UE 3&4 M3-3 Sécurité web

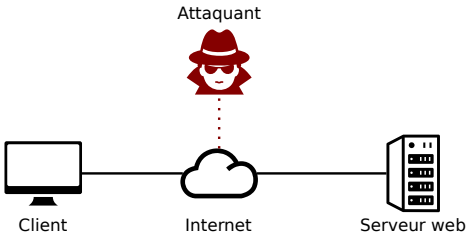
Chiffrement

Louis DEGUILLAUME
louis.deguillaume@u-bordeaux.fr

18 décembre 2022

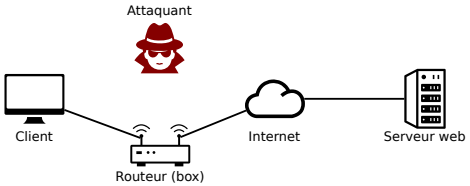
HTTP (HyperText Transfer Protocol)

- Les échanges sont en clairs
- Aucune garantie de :
  - Intégrité
  - Confidentialité
  - Identification du client/serveur
  - Authentification du client/serveur



MITM (Man In The Middle)

HTTP ne permet pas d'authentifier le client ou le serveur.

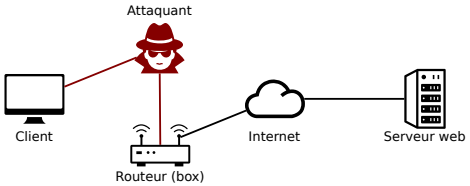


MITM (Man In The Middle)

HTTP ne permet pas d'authentifier le client ou le serveur.

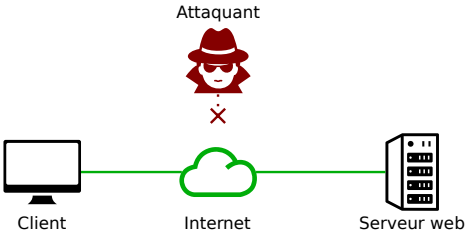
Il est donc possible de se mettre en interception et de :

- Observer ce qui est échangé (mot de passes, ...)
- Modifier les informations échangées (tromperie, ...)
- Empêcher l'accès à une ressource (déni de service, ...)



HTTPS (HyperText Transfer Protocol Secure)

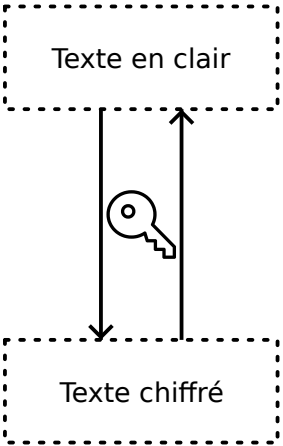
- Les échanges sont chiffrés
- Garantie de :
  - Intégrité
  - Confidentialité
  - Authentification du serveur (donc Identification)
  - Authentification possible du client (donc Identification)



Source : ssi.gouv.fr

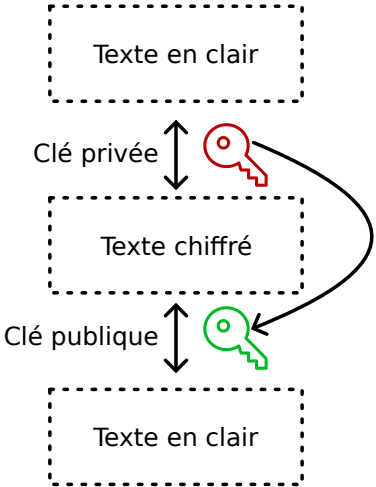
Le chiffrement symétrique

- Une même clé est utilisée pour chiffrer et déchiffrer le message
- Robuste, selon paramètres, comme :
  - pertinence de l'algorithme (ex : AES)
  - longueur de la clé (ex : 256 bits)
  - non prédictibilité de la clé
- peu coûteux en temps de traitement
- la clé doit être connue par l'expéditeur et le destinataire



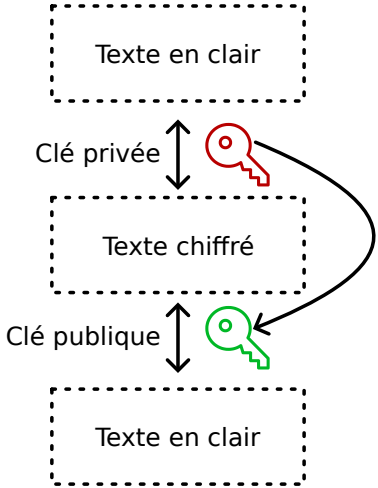
Le chiffrement asymétrique

- Une paire de clés est utilisée pour chiffrer et déchiffrer le message
- Basé sur la factorisation en nombres premiers
  - s'il est facile de multiplier 2 nombres premiers...
  - il est très difficile de les retrouver, surtout si ces nombres sont très grands
- Un message chiffré avec une clé ne peut être déchiffré que par l'autre
- La clé publique est créé à partir de la clé privée



Le chiffrement asymétrique

- Permet d'échanger des informations sans avoir besoin de partager un secret commun
- Coûteux en terme de temps de traitement
  - souvent utilisé pour échanger une clé symétrique (protocole HTTPS)
- Deux algorithmes principaux :
  - basé sur la factorisation complexe des grands nombres
  - basé sur les courbes elliptiques
- Inconvénient majeur : il est impossible d'être sûr que la clé publique fournie est bien celle de l'expéditeur.
  - passage par un tiers de confiance qui certifiera la clé de l'expéditeur
  - génération d'un **certificat numérique**

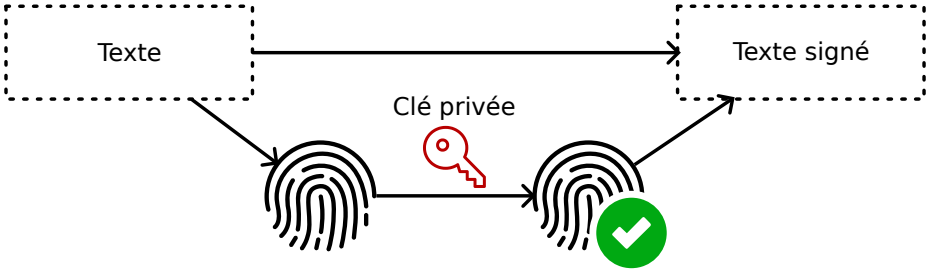


Le chiffrement asymétrique

Signature d'un document

Alice envoie un message à Bob, et on souhaite que Bob puisse s'assurer que c'est bien Alice qui l'a envoyé

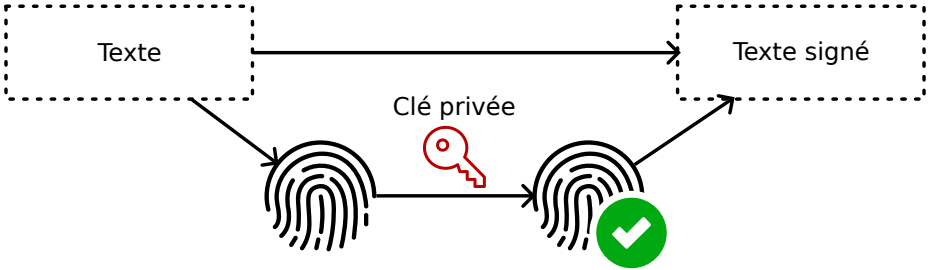
- Alice calcule l'empreinte de son message, et la chiffre avec sa clé privée
- Alice envoie le message et l'empreinte chiffrée à Bob



Signature d'un document

Alice envoie un message à Bob, et on souhaite que Bob puisse s'assurer que c'est bien Alice qui l'a envoyé :

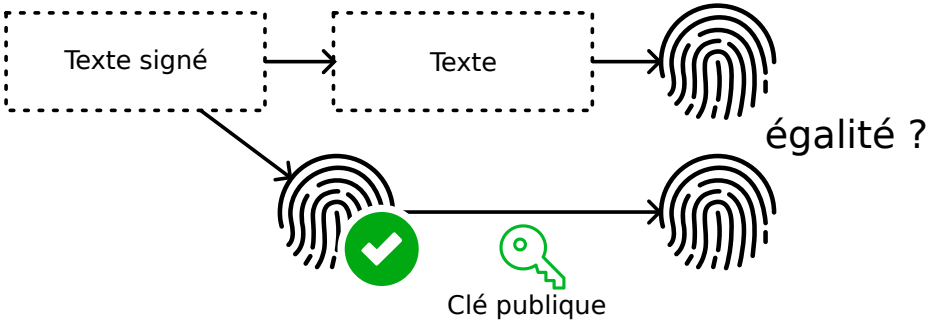
- Alice calcule l'empreinte de son message, et la chiffre avec sa clé privée
- Alice envoie le message et l'empreinte chiffrée à Bob



Signature d'un document

Bob reçoit le message :

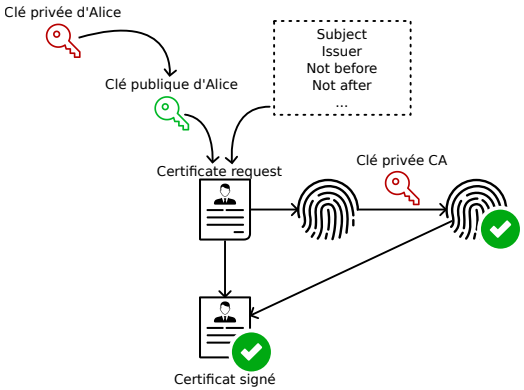
- il en calcule l'empreinte
- il déchiffre l'empreinte chiffrée fournie par Alice avec sa clé publique
- si les 2 empreintes sont identiques, le message provient bien d'Alice



Création d'un certificat

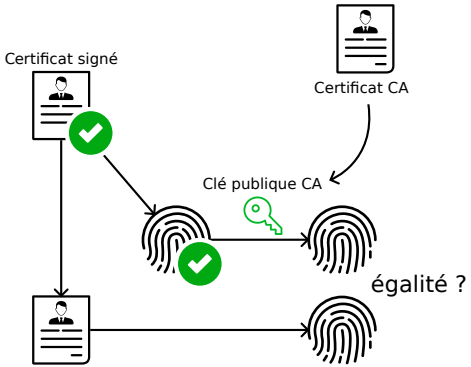
Alice créé un certificat et en demande la signature, entre autres :

- la clé publique d'Alice
- des informations complémentaires :
  - nom du site web correspondant (ou nom de la personne détentrice...)
  - des dates de validité
  - un numéro de série
  - l'adresse où trouver la liste des certificats révoqués
  - ...
- le condensat chiffré par l'autorité de certification



Vérifier un certificat

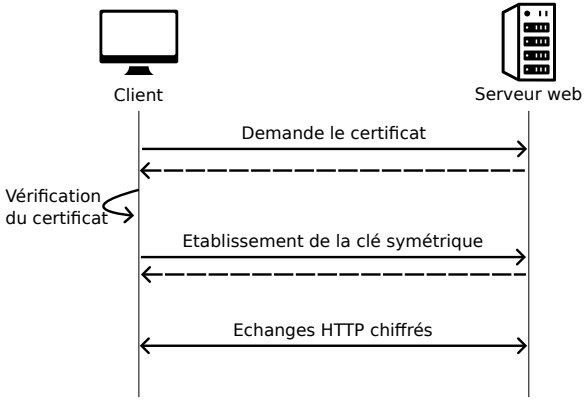
- Pour vérifier un certificat :
  - déchiffrer le condensat avec la clé publique l'autorité de certification (CA)
  - vérifier que le condensat obtenu est valide
- Les certificats des autorités de certification sont intégrés directement dans les navigateurs et les systèmes d'exploitation
  - la responsabilité en incombe aux éditeurs
- En cas de révocation du certificat d'une autorité, tous ceux qu'elle aura généré seront également révoqués



Principe de fonctionnement d'un échange HTTPS

Le navigateur se connecte à un site https

1. le serveur fournit son certificat et les algorithmes qu'il autorise
2. le client vérifie le certificat
3. il génère une partie de la clé symétrique, qu'il chiffre avec le certificat
4. le serveur fournit la seconde partie de la clé symétrique, chiffrée avec sa clé privée
5. le dialogue chiffré peut commencer



Tester le paramétrage d'un site https

La configuration peut être testée (si le site est accessible depuis internet) avec des outils mis en ligne

- `https://www.ssllabs.com/ssltest` : test complet
- `https://ssl-config.mozilla.org/` : outil pour donner de bonnes configurations à appliquer

Pour aller plus loin : créer un certificat utilisable dans son PC

- Pendant le développement, il faut pouvoir vérifier que la connexion https fonctionne :
  - cookies de session non chiffrés refusés
  - redirection systématique en https
  - etc.
- Les navigateurs récents refusent les certificats qui ne sont pas validés (Attention le CN ne suffit plus, il faut renseigner le SAN)
- Pour des usages personnels ou de développement :
  - création d'une autorité de certification, et intégration de son certificat dans le navigateur
  - génération des certificats avec cette AC

Crédits

Document inspiré du cours « Chiffrement »de E. Quinton, eric.quinton@protonmail.com