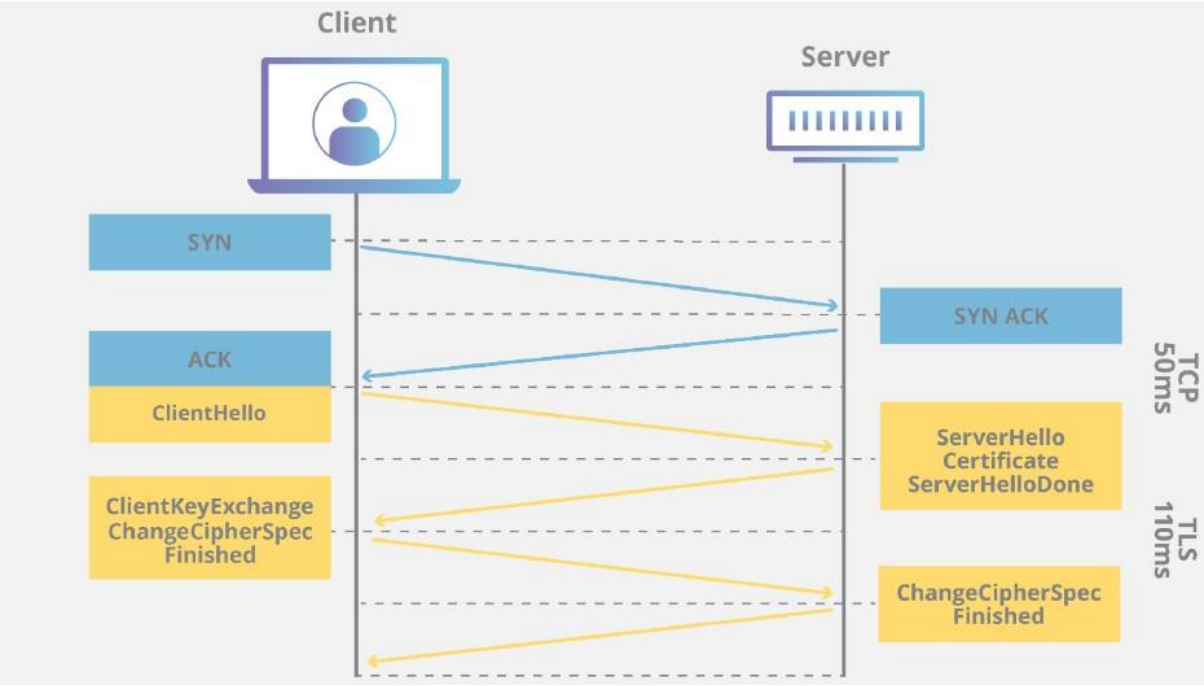
Name: Rithik Sarvesh Bharathiraja

UID: 120395246

Course: ENPM693

Task: Homework 6

Part – 1



This is a packet capture of the conversation between the client and badssl.com

Client hello

Following important parameters are sent to the server in the client_hello message

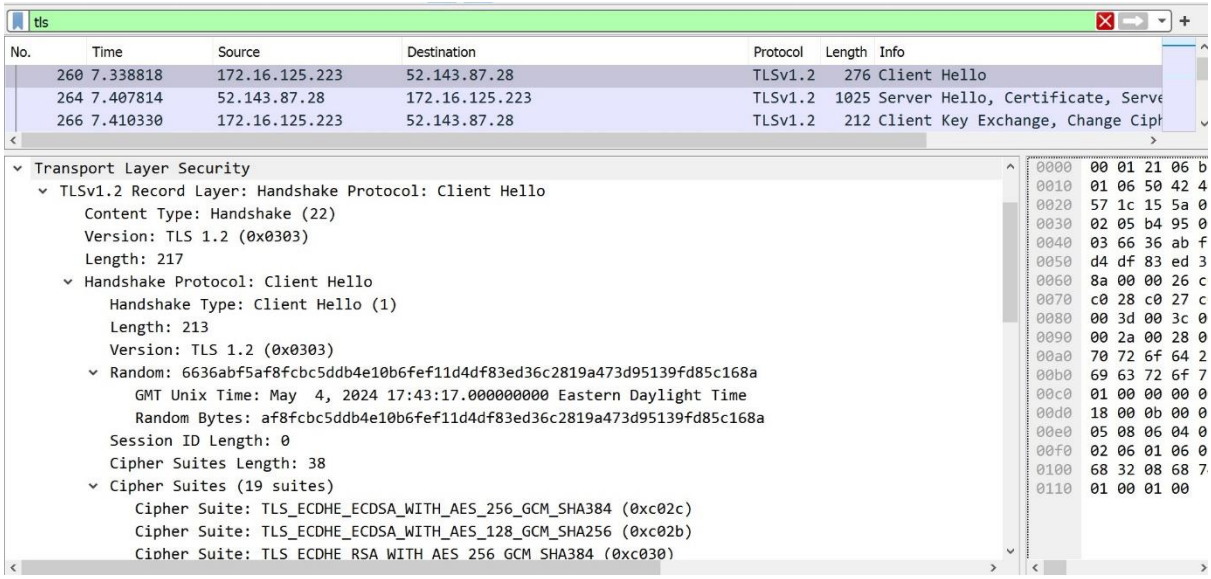
Version – TLS version

Timestamp – to prevent replay attack

random bytes - generated by the client

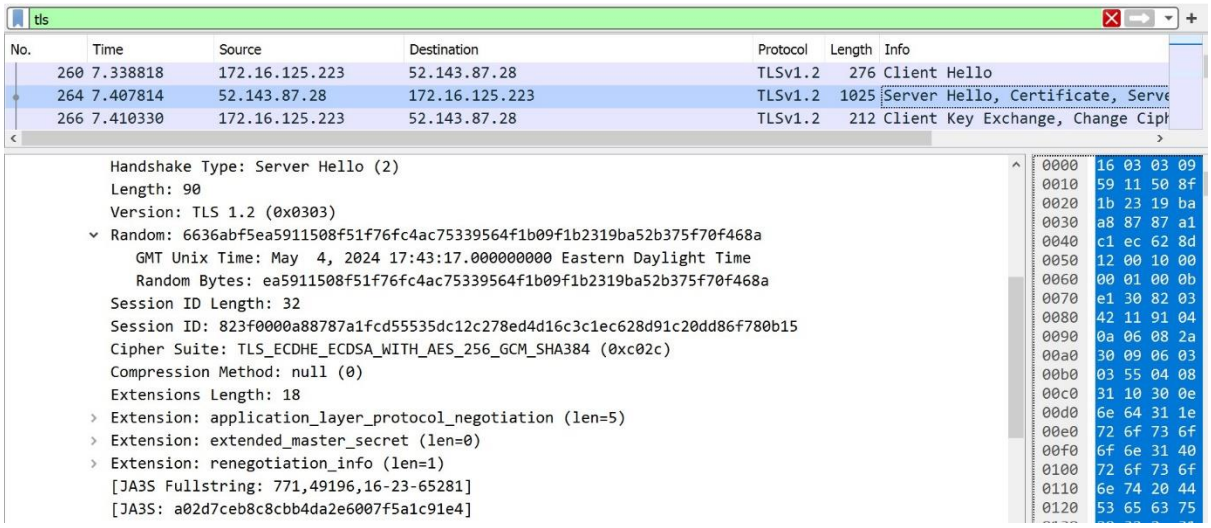
Session ID – To check whether the connection was established already.

Cipher suites – The types of cipher suites supported by the client

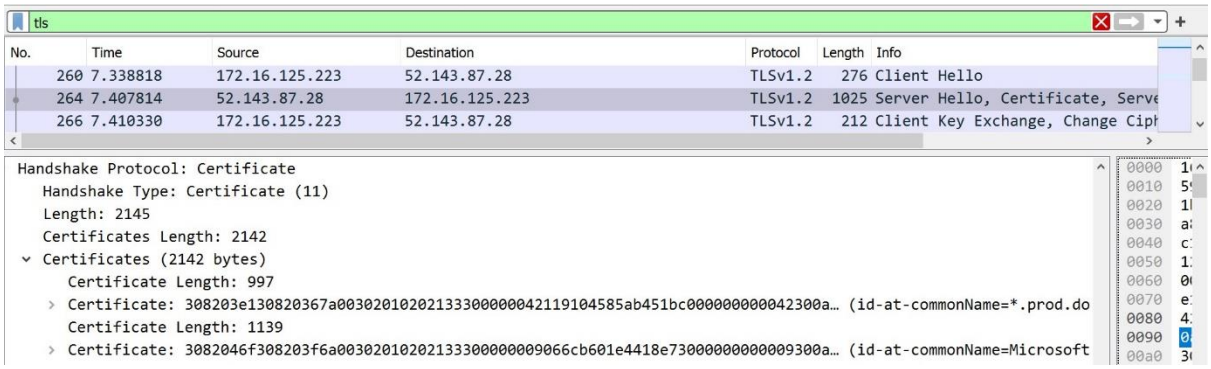


Server hello

In response to client hello, server sends the server_hello message. In this, server chooses the cipher suite and sends that to the client. Along with that, session is established and unique session ID is generated and sent to the client.



After server_hello, the server sends its certificate to the client.



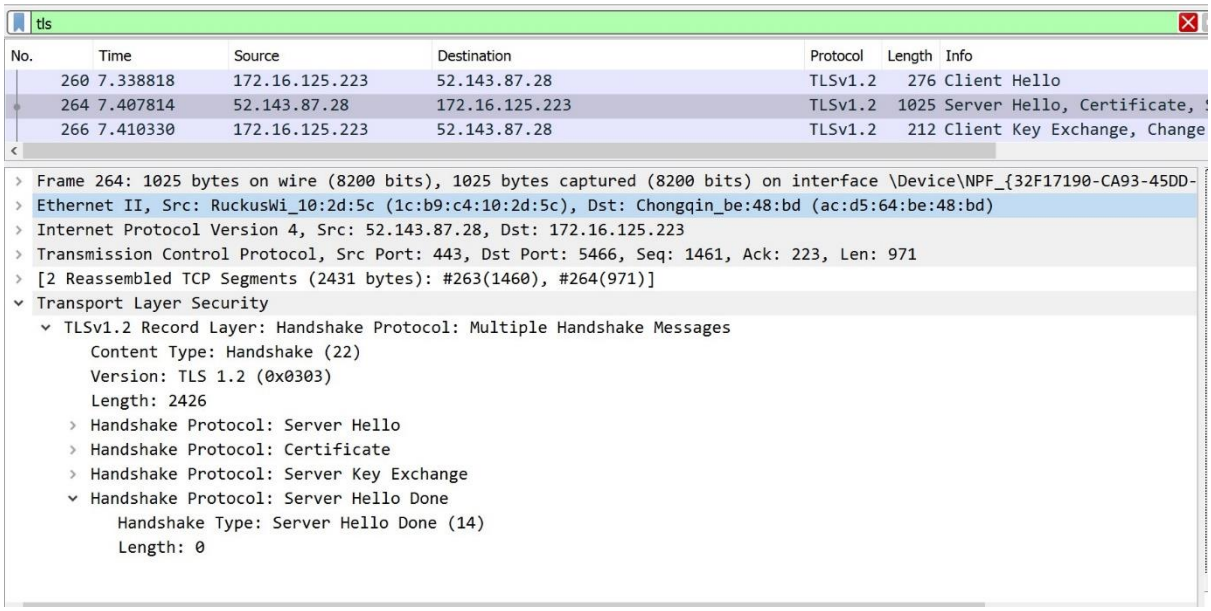
In certificate, it comes up with various important parameters like

- Signature
- Issuer
- Validity
- Subject

To exchange keys, server sends the key exchange algorithm and its parameter.

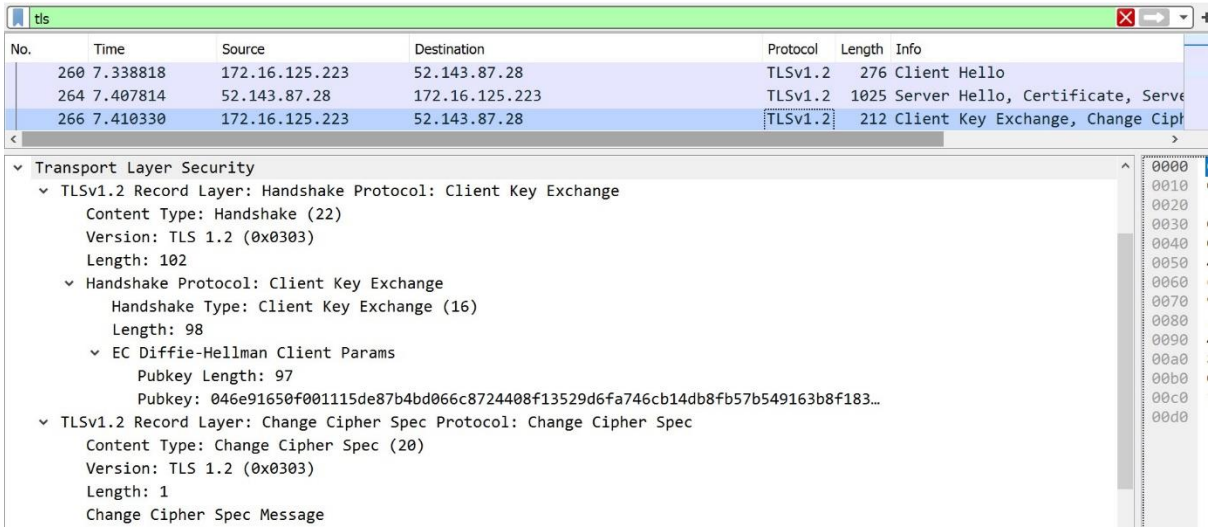
```
▾ Handshake Protocol: Server Key Exchange
    Handshake Type: Server Key Exchange (12)
    Length: 175
  ▾ EC Diffie-Hellman Server Params
      Curve Type: named_curve (0x03)
      Named Curve: secp384r1 (0x0018)
      Pubkey Length: 97
      Pubkey: 040e6f2d9416ff684086e1e3a9f0e011208c3cc8ab20c7affd9608ebb4696c0521f9313e…
    ▾ Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
        Signature Hash Algorithm Hash: SHA256 (4)
        Signature Hash Algorithm Signature: ECDSA (3)
      Signature Length: 70
      Signature: 304402200cfc9478f7ed24a7f7171f3d6d241311d9d0422faa6d1930fda18e96419304c6…
```

At last, the server sends the server_hello_done message to mark the end of server hello



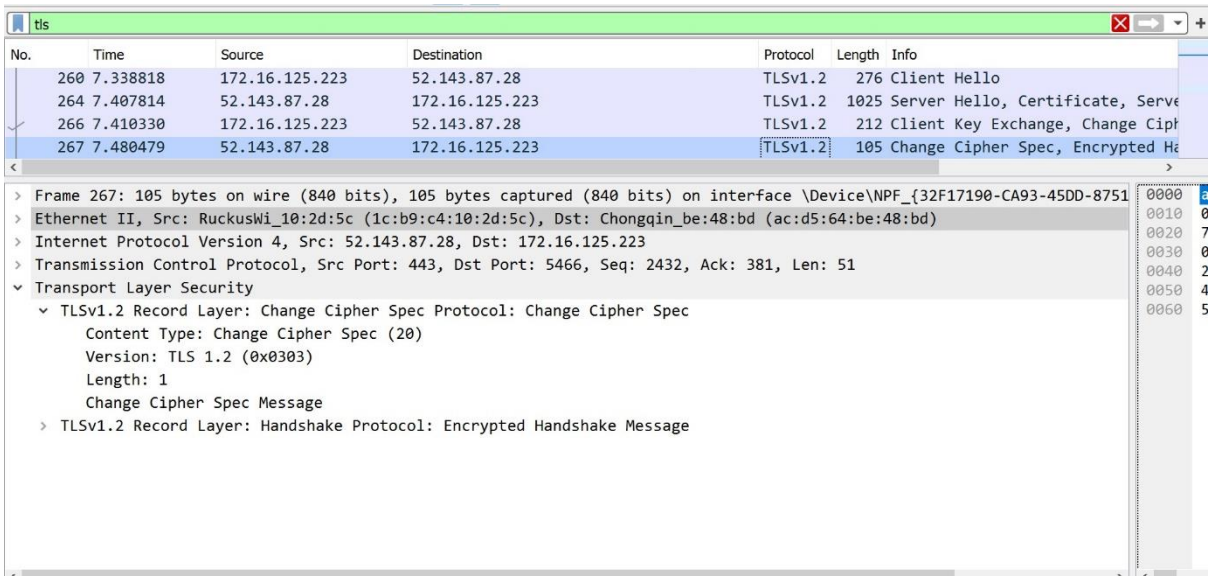
Client key exchange and cipher change spec

In response, the client sends its parameter for key exchange and it acknowledges that the server has chosen a cipher suite.



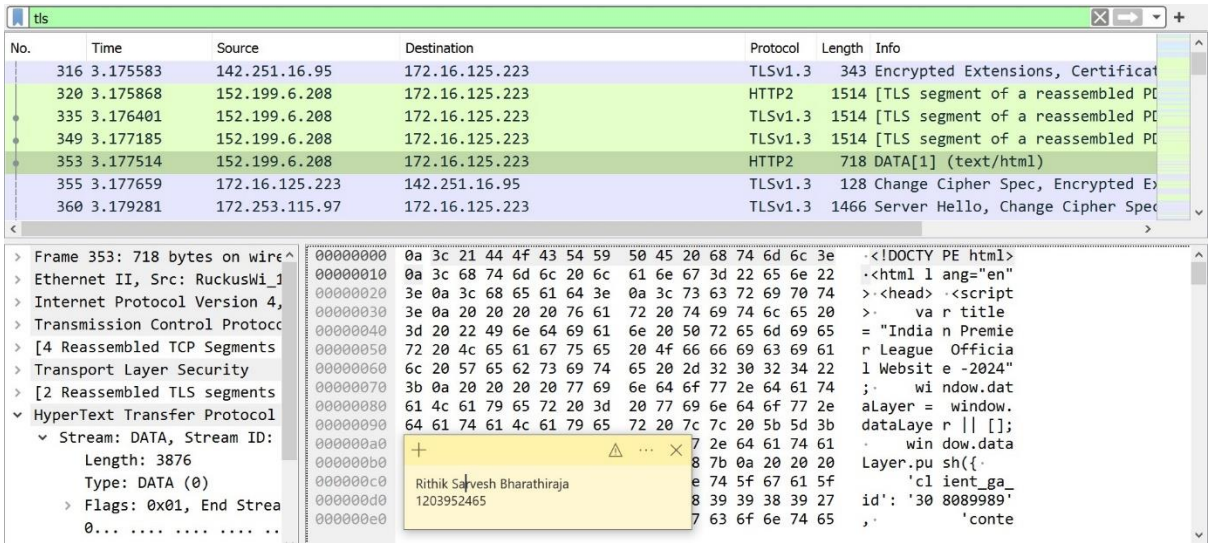
Server reply

Server acknowledges the client's acknowledgement and starts sending in the data after that.



Part 2

After following the given steps, packet capture was done after accessing https://www.iplt20.com website. Using the sslkeylog.txt, wireshark decrypts the data sent in the https. At the end, readable html tags were found in the below tab.


tls
No. Time Source Destination Protocol Length Info
316 3.175583 142.251.16.95 172.16.125.223 TLSv1.3 343 Encrypted Extensions, Certificat
320 3.175868 152.199.6.208 172.16.125.223 HTTP2 1514 [TLS segment of a reassembled PD
335 3.176401 152.199.6.208 172.16.125.223 TLSv1.3 1514 [TLS segment of a reassembled PD
349 3.177185 152.199.6.208 172.16.125.223 TLSv1.3 1514 [TLS segment of a reassembled PD
353 3.177514 152.199.6.208 172.16.125.223 HTTP2 718 DATA[1] (text/html)
355 3.177659 172.16.125.223 142.251.16.95 TLSv1.3 128 Change Cipher Spec, Encrypted Ex
360 3.179281 172.253.115.97 172.16.125.223 TLSv1.3 1466 Server Hello, Change Cipher Spec
Frame 353: 718 bytes on wire
Ethernet II, Src: RuckusWi_1
Internet Protocol Version 4,
Transmission Control Protocc
[4 Reassembled TCP Segments
Transport Layer Security
[2 Reassembled TLS segments
HyperText Transfer Protocol
Stream: DATA, Stream ID:
Length: 3876
Type: DATA (0)
Flags: 0x01, End Strea
0... ....
00000000 0a 3c 21 44 4f 43 54 59 50 45 20 68 74 6d 6c 3e ·<!DOCTY PE html>
00000010 0a 3c 68 74 6d 6c 20 6c 61 6e 67 3d 22 65 6e 22 ·<html l ang="en"
00000020 3e 0a 3c 68 65 61 64 3e 0a 3c 73 63 72 69 70 74 >·<head> ·<script
00000030 3e 0a 20 20 20 20 76 61 72 20 74 69 74 6c 65 20 >· va r title
00000040 3d 20 22 49 6e 64 69 61 6e 20 50 72 65 6d 69 65 = "India n Premie
00000050 72 20 4c 65 61 67 75 65 20 4f 66 66 69 63 69 61 r League Officia
00000060 6c 20 57 65 62 73 69 74 65 20 2d 32 30 32 34 22 l Websit e -2024"
00000070 3b 0a 20 20 20 20 77 69 6e 64 6f 77 2e 64 61 74 ;· wi ndow.dat
00000080 61 4c 61 79 65 72 20 3d 20 77 69 6e 64 6f 77 2e aLayer = window.
00000090 64 61 74 61 4c 61 79 65 72 20 7c 7c 20 5b 5d 3b dataLaye r || [];
000000a0 ·· win dow.data
000000b0 Layer.pu sh({·
000000c0 'cl ient_ga_
000000d0 id': '30 8089989'
000000e0 ,· 'conte
Rithik Sarvesh Bharathiraja
1203952465