

Pandit Deendayal Energy University
School of Technology
Department of ICT
Academic Year: 2022-23
Computer Communication and Networking Lab
20IC306P

Name: Janakar Patel

Roll No: 20BIT061

Experiment-10

Aim: To understand the working of TLS by using Wire-shark.

Software Tools required: Wire-shark.

Capturing packets in an TLS session

No.	Time	Source	Destination	Protocol	Length	Info
1669	6.933174	192.168.137.232	128.119.240.84	TCP	66	52718 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
1670	6.934098	192.168.137.232	128.119.240.84	TCP	66	52719 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
1677	7.027017	192.168.137.232	128.119.240.84	TCP	66	52720 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
1696	7.222465	128.119.240.84	192.168.137.232	TCP	66	443 → 52718 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
1697	7.222537	192.168.137.232	128.119.240.84	TCP	54	52718 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0
1698	7.222945	192.168.137.232	128.119.240.84	TLSv1.2	571	Client Hello
1699	7.239738	128.119.240.84	192.168.137.232	TCP	66	443 → 52719 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
1700	7.239846	192.168.137.232	128.119.240.84	TCP	54	52719 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0
1701	7.240215	192.168.137.232	128.119.240.84	TLSv1.2	571	Client Hello
1708	7.346266	128.119.240.84	192.168.137.232	TCP	66	443 → 52720 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
1709	7.346364	192.168.137.232	128.119.240.84	TCP	54	52720 → 443 [ACK] Seq=1 Ack=1 Win=132096 Len=0
1710	7.346656	192.168.137.232	128.119.240.84	TLSv1.2	571	Client Hello
1799	7.783752	128.119.240.84	192.168.137.232	TCP	54	[TCP Dup ACK 1696#1] 443 → 52718 [ACK] Seq=1 Ack=
1800	7.784326	128.119.240.84	192.168.137.232	TCP	54	[TCP Dup ACK 1696#2] 443 → 52718 [ACK] Seq=1 Ack=
1801	7.784326	128.119.240.84	192.168.137.232	TCP	54	[TCP Dup ACK 1696#3] 443 → 52718 [ACK] Seq=1 Ack=
1802	7.804529	128.119.240.84	192.168.137.232	TCP	54	[TCP Dup ACK 1699#1] 443 → 52719 [ACK] Seq=1 Ack=

A first look at the captured trace

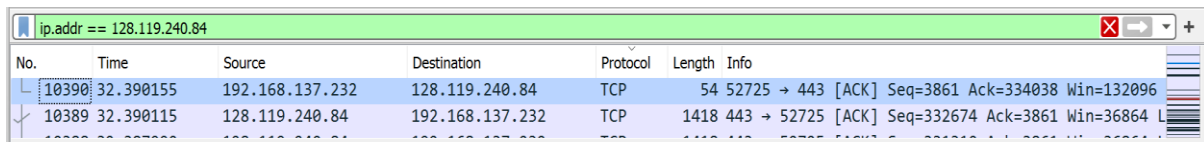
Q.1. What is the packet number in your trace that contains the initial TCP SYN message? (By “packet number,” we meant the number in the “No.” column at the left of the Wireshark display, not the sequence number in the TCP segment itself).

Ans: Packet Number in trace that contains the initial TCP SYN message: 10390

No.	Time	Source	Destination	Protocol	Length	Info
10390	32.390155	192.168.137.232	128.119.240.84	TCP	54	52725 → 443 [ACK] Seq=3861 Ack=334038 Win=132096
10389	32.390115	128.119.240.84	192.168.137.232	TCP	1418	443 → 52725 [ACK] Seq=332674 Ack=3861 Win=36864 L

Q.2. Is the TCP connection set up before or after the first TLS message is sent from client to server?

Ans: Yes, connection is setup after TLS message is sent from client to server.



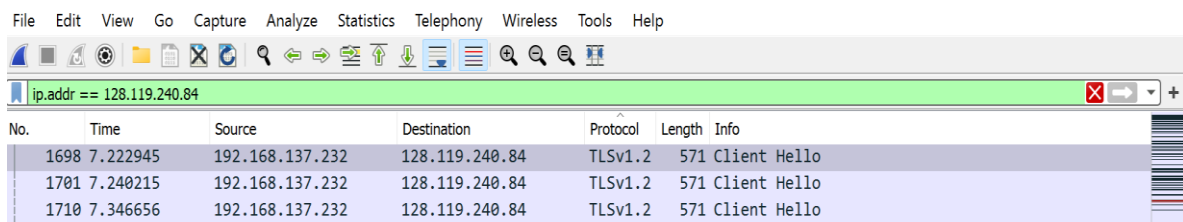
A screenshot of a Wireshark packet capture window. The filter bar at the top shows 'ip.addr == 128.119.240.84'. The packet list shows two TCP packets. Packet 10390 is a SYN packet from 192.168.137.232 to 128.119.240.84. Packet 10389 is an ACK packet from 128.119.240.84 to 192.168.137.232.

No.	Time	Source	Destination	Protocol	Length	Info
10390	32.390155	192.168.137.232	128.119.240.84	TCP	54	52725 → 443 [ACK] Seq=3861 Ack=334038 Win=132096
10389	32.390115	128.119.240.84	192.168.137.232	TCP	1418	443 → 52725 [ACK] Seq=332674 Ack=3861 Win=36864

The TLS Handshake: *Client Hello* message

Q.3. What is the packet number in your trace that contains the TLS *Client Hello* message?

Ans: Packet Number that contains the TLS *Client Hello* message: 1698

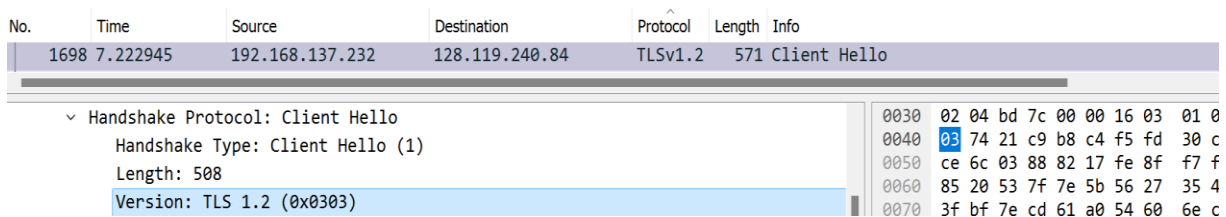


A screenshot of a Wireshark packet capture window. The filter bar at the top shows 'ip.addr == 128.119.240.84'. The packet list shows three TLSv1.2 Client Hello packets. Packet 1698 is the first Client Hello from 192.168.137.232 to 128.119.240.84.

No.	Time	Source	Destination	Protocol	Length	Info
1698	7.222945	192.168.137.232	128.119.240.84	TLSv1.2	571	Client Hello
1701	7.240215	192.168.137.232	128.119.240.84	TLSv1.2	571	Client Hello
1710	7.346656	192.168.137.232	128.119.240.84	TLSv1.2	571	Client Hello

Q.4. What version of TLS is your client running, as declared in the *Client Hello* message?

Ans: 1.2 version of TLS i.e. TLS 1.2 is the client running as declared in the *Client Hello* message.



A screenshot of the Wireshark packet details pane for packet 1698. It shows the handshake protocol as Client Hello, with a length of 508 bytes. The version is TLS 1.2 (0x0303). The raw packet data is shown in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1698	7.222945	192.168.137.232	128.119.240.84	TLSv1.2	571	Client Hello

Handshake Protocol: Client Hello
Handshake Type: Client Hello (1)
Length: 508
Version: TLS 1.2 (0x0303)

0030 02 04 bd 7c 00 00 16 03 01 0
0040 74 21 c9 b8 c4 f5 fd 30 c
0050 ce 6c 03 88 82 17 fe 8f f7 f
0060 85 20 53 7f 7e 5b 56 27 35 4
0070 3f bf 7e cd 61 a0 54 60 6e c

Q.5. How many cipher suites are supported by your client, as declared in the *Client Hello* message? A cipher suite is a set of related cryptographic algorithms that determine how session keys will be derived, and how data will be encrypted and be digitally signed via a HMAC algorithm.'

Ans: 17 cipher suites are supported by the client as declared in the *Client Hello* message.

No.	Time	Source	Destination	Protocol	Length	Info
	1698.7.222945	192.168.137.232	128.119.240.84	TLSv1.2	571	Client Hello
v Cipher Suites (17 suites)						0000 aa cd c4 bb 32 e7 d8 f8 83 04 2f ef 08 00
Cipher Suite: Reserved (GREASE) (0x5a5a)						0010 02 2d fb e6 40 00 80 06 00 00 c0 a8 89 e8
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)						0020 f0 54 cd ee 01 bb 2c b8 b7 e9 52 9c 1c 32
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)						0030 02 04 bd 7c 00 00 16 03 01 02 00 01 00 01
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)						0040 03 74 21 c9 b8 c4 f5 fd 30 c4 a2 ab 56 b1
Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)						0050 ce 6c 03 88 82 17 fe 8f f7 f8 8b e6 08 5b
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)						0060 85 20 53 7f 7e 5b 56 27 35 4b 27 f8 67 b2
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)						0070 3f bf 7e cd 61 a0 54 60 6e cc a5 06 31 a1
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)						0080 84 67 00 22 5a 5a 13 01 13 02 13 02 13 03
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)						0090 c0 2f c0 2c c0 30 cc a9 cc a8 c0 13 c0 14
Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xc031)						00a0 00 9d 00 2f 00 35 01 00 01 91 3a 3a 00 00
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc032)						00b0 00 0a 00 08 2a 2a 00 1d 00 17 00 18 00 33
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc013)						00c0 00 29 2a 2a 00 01 00 00 1d 00 20 e7 e4 ff
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc014)						00d0 12 2c 2d 80 2b 7d c0 51 fa d3 0c 79 6a 0c
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc015)						00e0 69 82 aa 44 0a 59 37 c7 9e c3 6d 00 0d 00
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc016)						00f0 10 04 03 08 04 04 01 05 03 08 05 05 01 08
Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)						0100 01 ff 01 00 01 00 44 69 00 05 00 03 02 68
Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)						0110 23 00 00 00 10 00 0e 00 0c 02 68 32 08 68
Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)						0120 70 2f 31 2e 31 00 05 00 05 01 00 00 00 00
Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)						0130 00 02 01 01 00 1b 00 03 02 00 02 00 2b 00
						0140 5a 5a 03 04 03 03 00 12 00 00 00 0b 00 02
						0150 00 17 00 00 00 00 00 17 00 15 00 00 12 77

Q.6. Your client generates and sends a string of “random bytes” to the server in the *Client Hello* message. What are the first two hexadecimal digits in the random bytes field of the *Client Hello* message? Enter the two hexadecimal digits (without spaces between the hex digits and without any leading '0x' , using lowercase letters where needed).

Ans: First two hexadecimal digits in the random bytes field of the *Client Hello* message: c4

No.	Time	Source	Destination	Protocol	Length	Info
1698	7.222945	192.168.137.232	128.119.240.84	TLSv1.2	571	Client Hello
▼ Transport Layer Security						
▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello						
Content Type: Handshake (22)						
Version: TLS 1.0 (0x0301)						
Length: 512						
▼ Handshake Protocol: Client Hello						
Handshake Type: Client Hello (1)						
Length: 508						
Version: TLS 1.2 (0x0303)						
▼ Random: 7421c9b8c4f5fd30c4a2ab56b167b9ce6c03888217fe8ff7f88be6085b55f485						
GMT Unix Time: Sep 28, 2031 19:46:24.000000000 India Standard Time						
Random Bytes: c4f5fd30c4a2ab56b167b9ce6c03888217fe8ff7f88be6085b55f485						

0040

03 74 21 c9 b8 c4 f5 fd 30 c4

0050

ce 6c 03 88 82 17 fe 8f f7 f8

0060

85 20 53 7f 7e 5b 56 27 35 4b

0070

3f bf 7e cd 61 a0 54 60 6e cc

0080

84 67 00 22 5a 5a 13 01 13 02

0090

c0 2f c0 2c c0 30 cc a9 cc a8

00a0

00 9d 00 2f 00 35 01 00 01 91

00b0

00 0a 00 08 2a 2a 00 1d 00 17

00c0

00 29 2a 2a 00 01 00 00 1d 00

00d0

12 2c 2d 80 2b 7d c0 51 fa d3

00e0

69 82 aa 44 0a 59 37 c7 9e c3

00f0

10 04 03 08 04 04 01 05 03 08

0100

01 ff 01 00 01 00 44 69 00 05

0110

23 00 00 00 10 00 0e 00 0c 02

0120

70 2f 31 2e 31 00 05 00 05 01

Q.7. What is the purpose(s) of the “random bytes” field in the *Client Hello* message? Note: you’ll have to do some searching and reading to get the answer to this question; see section 8.6 and in [RFC 5246](#) (section 8.1 in RFC 5246 in particular).

Ans: When resuming a session, the same master key is used to generate key block. So, use of client and server random bytes ensures that key block will be different in every handshake.

The TLS Handshake: *Server Hello* message

Q.8. What is the packet number in your trace that contains the TLS *Server Hello* message?

Ans: Packet Number that contains the TLS *Server Hello* message: 1861

ip.addr == 128.119.240.84						
No.	Time	Source	Destination	Protocol	Length	Info
1861	8.178996	128.119.240.84	192.168.137.232	TLSv1.2	1418	Server Hello
1866	8.184840	128.119.240.84	192.168.137.232	TLSv1.2	1277	Certificate, Server Key Exchange, Server Hello Do
1868	8.186938	192.168.137.232	128.119.240.84	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypte
1871	8.241223	128.119.240.84	192.168.137.232	TLSv1.2	1418	Server Hello

Q.9. Which cipher suite has been chosen by the server from among those offered in the earlier *Client Hello* message?

Ans: Cipher Suite chosen by the server from among those offered in the earlier *Client Hello* message: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

ip.addr == 128.119.240.84						
No.	Time	Source	Destination	Protocol	Length	Info
1861	8.178996	128.119.240.84	192.168.137.232	TLSv1.2	1418	Server Hello
Version: TLS 1.2 (0x0303)						
Length: 65						
Handshake Protocol: Server Hello						
Handshake Type: Server Hello (2)						
Length: 61						
Version: TLS 1.2 (0x0303)						
Random: 19f0d5484f29bdb941b559234652c98f743de5610582702658c08e85106ade3b						
GMT Unix Time: Oct 17, 1983 10:10:40.000000000 India Standard Time						
Random Bytes: 4f29bdb941b559234652c98f743de5610582702658c08e85106ade3b						
Session ID Length: 0						
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)						
0060 3b 00 c0 2f 00 00 15 00 00 00 00 f-						
0070 00 0b 00 04 03 00 01 02 00 23 00 0-						
0080 21 0b 00 13 1d 00 13 1a 00 07 32 3-						
0090 82 06 16 a0 03 02 01 02 02 10 30 9-						
00a0 1c de 05 eb 63 eb 08 72 72 71 30 0-						
00b0 48 86 f7 0d 01 01 0b 05 00 30 76 3-						
00c0 03 55 04 06 13 02 55 53 31 0b 30 0-						
00d0 08 13 02 4d 49 31 12 30 10 06 03 5-						
00e0 41 6e 6e 20 41 72 62 6f 72 31 12 3-						
00f0 04 0a 13 09 49 6e 74 65 72 6e 65 7-						
0100 0f 06 03 55 04 0b 13 08 49 6e 43 6-						
0110 31 1f 30 1d 06 03 55 04 03 13 16 4-						
0120 6d 6f 6e 20 52 53 41 20 53 65 72 7-						

Q.10. Does the *Server Hello* message contain random bytes, similar to how the *Client Hello* message contained random bytes? And if so, what is/are their purpose(s)?

Ans: Yes, the *Server Hello* message contain random bytes but not similar to the *Client Hello* message contained random bytes.

No.	Time	Source	Destination	Protocol	Length	Info
1861	8.178996	128.119.240.84	192.168.137.232	TLSv1.2	1418	Server Hello
Version: TLS 1.2 (0x0303)						
Length: 65						
Handshake Protocol: Server Hello						
Handshake Type: Server Hello (2)						
Length: 61						
Version: TLS 1.2 (0x0303)						
Random: 19f0d5484f29bdb941b559234652c98f743de5610582702658c08e85106ade3b						
GMT Unix Time: Oct 17, 1983 10:10:40.000000000 India Standard Time						
Random Bytes: 4f29bdb941b559234652c98f743de5610582702658c08e85106ade3b						
Session ID Length: 0						
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)						
Compression Method: null (0)						
0060 3b 00 c0 2f 00 00 15 00 00 00 00 00						
0070 00 0b 00 04 03 00 01 02 00 23 00 00						
0080 21 0b 00 13 1d 00 13 1a 00 07 32 30						
0090 82 06 16 a0 03 02 01 02 02 10 30 90						
00a0 1c de 05 eb 63 eb 08 72 72 71 30 00						
00b0 48 86 f7 0d 01 01 0b 05 00 30 76 30						
00c0 03 55 04 06 13 02 55 53 31 0b 30 00						
00d0 08 13 02 4d 49 31 12 30 10 06 03 50						
00e0 41 6e 6e 20 41 72 62 6f 72 31 12 30						
00f0 04 0a 13 09 49 6e 74 65 72 6e 65 70						
0100 0f 06 03 55 04 0b 13 08 49 6e 43 60						
0110 31 1f 30 1d 06 03 55 04 03 13 16 40						
0120 6d 6f 6e 20 52 53 41 20 53 65 72 70						
0130 41 30 1e 17 0d 32 33 30 33 31 30 30						

Q.11. What is the packet number in your trace for the TLS message part that contains the public key certificate for the www.cics.umass.edu server (actually the www.cs.umass.edu server)?

Ans: Packet Number that contains the TLS *Server Hello* message: 1876

The image shows a Wireshark packet capture of a TLSv1.2 Server Hello message. The packet number is 1876, source IP is 128.119.240.84, and destination IP is 192.168.137.232. The message type is 'Certificate, Server Key Exchange, Server Hello Do...'. The details pane shows the following structure:

- Length: 333
- Handshake Protocol: Server Key Exchange
 - Handshake Type: Server Key Exchange (12)
 - Length: 329
 - EC Diffie-Hellman Server Params
 - Curve Type: named_curve (0x03)
 - Named Curve: secp256r1 (0x0017)
 - Pubkey Length: 65
 - Pubkey: 04b277958b359784f4c2968d81877e985d68d57d1989a2d8fe562187470681bfcf4
 - Signature Algorithm: rsa_pkcs1_sha256 (0x0401)
 - Signature Length: 256
 - Signature: 12540f0e7865523af5c2e39fa310a8d70b119a5c23aed8ee3292edaa9e53cde2

The packet bytes pane shows the raw data in hexadecimal and ASCII format.

Q.12. A server may return more than one certificate. If more than one certificate is returned, are all of these certificates for www.cs.umass.edu? If not all are for www.cs.umass.edu, then who are these other certificates for? You can determine who the certificate is for by checking the id-at-commonName field in the returned certificate.

Ans: Yes there are more than one certificate. There are total 3 certificates. only one certificate for www.cs.umass.edu server other two are for 'IN Common RSA server CA' And 'USERTrust RSA Certification Authority'.

The image shows a Wireshark packet capture of a TLSv1.2 Certificate message. The packet number is 1876, source IP is 128.119.240.84, and destination IP is 192.168.137.232. The message type is 'Certificate, Server Key Exchange, Server Hello Do...'. The details pane shows the following structure:

- Handshake Protocol: Certificate
 - Handshake Type: Certificate (11)
 - Length: 4893
 - Certificates Length: 4890
 - Certificates (4890 bytes)
 - Certificate Length: 1842
 - Certificate: 3082072e30820616a00302010202103090854915311cde05eb63eb087272
 - Certificate Length: 1533
 - Certificate: 308205f9308203e1a00302010202104720d0fa85461a7e17a16402918463
 - Certificate Length: 1506
 - Certificate: 308205de308203c6a003020102021001fd6d30fca3ca51a81bbc640e3503

The packet bytes pane shows the raw data in hexadecimal and ASCII format.

Q.13. What is the name of the certification authority that issued the certificate for id-at-commonName=www.cs.umass.edu?

Ans: Name of the certification authority: Computer Science University of Massachusetts.

Q. 14. What digital signature algorithm is used by the CA to sign this certificate? Hint: this information can be found in signature subfield of the SignedCertificate field of the certificate for www.cs.umass.edu.

Ans: RSA is used by the CA to sign certificate.

ip.addr == 128.119.240.84						
No.	Time	Source	Destination	Protocol	Length	Info
1876	8.246968	128.119.240.84	192.168.137.232	TLSv1.2	1277	Certificate, Server Key Exchange, Server Hello Done

```

Handshake Type: Certificate (11)
Length: 4893
Certificates Length: 4890
  Certificates (4890 bytes)
    Certificate Length: 1842
    Certificate: 3082072e30820616a00302010202103090854915311cde05eb63eb08727271300d06092a... (id-at-commonName=www.cs.uma:
      signedCertificate
        version: v3 (2)
        serialNumber: 0x3090854915311cde05eb63eb08727271
        signature (sha256WithRSAEncryption)
          Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)

```

Q.15. Let's take a look at what a real public key looks like! What are the first four hexadecimal digits of the modulus of the public key being used by www.cics.umass.edu? Enter the four hexadecimal digits (without spaces between the hex digits and without any leading '0x', using lowercase letters where needed, and including any leading 0s after '0x'). Hint: this information can be found in subjectPublicKeyInfo subfield of the SignedCertificate field of the certificate for www.cs.umass.edu.

Ans: First four hexadecimal digits of the modulus of the public key being used by www.cics.umass.edu:3090

ip.addr == 128.119.240.84						
No.	Time	Source	Destination	Protocol	Length	Info
1876	8.246968	128.119.240.84	192.168.137.232	TLSv1.2	1277	Certificate, Server Key Exchange, Server Hello Done

```

  Certificates (4890 bytes)
    Certificate Length: 1842
    Certificate: 3082072e30820616a00302010202103090854915311cde05eb63eb08727271300d06092a... (id-at-commonName=www.cs.uma:
      signedCertificate
        version: v3 (2)
        serialNumber: 0x3090854915311cde05eb63eb08727271

```

Q.16. Look in your trace to find messages between the client and a CA to get the CA's public key information, so that the client can verify that the CA-signed certificate sent by the server is indeed valid and has not been forged or altered. Do you see such message in your trace? If so, what is the number in the trace of the first packet sent from your client to the CA? If not, explain why the client did not contact the CA.

Ans: No, client didn't contact CA.

Q.17. What is the packet number in your trace for the TLS message part that contains the *Server Hello Done* TLS record?

Ans: Packet Number that contains the TLS *Server Hello Done* message: 1866

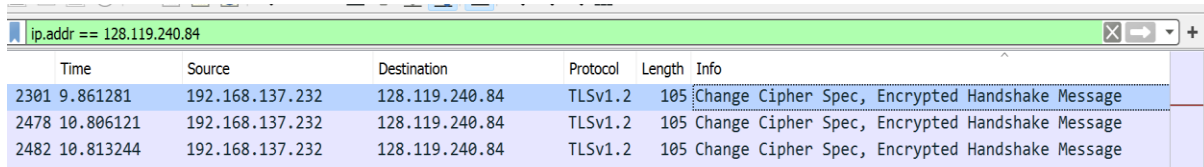
ip.addr == 128.119.240.84

	Time	Source	Destination	Protocol	Length	Info	
	1866	8.184840	128.119.240.84	192.168.137.232	TLSv1.2	1277	Certificate, Server Key Exchange, Server Hello Done

The TLS Handshake: wrapping up the handshake

Q.18. What is the packet number in your trace for the TLS message that contains the public key information, *Change Cipher Spec*, and *Encrypted Handshake* message, being sent from client to server?

Ans: Packet Number that contains the public key information, *Change Cipher Spec*, and *Encrypted Handshake* message: 2301



A screenshot of a Wireshark packet capture window. The filter bar at the top shows 'ip.addr == 128.119.240.84'. The packet list table below shows three packets:

Time	Source	Destination	Protocol	Length	Info
2301 9.861281	192.168.137.232	128.119.240.84	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
2478 10.806121	192.168.137.232	128.119.240.84	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
2482 10.813244	192.168.137.232	128.119.240.84	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message

Q.19. Does the client provide its own CA-signed public key certificate back to the server? If so, what is the packet number in your trace containing your client's certificate?

Ans: No, the client doesn't client provide its own CA-signed public key certificate back to the server.

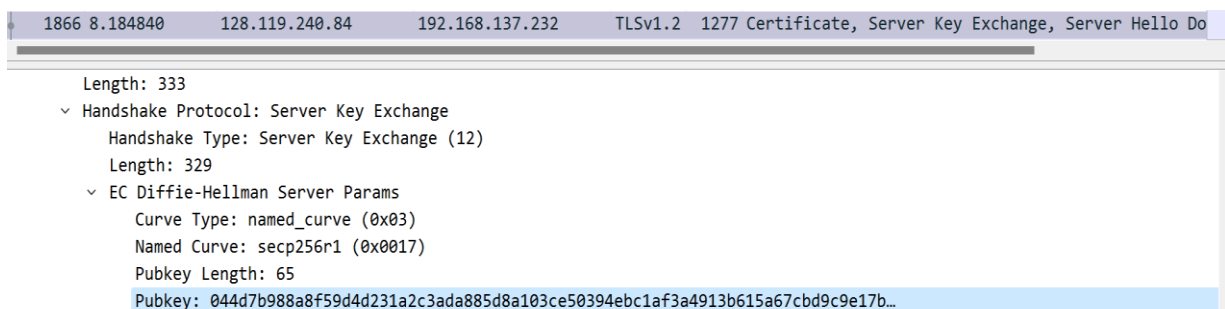
Application data

Q.20. What symmetric key cryptography algorithm is being used by the client and server to encrypt application data (in this case, HTTP messages)?

Ans: 'EC Diffie-Hellman Client Params' Algorithm is the symmetric key cryptography algorithm is being used by the client and server to encrypt application data.

Q.21 In which of the TLS messages is this symmetric key cryptography algorithm finally decided and declared?

Ans: In 1866 packet number, TLS message, symmetric key cryptography algorithm finally decided and declared.



A screenshot of a Wireshark packet capture window showing the details of packet 1866. The packet list table at the top shows:

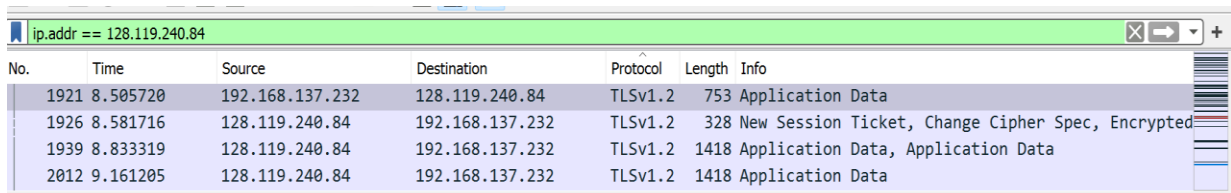
Time	Source	Destination	Protocol	Length	Info
1866 8.184840	128.119.240.84	192.168.137.232	TLSv1.2	1277	Certificate, Server Key Exchange, Server Hello Do

The details pane below shows the following structure:

- Length: 333
 - Handshake Protocol: Server Key Exchange
 - Handshake Type: Server Key Exchange (12)
 - Length: 329
 - EC Diffie-Hellman Server Params
 - Curve Type: named_curve (0x03)
 - Named Curve: secp256r1 (0x0017)
 - Pubkey Length: 65
 - Pubkey: 044d7b988a8f59d4d231a2c3ada885d8a103ce50394ebc1af3a4913b615a67cbd9c9e17b...

Q.22 What is the packet number in your trace for the first encrypted message carrying application data from client to server?

Ans: Packet Number that contains the first encrypted message carrying application data from client to server: 1921



No.	Time	Source	Destination	Protocol	Length	Info
1921	8.505720	192.168.137.232	128.119.240.84	TLSv1.2	753	Application Data
1926	8.581716	128.119.240.84	192.168.137.232	TLSv1.2	328	New Session Ticket, Change Cipher Spec, Encrypted
1939	8.833319	128.119.240.84	192.168.137.232	TLSv1.2	1418	Application Data, Application Data
2012	9.161205	128.119.240.84	192.168.137.232	TLSv1.2	1418	Application Data

Q.23. What do you think the content of this encrypted application-data is, given that this trace was generated by fetching the homepage of www.cics.umass.edu?

Ans: Alert