

Sécurisation des communications

J. Boucher

Lycée Pierre-Paul RIQUET, Terminale NSI

22 mars 2025

Plan

1 3. Protocole https

Retour sur la problématique initiale

- Comment garantir que le contenu échanger ne soit connu que de la source et de la destination ?

- Que le serveur auquel on se connecte est bien celui qu'il dit être ?

Retour sur la problématique initiale

- Comment garantir que le contenu échangé ne soit connu que de la source et de la destination ?
 - Le chiffrement (Diffie-Hellman + AES) empêche les routeurs et autres machines intermédiaire de lire le contenu des messages.
- Que le serveur auquel on se connecte est bien celui qu'il dit être ?
 - L'authentification (RSA) empêche l'exécution d'attaque de l'homme du milieu.

Retour sur la problématique initiale

Lorsqu'un utilisateur utilise son navigateur pour accéder à la page Web `http://snt-nsi.info/tnsi/`,

- 1 le navigateur isole le protocole (`http`), le nom du serveur (`snt-nsi.info`) et la ressource demandée (`/tnsi/`);
- 2 le navigateur effectue une requête DNS pour obtenir l'adresse IP du serveur (IPv4 : 51.178.82.21);
→ couche Internet : déterminer la route
- 3 le navigateur se connecte à la machine dont l'adresse IP est 51.178.82.21, en utilisant le *protocole TCP* sur le port 80;
→ couche transport : garantir l'intégrité des données
- 4 une fois la connexion établie, client et serveur échange des données en utilisant le protocole HTTP.
→ couche d'applications : protocoles de haut niveau

Retour sur la problématique initiale

Lorsqu'un utilisateur utilise son navigateur pour accéder à la page Web

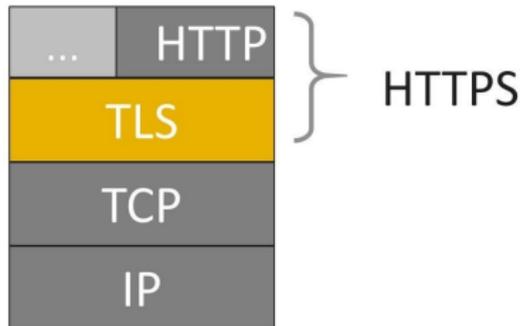
`https://snt-nsi.info/tnsi/`,

- 1 le navigateur isole le protocole (`http`), le nom du serveur (`snt-nsi.info`) et la ressource demandée (`/tnsi/`);
- 2 le navigateur effectue une requête DNS pour obtenir l'adresse IP du serveur (IPv4 : 51.178.82.21);
→ couche Internet : déterminer la route
- 3 le navigateur se connecte à la machine dont l'adresse IP est 51.178.82.21, en utilisant le *protocole TCP* sur le port 80;
→ couche transport : garantir l'intégrité des données
→ Utilisation du protocole TLS (Transport Layer Security) ou sécurité de la couche transport
- 4 une fois la connexion établie, client et serveur échange des données en utilisant le protocole HTTP.
→ couche d'applications : protocoles de haut niveau

Protocole HTTPS

Protocoles HTTPS et TLS

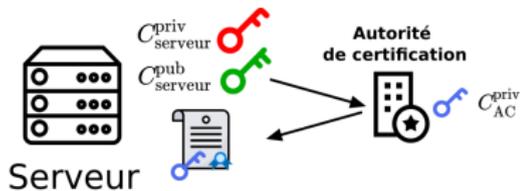
- Le **protocole HTTPS** utilise le protocole **TLS** (Transport Layer Security) ou *sécurité de la couche de transport* pour ajouter au protocole HTTP une phase d'authentification et d'échange de clé de chiffrement.
- Le **protocole TLS** ajoute une phase d'authentification du serveur et la mise en place sécurisée d'une clé de chiffrement symétrique, appelé *clé de session*.



Phase de mise en place, appelée « **poignée de main TLS** » (ou *TLS Handshake* en anglais).



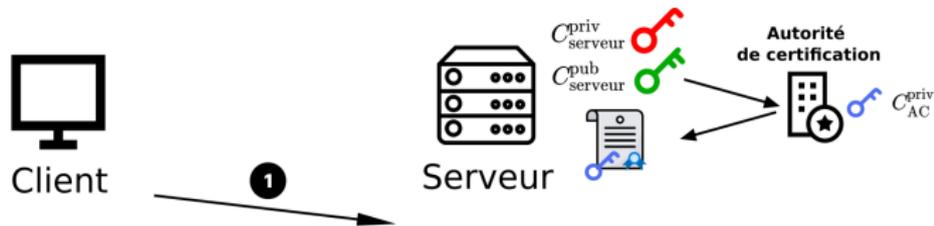
Client



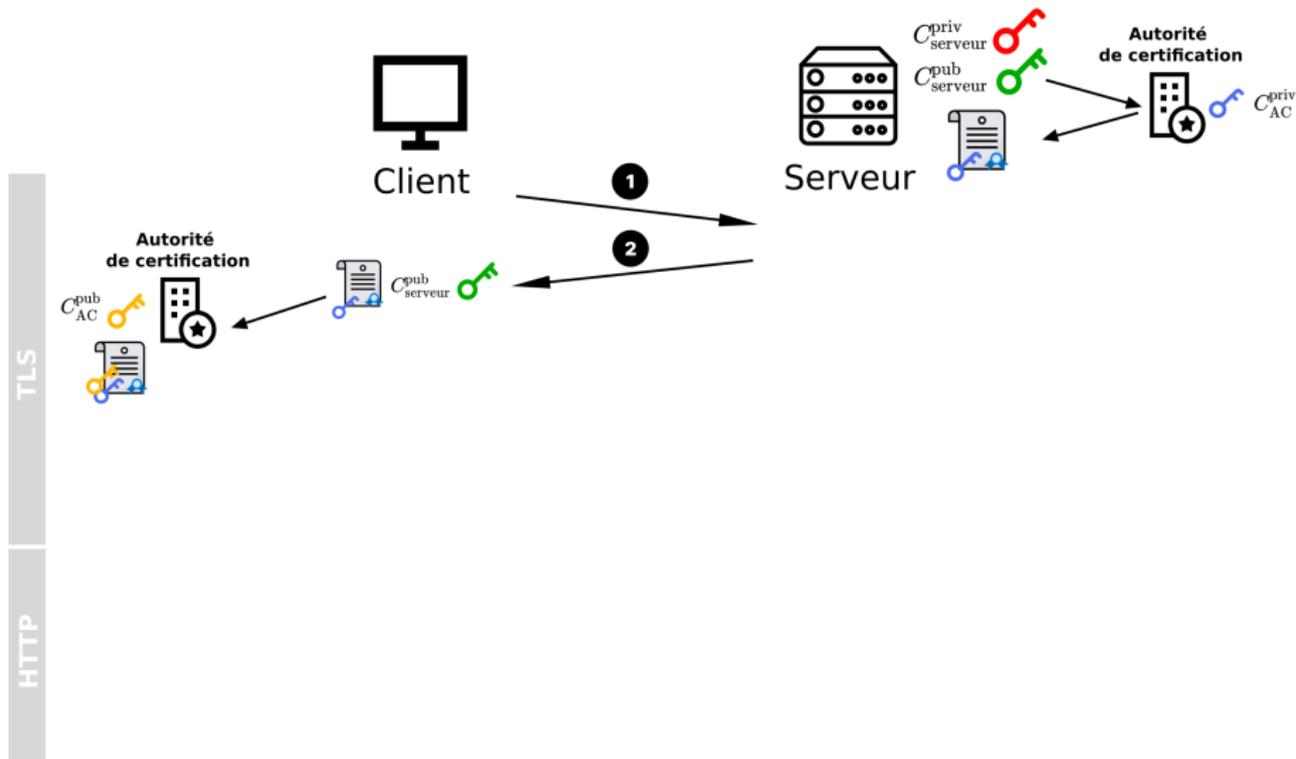
TLS

HTTP

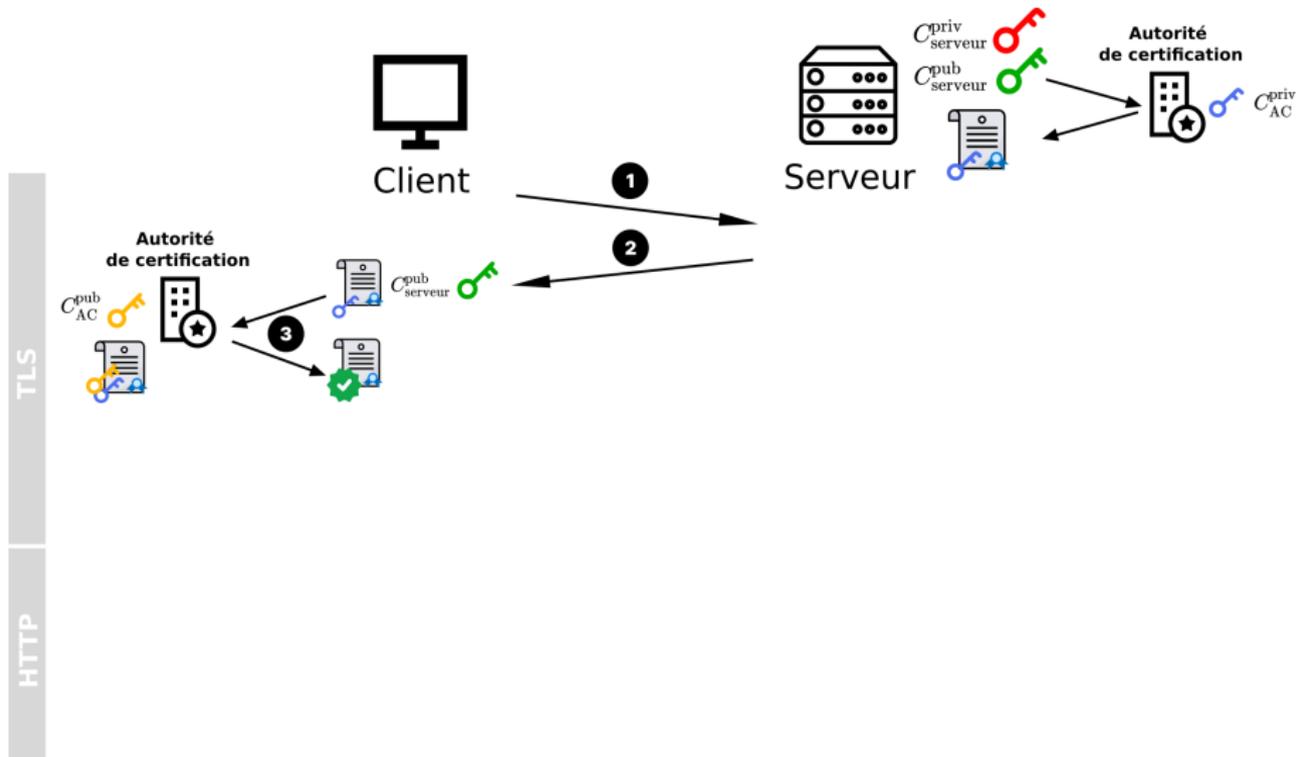
Phase de mise en place, appelée « poignée de main TLS » (ou *TLS Handshake* en anglais).



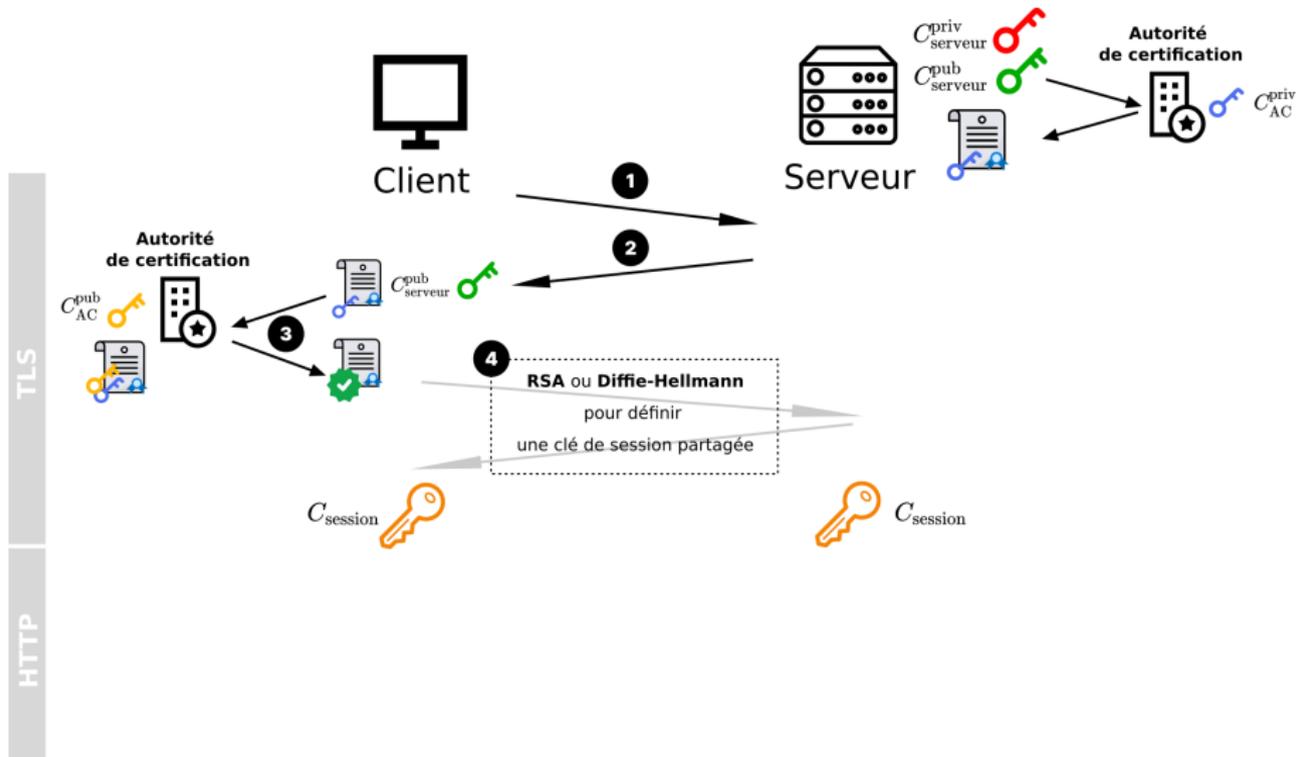
Phase de mise en place, appelée « poignée de main TLS » (ou *TLS Handshake* en anglais).



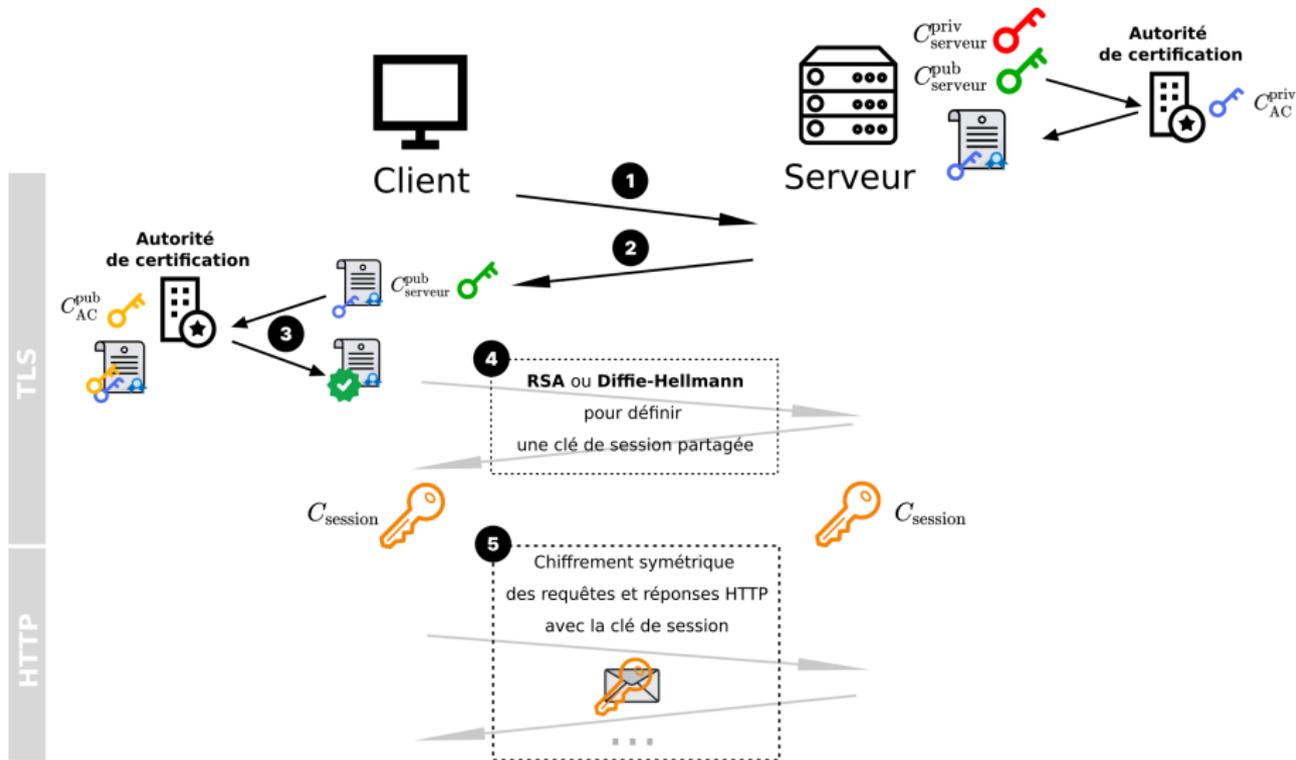
Phase de mise en place, appelée « poignée de main TLS » (ou *TLS Handshake* en anglais).



Phase de mise en place, appelée « poignée de main TLS » (ou *TLS Handshake* en anglais).



Phase de mise en place, appelée « poignée de main TLS » (ou *TLS Handshake* en anglais).

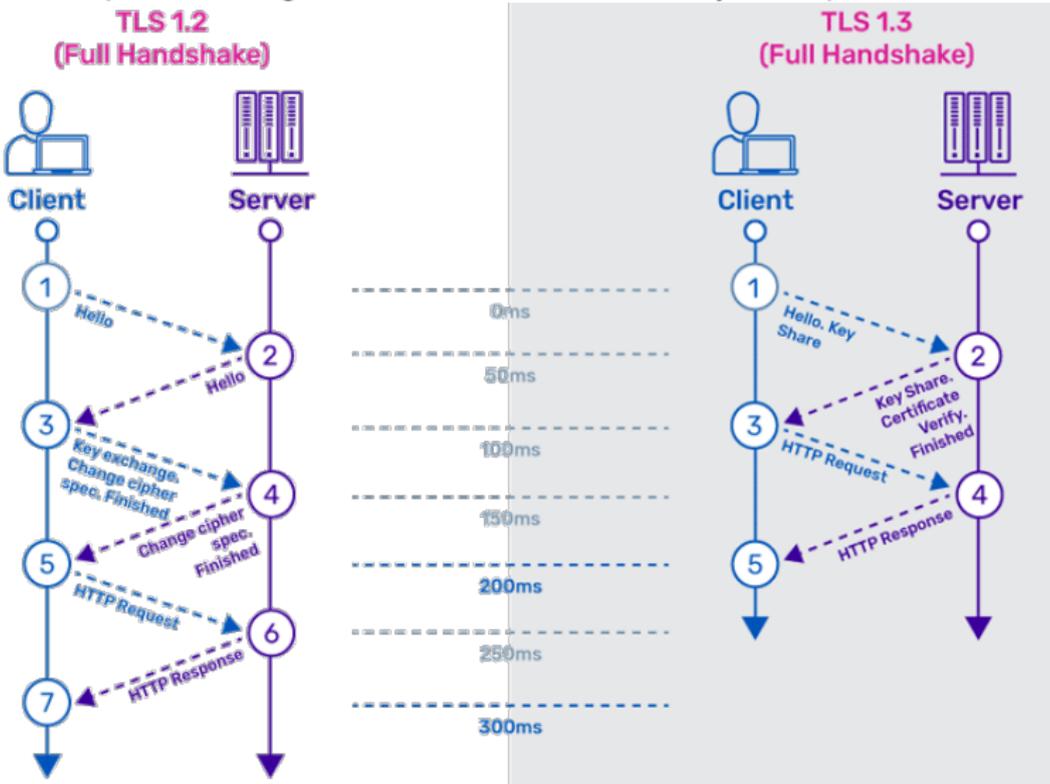


Phase de mise en place, appelée « poignée de main TLS » (ou *TLS Handshake* en anglais).

Client et serveur entame la phase initiale sur le port 443.

- 1 Le **client envoie un message initial** (nommé *Hello*) indiquant les différents algorithmes cryptographiques qu'il peut utiliser, ainsi que d'autres paramètres techniques.
- 2 Le **serveur renvoie sa réponse**, contenant entre autres le certificat contenant sa clé publique, signée par l'autorité de certification (AC).
- 3 Le **client vérifie le certificat** au moyen de la clé publique de l'AC.
- 4 Le **client** et le **serveur conviennent d'une clé de session** pour un algorithme symétrique. Il peuvent choisir :
 - de chiffrer une clé choisie par le client avec la clé publique du serveur ;
 - ou utiliser le protocole Diffie-Hellman pour convenir d'une clé de session partagée.
- 5 le serveur est authentifié par le client et les deux ont convenu d'une clé de session ; ils peuvent donc **échanger de manière sûre les message du protocole HTTP** en les chiffrant.

Remarque : Le protocole TLS 1.3 impose l'utilisation du protocole de Diffie-Hellman pour échanger une clé de chiffrement symétrique AES.



Informations sur la page - https://snt-nsi.info

Général Médias Permissions Sécurité

Identité du site web

Site web : snt-nsi.info **Nom de domaine**

Propriétaire : Ce site web ne fournit pas d'informations sur son propriétaire.

Vérifiée par : Let's Encrypt **CA** [Afficher le certificat](#)

Vie privée et historique

Ai-je déjà visité ce site web auparavant ? Oui, 217 fois

Ce site web conserve-t-il des informations sur mon ordinateur ? Non [Effacer les cookies et les données de sites](#)

Ai-je un mot de passe enregistré pour ce site web ? Non [Voir les mots de passe enregistrés](#)

Détails techniques

Connexion chiffrée : clés TLS_AES_128_GCM_SHA256, 128 bits, TLS 1.3 **Protocoles de chiffrement**

La page actuellement affichée a été chiffrée avant d'avoir été envoyée sur Internet.

Le chiffrement rend très difficile aux personnes non autorisées la visualisation de la page durant son transit entre ordinateurs. Il est donc très improbable que quelqu'un puisse lire cette page durant son transit sur le réseau.

Ce site web respecte la politique de transparence du certificat.

?

Certificats problématiques



Attention : risque probable de sécurité

Firefox a détecté une menace de sécurité potentielle et n'a pas poursuivi vers **self-signed.badssl.com**.
Si vous accédez à ce site, des attaquants pourraient dérober des informations comme vos mots de passe, e-mails, ou données de carte bancaire.

[En savoir plus...](#)

Retour (recommandé)

Avancé...

Certains problèmes peuvent survenir au moment de la poignée de main :

- utilisation d'un certificat non signé ;
- certificat signé par une entité inconnue.

→ La communication est chiffrée mais **pas authentifiée** !

Application 1 : Poignée de main TLS

On se place dans le contexte d'une poignée de main TLS entre client et serveur, telle que présentée précédemment.

- Le client est un navigateur Web classique, comme *Firefox*.
- Le serveur est configuré sur une machine dont le nom de domaine est `www.monsite.fr`.

Dans chacune des situations suivantes, dis quelles étapes de la *poignée de main TLS* échoue.

- 1 Le serveur web n'est pas configuré pour supporté le protocole HTTPS et ne sert des pages qu'en HTTP.
- 2 Le fichier contenant le certificat côté serveur est périmé.
- 3 L'utilisateur du navigateur pointe ce dernier vers l'URL `http://www.monsite.fr:443`.
- 4 L'administrateur du serveur a crée une paire de clé publique et privée, a signé le certidicat que le serveur envoie aux clients et effacé les clés.
- 5 Le navigateur commence à afficher le page de garde du site. Le câble connectant le serveur au réseau est coupé.

Application 1 : Poignée de main TLS

- 1 L'étape 1 échoue immédiatement, le navigateur ne trouve aucun serveur en écoute sur le port 443, la connexion TCP ne peut pas se mettre en place.
- 2 L'étape 3 échoue, le client ne procède pas à la validation du certificat.
- 3 L'étape 2 échoue car le client navigue vers le port HTTPS en HTTP, il n'envoie pas les bons paquets et le serveur ne peut donc pas répondre ou répond un message d'erreur.
- 4 L'étape 3 échoue et le navigateur affiche un message mentionnant un risque probable de sécurité. En effet, le navigateur ne pourra pas trouver une clé publique d'AC lui permettant de vérifier la signature.
- 5 Si la page de garde du site s'affiche, c'est que la requête HTTP récupérant la page a reçu une réponse du serveur. On est après l'étape 5, la connexion TCP est interrompue (car tous les paquets sont perdus).

Application 2 : Certificat et autorité de certification

- 1 Lance *Firefox* et ouvre les outils de développement (raccourci F12) puis va sur l'onglet *Réseau*.
- 2 Rend-toi à l'adresse `www.wikipedia.fr` puis clique dans l'onglet *Réseau* sur la première requête envoyée et va dans l'onglet *Sécurité* (tout à droite).
Retrouve les informations suivantes :
 - La version du protocole TLS utilisé
 - Le nom de l'autorité de certification
 - La période de validité du certificat d'identité
 - L'algorithme utilisé pour la signature
 - L'algorithme utilisé pour la suite du chiffrement
- 3 Retrouvez ces informations à partir de l'icône Cadenas à côté de la barre d'URL.

Bilan

- La sécurisation des communications sur internet repose sur l'utilisation de la **cryptographie**.
- Les algorithmes de chiffrement symétrique utilise la même clé pour chiffrer et déchiffrer des messages.
 - Le chiffrement AES offre un moyen sûr et efficace pour échanger des données.
- Les algorithmes de chiffrement asymétrique reposent sur l'utilisation d'une paire de clés publique et privée, ne nécessitant pas l'échange d'un secret partagé.
 - Le protocole de Diffie-Hellman permet aux participants d'échanger la clé symétrique de façon sûr.
 - Le système RSA permet d'authentifier les participants par le biais d'un tiers de confiance et de certificat.
- Le protocole HTTPS utilise le protocole TLS pour ajouter au protocole HTTP une phase d'authentification et d'échange de clé de chiffrement.