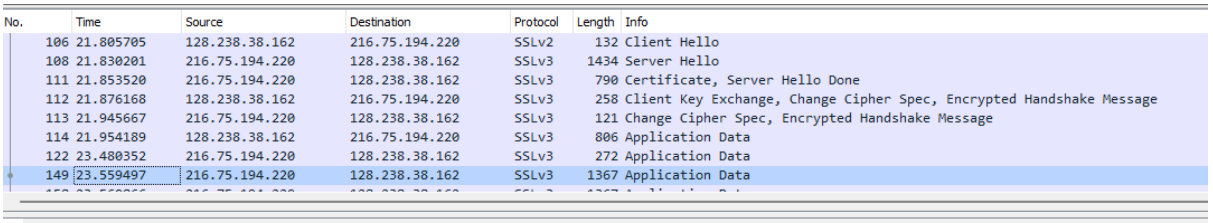
Lab_8a_Wireshark_SSL_v8.0

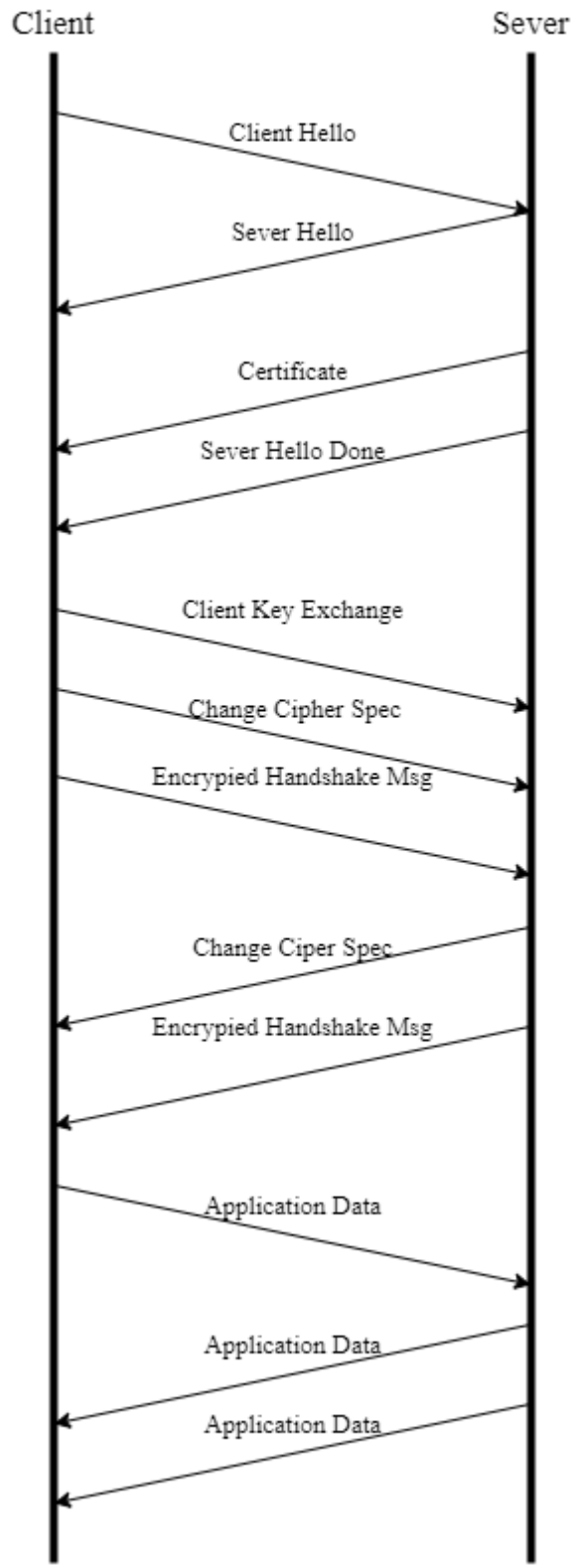
1.For each of the first 8 Ethernet frames, specify the source of the frame (client or server), determine the number of SSL records that are included in the frame, and list the SSL record types that are included in the frame. Draw a timing diagram between client and server, with one arrow for each SSL record.

Answer:



No.	Time	Source	Destination	Protocol	Length	Info
106	21.805705	128.238.38.162	216.75.194.220	SSLv2	132	Client Hello
108	21.830201	216.75.194.220	128.238.38.162	SSLv3	1434	Server Hello
111	21.853520	216.75.194.220	128.238.38.162	SSLv3	790	Certificate, Server Hello Done
112	21.876168	128.238.38.162	216.75.194.220	SSLv3	258	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
113	21.945667	216.75.194.220	128.238.38.162	SSLv3	121	Change Cipher Spec, Encrypted Handshake Message
114	21.954189	128.238.38.162	216.75.194.220	SSLv3	806	Application Data
122	23.480352	216.75.194.220	128.238.38.162	SSLv3	272	Application Data
149	23.559497	216.75.194.220	128.238.38.162	SSLv3	1367	Application Data

No.	Frame	Source	SSL Count	SSL Type
1	106	Client	1	Client Hello
2	108	Server	1	Server Hello
3	111	Server	2	Certificate Server Hello Done
4	112	Client	3	Client Key Exchange Change Cipher Spec Encrypted Handshake Message
5	113	Server	2	Change Cipher Spec Encrypted Handshake Message
6	114	Client	1	Application Data
7	122	Server	1	Application Data
8	149	Server	1	Application Data

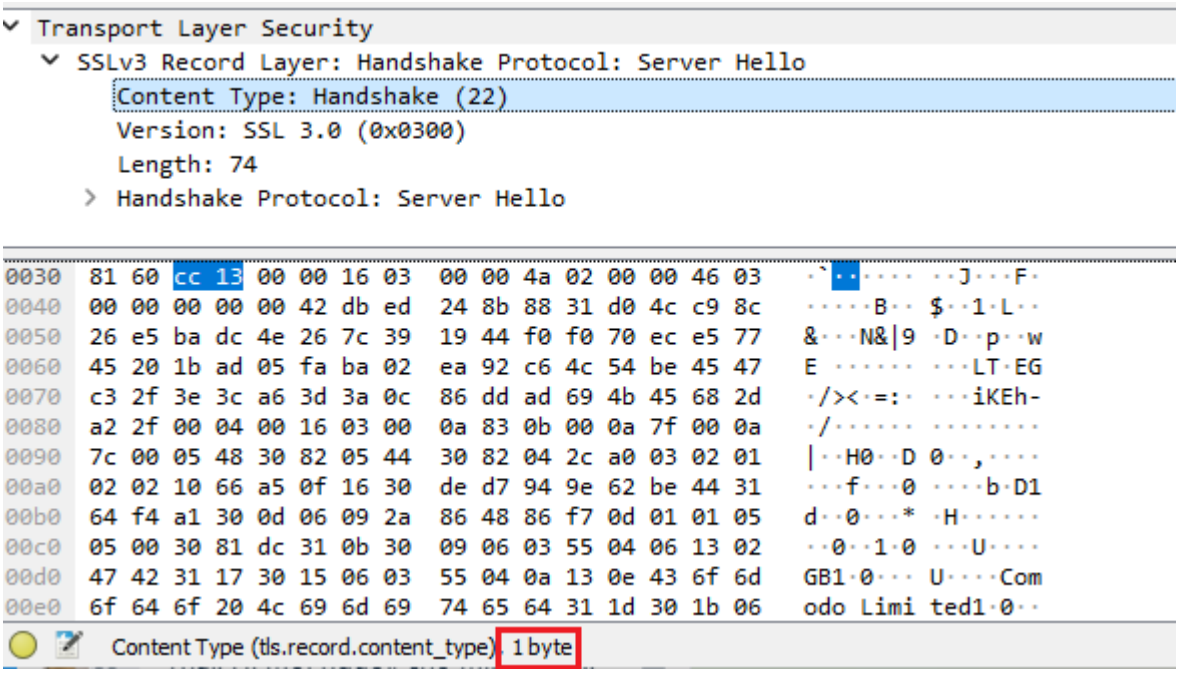


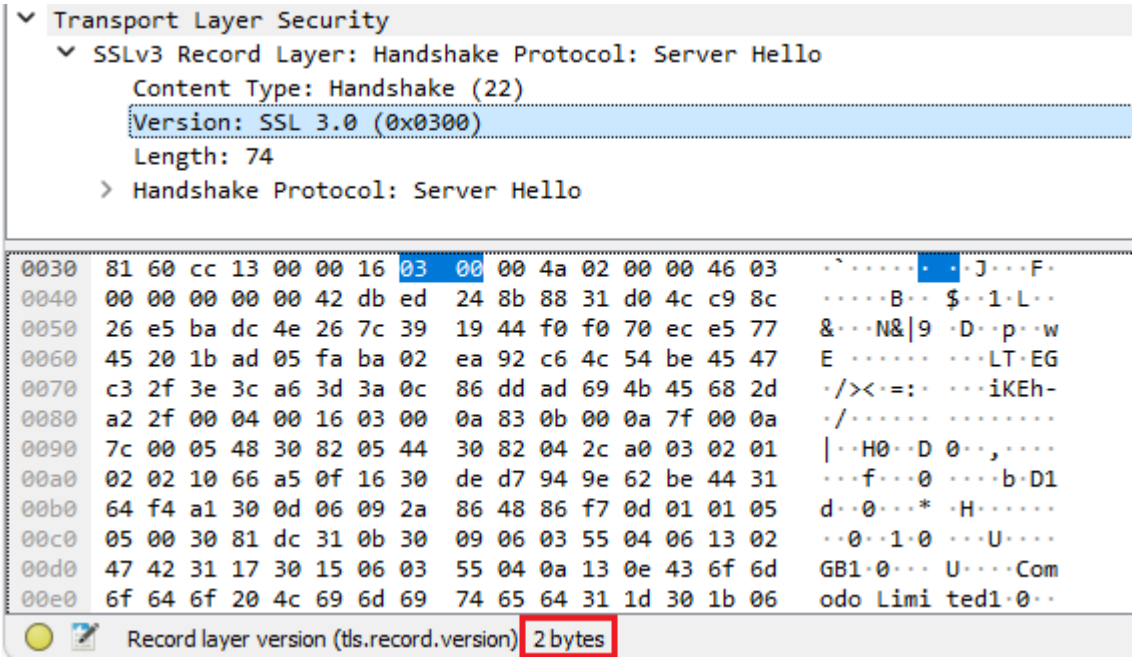
2.Each of the SSL records begins with the same three fields (with possibly different values). One of these fields is “content type” and has length of one byte. List all three fields and their lengths.

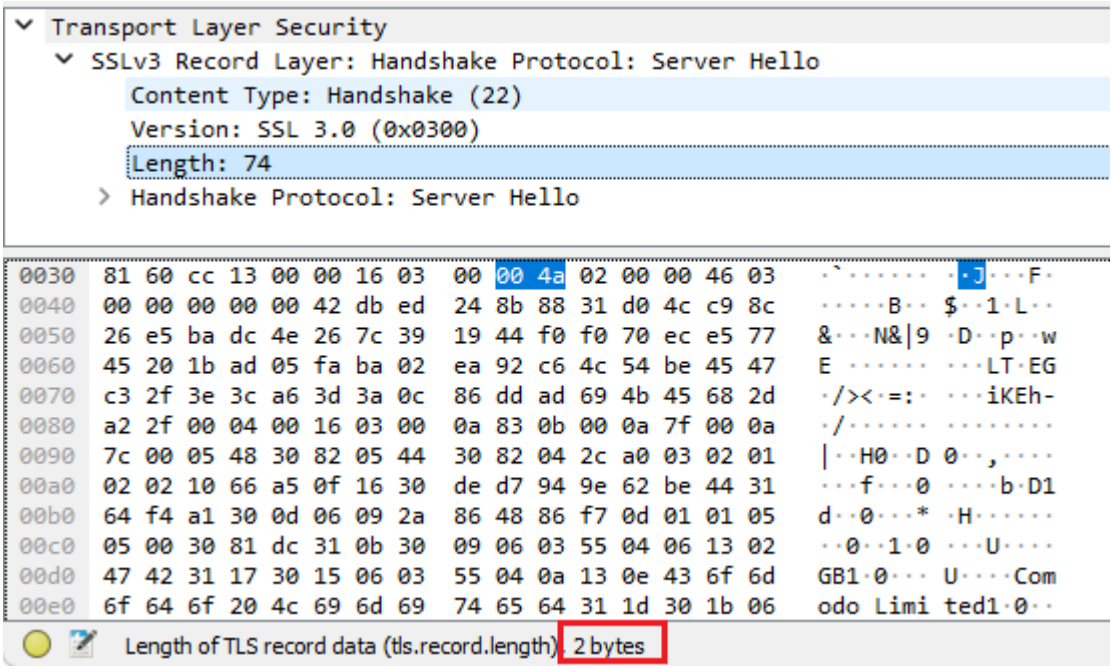
Answer:

- Content Type: 1 byte

- Version: 2 bytes
- Length: 2 bytes

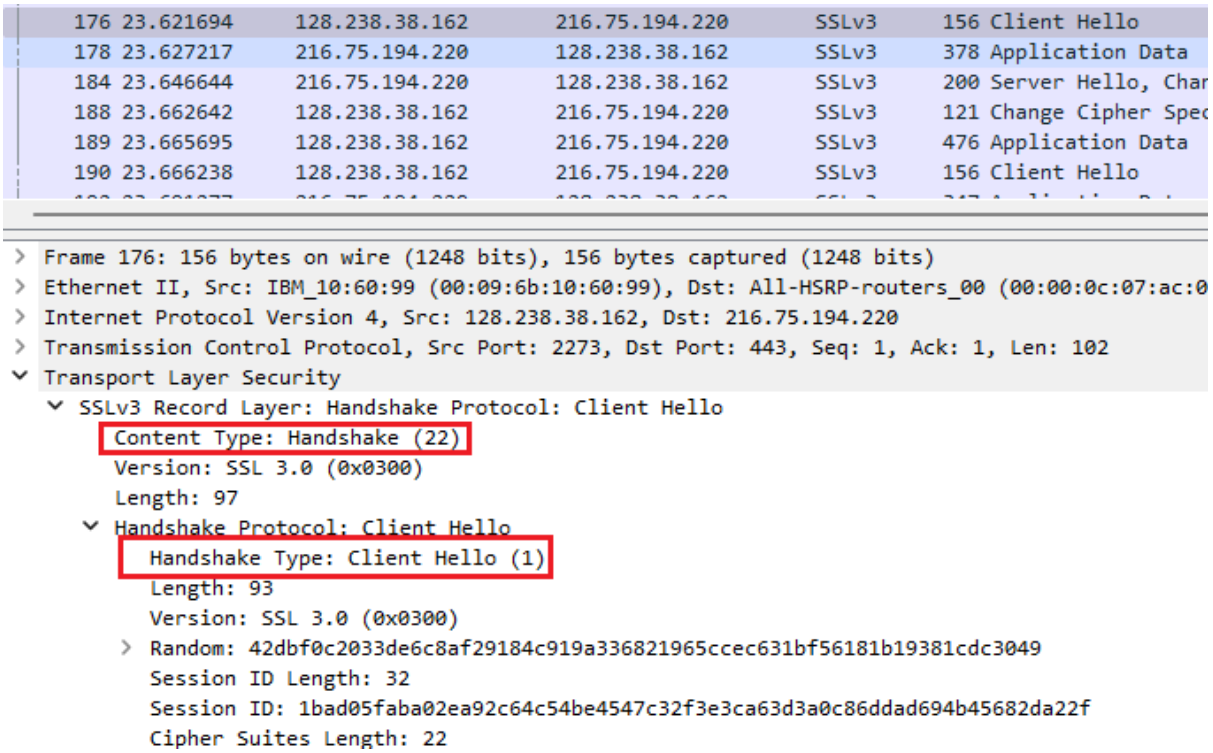






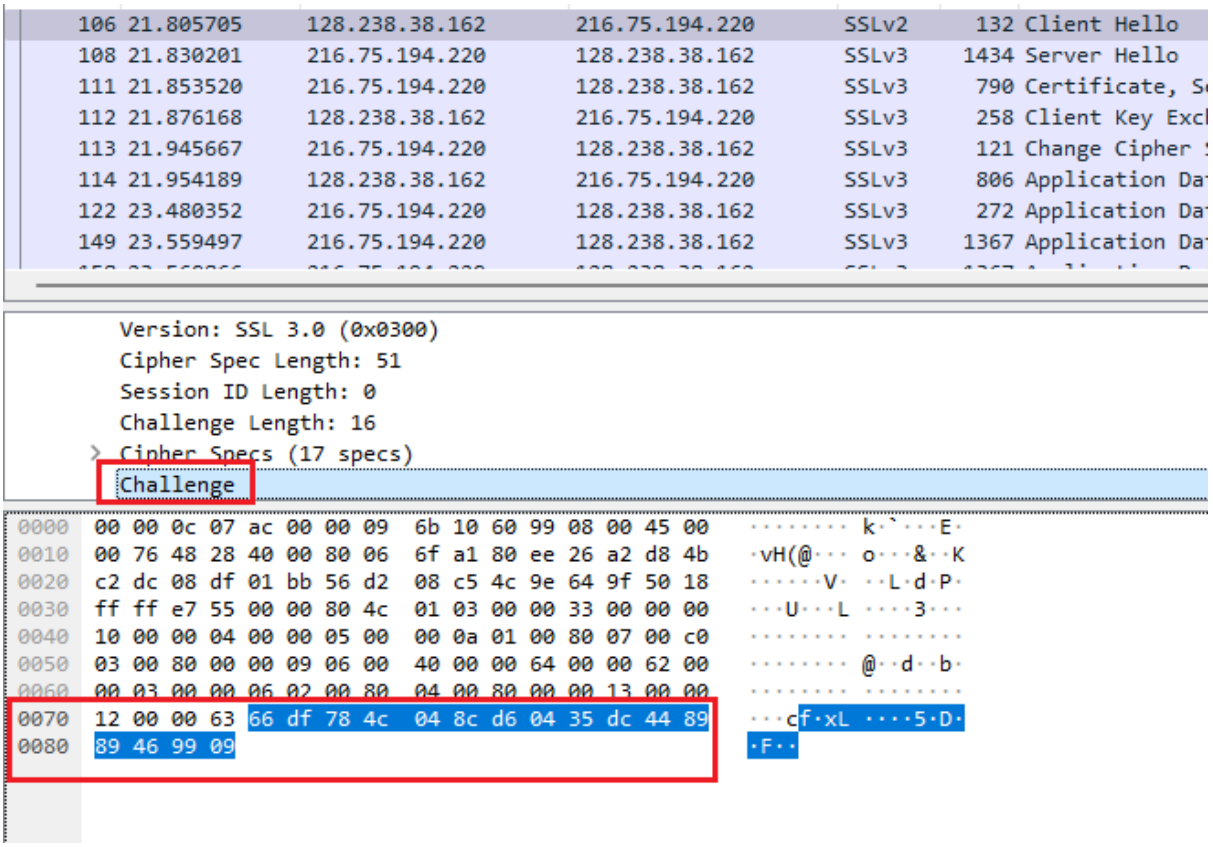
3.Expand the ClientHello record. (If your trace contains multiple ClientHello records, expand the frame that contains the first one.) What is the value of the content type?

Answer: The content type is 22, for Handshake Message, with a handshake type of 01, Client Hello



4.Does the ClientHello record contain a nonce (also known as a “challenge”)? If so, what is the value of the challenge in hexadecimal notation?

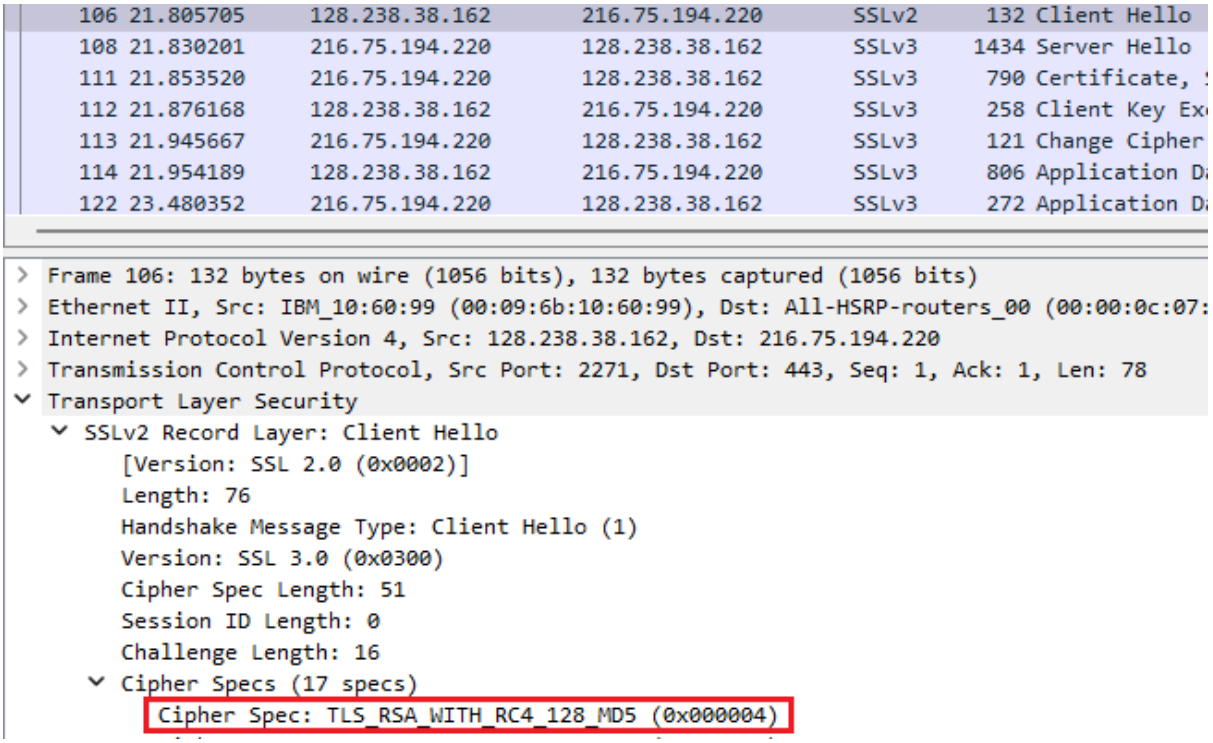
Answer: The ClientHello Record contains a Challenge and it is: 66 df 78 4c 04 8c d6 04 35 dc 44 89 89 46 99 09.



5.Does the ClientHello record advertise the cyber suites it supports? If so, in the first listed suite, what are the public-key algorithm, the symmetric-key algorithm, and the hash algorithm?

Answer:

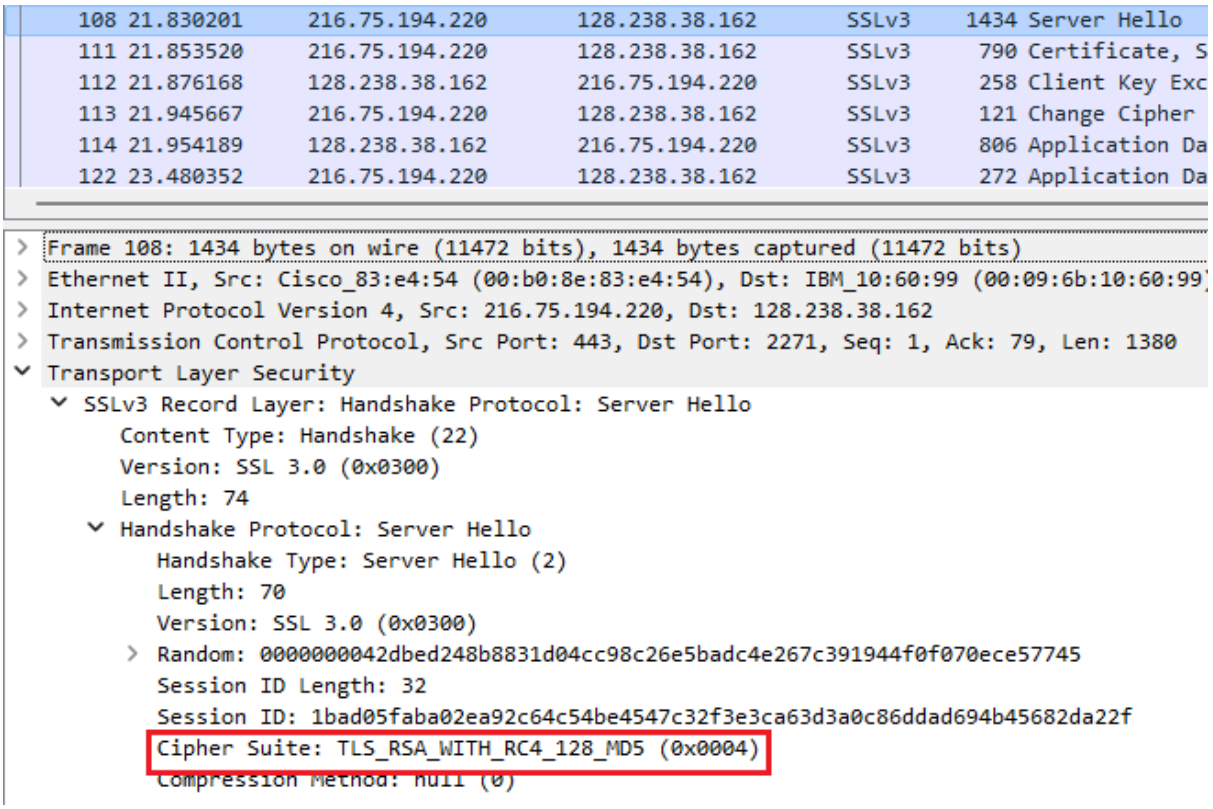
- Public key algorithm: RSA
- Symmetric-key algorithm: RC4
- Hash algorithm: MD5



6.Locate the ServerHello SSL record. Does this record specify a chosen cipher suite? What are the algorithms in the chosen cipher suite?

Answer:

- Public key algorithm: RSA
- Symmetric-key algorithm: RC4
- Hash algorithm: MD5

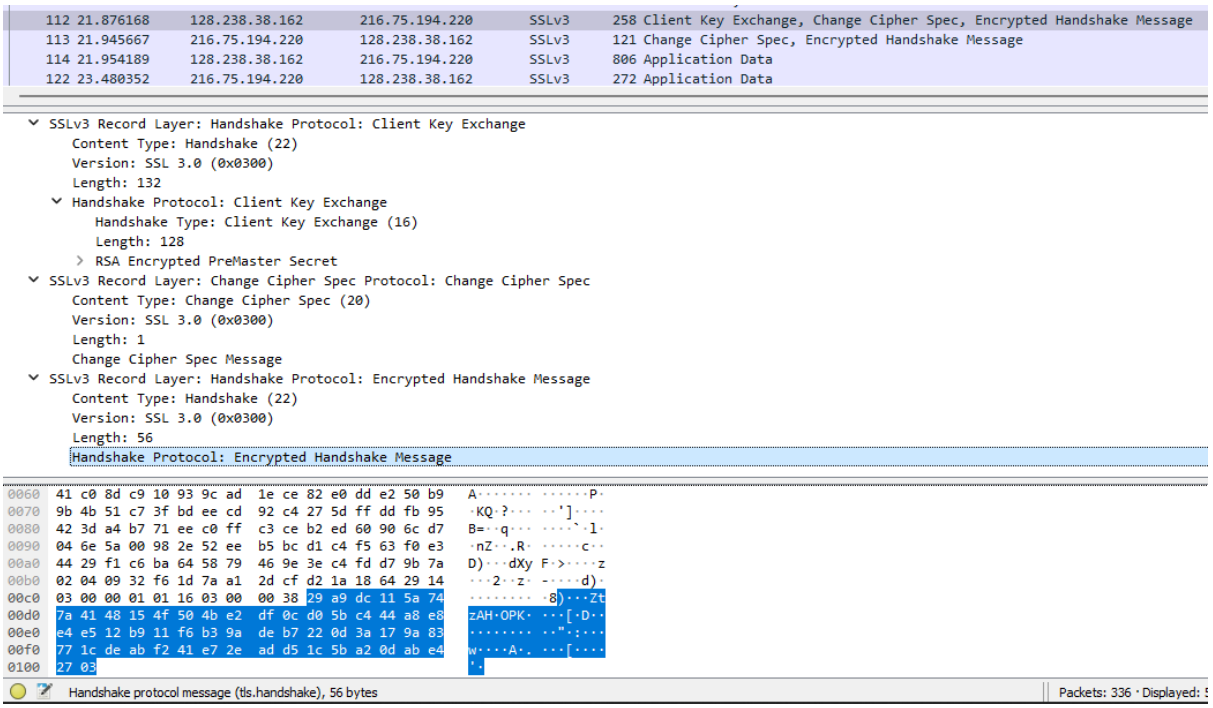


7.Does this record include a nonce? If so, how long is it? What is the purpose of the client and server nonces in SSL?
Answer: Yes, it is 32 bits long (28bits data + 4 bits time), it is used for attack preventing.

8.Does this record include a session ID? What is the purpose of the session ID?
Answer: Yes, the session ID in the record is an identifier for SSL session. This ID could let the client to resume the session later by using the session ID

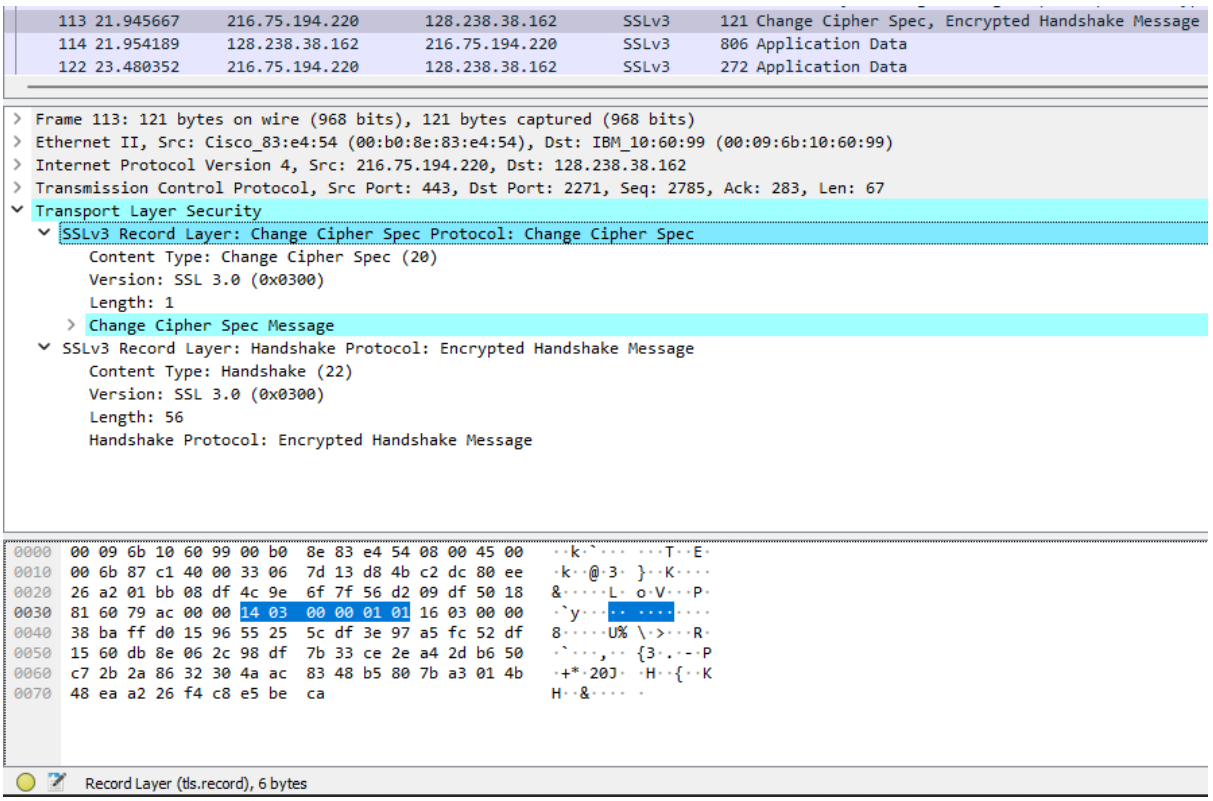
9.Does this record contain a certificate, or is the certificate included in a separate record. Does the certificate fit into a single Ethernet frame?
Answer: No, there is no certificate in this record. The certificate is in the separate record. Yes, the certificate fit into a single Ethernet frame

10.Locate the client key exchange record. Does this record contain a pre-master secret? What is this secret used for? Is the secret encrypted? If so, how? How long is the encrypted secret?
Answer: Yes, it does contain a pre-master secret. It is used by both the server and client to make a master secret, which used to generate session keys for MAC and encryption. The secret gets encrypted using the server's public key, which the client extracted from ther certificate sent by the sever. The encrypted secret is 56 bytes.



11.What is the purpose of the Change Cipher Spec record? How many bytes is the record in your trace?

Answer: The purpose of the Change Cipher Spec record is to indicate that the contents of the following SSL records sent by the client (data, not header) will be encrypted. This record is 6 bytes long: 5 for header and 1 for message segment



12.In the encrypted handshake record, what is being encrypted? How?
Answer: All handshake messages and MAC addresses are concatenated and encrypted. They are sent to the server.

13.Does the server also send a change cipher record and an encrypted handshake record to the client? How are those records different from those sent by the client?
Answer: Yes, the server will also send a Change Cipher Spec record and encrypted handshake to the client. The server's encrypted handshake record is different from that sent by the client because it contains the concatenation of all the handshake messages sent from the server rather than from the client. Otherwise the records would end up being the same.

14.How is the application data being encrypted? Do the records containing application data include a MAC? Does Wireshark distinguish between the encrypted application data and the MAC?
Answer: The symmetric encryption algorithm is used to encrypt the application data. Yes, the records containing application data include a MAC. No, Wireshark did not distinguish between the encrypted application data and the MAC.

15.Comment on and explain anything else that you found interesting in the trace.

Answer:

- The version of SSL used changes from SSLv2 in the initial ClientHello message to SSLv3 in all following message exchanges.
- Also, during resumes the handshake process is slightly different from the initial one. The client does not need another cert to the server never sends it. It just has to send a new nonce followed by Change Cipher Spec and Encypted Handshake records from the server to client. After a response from the client then application data can be sent