

1

CRYPTOGRAPHY (CTG)

Diploma in Cybersecurity and Digital Forensics (Dip in CSF)

Academic Year (AY) '21/'22 – Semester 2

WEEK 8.1

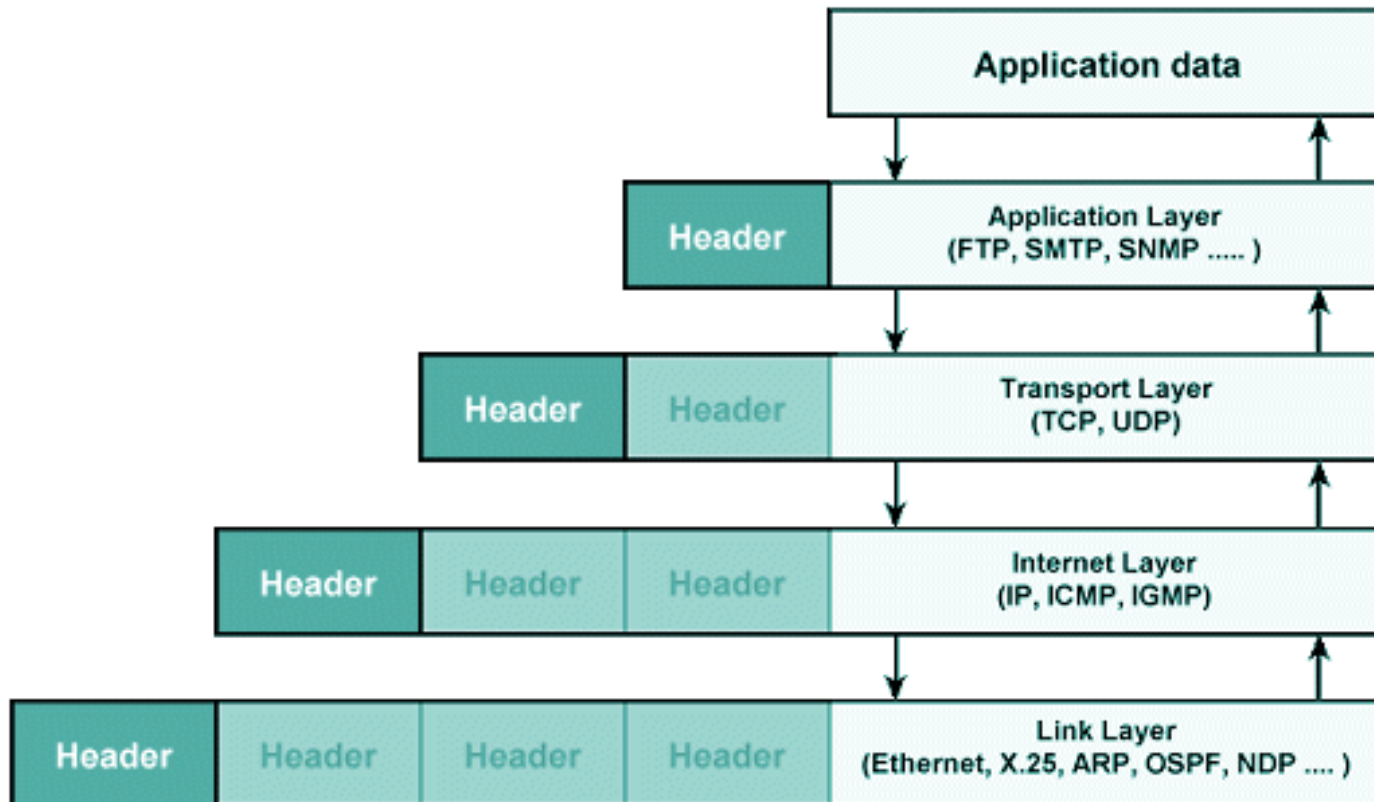
SECURE SOCKET LAYER (SSL) / TRANSPORT LAYER SECURITY (TLS)

2

What was the need for SSL/TLS?

No Data Encryption Mechanism in TCP/IP Layers!

3



Weakness in TCP/IP Protocol

4

- ❑ None of the layers provide encryption to protect data transmitted from one end to another end.
- ❑ When the World-Wide-Web(WWW) was first introduced in the late 80s, the web browser/client and the web server communicate with one another in plaintext using the Hypertext Transfer Protocol (HTTP). These raw and unencrypted data traveling across the Internet can be easily sniffed and read by anyone.
- ❑ This problem together with the need to making sure that the web client is communicating with the right web server and vice versa, making the world-wide-web not an ideal platform for eCommerce.

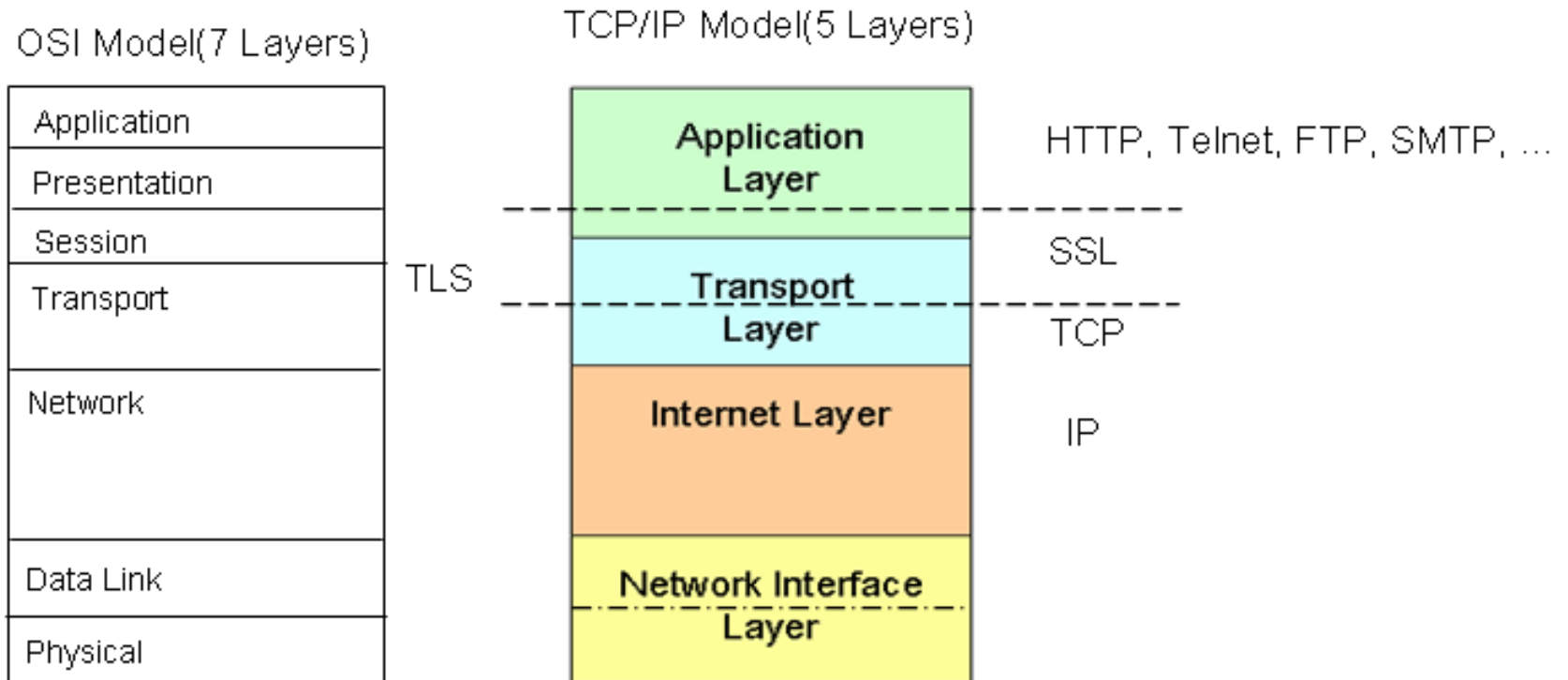
The Solution

5

- ❑ SSL (Secure Socket Layer) version 2 was first introduced in 1994 by Netscape to solve the mentioned problems.
- ❑ In the subsequent year, Netscape introduced SSLv3 and its success led to IETF(Internet Engineering Task Force) to adopt it and to enhance it to become a standard.
- ❑ Subsequently, IETF released the standards: TLS v1.0 and TLS v1.2 and TLS v1.3 in 1999, 2006, and 2008 respectively.
 - ❑ TLS stands for Transport Layer Security
 - ❑ HTTPS = HTTP over SSL/TLS used in all the browsers.
- ❑ SSL v3.0 must not be used due to many recent attacks such as POODLE attack
- ❑ TLS v1.2 and TLS v1.3 are recommended for security

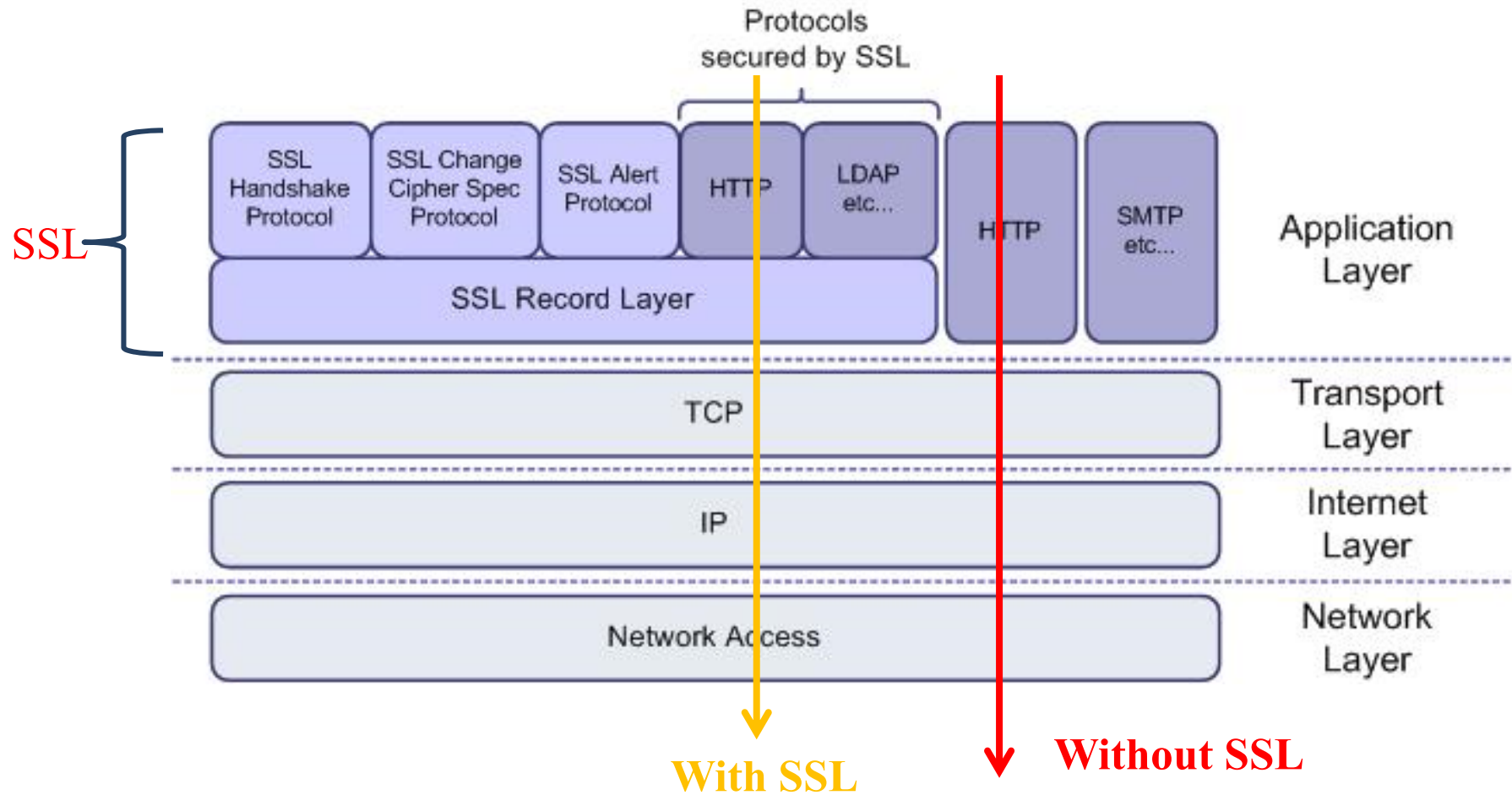
SSL/TLS in the TCP/IP Protocol Suite

6



SSL/TLS in the TCP/IP Protocol Suite

7



8

Transport Layer Security (TLS)

Purpose of TLS

9

- TLS – Transport Layer Security
- It uses
 - ▣ Encryption
 - ▣ Cryptographic hash functions or message digests
 - ▣ Digital signatures
- To provide a secure transport connection between applications (e.g., a web server and a browser)
 - ▣ Client- server authentication
 - ▣ Data confidentiality
 - ▣ Data origin authentication
 - ▣ Data integrity

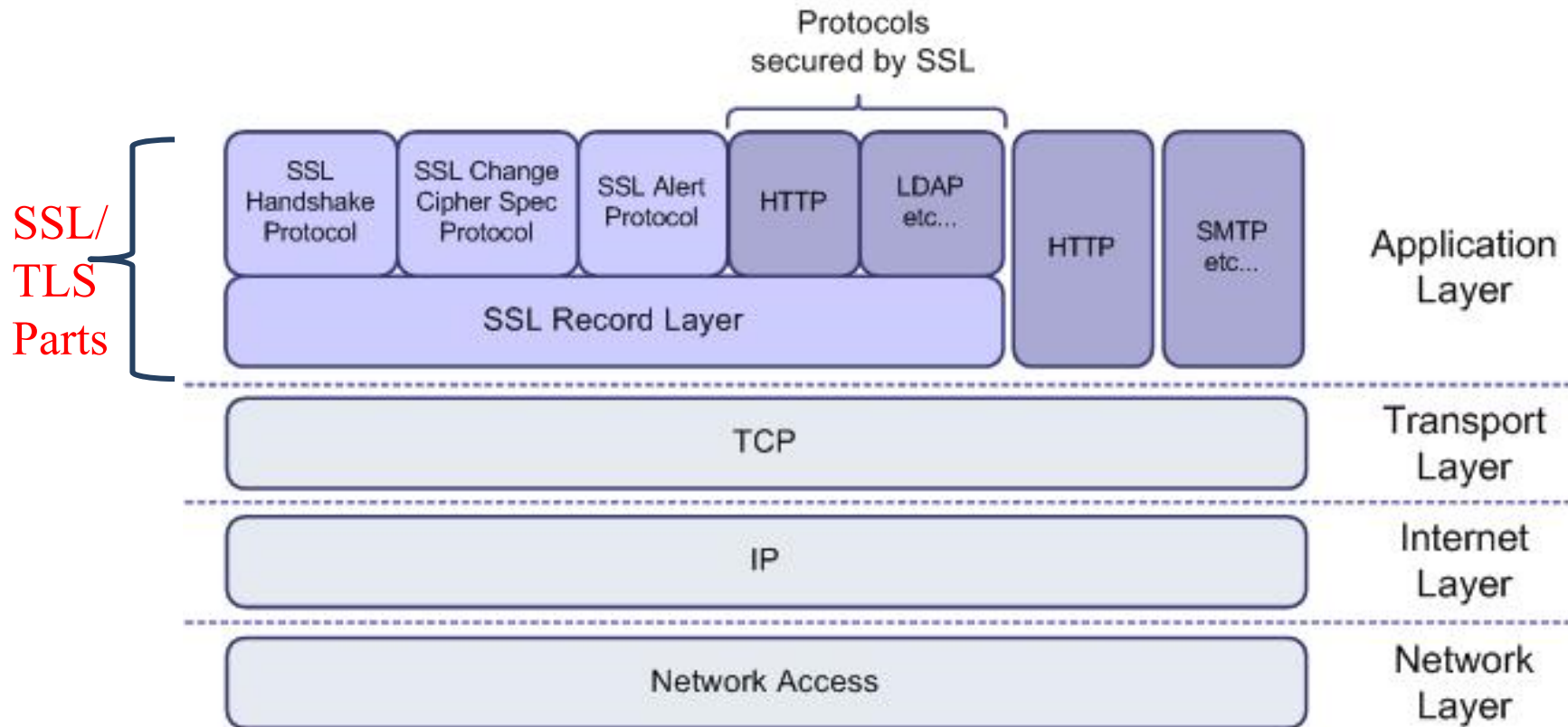
Two main parts of TLS

10

- ❑ Handshake Protocol
 - ▣ Establish shared secret key using public-key cryptography
 - ▣ Signed certificates for authentication
- ❑ Record Layer
 - ▣ Transmit data using negotiated key, encryption function
- ❑ We focus only on “Handshake Protocol”

Main Parts of SSL/TLS

11



12

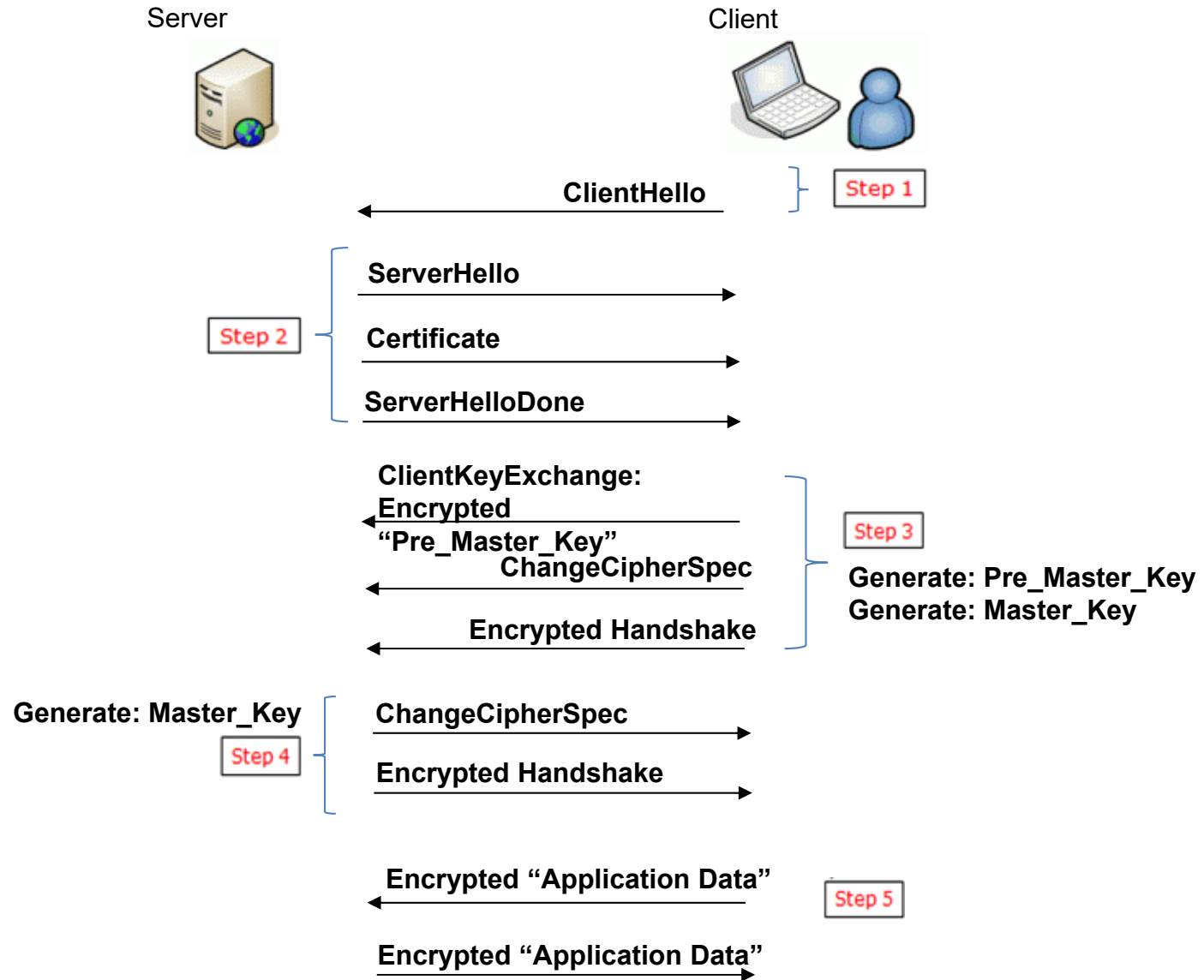
TLS - Handshake Protocol

Handshake Protocol

13

- ❑ To understand how the protocol works, you need to understand how the TLS secure channel is setup before encrypted data flow between two communicating devices.
- ❑ The channel is setup by the TLS Handshake.
- ❑ The handshake protocol describes the rules used to establish common cryptographic parameters as well as authenticating the server and optionally, the client.

Handshake Protocol



Step 1: Client Hello

15

- Client makes a connection request to the server by sending the client random number and a set of ciphers suites that it supports.

Filter: `tcp.stream eq 247 && ssl` Expression... Clear Apply Save

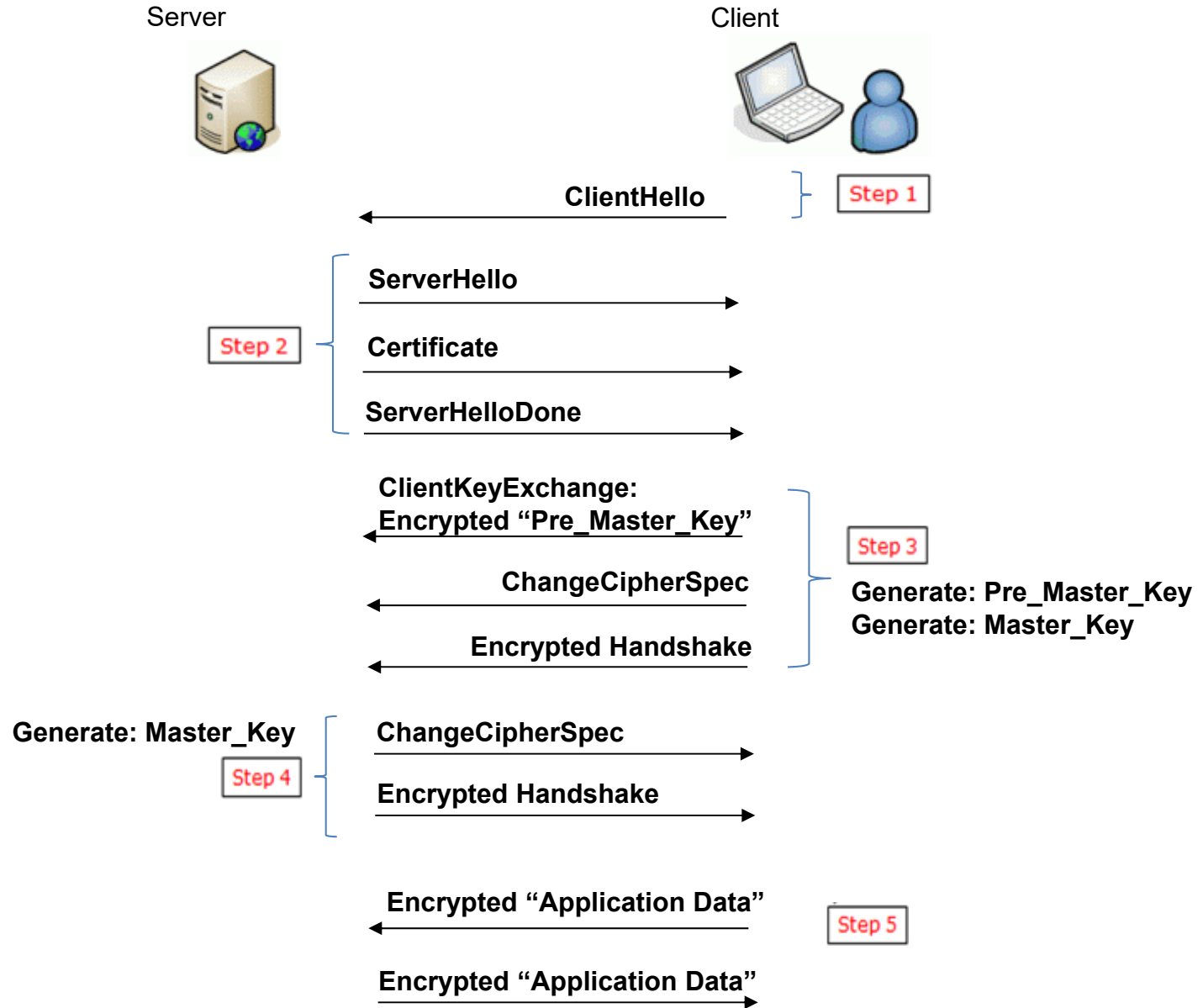
No.	Time	Source	Destination	Protocol	Length	Info
14928	304.581701000	192.168.1.3	23.74.219.219	TLSv1.2	274	Client Hello

Handshake Protocol: Client Hello

- Handshake Type: Client Hello (1)
- Length: 199
- Version: TLS 1.2 (0x0303)
- Random
 - GMT Unix Time: May 30, 2079 23:28:46.000000000 SGT
 - Random Bytes: ac6b76f8278f8abc7d02d9a8da3d86062fa58c96ad149f69...
 - Session ID Length: 0
 - Cipher Suites Length: 34
- Cipher Suites (17 suites)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
 - Cipher Suite: TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x009e)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc14)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc13)
 - Cipher Suite: TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc15)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
 - Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
 - Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
 - Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)

.....

Handshake Protocol – Step 2



Step 2: Server Hello

17

- The server responds with a server random number, the chosen cipher (from the client cipher suites it received), and server certificate.

Filter: tcp.stream eq 247 && ssl						
No.	Time	Source	Destination	Protocol	Length	Info
14928	304.581701000	192.168.1.3	23.74.219.219	TLSv1.2	274	Client Hello
14932	304.588029000	23.74.219.219	192.168.1.3	TLSv1.2	1514	Server Hello

Secure Sockets Layer
<ul style="list-style-type: none"> <ul style="list-style-type: none"> TLSv1.2 Record Layer: Handshake Protocol: Server Hello <ul style="list-style-type: none"> Content Type: Handshake (22) Version: TLS 1.2 (0x0303) Length: 72 <ul style="list-style-type: none"> Handshake Protocol: Server Hello <ul style="list-style-type: none"> Handshake Type: Server Hello (2) Length: 68 Version: TLS 1.2 (0x0303) <ul style="list-style-type: none"> Random <ul style="list-style-type: none"> GMT Unix Time: Jul 5, 2015 17:56:38.000000000 SGT Random Bytes: 7003cc602c6cbbc89dbb9f40eac97de9576cb1ddf68f8f53... Session ID Length: 0 Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035) Compression Method: null (0) Extensions Length: 28 <ul style="list-style-type: none"> Extension: server_name <ul style="list-style-type: none"> Type: server_name (0x0000) Length: 0 <ul style="list-style-type: none"> Extension: renegotiation_info <ul style="list-style-type: none"> Type: renegotiation_info (0xff01) Length: 1

Step 2: Server Certificate

SSL-Handshake-1.pcapng [Wireshark 1.12.1 (v1.12.1-0-g01b65bf from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: `tcp.stream eq 247 && ssl` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
14928	304.581701000	192.168.1.3	23.74.219.219	TLSv1.2	274	Client Hello
14932	304.588029000	23.74.219.219	192.168.1.3	TLSv1.2	1514	Server Hello
14933	304.588148000	23.74.219.219	192.168.1.3	TLSv1.2	1107	Certificate

Frame 14933: 1107 bytes on wire (8856 bits), 1107 bytes captured (8856 bits) on interface 0

- Ethernet II, Src: AztechEL_b2:fc:a6 (00:26:75:b2:fc:a6), Dst: Apple_1c:f9:a2 (88:1f:a1:1c:f9:a2)
- Internet Protocol Version 4, Src: 23.74.219.219 (23.74.219.219), Dst: 192.168.1.3 (192.168.1.3)
- Transmission Control Protocol, Src Port: 443 (443), Dst Port: 50963 (50963), Seq: 1449, Ack: 209, Len: 1041
- [2 Reassembled TCP Segments (2403 bytes): #14932(1371), #14933(1032)]

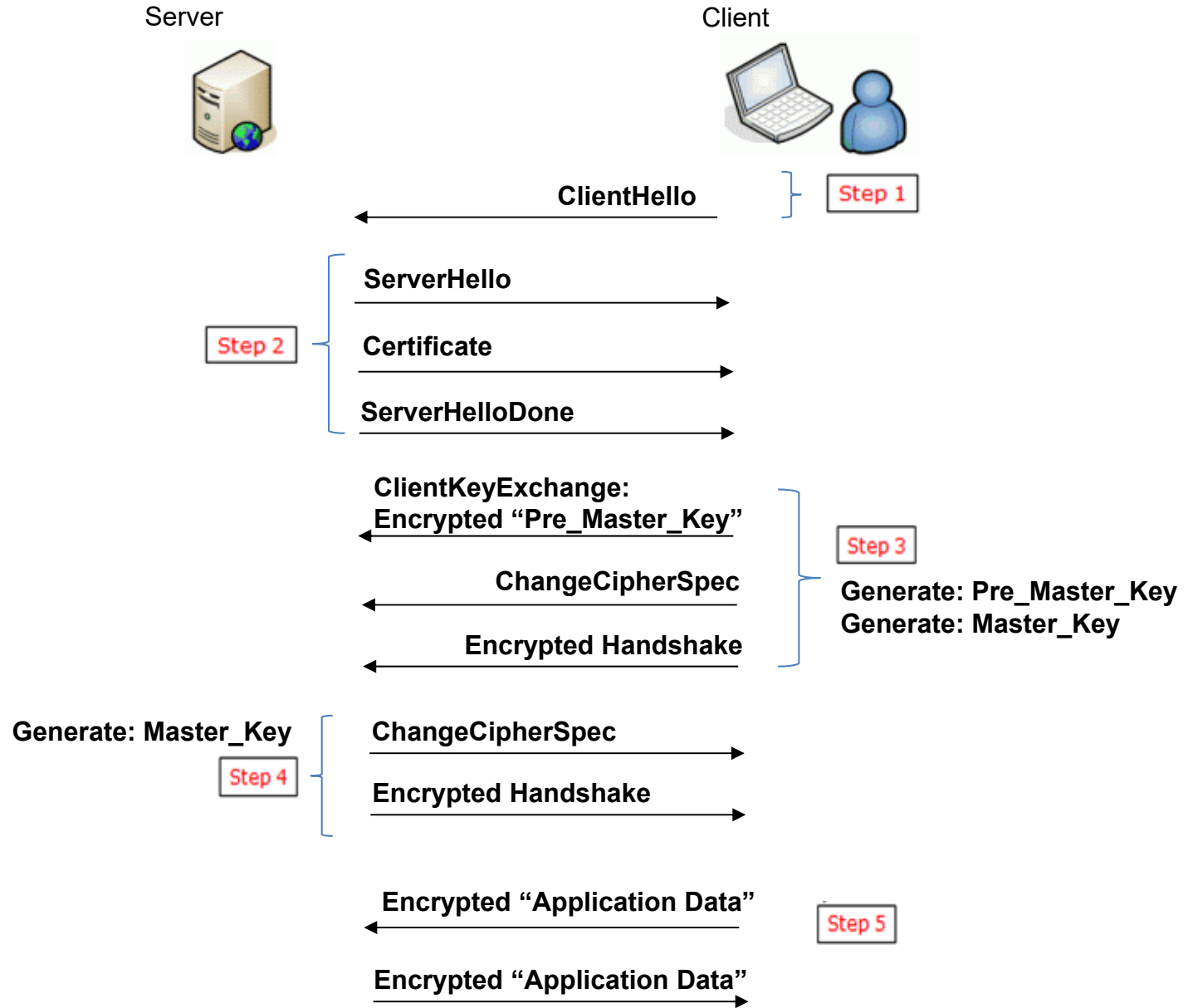
Secure Sockets Layer

- TLSv1.2 Record Layer: Handshake Protocol: Certificate
 - Content Type: Handshake (22)
 - Version: TLS 1.2 (0x0303)
 - Length: 2398
- Handshake Protocol: Certificate
 - Handshake Type: Certificate (11)
 - Length: 2394
 - Certificates Length: 2391
- Certificates (2391 bytes)
 - Certificate Length: 1396
 - Certificate (id-at-commonName=global.ibm.com,id-at-organizationName=IBM,id-at-localityName=Armonk,id-at-st
 - Certificate Length: 989
 - Certificate (id-at-commonName=GeoTrust SSL CA,id-at-organizationName=GeoTrust, Inc.,id-at-countryName=US)

Server Certificate

CA Certificate

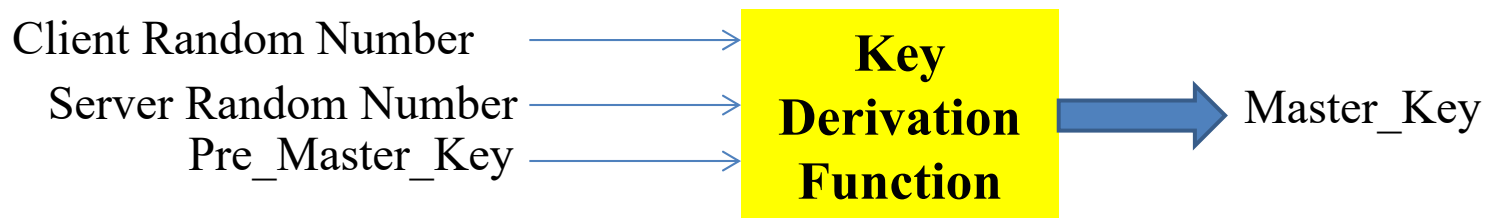
Handshake Protocol – Step 3



Step 3 - Client Key Exchange and Generation of Master_Key

20

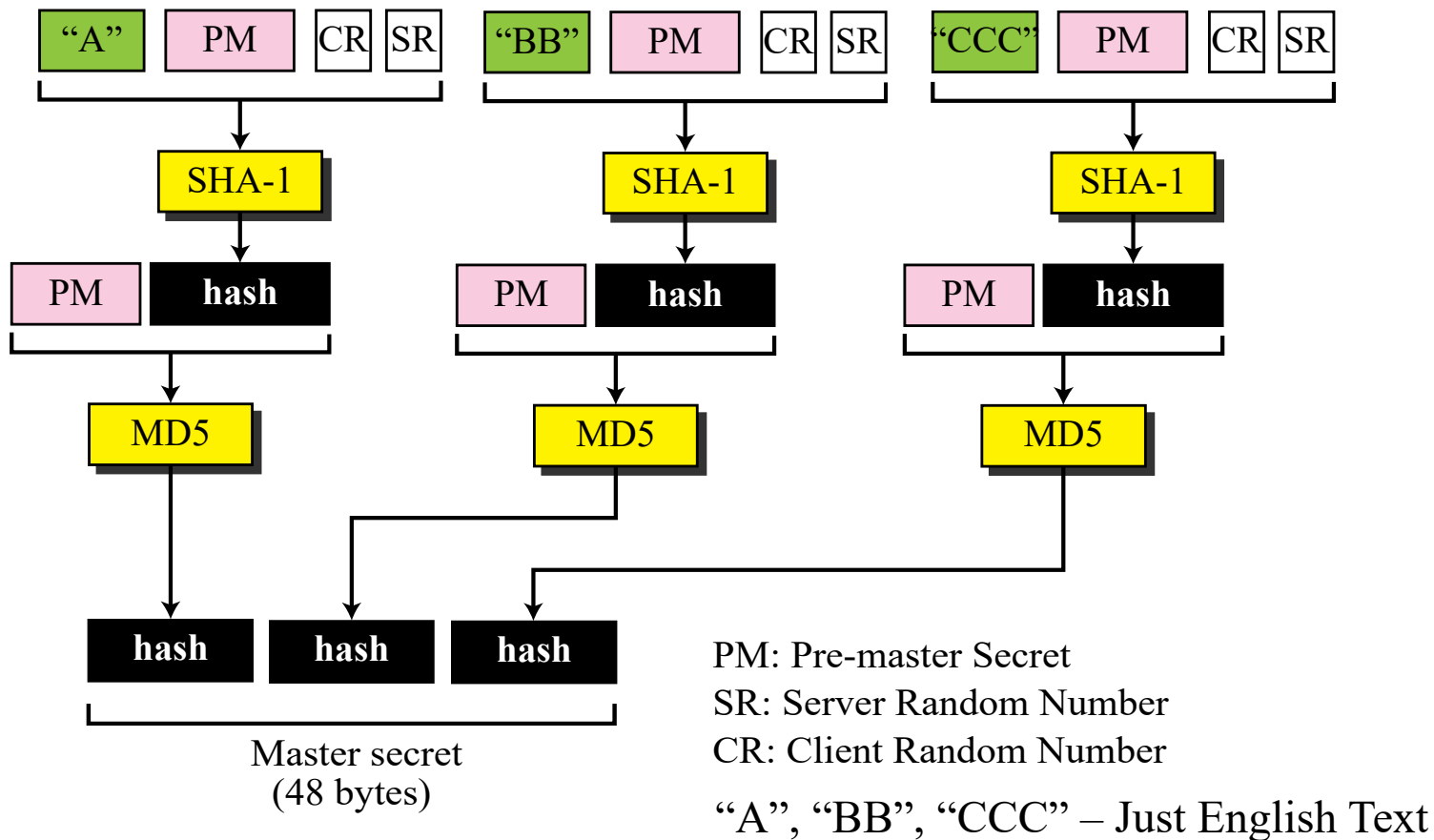
- The client verifies the identity of the Server using the server's certificate and server's CA certificate.
 - ▣ This step is not necessary for repeated session within certain time frame.
- The client then generates another random number called the "Pre_Master_Key".
- The client encrypts Pre_Master_Key with the server's public key. Note that the client received the server's public key via the server certificate in Step 2. The client sends the encrypted Pre_Master_Key as the Client Key Exchange.
- The client goes on to compute the "Master_Key" by putting 3 numbers (as shown below) through a Key Derivation Function.



```
master_secret = PRF(pre_master_secret, "master secret", ClientHello.random + ServerHello.random)
```

Step 3 - Client Key Exchange and Generation of Master_Key

21



Step 3: Client Change Cipher Spec

22

- The client informs the server that all messages sent from now on will be encrypted with the generated Master_Key.

Step 3: Client Encrypted Handshake Message

23

- ❑ The client hashes all the previously exchanged messages and encrypts the message digest (MAC-Message Authentication Code) with the Master Key using the agreed symmetric algorithm.
- ❑ The client then sends the server the encrypted MAC code.

Step 3 - Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message

24

Filter: **tcp.stream eq 247 && ssl** Expression... Clear Apply Save

Io.	Time	Source	Destination	Protocol	Length	Info
14928	304.581701000	192.168.1.3	23.74.219.219	TLSv1.2	274	Client Hello
14932	304.588029000	23.74.219.219	192.168.1.3	TLSv1.2	1514	Server Hello
14933	304.588148000	23.74.219.219	192.168.1.3	TLSv1.2	1107	Certificate
14935	304.589678000	192.168.1.3	23.74.219.219	TLSv1.2	408	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Mes

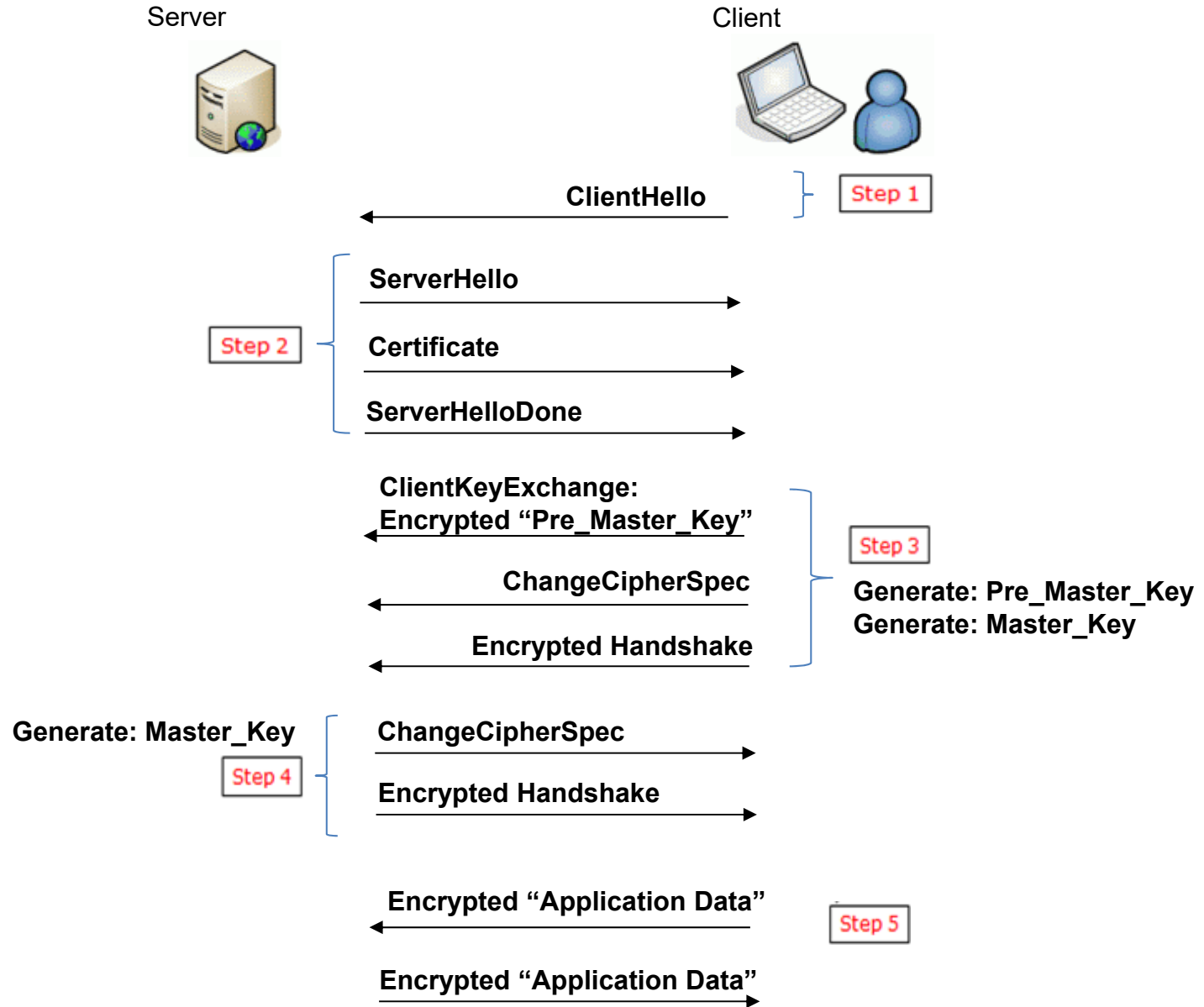
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 262

- Handshake Protocol: Client Key Exchange
 - Handshake Type: Client Key Exchange (16)
Length: 258
 - RSA Encrypted PreMaster Secret
 - Encrypted PreMaster length: 256
 - Encrypted PreMaster: 861377373aa4b37e6cc68b05c52f3a0898f4a31146ae9c5f...

▼ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
Content Type: Change Cipher Spec (20)
Version: TLS 1.2 (0x0303)
Length: 1
Change Cipher Spec Message

▼ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 64
Handshake Protocol: Encrypted Handshake Message

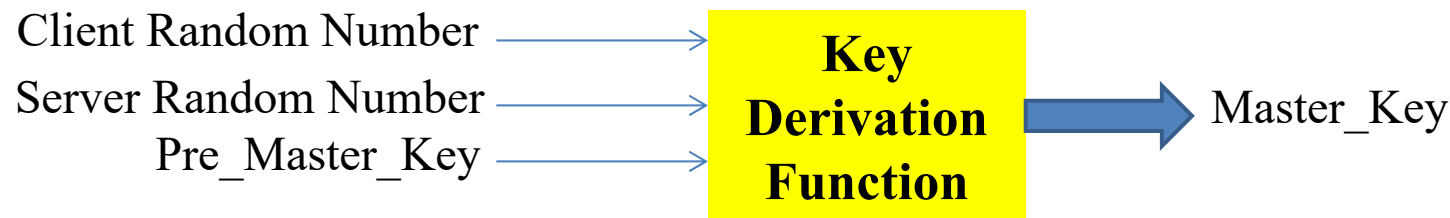
Handshake Protocol – Step 4



Step 4: Server Generates Master_Key

26

- ❑ The server decrypts the encrypted “Pre_Master_key” using its private key.
- ❑ It then goes on to compute the “Master_key” by putting the same three numbers through the same Key Derivation function.



Step 4: Verification of Client's Encrypted Handshake Message

27

- ❑ The server decrypts the client's encrypted handshake (MAC) with the Master_key
- ❑ It hashes all the previously exchanged messages and compares the hash with the decrypted MAC. If they match, it means all the previously exchanged messages were not tempered with.
- ❑ If the hashes do not match, the server terminates the handshake with an alert protocol.

Step 4: Change Cipher Spec

28

- The server informs the client that all messages sent from now on will be encrypted with the generated Master_Key.

Step 4: Encrypted Handshake Message

29

- ❑ The server hashes all the previously exchanged messages and encrypts the message digest (MAC-Message Authentication Code) with the Master_Key using the agreed symmetric algorithm.
- ❑ The server then sends the client the encrypted handshake (MAC) code.

Step 4: Server Change Cipher Spec, Encrypted Handshake Message

30

Wireshark 1.12.1 (v1.12.1-0-g01b65bf from master-1.12)

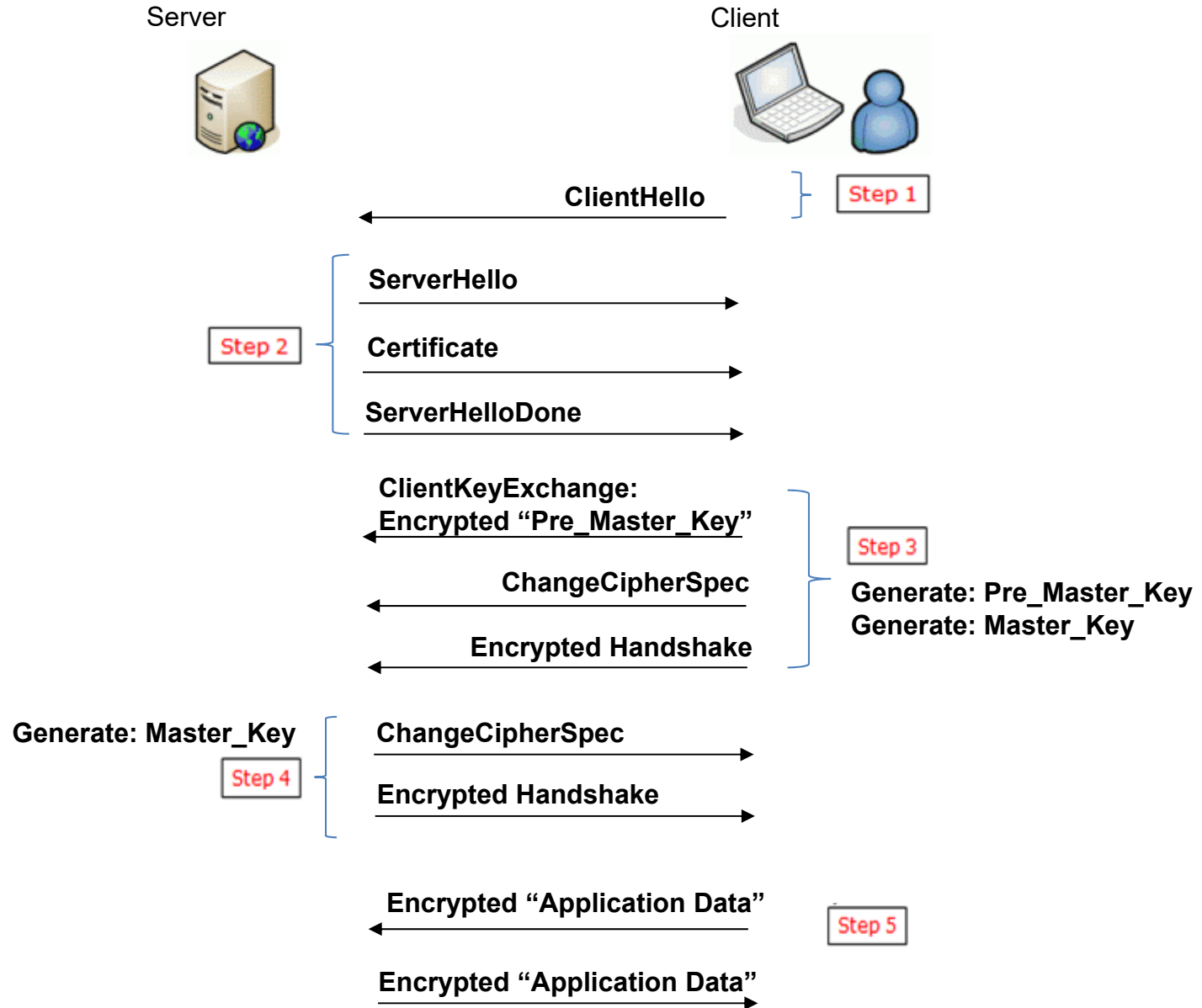
Filter: tcp.stream eq 247 && ssl

Time	Source	Destination	Protocol	Length	Info
14933	304.588148000	23.74.219.219	TLSv1.2	1107	Certificate
14935	304.589678000	192.168.1.3	TLSv1.2	408	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
14936	304.596860000	23.74.219.219	TLSv1.2	332	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message

Frame 14936: 332 bytes on wire (2656 bits), 332 bytes captured (2656 bits) on interface 0

- Ethernet II, Src: AztechEL_b2:fc:a6 (00:26:75:b2:fc:a6), Dst: Apple_1c:f9:a2 (88:1f:a1:1c:f9:a2)
- Internet Protocol Version 4, Src: 23.74.219.219 (23.74.219.219), Dst: 192.168.1.3 (192.168.1.3)
- Transmission Control Protocol, Src Port: 443 (443), Dst Port: 50963 (50963), Seq: 2490, Ack: 551, Len: 266
- Secure Sockets Layer
 - TLSv1.2 Record Layer: Handshake Protocol: New Session Ticket
 - TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
 - Content Type: Change Cipher Spec (20)
 - Version: TLS 1.2 (0x0303)
 - Length: 1
 - Change Cipher Spec Message
 - TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
 - Content Type: Handshake (22)
 - Version: TLS 1.2 (0x0303)
 - Length: 64
 - Handshake Protocol: Encrypted Handshake Message

Handshake Protocol – Step 5



Step 5: Verification of Server's Encrypted Handshake Message

32

- ❑ The client decrypts the server's encrypted handshake (MAC) with the Master_key
- ❑ It hashes all the previously exchanged messages and compares the hash with the decrypted MAC. If they match, it means all the previously exchanged messages were not tempered with.
- ❑ If the hashes do not match, the client terminates the handshake with an alert protocol.

Step 5: Encrypted Application Data

33

- ❑ If the messages were not tampered with, the client will start encrypting the application data using the Master_Key and send to the server. Upon receiving, the data will be decrypted using the same Master_Key.
- ❑ Likewise, the server will also encrypt the data using the Master_Key and send to the client. The client will decrypt the encrypted data using the Master_Key.
- ❑ The encrypted communication goes on until the client request termination.

Step 5: Encrypted Application Data

No.	Time	Source	Destination	Protocol	Length	Info
6	0.085490000	192.168.1.3	23.74.219.219	TCP	1514	[TCP segment of a
7	0.085490000	192.168.1.3	23.74.219.219	TLSv1.2	319	Application Data
8	1.596049000	23.74.219.219	192.168.1.3	TLSv1.2	855	Application Data

▶ Frame 7: 319 bytes on wire (2552 bits), 319 bytes captured (2552 bits) on interface 0
 ▶ Ethernet II, Src: Apple_1c:f9:a2 (88:1f:a1:1c:f9:a2), Dst: AztechEl_b2:fc:a6 (00:26:75:b2:fc:a6)
 ▶ Internet Protocol Version 4, Src: 192.168.1.3 (192.168.1.3), Dst: 23.74.219.219 (23.74.219.219)
 ▶ Transmission Control Protocol, Src Port: 50963 (50963), Dst Port: 443 (443), Seq: 1999, Ack: 2756, Len: 253
 ▶ [2 Reassembled TCP Segments (1701 bytes): #6(1448), #7(253)]

▼ Secure Sockets Layer
 ▼ TLSv1.2 Record Layer: Application Data Protocol: spdy
 Content Type: Application Data (23)
 Version: TLS 1.2 (0x0303)
 Length: 1696
 Encrypted Application Data: f6ee24a0c856bfcce04f5d1e54f8be02e4f5006f16840d44...

0000	17 03 03 06 a0 f6 ee 24 a0 c8 56 bf cc e0 4f 5d\$..V...0]
0010	1e 54 f8 be 02 e4 f5 00 6f 16 84 0d 44 d2 82 f8	.T.... o...D...
0020	c8 b1 14 b2 c5 a9 5a e5 eb 64 7c cf ef b4 d2 c2	...Z. .d
0030	52 41 5f ae 6a b3 c6 2f be 94 0e 08 36 ba d8 54	A_.j../6..T
0040	1c 4a b2 94 fe f6 5f 9a f6 f8 88 8f 46 f5 ed 91	.J....._F...
0050	35 a0 f8 5e f6 4e b1 e5 8e 45 bd 9f eb 15 d0 d3	5..^..N.. .E.....
0060	13 9b c5 8f cd 06 35 78 1e 6f 28 fb ad 26 84 c05x .o(..&..
0070	c1 d6 41 0a f7 fa d6 59 43 5a fa 19 9b 1e af 48	..A....Y CZ.....H
0080	a6 b1 91 42 5c 3a ac 5d 75 ca 7f a9 01 3f 54 42	...B\:.] u....?TB
0090	e0 da 24 fd 2b 9c 7d ad 9a 79 44 ce 85 f9 98 9f	..\$.+..}..yD.....
00a0	fa a5 bd 73 44 eb 37 80 a1 0f 22 f8 f6 f6 77 08	...sD.7. ..."...w.
00b0	0f 01 db 08 ee 31 dd 18 51 6e ae d5 92 af 8b 901.. Qn.....
00c0	3e 76 e3 c2 3e 81 a2 f6 fe dc fb 54 f9 4d a4 df	>v..>... ..T.M..
00d0	45 01 de c4 be 1e 59 d1 bd 27 4e 19 db 8b ab 41	E.....Y. .'N....A

35

Exercises

Wireshark Captures

36

- ❑ Download and install Wireshark (if you have not already done so)
 - ▣ <https://www.wireshark.org/download.html>
- ❑ Choose the appropriate installer (64-bit or 32-bit)
- ❑ Follow the instructions provided.
- ❑ Please note that WinPcap needs to be installed as well (it should come with it when you are installing Wireshark)

Wireshark Captures

37

- Analyze the packets provided in the PCAP folder
- List the following
 - ▣ 8 bytes of Client Random
 - ▣ First 4 Client Cipher Suites
 - ▣ Server Name
 - ▣ 8 bytes of Server Random
 - ▣ Server Chosen Cipher Suite
 - ▣ Server certificate
 - Issued by
 - Validity period
 - 8 bytes of public key
 - ▣ 8 bytes of Encrypted Pre Master Key
 - ▣ 8 bytes of Client Encrypted Handshake Message
 - ▣ 8 bytes of Server Encrypted Handshake Message

38

Summary

Summary

39

ClientHello

- Client sends server the version of TLS that it would like to use, a list of supported ciphers and a random string that the server will need later.

ServerHello

- The server sends the TLS version, the cipher it has chosen and a random string that the client will need later.

Certificate

- The server sends its certificate as proof of its identity.

ServerHelloDone

- The server tells the client that it is done passing parameters.

ClientKeyExchange

- The client sends pre-master secret which will be used by both sides to generate session keys.

ChangeCipherSpec

- The client informs the server that all messages sent from now on will be encrypted with the generated session key.

Finished

- The client sends a hash of the handshake components, encrypted with the generated session key. The server will use this to verify the integrity of the handshake process.

ChangeCipherSpec

- The server informs the client that all messages sent from now on will be encrypted with the generated session key.

Finished

- The server sends a hash of the handshake components, encrypted with the generated session key. The client will use this to verify the integrity of the handshake process.

The TLS handshake using the key exchange method

Summary

40

- You learnt
 - ▣ Need for SSL/TLS
 - ▣ Purpose of SSL/TLS
 - ▣ Understanding TLS – Handshake Protocol
 - ▣ Analyzing SSL/TLS packets using Wireshark
- <https://tools.ietf.org/html/rfc5280#page-23>