Faculté des Sciences et Techniques
(F. S. T.)

Département de Mathématiques et Informatique
(D. M. I.)

Laboratoire d'Algèbre de Cryptologie de Géométrie Algèbrique et Applications
(L. A. C. G. A. A.)

TITRE :

CRYPTOGRAPHIE

Demba SOW
Docteur en Mathématiques et Cryptologie
Enseignant chercheur au Département de Mathématiques/Informatique
demba1.sow@ucad.edu.sn, sowdembis@yahoo.fr

Année académique 2016-2017

Naissance spontanée de la cryptographie

Les écritures secrètes semblent être nées spontanément dès que, dans un pays, une partie importante de la population a su lire.

1.2.2 Entre 1000 et 1800 : l'éveil de l'occident

Jusque-là largement devancé par la science arabe, l'Occident développe la cryptographie et la cryptanalyse.

1.2.3 Entre 1800 et 1970 : l'essor des communications

- Les nouvelles techniques de communications (moyens de transports rapides, journaux, télégraphe, télégraphie sans fil) donne une nouvelle impulsion à la cryptologie.
- Les guerres modernes utilisent abondamment les télécommunications ;
- L'interception devient simple et le décryptement des informations devient vital.
- La cryptologie entre dans son ère industrielle.

1.2.4 La cryptologie moderne : de 1970 à nos jours

Cryptographie moderne

- Les ordinateurs et le réseau Internet font entrer la cryptologie dans son ère moderne.
- La grande invention de ces dernières décennies fut la cryptographie à clefs publiques.
- Le futur sera peut-être la cryptographie quantique, définitivement indécryptable.

1.3 Concepts de base

1.3.1 Cryptologie

- **Cryptographie :** Discipline incluant les principes, les moyens et les méthodes de transformation des données, dans le but de masquer leur contenu, d'empêcher leur modification ou leur utilisation illégale.
- **Cryptologie :** Science des messages secrets. Se décompose en cryptographie et cryptanalyse.
- Le mot cryptologie est souvent utilisé comme synonyme de cryptographie.

1.3.2 Stéganographie

- **Stéganographie :** contrairement à la cryptographie, qui chiffre des messages de manière à les rendre incompréhensibles, la stéganographie (en grec "l'écriture couverte") cache les messages dans un support ;
- **Stéganographie :** mode de communication secrète obtenu en dissimulant l'existence du message ;
- Par exemple des images ou un texte qui semble anodin ;
- L'idée est la même pour les grilles de Cardan et le «barn code» ;

le texte caché.

1.3.3 Lexique

- **Chiffrer :** transformer à l'aide d'une convention secrète, appelée clé, des informations claires en informations inintelligibles pour des tiers n'ayant pas la connaissance du secret ;
- **Déchiffrer :** retrouver les informations claires, à partir des informations chiffrées en utilisant la convention secrète de chiffrement ;
- **Décrypter :** retrouver l'information intelligible, à partir de l'information chiffrée sans utiliser la convention secrète de chiffrement ;
- Par contre, **crypter** ou **encrypter** n'a pas de sens clairement défini, mais sont parfois utilisés à tort comme synonymes de chiffrer ;
- **Cryptogramme :** Message chiffré ou codé ;
- **Chiffre :** Ensemble de procédés et ensemble de symboles (lettres, nombres, signes, etc.) employés pour remplacer les lettres du message à chiffrer. On distingue généralement les chiffres à transposition et ceux à substitution ;
- **Clé :** Dans un système de chiffrement, elle correspond à un nombre, un mot, une phrase, etc. qui permet, grâce à l'algorithme de chiffrement, de chiffrer ou de déchiffrer un message ;

1.4 Notion de cryptosystème

Un système cryptographique est un quintuplet $S = \{\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}\}$ avec :
- $\mathcal{P}$: ensemble fini de clairs (plain texts) ;
- $\mathcal{C}$: ensemble fini de chiffrés (cipher texts) ;
- $\mathcal{K}$: ensemble fini de clés (key space) ;
- $\mathcal{E}$: ensemble fini de règles de chiffrement (encryption rules)
- $\mathcal{D}$: ensemble fini de règles de déchiffrement (decryption rules)

$\forall\ k \in \mathcal{K},\ \exists\ e_k \in \mathcal{E}$ tel que $e_k : \mathcal{P} \to \mathcal{C}$
$\exists\ d_k \in \mathcal{D}$ tel que $d_k : \mathcal{C} \to \mathcal{P}$ et $d_k\ \circ\ e_k = id_{\mathcal{P}}$

1.5 Scénario d'un cryptosystème

1.5.1 Personnage

a. Un expéditeur **Alice** veut envoyer un message à un destinataire **Bob** en évitant les oreilles indiscrète d'**Eve**, et les attaques malveillantes de **Martin**.
b. Pour cela Alice se met d'accord avec Bob sur le cryptosystème qu'ils vont utiliser.
c. Ce choix n'a pas besoin d'être secret en vertu du principe de Kerckhoff.
d. Par exemple ils peuvent décider d'utiliser un des systèmes suivants :
- un code par blocs (block-cipher) à clé publique ou code asymétrique.
- un code par blocs à clef secrète ou code symétrique.
- un code par flots ou en continu (stream-cipher).

- Pour crypter le message M Alice le découpe en blocs $(m_i)_{1 \leq i \leq n}$, de même taille.
- Alice transforme chaque bloc m_i de texte en clair à l'aide de la fonction de chiffrement, e_{KC}, dépendant d'une clé de chiffrement, KC, en un bloc c_i, le texte chiffré.
- A l'autre extrémité Bob déchiffre, décrypte le message codé à l'aide la fonction de déchiffrement, d_{KD}, dépendant d'une clé de décryptage ou de déchiffrement, KD.

1.6 Qualités d'un cryptosystème

1.6.1 L'émetteur (Alice) veut être certain

- qu'une personne non-autorisée (Eve) ne peut pas prendre connaissance de ses messages ;
- que ses messages ne sont pas falsifiés par un attaquant malveillant (Martin) ;
- que le destinataire (Bob) a bien pris connaissance de ses messages et ne pourra pas nier l'avoir reçu (non-répudiation) ;
- que son message n'est pas brouillé par les imperfections du canal de transmission (cette exigence ne relève pas du cryptage mais de la correction d'erreur).

1.6.2 Le récepteur (Bob) veut être certain

- que le message reçu est authentique c'est à dire :
  - que le message n'a pas été falsifié par un attaquant malveillant (Martin) ;
  - que le messages vient bien d'Alice (autrement dit qu'un attaquant (Oscar) ne se fait pas passer pour Alice, mascarade).
- que l'expéditeur (Alice) ne pourra pas nier avoir envoyé le message (non-répudiation) ;
- que le message n'est pas brouillé (par les imperfections du canal de transmission ou par un brouillage intentionnel), autrement dit qu'il est identique à l'original que lui a envoyé Alice.

1.6.3 Qualités

Les qualités demandées à un système cryptographique sont résumées par les mots clés suivants :

1. **Intégrité des données :** le message ne peut pas être falsifié sans qu'on s'en aperçoive ;
2. **Identité des interlocuteurs du message :**
   - l'émetteur est sûr de l'identité du destinataire c'est à dire que seul le destinataire pourra prendre connaissance du message car il est le seul à disposer de la clé de déchiffrement ;
   - Authentification, le receveur est sûr de l'identité de l'émetteur grâce à une signature.
3. **Non-répudiation** se décompose en trois :
   - **non-répudiation d'origine :** l'émetteur ne peut nier avoir écrit le message ;
   - **non-répudiation de reception :** le receveur ne peut nier avoir reçu le message ;
   - **non-répudiation de transmission :** l'émetteur du message ne peut nier avoir envoyé le message.

Chapitre 2

CRYPTOGRAPHIE CLASSIQUE

2.1 Codes à répertoires

2.1.1 Définitions

En cryptographie, les systèmes à répertoires sont des systèmes de substitution qui reposent sur l'utilisation de tableaux de correspondance ou dictionnaires chiffrés.

Les éléments de ces tableaux sont représentés sous forme de lettres, de syllabes, de mots ou de phrases.

D'une taille importante, ces listes sont impossibles à retenir par coeur et doivent donc être écrites pour former un ouvrage que l'on peut nommer à choix :

- code
- répertoire
- dictionnaire chiffré
- système à dictionnaire
- tableau de chiffrement.

2.1.2 Avantages et Incovénients

Inconvénients

- Leur défaut évident et irréductible est d'exister à l'état de livre imprimé plus ou moins volumineux, mais sujet à perte, vol ou copie.
- L'histoire du chef d'état-major d'Osman-Pacha pendant la guerre russo-turque de 1877, parti en tournée d'inspection en emportant le code secret pendant que son malheureux général en chef recevait des dépêches sans pouvoir les traduire, est restée célèbre.

Avantages

- Par contre, leur avantage est d'être d'un emploi simple et rapide, et peu sujet aux erreurs s'il est utilisé par une personne suffisamment soigneuse.
- D'autre part, avec l'utilisation de 4 ou 5 symboles (que l'Administration des Télégraphes ne taxait que pour un mot), on peut figurer toute une phrase, beaucoup plus coûteuse à transmettre :
- d'où le succès de ces codes dans le **commerce et la finance**, où le secret n'a souvent pas une importance aussi considérable qu'en **diplomatie ou aux armées.**

- Il existe des répertoires à chiffres et d'autres à lettres.
- Le **Bentley's Complete Phrase Code Numbered** est un répertoire ordonné qui est à la fois à chiffres et à lettres (au choix de l'utilisateur).

Qu'ils soient à chiffres ou à lettres, les répertoires se divisent en deux catégories : les **répertoires ordonnés** et les **répertoires incohérents**.

1. **Répertoires ordonnés :**
   - Lorsque, dans le tableau de correspondance, les deux listes (mots et représentations) sont ordonnées toutes deux alphabétiquement ou numériquement, on dit que le répertoire est ordonné.
   - Dans les répertoires ordonnés, on utilise la même table pour chiffrer et déchiffrer.
2. **Répertoires incohérents :**
   - Pour compliquer la tâche des décrypteurs, on peut utiliser un ordre des mots incohérent (ordre non alphabétique).
   - On aura alors besoin de deux tables : une pour chiffrer et une autre pour déchiffrer.
   - De tels répertoires sont dits incohérents (on dit aussi désordonnés ou à bâtons rompus).

2.2 Chiffrement par transposition

2.2.1 Principe

- Elles consistent, par définition, à changer l'ordre des lettres.
- C'est un système simple, mais peu sûre pour de très brefs messages car il y a peu de variantes.
- Ainsi, un mot de trois lettres ne pourra être transposé que dans $6(= 3!)$ positions différentes.
- Par exemple, "col" ne peut se transformer qu'en "col", "clo", "ocl", "olc", "lco" et "loc".
- Lorsque le nombre de lettres croît, il devient de plus en plus difficile de retrouver le texte original sans connaître le procédé de brouillage.
- Ainsi, une phrase de 35 lettres peut être disposée de $35! = 1040$ manières différentes.
- Ce chiffrement nécessite un procédé rigoureux convenu auparavant entre les parties.
- Une transposition rectangulaire consiste à écrire le message dans une grille rectangulaire, puis à arranger les colonnes de cette grille selon un mot de passe donné (le rang des lettres dans l'alphabet donne l'agencement des colonnes).

2.2.2 Transposition par blocs

- Le message est découpé en des blocs de taille égale ;
- Puis sont rangés dans des matrices de n lignes et m colonnes ;
- Pour chiffrer :
  - les blocs sont rangés suivant les lignes (resp. les colonnes) dans des matrices
  - puis la lecture se fait suivant les colonnes (resp. les lignes) pour le chiffrement.
- Pour déchiffrer, on effectue les mêmes opérations à l'inverse.

– Le message est découpé en des blocs de taille égale à celle de la clé ;
– Puis sont rangés dans des matrices de n lignes (longueur de la clé) et m colonnes ;
– Pour chiffrer, on lit par colonne, dans l'ordre défini par la clé ;
– Pour déchiffrer, on effectue les mêmes opérations à l'inverse.

2.3 Chiffrement par substitution

2.3.1 Substitution monoalphabétique

Principe :
– Chaque lettre est remplacée par une autre lettre ou symbole.
– Parmi les plus connus, on citera le **chiffrement de César**, le **chiffrement affine**, ou encore les chiffres désordonnés.
– Tous ces chiffres sont sensibles à l'analyse de fréquence d'apparition des lettres (nombre de fois qu'apparait une même lettre dans un texte).
– De nos jours, ces chiffres sont utilisés pour le grand public, pour les énigmes de revues ou de journaux.

Historique :
– Historiquement, on recense des procédés de chiffrement remontant au Xème siècle avant JC.
– On trouve par exemple :
  – l'**Atbash des Hébreux** (-500) ;
  – la **scytale à Sparte** (-400) ;
  – le **carré de Polybe** (-125), ...

Exemples :

1. **Chiffrement de César (50 av. J-C) :**
   – Il s'agit d'un des plus simples et des chiffres classiques les plus populaires.
   – Son principe est un décalage des lettres de l'alphabet.
   – Dans les formules ci-dessous, p est l'indice de la lettre de l'aphabet, k est le décalage.
   – Pour le chiffrement, on aura la formule

$$C = E(p) = (p + k) \mod 26$$

   – Pour le déchiffrement, il viendra

$$p = D(C) = (C - k) \mod 26$$

   – Si on connait l'algorithme utilisé (ici **César**), la cryptanalyse par force brute est très facile.
   – En effet, dans le cas du chiffre de **César**, seules 25! clés sont possibles.
2. **Chiffrement affine :**
   – On dit qu'une fonction est affine lorsqu'elle est de la forme $x \mapsto ax + b$, c'est-à-dire un polynôme de degré 1.
   – Une fonction linéaire est une fonction affine particulière.

$$y = (ax + b) \mod 26,$$

où a et b sont des constantes, et où x et y sont des nombres correspondant aux lettres de l'alphabet ($A = 0, B = 1, ...$).

- On peut remarquer que si $a = 1$, alors on retrouve le chiffre de **César** où b est le décalage (le k du chiffre de **César**).

Propriété de neutralité :

- Si $b = 0$, alors "a" est toujours chiffré "A" car il ne subit aucun décalage.
- En effet, si aucun décalage n'a lieu, l'alphabet de départ se retrouve chiffré par lui même, et donc ne subit aucune modification.
- Pour le chiffrement affine, la clé est constituée de (k_1, k_2) où $k_1,\ k_2 \in 2[0, 25]$ et telle que $\gcd(k_1, 26) = 1$.
- Le chiffrement en lui-même est donné par

$$c_i = f(m_i) = k_1 * m_i + k_2 \mod 26.$$

- Pour le déchiffrement, il vient

$$m_i = f^{-1}(c_i) = k_1^{-1} * (c_i - k_2) \mod 26.$$

- Par le chiffrement affine, on obtient 312 clés possibles.
- En effet, pour obéir à la propriété de k_1, il n'y a que 12 choix possibles.
- Et puisque k_2 peut prendre n'importe valeur dans $[0, 25]$, il vient $12 * 26 = 312$.

Exemple :

- Soient la clé $= (k_1, k_2) = (3, 11)$.
- Transformation de chiffrement :

$$c_i = f(m_i) = 3 * m_i + 11 \mod 26$$

- Transformation de déchiffrement :

$$k_1^{-1} = 3^{-1} \mod 26 = 9 \ [car\ 3 * 9 \mod 26 = 1]$$

$$m_i = f^{-1}(c_i) = 9 * (c_i - 11) \mod 26$$

- Ainsi, pour une suite de lettres telle que $'NSA' \rightarrow 13\ 18\ 0\ \rightarrow 24\ 13\ 11\ \rightarrow' YNL'$.

2.3.2 Substitutions polyalphabétiques

1. **Chiffrement de Vigenère (1568) :**
   - C'est une amélioration décisive du chiffre de **César**.
   - Sa force réside dans l'utilisation non pas d'un, mais de 26 alphabets décalés pour chiffrer un message.
   - On parle du carré de **Vigenère**.
   - Ce chiffrement utilise une clé qui définit le décalage pour chaque lettre du message (A : décalage de 0 cran, B : 1 cran, C : 2 crans, ..., Z : 25 crans).

Chiffrer le texte "**CHIFFRE DE VIGENERE**" avec la clé "**BACHELIER**" (cette clé est éventuellement répétée plusieurs fois pour être aussi longue que le texte clair)

Clair	C	H	I	F	F	R	E	D	E	V	I	G	E	N	E	R	E
Clé	B	A	C	H	E	L	I	E	R	B	A	C	H	E	L	I	E
Décalage	1	0	2	7	4	11	8	4	17	1	0	2	7	4	11	8	4
Chiffré	D	H	K	M	J	C	M	H	V	W	I	I	L	R	P	Z	I

2. **Chiffre de Vernam (One Time Pad - 1917)**

   Le masque jetable est défini comme un chiffre de **Vigenère** avec la caractéristique que la clé de chiffrement a la même longueur que le message clair.

Clair	M	A	S	Q	U	E	J	E	T	A	B	L	E
Clé	X	C	A	A	T	E	L	P	R	V	G	Z	C
Décalage	23	2	0	0	19	4	11	15	17	21	6	25	2
Chiffré	J	C	S	Q	N	I	U	T	K	V	H	K	G

   Pour utiliser ce chiffrement, il faut respecter plusieurs propriétés :
   - choisir une clé aussi longue que le texte à chiffrer ;
   - utiliser une clé formée d'une suite de caractères aléatoires ;
   - protéger votre clé ;
   - ne jamais réutiliser une clé.

2.4 Chiffrement polygraphique

- Il s'agit ici de chiffrer un groupe de n lettres par un autre groupe de n symboles.
- On citera notamment le chiffre de **Playfair** et le chiffre de **Hill**.
- Ce type de chiffrement porte également le nom de **substitutions polygrammiques**.

1. **Chiffrement de Playfair (1854)**

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

Règle 1

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

Règle 2

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

Règle 3

   - On chiffre 2 lettres par 2 autres. On procède donc par digramme.
   - On dispose les 25 lettres de l'alphabet (W exclu car inutile à l'époque, on utilise V à la place) dans une grille de $5x5$, ce qui donne la clé.
   - La variante anglaise consiste à garder le W et à fusionner I et J.
   - Il y a 4 règles à appliquer selon les deux lettres à chiffrer lors de l'étape de substitution.
   - Pour le déchiffrement, on procède dans l'ordre inverse.

   Principe :

   (a) Si les lettres sont sur des "coins", les lettres chiffrées sont les 2 autres coins.
   Exemple : OK devient **VA**, **RE** devient **XI** ...

   (b) Si les lettres sont sur la même ligne, il faut prendre les deux lettres qui les suivent immédiatement à leur droite.

immédiatement en dessous.

(d) Si elles sont identiques, il faut insérer une nulle (habitellement le X) entre les deux pour éliminer ce doublon.

Exemple : "**balloon**" devient "**ba**" "**lx**" "**lo**" "**on**".

– Pour former ces grilles de chiffrement, on utilise un mot-clé secret pour créer un alphabet désordonné avec lequel on remplit la grille ligne par ligne.

– Ensuite, on comble la grille avec les lettres restantes de l'alphabet.

2. **Chiffrement de Hill (1929) :**

Ce cryptosystème généralise celui de Vigenère. Il a été publié par **L. S. Hill** en 1929.

Les lettres sont d'abord remplacées par leur rang dans l'alphabet. Les lettres P_k et P_{k+1} deviennent C_k et C_{k+1}.

$$\begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} \pmod{26}$$

– Les composantes de cette matrice doivent être des entiers positifs.
– De plus la matrice doit être inversible dans $\mathbb{Z}_{26}$.
– Cependant, sa taille n'est pas fixée à 2.
– Elle grandira selon le nombre de lettres à chiffrer simultanément.
– Chaque digramme clair (P_1 et P_2) sera chiffré (C_1 et C_2) selon :

$$C_1 \equiv aP_1 + bP_2 \mod 26$$

$$C_2 \equiv cP_1 + dP_2 \mod 26$$

Exemple de chiffrement :

(a) Alice prend comme clé de cryptage la matrice $\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}$

(b) pour chiffrer le message "**je vous aime**" qu'elle enverra à Bob. Après avoir remplacé les lettres par leur rang dans l'alphabet ($a = 1,\ b = 2,\ etc...$), elle obtiendra

$$C_1 = 9 * 10 + 4 * 5 \mod 26 = 110 \mod 26 = 6$$

$$C_2 = 5 * 10 + 7 * 5 \mod 26 = 85 \mod 26 = 7$$

– Elle fera de même avec les 3^e et 4^e lettres, 5^e et 6^e, etc.
– Elle obtiendra finalement le résultat de la figure suivante :

Lettres	j	e	v	o	u	s	a	i	m	e
Rangs (P_k)	10	5	22	15	21	19	1	9	13	5
Rangs chiffrés (C_k)	6	7	24	7	5	4	19	16	7	22
Lettres chiffrés	F	G	X	G	E	D	S	P	G	V

Exemple de déchiffrement :

(a) Pour déchiffrer, le principe est le même que pour le chiffrement :
– on prend les lettres deux par deux ;
– puis on les multiplie par une matrice :

$$\begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} (\mod 26)$$

(b) Cette matrice doit être l'inverse de matrice de chiffrement (modulo 26). Ordinairement, cet inverse est :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

(a) Pour déchiffrer le message d'Alice, Bob doit calculer :

$$\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}^{-1} = \frac{1}{43} \begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} (\mod 26) = 43^{-1} \begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} (\mod 26)$$

(b) Comme $\gcd(43, 26) = 1$, $(43)^{-1}$ existe dans $\mathbb{Z}_{26}$ et $(43)^{-1} = 23$.
Bob a la matrice de déchiffrement :

$$\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}^{-1} = 23 \begin{pmatrix} 7 & -4 \\ -5 & 9 \end{pmatrix} (\mod 26) = \begin{pmatrix} 161 & -92 \\ -115 & 207 \end{pmatrix} (\mod 26) = \begin{pmatrix} 5 & 12 \\ 15 & 25 \end{pmatrix}$$

– Bob prend donc cette matrice pour déchiffrer le message "FGXGE DSPGV".
– Après avoir remplacé les lettres par leur rang dans l'alphabet ($A = 1, B = 2, etc...$), il obtiendra :

$$P_1 = 5 * 6 + 12 * 7 \mod 26 = 114 \mod 26 = 10$$

$$P_2 = 5 * 6 + 25 * 7 \mod 26 = 265 \mod 26 = 5$$

– Il fera de même avec les 3^e et 4^e lettres, 5^e et 6^e, etc.

Il obtiendra finalement le résultat de la figure suivante :

Lettres chiffrés	F	G	X	G	E	D	S	P	G	V
Rangs chiffrés (C_k)	6	7	24	7	5	4	19	16	7	22
Rangs (P_k)	10	5	22	15	21	19	1	9	13	5
Lettres	j	e	v	o	u	s	a	i	m	e

Chapitre 3

CRYPTOGRAPHIE MODERNE

3.1 Concepts de base

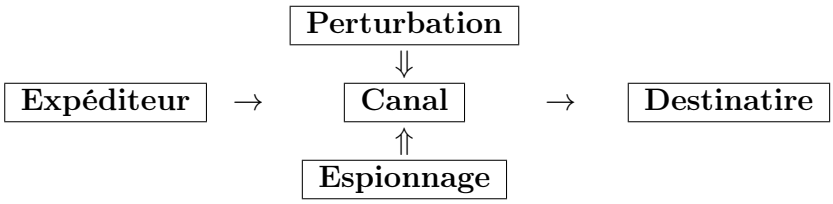
UN PEU DE VOCABULAIRE

- **Information :** élément de connaissance : texte, son, image, vidéo,...
- **Traiter/manipuler une information :** lire, écrire/modifier, effacer une information ;
- **Canal :** moyen de transmission permettant de convoyer une information entre deux partenaires d'une communication ; par exemple ligne téléphonique, fibre optique, moyen de communication sans fils,...
- **Entités :** quelqu'un (= personne) ou quelque chose (= machnie,...) capable de d'envoyer, de recevoir ou de traiter/manipuler une information. Dans la pratique, c'est l'un des partenaires d'une commmunication.
- **Un schémas de communication :** échange d'informations entre entités distantes

SCHEMA DE COMMUNICATION

Un schémas de communication (échange d'informations entre entités distantes) fait intervenir :

- **un expéditeur** (personne, machine,...),
- **un destinataire** (personne, machine,...),
- **un canal de transmission** (ligne téléphonique, fibre optique, systèmes de communication sans fils,...)
- **et un message** (information : texte, son, image, vidéo,...).



- Habituellment (ou systématiquement), le canal connait des perturbations (électromagnétique,...) et des personnes ou entités tiers interviennent pour espionner ou compromettre les communications.

- Avant la transmission, le message subit des transformations particulières appélées **codages** qui prennent en charge tous les problèmes (d'efficacité, de sécurité,...) qui se posent lors de la transmission.

d'erreurs et les codes secrets.

DEFINITION DES DIFFERENTS CODES

- **Codes de compréssion** : Lors d'une communication, l'information est transmise sous la forme d'une séquence de signaux et donc pour des soucis d'éfficacité (relativment à la capacité du canal), il faut utiliser des codes qui minimisent la longueur de la séquence (**Compression**).

- **Codes corecteurs d'erreurs :** Lors d'une transmission, après codage, des erreurs apparaissent car il y'a des signaux qui sont perdus ou altérés. A la reception, il est necessaire de pouvoir detecter et si possible de corriger les erreurs survenues (**Détection/Correction**).

- **Codes secrets** Lors d'une communication, un espion peut tenter de violer la confidentialité des données, de détourner et de modifier les données ou d'essayer d'usurper l'identitité de l'un des partenaires de la communication. Ainsi pour des raisons de sécurité, lors de l'envoie, un codage et/ou un protocole bien spécifique doit etre utilisé pour protéger la communication par rapport à des besoins de sécurité bien identifiés (Cryptographie).

TERMINOLOGIE : cryptologie, crytographie, cryptanalyse

- **Cryptologie** : c'est une science qui comporte deux branches la **cryptographie** et la **cryptanalyse**.

- **cryptographie** : traditionnellement c'est l'étude des méthodes permettant de transmettre des données de manière confidentielle. Afin de protéger un message, on lui applique une transformation qui le rend incompréhensible ; c'est ce qu'on appelle le **chiffrement**, qui, à partir d'un texte clair donne un **texte chiffré** ou **cryptogramme**. Inversement le **déchiffrement** est l'action qui permet de reconstruire le texte clair à partir du texte chiffré.
  Dans la cryptographie moderne, les transformations en question sont des fonctions mathématiques, appelées **algorithmes cryptographiques**, qui dépendent d'un paramètre appelé **clé**.

- **cryptanalyse** : à l'inverse, c'est l'étude des procédés cryptographiques dans le but de trouver des faiblesses et, en particulier, de pouvoir décrypter des textes chiffrés. Le **décryptement** est l'action consistant à retrouver le texte en clair sans connaitre la clé de déchiffrement.

Remarque 3.1.1. *Les termes "**cryptage**" et "**crypter**" sont des anglicismes dérivés de l'anglais* **to encrypt**, *souvent employés incorrectement à la place de **chiiffrement** et **chiffrer**.*

En toute rigueur, ces termes n'existent pas dans la langue française. Si le cryptage existait, il pourrait être défini comme l'inverse du décryptage, c'est-à-dire comme l'action consistant à obtenir un texte chiffré à partir d'un texte en clair sans connaître la clé.

SYSTEMES DE CRYPTOGRAPHIE

Un système de cryptographie est composé d'un quintuplet $(\mathcal{P}, \mathcal{C}, C_k, D_{k'}, \mathcal{K})$ où :

- $\mathcal{P}$ est un ensemble appelé espace des textes clairs

- $\mathcal{K}$ est un ensemble appelé espace des clés
- $Gen_{\mathcal{K}}$ un algorithme de génération de clés (=les éléments de $\mathcal{K}$) ;
- $C_k : \mathcal{P} \to \mathcal{C}$ est une fonction inversible à gauche appelée fonction de chiffrement et qui dépend d'un parametre k appelé clé.
- $D_{k'} : \mathcal{C} \to \mathcal{P}$ est la fonction inverse gauche de C_k (i.e $D_{k'} \circ C_k(m) = m, \forall m \in \mathcal{P}$) et est appelée fonction de déchiffrement (dépendant de la clé k'.)

TYPES DE SYSTEMES

En cryptographie les systémes peuvent être classés en deux catégories :

- les systèmes à clés secrètes (voir chapitre 2) ;
- les systèmes à clés publique/privée (voir chapitre 3) ;

Les systèmes à clés publique/privée sont, à leur tour, composés de deux familles :

- ceux basés sur **des algorithmes déterministes** (*pour une même donnée d'entrée, l'agorithme déterministe exécute toujours la même séquence d'operations et produit le même résultat*) :
- ceux basés sur **des algorithmes probabilistes** :(*pour une même donnée d'entrée, l'algorithme choisit la séquence d'opérations à exécuter avec une certaine probabilité et peut produire des résultats différents même si la même donnée est prise plusieurs fois en entrée*)

DEFINITION DES SYSTEMES

- **Système symétique** : un système est dit symétrique si une seule clé est utilisée pour le chiffrement et pour le déchiffrement. On parle dans ce cas **système à clés secrètes**.

- **Système asymétique** : si deux clés différentes sont utilisées, l'une pour le chiffrement, l'autre pour le déchiffrement, on parle de clés asymétriques. Dans ce modèle généralement, l'une des clé est publiée (**clé publique**) et l'autre est gardée par son propriétaire (**clé privée**). Par opposition aux systèmes à clés secrètes, les systèmes asymétriques sont aussi appelés **systèmes à clés publiques**.

- **Système hybride** : c'est l'utilisation des deux systèmes à la fois dans un schéma de communication.

3.2 Mécanismes et services de sécurité

La cryptologie est à la fois une science et une technologie. Science, dont les principes les plus récents sont encore l'occasion de nouvelles découvertes. Technologie, utile et nécessaire dans l'industrie de la sécurité et pour tous ceux qui veulent protéger leur information.

Si le but traditonnel de la cryptographie est d'élaborer des méthodes permettant de transmettre des données de manière confidentielle, **la cryptographie moderne s'attaque plus généralement aux problèmes de sécurité des communications**.

Le but est d'offrir un certain nombres de services de sécurité comme la **confidentialité** des données échangées, l'**authentification** des données transmises et des tiers, l'**intégrité** de l'information échangée et la **non-répudiation** des participants. Pour cela on utilise un certains nombres de **mécanismes** basés sur les **algorithmes cryptographies**.

La cryptologie couvre couramment quatre grandes fonctions de sécurité :

En général, on utilise le **chiffrement** au moyen d'une clé symétrique.

- **Authentification (L'identification)** : il s'agit de garantir l'origine d'une information.
  En général, on utilise la **signature numérique** avec un couple de clés dont celle permettant de créer les signatures est gardée secrète, et dont l'autre permettant de vérifier la signature est rendue publique.
  Le courrier électronique, un bon de commande transmis en ligne, un acte administratif peuvent être signés pour prouver leur origine et engager le signataire, à l'identique d'un paraphe sur le papier.
  L'identification : il s'agit de garantir l'identité et la qualité d'une personne qui souhaite accéder à des informations ou à des ressources matérielles.
  En général, on utilise le contrôle d'accès par mot de passe. Pour consulter son courrier électronique, pour se connecter à un ordinateur distant, ou pour entrer dans un lieu protégé, on peut ainsi s'assurer de l'identité du demandeur.

- **Intégrité** : il s'agit de garantir l'intégrité, c'est-à-dire l'absence de modification d'un message ou d'un document. On peut utiliser la **signature numérique** sous sa forme symétrique ou asymétrique, ou encore le **chiffrement**.
  Il est particulièrement important que, dans toute négociation et accord contractuel, on puisse vérifier qu'aucune modification du document électronique n'a été faite.

- **Non-répudiation** : il s'agit de garantir qu'aucun des partenaires d'une transaction ne pourra nier d'y avoir participée.

1. L'authentification n'est pas seulement pris en charge par des techniques de cryptographie et il existe plusieurs variantes et niveaux d'authentifications.

2. Dans un schéma de communication, l'authentification est le service le plus usité et peut être le plus fondamental.

3. **En général, plusieurs types d'authentifications sont couplés pour une sécurité avec plusieurs niveaux (barrières)** : on l'appelle **l'authentification forte ou multifacteurs.**

4. **Il y'a trois facteurs fondamentaux :**

   - **Ce que l'on sait** (mot de passe, phrase secrète, ...)
   - **Ce que l'on a** (clé, token usb, carte magnétique, carte à puce,...)
   - **Ce que l'on est** (Biométrie : empreintes digitales, voix, scanner rétinien, ADN/Chromosomes, ...)

3.3 Notions de cryptanalyse

3.3.1 Définitions

La cryptanalyse a pour principal objet, d'étudier les faiblesses des outils de sécurité produits par la cryptographie dans le but de les corriger ou nuire au système de communica-

Attaquant : Entité [nommée **Charlie**] suceptible d'agir sur un schémas de communication dans le but de nuire c'est à dire :

- de violer la confidentialité des données,
- de détourner et de modifier des données ou de récupérer des informations,
- d'usurper l'identitité de l'un des partenaires de la communication.

3.3.2 Types d'attaques

Il y'a deux classes d'attaques :

1. **Attaques passives** : l'attaquant écoute seulement. Donc, c'est une attaque qui cible la confidentialité.
2. **Attaques actives** : l'attaquant agit sur le système de commnucation pour :
   - détourner et modifier des données ou récupérer des informations,
   - usurper l'identitité de l'un des partenaires de la communication.

   Donc les attaques actives ciblent toutes les fonctions de sécurité : confidentialité, intégrité, authentification-identification et non répudiation.

Pour celà, l'attaquant cherche généralement à mettre à défaut l'une des primitives de cryptographie. Par exemple :

- **pour les algorithmes** : récupérer les clés de chiffrement, déchiffrer les cryptogrammes ou de casser complètement les algorithmes utilisés ;
- **pour les protocoles** : détourner l'objectif d'un protocole, compromettre le déroulement d'un protocole ;
- **pour le hachage** : fabriquer de fausses empreintes,
- **pour les signatures** : falsifier les signatures.

Ces attaques peuvent être dirigées :

- sur les modèles mathématiques utilisés pour fabriquer les primitives de cryptographie ;
- sur l'implémentation matériel et/ou logiciel des pritmitives de cryptographie ;
- sur les entités propriétaires (légitimes) de données ou d'objets cryptographiques (secrets) ;
- sur les acteurs légitimes d'un scénarios de communication ;
- sur la gestion (fabrication, distribution, stockage, tests de validité,...) de données ou d'objets cryptographiques ;

3.3.3 Sécurité d'un chiffrement

Un système de chiffrement est dit sûr si la probabilité d'obtenir une information sur le texte clair ou la clé de déchiffrement à partir du chiffré est presque nulle.

Celà veut dire que :

- toute information qu'on peut tirer du texte clair sera si faible qu'elle ne permettra pas de violer sa confidentialité partielle ou complète ;
- ou que toute information qu'on peut extraire de la clé sera si faible qu'elle ne permettra pas de la reconstituée substentiellement.

Il existe essentiellement deux modèles de sécurité dépendant du système de chiffrement en question :

frement (d'un système symétrique) indépendamment des moyens (calculatoire et de stockage) de l'adversaire. **La modélisation mathématique est basée sur la théroie de l'information formulée par Shanonn**.

- ▶ **Sécurité prouvée :** exhibie des critères pour que la sécurité d'un chiffrement (d'un système à clé publique) dépende de la résolution d'un problème mathématique calculatoirement difficile (reconnu comme tel !). **La modélisation mathématique est basée sur la complexité des algorithmes** qui permet de définir la sécurité calculatoire.

Dans le sécurité prouvé du modèle calculatoire, souvent, la dépence entre la sécurité del'algorithme et la difficulté du problème mathématique associé n'est pas une équivalenceet on se contente de modèles plus faibles tels l'**indistinguabilité qui stipule qu'on ne peut distinguer deux chiffrés distincts d'un même texte clair et celà necessite au moins un algorithme probabiliste**.

3.3.4 Taille des données

Un PC à $1GHz = 10^9 Hz$ effectue 10^9 opérations élémentaires par seconde.
Opérations élémentaires : affectation, instructions de contrôle, calcul binaire,...

PUISSANCE DE CALCUL DES MACHINES

Temps	Nbre operations / 1 PC	Nbre operat / 10^{18} PC
1s	10^9	...
1 an	$3,1.10^{16}$	$3,1.10^{34}$
1000 ans	$3,1.10^{19}$	$3,1.10^{37}$
10^9 ans	...	...
15.10^9 ans	$46,5.10^{25}$	$46,5.10^{43}$

- La vitesse de la lumière est de $300000kms/s = 3 \times 10^8 m/s$ donc elle traverse une pièce de 3 métres de largeur en un dix milliardième de seconde. Pendant ce temps, un PC à $1GHz$ peut effectuer 10 opérations élémentaires !.
- Un PC à $1GHz = 10^9 Hz$ effectue 10^9 opérations ou instructions élémentaires par seconde. Combien de temps faut-il pour qu'il puisse casser une clé de taille m par force brute ?
- Pour cela la machine doit tester toutes les 2^m clés possibles ! On suppose que le PC peut tester une clé par instruction c'est à dire 10^9 clés par seconde. Comme $10^9 = 2^{30}$ et $1an = 31536000s \cong 2^{25}s$, alors le PC peut tester 2^{55} clés par an d'où on a le tableau suivant :

SECURITE SUR LA TAILLE DES DONNEES "SECRETS"

taille m	Temps pour 1 PC	Temps pour $10^{18} = 2^{60}$ PCs
56	$2^{26}s = 2ans$	$2^{-59}s$=...
64	$2^{34}s = 2^9 ans = 512ans$	$... = 2^{-51}ans$
128	$2^{98}s = 2^{73}ans$	$2^{13}ans =$ **8192ans**
256	$2^{226} = 2^{201}ans$	$2^{141}ans \gg$ **age Univers**
1024	...	...
2048	$2^{2018} = ...$	$2^{1993} = ...$

▶ La taille des clés, des données chiffrées et des valeurs aléatoires secrètes doivent au moins être de taille **128**

▶ S'il n'y a pas d'autres attaques à prendre en compte en dehors de l'attaque par force brute, il n'y a pas de raison de choisir des des données sensibles de taille supérieur à **256**

SECURITE DES MOTS DE PASSE

– **Mots de passe de 8 caractères**

Alphabet	Alphabet et chiffre	Alphanumerique
$26^8 \sim 2^{38}$	$36^8 \sim 2^{42}$	$256^8 \sim 2^{64}$

– **Mots de passe de 22 caractères**

Alphabet	Alphabet et chiffre	Alphanumerique
$26^{22} \sim 2^{104}$	$36^{22} \sim 2^{118}$	$256^{22} \sim 2^{176}$

Ainsi :

▶ Un mot de passe qui protège des données pas trop précieuse doit au moins être de 8 caractères alphanumériques.

▶ Un mot de passe qui joue le role d'une clé de chiffrement doit au moins etre de 22 caractères alphanumériques.

Chapitre 4

CRYPTOGRAPHIE A CLES SECRETES

4.1 Notions de systèmes symétriques

Un système symétrique est un système construit avec une fonction ou un processus facilement reversible. Généralement on entend par systèmes symétriques les systèmes de chiffrement à clés sécrètes.

Dans un système symétrique, la clé secrète doit être partagée entre les entités en communication d'où la nécessité d'avoir un canal sûr.

Canal sûr :

- Se rencontrer, utiliser la valise diplomatique (avant 1976)
- Utiliser des techniques de cryptographie non symétrique (théoriquement depuis 1976)

Les algorithmes de chiffrements symétriques sont utilisés pour rendre le service de confidentialité et sont composés de deux catégories :

1. les algorithmes de chiffrements par blocs :
   DES, 3DES, Lucifer, FEAL, Blowfish, IDEA, AES

2. les algorithmes de chiffrements par flux :
   RC4, RC5, A5, ORYX, SEAL

4.2 Techniques et outils de base

Les systèmes symétriques utilisent plusieurs techniques ou outils dont les plus essentielles sont :

- **Permutation (Transposition)**

- ▶ **Chiffrement de Vernam ou One Time Pad (1918)**
- ▶ **Chiffrement par blocs**
- ▶ **Chiffrement sequentiel ou par flux (ou flot)**

4.2.1 Permutation

La permutation utilise généralement :
- la bijection d'un ensemble donnée ;
- le changer l'ordre des symboles dans un messages ;

PRINCIPE
- ▶ Si M est un message et π une permutation, pour caculer la transformé de M par π, on le décompose en morceaux de longueurs égales (**découpage en blocs**) à la longueur de π.
- ▶ Si le dernier morceau est incomplet (longueur plus courte que celle de π) on définit une procédure publique qui permet de faire du padding (= **compléter le bloc incomplet**).
- ▶ On calcule l'image de chaque morceau puis on juxtapose (**concaténation des blocs**) les résultats dans l'ordre du découpage.

NB La concaténation de A et B est noté habituellement $A \parallel B$.
EXEMPLE :

Si M=**Mon premier cours de crypto**" et $\pi = 102538674$,

on découpe $M = monpremie\ rcoursdec\ ryptozzzz$

puis on calcule les images des blocs

et enfin on concatène les résultats $\pi(M) = \pi(monpremie)\ \pi(rcoursdec)\ \pi(ryptozzzz)$

$\pi(M)$ =**OMNEPEMIR CROSUCDER YRPZTZZZO**

4.2.2 Substitution

PRINCIPE
- ▶ permutation sur l'ensemble des cas possibles appliquées à un nombre fini de cas ;
- ▶ remplacer chaque élément du texte clair (symbole, groupes de symboles) par un autre élément du texte clair (=message) ;

NB Parfois on utilise des substitutions qui ne sont pas reversibles.
EXEMPLE
- S_k = décaler les lettres de k rangs vers la droite dans l'ordre alphabétique".

  En numérotant les lettres de l'alphabet latins de 0 à 25 on voit que S_k est la permutation (globale)
  $S_k : \frac{\mathbb{Z}}{26\mathbb{Z}} :\to \frac{\mathbb{Z}}{26\mathbb{Z}} : x :\mapsto x + k \mod 26.$

bétique.

S_3=**chiffrement de CESAR**

– $S_{k_1,k_2,...,k_m} : (\frac{\mathbb{Z}}{26\mathbb{Z}})^m :\to (\frac{\mathbb{Z}}{26\mathbb{Z}})^m : (u_1,...,u_m) :\mapsto (u+k_1 \mod 26,...,u_m+k_m \mod 26)$ est une **substitution polyalphabétique**.

Dans ce cas, le message est divisé en blocs de longueur m

Si $S_{k_1,k_2,...,k_m}$ est fixé, le m-uplet $(k_1,...,k_m)$ est le mot de passe ou clé ;

$S_{k_1,k_2,...,k_m}$= **chiffrement de Vigenaire avec un mot de passe de longueur** m

4.2.3 Opérateur XOR

PRINCIPE

▶ $P \oplus Q$ est vrai si P **ou bien** Q est vrai. Si on pose $1 = vrai$ et $0 = faux$, on a : $P \oplus Q = Q \oplus P$, $P \oplus P = 0$ et $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$

– on a $0 \oplus 0 = 0$, $0 \oplus 1 = 0$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$.

▶ Le complément de a est $\overline{a} = 1 + a$. Donc $\overline{0} = 1$ et $\overline{1} = 0$: exemple : $\overline{1101011100} = 0010100011$

– Si on a deux chaines binaires de même longueur a et b on peut les additionner bit à bit, et on retrouve les proprités : $a \oplus b = b \oplus a$, $a \oplus a = 0$, $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ et $(a \oplus b) \oplus b = a$

▶ Si M est une chaine binaire $|M|$ représente la longueur de M. Par exemple on a $|011001000110001| = 15$.

4.2.4 Chiffrement de Vernam - Mauborgne - Vigenaire

PRINCIPE

▶ Chiffrer un message M **suffisamment long** avec une **clé parfaitement aléatoire** K **aussi longue que** M ;

▶ **Changer de clé à chaque chiffrement** (clé à usage unique= one time pad) ;

▶ $|M| = |K| \geq 160$; chiffrement : $M \oplus K = M'$; déchiffrement : $M' \oplus K = M$ donc celui qui chiffre et celui qui déchiffre effectue la même opération.

EXEMPLE

texte claire M	100011010111100011010111
clé K	010011110101010011110101
chiffrement $M' = M \oplus K$	110000100010110000100010
clé K	010011110101010011110101
Déchifrement $M' \oplus K$	100011010111100011010111

4.2.5 Algorithme de chiffrement par blocs (Cipher block)

PRINCIPE

Dans un chiffrement par bloc, le message est divisé en blocs de longueur égale :

- Si $\mathcal{C}_K$ est la fonction de chiffrement, on calcule $M'_j = \mathcal{C}_k(M_j)$ pour tout j.
- Si $\mathcal{D}_K$ est la fonction de déchiffrement, on calcule $\mathcal{D}_k(M'_j) = M_j$ pour tout j
- Il y'a plusieurs façons de combiner les différents blocs chiffrés.

  Par exemple, on peut chiffrer **blocs par blocs** ou **utiliser un bloc chiffré dans le suivant**.
- Généralement, **une fonction de chiffrement par bloc** contient une sous-fonction (principale) qui est itérée plusieurs fois (**chaque itération est appelée tour**) pour créer une **situation de surchiffrement dans le but d'augmenter la sécurité**.
- S'il y'a r tours à réaliser, il faut r sous clés $K_j, 1 \leq j \leq r$ de taille l qui sont dérivées de la clé de chiffrement K, via un **algorithme de génération de sous clés** .
- Si $\overline{\mathcal{C}}_k$ est la sous fonction principale de chiffrement, on calcule par surchiffrement :

  $T' \circ \overline{\mathcal{C}}_{K_r} \circ \overline{\mathcal{C}}_{K_{r-1}} .... \circ \overline{\mathcal{C}}_{K_2} \circ \overline{\mathcal{C}}_{K_1} \circ T(M_j)$ où T et T' sont deux fonctions qu'on applique respectivement au début et à la fin (ce choix T et T', dépend des algorithmes).

DEFINITION : chiffrement produit, chiffrement itératif

- On appelle **chiffrement produit** un chiffrement par blocs qui combine plusieurs transformations élémentairement (**substitutions, transpositions, opérations linéaires ou arithmétiques**)
- Un **chiffrement itératif** résulte de l'application itéré d'un chiffrement (en général un chiffrement produit).

4.2.6 Algorithme de chiffrement par flux (Stream cipher)

PRINCIPE

Le **chiffrement par flux (flot)** se fait sequentiellement en générant une clé aussi longue que le message à chiffrer. Chaque morceau (bit ou byte=octect=8bits) de la clé est composée via la fonction de chiffrement avec la portion de clé correspondante.

- Donc si le message est $M = m_1||m_2||...||m_{l-1}m_l$ et la clé est $K = k_1||k_2||...||k_{l-1}k_l$ alors le chiffrement se fait par morceaux : $c_i = \mathcal{C}_{k_i}(m_i)$ (où $\mathcal{C}_{k_i}$ est la fonction de chiffrement) et le déchiffrement se fait par morceaux : $d_i = \mathcal{D}_{k_i}(c_i) = m_i$ (où $\mathcal{D}_{k_i}$ est la fonction de déchiffrement).
- Par exemple : $c_i = m_i \oplus k_i$ et $d_i = c_i \oplus k_i = m_i$.

– Ainsi, le chiffrement de Vernam est un exemple de chiffrement à la fois de stream cipher et de bloc cipher.

- Le chiffrement par flux est adapté à des modes transmission où le message arrive morceaux par morceaux et si les équipements utilisés ont peu de ressource mémoire ou nécessite une transmission rapide par exemple : chiffrements en ligne , qui sont utilisées en particulier par les armées (sécurité), pour la téléphonie mobile (rapidité) GSM et son réseau (systéme $A51$: algorithme de chiffrement), etc..
- Il y a un autre avantage sur les chiffrements par flux en ce sens que si une erreur se produit sur m_i ou k_i alors cette erreur n'est pas propagée ; elle n'affecte que c_i.

ralement couplé avec le chiffrement.

4.3 Avantages et Inconvénients

AVANTAGES

- **Les algorithmes symétriques sont rapides** (parce qu'ils utilisent de petits entiers et des opérations rapides) ;
- En général, il semble que **les algorithmes symétriques sont plus faciles à fabriquer** (plus nombreux !) ;
- Le seul algorithme dont la sécurité est prouvée est un algorithme symétrique à savoir le chiffrement de **Vernam** ;

INCONVENIENTS

- **Confidentialité de la clé sécrète :** problème de partage de la clé à travers un canal sûr et problème de stockage de la clé ;
- **Durée de vie des clés assez courte ;**
- Peut de service de sécurité sont pris en charge par les systèmes symétriques par exemple : on ne peut déterminer qui entre les deux interlocuteurs légitimes, a chiffré un message ;
- **Distribution des clés :** si n personnes communiquent 2 à 2, il faut $C_n^2 = \frac{n(n-1)}{2}$ clés. Pour $n = 1000$ alors $C_n^2 = 499500$.
- En 1973, la **NBS** (National Bureau of Standard) des USA devenu le **NIST** (National Institut of Standard and Technology) a lancé un appel d'offre international pour un algorithme (méthode de chiffrement) donnant lieu à un niveau de sécurité élevé.
- **IBM** proposa **Lucifer** un algorithme développé au début des années 1970, qui fut évalué par la **NSA** (National Security Agency) des USA et publié en 1976 sous le nom de **DES** (Data Encryption Standard). Comme standard $ANSIX3.92$, le DES est proposé en 1974, publié dans le **Federal Register** en 1975, puis adopté comme standard en 1997 (FIPS-46)

4.4 TP sur quelques chiffrements symétriques

Chapitre 5

CRYPTOGRAPHIE A CLES PUBLIQUES

5.1 Introduction

Jusqu'à la fin des années soixante-dix, la cryptologie ne connaissait que les systèmes que l'on appelle maintenant "à clé symétrique". C'est le cas par exemple de l'algorithme DES.

Mais, en novembre 1976, W. Diffie et M.E. Hellman ont émis l'idée de systèmes à clé non-symétrique [1]. Il s'agissait là d'une révolution conceptuelle, dont l'exemple le plus connu est l'algorithme RSA, du nom de ses auteurs Rivest, Shamir et Adleman [3]. Dans ces systèmes, comme le nom l'indique, les clés de chiffrement et de déchiffrement sont différentes. Plus précisément, la connaissance de l'une ne doit pas permettre en pratique de retrouver l'autre.

Ces systèmes sont aussi appelés "à clé publique", une des deux clés pouvant être publiée sans nuire au secret de l'autre. Avec un tel système, n'importe qui peut envoyer à A un message chiffré. Il suffit pour cela d'utiliser la clé publique de A. Seul ce dernier, ayant sa clé privée, aura la capacité de le déchiffrer.

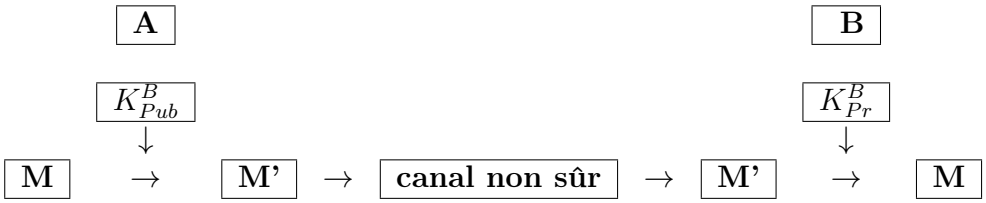
5.2 Notions de systèmes non-systèmes

Le système non symétrique peut être résumé comme suit :

- $k \neq k'$ et l'une des clé (soit k) est difficile à calculer à partir de k' ;
- k est alors publiée et est appelée clé publique, k' est gardée secrète par le propriétaire et est appelée clé privée.
- Il n'y a pas de necessité de canal sûr pour échanger les clés mais, il faut que les utilisateurs puissent s'assurer de l'authenticité des clés publiques.
- Ainsi, il est necessaire d'avoir un tiers de confiance (autorite de certification capable de certifier la validité d'une clé publique d'une entité bien identifiée) ou une chaine de confiance comme dans le cas de l'utilisation de GnuPG.
  Comme exemples d'algorithme à clé publique, on a : RSA, Mc-Eliece, El Gamal etc

5.2.1 Schéma d'un système à clé publique

En cryptographie à clé publique, c'est le destinaire du cryptogramme qui crée sa paire de clé publque/privée. Si Bob crée une paire de clé (K^B_{Pub}, K^B_{Pr}), Alice peut lui envoyer un



5.2.2 Problèmes des systèmes à clé publique

Pour les systèmes à clé publique, le problème du canal sûr de communication ne se pose pas, mais néanmoins deux autres problèmes sont soulevés.

- **Confidentialité de la clé privée** (gardée sur un ordinateur ou une clé USB appelée Token) ;
- **Intégrité de la clé publique** (publiée dans ce qu'on appel un annuaire) ;

Pour garantir l'intégrité de la clé publique de Bob, les différentes parties en communication conviennent de s'en référer à un Tiers de Confiance appelé **Autorité de Confiance** (AC) qui, si elle est sollicitée peut garantir si une clé donnée appartient bien à Bob et si elle n'est pas compromise.

L'utilisation des systèmes à clé publique :

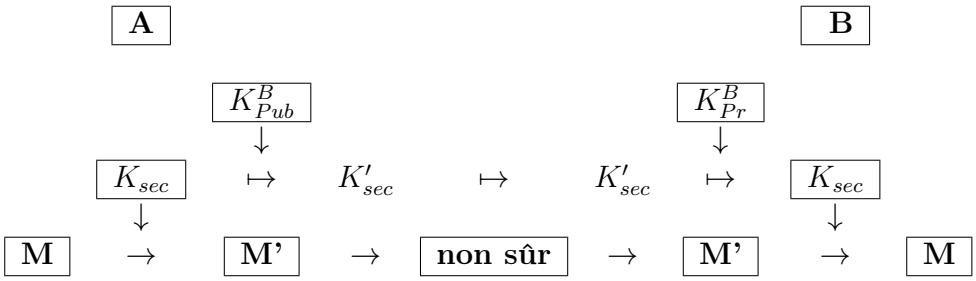
- **à grande échelle**, nécessite des systèmes complexes appelés PKI (Public Key Infrastructure).
- **à petite echelle**, nécessite une chaine de confiance à défaut d'un PKI : c'est le cas de l'Utilisation de Gnu-PG (voir TP) pour le chiffrement des mails sur internet
- Les algorithmes à clé publique utilisent des fonctions à sens unique (voir point suivant) dont la sécurité repose sur l'existence de problèmes difficiles comme la factorisation.
- Tous les algorithmes qui résolvent ces problèmes difficiles sont au moins **sous-exeponentiels** et on montre que les puissances de calculs actuels et les attaques connues imposent de choisir des tailles de clés de plus de 1000 bits (Par exemple le module **RSA** est à 2048 bits actuellement).
- L'utilisation de nombres très grands par les algorithmes à clés publiques ainsi que d'opérations couteuses comme l'exponentiation font ques ces derniers ne sont pas adaptés au chiffrement de données de grande taille.

Ainsi les algorithmes à clé publique sont essentiellement utilisés :

- **pour chiffrer des clés symétriques** qui à leur tour seront utilisées pour chiffrer les données pour la **confidentialité** (système hybride : voir point suivant) ;
- **pour signer des messages** pour garantir la **non répudiation** ;
- **pour construire des protocoles (authentification, identification, partage de secrets,...)** ;

Si Alice veut envoyer un message chiffré à Bob, elle génère une clé secrète K_{sec} et l'utilise pour chiffrer le message clair M en M' puis chiffre la clé K_{sec} en K'_{sec} en utilisant la clé publique de Bob K^B_{Pub}. Enfin elle envoie le couple (M', K'_{sec}) à Bob à travers un canal non nécessairement sûr.

A l'arrivée, Bob déchiffre K'_{sec} en K_{sec} en utilisant sa clé privée K^B_{Pr}, puis déchiffre M' en M en utilisant K_{sec}



5.3 Protocoles cryptographique

5.3.1 Définition

- De façon générale, un protocole est série finie d'étapes conçu pour accomplir une tâche avec au moins deux partenaires.
- Un protocole cryptographique est une suite de règles déterminant l'ensemble des opérations cryptographiques nécessaires et leur séquence pour sécuriser une communication (une transaction, un échange de données, ..) entre plusieurs entités.
- Un protocole cryptographique doit permettre à des personnes qui ne se font pas confiance en général d'échanger en toute sécurité et en présence de personnes malveillantes. Il doit donc empêcher l'espionnage et la tricherie sous toutes leurs formes.
- **Un protocole de cryptographie doit avoir un objectif de securité bien précis : identification, échange de clés, authentification, preuve de connaissance etc.**

5.3.2 Echange de clé Dieffe-Hellman

Pour que Alice et Bob échangent une clé, ils s'entendent d'abord sur un groupe G (noté multiplicative) puis effectuent les étapes suivantes.

- Alice et bob choisissent un élément g d'ordre premier suffisamment grand dans G ;
- Alice choisie un parametre secret (valeur aléatoire $a < p$) et calcule g^a dans G et transmet publiquement g^a à Bob ;
- Bob choisie un parametre secret (valeur aléatoire $b < p$) et calcule g^b dans G et transmet publiquement g^b à Bob ;
- Chacun deux cacule la valeur commune $k = (g^b)^a = (g^a)^b$ qui constituera leur clé privé ou comme outil pour fabriquer la clé privée.

calculer $k = (g^b)^a = (g^a)^b$. Ce problème est basé sur le logerithme discret mais reste moins difficle en théorie.

Cette d'échange de clés Diffe-hellmann a connu depuis lors différentes généralisations.

5.4 TP sur les Algorithmes à clé publique

Nous présentons ici les deux cryptosystèmes asymétriques les plus classiques : RSA [3] et Elgamal [2] sur des corps premiers. Ces deux systèmes ont l'avantage d'être très simples à décrire, du moins dans leur version mathématique.

5.4.1 RSA

Du nom de ses inventeurs (Rivest, Shamir et Adleman), le cryptosystème RSA [3] est le système asymétrique le plus utilisé au monde. Il s'appuie sur la difficulté de la factorisation d'entiers.

Algorithme de génération des clés

- On choisit un entier n(modulo) tel que $n = pq$ avec p et q deux nombres premiers assez grands ($|p| = |q| \geq 1024\ bits$) ;
- On calcule $\varphi(n) = (p-1)(q-1)$. φ est l'indicateur d'Euler et $\varphi(n)$ le nombre d'éléments inversibles de $\frac{\mathbb{Z}}{n\mathbb{Z}}$ (c'est à dire le cardinal du groupe $(\frac{\mathbb{Z}}{n\mathbb{Z}})^*$) ;
- On choisit e tel qu'il soit plus petit que $\varphi(n)$ et $\text{pgcd}(e, \varphi(n)) = 1$;
- On calcule d tel que $ed = 1 \mod \varphi(n)$ (avec l'algorithme étendu d'Euclide).
- Clé publique (e, n) : (généralement $e = 2^{16} + 1$ est fixé) ;
- Clé privée (d, n). p, q et $\varphi(n)$ doivent rester secrets (on peut détruire $\varphi(n)$) .

Algorithme de chiffrement

▲ Soit $M \in \frac{\mathbb{Z}}{n\mathbb{Z}}$ un message, alors le chiffré est : $C = M^e \mod n$;

Algorithme de déchiffrement

- On déchiffre un message chiffré $C \in \frac{\mathbb{Z}}{n\mathbb{Z}}$ par : $C^d \mod n = M$, (si le déchiffrement est valide) ;
- on utilise l'algorithme de calcul de puissance rapide car d est très grand.

5.4.2 ElGamal

Le chiffrement d'El Gamal est un système de cryptage à clé publique inventé en 1984 et utilisé pour chiffrer mais aussi pour signer des messages. Ce système est assez peu utilisé en tant qu'algorithme de chiffrement pur, mais plutôt dans les systèmes de signatures ; il est d'ailleurs à l'origine du DSS (Digital Signature Standard), devenu une norme fédérale en 1994. Ce chiffrement tire son nom, comme RSA, du nom de son créateur.

Nous présentons El Gamal dans un groupe G quelconque.

Algorithme de génération des clés

- Choisir G un groupe et $g \in G$ d'ordre q, ($n = \#G \geq 1024,\ |q| \geq 160,\ q\ divise\ n$) ;
- Alice choisit a aléatoire dans $]1, n-1[$, et calcule $h = g^a$ dans G, ($|a| = |q|$) ;
- La clé publique est (g, h, G) ;
- La clé secrète est a.

Algorithme de chiffrement

- ▶ Choisir un nombre aléatoire $k \in]1, n-1[$ assez grand, $(|a| = |q|)$;
- ▶ Pour chiffrer un message clair m, Bob calcule :
  - ♦ $m_1 = g^k \in G$;
  - ♦ $m_2 = m.h^k \in G$.
- ▶ Chiffré $c = (m_1, m_2)$.

Algorithme de déchiffrement

- ▶ Clé privée a ;
- ▶ Alice reçoit le chiffré $c = (m_1, m_2)$;
- ▶ Elle calcule : $m' = m_1^{p-1-a} \cdot m_2 \in G$.

5.4.3 Nouvelle variante d'El Gamal

Du noms de ses inventeurs (Djiby et Demba) est un nouveau schéma de chiffrement probabiliste et quelques schémas de signatures numériques similaires dans un certain sens, à ceux basés sur le cryptosystème ElGamal.
Ses schémas sont basés sur le problème de logarithme discret mais sont plus sûrs que ceux d'ElGamal dans un certain sens. L'algorithme de génération de clés pour les mécanismes de chiffrement et de signature, proposé, est une légère modification de celui d'ElGamal. Mais, il n'est pas toujours nécessaire de publier le générateur du groupe (ou du sous-groupe) et son ordre ; d'où, on peut utiliser ce fait pour parer à certaines attaques connues et basées sur des générateurs dits faibles.

Algorithme de génération des clés

Pour créer une clé publique, Bob devra faire ce qui suit :

- ▶ Choisir un grand nombre premier aléatoire (ou un strong prime en anglais)p , en effectuant ce qui suit :
  a. Choisir un nombre premier q tel que$2^{159} < q < 2^{160}$;
  b. Choisir t pour $0 \leq t \leq 8$, et choisir un nombre premier p où $2^{511+64t} < p < 2^{512+64t}$, avec la propriété que q divise $(p-1)$;
  c. Choisir un générateur α du groupe cyclique unique d'ordre q dans $\mathbb{Z}_p{}^*$;
     i. Choisir un élément $g \in \mathbb{Z}_p{}^*$ et calculer $\alpha = g^{(p-1)/p} \mod p$;
     ii. Si $\alpha = 1$ alors retourner à l'étape (i).
- ▶ Choisir un élément de groupe $g \in (\frac{\mathbb{Z}}{p\mathbb{Z}})^*$ avec un ordre suffisamment grand : $d = O(g)$ [d'où d divise $p-1$ et si , g est le générateur de $(\frac{\mathbb{Z}}{p\mathbb{Z}})^*$] ;
- ▶ Choisir un nombre aléatoire $2 < k < d$ suffisamment grand et calculer kd ;
- ▶ Choisir un nombre aléatoire $2 < s < d$ suffisamment grand ;
- ▶ Calculer avec l'algorithme de division euclidienne le couple (r, t) tel que $kd = rs + t$ où $t = kd \mod s$ et $r = \lfloor \frac{kd}{s} \rfloor$;
- ▶ Si $t \leq 1$, retourner à l'étape 4 ; [Noter que $r = \lfloor \frac{kd}{s} \rfloor$ et $\lfloor \frac{s}{t} \rfloor$ doivent être suffisamment grands] ;
- ▶ Calculer $\gamma = g^s \mod p$ et $\delta = g^t \mod p$ [Noter que $\gamma \neq 1$ et $\delta \neq 1$] ;
- ▶ La clé publique de Bob est $((\gamma, \delta), p)$ et la clé privée de Bob est $((r, s, t), p)$;

Algorithme de chiffrement

Pour chiffrer un message pour Bob, Alice devra faire comme suit :

- ▶ Prendre $((\gamma, \delta), p)$, la clé publique de Bob ;

$\lambda = \delta^{\alpha} \mod p$ et $\lambda = \delta^{\alpha} \mod p$, (d'où $m_1 \neq 1 \mod p$ et $\lambda \neq 1 \mod p$) ;

- Transformer le message m comme un élément de $(\frac{Z}{pZ})^*$;
- Calculer $m_2 = \lambda m$;
- Le texte chiffré est (m_1, m_2) ;

Algorithme de déchiffrement

Pour déchiffrer un message chiffré avec sa clé publique, Bob devra faire comme suit :

- Prendre (r, p), la clé privée de Bob (en premier lieu) ;
- Prendre (m_1, m_2) le texte chiffré ;
- Calculer m_1^r et $m_1^r m_2$ dans $(\frac{\mathbb{Z}}{p\mathbb{Z}})^*$;
- Le texte clair est $m_1^r m_2$;

5.4.4 Rabin

Le cryptosystème de Rabin est un cryptosystème asymétrique basé sur la difficulté du problème de la factorisation (comme RSA). Il a été inventé en 1979 par Michael Rabin : c'est le premier cryptosystème asymétrique dont la sécurité se réduit à l'intractabilité de la factorisation d'un nombre entier.

Algorithme de génération des clés

Comme pour tous les algorithmes de cryptographie asymétrique, le cryptosystème de Rabin fait usage d'une clé publique et d'une clé privée. La clé publique est utilisée pour chiffrer et n'est pas secrète, tandis que la clé privée est secrète et ne doit être connue que de son propriétaire : le destinataire du message (afin qu'il soit le seul à pouvoir décrypter). Explicitement, la génération de clés est comme suit.

- Choisisr deux grands nombres premiers, p et q, au hasard ;
- Posons $n = pq$, ce qui fait de n la clé publique. Les nombres premiers p et q constituent la clé privée ;

Pour chiffer, on n'a besoins que de la clé publique, n. Pour déchiffrer, les facteurs de n, p et q, sont nécessaires.

Algorithme de chiffrement

- Choisisr deux grands nombres premiers, p et q, au hasard ;Pour le chiffrement, seulement la clé publique, n, est utilisée. On produit le texte chiffré à partir du texte en clair m comme suit. Le texte chiffré c se détermine comme suit.
  $$c = m^2 \mod n.$$
- Autrement dit, c est le résidu quadratique du carré du texte en clair, pris modulo n. En pratique du chiffrement par bloc est généralement utilisé.

Algorithme de déchiffrement

- Pour déchiffrer, la clé privée est nécessaire. Le processus est comme suit. Les racines carrées
  $$m_p = \sqrt{c} \mod p \text{ et } m_q = \sqrt{c} \mod q \text{ sont calculées.}$$
- L'algorithme d'Euclide étendu permet de calculer y_p et y_q, tels que
  $$y_p \cdot p + y_q \cdot q = 1.$$

Chapitre 6

FONCTIONS DE HACHAGE ET SIGNATURE NUMERIQUE

6.1 Fonctions de hachage

6.1.1 Définition

Une fonction de hachage est une fonction publique $h : \{0,1\}^* \to \{0,1\}^n$ telle que :

- h transforme un message (binaire) de longueur quelconque en un message de longueur fixe ; (**Fonction de Compression**)
- pour tout x, $h(x)$ est facile à calculer ; (**Facilement calculable**)

Les images $h(x)$ sont appelés ***hache*** ou ***empreinte***.

On dira que c'est **fonction de hachage pour la cryptographie** si en plus la fonction a les propriétés suivantes :

- pour presque tout y, il est difficile de trouver x tel que $h(x) = y$; (**Fonction à sens unique sans trappe**)
- Pour presque tout x, il est difficile de trouver x' tel que $h(x) = h(x')$ est faible ; (**Faiblement resistante au collusions**)
- Il est difficile de trouver un couple (x, x') tel que $h(x) = h(x')$ (**Fortement resistante au collusions**).

Pour résumer on retiendra que :

Une fonction de hachage en cryptographie est une fonction publique à sens unique sans trappe (donc facile à calculer et difficile à inverser pour tout le monde) $h : \mathcal{P} \to \mathcal{H}$, **qui transforme un message de longueur quelconque en un message de longueur fixe**.

De plus la probabilité, pour qu'il y ait une collusion doit être faible [c'est-à-dire il doit être difficle de trouver $x \neq x'$ tels que $h(x) = h(x')$] ;

6.1.2 Intégrité

Si x est un message alors pour garantir l'intégrité de x en envoie ou stocke le couple $(x, h(x))$ où $h(x)$ est l'empreinte de x via une fonction de hachage h. Le message est considéré intégre s'il est bien accompagné par son empreinte qu'on peut falsifier.

Les fonctions de hachages comptent deux familles

2. celles n'utilisant pas de clés

Les fonctions de hachage sont utilisés pour :
- l'intégrité
- construire des générateurs aléatoires cryptographiquement sûr ;
- pour la modélisation théorique des fonctions à sens unique tel que le modèle de l'oracle aléatoire.

6.1.3 Algorithemes de hachage

MD : Message Digest,
SHA : Secure Hash Algorithm
SHS : Secure hash Standard
- MD4, Rivest, 1990
- MD5, Rivest, 1992, empreinte sur 128 bits, RFC 1321
- SHA (NIST-1993), FIPS 180, SHA1, empreinte sur 160 bits, FIPS 180-1
- SHS (2001) FIPS 180-2 inclut SHA1 et SHA2 (SHA-256, SHA-384, SHA-512)
- RIPEMD 160

RFC Request For Comment

6.2 Signature numérique

6.2.1 Définition 1

Une signature (digitale-manuelle ou numérique-cryptographique) est un procédé, qui, appliqué à un message, garantit la non répudiation par le signataire et donc réalise les deux objectifs suivants
- identification unique du signataire,
- et preuve d'accord sur le contenu du document.

Elle doit posséder les propriétés suivantes :
- Unique
- Impossible à usurper
- Impossible à répudier par son auteur,
- facile à vérifier par un tiers,
- facile à générer

6.2.2 Modélisation des systèmes de signatures numériques (version1)

Un système de signature est composé d'un quintuplet $(\mathcal{P}, \mathcal{S}, S_{k'}, V_k, \mathcal{K})$ où :
- $\mathcal{P}$ est un ensemble appelé espace des textes clairs ;
- $\mathcal{S}$ est un ensemble appelé espace des signatures ;
- $S_{k'} : \mathcal{P} \to \mathcal{S}$ est une fonction injective dite fonction de signature (non nécessairement bijective) qui dépend d'un paramètre k' appelé clé privée.
- $V_k : \mathcal{P}x\mathcal{S} \to \{vraie, faux\}$ est la fonction de vérification de signature binaire telle que $V_k(m, s) =$ vraie si et seulement si $S_{k'}(m) = s$ (dépendant de la clé publique k) .
- $\mathcal{K}$ l'ensemble des paramètres utilisés est l'espace des clés.

On a vu les 5 points communs entre les signatires manuelle et numérique dans la première définition. Ici on regarde les différences.

Signature manuelle

1. Associé physiquement au document signé ;
2. Identique pour tous les documents venant d'un même signataire ;
3. Habituellment à la dernière page.

Signature numérique

1. Peut être stockée et envoyée indépendamment du document signé ;
2. Fonction du document même si le signataire signe avec la même clé privée
3. Couvre l'entièreté du document (dépend de tout le message)

6.2.4 Exemple de signatures numériques

Une façon simple d'avoir un algorithme de signature est d'utiliser un algorithme de chiffrement à clé publique bijective (comme RSA) :

Pour un chiffrement à clé publique on a :
- $C_k : \mathcal{P} \to \mathcal{C}$ fonction de chiffrement
- $D_{k'} : \mathcal{C} \to \mathcal{P}$ fonction de déchiffrement inverse à gauche de C_k c'est à dire : $D_{k'} \circ C_k(m) = m, \forall m \in \mathcal{P}$).

On peut le tranformer en algo de signature si C_k est aussi l'inverse à gauche de $D_{k'}$ et donc $D_{k'} = C_k^{-1}$.

Dans ce cas, on prend l'algorithme de "déchiffrement" comme étant l'agorithme de signature.

Pour signer m on calcule $D_{k'}(m) = s$ et pour vérifier on calcule $C_k(s)$ et on compare avec m. .

6.2.5 Problème d'intégrité

La signature, telle que présentée jusqu'à présent, ne garantit pas l'intégrité (qu'elle soit manuelle ou numérique).

♦ Dans le cas manuelle c'est l'observation et l'analyse du document qui permet de croire ou de s'assurer que le texte n'a pas été modifié.

♦ Dans le cas numérique, voyons d'abord cet exemple.
- On suppose que la fonction de signature est la bijection réciproque de la fonction de chiffrement : $S_{k'} = D_{k'} = C_k^{-1}$ de Bob,
- On suppose que la fonction de signature est un homomorphisme de semi-group (multiplicatif) : $S_{k'}(m_1 m_2) = S_{k'}(m_1) S_{k'}(m_2)$ c'est-à-dire que la signature du produit est égale au produit des signatures.
- On suppose que Charlie a réussi à faire signer un message m_0 par Bob : $S_{k'}(m_0) = s_0$ Soit k et k' les clés publique et privée respectives de Bob

1. Pour signer m, Bob calcule $S_{k'}(m) = D_{k'}(m) = s$

3. Charlie intercepte (m, s) et calcule $s' = s.s_0$ et $m' = mm_0$.
4. Charlie envoie à Alice (m', s').
5. Alice prend la clé publique de Bob k et calcule $C_k(s')$ et compare avec m'. Mais $C_k(s') = C_k(ss_0) = C_k(s)C_k(s_0) = mm_0 = m'$.
6. **Donc Charlie conclut que le message a été signé par Bob ! Ce qui est faux ! Bob à simplement participer à la signature ! Comme le message a été modifié, il n'est pas intègre (authentique).**

Pour résoudre ce problème, on hache le message avant de le signer pour garantir l'intégrité. La fonction de hachage ne doit pas être homomorphique ! ! !

6.2.6 Définition 2

♦ **Données ajoutées à une unité de données ou transformation cryptographie d'une unité de données, permettant à un destinataire de vérifier la source et l'intégrité de l'unité de données, garantissant la non répudiation et protégeant contre la contrefaçon**

♦ **Une signature numérique doit fournir les services d'authentification (de l'origine des données et de leur intégrité) et de non répudiation (pour le signataire)**

6.2.7 Modélisation des signatures numériques (version 2)

Un système de **signature avec appendice** est composé d'un 6-tuplet $(\mathcal{P}, \mathcal{H}, \mathcal{S}, S_{k'}, V_k, \mathcal{K})$ où :

- $\mathcal{P}$ est un ensemble appelé espace des textes clairs ;
- $\mathcal{S}$ est un ensemble appelé espace des signatures ;
- $h : \mathcal{P} \to \mathcal{H}$ une fonction de hachage,
- $S_{k'} : \mathcal{H} \to \mathcal{S}$ est une fonction injective dite fonction de signature (non nécessairement bijective) qui dépend d'un paramètre k' appelé clé privée.
- $V_k : \mathcal{P}x\mathcal{S} \to \{vraie, faux\}$ est la fonction de vérification de signature binaire telle que $V_k(m, s) =$ vraie si et seulement si $S_{k'}\big(h(m)\big) = s$ (dépendant de la clé publique k).
- $\mathcal{K}$ l'ensemble des paramétres utilisés est l'espace des clés.

Bibliographie

[1] Diffie, W., Hellman, M.E. "*New Directions in Cryptography*". IEEE-IT, 22, n° 6, Nov. 1976.

[2] T. E. Gamal. *A public key cryptosystem and a signature scheme based on discrete logarithms*. IEEE Trans. Inform. Theory, 31 :469-472, 1985.

[3] Rivest, R., Shamir, A., Adleman, L. "*A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*". Communication of the ACM, vol. 21, n° 2, Feb. 1978, pp. 41-54.