Résumé rappels réseau

Modèle OSI

Le modèle OSI se compose de 7 couches. Dans un contexte sécurité, le modèle OSI se voit ajouter une couche supplémentaire plus abstraite, la couche 8 associé à l'utilisateur.

8	Utilisateur
7	Application
6	Présentation
5	Session
4	Transport
3	Réseau
2	Liaison de donnée
1	Physique

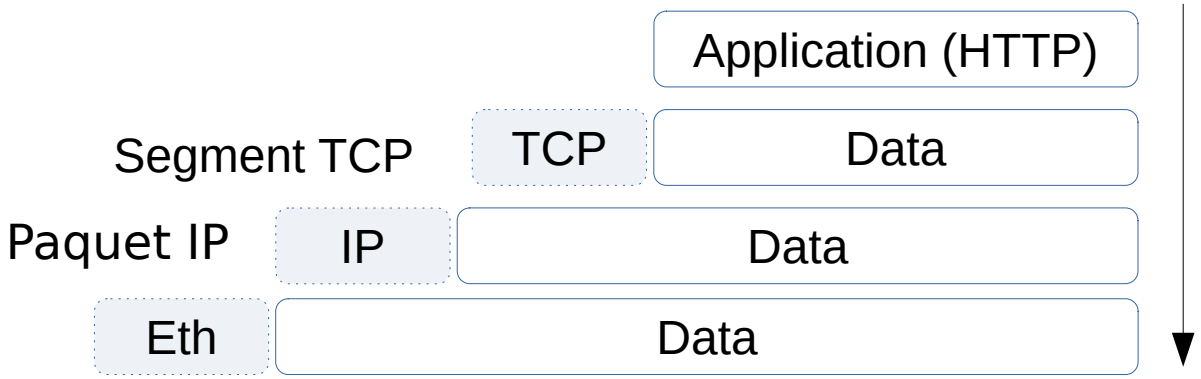
Modèle TCP/IP

Le modèle TCP/IP est lui composé de 4 couches

4 Application
3 Transport
2 Réseau
1 Couche d'accès réseau

Principe d'encapsulation et entêtes

Chaque couche supérieur devient la donnée de la couche inférieur chaque couche ajoutant une entête qui sera propre au protocole utilisé.



Résumé Réseau

Ethernet

Les équipements d'un même réseau communiquent grâce à leurs adresses MAC.
Composition d'une adresse MAC :

00:0d:b4:04:ee:38

6 octets représentés sous forme hexadécimal séparé par un « : ».

Les trois premiers Octets définissent l'OUI « *Organisationally Unique Identifier* » ou « *Identifiant unique d'organisation* ».Il est attribué à un constructeur.

Les trois octets suivants constituent le reste de l'adresse (*NIC, Network Interface Controller*)

C'est le protocole ARP qui permet d'obtenir les adresses MAC sur un réseau.
Une fois obtenu, les adresses MAC sont stockés dans une table ARP qui contient l'association MAC/IP.

Une attaque répandu sur cette couche est l'attaque de type Man In The Middle par empoisonnement de cache ARP vu en cours.

IP

RFC : 791
Il existe deux versions d'IP :

- La version 4 qui code les IP sur 32 bits et offre donc 2^{32} IPs soit 4 294 967 296 IPs
- La version 6 est codé sur 128 bits et offre donc la possibilité d'avoir 2^{128} Ips soit $3{,}402823669 \times 10^{38}$ Ips

C'est la limite du nombre d'IP proposé par la version 4 qui à principalement nécessité le développement de la version 6.

La fragmentation

La fragmentation consiste à la division d'un paquet IP en plusieurs paquets IP en fonction de la taille de la MTU de l'interface réseau par laquelle le paquet transite sauf si le paquet positionne un flags indiquant qu'il en doit pas être fragmenté.
Chaque carte réseau dispose d'une MTU (Maximum Transmission Unit) qui définis la taille maximal d'un paquet pouvant transiter au travers de la carte.

Drapeaux/Flags

Champ ip de 3 bits pouvant prendre 3 valeurs :

- 0 : Bit Réservé
- 1 : Don't fragment
- 2 : More Fragment

Offset

Champ sur 13 bits spécifiant la taille du fragment reçu par rapport au début du paquet IP non fragmenté.
Le premier fragment dispose d'un offset à 0.

TTL

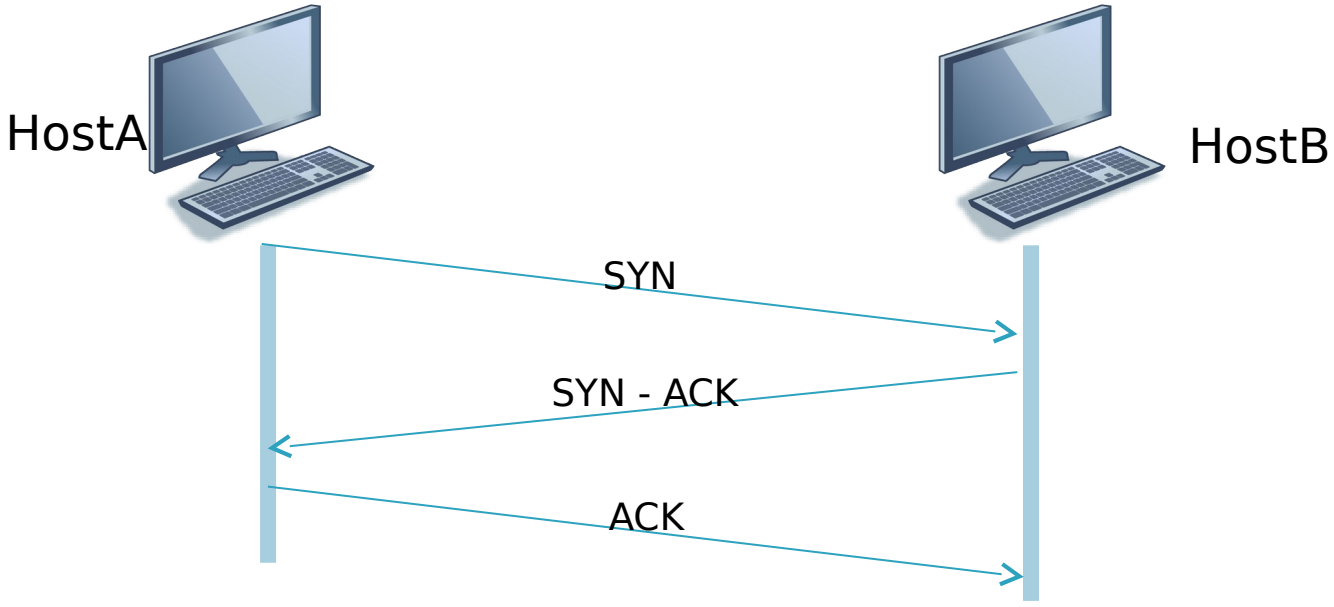
Permet de définir la durée de vie d'un paquet sur le réseau. Codé sur 1 Octet → 2^8=256 sauts maximum.

Routage

Lorsque l'ip de destination ne fais pas partie du réseaux connu par ma carte réseau, alors le paquet doit-être routé.
C'est à dire qu'il appartiendra à un autre équipement nommé passerelle de délivrer le paquet.
Cette décision se prends après avoir consulté la table de routage.
Le routage peut être statique ou dynamique.

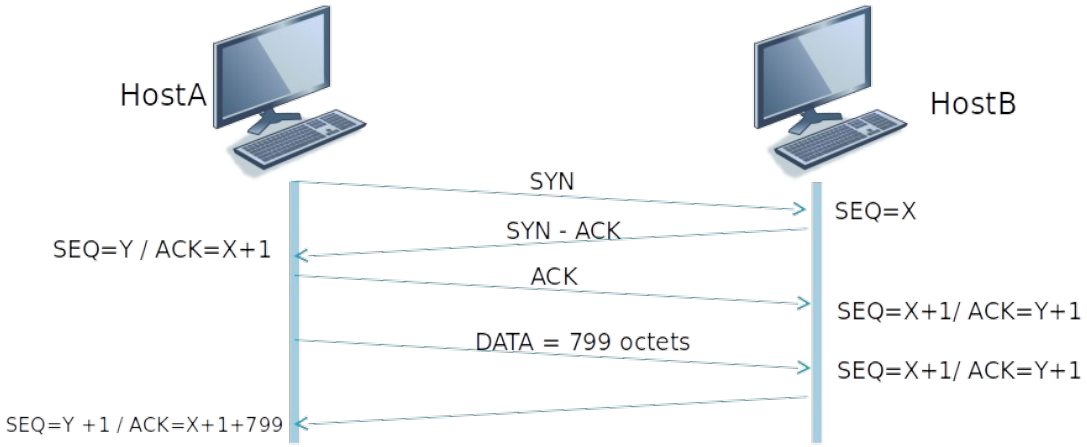
TCP

Chaque nouvelle connexion TCP début par une poignée de main en trois étapes (Three-way handshake) :



TCP utilise un mécanisme utilisant des numéros de séquences et d'acquittement permettant de connaître la quantité de donnée transmise et reçu par un correspondant.

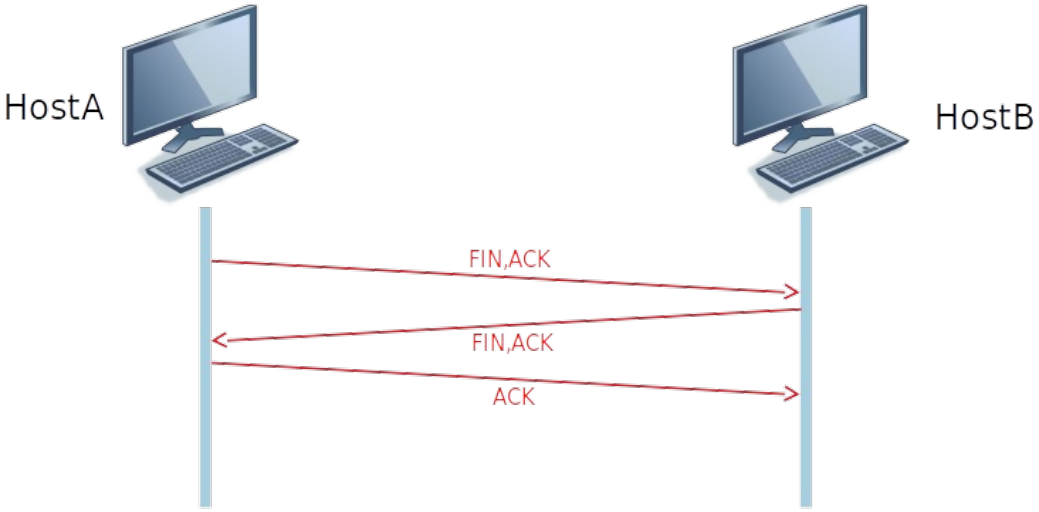
Un numéro de séquence est initialisé par le client et un autre est initialisé par le serveur.
Ceux-ci s'incrémenteront de la quantité de donnée transféré.
Le correspondra acquittera régulièrement les numéros de séquence correspondant à la quantité de donnée reçu.



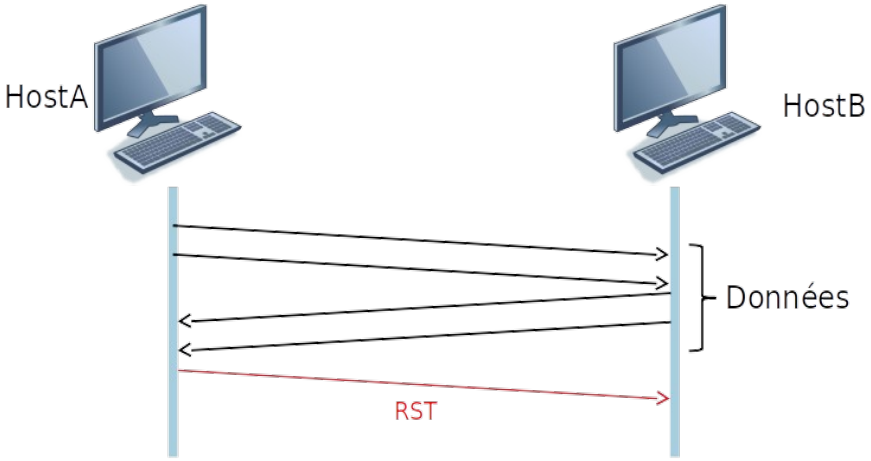
TCP utilise un port source définit aléatoirement et principalement supérieur à 1024 (sauf cas spécifique).
Le port de destination détermine le service ciblé (ex : 25 SMTP)

Une connexion TCP peut se terminer de deux manières :

1. Proprement avec une séquence de fermeture FIN-ACK / FIN-ACK /ACK



2. Brutalement avec un RESET



MSS

La MSS (Maximum Segment Size) est annoncé par le client et par le serveur dans le handshake TCP.

La MSS définis la quantité de données transmise sur TCP pouvant être géré par un équipement.

ICMP

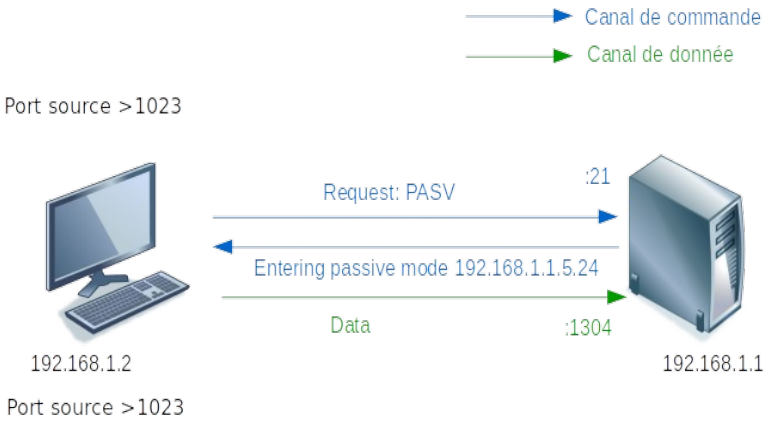
ICMP (Internet Control Message Protocol) RFC 792 est un protocole permettant d'envoyer des messages de contrôle et d'erreur sur un réseau.
Il fonctionne avec une association type/code permettant de définir le message à véhiculer

FTP

FTP RFC 959 est un protocole utilisé pour permettre le transfert de fichiers.
Il s'agit d'un protocole à connexion fille utilisant donc deux connexions/canaux pour fonctionner.
Un canal de commande (port 21) et un canal de donnée.
Il a la particularité d'utiliser deux modes de fonctionnement qui va définir le sens d'établissement du canal de donnée.

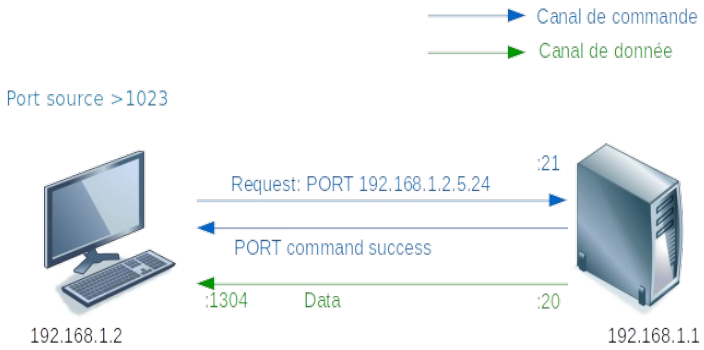
Le mode passif

La connexion de donnée se fait du client → serveur.



Le mode Actif

La connexion de donnée se fait dans le sens serveur → client



Rémy Decrock

Résumé analyse de trafic

Outils

Deux outils principaux :

1. Tcpdump
   → Outil en ligne de commande disponible sur beaucoup de plate formes basées sur unix (Linux, BSD…)
   → Utilise un ensemble d'arguments permettant de filtrer le trafic, arguments qui peuvent-être combinés avec des opérateurs logiques (and, or, not) ex :

   ```
   tcpdump -ni eth1 src 192.168.1.1 and dst 91.212.116.102 and port 80
   ```

2. Wireshark
   → Outil graphique et multi plate forme. Il est basé sur la lib winpcap.
   → Permet de capture et d'analyser en détails des captures, il permet également d'importer des captures effectués depuis tcpdump (si option-w utilisé).

Ces deux outils permettent de mettre en évidence les informations qui transitent au travers de protocoles en clair comme HTTP, FTP ou encore SIP.

Usages

L'analyse de trafic à plusieurs objectifs :

- Debugger une problématique réseau
- « Écouter » passivement le trafic pour identifier des communications non désirés
- Dans le cadre d'audit de sécurité

Résumé
Problèmes utilisateurs

Problématiques associées à l'utilisateur

Social Engineering
Négligence (post-it….)
Manque de culture informatique conduisant créer des vecteurs d'attaques (phishing...)

Problématiques associées à l'administrateur

Absence de politique de sécurité
Utilisation de technologies obsolète
Manque de rigueur (mot de passe par défauts etc...)

Exploitation

Des moteurs de recherches tels que Shodan permettent de cibler des équipements avec des mot de passe par défaut connus et facilite les attaques.

L'utilisation de scripts, logiciels trouvés sur des forums est à proscrire car vous n'avez aucune garantie de la robustesse de la technologie utilisée.

Mitigation

Il est important que les utilisateurs soient sensibilisés à la sécurité informatique mais également les administrateurs réseaux.
Beaucoup d'administrateurs réseau n'ont pas de connaisance en sécurité mais uniquement en administration. Il faut donc également qu'ils soient formés sur les différents aspects de la sécurité (réseau/physique/donnée...)

Résumé
Hash & Chiffrement

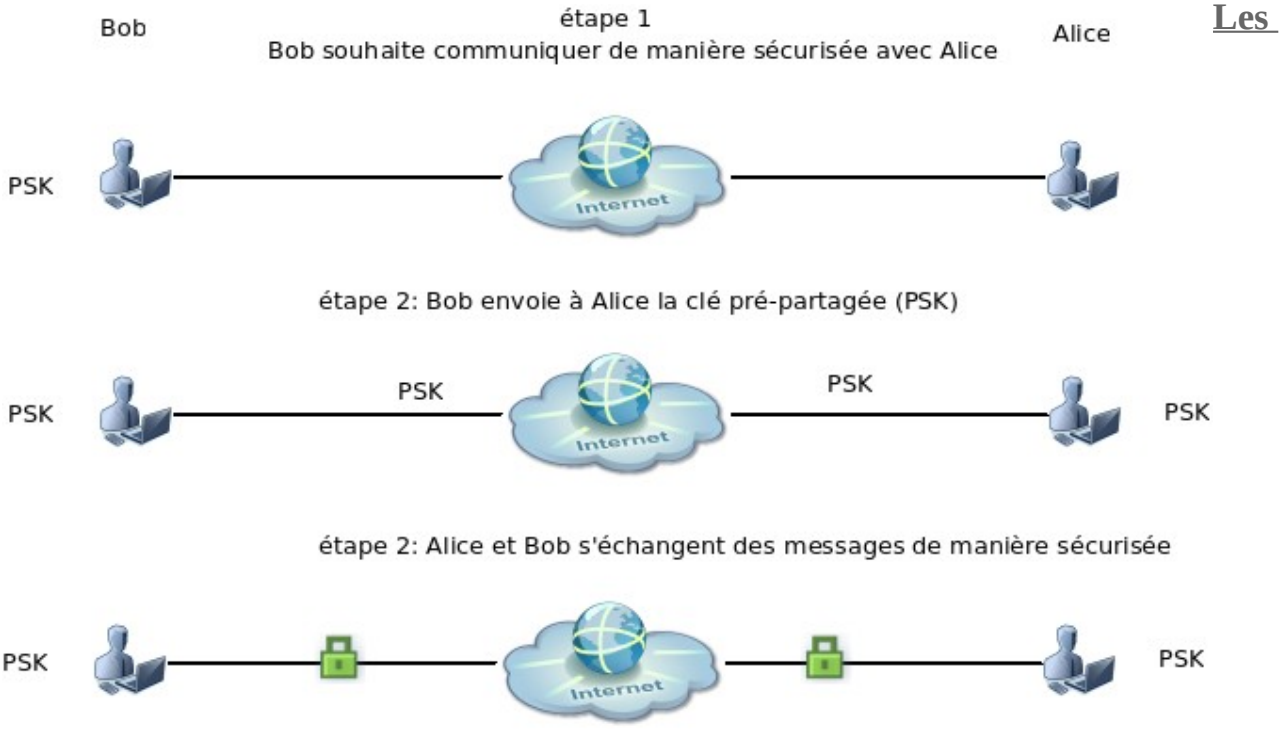
Terminologie & définition

Le chiffrement c'est le fait de rendre un message illisible sauf pour la personne possédant la clé de déchiffrement.
Le terme de cryptage est à proscrire comme définis par l'ANSSI (RGS_V2-0_B1) car c'est un abus de langage qui techniquement en français n'a aucun sensibiliser.
Le chiffrement ne doit pas être confondu avec un le hash ou fonction de hashage.

Chiffrement Symétrique

Le chiffrement symétrique se distingue par le fait que la clé utilisé pour chiffré est la même que celle utilisé pou déchiffrer.

Les



avantages de ce système sont :

→ consomme peu de CPU
→ rapide

Inconvénients :

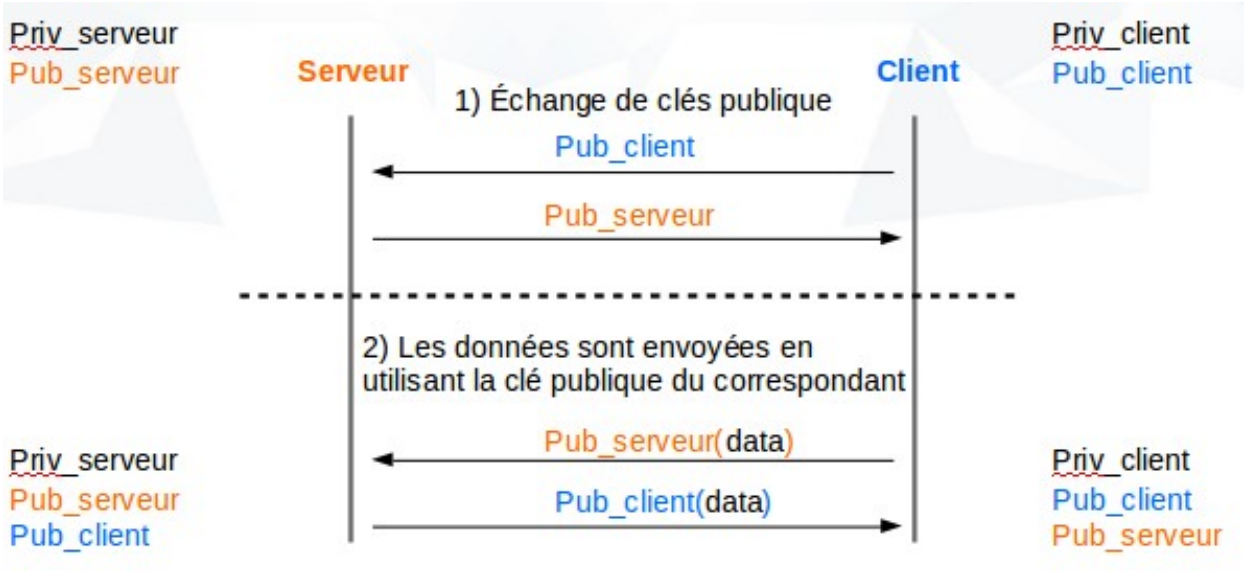
→ gestion des clés difficile (une clé par correspondant)
→ Échange de la clé

Une solution parmi tant d'autres pour chiffrer un document avec un chiffrement symétrique est d'utiliser openssl :

```
Openssl enc -aes-256-cbc -in fichier_a-chiffrer -out fichier_de_sortie
Openssl enc d -aes-256-cbc -in fichier_a_déchiffrer -out fichier de sortie
```

Chiffrement Asymétrique

Le chiffrement Asymétrique utilise deux clés, un privé et l'autre publique (biclé).
La clé privé permet de déchiffrer ce qui a été chiffré avec la clé publique et vice-versa.



La clé la privé est conservé par l'émetteur tandis que la clé publique est en libre accès.

On dit que la clé privé sert à « signer » et la clé publique à « chiffrer ».

Avantages :

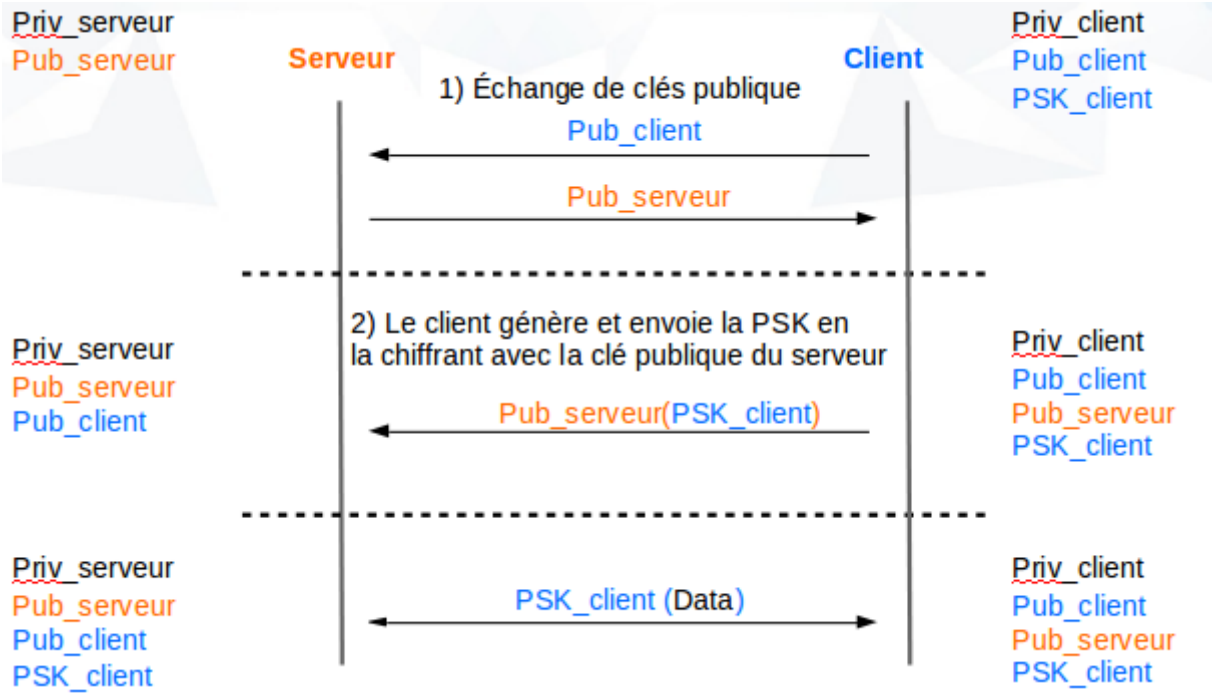
→ gestion des clés plus facile
→ Pas de partage de clé secrète

Inconvénients :

→ Consomme beaucoup de CPU
→ Plus lent

Chiffrement Hybride

Afin de palier au faiblesse du chiffrement asymétrique et symétrique, on mélange les deux systèmes. On va utiliser le système asymétrique afin de protéger l'échange d'une clé symétrique.



Hash

Le hash est une empreinte numérique et n'est pas quelque chose pensé pour être réversible. Il permet de vérifier l'intégrité d'un message, d'un fichier etc. :

```
>echo test | sha256sum
f2ca1bb6c7e907d06dafe4687e579fce76b37e4e93b7605022da52e6ccc26fd2
>echo tesst | sha256sum
9dff28c57d3e32a21c85089288b748a43d5092e9efa09a40ff0a49ffbb848453
```

Ci-dessus le fait de changer une lettre change totalement l'empreinte, seule le mot « test » génère le hash « f2ca1bb6c7e907d06dafe4687e579fce76b37e4e93b7605022da52e6ccc26fd2 » avec l'algorithme sha256

Certificats

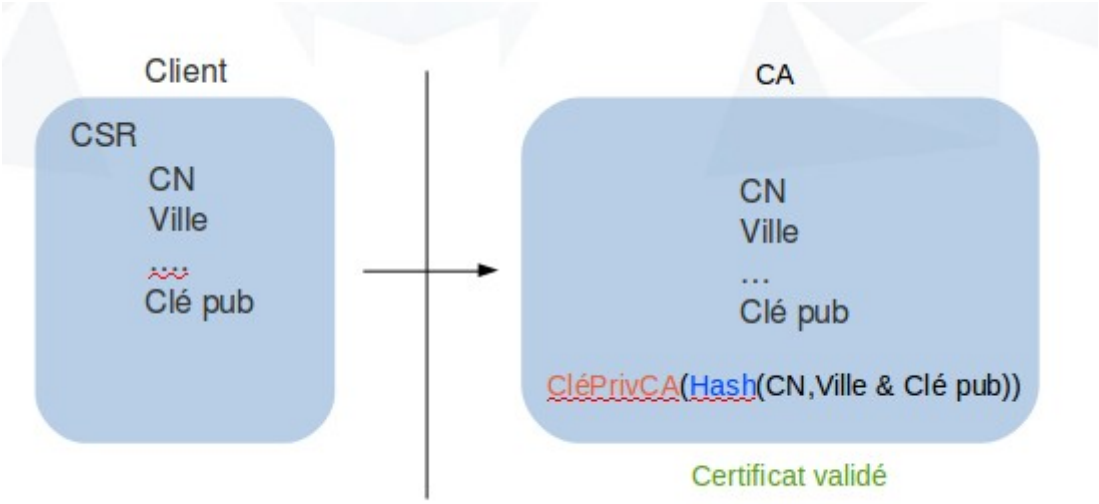
Le certificat permet de vérifier l'identité d'un site ou d'une personne.
Il se base sur le standard X509 et il s'agit d'un système centralisé basé sur une ICP (infrastructure à clé publique) ou PKI (public key infrastructure).

Création d'un certificat

Afin d'obtenir un certificat, il faut d'abord suivre les étapes suivantes :

1. génération d'un CSR (certificat signing request) et d'un biclé. Ce fichier CRS contient les informations du site (CN, email….) ainsi que l'a clé publique, la clé privé reste privé.

2. Le CSR est envoyé à une autorité de certification qui va vérifier votre identité puis signer un hash de l'ensemble des informations du CSR avec sa clé privé.



Vérification du certificat

Le navigateur est en mesure de reconnaître un site valide d'une usurpation grâce à son magasin de certificat.
Celui-ci contient les clés publiques de CA considérés comme valides par le système ou le navigateur.
Ainsi, lorsque le site vous envoie son certificat, votre navigateur va vérifier si une clé publique de son magasin est associée à la CA qui à signé le certificat.
Si c'est le cas, il va utiliser la clé publique pour déchiffrer le hash des informations du certificat chiffré par la CA et le comparé avec le hash calculé par le navigateur à partir des informations en clair du certificat.
Si le déchiffrement se déroule correctement et que les hash correspondent, alors le site est considéré comme valide.

TLS

TLS est un protocole de couche 5 permettant d'établir des sessions sécurisés.
Il est le successeur de SSL depuis 1999
En TLS 1.2 voici les différentes étapes du handshake (7 étapes) :

1. Client hello
2. Server hello
3. Certificate
4. ServerHelloDone
5. ClientKeyExchange
6. ChangeCipherSpec
7. Finished

Résumé OpenSSH

Qu’est-ce que SSH ?

Le protocole SSH est un protocole de communication sécurisé.

Qu’est-ce qu’OpenSSH

OpenSSH est l’implémentation libre de SSH, c’est l’implémentation la plus utilisé à ce jour.

Authentification

L’authentification peut se faire :

- Par mot de passe
  la syntaxe en ligne de commande est :

```
ssh  user@ip
```

  - Avantages :
    - Simple d’utilisation
  - Inconvénients
    - Sensible au brute force
    - Nécessite de communiquer le mot de passe

- Par clé
  la syntaxe en ligne de commande est :

```
ssh  -i chemin_clé_privé  user@ip
```

  - Avantages :
    - Insensible au brute force
    - Plus de souci de partage de mot de passe
  - Inconvénients :
    - Un peu plus complexe à l’utilisation

Usages

SSH est fréquemment utilisé pour :

- Prendre la main à distance sur des équipements (voir Authentification)

- Transférer des fichiers (scp)

```
scp source[@ip:/chemin_fichier] destination[@ip/chemin_fichier]
ex : scp test@ip.fr:/tmp/fichier_a_telecharger /tmp
```

- Y faire passer des connexions TCP (tunnels SSH) :
  - via une socket locale en 1 pour 1 avec l'options -L

```
ssh -L dstport_local:ip_sortie_tunnel:dstport_sortie_tunnel user@ip
```

  - via une socket distance (reverse) en 1 pour 1 avec l'options -R

```
ssh -R dstport_local:ip_sortie_tunnel:dstport_sortie_tunnel user@ip
```

  - via une socket socks5 dynamique en 1 pour N avec l'options -D

```
ssh -D PORT_socks5 user@ip
```