

Lab 07_TLS

I. Capturing and analyzing Ethernet frames

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).

The packet number in your trace that contains the initial TCP SYN message: 17

The screenshot shows a Wireshark packet capture of a TCP connection. Packet 17 is highlighted, showing a SYN message from 192.168.1.245 to 128.119.240.84. The packet details pane shows the Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol layers. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
17	14:29:29.400	192.168.1.245	128.119.240.84	TCP	78	51146 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=465227084 TSecr=0 SACK_PERM
26	14:29:29.479	128.119.240.84	192.168.1.245	TCP	74	443 → 51146 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSval=248562440 TSecr=465227084
27	14:29:29.479	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=465227082 TSecr=248562440
28	14:29:29.479	192.168.1.245	128.119.240.84	TLSv1.2	583	Client Hello
31	14:29:29.557	128.119.240.84	192.168.1.245	TCP	66	443 → 51146 [ACK] Seq=1 Ack=518 Win=30080 Len=0 TSval=248562518 TSecr=465227082
32	14:29:29.558	128.119.240.84	192.168.1.245	TLSv1.2	1514	Server Hello
33	14:29:29.558	128.119.240.84	192.168.1.245	TCP	1266	443 → 51146 [PSH, ACK] Seq=2897 Ack=518 Win=30080 Len=1200 TSval=248562518 TSecr=465227082
34	14:29:29.558	128.119.240.84	192.168.1.245	TCP	66	51146 → 443 [ACK] Seq=518 Ack=2897 Win=129600 Len=0 TSval=465227160 TSecr=248562518
35	14:29:29.558	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=2897 Win=129600 Len=0 TSval=465227160 TSecr=248562518
36	14:29:29.558	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=2897 Win=129600 Len=0 TSval=465227160 TSecr=248562518

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

TLS is built on top of TCP/IP, TCP connection has to be established (and thus the TCP handshake successfully finished) before the TLS handshake can start, the client must first complete the 3-way TCP handshake with the server.



The screenshot shows the same Wireshark packet capture as before, but now packet 35 is highlighted, showing the final ACK message from the client to the server. The packet details pane shows the Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol layers. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
17	14:29:29.400	192.168.1.245	128.119.240.84	TCP	78	51146 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=465227084 TSecr=0 SACK_PERM
26	14:29:29.479	128.119.240.84	192.168.1.245	TCP	74	443 → 51146 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSval=248562440 TSecr=465227084
27	14:29:29.479	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=465227082 TSecr=248562440
28	14:29:29.479	192.168.1.245	128.119.240.84	TLSv1.2	583	Client Hello
31	14:29:29.557	128.119.240.84	192.168.1.245	TCP	66	443 → 51146 [ACK] Seq=1 Ack=518 Win=30080 Len=0 TSval=248562518 TSecr=465227082
32	14:29:29.558	128.119.240.84	192.168.1.245	TLSv1.2	1514	Server Hello
33	14:29:29.558	128.119.240.84	192.168.1.245	TCP	1266	443 → 51146 [PSH, ACK] Seq=2897 Ack=518 Win=30080 Len=1200 TSval=248562518 TSecr=465227082
34	14:29:29.558	128.119.240.84	192.168.1.245	TCP	66	51146 → 443 [ACK] Seq=518 Ack=2897 Win=129600 Len=0 TSval=465227160 TSecr=248562518
35	14:29:29.558	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=2897 Win=129600 Len=0 TSval=465227160 TSecr=248562518
36	14:29:29.558	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=2897 Win=129600 Len=0 TSval=465227160 TSecr=248562518

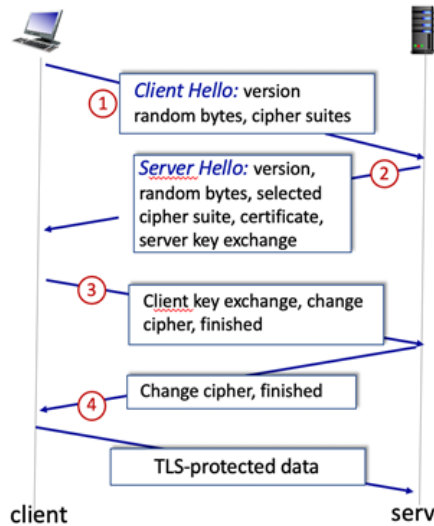


Figure 2: TLS handshaking

- The TLS Handshake: Client Hello message

The image shows a Wireshark capture of a TLS Client Hello message. The packet list on the left shows the following details:

- 28 14:29:29.479 192.168.1.245 128.119.240.84 TLSv1.2 583 Client Hello
- 31 14:29:29.557 128.119.240.84 192.168.1.245 TCP 66 443 → 51146 [ACK] Seq=1 Ack=518 Win=30080 Len=0 TSval=248562518 TSecr=248527082
- 32 14:29:29.558 128.119.240.84 192.168.1.245 TLSv1.2 1514 Server Hello

The packet details pane shows the following structure for the Client Hello message:

- Handshake Protocol: Client Hello (1)
 - Length: 508
 - Version: TLS 1.2 (0x0303)
 - Random: 421623e04b909a780b955b1a679367e8af0312ec2362979794c50c162089004b
 - GMT Unix Time: Feb 19, 2005 00:20:32.000000000 SE Asia Standard Time
 - Random Bytes: 4b909a780b955b1a679367e8af0312ec2362979794c50c162089004b
 - Session ID Length: 32
 - Session ID: 9cb2d5b5089902aa2ad429db71eb11800afb2c4b0d335cc63f7bcc8defe8787d2
 - Cipher Suites Length: 34
 - Cipher Suites (17 suites):
 - TLS_AES_128_GCM_SHA256 (0x1301)
 - TLS_CHACHA20_POLY1305_SHA256 (0x1303)
 - TLS_AES_256_GCM_SHA384 (0x1302)
 - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
 - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
 - TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
 - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc0ca8)
 - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
 - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
 - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
 - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc009)
 - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
 - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc014)
 - TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009d)
 - TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009c)
 - TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)

The packet bytes pane shows the raw data of the message, including the random bytes and the cipher suite list.

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

the packet number in your trace that contains the TLS *Client Hello* message is: 28

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

version of TLS is your client running: 1.0

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.

They are 17 Cipher suites supported by your client.

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). *Hint: be careful to fully dig into the Random field to find the Random Bytes subfield (do not consider the GMT UNIX Time subfield of Random).*

The first two hexadecimal digits in the random bytes field of the *Client Hello* message are: 4b

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).

Random bytes (28 bytes) should be generated by the client using a secure random number generator. The source of entropy for random number generation (to generate the key for encryption) will depend on the operating system and implementation of the client software.

Its main purpose is to stop replay attacks. In essence, the addition of this unique number prevents an attacker from being able to send the client the very same packets that it had received in a previous connection.

The image shows a Wireshark packet capture of a TLSv1.2 Client Hello message. The packet list pane shows the following packets:

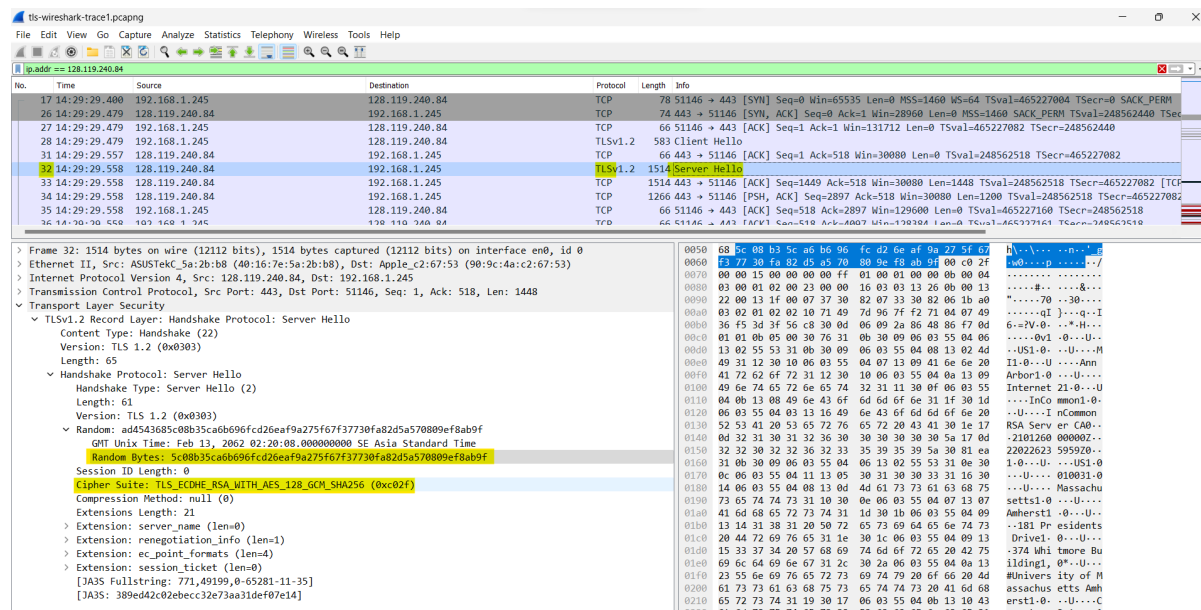
- 17 14:29:29.400 192.168.1.245 128.119.240.84 TCP 78 51146 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=465227084 TSecr=0 SACK_PERM=0
- 26 14:29:29.479 128.119.240.84 192.168.1.245 TCP 74 443 → 51146 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=0 TSval=248562440 TSecr=0
- 27 14:29:29.479 192.168.1.245 128.119.240.84 TCP 66 51146 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=465227082 TSecr=248562440
- 28 14:29:29.479 192.168.1.245 128.119.240.84 TLSv1.2 583 Client Hello
- 31 14:29:29.557 128.119.240.84 192.168.1.245 TCP 66 443 → 51146 [ACK] Seq=1 Ack=518 Win=30080 Len=0 TSval=248562518 TSecr=465227082
- 32 14:29:29.558 128.119.240.84 192.168.1.245 TLSv1.2 1514 Server Hello

The packet details pane shows the following structure for the selected packet (28):

- Frame 28: 583 bytes on wire (4664 bits), 583 bytes captured (4664 bits) on interface en0, id 0
- Ethernet II, Src: Apple_C2:67:53 (90:9c:4a:c2:67:53), Dst: ASUSTek_5a:2b:b8 (40:16:7e:5a:2b:b8)
- Internet Protocol Version 4, Src: 192.168.1.245, Dst: 128.119.240.84
- Transmission Control Protocol, Src Port: 51146, Dst Port: 443, Seq: 1, Ack: 1, Len: 517
- 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: 421623c4b090a780b955b1a679367e8af0312ec2362979794c50c162089004b
 - GMT Unix Time: Feb 19, 2005 00:20:32.000000000 SE Asia Standard Time
 - Random Bytes: 4b090a780b955b1a679367e8af0312ec2362979794c50c162089004b
 - Session ID Length: 32
 - Session ID: 9cb2d50509092aa2ad429db71eb11800afb2c4b0d335cc63f7bcc8defe8787d2
 - Cipher Suites Length: 34
 - Cipher Suites (17 suites)
 - Compression Methods Length: 1
 - Compression Methods (1 method)
 - Extensions Length: 401
 - Extension: server_name (len=23)
 - Extension: extended_master_secret (len=0)
 - Extension: renegotiation_info (len=1)
 - Extension: supported_groups (len=14)
 - Extension: ec_point_formats (len=2)
 - Extension: session_ticket (len=0)
 - Extension: application_layer_protocol_negotiation (len=14)
 - Extension: status_request (len=5)

The packet bytes pane shows the raw data of the Client Hello message, with the random bytes field highlighted in blue.

- **The TLS Handshake: Server Hello message**



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

The packet number in your trace that contains the TLS *Server Hello* message is: 32

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

Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

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)?

Server: Random Bytes: 5c08b35ca6b696fcd26eaf9a275f67f37730fa82d5a570809ef8ab9f

Client: Random Bytes: 4b909a780b955b1a679367e8af0312ec2362979794c50c162089004b

Different.

- **The TLS Handshake: Server Hello message - public key certificate**

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)?

The packet number in your trace for the TLS message part that contains the public key certificate for the www.cics.umass.edu server is 37

No.	Time	Source	Destination	Protocol	Length	Info
7	14:29:27.922	34.226.161.166	192.168.1.245	TLSv1.2	599	Application Data
12	14:29:27.926	192.168.1.245	34.226.161.166	TLSv1.2	496	Application Data
28	14:29:29.479	192.168.1.245	128.119.240.84	TLSv1.2	583	Client Hello
32	14:29:29.558	128.119.240.84	192.168.1.245	TLSv1.2	1514	Server Hello
37	14:29:29.559	128.119.240.84	192.168.1.245	TLSv1.2	1294	Certificate, Server Key Exchange, Server Hello Done
39	14:29:29.571	192.168.1.245	128.119.240.84	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message

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.

If more than one certificate is returned, are all of these certificates not for www.cs.umass.edu.

Certificate: `id-at-commonName=InCommon RSA Server CA`

Certificate: `id-at-commonName=USERTrust RSA Certification Authority`

```

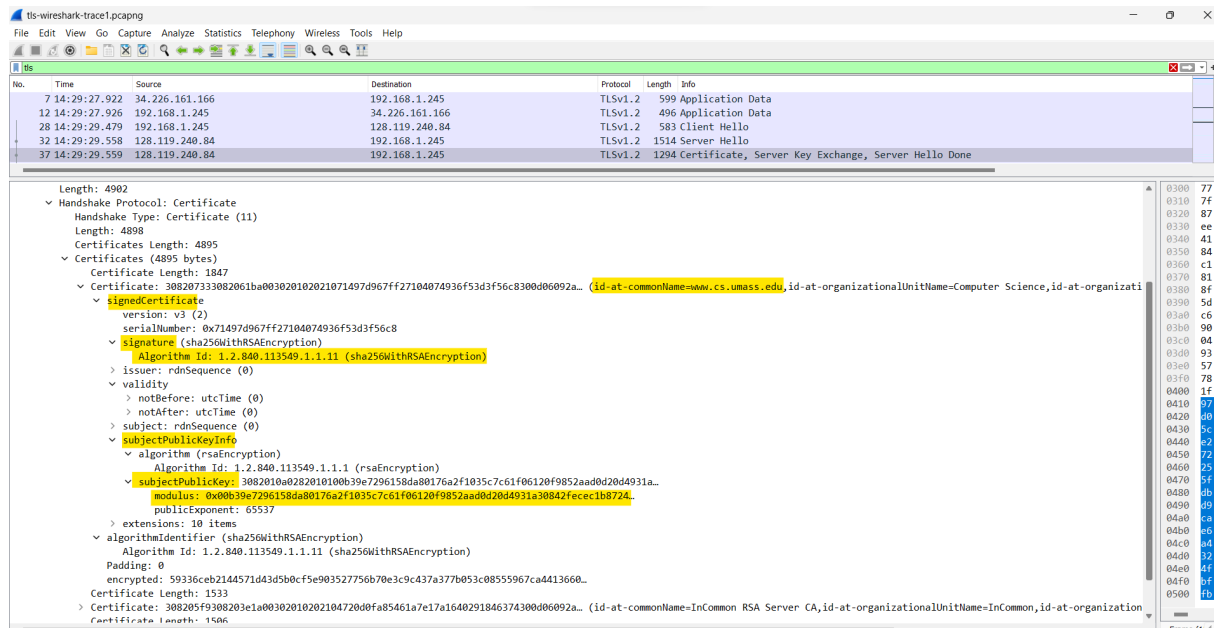
> Frame 37: 1294 bytes on wire (10352 bits), 1294 bytes captured (10352 bits) on interface en0, id 0
> Ethernet II, Src: ASUSTekC_5a:2b:b8 (48:16:7e:5a:2b:b8), Dst: Apple_c2:67:53 (90:9c:4a:c2:67:53)
> Internet Protocol Version 4, Src: 128.119.240.84, Dst: 192.168.1.245
> Transmission Control Protocol, Src Port: 443, Dst Port: 51146, Seq: 4097, Ack: 518, Len: 1228
> [4 Reassembled TCP Segments (4907 bytes): #32(1378), #33(1448), #34(1200), #37(881)]
< Transport Layer Security
  < TLSv1.2 Record Layer: Handshake Protocol: Certificate
    < Content Type: Handshake (22)
      < Version: TLS 1.2 (0x0303)
        < Length: 4902
          < Handshake Protocol: Certificate
            < Handshake Type: Certificate (11)
              < Length: 4898
                < Certificates Length: 4895
                  < Certificates (4895 bytes)
                    < Certificate Length: 1847
                      < Certificate: 308207333082061ba003020102021071497d967ff27104074936f53d3f56c8300d06092a... (id-at-commonName=www.cs.umass.edu,id-at-organizationalUnitName=Computer Science,id-at-organizationName=University of Massachusetts Amherst,id-at-stre
                    < Certificate: 308205f9308203e1a00302010202104720d0fa85461a7e17a1640291846374300d06092a... (id-at-commonName=InCommon RSA Server CA,id-at-organizationalUnitName=InCommon,id-at-organizationName=The USERTRUST Network
                    < Certificate: 308205de308203c6a003020102021001fd6d30fca3ca51a81bbcb640e35032d300d06092a... (id-at-commonName=USERTrust RSA Certification Authority,id-at-organizationName=The USERTRUST Network
  < Transport Layer Security
  
```

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

Certificate:
 308207333082061ba003020102021071497d967ff27104074936f53d3f56c8300d06092a...
 (id-at-commonName=www.cs.umass.edu,id-at-organizationalUnitName=Computer Science,id-at-organizationName=University of Massachusetts Amherst,id-at-stre

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.

Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)



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.

The first four hexadecimal digits of the modulus of the public key being used by www.cics.umass.edu are: 00b3

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.

Not see. The client did not contact the CA because the certificate is signed by a CA about which the client has no knowledge at all. In this case the client treats the certificate as invalid.

The *Server Hello* message is always terminated by an explicit *Server Hello Done* record.

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

The packet number in your trace for the TLS message part that contains the *Server Hello Done* TLS record is: 37

No.	Time	Source	Destination	Protocol	Length	Info
7	14:29:27.922	34.226.161.166	192.168.1.245	TLSv1.2	599	Application Data
12	14:29:27.926	192.168.1.245	34.226.161.166	TLSv1.2	496	Application Data
28	14:29:29.479	192.168.1.245	128.119.240.84	TLSv1.2	583	Client Hello
32	14:29:29.558	128.119.240.84	192.168.1.245	TLSv1.2	1514	Server Hello
37	14:29:29.559	128.119.240.84	192.168.1.245	TLSv1.2	1294	Certificate, Server Key Exchange, Server Hello Done

> Frame 37: 1294 bytes on wire (10352 bits), 1294 bytes captured (10352 bits) on interface en0, id 0
 > Ethernet II, Src: ASUSTekC_5a:2b:b8 (40:16:7e:5a:2b:b8), Dst: Apple_c2:67:53 (90:9c:4a:c2:67:53)
 > Internet Protocol Version 4, Src: 128.119.240.84, Dst: 192.168.1.245
 > Transmission Control Protocol, Src Port: 443, Dst Port: 51146, Seq: 4097, Ack: 518, Len: 1228
 > [4 Reassembled TCP Segments (4907 bytes): #32(1378), #33(1448), #34(1200), #37(881)]
 > Transport Layer Security
 > Transport Layer Security
 > TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
 > TLSv1.2 Record Layer: Handshake Protocol: **Server Hello Done**

- **The TLS Handshake: wrapping up the handshake**

No.	Time	Source	Destination	Protocol	Length	Info
32	14:29:29.558	128.119.240.84	192.168.1.245	TLSv1.2	1514	Server Hello
37	14:29:29.559	128.119.240.84	192.168.1.245	TLSv1.2	1294	Certificate, Server Key Exchange, Server Hello Done
39	14:29:29.571	192.168.1.245	128.119.240.84	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
40	14:29:29.652	128.119.240.84	192.168.1.245	TLSv1.2	340	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
41	14:29:29.653	192.168.1.245	128.119.240.84	TLSv1.2	970	Application Data

> Flags: 0x018 (PSH, ACK)
 Window: 235
 [Calculated window size: 30080]
 [Window size scaling factor: 128]
 Checksum: 0x0a7b [unverified]
 [Checksum Status: Unverified]
 Urgent Pointer: 0
 > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 > [Timestamps]
 > [SEQ/ACK analysis]
 TCP payload (274 bytes)
 > Transport Layer Security
 > TLSv1.2 Record Layer: Handshake Protocol: New Session Ticket
 Content Type: Handshake (22)
 Version: TLS 1.2 (0x0303)
 Length: 218
 > Handshake Protocol: New Session Ticket
 Handshake Type: New Session Ticket (4)
 Length: 214
 > TLS Session Ticket
 Session Ticket Lifetime Hint: 7200 seconds (2 hours)
 Session Ticket Length: 208
 Session Ticket: 4db49c9da1802e820e32392d8149f879446dcce9922b07240a22dac8780129233635807b...
 > 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: 40
 ..

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?

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 is: 39

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?

The packet number in your trace containing your client's certificate is: 40

- Application data

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

TLS protocol

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

The symmetric key cryptography algorithm is decided and declared in the "ClientKeyExchange" message. This message contains the client's pre-master secret, which is used to generate the symmetric key for encryption and decryption of data in the TLS session.

No.	Time	Source	Destination	Protocol	Length	Info
32	14:29:29.558	128.119.240.84	192.168.1.245	TLSv1.2	1514	Server Hello
33	14:29:29.558	128.119.240.84	192.168.1.245	TCP	1514	443 → 51146 [ACK] Seq=1449 Ack=518 Win=30080 Len=1448 TSval=248562518 TSecr=465227082 [TCP Reset]
34	14:29:29.558	128.119.240.84	192.168.1.245	TCP	1266	443 → 51146 [PSH, ACK] Seq=2897 Ack=518 Win=30080 Len=1200 TSval=248562518 TSecr=465227082 [ACK]
35	14:29:29.558	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=2897 Win=129600 Len=0 TSval=465227160 TSecr=248562518
36	14:29:29.558	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=4097 Win=128384 Len=0 TSval=465227161 TSecr=248562518
37	14:29:29.559	128.119.240.84	192.168.1.245	TLSv1.2	1294	Certificate, Server Key Exchange, Server Hello Done
38	14:29:29.559	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=5325 Win=129792 Len=0 TSval=465227161 TSecr=248562520
39	14:29:29.571	192.168.1.245	128.119.240.84	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
40	14:29:29.652	128.119.240.84	192.168.1.245	TLSv1.2	340	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
41	14:29:29.653	192.168.1.245	128.119.240.84	TLSv1.2	970	Application Data
42	14:29:29.740	128.119.240.84	192.168.1.245	TLSv1.2	1514	Application Data, Application Data
43	14:29:29.741	128.119.240.84	192.168.1.245	TCP	1514	443 → 51146 [ACK] Seq=7047 Ack=1548 Win=31872 Len=1448 TSval=248562700 TSecr=465227253 [ACK]
44	14:29:29.741	128.119.240.84	192.168.1.245	TCP	1514	443 → 51146 [ACK] Seq=8495 Ack=1548 Win=31872 Len=1448 TSval=248562700 TSecr=465227253 [ACK]
45	14:29:29.741	128.119.240.84	192.168.1.245	TCP	1514	443 → 51146 [ACK] Seq=9943 Ack=1548 Win=31872 Len=1448 TSval=248562700 TSecr=465227253 [ACK]

> Frame 39: 192 bytes on wire (1536 bits), 192 bytes captured (1536 bits) on interface en0, id 0
 > Ethernet II, Src: Apple_c2:67:53 (90:9c:4a:c2:67:53), Dst: ASUSTekC_5a:2b:b8 (40:16:7e:5a:2b:b8)
 > Internet Protocol Version 4, Src: 192.168.1.245, Dst: 128.119.240.84
 > Transmission Control Protocol, Src Port: 51146, Dst Port: 443, Seq: 518, Ack: 5325, Len: 126
 > Transport Layer Security
 > TLSv1.2 Record Layer: Handshake Protocol: Client Key Exchange
 Content Type: Handshake (22)
 Version: TLS 1.2 (0x0303)
 Length: 70
 > Handshake Protocol: Client Key Exchange
 > 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: 40
 Handshake Protocol: Encrypted Handshake Message

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

The packet number in your trace for the first encrypted message carrying application data from client to server is 41

No.	Time	Source	Destination	Protocol	Length	Info
34	14:29:29.558	128.119.240.84	192.168.1.245	TCP	1266	443 → 51146 [PSH, ACK] Seq=2897 Ack=518 Win=30080 Len=1200 TSval=248562518 TSecr=465227082
35	14:29:29.558	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=2897 Win=129600 Len=0 TSval=465227160 TSecr=248562518
36	14:29:29.558	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=4097 Win=128384 Len=0 TSval=465227161 TSecr=248562518
37	14:29:29.559	128.119.240.84	192.168.1.245	TLSv1.2	1294	Certificate, Server Key Exchange, Server Hello Done
38	14:29:29.559	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=5325 Win=129792 Len=0 TSval=465227161 TSecr=248562520
39	14:29:29.571	192.168.1.245	128.119.240.84	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
40	14:29:29.652	128.119.240.84	192.168.1.245	TLSv1.2	340	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
41	14:29:29.653	192.168.1.245	128.119.240.84	TLSv1.2	970	Application Data
42	14:29:29.740	128.119.240.84	192.168.1.245	TLSv1.2	1514	Application Data, Application Data
43	14:29:29.741	128.119.240.84	192.168.1.245	TCP	1514	443 → 51146 [ACK] Seq=7047 Ack=1548 Win=31872 Len=1448 TSval=248562700 TSecr=465227253 [TC
44	14:29:29.741	128.119.240.84	192.168.1.245	TCP	1514	443 → 51146 [ACK] Seq=8495 Ack=1548 Win=31872 Len=1448 TSval=248562700 TSecr=465227253 [TC
45	14:29:29.741	128.119.240.84	192.168.1.245	TCP	1514	443 → 51146 [ACK] Seq=9943 Ack=1548 Win=31872 Len=1448 TSval=248562700 TSecr=465227253 [TC
46	14:29:29.741	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=1548 Ack=8495 Win=129600 Len=0 TSval=465227340 TSecr=248562700
47	14:29:29.741	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=1548 Ack=11391 Win=126720 Len=0 TSval=465227340 TSecr=248562700

> Frame 41: 970 bytes on wire (7760 bits), 970 bytes captured (7760 bits) on interface en0, id 0
 > Ethernet II, Src: Apple_C2:67:53 (90:9c:4a:c2:67:53), Dst: ASUSTeK_5a:2b:b8 (40:16:7e:5a:2b:b8)
 > Internet Protocol Version 4, Src: 192.168.1.245, Dst: 128.119.240.84
 > Transmission Control Protocol, Src Port: 51146, Dst Port: 443, Seq: 644, Ack: 5599, Len: 904
 > Transport Layer Security
 > TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
 Content Type: Application Data (23)
 Version: TLS 1.2 (0x0303)
 Length: 899
 Encrypted Application Data: 00000000000000104a00e834c9f8aa2487eacacaca6fe67a86a3783533667b228bd9cd2...
 [Application Data Protocol: Hypertext Transfer Protocol]

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?

No.	Time	Source	Destination	Protocol	Length	Info
34	14:29:29.558	128.119.240.84	192.168.1.245	TCP	1266	443 → 51146 [PSH, ACK] Seq=2897 Ack=518 Win=30080 Len=1200 TSval=248562518 TSecr=465227082
35	14:29:29.558	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=2897 Win=129600 Len=0 TSval=465227160 TSecr=248562518
36	14:29:29.558	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=4097 Win=128384 Len=0 TSval=465227161 TSecr=248562518
37	14:29:29.559	128.119.240.84	192.168.1.245	TLSv1.2	1294	Certificate, Server Key Exchange, Server Hello Done
38	14:29:29.559	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=518 Ack=5325 Win=129792 Len=0 TSval=465227161 TSecr=248562520
39	14:29:29.571	192.168.1.245	128.119.240.84	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
40	14:29:29.652	128.119.240.84	192.168.1.245	TLSv1.2	340	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
41	14:29:29.653	192.168.1.245	128.119.240.84	TLSv1.2	970	Application Data
42	14:29:29.740	128.119.240.84	192.168.1.245	TLSv1.2	1514	Application Data, Application Data
43	14:29:29.741	128.119.240.84	192.168.1.245	TCP	1514	443 → 51146 [ACK] Seq=7047 Ack=1548 Win=31872 Len=1448 TSval=248562700 TSecr=465227253 [TC
44	14:29:29.741	128.119.240.84	192.168.1.245	TCP	1514	443 → 51146 [ACK] Seq=8495 Ack=1548 Win=31872 Len=1448 TSval=248562700 TSecr=465227253 [TC
45	14:29:29.741	128.119.240.84	192.168.1.245	TCP	1514	443 → 51146 [ACK] Seq=9943 Ack=1548 Win=31872 Len=1448 TSval=248562700 TSecr=465227253 [TC
46	14:29:29.741	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=1548 Ack=8495 Win=129600 Len=0 TSval=465227340 TSecr=248562700
47	14:29:29.741	192.168.1.245	128.119.240.84	TCP	66	51146 → 443 [ACK] Seq=1548 Ack=11391 Win=126720 Len=0 TSval=465227340 TSecr=248562700

> Frame 41: 970 bytes on wire (7760 bits), 970 bytes captured (7760 bits) on interface en0, id 0
 > Ethernet II, Src: Apple_C2:67:53 (90:9c:4a:c2:67:53), Dst: ASUSTeK_5a:2b:b8 (40:16:7e:5a:2b:b8)
 > Internet Protocol Version 4, Src: 192.168.1.245, Dst: 128.119.240.84
 > Transmission Control Protocol, Src Port: 51146, Dst Port: 443, Seq: 644, Ack: 5599, Len: 904
 > Transport Layer Security
 > TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
 Content Type: Application Data (23)
 Version: TLS 1.2 (0x0303)
 Length: 899
 Encrypted Application Data: 00000000000000104a00e834c9f8aa2487eacacaca6fe67a86a3783533667b228bd9cd2...
 [Application Data Protocol: Hypertext Transfer Protocol]

Encrypted Application Data:
 0000000000000000104a00e834c9f8aa2487eacacaca6fe67a86a3783533667b228bd9cd2...

24. What packet number contains the client-to-server TLS message that shuts down the TLS connection? Because TLS messages are encrypted in our Wireshark traces, we can't actually look *inside* a TLS message and so we'll have to make an educated guess here.

Packet number contains the client-to-server TLS message that shuts down the TLS connection: 41