Introduction à la cryptographie

Cryptographie

La **cryptographie** désigne l'ensemble des techniques de protection des messages grâce à leur **chiffrement**, c'est-à-dire la transformation d'un message **clair** (non chiffré) en un **cryptogramme** (message inintelligible si on ne dispose pas de la clé de **déchiffrement**).

En anglais on parle de **plaintext** pour le message en clair et de **ciphertext** pour le cryptogramme.

On appelle aussi **décryptage** (ou décryptement), l'action de déchiffrer un message illégitimement. Si le code de chiffrement (ou la clé) n'est pas connu on parle alors de **cryptanalyse** (fam. **Cassage**).

La cryptographie ne sert pas seulement à garder une **confidentialité** des messages, mais aussi une **authenticité** et une **intégrité**. On l'utilise aussi pour la **non répudiation** de transaction.

Symétrique versus asymétrique

Il existe deux types de clé de chiffrement:

- les clés symétriques. La même clé est utilisée pour le chiffrement et le déchiffrement. Cette clé doit donc être une **clé secrète**.
- les clés asymétriques. Une **clé publique** est utilisée pour le chiffrement, tandis qu'une **clé secrète** est utilisée pour le déchiffrement. On a donc une paire de clé.

Chiffrement symétrique

Le chiffrement symétrique existe depuis les premiers écrits. On peut imaginer beaucoup de chiffrement symétrique. La clé secrète peut être l'algorithme de chiffrement lui-même.

Quelques exemples célèbres de chiffrement symétrique sont le **chiffrement de César**, **la technique assyrienne**, **le chiffre de Vigénère**, et bien d'autres.

Le principe du chiffrement symétrique est simple. Si **A** veut transmettre un message à **B** il procède comme suit: à partir d'un message (**M**) à transmettre **A** utilise un algorithme de chiffrement utilisant une clé secret (**K**). Il transmet alors le message chiffré (**MC)** à **B**. **B** déchiffre alors le message en utilisant la clé **K**.

Voici une liste non exhaustive des algorithmes de chiffrement symétrique les plus utilisés: DES, **3DES**, AES, RC4, RC5.

Les avantages d'un système de chiffrement symétrique sont:

- La rapidité des algorithmes de chiffrement et déchiffrement.

Les désavantages sont:

- La gestion des clés (en effet il faut environ **n * (n-1) / 2** clés pour que n personnes puisse échanger des messages entre elles
- L'échange de la clé secrète

Chiffrement asymétrique

Le chiffrement asymétrique n'existe que depuis quelques dizaines d'années (vers 1970). Il a été crée pour pallier aux désavantage de la gestion et l'échange des clés du chiffrement symétrique.

Le principe est le suivant: si **A** veut transmettre un message **M** à **B,** il commence par demander la clé publique (**PK**) de **B** (à un serveur de clés par exemple, ou directement à **B**). Une fois cette clé obtenue il chiffre le message **M** et l'envoi à **B. B** utilise alors sa clé secrète (**SK**) pour déchiffrer le message.

Les algorithmes de chiffrement asymétrique fonctionnent donc avec une paire de clé. La **clé publique** est générée à partir de la **clé secrète,** et ces deux clés ont la plupart du temps la forme de très grand nombre premiers. Ces clés sont générées à partir de grand nombres aléatoires dont on test la primalité grâce à un algorithme (le test de primalité de Rabin-Miller par exemple).

Voici les deux algorithmes de chiffrement asymétrique les plus utilisés: **RSA** et DSA.

Les avantages d'un système de chiffrement symétrique sont:

- Aucun échange de clé secrète
- Facilité d'accès à la clé publique.

Les désavantages sont:

- Leur lenteur (par apport aux algorithmes de chiffrement symétrique)
- Il faut s'assurer que la clé publique est bien celle du destinataire!

Symétrique plus asymétrique: la bonne solution

Pour obtenir un canal de communication sécurisé entre deux entités il n'est pas souhaitable de n'utiliser que la méthode de chiffrement asymétrique (à cause de sa lenteur).

La solution est donc d'utiliser le chiffrement asymétrique uniquement pour effectuer l'échange de la clé secrète nécessaire au chiffrement symétrique. On nullifie ainsi les désavantages du chiffrement symétrique.

Cela semble donc être la bonne solution. Mais il reste, dans cette méthode, un des désavantages du chiffrement asymétrique, il faut toujours s'assurer que la clé publique est bien celle du destinataire.

Les fonctions de hachage

Une fonction de **hachage** est une fonction à sens unique (**one-way function**) permettant d'obtenir un condensé d'un message (fam. **haché**). Une bonne fonction de hachage doit avoir une probabilité très forte de générer deux hachés différents pour deux messages différents.

Quelques fonctions de hachage souvent utilisées: MD5, SHA-1, SHA-256.

Les utilités d'une fonction de hachage sont multiples. Mais la principale est de garantir l'intégrité d'un message. En effet si **A** envoie à **B** un message **M** ainsi que le haché du message **M** (**MH**), **B** pourra vérifier l'intégrité du message **M** (et de **MH**) en générant a nouveau le haché du message (**MH2**) et en le comparant avec le haché **MH.**

Malheureusement cette méthode ne garantie pas l'authenticité du message **M** (c'est dire qu'on ne peut pas vérifier qu'il a bien été envoyé par **A**).

Asymétrique + hachage = authentification ?

Si l'on rajoute une étape de chiffrement à la méthode de hachage l'on peut obtenir une authentification. En effet si **A** chiffre le haché **MH** grâce à sa clé privé, **B** n'a plus qu'à utiliser la clé publique de **A** pour le déchiffrer et le comparer à **MH2**. On parle alors de **signature** électronique.

Mais cela ne suffit pas, car il faut encore être sûr que la clé publique utilisée par **B** est bien la clé publique de **A** et pas celle d'un éventuel hacker.

Les certificats: la solution

Un certificat est un fichier composé de deux parties. La première comprend diverses informations. On y trouve le nom du propriétaire (ou de l'organisation), sa clé publique, et les dates de validité de cette clé.

La deuxième partie du certificat contient le haché de la première partie, signé par une autorité de certification (**CA** pour Certification Authority).

Si **A** veut communiquer avec **B**, il lui suffit de se procurer le certificat de **B**. Pour vérifier qu'il s'agit bien du certificat de **B** (et donc qu'il contient bien sa clé publique), il déchiffre la signature électronique du certificat (la deuxième partie), grâce a la clé publique de l'autorité de certification et compare le résultat obtenu avec le haché de la première partie. Si les deux hachés correspondent, il s'agit bien du certificat de **B**.

Cela résolve le dernier désavantage du chiffrement asymétrique, en effet il est maintenant facile d'obtenir la clé publique d'une personne grâce à son certificat.

SSL: la synthèse

SSL regroupe toutes les méthodes citées plus haut. En effet voici son fonctionnement:

1) Le client (le browser) qui se connecte à un site sécurisé par SSL fourni au serveur la liste des algorithmes de chiffrement qu'il connaît et demande le certificat du serveur.

2) Le serveur lui renvois alors le certificat signé par une autorité de certification, ainsi que la méthode de chiffrement symétrique à utiliser.

3) Le client vérifie l'intégrité et l'authenticité du certificat. Puis il génère une clé secrète, chiffre cette clé avec la clé publique du serveur contenu dans le certificat et lui envoi cette clé (nommée clé de session).

4) Le serveur déchiffre la clé de session avec sa clé privée. Les deux entités vont désormais utiliser cette clé de session et la méthode de chiffrement symétrique pour communiquer sur le canal sécurisé ainsi crée.