SSL and TLS

- CEG 6430/4430 Cyber Network Security
- Junjie Zhang
- junjie.zhang@wright.edu
- Wright State University

What is SSL/TLS?

- SSL - Secure Sockets Layer
- TLS - Transport Layer Security

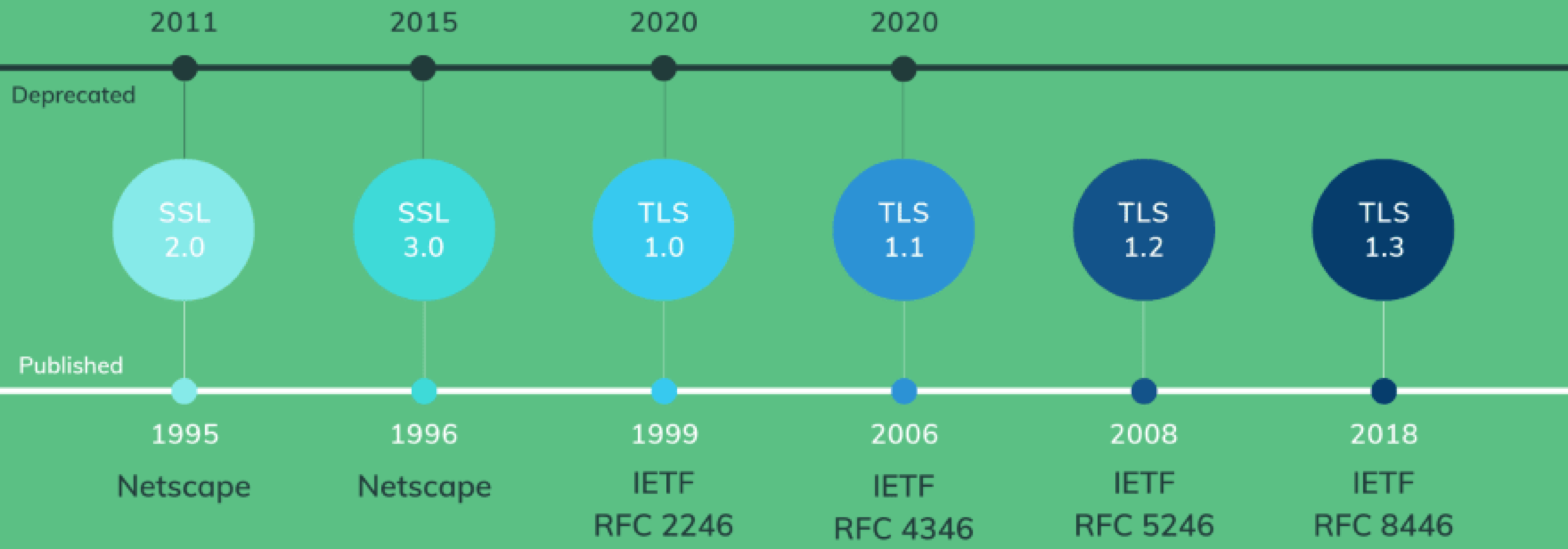
What is SSL/TLS?

SSL/TLS creates a protected tunnel across the Internet capable of offering communications with

- confidentility using encryption,
- integrity using HMAC and digital signature,
- and authenticity using PKI.

But why two names for the same thing?

THE HISTORY OF SSL / TLS



Deprecated	2011	2015	2020	2020		
	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
Published	1995	1996	1999	2006	2008	2018
	Netscape	Netscape	IETF RFC 2246	IETF RFC 4346	IETF RFC 5246	IETF RFC 8446

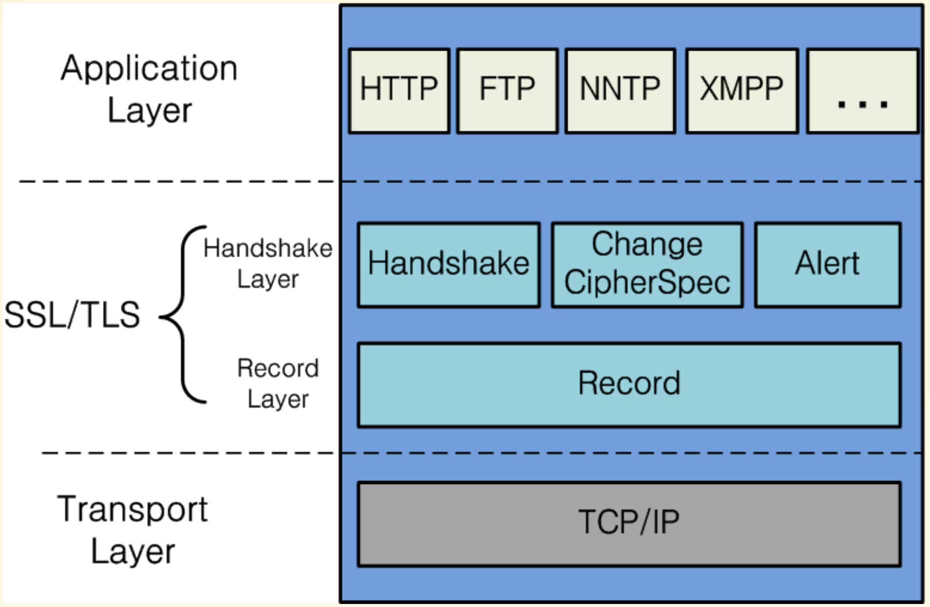
IETF: Internet Engineering Task Force

Some Interesting Measurements of SSL/TLS

Visit here for fresh data.

SSL/TLS

- It is commonly used as HTTPS, where data is transfered via the HTTP protected by SSL.
- It is also used in many other protcocols.



Key Players in SSL/TLS - Same as PKI

- Client
  - A process that initiates the TLS Handshake (e.g., a browser, your car, and etc.).
- Server
  - A process that receives a TLS Handshake request (e.g., a web server).
- Certificate Authority
  - A service that issues Certificates.

Trust Relationship

- A client trusts the CA
- A client trusts the server(s) CA trusts

The CA Market

As of July 2022, analysis based on Alexa top 10 million:

CA	Market Share
IdenTrust	48.9%
DigiCert	18.7%
Sectigo	15.5%
Let's Encrypt	8.2%
GoDaddy	6.1%
GlobalSign	2.7%

Steps - A Bigger Picture

- Server acquires Certificate
- Client verifies
- Establish session keys

Server Acquires Certificate

- CA has a public key and a private key.
- CA generates a self-signed certificate.
- Server acquires a certificate
  - Server generates a public and private key pair.
  - Server generates a Certificate Signing Request (CSR), which contains server's public key and is signed using Server's private key.
  - Server sends this signed CSR to CA

Server Acquires Certificate

- CA inspects and validates the CSR.
- CA generates a cerficiate using information from CSR.
- CA signs this certificate uisng CA's private key.
- CA sends the signed certificate to the server.

Client Verifies Server

- Client wants to connect to the Server securely
  - Client is pre-installed with CA's self-signed certificate, thereby having its public key.
- Client requests and receives Server's certificate.

Client Verifies Server

- Client validates Certificate is legitimate (i.e., it is actually generated by the CA) using CA's public key, which is included in CA's self-signed certificate.
- Client validates that Server is tha actual owner of Certificate.
    - Client does this using SSL/TLS handshake.
    - It verfies that Server has the private key corresponding to the public key, which is included in the received CA.

Establish session keys

- SSL/TLS handshake also produces session keys used for
    - Symmetric Encryption
    - Message Authentication Code (MAC)

TLS - Zoom-In: Protocols

TLS Record Protocol

The Record Protocol defines the format of the records used by TLS.

- When a host sends out a TLS message, TLS puts this message in records regardless this message is for control or data.
- [Header|Payload|MAC|Padding]

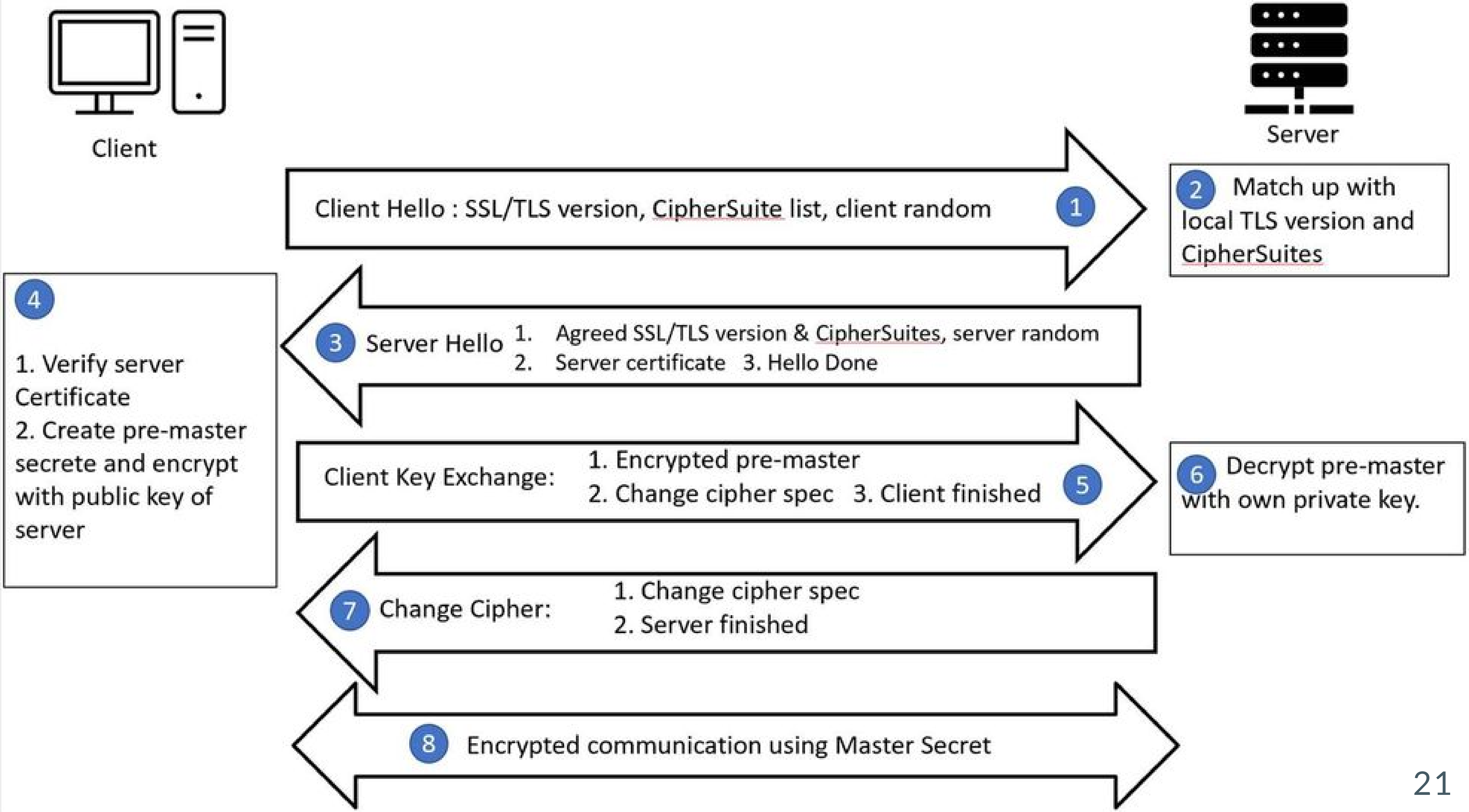
TLS Record Protocol - Format

TLS Alert & Change Cipher Spec Protocol

- The Alert Protocol is used for peers to exchange signal messages, mainly for reporting the cause of failure.
- The Change Cipher Spec Protocol is to change the encryption method used by the client and server. It is used as part of handshake process to switch to symmetric key encryption.

TLS Handshake & Application Protocol

- The Handshake protocol is used to establish the secure channel.
- The Application Protocol is used for data transmission using the established channel.



Client

Server

1 Client Hello : SSL/TLS version, CipherSuite list, client random

2 Match up with local TLS version and CipherSuites

3 Server Hello
1. Agreed SSL/TLS version & CipherSuites, server random
2. Server certificate 3. Hello Done

4
1. Verify server Certificate
2. Create pre-master secrete and encrypt with public key of server

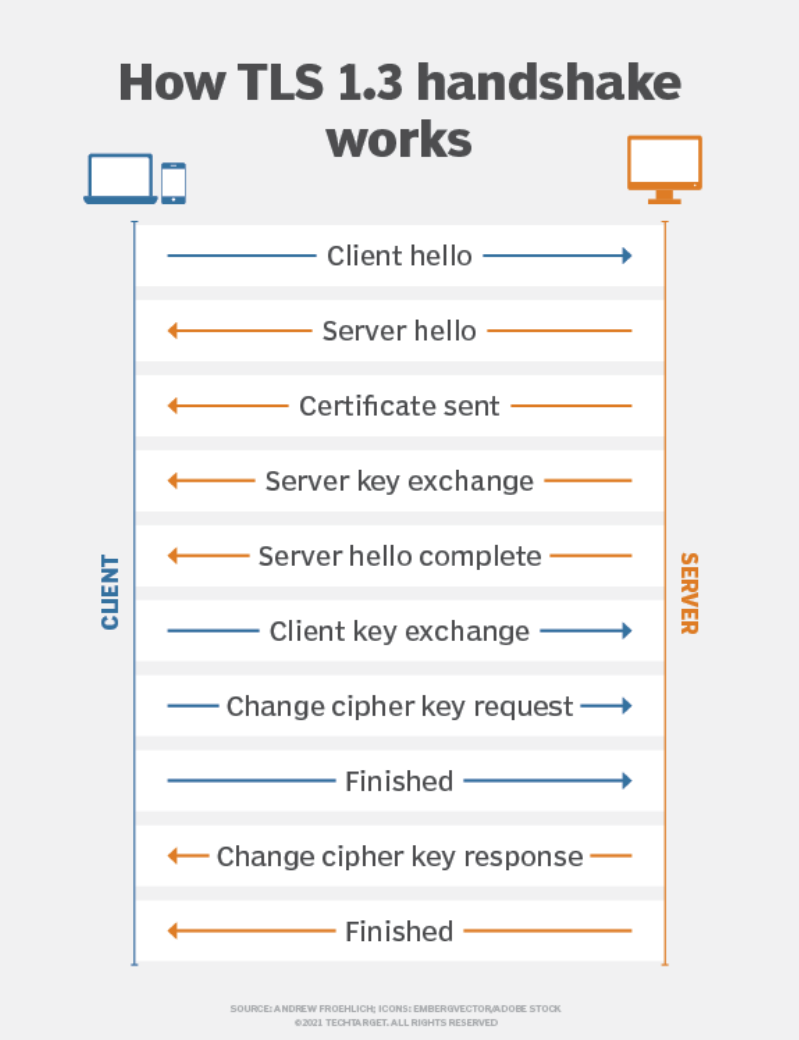
5 Client Key Exchange:
1. Encrypted pre-master
2. Change cipher spec 3. Client finished

6 Decrypt pre-master with own private key.

7 Change Cipher:
1. Change cipher spec
2. Server finished

8 Encrypted communication using Master Secret

Server key exchange is used for certain key exchange algorithms. When it happens, it is sent before the client key exchange.



How TLS 1.3 handshake works

CLIENT

SERVER

Client hello →

← Server hello

← Certificate sent

← Server key exchange

← Server hello complete

Client key exchange →

Change cipher key request →

Finished →

← Change cipher key response

← Finished

SOURCE: ANDREW FROEHLICH; ICONS: EMBERGVECTOR/ADOBE STOCK
©2021 TECHTARGET. ALL RIGHTS RESERVED

Handshake

- Client Hello
  - TLS version
  - Cipher suites supported by the client
  - A random string called "client random"
- Server Hello
  - Cipher selected by the server
  - A random string called "server random"

Handshake

- Server Certificate
  - The server's X.509 certificate chain
    - main certificate
    - intermediary certificates with the correct order
    - the root certificate is omitted
- Server Hello Done
  - Notification to the client that the server's handshake negotiation is done

Handshake

- Client Key Exchange
    - The client and the server agrees with a **pre_master_secret**
        - One possibility using RSA:
            - The client generates a random **pre_master_secret**
            - The client encrypts it using the server's public key extracted from verified public-key certificate
            - The client sends it to the server
    - The client and the server will the generate master secret locally.

Handshake

- Client Change Cipher Spec
  - Notify the server that further communication from client to server will be authenticated and encrypted.
- Server Change Cipher Spec
  - Notify the client that further communication from server to client will be authenticated and encrypted.

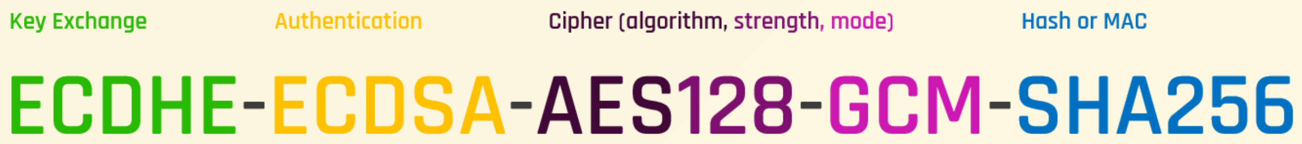
Handshake

- Client Finished
  - Encrypted Verification - a hash and MAC of previous handshake messages.
  - Proves to the server that the client has correct session keys.
- Server Finished
  - Encrypted Verification - a hash and MAC of previous handshake messages.
  - Proves to the client that the server has correct session keys.

Handshake Details - Cipher Suites

Each cipher suite includes

- Key Exchange Protocol - to generate necessary keys
- Authentication - to authenticate parameters for pre_master_secret
- Symmetric Encryption - to encrypt bulk data
- Hashing Algorithm - to be used by MAC for data integrity



ECDHE-ECDSA-AES128-GCM-SHA256

Handshake Details - Cipher Suites

inna.org Cipher Suites

Handshake Details - TLS Key Generation

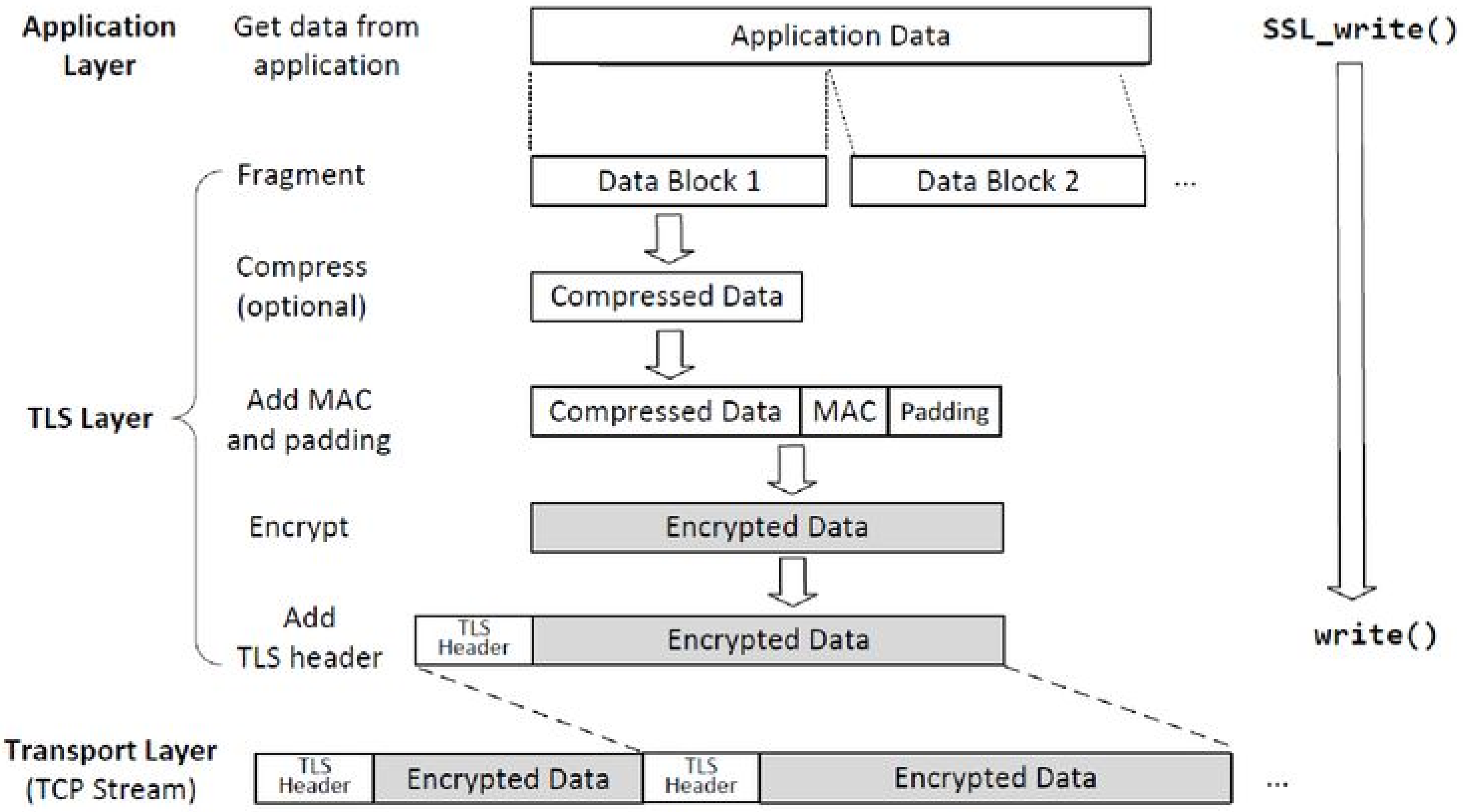
- The client and the server will locally calculate both master_secret and key_block:
    - master_secret = PRF(pre_master_secret, "master secret", client_random + server_random)
    - key_block = PRF(master_secret, "key expansion", server_random + client_random)
        - two MAC keys
        - two encryption keys
        - two initialization vectors

Certificate Verfication

- The client does verify
  - expiration date
  - signature using trusted CAs' self-signed certificates
- The client does not verify
  - the identity information contained in the certificate with the identity of the intended server. (**This is essential for security, but it is the responsibility of applications**.)
- Not checking server's hostname is a common security flaw in programs based on TLS. (Georgiev, 2012)

Application Protocol - Sending Data Over TLS

Sending Data with the TLS Record Protocol



Demo - Sniffing and Parsing TLS Packets Using Wireshark

Reference

A good tutorial on TLS handshake

A good tutorial on cipher suites