Sécurité au niveau transport (Transport Layer Security)

- SSL/TLS
- SSL/TLS Structure
- Sous-protocoles de SSL
- Mémorisation des sessions
- Ports utilisés par SSL
- Liens

Sécurité au niveau transport

Lorsque l'on veut offrir un seul service sécurisé (web, mail, login), il est plus intéressant de sécuriser les données au niveau transport qu'au niveau réseau.

Les deux protocoles les plus utilisés sont :

- SSH: qui permet de faire des logins sécurisés.
- SSL/TLS: permet de sécuriser n'importe quel service basé sur TCP (https, pop3s, esmtp,...)

Exemple de crypto systèmes : SSL/TLS

- **SSL** (Secure Socket Layer) est un protocole de sécurisation des échanges au niveau de la couche application. Il est implémenté au dessus de la couche TCP.
- **SSL permet d'assurer les** services de sécurité **suivants:**
  - confidentialité : assurée par les algorithmes à chiffrement symétrique de blocs comme DES, 3DES, ou AES
  - intégrité : assurée par l'utilisation de MAC (Message Authentication Code) basés sur les fonctions de hachage MD5 (16 octets) ou SHA-1 (20 octets).
  - authentification : SSL permet l'authentification des 2 entités (authentification client facultative) basé sur des certificats, et l'authentification des données grâce aux MAC.

Protocole SSL

- Services de SSL:
  - Authentification (forte) avec certificats,
  - Chiffrement,
  - Compression,
  - Négociation des algorithmes d'authentification/cryptage,
  - Gestion de clés de session.

- La version actuelle de SSL est la 3.1

TLS : Transport Security Layer

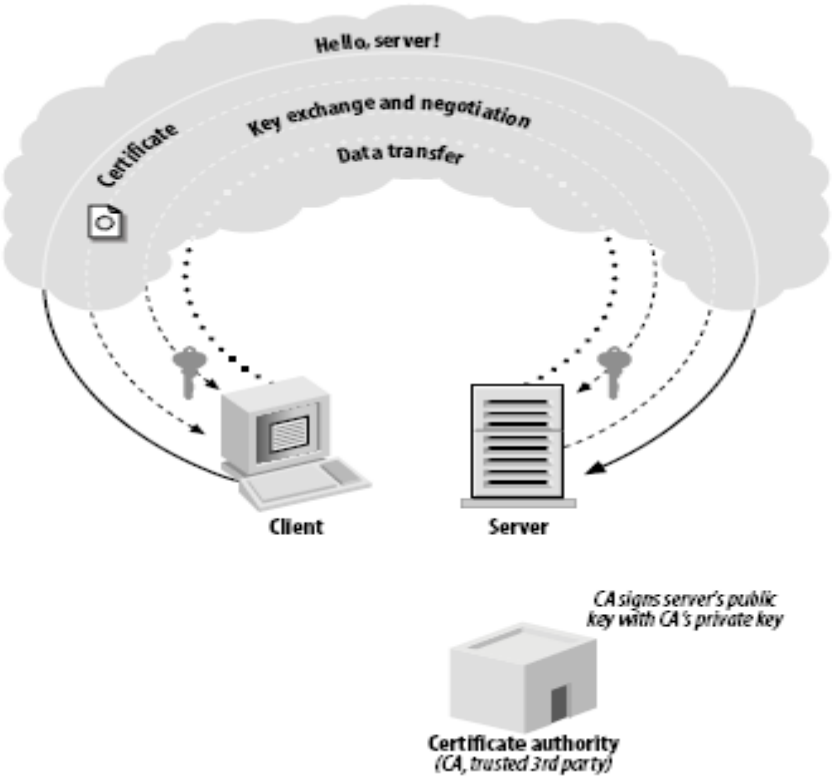
- Netscape a remis le développement de SSL à l'IETF (Task Force d'Internet)

- TLS 1.0 est la nouvelle version de SSL

- TLS 1.0 est parfois appelé SSL 3.1

- Tous les serveurs web et browser récents supportent TLS 1.0

Échanges SSL/TLS



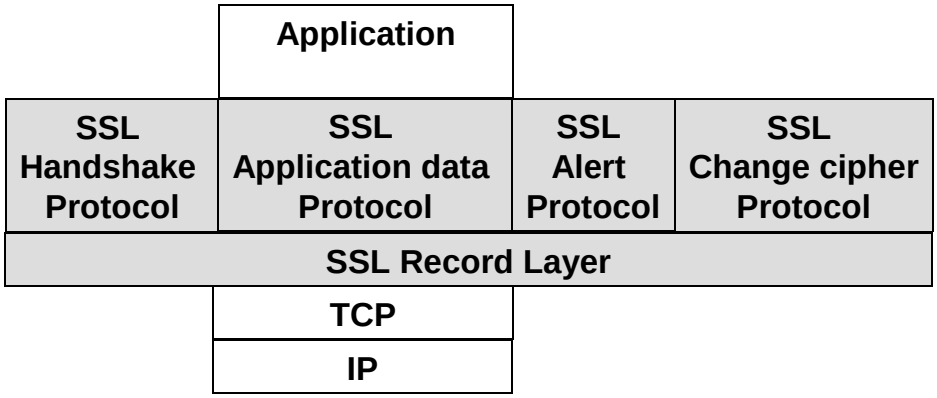
Source : chapitre de livre *Network Security with OpenSSL, O'Reilly Editions.*

SSL/TLS structure

- TLS s'insère entre la couche application et la couche transport (TCP)

- TLS contient deux couches:
  - La couche inférieure (record layer) qui applique les mesures de sécurité au données échangées
  - La couche supérieure qui traite quatre types de messages

SSL/TLS structure



Les sous-protocoles de SSL

1) **Handshake** : permet l'authentification mutuelle du client et serveur, la négociation des algorithmes de chiffrement, de hachage, et l'échange des clés symétriques.

2) **SSL Change Cipher Spec** : indique au protocole Record Layer la mise en place des algorithmes de chiffrement qui viennent d'être négociés.

3) **SSL Record Layer** : permet de garantir la confidentialité et l'intégrité.

4) **SSL Alert** : permet de signaliser à l'application des erreurs ou des avertissements concernant la session en cours.

.

5) **SSL Data Protocol** : passe les données d'une application de manière transparente au record layer.

Protocole Handshake

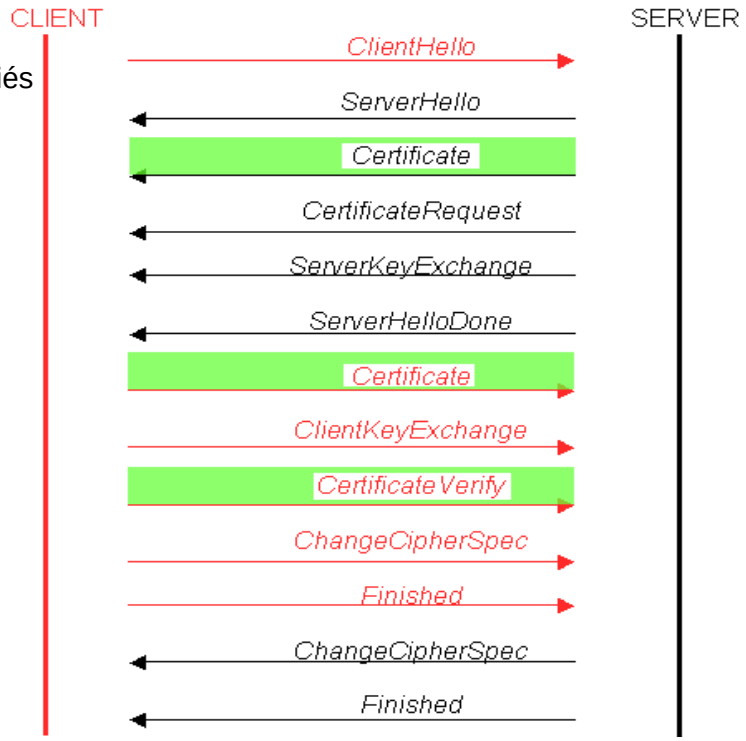
- Négociation de :
  - La version du protocole (SSL 3.1, TLS 1.0)
  - Les algorithmes:
    - ➢ échange de clé (RSA, Diffie-Hellman)
    - ➢ chiffrement (DES, 3DES, IDEA, RC4,…)
    - ➢ authentification (HMAC-MD5, HMAC-SHA)
    - ➢ compression des données
    - ➢ le client propose les algorithmes désirés dans l'ordre de préférence, le serveur choisit

Déroulement d'un handshake SSL (1)

en **noir**, échanges initiés par le serveur.

en **rouge**, échanges initiés par le client.

en **vert**, échanges correspondants à l'authentification



Déroulement d'un handshake SSL (2)

Les échanges se déroulent en deux phases :

1) authentification du serveur :

- **Suite à une requête d'un client, le serveur envoie son certificat, la signature du certificat et la liste des algorithmes cryptographiques, qu'il souhaite négocier.**
- **Le client vérifie la validité du certificat à l'aide de la clé publique du CA contenue dans le navigateur.**
- **Si le certificat est valide, le client génère un pré-master secret (PMS) qui servira à dériver le master secret (MS).**
- **Ce PMS, chiffré avec la clé publique du serveur, est transmis au serveur. Les données échangées entre le client et le serveur sont chiffrées et authentifiées à l'aide de clés dérivées de la clé maître.**

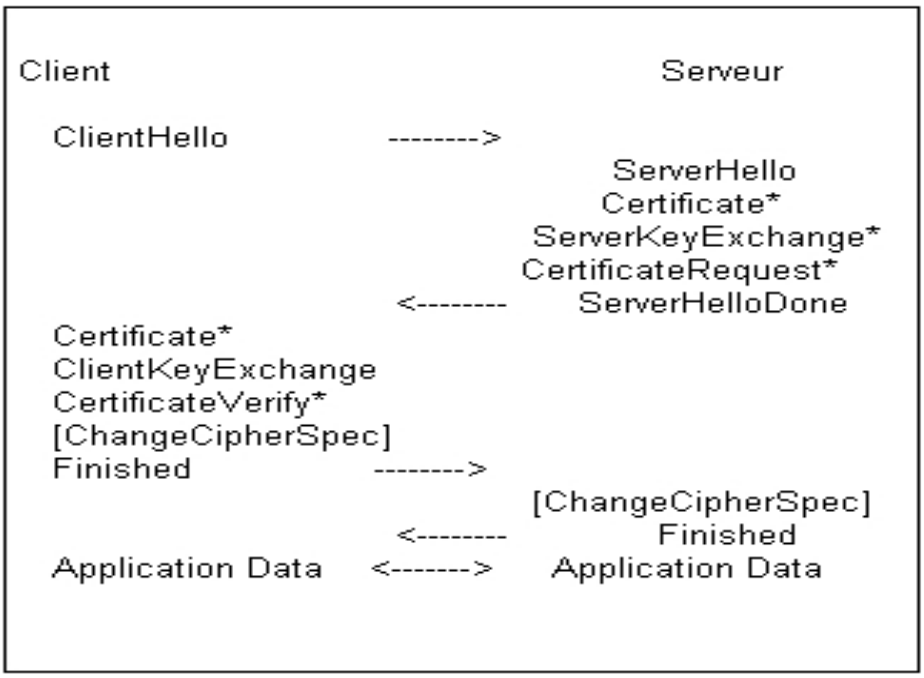
Déroulement d'un handshake SSL (3)

Les échanges se déroulent en deux phases :

2) authentification (optionnelle) du client :

- **Le serveur (et seulement lui) peut demander au client de s'authentifier en lui demandant tout d'abord son certificat.**
- **Le client réplique en envoyant ce certificat.**
- **L'authentification du client est facultative et au bon vouloir du serveur.**

Déroulement d'un handshake SSL (4)



Déroulement d'un handshake SSL (5)

- Client Hello & Server Hello :
  - **version**: version du protocole SSL,
  - **Random**: nombre aléatoire,
  - **session_id**: l'identificateur de session,
  - **Cipher_suite**: liste des suites de chiffrement choisies,
  - **algo** : liste des méthodes de compression.
- Server Key Exchange : contient le certificat de signature.
- Server Hello Done : indique la fin de l'envoi de message.
- Client Key Exchange : contient le ***PreMastersecret*** crypté à l'aide de la clé publique du serveur.
- Finished : fin du protocole Handshake et le début de l'émission des données.

Les ports utilisées par SSL

Pour accéder à un serveur qui utilise SSL, on ajoute un "s" lors de la spécification du protocole.
(https://www.monserveur.com)

HTTPS (HTTP en SSL)	**443**
SMTPS (SMTP en SSL)	**465**
NNTPS	**563**
LDAPS (LDAP en SSL)	**636**
POP3S	**995**
IMAPS	**995**

Quelques Liens

- Network Security with OpenSSL (O'REILLY Éditions)
- http://www.apprendre-en-ligne.net/crypto/activites/index.html
- Chapitre de Livre : la sécurité des systèmes d'information
- Transparents Ph. Oechslin, Sécurité des Réseaux, 2003
- Site La sécurité informatique (www.securiteinfo.com/cryptographie/ssl.shtml)