

Wireshark Lab:

TLS v8.1

In this lab, we'll investigate Transport Layer Security (known as TLS) and aspects of the authentication, data integrity, and confidentiality services provided by TLS. TLS is the successor to the now-deprecated Secure Sockets Layer (SSL). So, it'd be preferable to do this v8.1 TLS Wireshark lab, rather than the v8.0 SSL Wireshark Lab.

We'll investigate TLS by analyzing a Wireshark packet trace captured during the retrieval of a web page via HTTPS - a secure version of HTTP, which implements TLS on top of HTTP. We'll look at TLS's client-server handshaking protocol in some detail, since that's where most of the interesting action happens.

1. Capturing packets in an TLS session

The first step in this lab is to capture the packets in an TLS session. To do this, you should startup Wireshark and begin packet capture, retrieve the homepage from <https://www.cics.umass.edu> using the browser of your choice, and then stop Wireshark packet capture. The 's' after 'http' will cause the Hypertext Transfer Protocol Secure (HTTPS) - an extension of HTTP - to be used to securely retrieve the homepage from www.cics.umass.edu. Here, "securely" means that the www.cics.umass.edu server will be authenticated by your web browser, that the transmission of your client HTTP GET request and the server's reply will be encrypted, and the integrity of all message content will be cryptographically verified. Of course, the authentication, integrity and encryption of a computer science department's webpage may not be as critical as that for Internet commerce and banking sites, but the same TLS protocol and TLS messages are used in all cases.

2. A first look at the captured trace

Let's first set Wireshark's display so that only the packets to and from www.cics.umass.edu, whose IP address is 128.119.240.84, are displayed. To do this, enter

```
ip.addr == 128.119.240.84
```

in Wireshark's display filter window. Your screen should look similar to what is shown in Figure 1.

Figure 1: Wireshark display showing TCP and TLS message to/from 128.119.240.84

It's important to keep in mind that an Ethernet frame (containing an IP datagram containing an TCP segment) may contain one or more TLS records. (This is very different from HTTP, for which each frame contains either one complete HTTP message or a portion of a HTTP message.) Also, a TLS record may not completely fit into an Ethernet frame, in which case multiple frames will be needed to carry the record.

We've said earlier that HTTPS implements TLS running "over" TCP. That means that a TCP connection must first be established between your browser and the web server for www.cics.umass.edu before TLS and HTTP messages can be exchanged, just as we saw

with the vanilla (non-TLS) HTTP protocol.

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).

Packet Number 853:

The image shows a Wireshark packet capture window titled "Wi-Fi". The filter bar at the top shows "ip.addr == 34.227.156.202". The packet list on the left shows a series of packets. Packet 853 is highlighted in blue. The packet details pane on the right shows the structure of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Text item	Info
853	7.829748	192.168.1.207	34.227.156.202	TCP	66	✓	56609 → 443 [SYN] Seq=0 Win=64240 Len=0
854	7.921549	34.227.156.202	192.168.1.207	TCP	66	✓	443 → 56609 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
855	7.921603	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131 Len=0
856	7.922094	192.168.1.207	34.227.156.202	TCP	1486	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131 Len=0
857	7.922094	192.168.1.207	34.227.156.202	TLSv1.2	447	✓	Client Hello (SNI=www.cics.umass.edu. Len=447)
858	8.010550	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=1 Ack=1433 Win=64240 Len=0
859	8.010550	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=1 Ack=1826 Win=64240 Len=0
860	8.010550	34.227.156.202	192.168.1.207	TLSv1.2	2918	✓	Server Hello (Len=2918)
861	8.010597	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1826 Ack=2865 Win=64240 Len=0
862	8.010781	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=2865 Ack=1826 Win=64240 Len=0
863	8.010781	34.227.156.202	192.168.1.207	TLSv1.2	1000	✓	Certificate, Server Key Exchange, Server Hello Done (Len=1000)
864	8.010817	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1826 Ack=5243 Win=64240 Len=0
865	8.012261	192.168.1.207	34.227.156.202	TLSv1.2	180	✓	Client Key Exchange, Change Cipher Spec, Application Data (Len=180)
866	8.012387	192.168.1.207	34.227.156.202	TLSv1.2	153	✓	Application Data (Len=153)
867	8.012481	192.168.1.207	34.227.156.202	TLSv1.2	550	✓	Application Data (Len=550)
872	8.096897	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=5243 Ack=2547 Win=64240 Len=0
873	8.096897	34.227.156.202	192.168.1.207	TLSv1.2	312	✓	New Session Ticket, Change Cipher Spec, Application Data (Len=312)
874	8.096897	34.227.156.202	192.168.1.207	TLSv1.2	132	✓	Application Data (Len=132)
875	8.096897	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=5579 Ack=2547 Win=64240 Len=0
876	8.096934	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2547 Ack=7011 Win=64240 Len=0
877	8.096983	34.227.156.202	192.168.1.207	TCP	5782	✓	443 → 56609 [ACK] Seq=7011 Ack=2547 Win=64240 Len=0
878	8.096994	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2547 Ack=12739 Win=64240 Len=0
879	8.097100	34.227.156.202	192.168.1.207	TCP	4350	✓	443 → 56609 [ACK] Seq=12739 Ack=2547 Win=64240 Len=0

Packet 853 details:

- Total Length: 52
- Identification: 0x5577 (21879)
- 010. = Flags: 0x2, Don't fragment
 - 0... = Reserved bit: Not set
 - .1... = Don't fragment: Set
 - ..0. = More fragments: Not set
 - ...0 0000 0000 0000 = Fragment Offset: 0
- Time to Live: 128
- Protocol: TCP (6)
- Header Checksum: 0x2328 [validation disabled]
[Header checksum status: Unverified]
- Source Address: 192.168.1.207
- Destination Address: 34.227.156.202
- [Stream index: 12]
- Transmission Control Protocol, Src Port: 56609, Dst Port: 443, Seq: 0, Len: 0
 - Source Port: 56609
 - Destination Port: 443
 - [Stream index: 14]

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

The TCP connection is set up before the first TLS message is sent

Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr == 34.227.156.202

No.	Time	Source	Destination	Protocol	Length	Text item	Info
853	7.829748	192.168.1.207	34.227.156.202	TCP	66	✓	56609 → 443 [SYN] Seq=0 Win=64240 Len=0
854	7.921549	34.227.156.202	192.168.1.207	TCP	66	✓	443 → 56609 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
855	7.921603	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131 Len=0
856	7.922094	192.168.1.207	34.227.156.202	TCP	1486	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131 Len=0
857	7.922094	192.168.1.207	34.227.156.202	TLSv1.2	447	✓	Client Hello (SNI=www.cics.umass.edu)
858	8.010550	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=1 Ack=1433 Win=0 Len=0
859	8.010550	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=1 Ack=1826 Win=0 Len=0
860	8.010550	34.227.156.202	192.168.1.207	TLSv1.2	2918	✓	Server Hello
861	8.010597	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1826 Ack=2865 Win=0 Len=0
862	8.010781	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=2865 Ack=1826 Win=0 Len=0
863	8.010781	34.227.156.202	192.168.1.207	TLSv1.2	1000	✓	Certificate, Server Key Exchange, Server Certificate
864	8.010817	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1826 Ack=5243 Win=0 Len=0
865	8.012261	192.168.1.207	34.227.156.202	TLSv1.2	180	✓	Client Key Exchange, Change Cipher Spec
866	8.012387	192.168.1.207	34.227.156.202	TLSv1.2	153	✓	Application Data
867	8.012481	192.168.1.207	34.227.156.202	TLSv1.2	550	✓	Application Data
872	8.096897	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=5243 Ack=2547 Win=0 Len=0
873	8.096897	34.227.156.202	192.168.1.207	TLSv1.2	312	✓	New Session Ticket, Change Cipher Spec
874	8.096897	34.227.156.202	192.168.1.207	TLSv1.2	132	✓	Application Data
875	8.096897	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=5579 Ack=2547 Win=0 Len=0
876	8.096934	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2547 Ack=7011 Win=0 Len=0
877	8.096983	34.227.156.202	192.168.1.207	TCP	5782	✓	443 → 56609 [ACK] Seq=7011 Ack=2547 Win=0 Len=0
878	8.096994	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2547 Ack=12739 Win=0 Len=0
879	8.097100	34.227.156.202	192.168.1.207	TCP	4350	✓	443 → 56609 [ACK] Seq=12739 Ack=2547 Win=0 Len=0
880	8.097131	192.168.1.207	34.227.156.202	TLSv1.2	0	✓	Application Data

Total Length: 52
Identification: 0x5577 (21879)

- 010. = Flags: 0x2, Don't fragment
 - 0... = Reserved bit: Not set
 - .1.. = Don't fragment: Set
 - ..0. = More fragments: Not set
- ...0 0000 0000 0000 = Fragment Offset: 0
- Time to Live: 128
- Protocol: TCP (6)
- Header Checksum: 0x2328 [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 192.168.1.207
- Destination Address: 34.227.156.202
- [Stream index: 12]

Transmission Control Protocol, Src Port: 56609, Dst Port: 443, Seq: 0, Len: 0

- Source Port: 56609
- Destination Port: 443
- [Stream index: 14]

3. The TLS Handshake: Client Hello message

We learned that, as shown in Figure 2, the client-server TLS handshake begins with the client sending a TLS Client Hello message.

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

Packet Number 857

The screenshot shows a Wireshark packet capture of a TLS handshake. The packet list on the left shows packet 857, which is a TLSv1.2 Client Hello message. The packet details pane on the right shows the structure of the Client Hello message, including the TLS version (1.0), cipher suites, and the random value.

No.	Time	Source	Destination	Protocol	Length	Text	Info
853	7.829748	192.168.1.207	34.227.156.202	TCP	66	✓	56609 → 443 [SYN] Seq=0 Win=64240 Len=0
854	7.921549	34.227.156.202	192.168.1.207	TCP	66	✓	443 → 56609 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
855	7.921603	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131 Len=0
856	7.922094	192.168.1.207	34.227.156.202	TCP	1486	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131 Len=0
857	7.922094	192.168.1.207	34.227.156.202	TLSv1.2	447	✓	Client Hello (SNI=www.cics.umass.edu.)

Packet 857 details:

- Total Length: 52
- Identification: 0x5577 (21879)
- 010. = Flags: 0x2, Don't fragment
- 0... = Reserved bit: Not set
- 1... = Don't fragment: Set
- ..0. = More fragments: Not set
- ...0 0000 0000 0000 = Fragment Offset: 0
- Time to Live: 128
- Protocol: TCP (6)
- Header Checksum: 0x2328 [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 192.168.1.207
- Destination Address: 34.227.156.202
- [Stream index: 12]
- Transmission Control Protocol, Src Port: 56609, Dst Port: 443, Seq: 0, Len: 0
- Source Port: 56609
- Destination Port: 443
- [Stream index: 14]

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

Version: TLS 1.0 (0x0301)

The screenshot shows the details of the TLS Client Hello message (packet 857). The version is highlighted as TLS 1.0 (0x0301).

Frame 857: 447 bytes on wire (3576 bits), 447 bytes captured (3576 bits) on interface \Device\NPF_{FE59B5CF-7C5A-49A0-8CB4-84451B1E...}

Ethernet II, Src: AzureWaveTec_cd:1d:a9 (10:68:38:cd:1d:a9), Dst: SagemcomBroa_4d:b1:21 (4c:19:5d:4d:b1:21)

Internet Protocol Version 4, Src: 192.168.1.207, Dst: 34.227.156.202

Transmission Control Protocol, Src Port: 56609, Dst Port: 443, Seq: 1433, Ack: 1, Len: 393

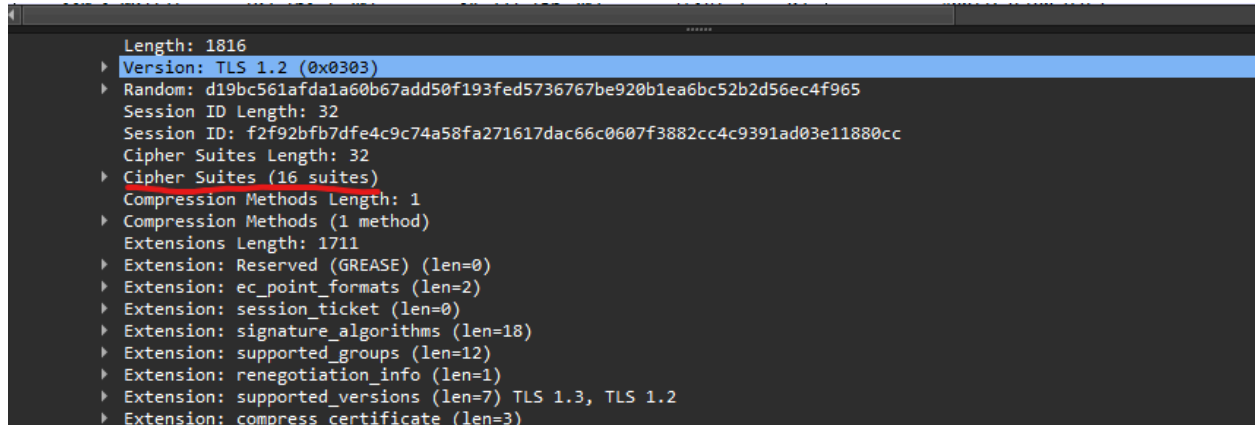
[2 Reassembled TCP Segments (1825 bytes): #856(1432), #857(393)]

Transport Layer Security

- TLSv1.2 Record Layer: Handshake Protocol: Client Hello
- Content Type: Handshake (22)
- Version: TLS 1.0 (0x0301)
- Length: 1820
- Handshake Protocol: Client Hello

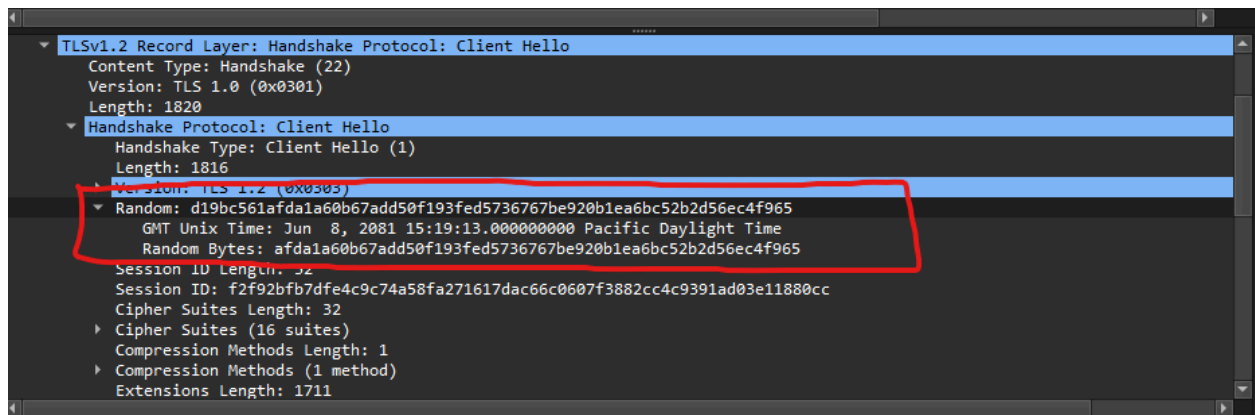
5. How many cipher suites are supported by your client, as declared in the Client Hello message? A cipher suite is a set of related cryptographic algorithms that determine how session keys will be derived, and how data will be encrypted and be digitally signed via a HMAC algorithm.

16 Cipher Suites



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 of the Random Bytes: af



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).

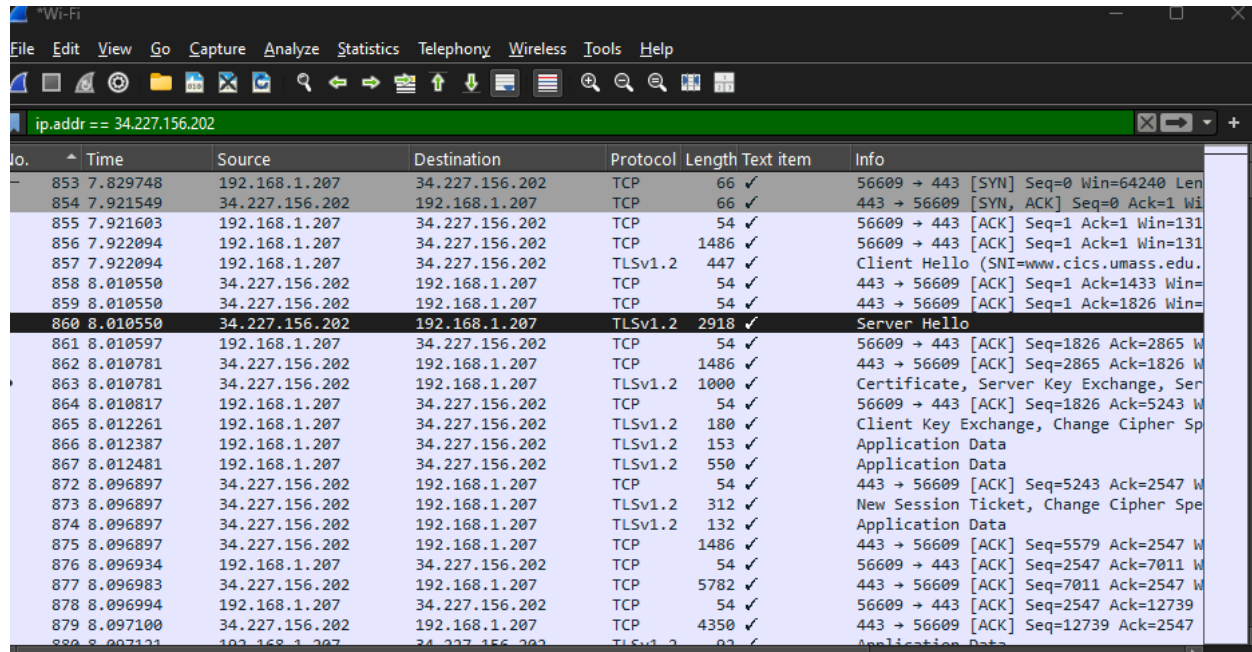
The random bytes field plays a foundational role in TLS security by ensuring uniqueness, contributing to secure key generation, and safeguarding against replay attacks. Without this field, the handshake process would be vulnerable to several cryptographic and operational weaknesses.

3. The TLS Handshake: Server Hello message

Next, let's take a look at the second step of the TLS handshake, the TLS Server Hello message, which is sent in response to the earlier TLS Client Hello message.

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

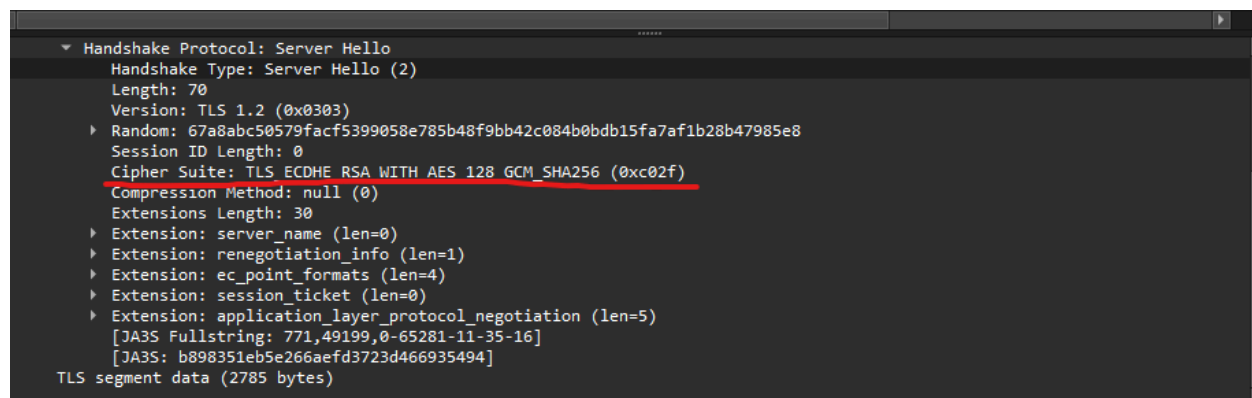
Packet Number 860



No.	Time	Source	Destination	Protocol	Length	Text item	Info
853	7.829748	192.168.1.207	34.227.156.202	TCP	66	✓	56609 → 443 [SYN] Seq=0 Win=64240 Len=0
854	7.921549	34.227.156.202	192.168.1.207	TCP	66	✓	443 → 56609 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
855	7.921603	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131 Len=0
856	7.922094	192.168.1.207	34.227.156.202	TCP	1486	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131 Len=0
857	7.922094	192.168.1.207	34.227.156.202	TLSv1.2	447	✓	Client Hello (SNI=www.cics.umass.edu)
858	8.010550	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=1 Ack=1433 Win=0 Len=0
859	8.010550	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=1 Ack=1826 Win=0 Len=0
860	8.010550	34.227.156.202	192.168.1.207	TLSv1.2	2918	✓	Server Hello
861	8.010597	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1826 Ack=2865 Win=0 Len=0
862	8.010781	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=2865 Ack=1826 Win=0 Len=0
863	8.010781	34.227.156.202	192.168.1.207	TLSv1.2	1000	✓	Certificate, Server Key Exchange, Server Hello Done
864	8.010817	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1826 Ack=5243 Win=0 Len=0
865	8.012261	192.168.1.207	34.227.156.202	TLSv1.2	180	✓	Client Key Exchange, Change Cipher Spec
866	8.012387	192.168.1.207	34.227.156.202	TLSv1.2	153	✓	Application Data
867	8.012481	192.168.1.207	34.227.156.202	TLSv1.2	550	✓	Application Data
872	8.096897	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=5243 Ack=2547 Win=0 Len=0
873	8.096897	34.227.156.202	192.168.1.207	TLSv1.2	312	✓	New Session Ticket, Change Cipher Spec
874	8.096897	34.227.156.202	192.168.1.207	TLSv1.2	132	✓	Application Data
875	8.096897	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=5579 Ack=2547 Win=0 Len=0
876	8.096934	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2547 Ack=7011 Win=0 Len=0
877	8.096983	34.227.156.202	192.168.1.207	TCP	5782	✓	443 → 56609 [ACK] Seq=7011 Ack=2547 Win=0 Len=0
878	8.096994	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2547 Ack=12739 Win=0 Len=0
879	8.097100	34.227.156.202	192.168.1.207	TCP	4350	✓	443 → 56609 [ACK] Seq=12739 Ack=2547 Win=0 Len=0
880	8.097131	192.168.1.207	34.227.156.202	TLSv1.2	02	✓	Application Data

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

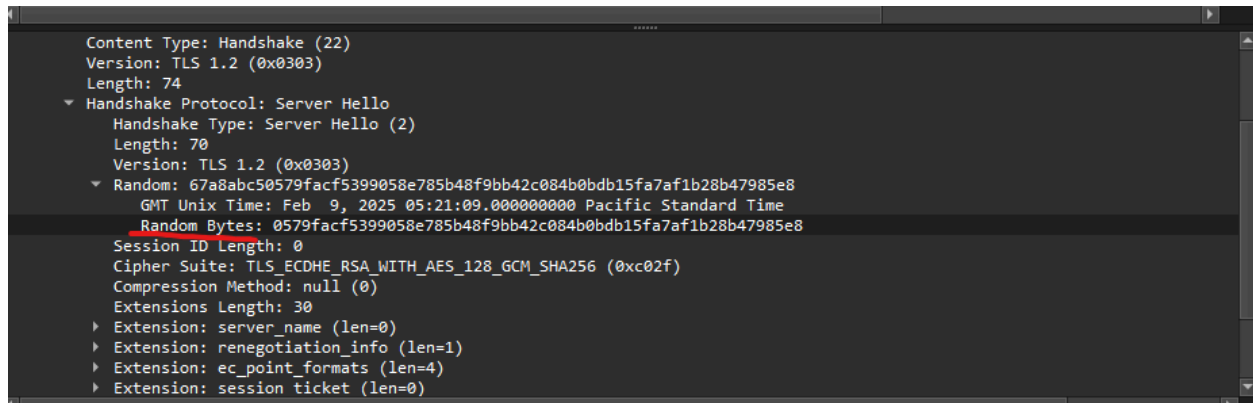
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)



Handshake Protocol: Server Hello
Handshake Type: Server Hello (2)
Length: 70
Version: TLS 1.2 (0x0303)
Random: 67a8abc50579facf5399058e785b48f9bb42c084b0bdb15fa7af1b28b47985e8
Session ID Length: 0
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
Compression Method: null (0)
Extensions Length: 30
Extension: server_name (len=0)
Extension: renegotiation_info (len=1)
Extension: ec_point_formats (len=4)
Extension: session_ticket (len=0)
Extension: application_layer_protocol_negotiation (len=5)
[JA3S Fullstring: 771,49199,0-65281-11-35-16]
[JA3S: b898351eb5e266aefd3723d466935494]
TLS segment data (2785 bytes)

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)?

Yes, the Server Hello message does contain random bytes. It provides for robust cryptographic key generation and securing the communication channel.



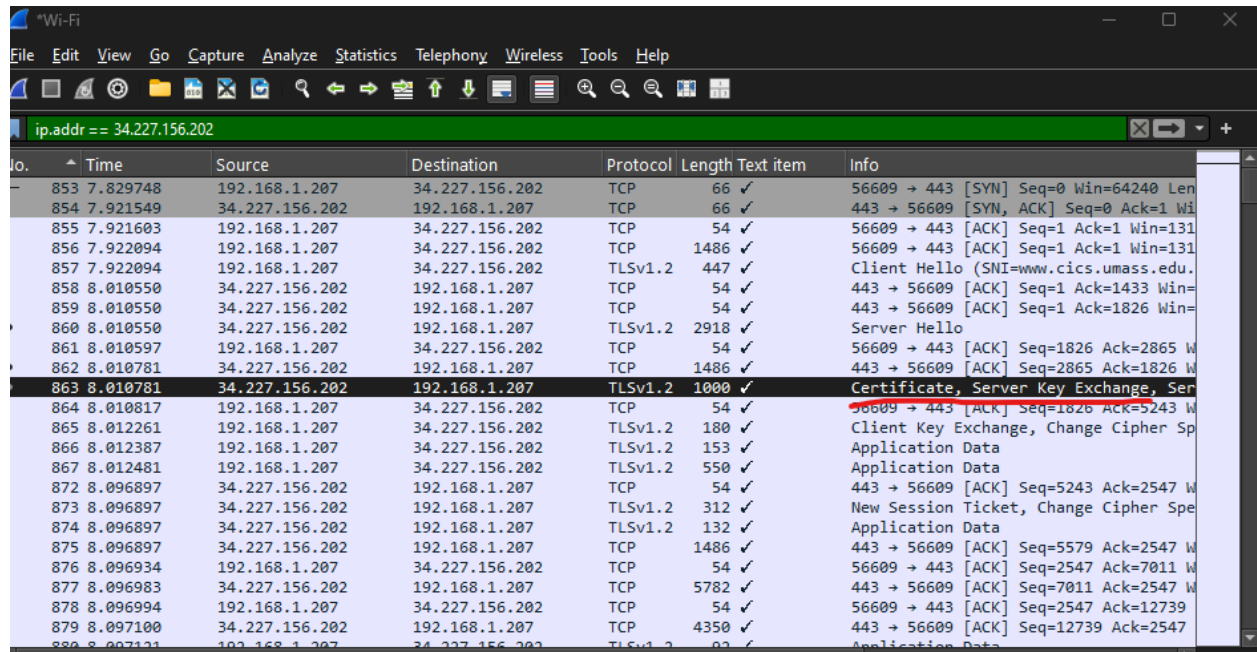
In response to the initial Client Hello message, the server also sends back two important additional pieces of information (that may be contained in a different packet than the initial part of the Server Hello message). The first important piece of information returned is a certificate issued by a trusted certification authority (CA; see section 8.3 in the text) that binds the public key (and other information) of the web server to that web server's identity. Your client may use the server's public key for a number of different purposes, including deriving the symmetric keys to encrypt data being sent over this HTTPS/TLS/TCP session and for generating HMAC digital signatures. You can, of course, look at the server's certificate in Wireshark. You can also view the server's certificate in your web browser after the www.cs.umass.edu server has responded to your request (and the certificate formatted a lot prettier too) – here's how.

The second piece of additional information returned from the server are parameters needed for the symmetric encryption of the data (HTTP messages in this case) being exchanged.

Let's dig a bit deeper into the 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)?

Packet Number 863



No.	Time	Source	Destination	Protocol	Length	Text item	Info
853	7.829748	192.168.1.207	34.227.156.202	TCP	66	✓	56609 → 443 [SYN] Seq=0 Win=64240 Len=0
854	7.921549	34.227.156.202	192.168.1.207	TCP	66	✓	443 → 56609 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
855	7.921603	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131 Len=0
856	7.922094	192.168.1.207	34.227.156.202	TCP	1486	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131 Len=0
857	7.922094	192.168.1.207	34.227.156.202	TLSv1.2	447	✓	Client Hello (SNI=www.cics.umass.edu.)
858	8.010550	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=1 Ack=1433 Win=0 Len=0
859	8.010550	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=1 Ack=1826 Win=0 Len=0
860	8.010550	34.227.156.202	192.168.1.207	TLSv1.2	2918	✓	Server Hello
861	8.010597	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1826 Ack=2865 Win=0 Len=0
862	8.010781	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=2865 Ack=1826 Win=0 Len=0
863	8.010781	34.227.156.202	192.168.1.207	TLSv1.2	1000	✓	Certificate, Server Key Exchange, Server Hello
864	8.010817	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1826 Ack=5243 Win=0 Len=0
865	8.012261	192.168.1.207	34.227.156.202	TLSv1.2	180	✓	Client Key Exchange, Change Cipher Spec
866	8.012387	192.168.1.207	34.227.156.202	TLSv1.2	153	✓	Application Data
867	8.012481	192.168.1.207	34.227.156.202	TLSv1.2	550	✓	Application Data
872	8.096897	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=5243 Ack=2547 Win=0 Len=0
873	8.096897	34.227.156.202	192.168.1.207	TLSv1.2	312	✓	New Session Ticket, Change Cipher Spec
874	8.096897	34.227.156.202	192.168.1.207	TLSv1.2	132	✓	Application Data
875	8.096897	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=5579 Ack=2547 Win=0 Len=0
876	8.096934	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2547 Ack=7011 Win=0 Len=0
877	8.096983	34.227.156.202	192.168.1.207	TCP	5782	✓	443 → 56609 [ACK] Seq=7011 Ack=2547 Win=0 Len=0
878	8.096994	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2547 Ack=12739 Win=0 Len=0
879	8.097100	34.227.156.202	192.168.1.207	TCP	4350	✓	443 → 56609 [ACK] Seq=12739 Ack=2547 Win=0 Len=0
880	8.097131	192.168.1.207	34.227.156.202	TLSv1.2	0	✓	Application Data

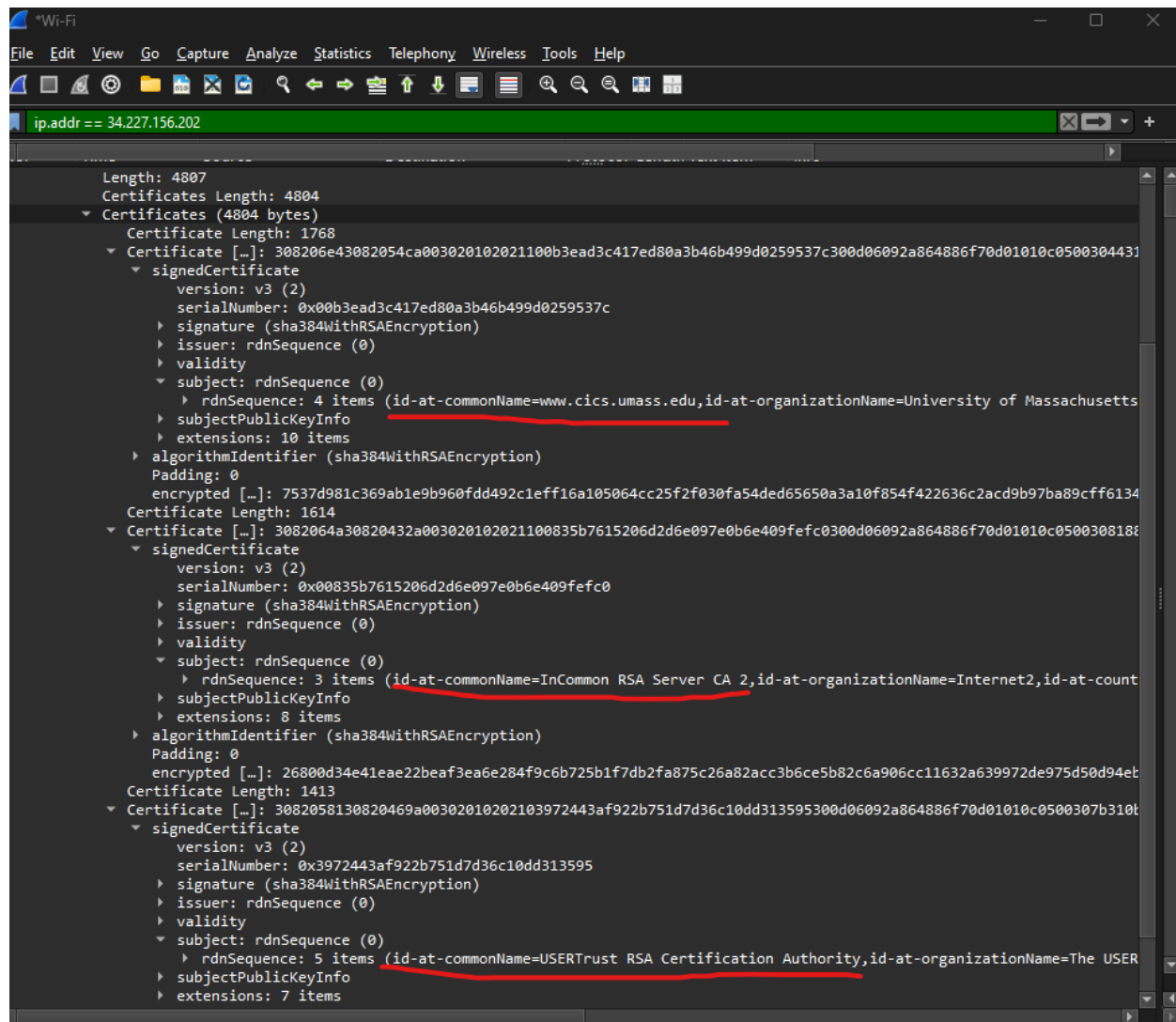
.12. A server may return more than one certificate. If more than one certificate is returned, are all of these certificates for www.cs.umass.edu? If not all are for www.cs.umass.edu, then who are these other certificates for? You can determine who the certificate is for by checking the id-at-commonName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

Second Certificate:

- Issuer: InCommon RSA Server CA 2

Third Certificate:

- Issuer: USERTrust RSA Certification Authority



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

- InCommon RSA Server CA 2

```

Certificates (4804 bytes)
Certificate Length: 1768
Certificate [...]: 308206e43082054ca003020102021100b3ead3c417ed80a3b46b499d0259537c300d06092a864886f70d01010c05003044310b3009
  signedCertificate
    version: v3 (2)
    serialNumber: 0x00b3ead3c417ed80a3b46b499d0259537c
    signature (sha384WithRSAEncryption)
    issuer: rdnSequence (0)
      rdnSequence: 3 items (id-at-commonName=InCommon RSA Server CA 2,id-at-organizationName=Internet2,id-at-countryName=US)
        RDNSequence item: 1 item (id-at-countryName=US)
          RelativeDistinguishedName item (id-at-countryName=US)
            Object Id: 2.5.4.6 (id-at-countryName)
            CountryName: US
          RDNSequence item: 1 item (id-at-organizationName=Internet2)
            RelativeDistinguishedName item (id-at-organizationName=Internet2)
          RDNSequence item: 1 item (id-at-commonName=InCommon RSA Server CA 2)
            RelativeDistinguishedName item (id-at-commonName=InCommon RSA Server CA 2)
        validity
      subject: rdnSequence (0)
        rdnSequence: 4 items (id-at-commonName=www.cics.umass.edu,id-at-organizationName=University of Massachusetts Amherst)
          RDNSequence item: 1 item (id-at-countryName=US)
            RelativeDistinguishedName item (id-at-countryName=US)
              Object Id: 2.5.4.6 (id-at-countryName)
              CountryName: US
            RDNSequence item: 1 item (id-at-stateOrProvinceName=Massachusetts)
              RelativeDistinguishedName item (id-at-stateOrProvinceName=Massachusetts)
                Object Id: 2.5.4.8 (id-at-stateOrProvinceName)
                DirectoryString: printableString (1)
            RDNSequence item: 1 item (id-at-organizationName=University of Massachusetts Amherst)
              RelativeDistinguishedName item (id-at-organizationName=University of Massachusetts Amherst)
                Object Id: 2.5.4.10 (id-at-organizationName)

```

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.
signature (sha384WithRSAEncryption)

```

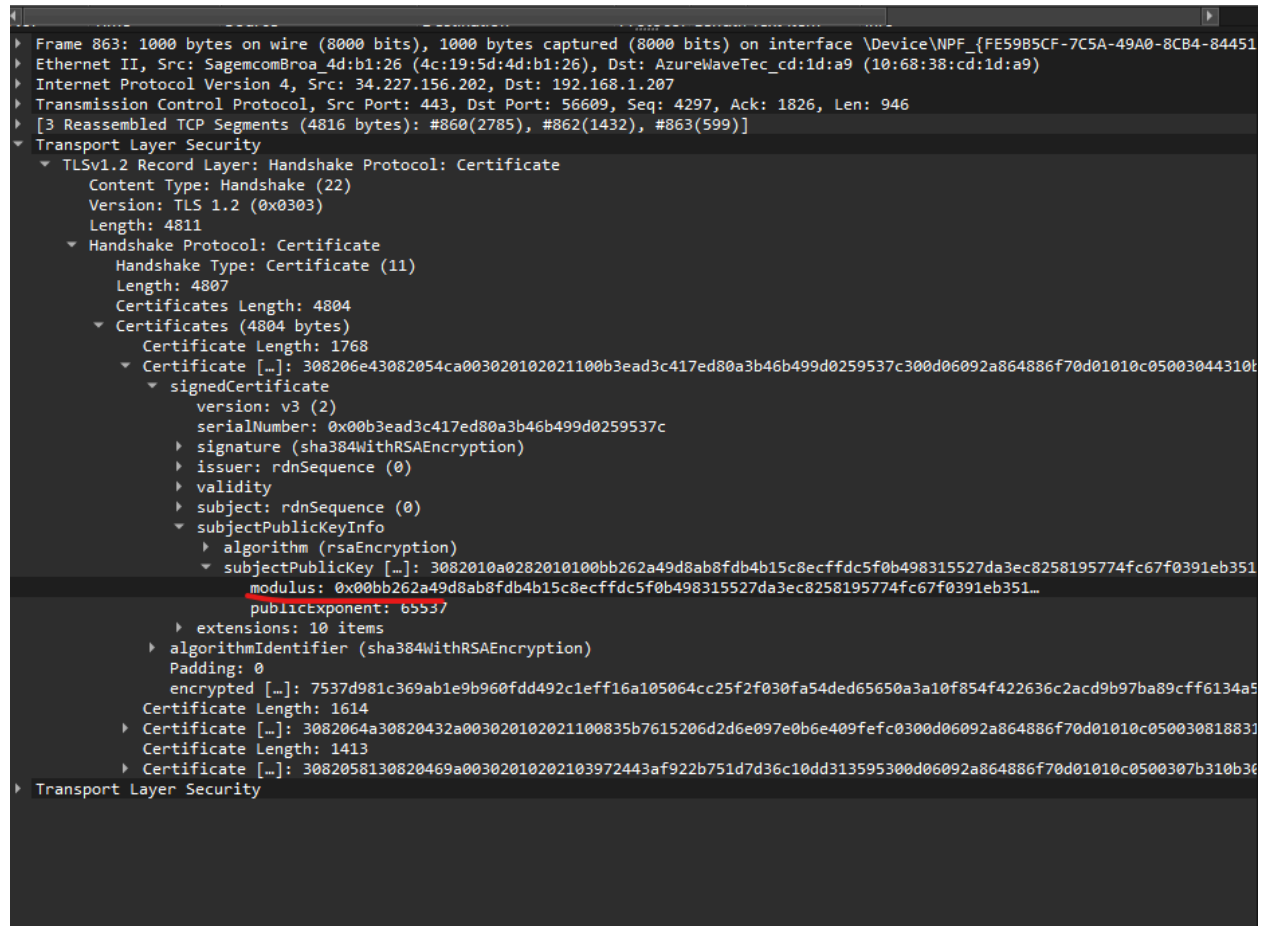
Frame 863: 1000 bytes on wire (8000 bits), 1000 bytes captured (8000 bits) on interface \Device\NPF_{FE59B5CF-7C5A-49A0-8CB4-844...}
Ethernet II, Src: SagemcomBroa_4d:b1:26 (4c:19:5d:4d:b1:26), Dst: AzureWaveTec_cd:1d:a9 (10:68:38:cd:1d:a9)
Internet Protocol Version 4, Src: 34.227.156.202, Dst: 192.168.1.207
Transmission Control Protocol, Src Port: 443, Dst Port: 56609, Seq: 4297, Ack: 1826, Len: 946
[3 Reassembled TCP Segments (4816 bytes), #860(2785), #862(1432), #863(599)]
Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 4811
    Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 4807
      Certificates Length: 4804
      Certificates (4804 bytes)
        Certificate Length: 1768
        Certificate [...]: 308206e43082054ca003020102021100b3ead3c417ed80a3b46b499d0259537c300d06092a864886f70d01010c05003044310b3009
          signedCertificate
            version: v3 (2)
            serialNumber: 0x00b3ead3c417ed80a3b46b499d0259537c
            signature (sha384WithRSAEncryption)
            Algorithm: id-1.2.840.113549.1.1.12 (sha384WithRSAEncryption)
            issuer: rdnSequence (0)
              rdnSequence: 3 items (id-at-commonName=InCommon RSA Server CA 2,id-at-organizationName=Internet2,id-at-countryName=US)
                RDNSequence item: 1 item (id-at-countryName=US)
                  RelativeDistinguishedName item (id-at-countryName=US)
                    Object Id: 2.5.4.6 (id-at-countryName)
                    CountryName: US
                  RDNSequence item: 1 item (id-at-organizationName=Internet2)
                    RelativeDistinguishedName item (id-at-organizationName=Internet2)
                  RDNSequence item: 1 item (id-at-commonName=InCommon RSA Server CA 2)
                    RelativeDistinguishedName item (id-at-commonName=InCommon RSA Server CA 2)
                validity
              subject: rdnSequence (0)
                rdnSequence: 4 items (id-at-commonName=www.cics.umass.edu,id-at-organizationName=University of Massachusetts Amherst)
                  RDNSequence item: 1 item (id-at-countryName=US)
                    RelativeDistinguishedName item (id-at-countryName=US)
                      Object Id: 2.5.4.6 (id-at-countryName)
                      CountryName: US
                    RDNSequence item: 1 item (id-at-stateOrProvinceName=Massachusetts)
                      RelativeDistinguishedName item (id-at-stateOrProvinceName=Massachusetts)
                        Object Id: 2.5.4.8 (id-at-stateOrProvinceName)
                        DirectoryString: printableString (1)
                    RDNSequence item: 1 item (id-at-organizationName=University of Massachusetts Amherst)
                      RelativeDistinguishedName item (id-at-organizationName=University of Massachusetts Amherst)

```

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.

First 4 hexadecimal digits for modulus of the public key:

bb26



```
Frame 863: 1000 bytes on wire (8000 bits), 1000 bytes captured (8000 bits) on interface \Device\NPF_{FE59B5CF-7C5A-49A0-8CB4-84451}
Ethernet II, Src: SagemcomBroa_4d:b1:26 (4c:19:5d:4d:b1:26), Dst: AzureWaveTec_cd:1d:a9 (10:68:38:cd:1d:a9)
Internet Protocol Version 4, Src: 34.227.156.202, Dst: 192.168.1.207
Transmission Control Protocol, Src Port: 443, Dst Port: 56609, Seq: 4297, Ack: 1826, Len: 946
[3 Reassembled TCP Segments (4816 bytes): #860(2785), #862(1432), #863(599)]
Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 4811
    Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 4807
      Certificates Length: 4804
      Certificates (4804 bytes)
        Certificate Length: 1768
        Certificate [...]
          signedCertificate
            version: v3 (2)
            serialNumber: 0x00b3ead3c417ed80a3b46b499d0259537c
            signature (sha384WithRSAEncryption)
            issuer: rdnSequence (0)
            validity
            subject: rdnSequence (0)
            subjectPublicKeyInfo
              algorithm (rsaEncryption)
              subjectPublicKey [...]
                modulus: 0x00bb262a49d8ab8fdb4b15c8ecffdc5f0b498315527da3ec8258195774fc67f0391eb351...
                publicExponent: 65537
              extensions: 10 items
            algorithmIdentifier (sha384WithRSAEncryption)
            Padding: 0
            encrypted [...]
          Certificate Length: 1614
        Certificate [...]
          Certificate Length: 1413
        Certificate [...]
          Certificate Length: 1413
Transport Layer Security
```

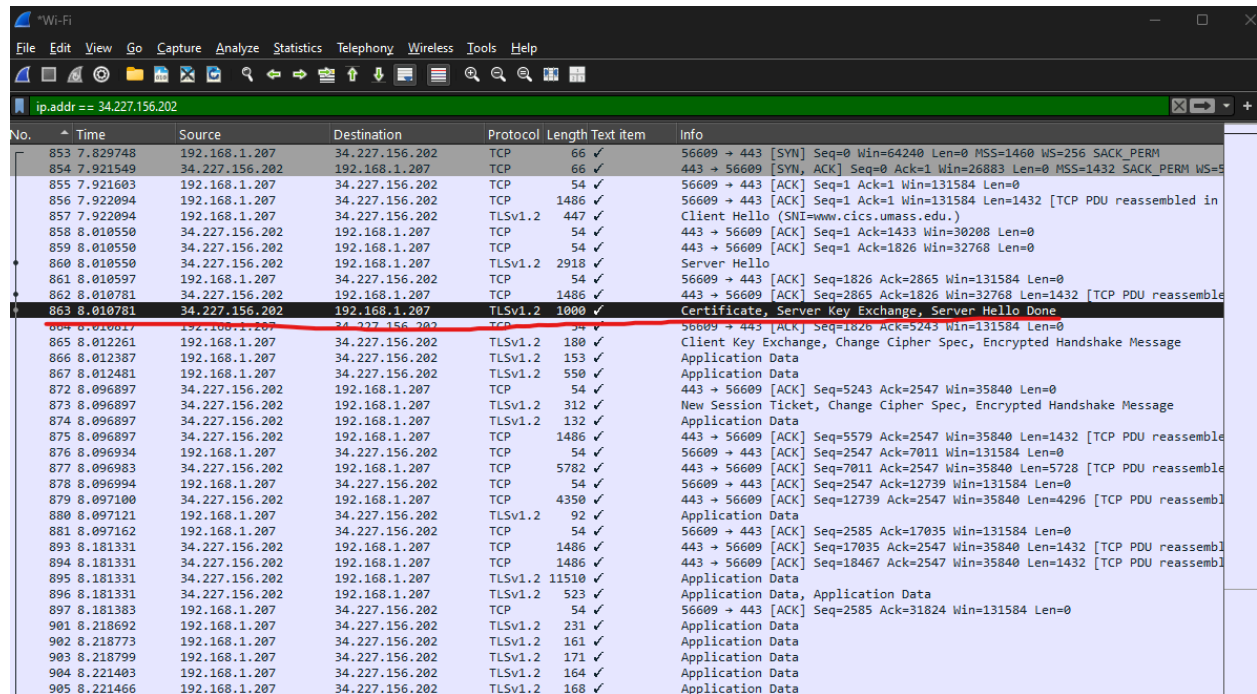
16. Look in your trace to find messages between the client and a CA to get the CA's public key information, so that the client can verify that the CA-signed certificate sent by the server is indeed valid and has not been forged or altered. Do you see such message in your trace? If so, what is the number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

There is no packet in the trace showing the client contacting the CA. The certificate validation process is handled by the client using its local store of trusted certificates

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?

Packet Number 863 contains the Server Hello Done TLS record



Wireshark packet capture showing a TLS handshake. The packet list on the left shows packet 863 selected. The packet details pane on the right shows the structure of the TLS record.

No.	Time	Source	Destination	Protocol	Length	Text item	Info
853	7.829748	192.168.1.207	34.227.156.202	TCP	66	✓	56609 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
854	7.921549	34.227.156.202	192.168.1.207	TCP	66	✓	443 → 56609 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1432 SACK_PERM WS=5
855	7.921603	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0
856	7.922094	192.168.1.207	34.227.156.202	TCP	1486	✓	56609 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=1432 [TCP PDU reassembled in
857	7.922094	192.168.1.207	34.227.156.202	TLSv1.2	447	✓	Client Hello (SNI=www.cics.umass.edu.)
858	8.010550	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=1 Ack=1433 Win=30208 Len=0
859	8.010550	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=1 Ack=1826 Win=32768 Len=0
860	8.010550	34.227.156.202	192.168.1.207	TLSv1.2	2918	✓	Server Hello
861	8.010597	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1826 Ack=2865 Win=131584 Len=0
862	8.010781	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=2865 Ack=1826 Win=32768 Len=1432 [TCP PDU reassembled
863	8.010781	34.227.156.202	192.168.1.207	TLSv1.2	1000	✓	Certificate, Server Key Exchange, Server Hello Done
864	8.010807	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=1826 Ack=5243 Win=131584 Len=0
865	8.012261	192.168.1.207	34.227.156.202	TLSv1.2	180	✓	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
866	8.012387	192.168.1.207	34.227.156.202	TLSv1.2	153	✓	Application Data
867	8.012481	192.168.1.207	34.227.156.202	TLSv1.2	550	✓	Application Data
872	8.096897	34.227.156.202	192.168.1.207	TCP	54	✓	443 → 56609 [ACK] Seq=5243 Ack=2547 Win=35840 Len=0
873	8.096897	34.227.156.202	192.168.1.207	TLSv1.2	312	✓	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
874	8.096897	34.227.156.202	192.168.1.207	TLSv1.2	132	✓	Application Data
875	8.096897	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=5579 Ack=2547 Win=35840 Len=1432 [TCP PDU reassembled
876	8.096934	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2547 Ack=7011 Win=131584 Len=0
877	8.096983	34.227.156.202	192.168.1.207	TCP	5782	✓	443 → 56609 [ACK] Seq=7011 Ack=2547 Win=35840 Len=5728 [TCP PDU reassembled
878	8.096994	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2547 Ack=12739 Win=131584 Len=0
879	8.097100	34.227.156.202	192.168.1.207	TCP	4350	✓	443 → 56609 [ACK] Seq=12739 Ack=2547 Win=35840 Len=4296 [TCP PDU reassembled
880	8.097121	192.168.1.207	34.227.156.202	TLSv1.2	92	✓	Application Data
881	8.097162	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2585 Ack=17035 Win=131584 Len=0
893	8.181331	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=17035 Ack=2547 Win=35840 Len=1432 [TCP PDU reassembled
894	8.181331	34.227.156.202	192.168.1.207	TCP	1486	✓	443 → 56609 [ACK] Seq=18467 Ack=2547 Win=35840 Len=1432 [TCP PDU reassembled
895	8.181331	34.227.156.202	192.168.1.207	TLSv1.2	11510	✓	Application Data
896	8.181331	34.227.156.202	192.168.1.207	TLSv1.2	523	✓	Application Data
897	8.181383	192.168.1.207	34.227.156.202	TCP	54	✓	56609 → 443 [ACK] Seq=2585 Ack=31824 Win=131584 Len=0
901	8.218692	192.168.1.207	34.227.156.202	TLSv1.2	231	✓	Application Data
902	8.218773	192.168.1.207	34.227.156.202	TLSv1.2	161	✓	Application Data
903	8.218799	192.168.1.207	34.227.156.202	TLSv1.2	171	✓	Application Data
904	8.221403	192.168.1.207	34.227.156.202	TLSv1.2	164	✓	Application Data
905	8.221466	192.168.1.207	34.227.156.202	TLSv1.2	168	✓	Application Data