Solution to Wireshark Lab: SSL

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.

Frame	Source	SSL Count	SSL Type
106	Client	1	Client Hello
108	Server	1	Server Hello
111	Server	2	Certificate Server Hello Done
112	Client	3	Client Key Exchange Change Cipher Spec Encrypted Handshake Message
113	Server	2	Change Cipher Spec Encrypted Handshake Message
114	Client	1	Application Data
122	Server	1	Application Data
127	Server	1	Application Data

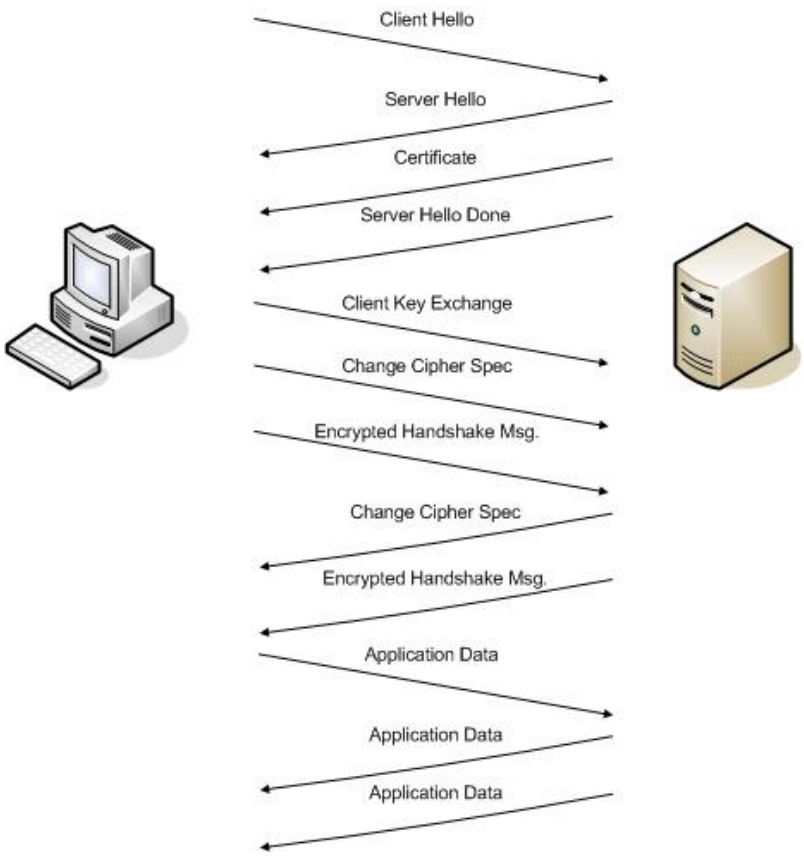

Fig. 1: Timing Diagram

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.

   The first three fields are:

   Content Type 1 byte
   Version 2 bytes
   Length 2 bytes.

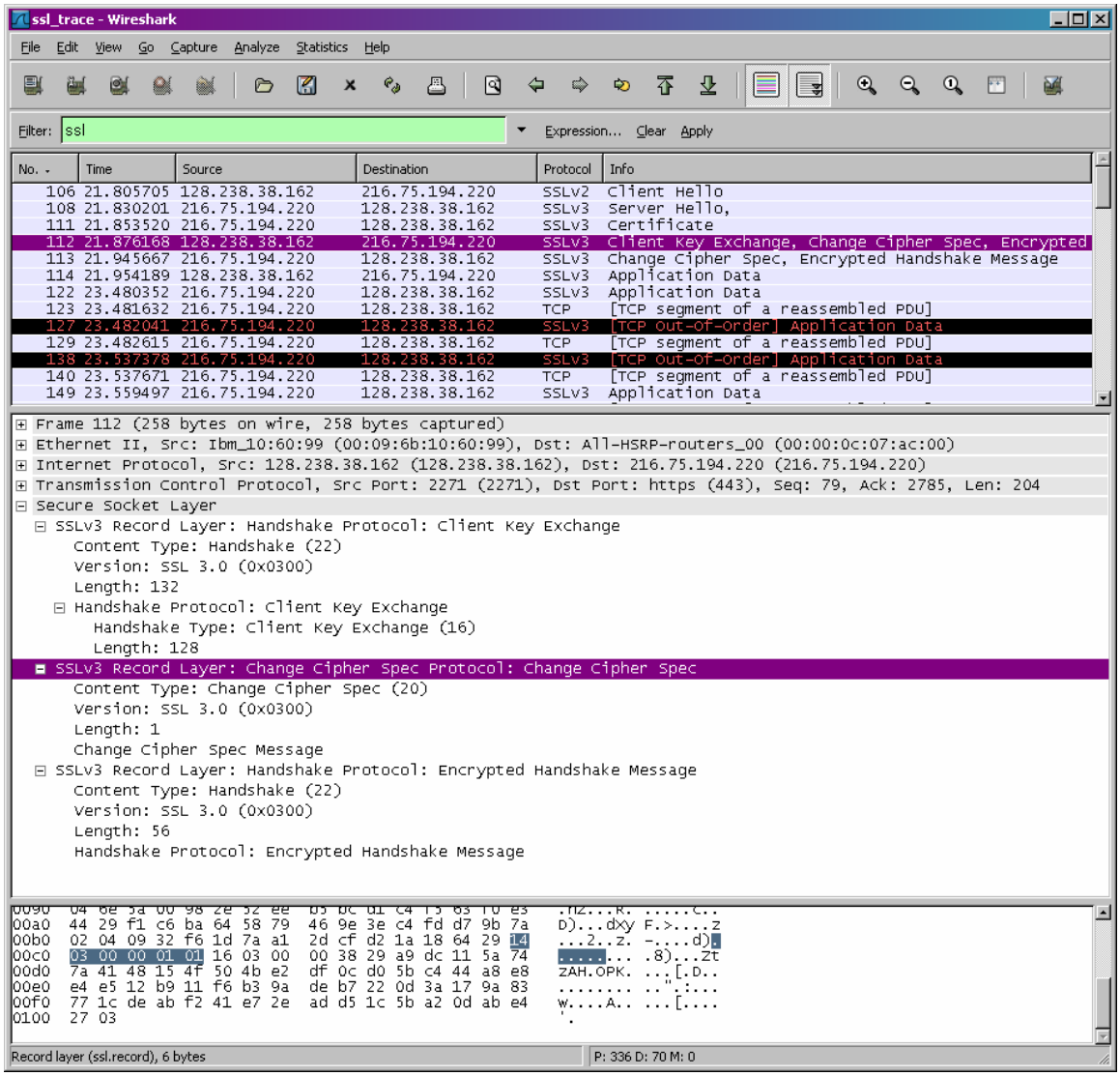

Fig. 2: SSL Frames

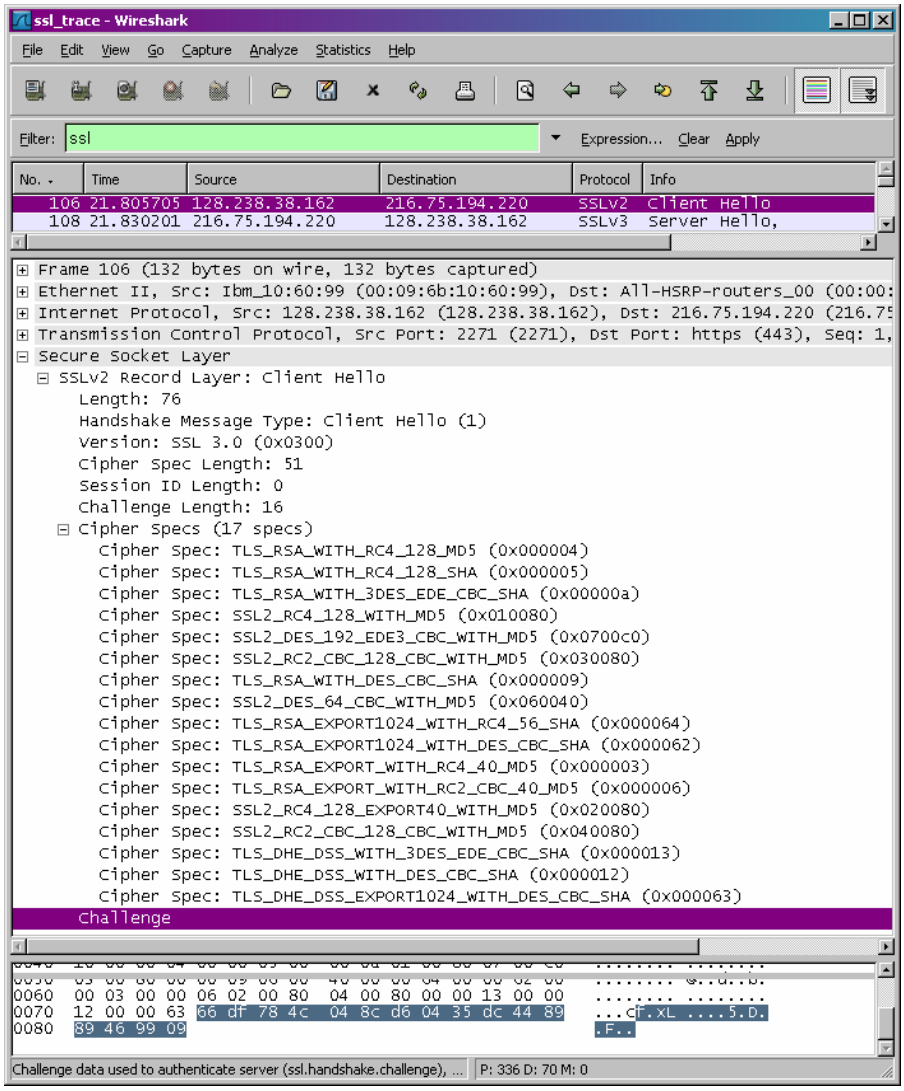

Fig. 3: Client Hello record

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?

   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?

   The client hello challenge is 66df 784c 048c d604 35dc 4489 8946 9909

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?

   The first listed suite uses RSA for public key crypto, RC4 for the symmetric-key cipher and uses the MD5 hash algorithm.

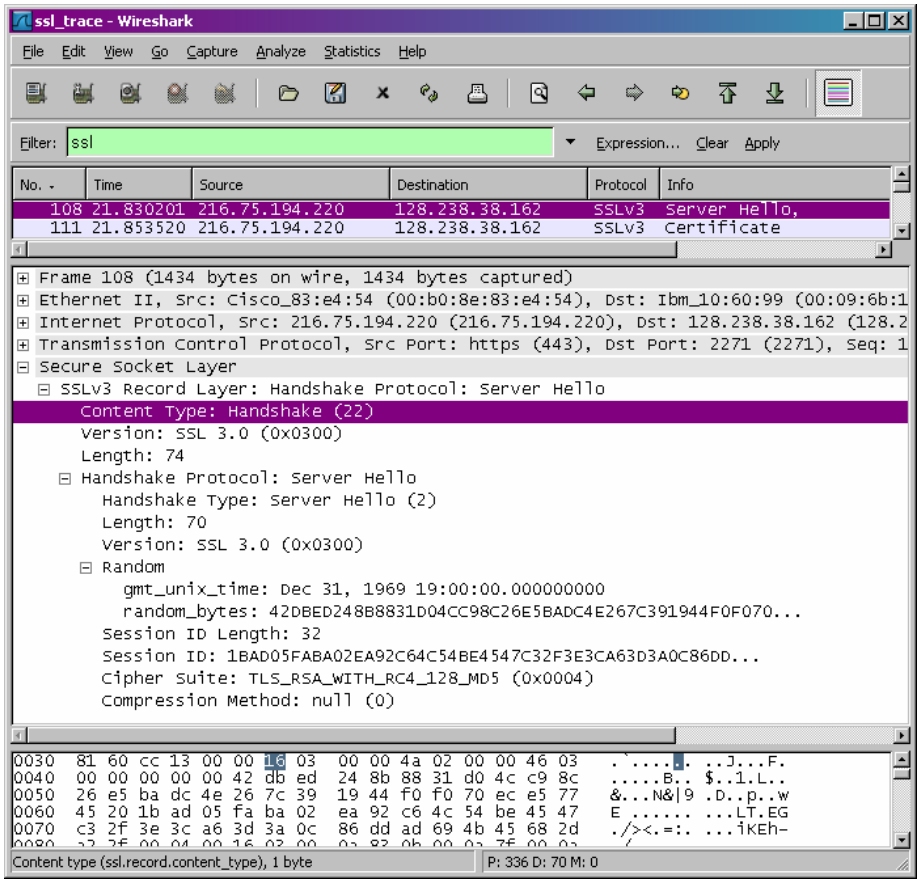

Fig. 3: Server Hello record

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

   The cipher suite uses RSA for public key crypto, RC4 for the symmetric-key cipher and uses the MD5 hash algorithm.

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?

   Yes, this record does include a nonce listed under Random. The nonce is 32 bits long, 28 for data plus 4 for the time. The purpose of the client and server nonces in SSL are to prevent replay attacks.

8. Does this record include a session ID? What is the purpose of the session ID?

   Yes, there is a session ID included. The purpose of the session ID is to provide a unique persistent identifier for the SSL session, which is sent in the clear. The client may resume the same session later by using the server-provided session ID when it sends the ClientHello.

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?

   This record does not contain a certificate, it is included in a separate record. The certificate fits into a single Ethernet frame.

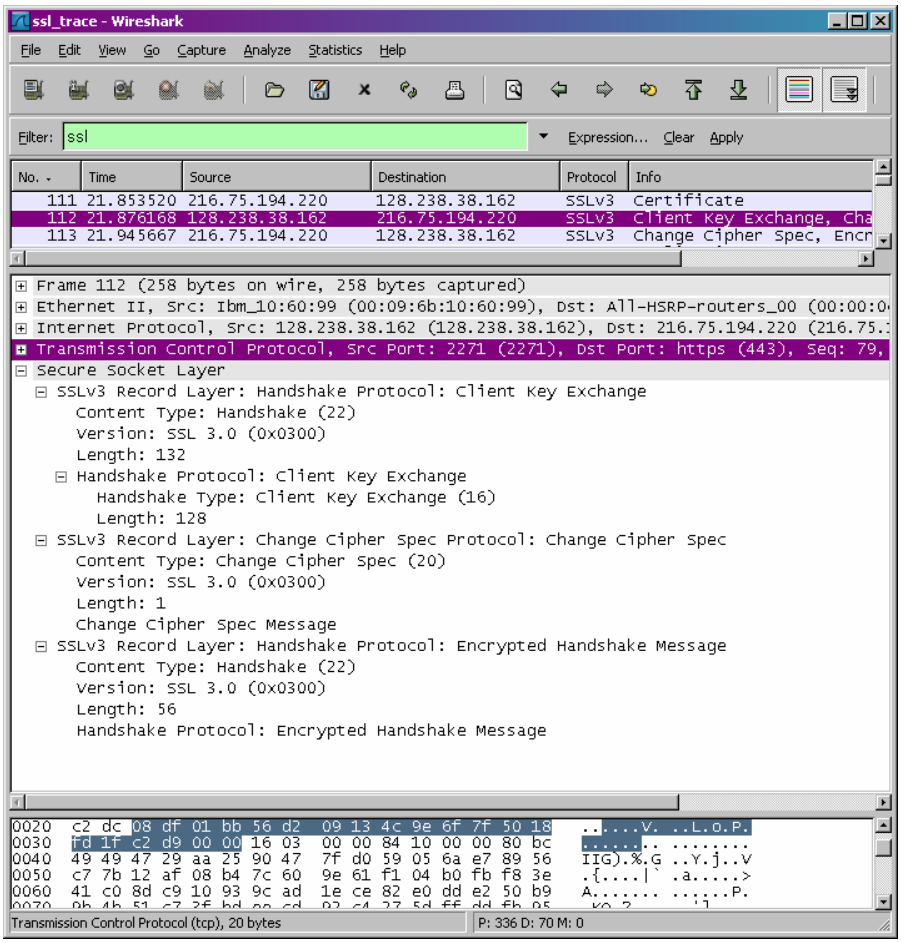

Fig. 4: Client key exchange, Change Cipher Spec & Encrypted Handshake records

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?

    Yes, this record contains a pre-master secret. The pre-master secret is used by both the server and client to produce a master secret, which is used to generate a set of session keys for MAC and encryption. The secret is encrypted using server's public key, which the client has extracted from the certificate record sent to them by the server. The encrypted secret is 128 bytes long.

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

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

12. In the encrypted handshake record, what is being encrypted? How?

    In the encrypted handshake record, a MAC of the concatenation of all previous handshake messages sent from this client is generated and 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?

    Yes, the server 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 are the same as those sent by the client.

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?

    The application data is encrypted using symmetric key encryption algorithm chosen in the handshake phase (in my case, RC4) using the symmetric encryption keys generated using the pre-master key and nonces (from both client and server). The client encryption key is used to encrypt the data being sent from client to server and the server encryption key is used to encrypt the data being sent from the server to the client.

    The record containing the application data does include a MAC; however, Wireshark does not distinguish between the encrypted application data and the MAC

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

    The initial ClientHello message uses SSLv2 (version 2). However, when the server replies with a frame using SSLv3 (version 3), the subsequent SSL message exchange is all in version 3 format.

    There are several times where the session is resumed. The handshake process here differs from the initial handshake detailed in Fig. 1. To resume the session, the client sends a ClientHello message including the SessionID first sent from the server to the client during the initial SSL handshake. The Server does not need to respond with a certificate, since the client already has it. Instead, a ServerHello containing a new nonce is sent, followed by Change Cipher Spec and Encrypted Handshake records from server to client. The client then responds with Change Cipher Spec and Encrypted Handshake records, then application data is sent.