Cryptographie - Généralités

1. Cryptologie
2. Cryptographie
3. Cryptanalyse
4. Sécurité
5. Stéganographie
6. Transposition
7. Substitution
8. One time pad
9. Chaînage

1. Cryptologie

La *cryptologie* comprend

- **la *cryptographie* ("secret writing") qui est le *chiffrement* ou *cryptage* de messages en clair et le *déchiffrement* ou *décryptage* de messages codés, connaissant la clé.**
- **la *cryptanalyse* qui est l'art de décrypter des messages codés sans connaître la clé ("code breaking").**

$$E(m) = c, \quad D(c) = m$$

2. Cryptographie

Un *système cryptographique* ou *cryptosystème* est composé d'un algorithme de cryptage (chiffrement) et d'un algorithme de décryptage (déchiffrement).

Types de cryptosystèmes

- **Systèmes à usage restreint: les algorithmes de chiffrement et de déchiffrement sont secrets. La sécurité repose sur leur confidentialité.**
- **Systèmes à usage général: la confidentialité ne repose pas sur l'algorithme, mais sur une *clé*. Tout le monde peut utiliser le système.**

Les cryptosystèmes modernes sont des systèmes à clé.

$$E_K(m) = c, \quad D_{K'}(c) = m$$

Deux classes d'algorithmes:

- **Algorithmes *à clé secrète* ou algorithmes *symétriques*.**
- **Algorithmes *à clé publique*.**

- ***Algorithmes à clé secrète***
  La clé de décryptage se calcule (simplement) à partir de la clé de cryptage et vice-versa; équivaut à une même clé.

  Deux catégories:

  - ***stream ciphers*: travaillent bit à bit ou octet à octet. Exemple : xor, one time pad.**
  - ***block ciphers*: travaillent sur des groupes de bits (par ex. 64 bits, DES).**

- ***Algorithmes à clé publique***
  La clé de cryptage est publique, la clé de décryptage est secrète et non calculable en temps raisonnable.

Objectifs

Confidentialité **: le message crypté doit rester secret. ne peut être décrypté par un tiers.**

Authentification **: assurance de l'authenticité, notamment de l'expéditeur ou de l'origine.**

Intégrité **: assurance que le message n'a pas été modifié durant la transmission.**

Nonrépudiation **: l'expéditeur ne peut pas nier, ultérieurement, avoir envoyé le message.**

Techniques

Signature **: un moyen d'associer l'expéditeur à un message.**

Certificat **: attestation qui corrobore (confirme) une affirmation.**

Tiers de confiance **: c'est lui qui délivre les certificats.**

Estampillage **: apposition de dates ou autres signes assurant l'unicité du message.**

3. Cryptanalyse

La *cryptanalyse* est la science du décryptage sans connaissance de la clé. Une tentative d'analyse est une *attaque*.

Une attaque n'est pas générale, mais exploite des situations particulières.

***Attaque à texte chiffré* :**
l'analyste dispose de textes chiffrés $c_1, \ldots, c_n$.

***Attaque à texte clair connu* :**
l'analyste dispose de textes en clair $m_1, \ldots, m_n$ et de leur chiffrement $c_1, \ldots, c_n$.

***Attaque à texte clair choisi* :**

l'analyste peut choisir les textes

$$m_1, \ldots, m_n$$

et obtenir

$$c_1, \ldots, c_n$$

Un cas particulier : l'analyste peut choisir m_i en fonction de $m_1, \ldots, m_{i-1}$ et $c_1, \ldots, c_{i-1}$. *C'est le cas des systèmes à clé publique.*

***Attaque biaisée* :**

l'analyste achète la clé, ou force quelqu'un à la lui fournir. C'est souvent l'attaque la plus efficace.

4. Sécurité

La sécurité mesure la difficulté de casser l'algorithme. Mais cette mesure est relative.
Un algorithme est sûr
- si le coût pour casser l'algorithme est supérieur à la valeur des données cryptées;
- si le temps nécessaire pour casser l'algorithme est supérieur au temps pendant lequel le message doit rester secret.

Un algorithme est *inconditionnellement sûr* si un cryptanalyste ne peut pas trouver le texte en clair, quelle que soit la quantité de messages chiffrés dont il dispose.
Un algorithme est *algorithmiquement sûr* s'il ne peut être cassé avec les ressources disponibles (vague).

La *complexité* d'une attaque est mesurée par la place et le temps requis.

Exemple:

Complexité en temps : $2^{128} = 10^{40}$ opérations.
Moyens : Un million de stations travaillant à un milliard d'opérations par seconde.
Temps requis : Avec 10^{15} opérations par seconde, il faut 10^{25} secondes, c'est-à-dire de l'ordre d'un milliard de milliards d'années.

5. Exemple : Stéganographie

Le message à transmettre est caché à l'intérieur d'un texte beaucoup plus grand.

Exemples historiques : encre invisible, petits points sur des caractères particuliers, grille masquant des portions du texte.

Exemples plus récents : on cache un message dans une image graphique en modifiant le dernier bit de la couleur de chaque pixel. Une version plus efficace est le *water-marking*, qui est une estampille (en principe invisible) ajoutée à une image et qui en certifie l'origine (donc le copyright).

6. Exemple : Transposition

Une *transposition* ou *anagramme* d'un texte

$$t = a_1 \cdots a_n$$

est le texte

$$a_{\tau(1)} \cdots a_{\tau(n)}$$

où τ est une permutation de $\{1, \ldots, n\}$.

La transposition de permutation τ^{-1} redonne le texte original.

Une transposition augmente la "diffusion" dans un texte (les digrammes n'en sont plus).
Un texte long est découpé en blocs transposés séparément.

Exemple: Transposition de colonnes

Soit $n = pq$. On écrit le texte en p lignes de longueur a, puis on lit les colonnes, dans l'ordre donné par la permutation.

3	**2**	4	1
e	n	v	a
h	i	r	m
a	r	n	e

donne

$$amenirehavrn$$

Substitution

Une *substitution* est une permutation σ de l'alphabet. Une substitution doit augmenter la "confusion" d'un texte. Le texte

$$m = a_1 \cdots a_n$$

donne

$$c = sigma(a_1) \cdots \sigma(a_n)$$

La substitution de permutation σ^{-1} redonne le texte original.

Exemple :

- **Code de César**
  $\sigma(a) = a + 1 \bmod 26$, **IBM→HAL**
- `rot13` : $\sigma(a) = a + 13 \bmod 26$

Code de Vigénère

On choisit une clé $k = k_1 \cdots k_p$. On répète la clé jusqu'à obtenir la longueur du message $m = a_1 \cdots a_n$. Le message crypté est:

$$c_i = k_i + a_i \bmod 26$$

ou encore

$$c = k + m \bmod 26$$

Le décodage est par soustraction.

Xor simple

C'est un algorithme de cryptage utilisé dans de nombreux logiciels commerciaux, et autorisé par la NSA pour l'industrie des téléphones cellulaires aux Etats-Unis.

La clé est une chaîne de caractères k et le codage est

$$m \oplus k = c$$

Le décodage est

$$c \oplus k = m$$

Ce système n'est pas sûr.

7. One time pad ou à masque jetable

Données :

- **le message m à crypter.**
- **un *masque* ou *pad* qui est une chaîne de caractères k tirée au hasard, et de même longueur que m**

Cryptage : par addition modulo 26 ou par xor:

$$c = k \oplus m$$

Decryptage : le destinataire possède une copie du masque, et fait l'opération inverse:

$$m = c \oplus k$$

- **Si masque est parfaitement aléatoire, le texte crypté est parfaitement aléatoire: il n'y a pas de moyen d'attaque.**
- **Le destinataire doit connaître le masque.**
- **On doit savoir engendrer des séquences aléatoires.**

8. Chaînage

Donnée : un cryptosystème (E, D) par blocs de taille n. Cela peut être RSA ou DES.

Objectif : envoyer un message m long.

Méthode :

- **on le découpe en segments de taille n:**

$$m = m_1 \cdots m_k$$

- **on crypte chaque segment;**
- **on *chaîne* les segments en les faisant interagir, en clair ou cryptés, pour augmenter la diffusion.**

Quatre modes de chaînage existent:

- **ECB (Electronic codebook)**
- **CBC (Cipher block chaining)**
- **CFB (Cipher feedback)**
- **OFB (Output feedback)**

ECB (Electronic codebook)

C'est l'absence de chaînage:

On calcule

$$c_i = E(m_i)$$

et on envoit

$$c = c_1 \cdots c_k$$

On décrypte par

$$m_i = D(c_i)$$

- **Ce chaînage expose à des attaques statistiques, car deux blocs égaux de la source donnent deux blocs chiffrés égaux.**
- **Peut être vu comme une substitution, l'alphabet étant formé des blocs de taille n.**

CBC (Cipher Block Chaining)

On choisit un premier bloc c_0.

On crypte par

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

et on envoit c_0 et les c_i.

On déchiffre par

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

Les erreurs de transmission sur les c_i ne se propagent pas.

CFB (Cipher Feedback)

- **On prend une taille $t < n$.**
  On découpe le message en blocs de taille t

$$m = m_1 \cdots m_k$$

  Un registre à décalage à n bits initialisés à s_0.

- **Chiffrage par**

$$p_i = E(s_{i-1})_t \qquad (t \text{ bits de gauche})$$
$$c_i = m_i \oplus p_i$$
$$s_i = (2^t s_{i-1} + c_i) \bmod 2^n \quad \text{(décalage et addition)}$$

- **Déchiffrage : changer la formule du milieu en**

$$m_i = c_i \oplus p_i$$

- **NB. Il faut une clé secrète.**

OFB (Output Feedback)

Les CFB et OFB sont utilisés lorsque l'unité de transmission est de taille t ($= 1$ ou $= 8$ souvent) est inférieure à la taille d'un bloc crypté ($n = 64$).

Pour OFB, on a $t \leq n$. Le calcul des s_i est modifié en:

$$s_i = (2^t s_{i-1} + p_i) \bmod 2^n$$

Cas fréquent: $t = n$. Dans ce cas, $p_i = s_i$ et

$$s_i = E(s_{i-1})$$

Variante (*counter mode*) : On prend $s_i = 1 + s_{i-1}$.