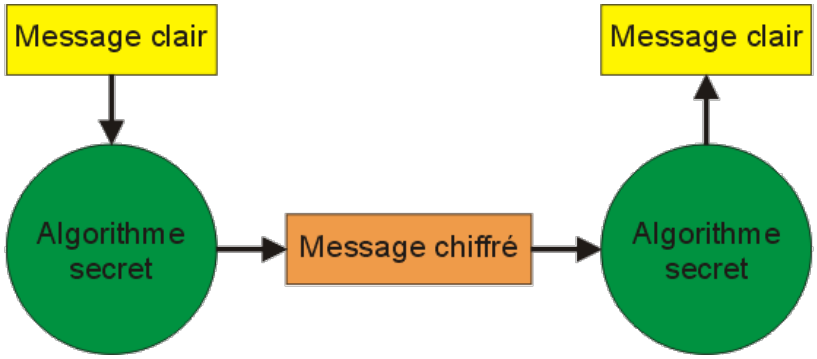
13 : Cryptographie pour la Cybersécurité

I - Introduction

La **cryptographie** consiste à chiffrer des données pour les rendre **confidentielles**. C'est la base de tout échange d'informations **sécurisé**.

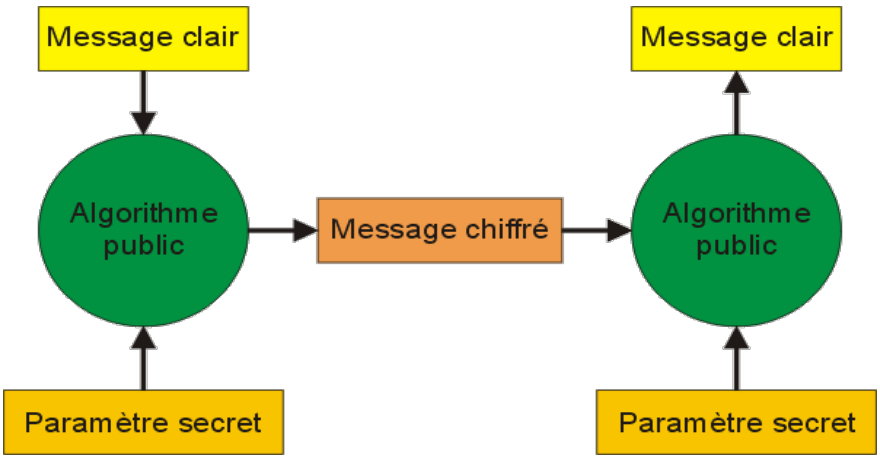
II - Sécurisation d'un échange d'informations avec un algorithme secret

Ici on dispose d'un **algorithme de chiffrement secret** qui assure à lui seul la confidentialité du message. Un tel procédé, cependant, n'est pas considéré comme sûr, si quelqu'un réussis à reconstituer l'algorithme alors il n'y aura plus de secret.



III - Sécurisation d'un échange d'informations avec un algorithme et une clé

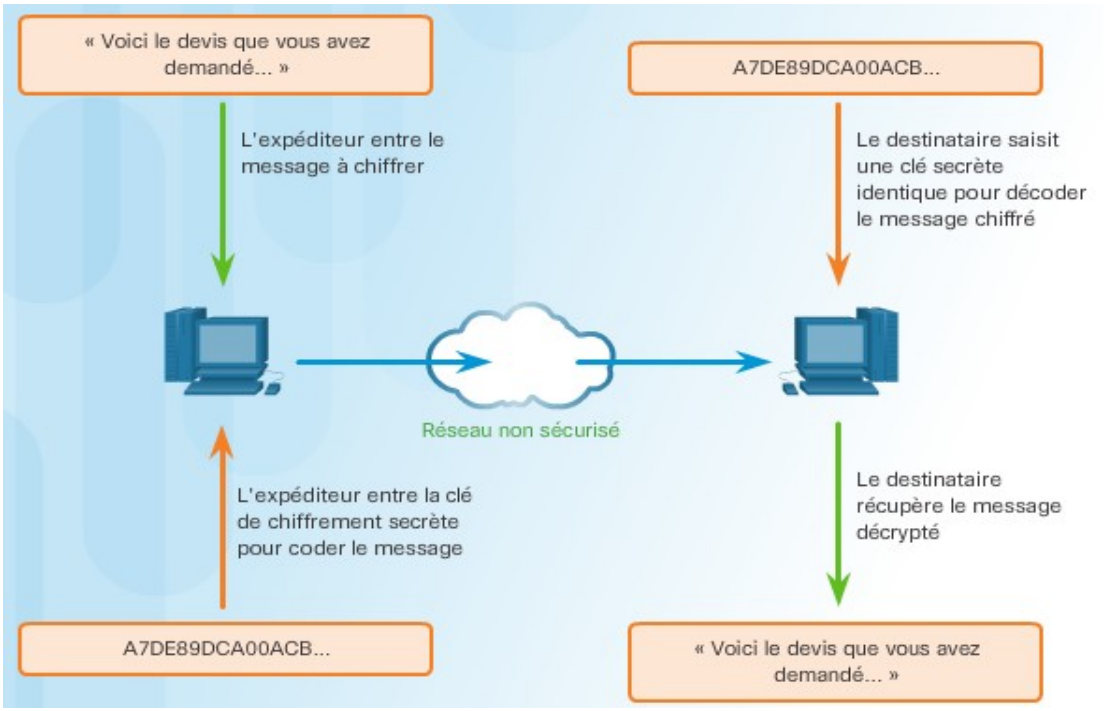
Ici on utilise un **algorithme de chiffrement public**, que tout le monde peut analyser et utiliser, mais qui exploitera un paramètre de chiffrement (**clé de chiffrement**) qui, lui ne sera pas partagé.



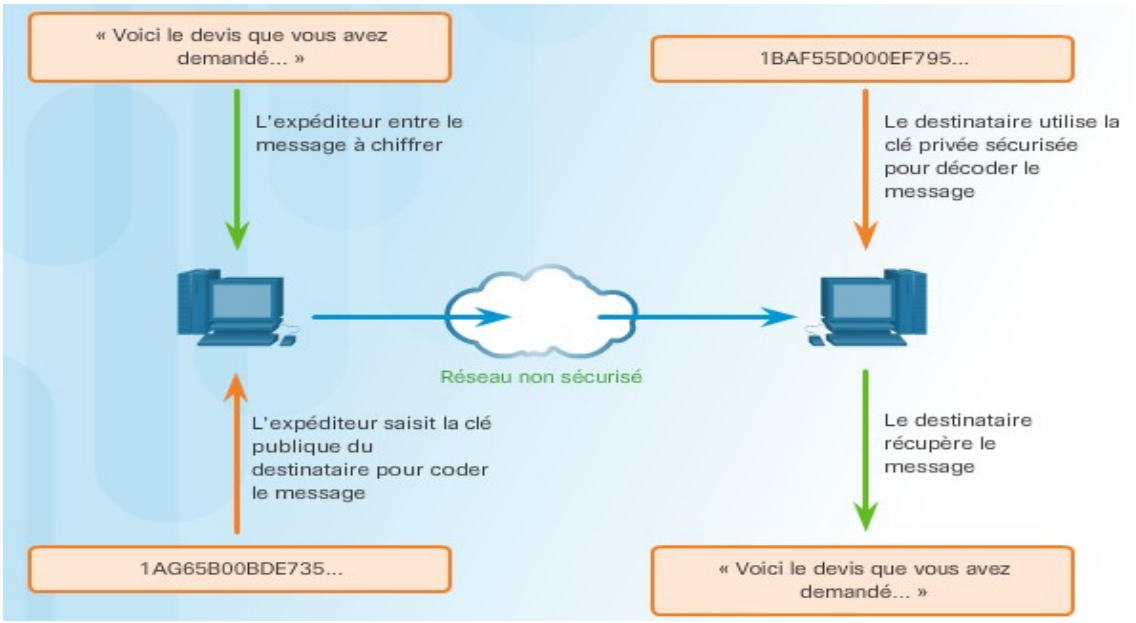
Ce principe, qui peut éventuellement adopter des **combinaisons de clés**, reste à l'heure actuelle le procédé le plus sûr. Ici, pour déchiffrer le message, il faudra la bonne clé, l'algorithme étant public.

IV - Chiffrement Symétrique et Asymétrique

Chiffrement Symétrique : On utilise **une clé** de chiffrement et de déchiffrement identiques. Inconvénient : Comment transmettre cette clé de chiffrement en toute sécurité ?



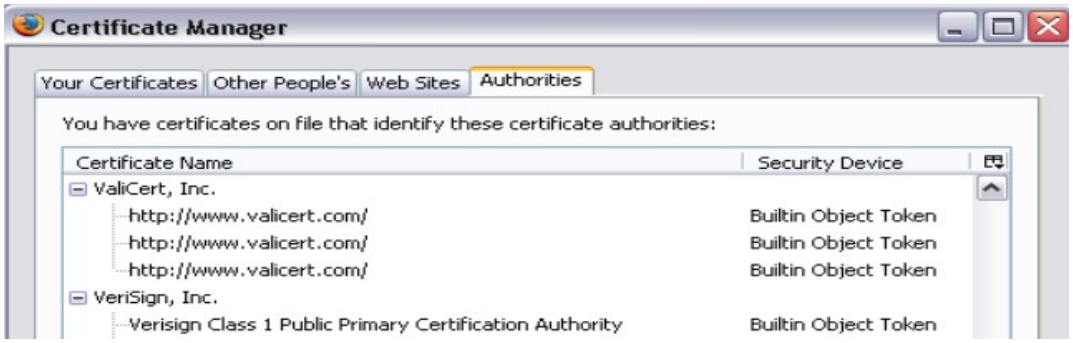
Chiffrement Asymétrique : On utilise 2 clés, une **clé publique** qui peut être partagée et une **clé privée**. Ce qui est **chiffré** avec une **clé publique** ne peut être **déchiffré** qu'avec la **clé privée correspondante**.



Remarque : La **clé publique** est unique, il suffit de la récupérer via un **certificat numérique signé** par un dépositaire, dit « **tiers de confiance** » ou **CA** : **C**ertificate **A**uthority.

V - Les certificats numériques

Lorsqu'on s'adresse à un organisme de confiance (**CA**) pour récupérer une clé publique, il l'envoi avec un **certificat numérique signé**. Les certificats sont à la norme **X.509**. Par exemple un navigateur installe des certificats de divers organismes qui contiennent une clé publique :



VI - Protocole Telnet

Telnet permet d'ouvrir un shell sur un hôte distant. Il utilise le protocole de transport **TCP** et le port **23**. Avec **Telnet** la totalité de la transaction passe en **clair** sur le réseau. Ce protocole n'est plus utilisé en raison de son manque de sécurité.

VII - Protocole SSH

SSH (Secure Shell) permet d'ouvrir un shell sur un hôte distant. Il utilise le protocole de transport **TCP** et le port **22**. Avec **SSH** la totalité de la transaction entre un client et le serveur est cryptée grâce au **chiffrement symétrique** et **asymétrique**.

Le dialogue **SSH** peut se résumer par les étapes suivantes :

- Le serveur envoie sa **clé publique** au client ;
- Le client génère une **clé secrète** (**privée**) et l'envoie au serveur, en cryptant l'échange avec la **clé publique** du serveur (**chiffrement asymétrique**) ;
- Le client et le serveur peuvent alors établir un **canal sécurisé** grâce à la **clé secrète** commune (**chiffrement symétrique**).

SSH possède deux mécanismes différents d'authentification :

- **Authentification par mot de passe** : Couple Nom de compte / Mot de passe ;
- **Authentification par clés** : Couple de **clés privée/publique**.

Utilisé généralement avec un mécanisme d'**authentification par mot de passe**. Lors de la première connexion du client au serveur, le serveur propose d'envoyer **la clé publique** au client :

```
jcabianca@jcabianca-HP-PC:~$ ssh etudiant@192.168.43.45
The authenticity of host '192.168.43.45 (192.168.43.45)' can't be established.
ECDSA key fingerprint is SHA256:Cf2h/nVfzceNWJnxtFh2iDIMPYmHNHNaac0aTMnBiRk.
Are you sure you want to continue connecting (yes/no)?
```

On accepte en saisissant **yes**. La **la clé de chiffrement** est maintenant sauvegardée sur le client.