA$

  $5 = (8*6 + 9*bA) \bmod 10$

L'algorithme d'Euclide étendu donne bA = 3, Signature de M = (6,3)

3- La correspondant B reçoit M = 5et (6 ,3), pour confirmer la signature de A, il calcul : $y^{aA}_A a^{bA}_A \bmod nA = g^M_A \bmod nA$

$3^6 .6^3 \bmod 11 = 2^5 \bmod 11 = 10$

Il est sûr que c'est A qui a signé le message M.

Chiffrement de déchiffrement

Un correspondant B envoie un message M à A, en utilisant la clé publique de A :
K^A_p = (yA, gA , nA). Le message chiffré sera composé de deux parties comme suit :

$C1 = g^{kB}_A \bmod nA$
$C2 = y^{kB}_A \bmod nA$

kB est un nombre aléatoire choisi par B tel que PGCD (kB, nA – 1) =1

C = C1C2 le message chiffré dont la taille est le double de celle de M.

Le correspondant A reçoit C et le déchiffre avec sa clé privée xA en utlisant l'expression suivante :

$M = C2 / C^{xA}_1 \bmod nA$

En effet,

$C2/ C^{xA}_1 \bmod nA = y^{kB}_A \bmod nA / [g^{xAkB}_A \bmod nA] \bmod nA$
$= (g^{xAkB}_A \bmod nA) \, M \bmod nA \; / \; [g^{xAkB}_A \bmod nA] \bmod nA$
$= M.$

Exemple

Clés publique et privée de A : KAp = (yA, gA, nA) = (3, 2, 11) et KApr = xA =8

B chiffre M=5 en utilisant kB = 3, PGCD(3,10) =1

C1 = 23 mod 11 = 8

C2 = 3^3. 5 mod 11 =25

B envoie à A C1C2, A utilise sa clé privée xA = 8,

M = 25 / 8^8 mod 11 =25 / 5 =5.

Conclusion

Le chiffrement / déchiffrement, l'authentification et la signature d'ElGamal ne peuvent être effectués que sur des messages dont la taille est entre 8 à 64 caractères (de 64 bits à 512 bits). Ceci est dû à la complexité du calcul de l'exponentielle modulaire qui utilise des clés publiques ou privées représentées par des nombres de grandes valeurs de 100 à 500 chiffres (de 512 bits à 2048 bits).

Exemple de clé publique sur 1024 bits en Hexadécimal :

3081 8902 8181 00CF 8424 B08C CD71 9110 7E44 2B2E 8014 35F0 49CE
8C3E 8CA9 3516 5FC7 9EB8 B4D2 9A89 637C 20C4 DB30 97AF ECB3
37F2 A000 00E8 E350 BA90 2B20 EEE5 9D5B 4A87 E0D5 895A B6A4
05A6 B2C4 2715 555F 6699 0A68 95AD 3963 6071 4C00 8431 7693 7EC0
20F9 8C31 EC2A 8585 9054 3478 4DD1 366B 9024 67B1 E8C8 C812
6EE9 E35B 5D04 700D 7C28 2702 0301 0001

Les algorithmes asymétriques sont très coûteux en temps de calcul

Utilisation courante : Echange des clés de session des algorithmes symétriques (8 caractères (64 bits) à 32 caractères (256 bits)) en utilisant le service de chiffrement / déchiffrement.

Signature et l'authentification des messages plus longs n'est possible que sur des résumés des messages

Solution adoptée :

Fonction de hachage → résumé → signature et authentification

Chapitre 4 : Authentification, hachage, signature et gestion de clés

1. Signature numérique :

La signature numérique doit avoir les mêmes caractéristiques qu'une signature manuscrite à savoir :

- Elle ne peut être limitée pour prouver que la signataire a délabrement signé son document.
- Elle doit authentifier la signature
- Elle n'appartient qu'à un seul document : pas réutilisable
- Un document signé ne peut être modifié (intégralité)
- Elle ne peut être reniée (non répudiation).

Existence d'une relation biunivoque entre un document signé et la signature qui l'accompagne.

Les algorithmes asymétriques permettent une signature numérique des messages très court (<256 bits) ➔ pour les messages de grandes tailles ➔ Utilisation de fonction de contraction ou de condensation appelée fonction de hachage à sens unique. ➔ Génération de résumé appelé une empreinte digitale ou un condensé ou un digit du document initial.

- Notion de fonction de hachage à sens unique (one way hach function)
C'est une fonction mathématique facile à calculer dans un sens (chiffrement facile) et est extrêmement difficile à calculer dans le sens inverse (déchiffrement difficile).

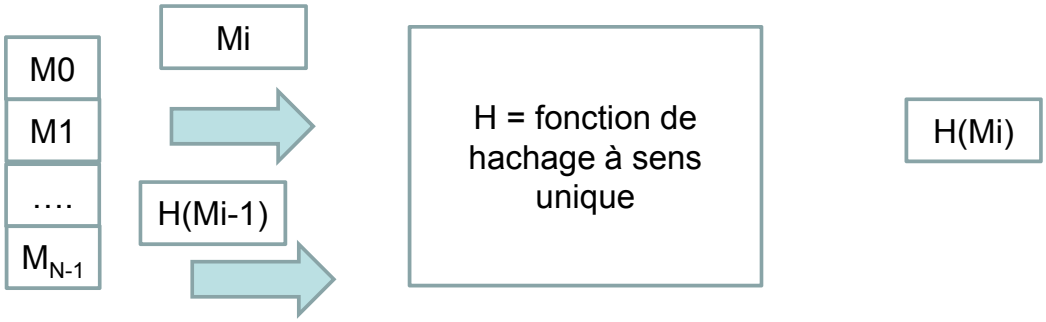
M➔ H(M)➔ empreinte digitale

Propriété de H(M) :

- M est de taille variable ➔ empreinte h=H(M) de taille fixe
- Il est facile de calculer h à partir de M
- Impossible de calculer M à partir de H(M)
- Rapidité du calcul de l'empreinte
- Impossible de générer deux documents différents ayant la même empreinte. (H(M) est sans collision)
- Il est très difficile de trouver deux messages aléatoires qui donnent la même signature.

Principe de fonctionnement :

Le message M = M0 M1 …M_{N-1} les blocs Mi sont de même taille. Chaque bloc Mi subira un hachage : H(Mi) = f(Mi, H(Mi-1))



Pour le premier bloc M0, une constante sera utilisée comme valeur initiale, pour le bloc M1, H(M0) représente une valeur d'initiation et ainsi de suite. A la fin du hachage H (M_{N-1}) représente l'empreinte digitale du message.

- M de taille quelconque ➔ H(M) de taille fixe (valeurs standards 128 ou 160 bits)
Particularité des fonctions de hachage :
  - Algorithmes associés sont publiques
  - Une fois M haché on ne peut plus restitué M (sens unique)

- Assure l'intégralité mais ne peuvent pas authentifier l'auteur de M ➔

- Nécessité de la cryptographie asymétrique qui offre le service d'authentification ➔

- H(M) et RSA ou ELGAMAL == à signature numérique ➔

- Fonction de hachage à sens unique avec clé (ou à brèche secrète)

Création d'une signature numérique

Le correspondant A signe un message M en utilisant une fonction de hachage H(M) et en chiffrant le résultat avec sa clé privée K^{A}_{pr} le résultat est transmis à un correspondant B.

$$\mathbf{S(M)=D_{K^{A}_{pr}}(H(M))}$$

Remarque :

Le message M peut être clair ou chiffré avec un algorithme symétrique dont la clé a été préalablement échangée entre A et B en utilisant un algorithme asymétrique. Dans ce cas B reçoit C= $E_{K^{B}_{p}}$(M) et S(M)

Vérification d'une signature numérique

Le correspondant B reçoit Mr et S(M) . Il utilise la clé publique de A (K^{A}_{p}) pour déchiffrer la signature S(M).

$$H(M)=E_{K}{}^{A}{}_{p}(S(M))=E_{K}{}^{A}{}_{p}(D_{k}{}^{A}{}_{pr}(H(M))$$

Et hache Mr pour calculer H(Mr)

Si H(Mr)=H(M) la signature est valide sinon M est à jeté

Inconvénient majeur du protocole de la signature numérique : certification de la clé publique du correspondant A.

Exemples d'utilisation

Signature en utilisant :

Un algorithme de hachage :
- MD4 : Message Digist version 4 (produit des empreintes de 128 bits)
- MD5 : Message Digist version 5 (produit des empreintes de 128 bits)
- SHA2 : Hach Algorithm version 2
- RIPEMD 160 : Ripe Message Digest

Et un algorithme asymétrique :
RSA
ELGAMAL

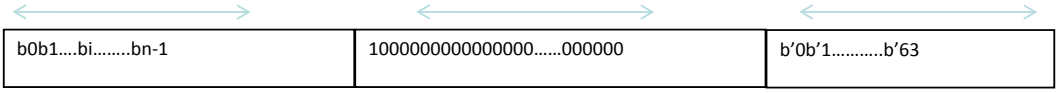
- ou l'utilisation du :
Standard de signature numérique (DSS)
DSA Digital Signature

Description de l'algorithme de hachage MD5

MD5 est une version améliorée de MD4. Le principe de base de cet algorithme, conçu par R. Rivest, est fondé sur la manipulation de blocs de 512 bits chacun (64 caractères). Chaque bloc est décomposé sous la forme de 16 sous blocs de 32 bits chacun. Ces sous blocs constituent une des entrées de l'algorithme. 4 fonctions non linéaires sont utilisées. Chaque fonction est utilisée dans une ronde (MD5 possède 4 rondes). Chaque ronde effectue 16 itérations. La sortie de l'algorithme est un ensemble de 4 variables de 32 bits. De ce fait, le résultat de hachage est codé sur 128 bits.

Étape 1 : **pré traitement du message**
Le message M peut avoir une taille quelconque, il doit être complété de manière à ce que sa taille soit multiple de 512 bits (M mod 512 =0)
Message multiple de 512 bits est M'.

Message M initial Complément de M taille de M



Le complément de M représente un remplissage qui commence avec 1 seul et d'autant de 0 que nécessite et ensuite la taille initiale de M est codée sur 64 bits et est placée à la fin du message.
- Découpage de M' en blocs de 512 bits chacun

B0	B1	…..	Bm

Chaque bloc Bi découpé en 16 sous blocs de 32 bits chacun
Bi ➔ M[0], M[1], …, M[15]

Étape 2 : **Initialisation pour le traitement de B0**
On définit 4 variables (variables de chaînage) A,B,C,D de 32 bits initialisées par :
A = 01 23 45 67
B = 89 AB CD EF
C = FE DC BA 98
D = 76 54 32 10
Ces 4 variables sont copies dans 4 autres variables :
AA = A, BB = B, CC = C, DD = D

Étape 3 : Calcul itérative des 4 rondes (4*16=64 itérations)
Dans chaque ronde , on calcul de nouvelles valeurs des variables A, B, C, D à partir des anciennes valeurs en utilisant dans chaque ronde une fonction non linéaire (pour évoluer ces variables de manière non-linéaire, et d'assurer les propriétés de sens unique).

Ronde 1

F(X,Y,Z)=(X **AND** Y) OR (**NOT**(X) **AND** Z)

Ronde 2

G(X,Y,Z)=(X **AND** Z) OR (Y **AND NOT**(Z))

Ronde 3

H(X,Y,Z)=(X **XOR** Y **XOR** Z)

Ronde 4

I(X,Y,Z)=Y **XOR** (X **XOR NOT**(Z))

Où X, Y et Z dénotent l'un des variables A, B,C ou D. Ces fonctions renvoient des valeurs sur 32 bits qui seront ajoutées à la 4 eme variable (non impliquée dans le calcul), et au sous bloc considéré et à une constante T. Ce nouveau résultat sera ensuite décalé circulairement vers la gauche de S positions. Enfin, le résultat de ce décalage est ajouté à l'une des variables A,B,C ou D.

Détails du calcul de A,B,C et D
M[j] : j (0 à 15) ème bloc
T[i] : constante calculée à l'itération i par 429467296 . abs(sin[i]) (i en radian).
S[i] : nombre de décalages correspondant à l'itération i.

Ronde 1

A = B + [(A + F(B,C,D) + M[j] + T[i])<<<S[i]]
B = C + [(B + F(C,D,A) + M[j] + T[i])<<<S[i]]
C = D + [(C + F(D,A,B) + M[j] + T[i])<<<S[i]]
D = A + [(D + F(A,B,C) + M[j] + T[i])<<<S[i]]

Ronde 2

A = B + [(A + G(B,C,D) + M[j] + T[i])<<<S[i]]
B = C + [(B + G(C,D,A) + M[j] + T[i])<<<S[i]]
C = D + [(C + G(D,A,B) + M[j] + T[i])<<<S[i]]
D = A + [(D + G(A,B,C) + M[j] + T[i])<<<S[i]]

Ronde 3

A = B + [(A + H(B,C,D) + M[j] + T[i])<<<S[i]]
B = C + [(B + H(C,D,A) + M[j] + T[i])<<<S[i]]
C = D + [(C + H(D,A,B) + M[j] + T[i])<<<S[i]]
D = A + [(D + H(A,B,C) + M[j] + T[i])<<<S[i]]

Ronde 4

A = B + [(A + I(B,C,D) + M[j] + T[i])<<<S[i]]
B = C + [(B + I(C,D,A) + M[j] + T[i])<<<S[i]]
C = D + [(C + I(D,A,B) + M[j] + T[i])<<<S[i]]
D = A + [(D + I(A,B,C) + M[j] + T[i])<<<S[i]]

A la fin de la 4eme ronde on fait les mise à jour des variables :
A = A + AA
B = B + BB
C = C + CC
D = D + DD
Ces 4 variables seront utilisées comme entrée pour le **traitement du bloc B1**.
Les 4 rondes sont effectuées autant de fois qu'il y a de blocs de 512 bits dans le message complété M'.

Étape 4 : Empreinte du message M

L'empreinte du message sera codée sur 128 bits en regroupant les 4 variables A, B, C et D de 32 bits.

Concurrents de MD5

SHA 1 : empreinte sur 160 bits
SHA 2 : 3 choix 256, 384 ou 512 bits
RIPEMD 160 : empreinte sur 160 bits

Le standard de la signature numérique DSS

L'algorithme de signature à clé publique (DSA : Digital Signature Algorithme) a été proposée par le NIST en 1991, comme standard de signature numérique (DSS : Digital Signature Standard).
Le DSA est une variable de l'algorithme de signature d'ElGamal.

- Génération d'une paire de clés d'un correspondant :
n nombre premier entre 512 et 1024 bits
q nombre premier avec (n-1) de 160 bits
calculer $g=h^{(n-1)/q} \bmod n$ où $h<n-1$ tel que $g>1$
calculer $y=g^x \bmod n$ avec $x<q$

➔

une clé publique Kp=(q,y,g,n)
une clé privée Kpr=x

Signature de l'empreinte H(M) d'un message M

Choix aléatoire de K<q

Calculer :
$R = (g^k \bmod n) \bmod q$ 1ere partie de la signature
$S = (K^{-1} (H(M)+x*R)) \bmod q$ 2eme partie de la signature
(R,S) représente la signature du message M, en utilisant la fonction de hachage SHA1.

Un correspondant A ayant la clé publique (q_a,y_a,g_a,n_a) hache son message et le signe en utilisant sa clé privée x_a et K_a, il envoie à B l'empreinte $H_a(M)$, le message M en clair ou chiffré et la signature (R_a, S_a).

Vérification de la signature
Un correspondant B reçoit l'empreinte $H_a(M)$ et (R_a, S_a). Il utilisera la clé publique de A pour calculer :

$W = S^{-1}_A \bmod q_A$
$U1 = (H_A(M)*W \bmod q_A$
$U2 = R_A*W \bmod q_A$
$V = ((g^{U1}_A * y^{U2}_A) \bmod n_A) \bmod q_A$

Si $V=R_A$ alors la signature est vérifiée
➔ A a bien signé le message M en utilisant $H_A(M)$ avec sa clé privée x_A et le nombre K_A.
Exemple :
Génération d'une paire de clés d'un correspondant A :
n_A= 124540019 nombre premier entre 512 et 1024
q_A= 17389 nombre premier avec (n-1) de 160 bits
$(n-1)/q_A$=7162 et h=11021752 < 124540018
$g_A = 110217528^{7162} \bmod 124540019 = 10083255 > 1$
$x_A = 12496 < q_A$
$y_A = 10083255^{12496} \bmod 124540019 = 119946265$
une clé publique $Kp^A = (q_A,y_A,g_A,n_A)$
une clé privée $Kpr^A=x_A$
Signature par A de l'empreinte HA(M) d'un message M
Choix aléatoire de KA=9557 $<q_A$
calculer :
1ere partie de la signature
$R_A=(10083255^{9557} \bmod 124540019) \bmod 17389 = 34$

2eme partie de la signature
$S_A=(9557^{-1} (5246+12496*34)) \bmod 17389 = 13049$
Avec H(M)=5246
A envoie à B (34, 13049)

Vérification de la signature

Le correspondant B reçoit d'un correspondant A l'empreinte HA(M) (le message en clair ou chiffré et (R_A=34, S_A=13049). Il utilisera la clé publique de A pour calculer :

W= 34^{-1} mod 117389 = 1799

U1=(5246*1799) mod 117389 = 12716

U2 = 13049* 1799 mod 117389 = 8999

V=(($10083255^{12716}{}_A$*$y^{8999}{}_A$) mod n_A) mod 117389 =34

V=RA=34 alors la signature est validée

➔A a bien signé le message M en utilisant H_A(M) avec sa clé privée x_A = 12496 et le nombre K_A= 9557.

Gestion des clés et certificat numérique

Le problème majeur de la cryptographie asymétrique est la publication des clés publiques des différents utilisateurs. Cette publication doit offrir l'assurance que :

- Les clés publiques sont bien celles des utilisateurs à qui elles sont associées.
- L'organisme qui publie les clés publiques doit être digne de confiance (tiers de confiance).
- Les clés publiques sont validées.

Pour cela, on a recours à un **certificat numérique**. Un certificat numérique est un document numérique fixant les relations qui existent entre une clé publique et son propriétaire (un utilisateur, une application, un site, une machine…).

La gestion technique et administrative de ces certificats sont assurées par une infrastructure de gestion des clés IGC ou Public Key Infrastructure (PKI). Une PKI repose sur les organismes suivants :

- **L'autorité de certification AC**, dont la fonction est de définir la politique de certification (PC) et de la faire appliquer, garantissant ainsi un certain niveau de confiance aux utilisateurs.
- **L'autorité d'enseignement (AE)**, dont la fonction est de vérifier que le demandeur est bien la personne qu'il prétend être, conformément aux règles définies par l'autorité de certification. Elle garantit la validité des informations contenues dans le certificat. L'autorité d'enseignement est le lien entre l'opérateur de certification et l'abonné.
- **L'opérateur de certification (OC)**, dont la fonction est d'assurer la fourniture et gestion du cycle de vie des certificats. Son rôle consiste à mettre en œuvre une plate forme opérationnelle, fonctionnelle, sécurisée, dans le respect des exigences énoncées dans la politique de certification (PC) et dont les modalités sont détaillées dans la déclaration des pratiques de certification (DPC).

Annexe : Norme X.509

X 509 est un standard de cryptographie de l'Union international des télécommunications pour els infrastructures à clés publique. X509 établit entre autres les formats standard de certificats électronique et un algorithme de validation de chemin de certification.
X509 a été créé en 1988 dans le cadre du standard X500. Il repose sur un système hiérarchique d'autorités de certifications.

Un certificat X.509 est décomposé en deux parties :
• La partie contenant les informations (version, n° de série du certificat, algorithme utilisé , nom, date de validité, clé publique du propriétaire du certificat, …)
• La partie contenant la signature de l'autorité de certification

Exercice :
Discuter les cas suivants:
1. Deux certificats différents contiennent la même clé publique.
2. Deux certificats différents ont la même signature.