JULES HATCHUEL

ANTHONY RUGGERI

COMPTE RENDU HTTPS / SSL / TLS

Protocoles étudiés :

HTTP : HyperText Transfer Protocol

Protocole d'échange d'informations client / serveur web

HTTPS : HyperText Transfer Protocol Secure

Protocole d'échange d'informations client / serveur web de façon sécurisée (HTTP + TLS)

SSL : Secure Sockets Layer

Protocole d'ajout de couche cryptée asymétrique (obsolète).

TLS : Transport Layer Security

Protocole d'ajout de couche cryptée asymétrique (le plus récent).

Résumé :

- HTTPS :

Le rôle de HTTPS est le même que http : envoyer des requêtes à un serveur web, auquel est ajouté TLS qui permet de chiffrer ces échanges. Lorsqu'un client demande un transfert de données sécurisé (une requête https), il fournit au serveur une liste de protocoles pris en charge. Le serveur indique alors avec quel protocole il est compatible. Si les conditions sont réunies, il fournit aussi un certificat attestant de sa propre authenticité. Des organismes peuvent jouer le rôle de garant sur l'authenticité de ces certificats : les autorités de certification. Sinon, le client peut préenregistrer un certificat pour pouvoir ensuite vérifier.

- SSL :

SSL est un protocole développé par Netscape en 1994. Il permet le chiffrement d'informations transitant entre un serveur et un client. La première version officielle (2.0) sort en 1995. Il est définitivement abandonné en 2015 car ne remplit pas toutes les conditions de sécurité nécessaires. Il est suivi de TLS qui comble ces lacunes.

- TLS :

La première version (1.0) succède à SSL, après le constat de l'obsolescence de SSL, l'IETF ou Internet Engineering Task Force (un organisme à but non lucratif) élabore ce protocole. Il vient, comme SSL, se greffer à HTTP pour en crypter les échanges et y apporter des solutions concernant les problèmes de sécurité.

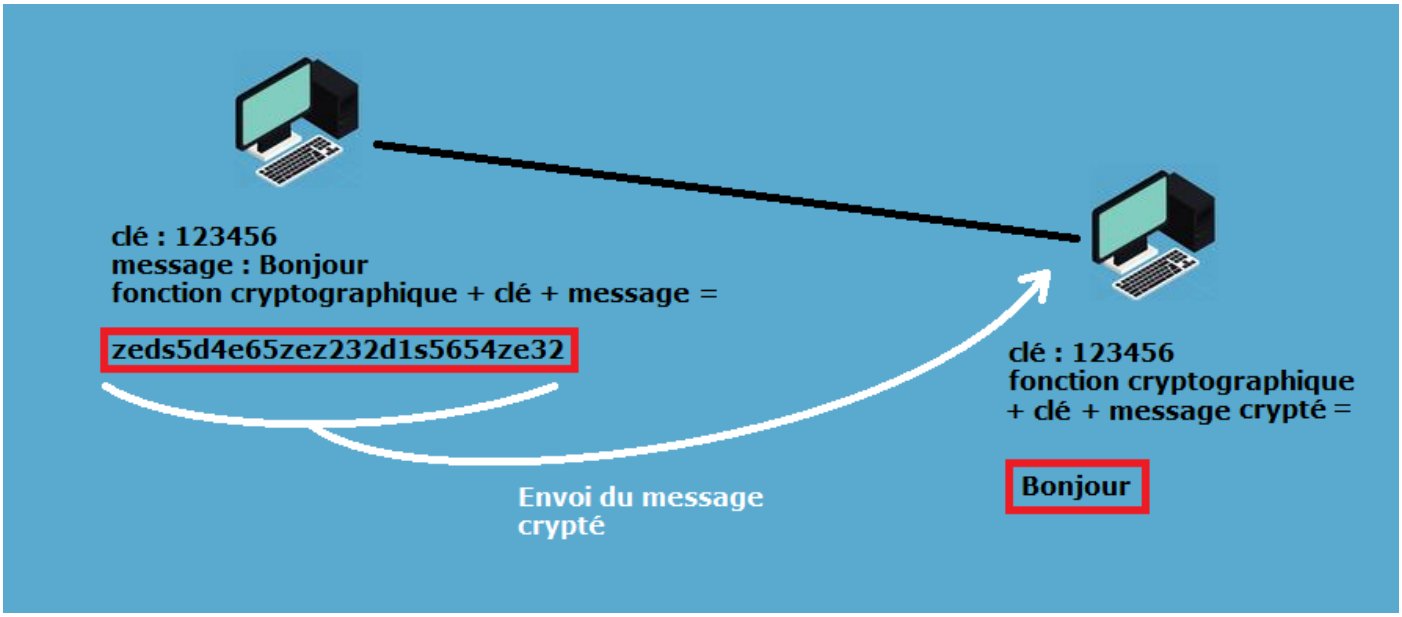
Qu’est-ce qu’un échange considéré comme sécurisé :

On considère comme sécurisé un système permettant l’échange d’information au sein duquel sont garantis ces trois points :

- Confidentialité : l’information n’est pas accessible à un tiers autre que le destinataire
- Authenticité : l’expéditeur est sûr de l’identité du destinataire et le destinataire sûr de l’identité de l’expéditeur
- Intégrité : l’information n’a pas été modifiée

Fonctionnement symétrique / asymétrique :

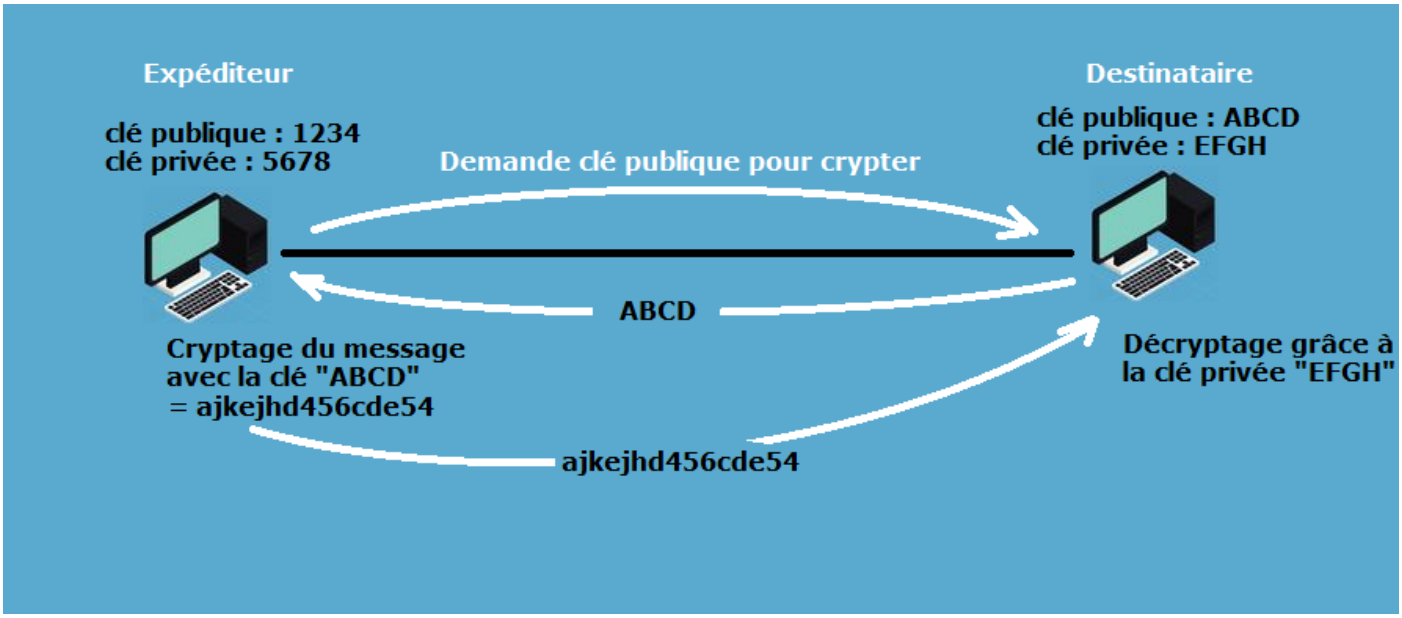
- Cryptographie à clé symétrique : l’expéditeur et le destinataire ont la même clé qui sert à crypter et décrypter une information. Elle est efficace mais oblige la source et la destination à se mettre d’accord sur une seule et même clé, ce qui n’est pas possible lorsqu’une distance sépare les deux points car cela impliquerait l’envoi de la clé, et ce ne serait donc pas sécurisé.



Avec cette technique il est donc possible d’intercepter le message sur le réseau, mais il faut pouvoir le décrypter ce qui prends du temps sans la clé.

Pour pallier le problème de l’échange de clé sur un système à clé symétrique il existe la technique de cryptographie asymétrique.

- Cryptographie à clé asymétrique : deux clés sont créées par participant (expéditeur et destinataire). Ces paires de clés sont liées mathématiquement de telle sorte que l'une sert à crypter mais pas à décrypter et l'autre à décrypter mais pas à crypter.
  L'une sert donc uniquement à crypter l'information (la clé publique), l'autre à la décrypter (la clé privée). Lorsque la clé privée est créée elle reste où elle est et n'est jamais communiquée. Lors d'un échange d'information, l'expéditeur demande au destinataire sa clé publique afin de crypter le message qu'il veut lui transmettre, le destinataire envoie donc sa clé publique.
  A ce stade, vu que cette clé ne sert qu'à crypter une information, elle peut être communiquée en clair sans avoir peur que quelqu'un d'autre la récupère et la garde.
  Grâce à cette clé, l'expéditeur crypte son message qu'il envoit ensuite au destinataire qui peut le décrypter grâce à sa clé privée.



La cryptographie asymétrique apporte la solution au problème de la confidentialité nécessaire à l'échange mais ne garantie pas l'authenticité ni l'intégrité de l'information.

En effet, quelqu'un peut se faire passer pour le destinataire et remplacer les clés par les siennes (c'est l'attaque de l'homme du milieu) et ainsi intercepter tous les échanges. Après avoir reçu la clé publique du destinataire, il la remplace par la sienne, l'expéditeur pense alors envoyer le message crypté avec la clé publique du destinataire alors qu'il s'agit de celle d'un intrus qui peut alors décrypter le message avec sa clé privée car il a été crypté avec sa clé publique. Au choix ensuite de replacer le message crypté avec la clé du vrai destinataire ou pas, et intègre ou pas.

C'est là qu'intervient TLS qui ajoute les couches de sécurité nécessaires : une autorité de certification reconnue attestant de l'authenticité, et de l'intégrité grâce à une signature qui, une fois décryptée est censée donner des informations identiques à celles données par le serveur.

Fonctionnement général de TLS :

Lorsqu'un client souhaite initier une connexion sécurisée il en fait la demande au serveur en lui indiquant les versions de protocoles pris en charge, le serveur choisis la version qu'il préfère (en général la plus récente commune) puis lui réponds en lui envoyant un certificat contenant :

- des informations publiques en clair comme le nom de l'organisation, le lieu, le type et la version des algorithmes utilisés, ces informations sont hachées, et cryptées avec la clé publique du serveur, c'est la signature : le hachage est censé être unique, son cryptage aussi, ce qui garantie l'unicité de la signature. Cette signature est concaténée aux informations au sein du certificat, puis envoyées avec la clé publique.

Le protocole implique que le client soit compatible avec ce protocole. Ce qui implique que le client ou navigateur détient déjà la liste des clés publiques des autorités de certifications. Lorsqu'il reçoit le certificat du serveur il peut alors procéder aux vérifications : si l'une des clés détenues par le navigateur permet de déchiffrer la signature du certificat et que les données obtenues sont identiques à celles contenues en en-tête de ce certificat, c'est qu'il est signé par une autorité de certification (car il a été déchiffré avec une clé intégrée de façon « native » au navigateur).

Si le déchiffrement de la signature ne se fait pas avec les clés natives mais avec la clé publique envoyée par le serveur, c'est qu'il est auto signé, c'est au client de faire son choix d'accepter ce certificat ou non. En intranet, cette solution peut suffir.

Dernier cas : la signature n'est pas décryptable, c'est que la clé fournie n'est pas authentique, la connexion ne peut pas se faire.

NB : les autorités de certifications peuvent prendre un certain temps pour mettre à jour la liste des certificats révoqués, c'est pourquoi a été mis en place le protocole OCSP qui met en place les règles de disponibilité. Le client peut interroger un répondeur OCSP qui lui répondra sur la validité d'un certificat ou non.

La plupart des autorités de certifications sont vénales mais il en existe des gratuites, voici un tableau non exhaustif :

Autorité	Let's Encrypt	Comodo CA	GlobalSign	AlphaSSL
Lien	Lien	Lien	Lien	Lien
Prix	Gratuit	88$/an	249$/an	49$/an

Sur le site https://www.certificat.fr on peut comparer selon nos besoins les meilleures offres.

Sur le site https://caniuse.com/#search=tls%201.3 on peut consulter les versions de navigateurs nécessaires pour TLS 1.3.

Lien vers la spécification TLS 1.3 : https://tools.ietf.org/html/rfc8446

Sources :

HTTPS : de SSL à TLS 1.3 : https://openweb.eu.org/articles/https-de-ssl-a-tls-1-3

Wikipedia : TLS : https://fr.wikipedia.org/wiki/Transport_Layer_Security

YouTube, L'Informateur : https://www.youtube.com/channel/UCMzZh0q-rcd9yDEOTXAH90g