SSL/TLS

ACHRAF FAYAD

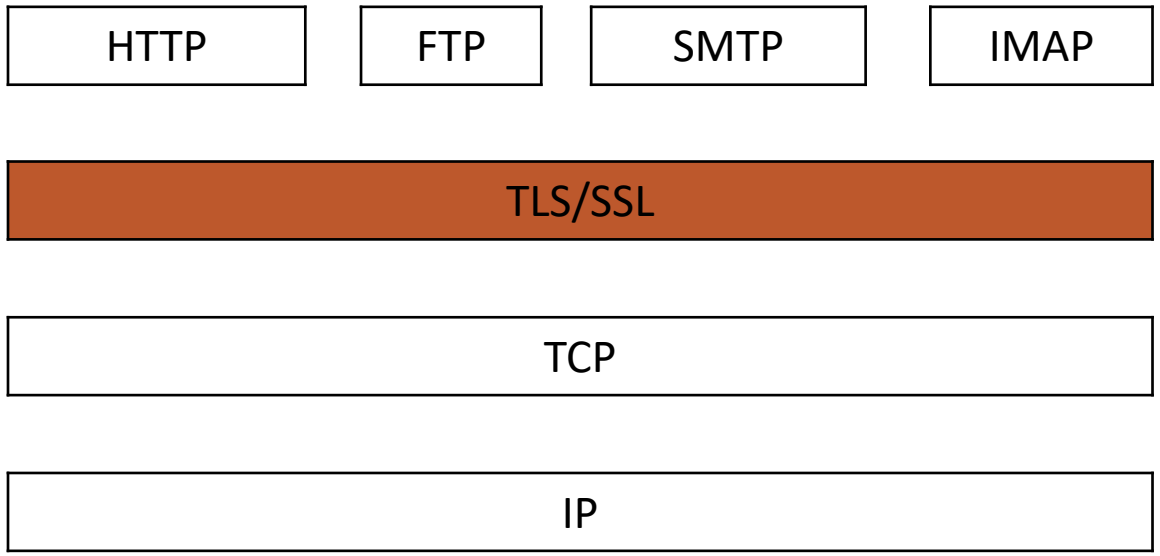
ECE

2017/2018

Introduction TLS/SSL

- It is essentially a protocol that provides a secure channel between two machines operating over the Internet or an internal network.
- The Secure Sockets Layer (SSL) and Transport Layer Security (TLS) is the most widely deployed security protocol used today.
  - Integration in browsers
  - Operation for Application layer
  - Don’t have to make a particular configuration client’s side.
  - Transparent solution to secure the applicative exchange.

Introduction TLS/SSL



Introduction TLS/SSL

SSL concepts and practices, including:

- SSL communications
- Certificate authorities
- Public key infrastructures
- Symmetric and asymmetric key pairs
- Cryptographic hash functions
- Encryption algorithms

Introduction TLS/SSL

SSL was developed by Netscape and integrated in browser Navigator

- SSL V1.0 (early 1994)
  - Tested, but wasn't released
- SSL V2.0 (and of 1994)
  - Supported by browsers, but not recommended
- SSL V3.0 (début 1997)

Attack Bleichenbacher

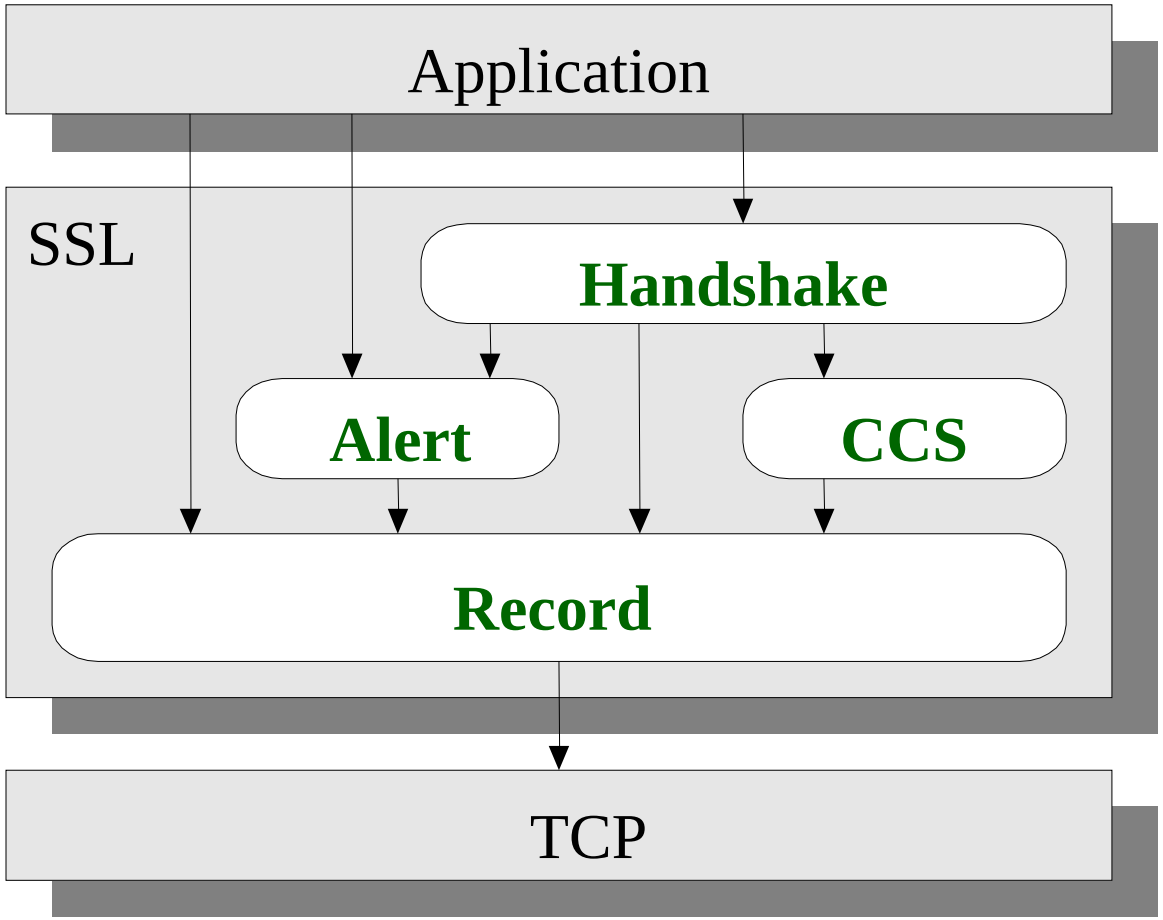
Introduction TLS/SSL

- TLS V1.0 (1999)
  - Transport Layer Secure
  - RFC2246 (Standard)
  - upgraded from SSL 3.0
  - Correction attack Bleichenbacher
  - SSL 3.0 and TLS 1.0 don't interoperate
- TLS V1.1 (2006)
  - RFC 4346 (standard)
  - is an update to TLS 1.0
  - Correction de Attack Rogway (BEAST) (2002)
  - TLS V1.2 August of 2008
- TLS V1.2 (2008)
  - RFC5246 (Standard)
  - Including extensions
  - Add a new suite gryptography (HMAC sha256, …)

TSL/SSL Services

- Authentication
  - Server, Client (optional)
  - Use Certificates X509 V3
- Confidentiality
  - Negotiation of encryption symmetric algorithms, key generated at the session establishment.
- Integrity
  - HMAC (Hash function with a secret cryptographic key)
- replay attack
  - Sequence number

TLS/SSL Protocols

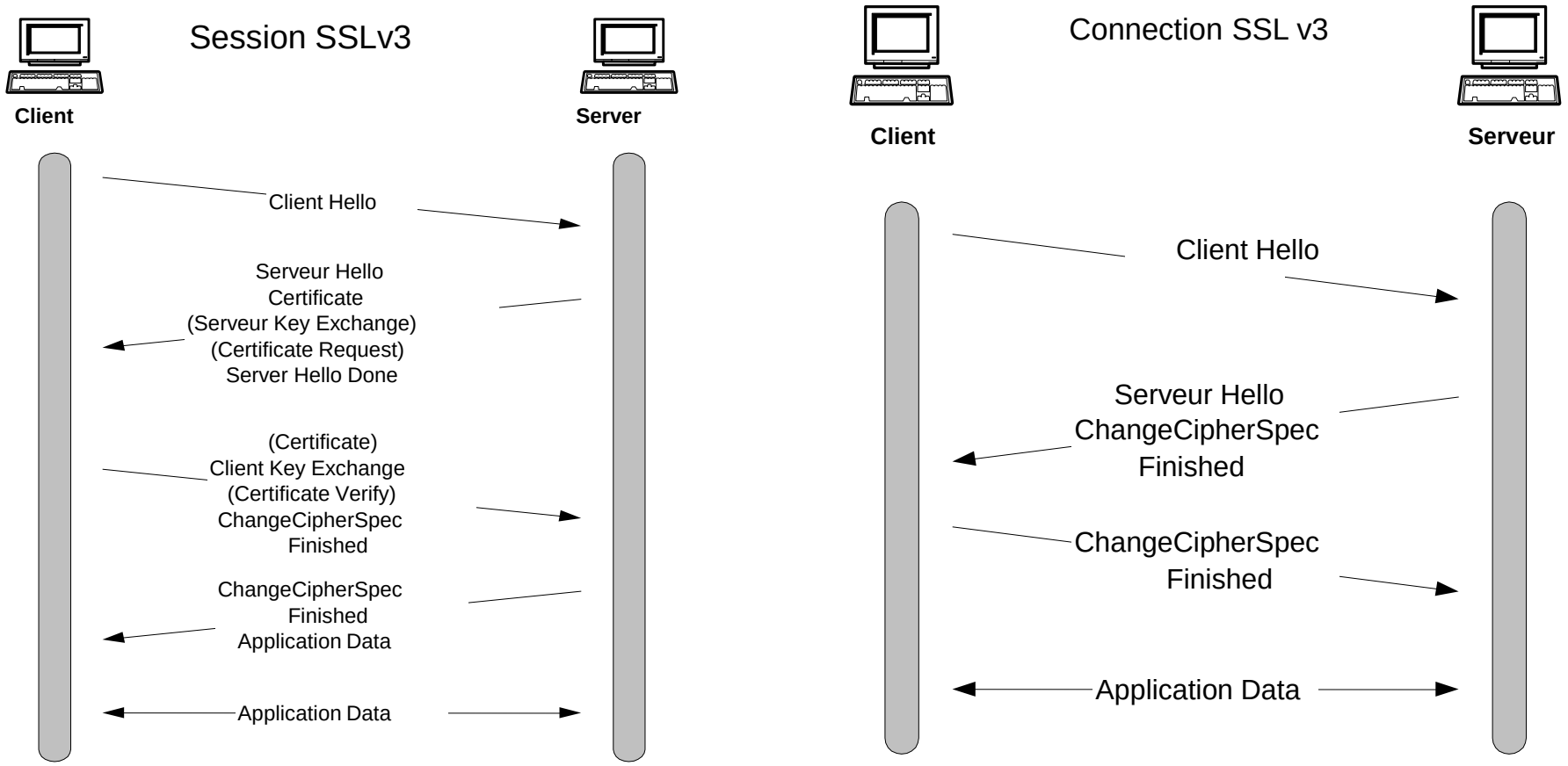


Handshake Protocol

Allow server & client to :

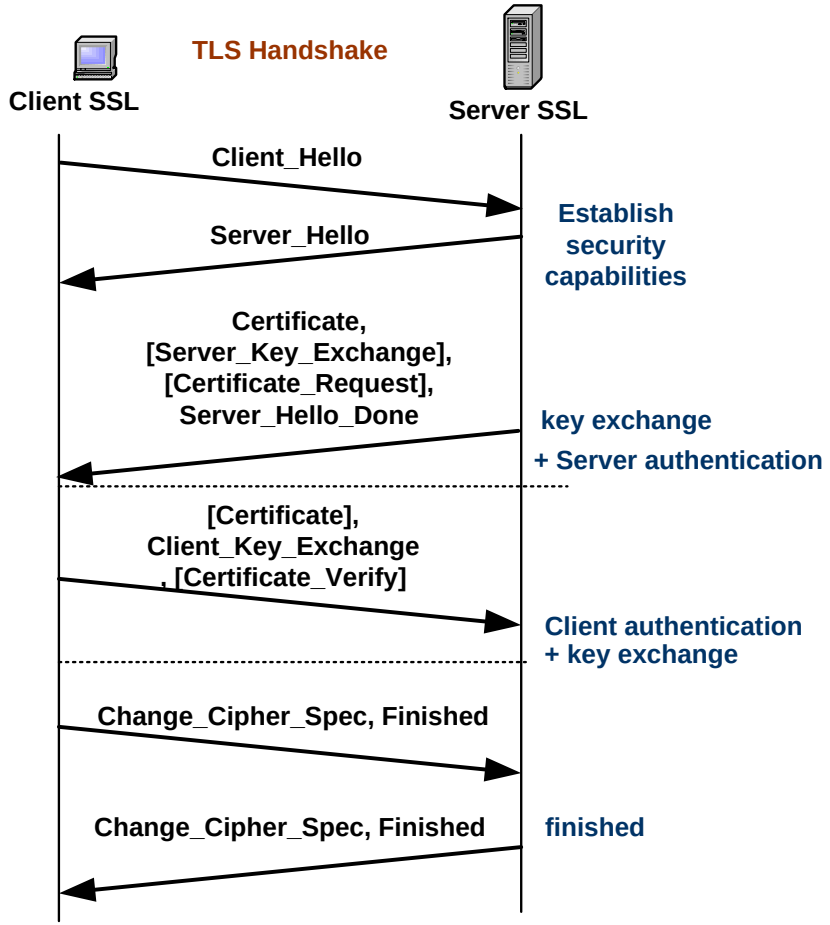
- Authenticate each other
  - Use X509 v3 Certificates (client optional)
- Negotiate encryption algorithm
- Negotiate hash function to be used
- Exchange and generate cryptographic keys to be used

Handshake Protocol



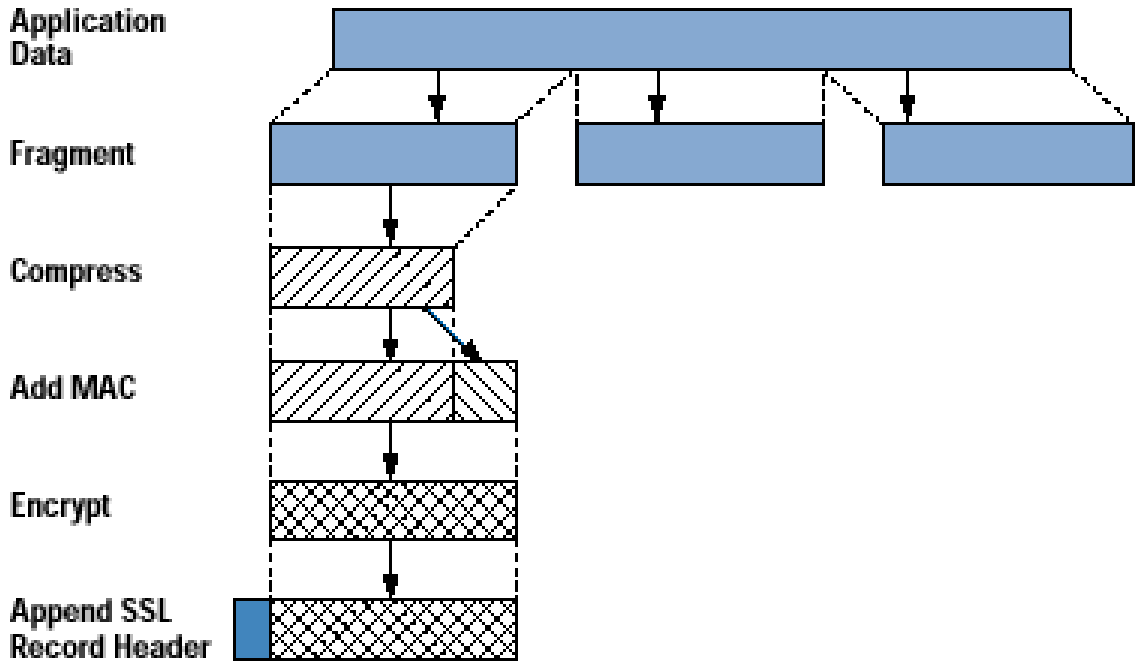
Handshake Protocol

- Handshake Protocol contains 3 phases:
  1. Exchange security parameters and server authentication
  2. Client Authentication (optional)
  3. initializing the data encryption



Record Protocol

- Use symmetric encryption whit a shared secret key defined by handshake protocol
- Receive data from up layers (Hanshake, Alert, CCS, HTTP).



Alert Protocol / CCS

- Alert Protocol signals problems with an SSL session.
  - Alert messages convey the severity of the message and a description of the alert.
  - Upon transmission or receipt of a fatal alert message, both parties immediately close the connection.
- Specific alert:
  - Unexpected message, bad record mac, decompression failure, handshake failure, illegal parameters
  - Close notify, no certificate, bad certificate, ……..
- ChangeCipherSpec (CCS) signals to Record any changes of security parameters.