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

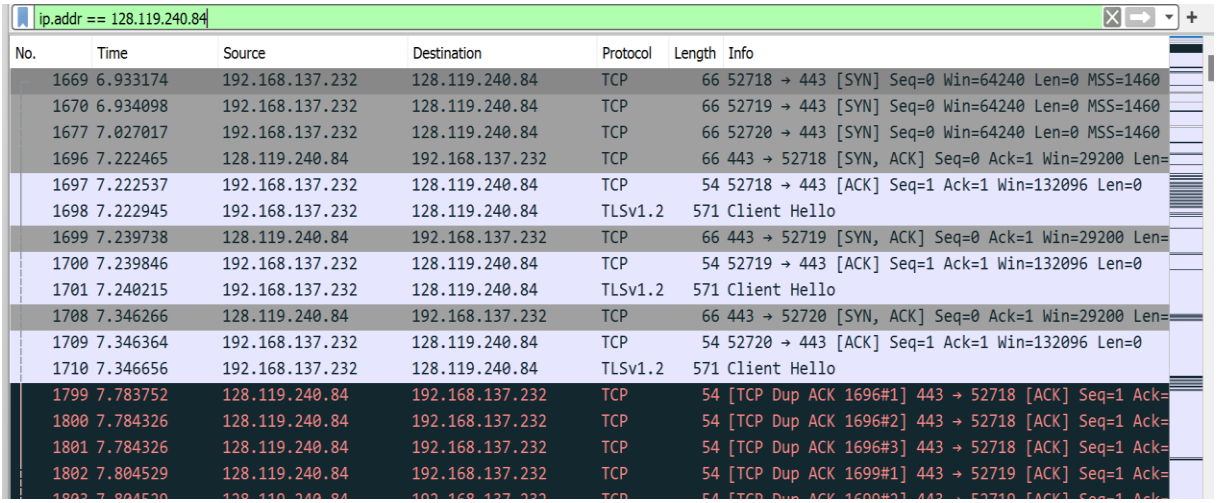
Name: Janakar Patel
Roll No: 20BIT061

Experiment-10

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

Software Tools required: Wire-shark.

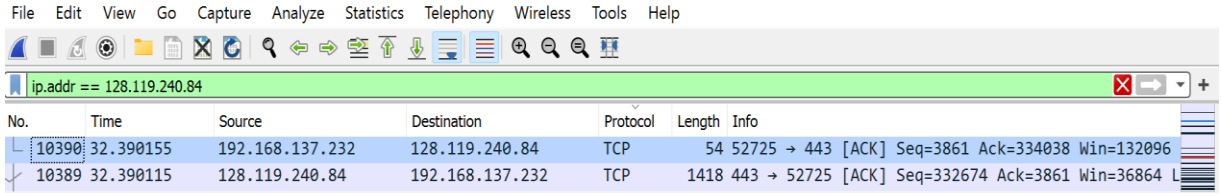
Capturing packets in an TLS session



A first look at the captured trace

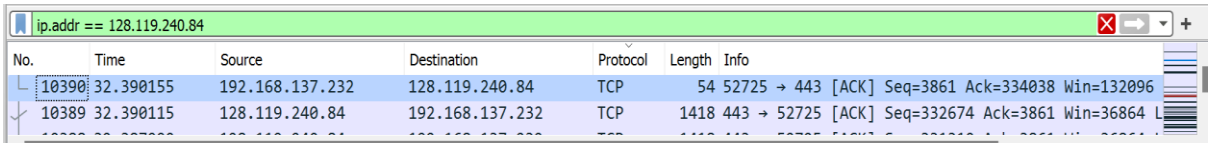
Q.1. What is the packet number in your trace that contains the initial TCP SYN message? (By "packet number," we meant the number in the "No." column at the left of the Wireshark display, not the sequence number in the TCP segment itself).

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



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

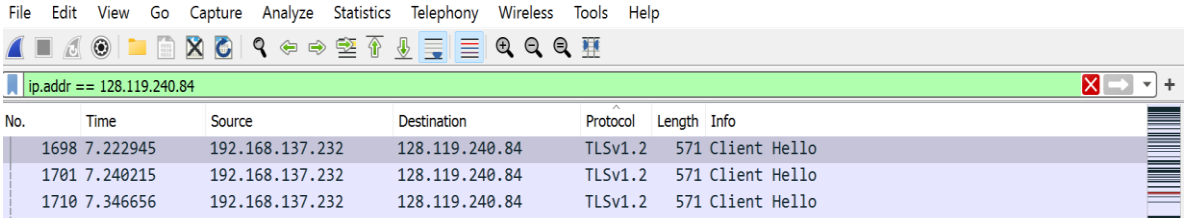
Ans: Yes, connection is setup after TLS message is sent from client to server.



The TLS Handshake: *Client Hello* message

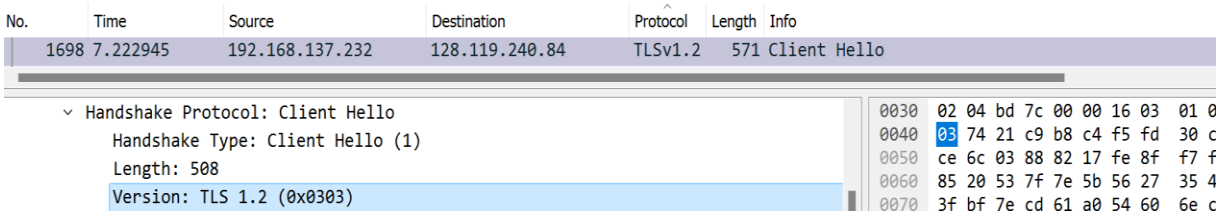
Q.3. What is the packet number in your trace that contains the TLS *Client Hello* message?

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



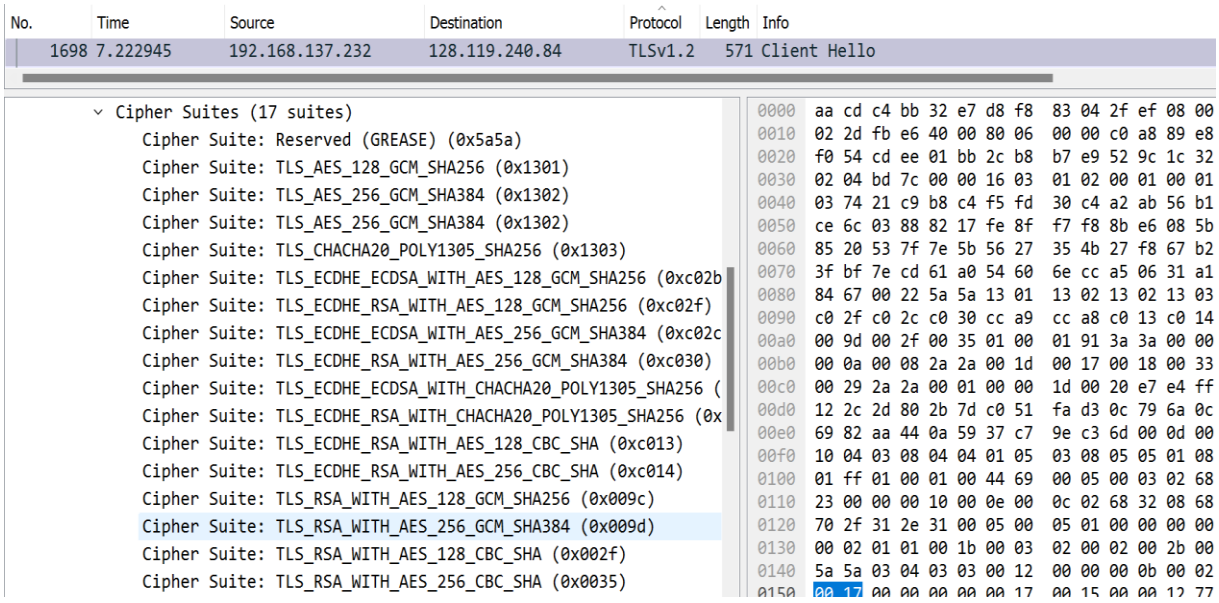
Q.4. What version of TLS is your client running, as declared in the *Client Hello* message?

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



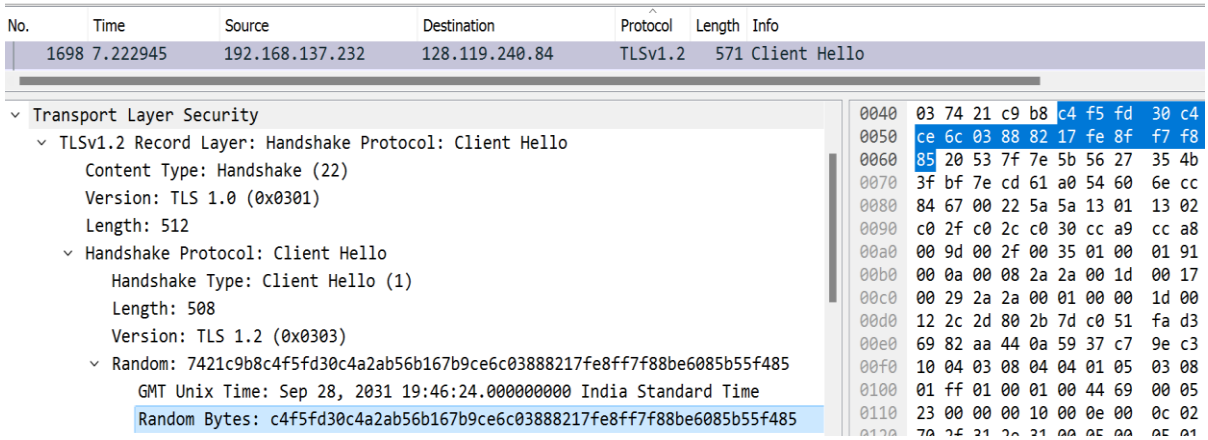
Q.5. How many cipher suites are supported by your client, as declared in the *Client Hello* message? A cipher suite is a set of related cryptographic algorithms that determine how session keys will be derived, and how data will be encrypted and be digitally signed via a HMAC algorithm.`

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



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



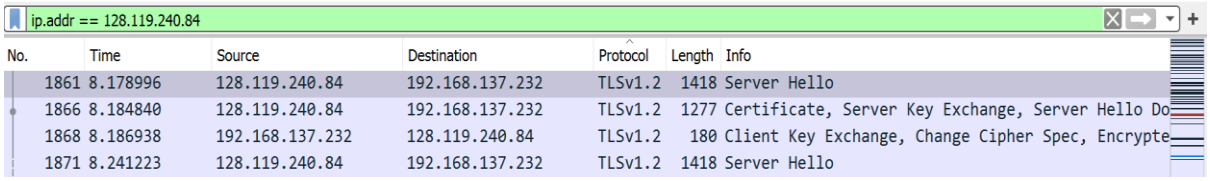
Q.7. What is the purpose(s) of the "random bytes" field in the *Client Hello* message? Note: you'll have do some searching and reading to get the answer to this question; see section 8.6 and in RFC 5246 (section 8.1 in RFC 5246 in particular).

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

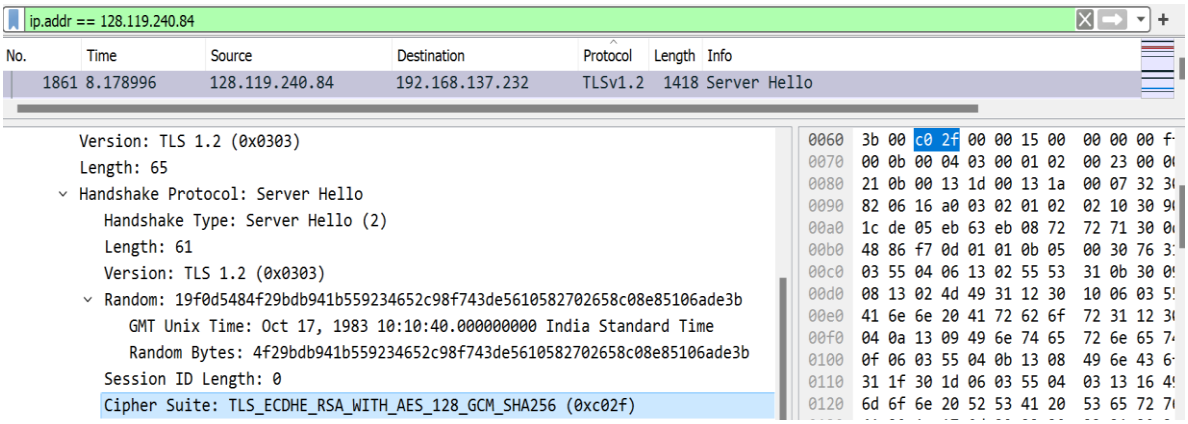
Q.8. What is the packet number in your trace that contains the TLS *Server Hello* message?

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



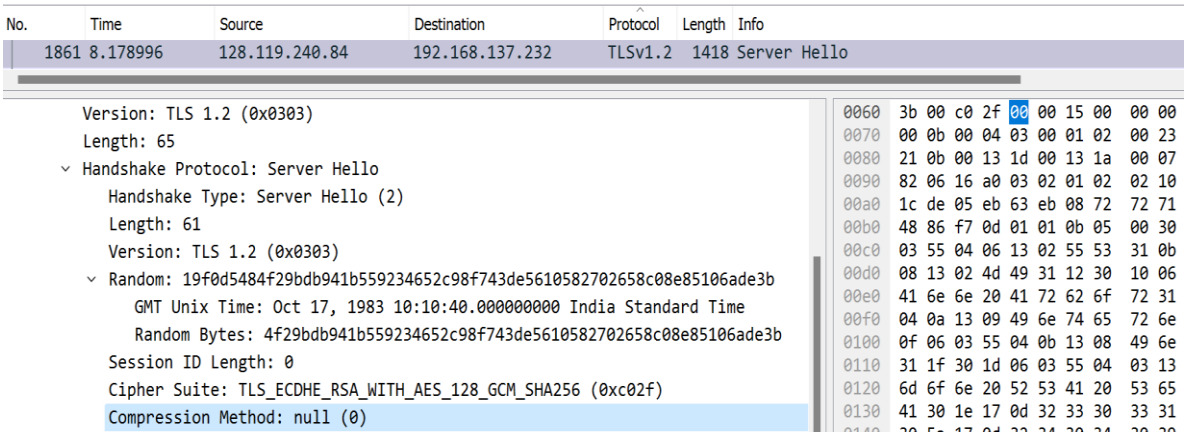
Q.9. Which cipher suite has been chosen by the server from among those offered in the earlier *Client Hello* message?

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)



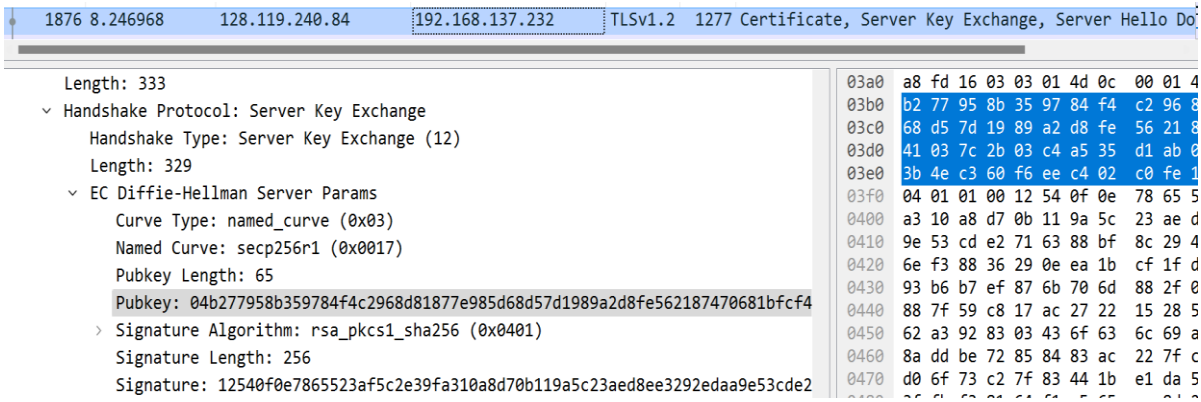
Q.10. Does the *Server Hello* message contain random bytes, similar to how the *Client Hello* message contained random bytes? And if so, what is/are their purpose(s)?

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



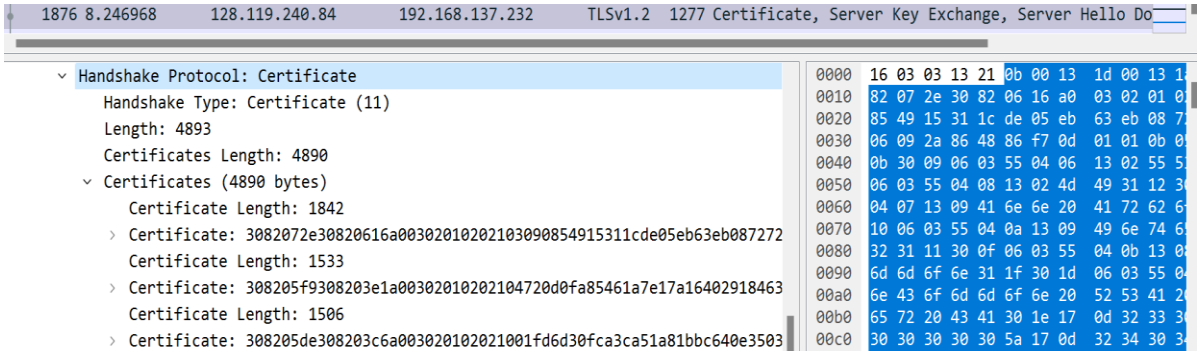
Q.11. What is the packet number in your trace for the TLS message part that contains the public key certificate for the www.cics.umass.edu server (actually the www.cs.umass.edu server)?

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



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 retuned 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'.



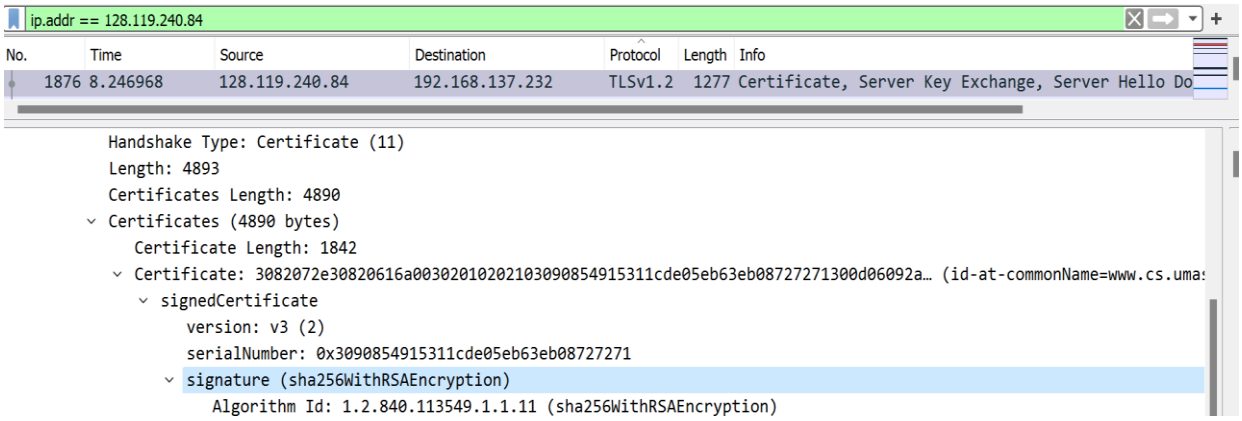
Q.13.What is the name of the certification authority that issued the certificate for id-a

commonName=www.cs.umass.edu?

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

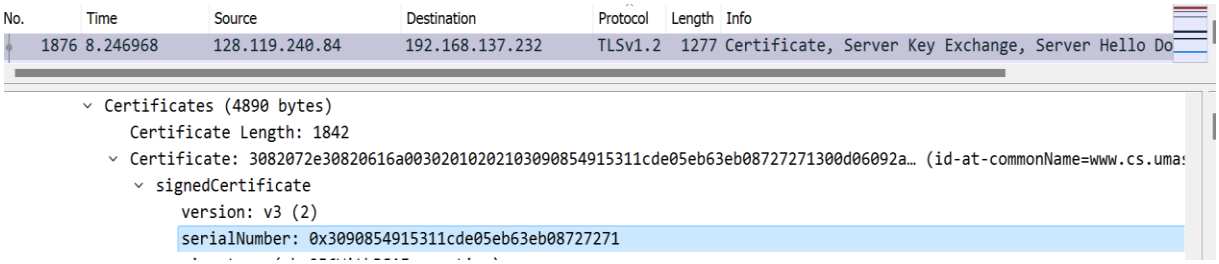
Q. 14. What digital signature algorithm is used by the CA to sign this certificate? Hint: this information can be found in signature subfield of the SignedCertificate field of the certificate for www.cs.umass.edu.

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



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

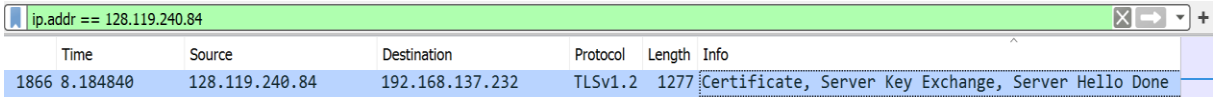


Q.16. Look in your trace to find messages between the client and a CA to get the CA's public key information, so that the client can verify that the CA-signed certificate sent by the server is indeed valid and has not been forged or altered. Do you see such message in your trace? If so, what is the number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

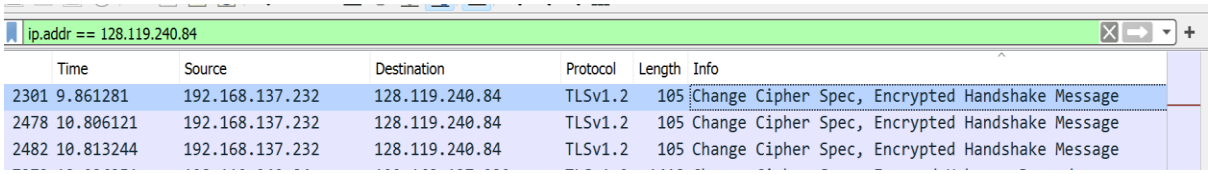
Ans: Packet Number that contains the TLS *Server Hello Done* message: 1866



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



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.

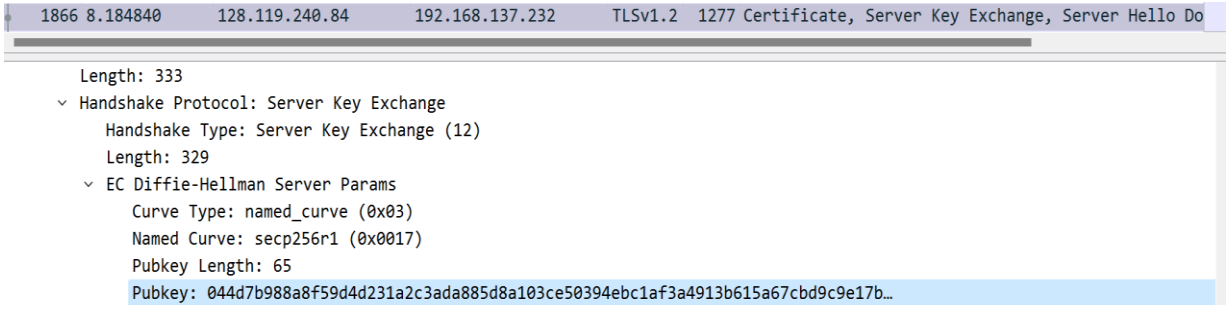
Application data

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.

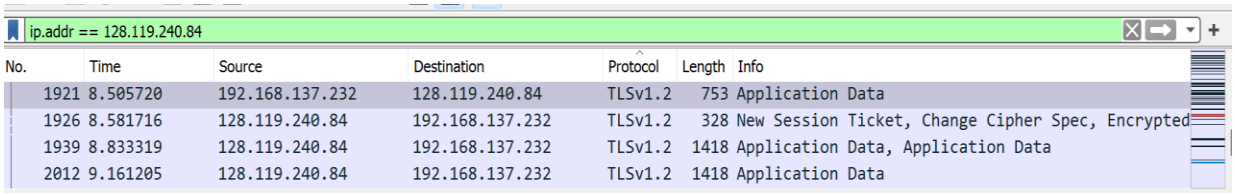
Q.21 In which of the TLS messages is this symmetric key cryptography algorithm finally decided and declared?

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



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



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