CH9 Sécurisation et chiffrement des communications

Comment se connecter au site https://gitlab.com/edaldegan/nsi_cours ?

1	L'utilisateur saisie l'adresse dans le navigateur.
2	Le navigateur demande au serveur DNS l'adresse IP du serveur web correspondant. (11112331)
3	Le navigateur établit une connexion TCP vers l'IP du serveur web sur le port 80.
4	La connexion étant établie, le client et le serveur échange des données.

Dans ce cas de figure, **sécuriser la communication**, c'est garantir que :
- *les échanges client-serveur ne sont lisibles que par la source et le destinataire.*
- *le serveur auquel le client se connecte est bien celui auquel on pense se connecter*

1. Les méthodes de chiffrement

Pour assurer la sécurisation de la communication, on va pouvoir **chiffrer** la communication.

Un peu de vocabulaire :
- **Coder** (ou ***encoder***) = Représenter de l'information par un ensemble de signes prédéfinis.
- **Décoder** = Interpréter un ensemble de signes pour en extraire l'information qu'ils représentent.
- **Chiffrer** = Rendre une suite de symbole incompréhensible au moyen d'une ***clé de chiffrement.***
- **Déchiffrer =** Retrouver la suite de symbole originale à partir du message chiffré en utilisant la ***clé de chiffrement***. On parle de *Décrypter* lorsqu'on n'utilise pas la clé.

A Retenir :

Dans le cas où BOB envoie un message à ALICE, on retrouve les 2 méthodes de chiffrement utilisées :

- Le chiffrement symétrique

BOB et ALICE disposent tous les deux des fonctions suivantes :
C(*m, k*) est la fonction de chiffrement qui produit un message chiffré **s.**
D(*s, k*) est la fonction de déchiffrement qui produit le message en clair **m.**

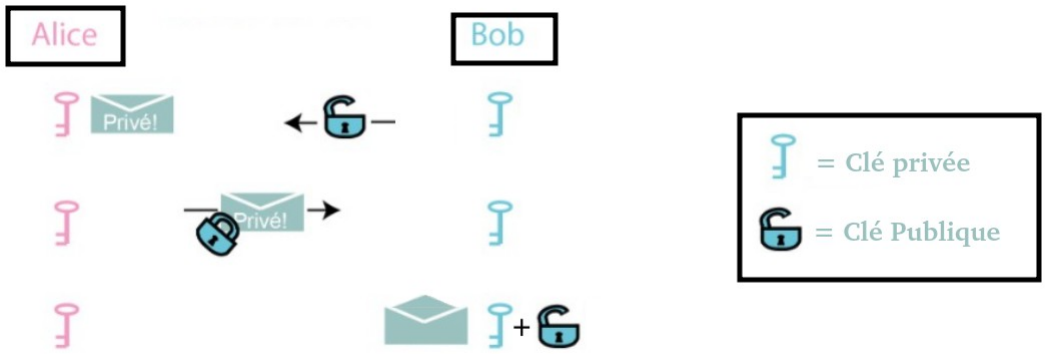
$$C : (m, k) \rightarrow s \quad \text{et} \quad D : (s, k) \rightarrow m$$

1 - *BOB utilise la fonction* ***C(m, k)*** *pour chiffrer le message m et envoie le résultat s à ALICE.*
2 - *ALICE utilise la fonction* ***D(s, k)*** *pour déchiffrer s reçu et retrouver le message m.*

Dans les 2 sens, on utilise **k** la clé de chiffrement, d'où le nom de *symétrique*.

- Le chiffrement asymétrique (1976, Diffie & Hellman)

BOB et ALICE disposent chacun de leur propres « cadenas + clés », un des « cadenas » va circuler sur le réseau : c'est la ***clé publique***. Une fois fermé, ce « cadenas » ne pourra s'ouvrir qu'avec la ***clé privée*** associée, qui elle n'a pas circulé sur le réseau.



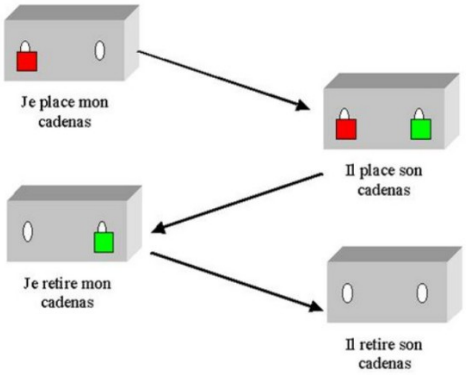
1 - *BOB envoie à ALICE son cadenas (clé publique), il garde sa clé (clé privée).*
2 - *ALICE renvoie à BOB le message chiffré avec la clé publique. Personne n'en possède la clé excepté BOB.*
3 - *BOB reçoit le message chiffré et peut le déchiffre à l'aide de sa clé (clé privée).*

Exemples :

- Certains chiffrements symétriques

Nom	Type	Clé	Attaque possible
Chiffre de César (-50 av. J.-C.)	Monoalphabétique	Décalage de l'alphabet	- Force brute - Analyse fréquentielle
Chiffre de Vigenère (1586)	Polyalphabétique	Décalage différent pour chaque lettre	Cryptanalyse de Babbage(1854) et Kasiski (1863)
Chiffre de Vernam (1917)	Masque jetable	Aléatoire et aussi longue que le message	Incassable
AES (Advanced Encryption Standard) (2000)	Permutations itérées de blocs (10 à 14 tours)	Clés de 128 à 256 bits	Non cassé à ce jour

- Chiffrement asymétrique à cadenas :

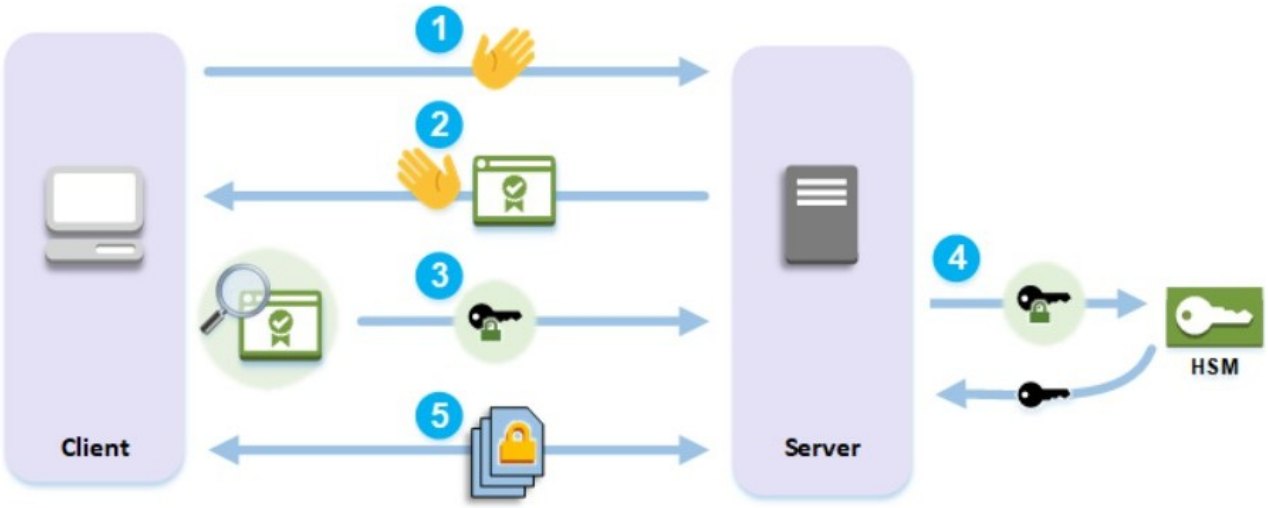


→ **ACT1 Chiffrer pour sécuriser TRAVAIL A RENDRE**

2. Les protocoles HTTPS et TLS

La connexion au site web https://gitlab.com/edaldegan/nsi_cours utilise un protocole particulier : le HTTPS qui lui permet de sécuriser les échanges grâce à un certificat qui sert de à authentifier la source et le destinataire.

Pour établir une connexion HTTPS, votre serveur web effectue un processus de négociation avec les clients (RSA utilisé, voir https://fr.wikipedia.org/wiki/Chiffrement_RSA).
Dans le cadre de ce processus, le serveur décharge une partie du traitement cryptographique sur les HSM (Hardware Security Module).



1. Le client envoie un message Hello au serveur.
2. Le serveur répond par un message Hello et envoie son certificat de serveur.
3. Le client effectue les actions suivantes :

a. Il vérifie que le certificat du serveur SSL/TLS est signé par un certificat racine auquel il fait confiance.

b. Il extrait la clé publique du certificat du serveur.

c. Il génère un prémaster secret et le chiffre avec la clé publique du serveur.

d. Il envoie le prémaster secret chiffré au serveur.

4. Pour déchiffrer le prémaster secret du client, le serveur l'envoie au HSM. Le HSM utilise la clé privée dans le HSM pour déchiffrer le prémaster secret, puis il envoie le prémaster secret au serveur. Indépendamment, le client et le serveur utilisent tous les deux le prémaster secret et certaines informations issues des messages Hello pour calculer un secret principal.

5. Le processus de négociation se termine. Dans le reste de la session, tous les messages envoyés entre le client et le serveur sont chiffrés avec des dérivés du secret principal.

Autres sources :
https://www.cnil.fr/fr/comprendre-les-grands-principes-de-la-cryptologie-et-du-chiffrement
https://www.ssi.gouv.fr/particulier/bonnes-pratiques/crypto-le-webdoc/cryptologie-art-ou-science-du-secret/