

Introduction à la Sécurité

1.1 Objectifs de la sécurité

La sécurité vise à assurer plusieurs propriétés :

La confidentialité : c'est la propriété qui garantit que les informations transmises ne sont compréhensibles que par les entités autorisées.

L'authentification : c'est la propriété qui consiste à vérifier l'identité d'un utilisateur avant de lui donner l'accès à une ressource.

L'intégrité : c'est la propriété qui consiste à vérifier si les informations n'ont pas été modifiées durant la transmission.

La disponibilité : c'est la propriété qui permet de garantir l'accès aux données.

La non-répudiation : c'est la propriété qui permet d'avoir une preuve comme quoi un utilisateur a envoyé (ou reçu) un message particulier. Cette propriété permet d'empêcher l'utilisateur de nier l'envoi (ou réception) du message en question.

1.2 Les scénarios d'attaques

1.2.1 Attaque passive

Dans ce genre d'attaques, les informations ne sont pas modifiées. L'attaquant collecte seulement les informations qui circulent sur le réseau.

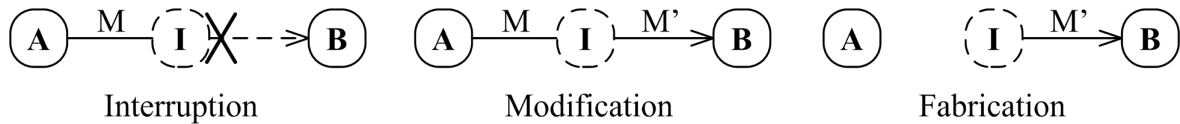
1.2.2 Attaque active

Il y a trois cas possibles pour mener une attaque active :

L'interruption : l'intrus intercepte le message envoyé par l'utilisateur \mathcal{A} pour \mathcal{B} et l'interrompt.

La modification : l'intrus intercepte le message envoyé par l'utilisateur \mathcal{A} et le modifie avant de le faire suivre à l'utilisateur \mathcal{B} .

La fabrication : L'intrus fabrique un message et l'envoie à l'utilisateur \mathcal{B} en se passant pour l'utilisateur \mathcal{A} .



1.3 Concepts de base sur la cryptographie

Chiffrement et déchiffrement : consiste à transformer une donnée afin de la rendre incompréhensible par un utilisateur autre que celui qui l'a créée et celui qui va la recevoir.

Chiffré : c'est le résultat de chiffrement d'une donnée.

Clé : il s'agit d'un paramètre impliqué dans les opérations de chiffrement et de déchiffrement et qui est partagé entre l'émetteur et le récepteur.

Cryptanalyse : elle a pour but de retrouver la donnée en claire à partir d'un ou plusieurs chiffrés sans connaître les clés et/ou l'algorithme de chiffrement.

Canal : moyen de transport de l'information.

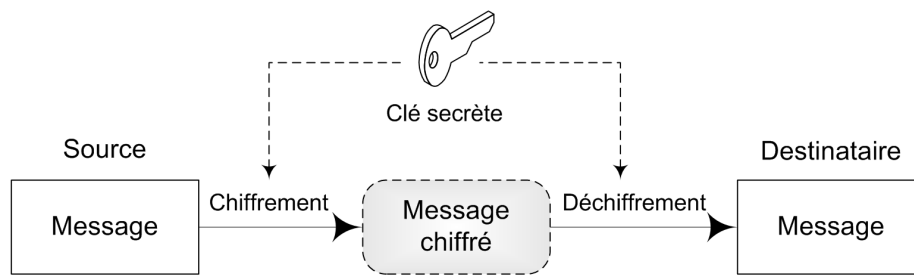
Canal sécurisé : canal où l'intrus n'a pas la possibilité d'altérer les messages.

Canal sécuritaire : canal qui n'est pas physiquement accessible à l'intrus.

1.4 Les algorithmes cryptographiques

1.4.1 Algorithmes à clés symétriques

Dans ce type d'algorithme, la même clé est utilisée à la fois pour le chiffrement et le déchiffrement (DES, AES, IDEA, ... etc.).

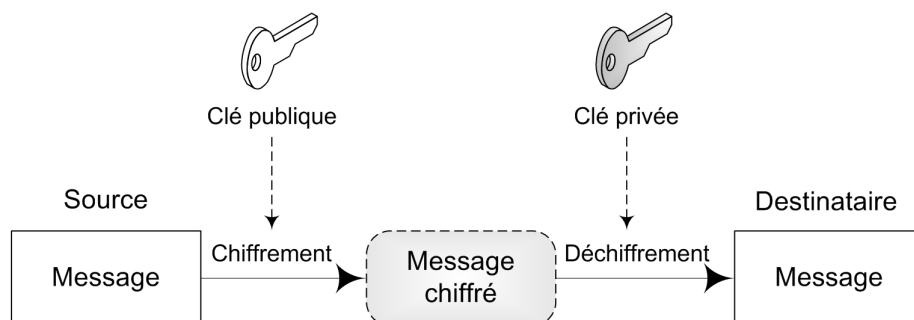


1.4.1.1 Exemple

- ★ Le message $\mathcal{M} = 139$.
- ★ L'opérateur de chiffrement et de déchiffrement : \oplus (XOR).
- ★ La clé $\mathcal{K} = 199$.
- ★ $\mathcal{C} = \mathcal{M} \oplus \mathcal{K} = 139 \oplus 199 = 76$.
- ★ $\mathcal{M} = \mathcal{C} \oplus \mathcal{K} = 76 \oplus 199 = 139$.

1.4.2 Algorithmes à clés asymétriques

Dans ce type d'algorithmes, chaque entité possède une paire de clés : *publique* et *privée*. La clé publique est utilisée pour le chiffrement et la clé privée pour le déchiffrement (RSA, Elgamal, Rabin, Merkel-Hellman, ... etc.).



La signature numérique consiste à générer un condensé à partir d'un message et le chiffrer en utilisant la clé privée de l'émetteur. Ce dernier, ensuite, envoie le message en clair avec la signature. Pour vérifier la validité de cette signature, le récepteur recalcule le condensé à partir du message en clair et déchiffre la signature. Ensuite, il vérifie l'égalité des deux résultats.

1.4.3 Fonctions de hachage

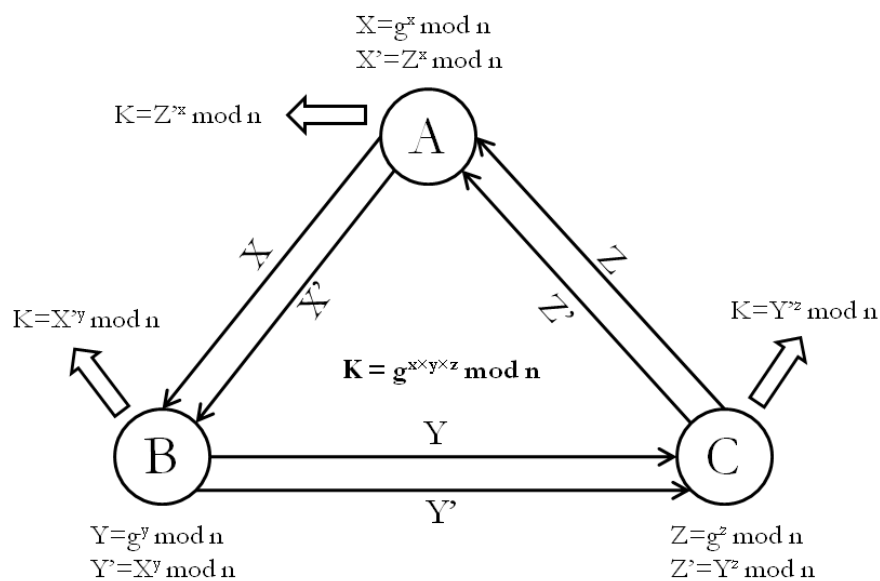
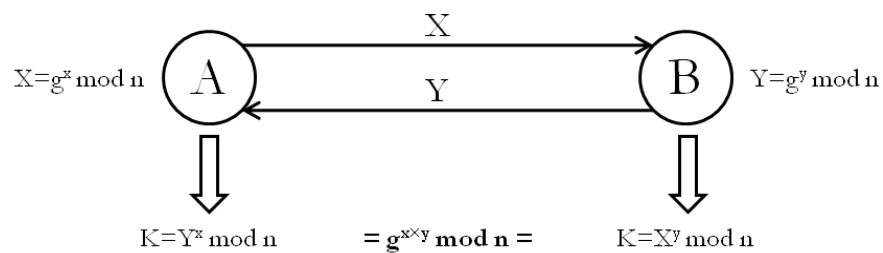
Il s'agit de la troisième famille d'algorithmes cryptographiques. Le principe est qu'un message \mathcal{M} de longueur quelconque est transformé en une valeur h de longueur fixe et inférieure à celle du départ ($h = H(\mathcal{M})$) et qui représente d'une manière unique le message \mathcal{M} . Deux caractéristiques importantes sont les suivantes :

1. A partir de h , il est impossible de retrouver \mathcal{M} .
2. Etant donné $h = H(\mathcal{M})$, il est impossible de trouver \mathcal{M}' tel que $H(\mathcal{M}') = h$.

Algorithmes de Chiffrement à Clés Publiques

2.1 Protocole de Diffie-Hellman

Le protocole d'échange de Diffie-Hellman est une méthode par laquelle deux utilisateurs peuvent partager une clé de chiffrement symétrique. Le protocole tire sa sécurité du problème du Logarithme discret. Il peut s'exécuter avec deux ou plusieurs utilisateurs.



2.2 Chiffrement de RSA

L'algorithme de chiffrement de RSA tire sa sécurité du problème de factorisation des grands nombres entiers. L'algorithme se déroule en trois étapes :

1. *Création de clés* : On choisit deux nombres premiers p et q et on calcule n et $\phi(n)$ tels que $n = p \times q$ et $\phi(n) = (p - 1) \times (q - 1)$. Ensuite, on choisit $e < \phi(n)$ tel que $\text{pgcd}(e, \phi(n)) = 1$, et enfin on calcule d tel que $e \times d \bmod \phi(n) = 1$. La clé publique est (e, n) et la clé privée est (d, n) .
2. *Chiffrement* : Le message à chiffrer doit être strictement inférieur à n . Pour chiffrer un message \mathcal{M} , on calcule $\mathcal{C} = \mathcal{M}^e \bmod n$.
3. *Déchiffrement* : Pour déchiffrer le message, on calcule $\mathcal{M} = \mathcal{C}^d \bmod n$.

2.2.1 Exemple

- ★ $p = 19, q = 23 \Rightarrow n = 19 \times 23 = 437$ et $\phi(n) = (19 - 1) \times (23 - 1) = 396$.
- ★ $e = 17 \Rightarrow 17d \bmod 396 = 1 \Rightarrow d = 233$.
- ★ Pour $\mathcal{M} = 351 \Rightarrow \mathcal{C} = 351^{17} \bmod 437 = 150$.
- ★ $\mathcal{M} = 150^{233} \bmod 437 = 351$.

2.3 Chiffrement de Rabin

L'algorithme de chiffrement de *Rabin* tire sa sécurité du problème de factorisation des grands nombres entiers. L'algorithme se déroule en trois étapes :

1. *Création de clés* : On choisit deux nombres premiers p et q , tels que $p \bmod 4 = 3$ et $q \bmod 4 = 3$. La clé publique est $n = p \times q$ et la clé privée est (p, q) .
2. *Chiffrement* : Le message à chiffrer doit être strictement inférieur à n . Pour chiffrer un message \mathcal{M} , on calcule $\mathcal{C} = \mathcal{M}^2 \bmod n$.
3. *Déchiffrement* : Pour déchiffrer le message, on calcule tout d'abord m_p, m_q, y_p , et y_q tels que $m_p = \mathcal{C}^{(p+1)/4} \bmod p$, $m_q = \mathcal{C}^{(q+1)/4} \bmod q$, et $p \times y_p + q \times y_q = 1$. Ensuite, on calcule \mathcal{R} et \mathcal{S} , tels que $\mathcal{R} = (p \times m_q \times y_p + q \times m_p \times y_q) \bmod n$, et $\mathcal{S} = (p \times m_q \times y_p - q \times m_p \times y_q) \bmod n$. L'une des solutions $\{\mathcal{R}, \mathcal{S}, n - \mathcal{R}, n - \mathcal{S}\}$ est le message \mathcal{M} .

2.3.1 Exemple

- ★ $p = 11, q = 23 \Rightarrow n = 253$.
- ★ Pour $\mathcal{M} = 158 \Rightarrow \mathcal{C} = 158^2 \bmod 253 = 170$.
- ★ $m_p = 170^{(11+1)/4} \bmod 11 = 4$.
- ★ $m_q = 170^{(23+1)/4} \bmod 23 = 3$.
- ★ $11y_p + 23y_q = 1 \Rightarrow y_p = -2$ et $y_q = 1$.
- ★ $\mathcal{R} = (11 \times 3 \times (-2) + 23 \times 4 \times 1) \bmod 253 = 26$.
- ★ $\mathcal{S} = (11 \times 3 \times (-2) - 23 \times 4 \times 1) \bmod 253 = 95$.
- ★ Les solutions possibles sont : $\{26, 95, (253 - 26) = 227, (253 - 95) = \underline{158}\}$.

2.4 Chiffrement de Merkle-Hellman

L'algorithme de chiffrement de *Merkle-Hellman* tire sa sécurité du problème de *sac-à-dos*. Etant données des valeurs entières $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_k$ et un poids \mathcal{T} . Est-il possible de déterminer des valeurs binaires b_1, b_2, \dots, b_k de telle sorte que $\mathcal{T} = b_1\mathcal{P}_1 + b_2\mathcal{P}_2 + \dots + b_k\mathcal{P}_k$? Si la suite des poids \mathcal{P}_i est super-croissante (chaque poids \mathcal{P}_i est strictement supérieur à la somme de tous les poids précédents), alors il existe un algorithme polynomial permettant de déterminer les valeurs de b_i :

```

Pour  $i=k$  à  $1$  Faire
    Si  $T \geq P_i$  Alors  $T \leftarrow T - P_i$ ;
                         $b_i \leftarrow 1$ ;
    Sinon  $b_i \leftarrow 0$ ;
Fin pour
Si  $T = 0$  Alors  $\{b_1, b_2, \dots, b_k\}$  est la solution ;
Sinon Il n'existe pas de solutions ;
  
```

On peut vérifier qu'avec la suite super-croissante $\{2, 3, 6, 12\}$ pour $\mathcal{T} = 15$ on obtient la solution $\{0, 1, 0, 1\}$.

La résolution d'une suite non super-croissante est un problème NP-complet. La sécurité du système de chiffrement *Merkle-Hellman* est basé sur cette difficulté. Il se déroule en trois étapes :

1. *Création de clés* : On choisit une suite super-croissante $\mathcal{A} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_k\}$.

Ensuite, on choisit deux nombres n et m tel que m est supérieur à la somme de tous les \mathcal{P}_i , et n n'a de facteur commun avec aucun nombre de la suite.

On calcule, ensuite, la suite non super-croissante $\mathcal{B} = \{\mathcal{P}'_1, \mathcal{P}'_2, \dots, \mathcal{P}'_k\}$ tel que $\mathcal{P}'_i = n\mathcal{P}_i \bmod m$. La clé publique est \mathcal{B} , et la clé privée est (\mathcal{A}, n, m) .

2. *Chiffrement* : Le message à chiffrer doit être une suite binaire tel que $\mathcal{M} = b_1b_2\dots b_k$. On calcule $\mathcal{C} = b_1\mathcal{P}'_1 + b_2\mathcal{P}'_2 + \dots + b_k\mathcal{P}'_k$.
3. *Déchiffrement* : Pour déchiffrer le message, on calcule tout d'abord n^{-1} tel que $n \times n^{-1} \bmod m = 1$. Ensuite, on calcule $\mathcal{T} = n^{-1} \times \mathcal{C} \bmod m$. Enfin, on calcule les valeurs b_i du message \mathcal{M} en utilisant l'algorithme de résolution d'une suite super-croissante.

2.4.1 Exemple

- ★ $\mathcal{A} = \{2, 3, 6, 13, 27, 52\}$, $n = 31$ et $m = 105 \Rightarrow \mathcal{B} = \{62, 93, 81, 88, 102, 37\}$.
- ★ Pour $\mathcal{M} = 53 = 110101_{(2)} \Rightarrow \mathcal{C} = 1 \times 62 + 1 \times 93 + 0 \times 81 + 1 \times 88 + 0 \times 102 + 1 \times 37 = 280$.
- ★ $31 \times n^{-1} \bmod 105 = 1 \Rightarrow n^{-1} = 61$.
- ★ $\mathcal{T} = (61 \times 280) \bmod 105 = 70 = 1 \times 2 + 1 \times 3 + 0 \times 6 + 1 \times 13 + 0 \times 27 + 1 \times 52 \Rightarrow \mathcal{M} = 110101_{(2)} = 53$.

2.5 Chiffrement d'Elgamal

L'algorithme de chiffrement d'*Elgamal* tire sa sécurité du problème du Logarithme discret qui est un problème NP-complet. Il se déroule en trois étapes :

1. *Création de clés* : On choisit un nombre premier p et deux nombres a et g inférieur à p . Ensuite, on calcule $\mathcal{A} = g^a \bmod p$. La clé publique est (\mathcal{A}, g, p) et la clé privée est a .
2. *Chiffrement* : Le message à chiffrer doit être strictement inférieur à p . Pour chiffrer le message \mathcal{M} , on choisit un nombre aléatoire b strictement inférieur à p et premier avec $p - 1$. Ensuite, on calcule \mathcal{B} et \mathcal{C} , tels que $\mathcal{B} = g^b \bmod p$ et $\mathcal{C} = \mathcal{M} \times \mathcal{A}^b \bmod p$. Le message chiffré est $(\mathcal{B}, \mathcal{C})$.

3. *Déchiffrement* : Pour déchiffrer le message, on calcule $\mathcal{M} = \mathcal{C} \times \mathcal{B}^{(p-a-1)} \bmod p$.

2.5.1 Exemple

- ★ $p = 23, g = 7, a = 6 \Rightarrow \mathcal{A} = 7^6 \bmod 23 = 4$.
- ★ Pour $\mathcal{M} = 7$ et $b = 3 \Rightarrow \mathcal{B} = 7^3 \bmod 23 = 21$ et $\mathcal{C} = (7 \times 4^3) \bmod 23 = 11$.
- ★ $\mathcal{M} = (11 \times 21^{(23-6-1)}) \bmod 23 = 7$.

2.5.2 Signature numérique

Pour calculer la signature numérique d'un message \mathcal{M} , on choisit un nombre aléatoire b strictement inférieur à p et premier avec $p - 1$. Ensuite, on calcule $\mathcal{B} = g^b \bmod p$ et \mathcal{C} tel que $\mathcal{M} = (a \times \mathcal{B} + b \times \mathcal{C}) \bmod (p - 1)$. Pour vérifier la validité de la signature $(\mathcal{M}, \mathcal{B}, \mathcal{C})$, on vérifie l'égalité $\mathcal{A}^{\mathcal{B}} \times \mathcal{B}^{\mathcal{C}} \bmod p = g^{\mathcal{M}} \bmod p$.

2.5.2.1 Exemple

- ★ $p = 23, g = 7, a = 6, \mathcal{A} = 4, \mathcal{M} = 7, b = 3$.
- ★ $\mathcal{B} = 7^3 \bmod 23 = 21$.
- ★ \mathcal{C} tel que $(6 \times 21 + 3\mathcal{C}) \bmod 22 = 7 \Rightarrow \mathcal{C} = 19$.
- ★ Vérification : $4^{21} \times 21^{19} \bmod 23 = 7^7 \bmod 23 = 5$.

