LES PROTOCOLES DE SÉCURISATION DES COMMUNICATIONS

L'utilisation de protocoles permettant de sécuriser les communications est fondamentale. Leur but est de garantir la confidentialité des données (chiffrement des données), l'authentification des serveurs (certificats) et, de plus, ils incluent des mécanismes garantissant l'intégrité des données. Quand on navigue sur Internet, on voit souvent un cadenas 🔒 à côté de l'URL. Ce symbole indique que la connexion est sécurisée grâce au protocole TLS (Transport Layer Security).

1. Mais TLS c'est quoi ?

TLS (Transport Layer Security) est un protocole de chiffrement qui assure la sécurité de bout en bout des données échangées entre applications sur Internet. On peut identifier son utilisation dans le navigateur web grâce à l'icône de cadenas. Il peut également être utilisé pour d'autres applications telles que la messagerie électronique, les transferts de fichiers, la voix sur IP, ainsi que pour des services Internet comme DNS et NTP.

TLS est une évolution du protocole SSL (Secure Socket Layers), initialement développé par Netscape Communications Corporation en 1994 pour sécuriser les sessions web. TLS a été spécifié dans la norme RFC 2246 en 1999 comme protocole indépendant des applications (SSL 3.0 est désormais considéré comme non sécurisé et a été déconseillé depuis juin 2015).

TLS ne sécurise pas les données sur les terminaux. Il assure simplement la transmission sécurisée des données sur Internet, évitant ainsi toute lecture illicite ou altération du contenu.

TLS est implémenté sur TCP afin de chiffrer les protocoles de la couche application tels que HTTP, FTP, SMTP et IMAP.

2. Fonctionnement du protocole TLS

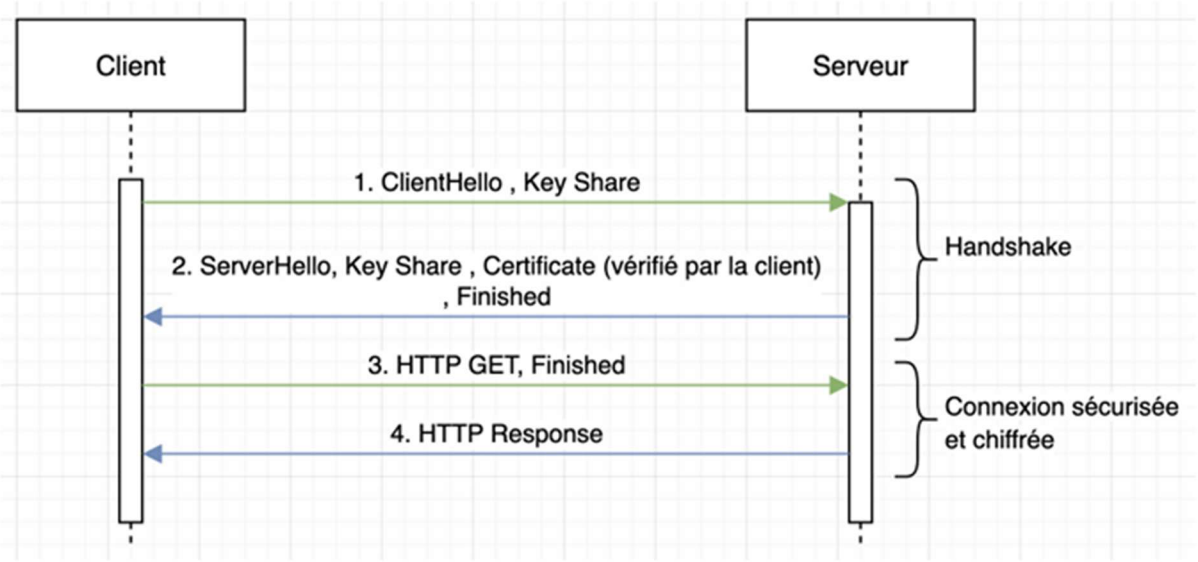
TLS repose sur des mécanismes de chiffrement avancés, combinant cryptographie symétrique et asymétrique, ainsi que sur un système de certificats numériques permettant d'authentifier les serveurs.

Chiffrement symétrique	Chiffrement asymétrique
Une même clé secrète est utilisée pour chiffrer et déchiffrer les données.	Utilise deux clés différentes : une publique (connue de tous) et une privée (gardée secrète).
Elle est rapide et efficace, mais nécessite que les deux parties aient déjà la clé de façon sécurisée.	Permet à quelqu'un de chiffrer un message avec la clé publique du destinataire, que seul le destinataire peut déchiffrer avec sa clé privée.
Processus rapide	Processus lent

TLS commence par utiliser le chiffrement asymétrique pour échanger une clé de session. Ensuite, il passe au chiffrement symétrique pour échanger les données rapidement. La clé de session est temporaire : elle est supprimée à la fin de la communication.

TLS permet également qu'un client se connectant à un serveur puisse valider la propriété de la clé publique du serveur (permet d'être sûr de l'identité du serveur avec lequel on dialogue). Cette validation s'effectue généralement à l'aide d'un certificat numérique X.509 émis par une autorité de certification (AC) tierce, qui atteste de l'authenticité de la clé publique.

Protocole TLS 1.3



1. Le client envoie au serveur le message ClientHello qui contient :

- Une liste de chiffrements (algorithmes de sécurité) qu'il supporte ;
- Sa clé publique éphémère (via Key Share) pour établir un secret partagé.

C'est le début de la négociation sécurisée.

2. Le serveur répond avec un ServerHello indiquant les paramètres de connexion négociés :

- Les algorithmes compatibles et sa propre clé publique éphémère(Key Share) ;
- Son certificat pour prouver son identité (ex : certif. HTTPS signé par une autorité de confiance) ;
- Il termine le handshake en envoyant Finished (valide l'échange des clés).

À la fin de cette étape, les deux côtés ont un secret commun (clé de session) utilisé pour chiffrer les données.

3. Le client combine son message Finished avec sa première requête chiffrée (HTTP GET dans le schéma) dans le même message TLS.

4. À ce stade, le handshake TLS est terminé, la clé de session est en place. Toute la communication est désormais intégralement chiffrée. Même les métadonnées comme les en-têtes HTTP sont chiffrées (contrairement à HTTPS en TLS 1.2 ou certains champs pouvaient être visibles).

C'est une réponse HTTP « classique » mais encapsulée dans TLS.

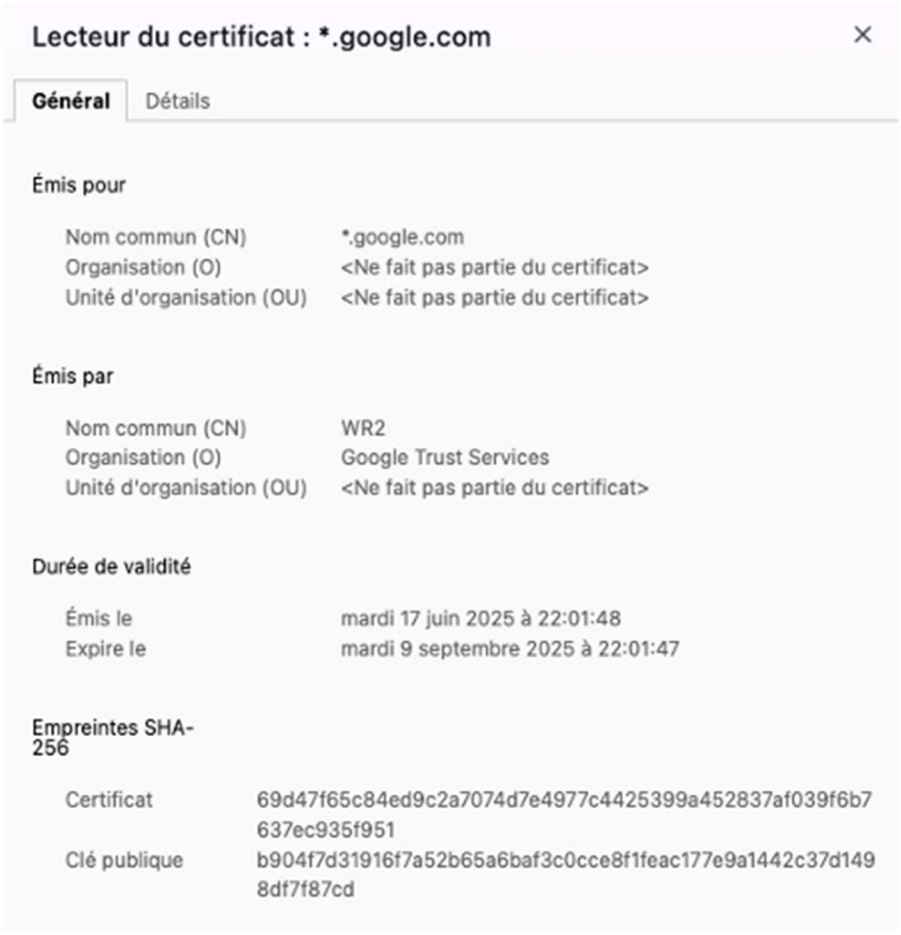
Pour garantir l'authenticité d'un serveur, celui-ci utilise un certificat numérique X.509 délivré par une autorité de certification (AC) reconnue. Cette autorité agit comme un tiers de confiance et atteste que le serveur est bien celui qu'il prétend être. Dans certains cas, un serveur peut utiliser un certificat auto-signé. Cela signifie qu'il s'est « signé lui-même » sans passer par une autorité.

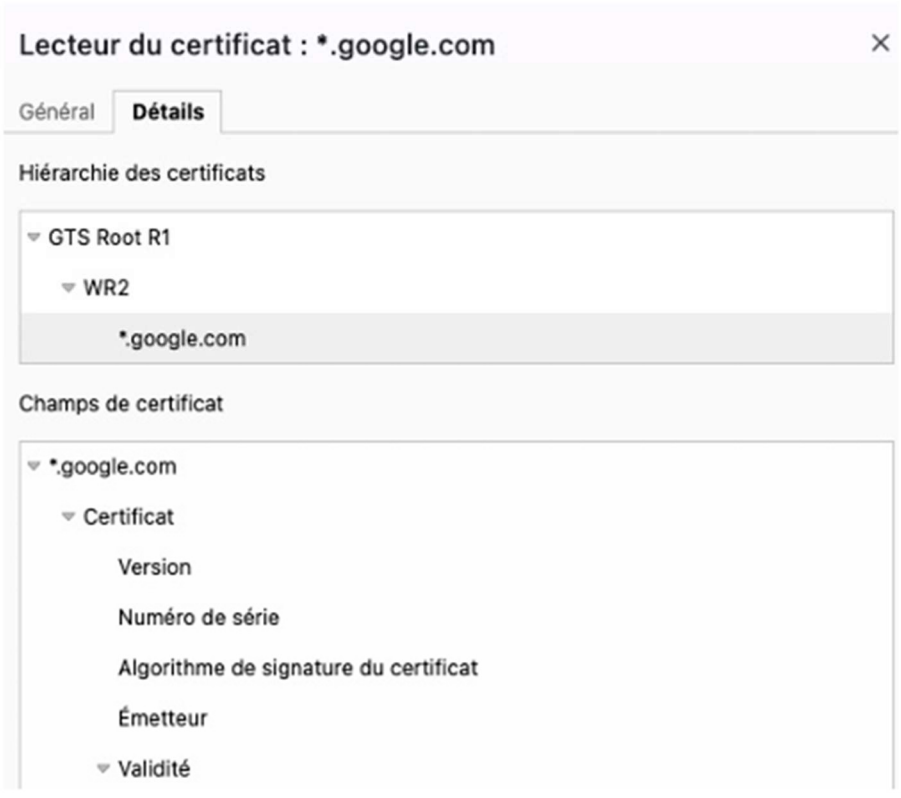
Dans ce cas, le client (ex : navigateur web) n'a aucun moyen de vérifier automatiquement la fiabilité du serveur. C'est pourquoi un avertissement de sécurité est affiché.

Un certificat X.509, c'est un fichier, généralement au format : pem, .crt ou .cer, qui contient :

- La clé publique du serveur ;
- L'identité du propriétaire (ex. nom de domaine, organisation, etc.) ;
- La signature numérique de l'autorité de certification (AC) ;
- La période de validité ;
- Le numéro de série ;
- L'algorithme utilisé (RSA, ECC…).

Voici un exemple de certificat :


Lecteur du certificat : *.google.com
Général
Détails
Émis pour
Nom commun (CN) *.google.com
Organisation (O) <Ne fait pas partie du certificat>
Unité d'organisation (OU) <Ne fait pas partie du certificat>
Émis par
Nom commun (CN) WR2
Organisation (O) Google Trust Services
Unité d'organisation (OU) <Ne fait pas partie du certificat>
Durée de validité
Émis le mardi 17 juin 2025 à 22:01:48
Expire le mardi 9 septembre 2025 à 22:01:47
Empreintes SHA-256
Certificat 69d47f65c84ed9c2a7074d7e4977c4425399a452837af039f6b7637ec935f951
Clé publique b904f7d31916f7a52b65a6baf3c0cce8f1feac177e9a1442c37d1498df7f87cd


Lecteur du certificat : *.google.com
Général
Détails
Hiérarchie des certificats
GTS Root R1
WR2
*.google.com
Champs de certificat
*.google.com
Certificat
Version
Numéro de série
Algorithme de signature du certificat
Émetteur
Validité