# Comment fonctionne HTTPS

Me permettez de vous expliquer comment fonctionne HTTPS, y compris SSL, les certificats, le processus de chiffrement et les différences entre TLS 1.2 et TLS 1.3, de manière claire et étape par étape.

---

### Qu'est-ce que HTTPS ?

HTTPS signifie **Hypertext Transfer Protocol Secure**. Il s'agit de la version sécurisée de HTTP, le protocole utilisé pour transférer des données entre votre navigateur et un site web. La partie "sécurisée"provient de l'ajout d'une couche de chiffrement et d'authentification, fournie par **SSL (Secure Sockets Layer)** ou son remplacement moderne, **TLS (Transport Layer Security)**. Donc, HTTPS est essentiellement HTTP fonctionnant sur TLS (ou historiquement SSL) pour garder vos données en sécurité.

---

### Comment fonctionne HTTPS ? La poignée de main TLS

Lorsque vous visitez un site web utilisant HTTPS (par exemple, `https://example.com`), votre navigateur et le serveur du site établissent une connexion sécurisée avant d'échanger des données. Ce processus s'appelle la **poignée de main TLS**. Voici comment cela fonctionne en étapes simples :

1. **Client Hello** :
    - Votre navigateur envoie un message au serveur disant, "Bonjour ! Voici la version TLS que je supporte (par exemple, TLS 1.3), les algorithmes de chiffrement (suites de chiffrement) que je peux utiliser, et une chaîne aléatoire de bytes (client aléatoire)."
2. **Server Hello** :
    - Le serveur répond, "Bonjour en retour ! J'utiliserai cette version TLS et cette suite de chiffrement de votre liste. Voici ma chaîne aléatoire (server aléatoire)."
3. **Certificat** :
    - Le serveur envoie son **certificat SSL**, qui inclut sa **clé publique** et est signé par une **Autorité de Certification (CA)** de confiance. Ce certificat prouve l'identité du serveur.
4. **Échange de clé client** :
    - Votre navigateur vérifie le certificat pour s'assurer qu'il est valide et signé par une CA de confiance. S'il passe, le navigateur génère un **secret pré-maître**, le chiffre avec la clé publique du serveur, et l'envoie au serveur.
5. **Clés de session** :

- À la fois le navigateur et le serveur utilisent le client aléatoire, le serveur aléatoire et le secret pré-maître pour générer indépendamment la même **clé de session**. Cette clé est utilisée pour chiffrer et déchiffrer toutes les données pendant la session.

6. **Terminé** :
    - Les deux côtés envoient un message “terminé”, chiffré avec la clé de session, pour confirmer que la connexion sécurisée est prête.

Une fois la poignée de main terminée, toutes les données (comme les pages web, les formulaires ou les fichiers) sont chiffrées avec la clé de session, les rendant illisibles pour quiconque pourrait les intercepter.

---

**Qu’est-ce que les certificats SSL et comment fonctionnent-ils ?**

Un **certificat SSL** est un document numérique qui prouve l’identité d’un site web et permet le chiffrement. Voici ce que vous devez savoir :

- **Contenu** : Le certificat inclut le nom de domaine du site web, sa clé publique et une signature numérique d’une **Autorité de Certification (CA)**.
- **But** : Il garantit que le serveur est légitime (par exemple, vous vous connectez vraiment à `example.com`, pas à un site faux) et fournit la clé publique pour le chiffrement.
- **Vérification** : Votre navigateur vérifie :
    1. Le certificat est-il valide (non expiré ou révoqué) ?
    2. Est-il signé par une CA de confiance ? (Les navigateurs ont une liste intégrée de CA de confiance, comme DigiCert ou Let’s Encrypt.)
- Si les vérifications passent, le navigateur fait confiance au serveur et poursuit la poignée de main.

La CA agit comme un tiers de confiance vouchant pour le site web. Sans cela, les attaquants pourraient prétendre être n’importe quel site et voler vos données.

---

**L’algorithme de chiffrement**

Le chiffrement dans HTTPS repose sur une combinaison de **cryptographie asymétrique** et **symétrique** :

1. **Cryptographie asymétrique** (pendant la poignée de main) :
    - Utilise une **clé publique** (partagée ouvertement) et une **clé privée** (gardée secrète par le serveur).

- Le navigateur chiffre le secret pré-maître avec la clé publique du serveur. Seul le serveur, avec sa clé privée, peut le déchiffrer.
- Exemples d'algorithmes : RSA ou Cryptographie à Courbes Elliptiques (ECC).

2. **Cryptographie symétrique** (pour la session) :
   - Une fois la clé de session créée, les deux côtés l'utilisent pour chiffrer et déchiffrer les données.
   - Cela est plus rapide que le chiffrement asymétrique et idéal pour les transferts de données volumineux.
   - Exemple d'algorithme : AES (Advanced Encryption Standard).

La poignée de main utilise le chiffrement asymétrique pour partager la clé de session de manière sécurisée, puis le chiffrement symétrique prend le relais pour l'efficacité.

---

**Différences entre TLS 1.2 et TLS 1.3**

**TLS 1.2** et **TLS 1.3** sont des versions du protocole TLS, TLS 1.3 étant la version plus récente et améliorée. Voici les principales différences :

| Fonctionnalité | TLS 1.2 | TLS 1.3 |
|---|---|---|
| **Vitesse de poignée de main** | Plusieurs allers-retours entre le client et le serveur, ajoutant de la latence. | Moins d'allers-retours, souvent juste un, pour des connexions plus rapides. Peut envoyer des données immédiatement ("0-RTT") lors des reconnexions. |
| **Suites de chiffrement** | Prend en charge les algorithmes de chiffrement plus anciens et moins sécurisés (par exemple, SHA-1). | Supprime les suites de chiffrement obsolètes, n'utilise que les modernes et sécurisées (par exemple, AES-GCM). |
| **Confidentialité** | Certaines données de poignée de main (comme les certificats) sont envoyées en texte brut. | Presque toute la poignée de main est chiffrée, cachant plus de détails aux écouteurs. |
| **Rétablissement 0-RTT** | Non disponible. | Permet l'envoi instantané de données lors de la reconnexion, mais risque des attaques par replay si pas sécurisé correctement. |
| **Sécurité** | Vulnérable à certaines attaques en raison de fonctionnalités plus anciennes. | Plus robuste par conception, en supprimant les options faibles. |

En résumé, TLS 1.3 est **plus rapide**, **plus sécurisé** et **plus privé** que TLS 1.2, ce qui en fait la norme pour le HTTPS moderne.

---

**Tout rassembler**

HTTPS sécurise la communication web en : 1. Utilisant la **poignée de main TLS** pour établir une connexion chiffrée. 2. S'appuyant sur les **certificats SSL** provenant de CA de confiance pour vérifier l'identité du serveur. 3. Combinaison de **cryptographie asymétrique** (pour la poignée de main) et de **cryptographie symétrique** (pour le transfert de données) pour protéger vos données. 4. Évoluant avec des protocoles comme **TLS 1.3**, qui améliore TLS 1.2 avec une meilleure vitesse, sécurité et confidentialité.

C'est le cœur de la manière dont HTTPS garde votre navigation en sécurité ! Faites-moi savoir si vous souhaitez que je plonge plus profondément dans une partie.