**Question 1:**

For each of the first 8 Ethernet frames (after you apply SSL filter, i.e., frame 106), specify 1) the frame number, 2) frame source (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.

**Hint:** You may need to see the details of each frame.

|  |  |  |  |
| --- | --- | --- | --- |
| Frame No. | Source | No. of SSL Records | Record Type |

|  |  |  |  |
| --- | --- | --- | --- |
| **Frame No.** | **Source** | **No. of SSL Records** | **Record Type** |
| 106 | 128.238.38.162 | 1 | Client Hello (SSLv2) |
| 108 | 216.75.194.220 | 1 | Server Hello (SSLv3) |
| 111 | 216.75.194.220 | 2 | Certificate (SSLv3)  Server Hello Done (SSLv3) |
| 112 | 128.238.38.162 | 3 | Client Key Exchange (SSLv3)  Change Cipher Spec (SSLv3)  Encrypted Handshake Message (SSLv3) |
| 113 | 216.75.194.220 | 2 | Change Cipher Spec (SSLv3)  Encrypted Handshake Message (SSLv3) |
| 114 | 128.238.38.162 | 1 | http-over-tls (SSLv3) |
| 122 | 216.75.194.220 | 1 | http-over-tls (SSLv3) |
| 149 | 216.75.194.220 | 1 | http-over-tls (SSLv3) |

**Question 2:**

Draw a timing diagram between client and server, with one arrow for each SSL record. (You can upload a picture of your drawing)

A piece of paper with writing on it

Description automatically generated with medium confidence

**Question 3:**

Answer the following questions about **ClientHello**:

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

**Hint:** Pick any ClientHello frame that you see, i.e., frame 106. You may need to print the raw information by adding option –T jsonraw to the command, which will print the details in JSON with some raw information in hexadecimal.

*Answer:*

Yes, it does contain Challenge.

In hexadecimal: 66 df 78 4c 04 8c d6 04 35 dc 44 89 89 46 99 09

**Question 4:**

Answer the following questions about **ClientHello**:

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:**

Yes, the Client Hello record advertise the Cyber Suits it supported.

Text

Description automatically generatedPublic-key Algorithm:

* RSA

Symmetric-key Algorithm:

* RC4

Hash Algorithm:

* MD5

**Question 5:**

Answer the following questions about **ServerHello**:

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

**Answer:**

Yes, the chosen cipher Suite is TLS\_RSA\_WITH\_RC4\_128\_MD5

Graphical user interface, text, application, email

Description automatically generated

**Question 6:**

Answer the following questions about **ServerHello**:

Does this **ServerHello** record include a nonce? If so, how long is it? What is the purpose of the client and server nonces in SSL?

**Answer text**

Yes:

42dbed248b8831d04cc98c26e5badc4e267c391944f0f070ece57745

28 bytes

To prevent replay attacks

**Question 7:**

Does this **ServerHello** record include a session ID? What is the purpose of the session ID?

Yes Session ID: 1bad05faba02ea92c64c54be4547c32f3e3ca63d3a0c86ddad694b45682da22f

To identify the nonce and create a unique session that will be thrown away for freshness. (check!)

**Question 8:**

Does this **ServerHello** record contain a certificate, or is the certificate included in a separate record. Does the certificate fit into a single Ethernet frame?

No it doesn’t contain a certificate, the certificate is included in a separate record. The total size of the SSL record containing the certificate is 2696 byte which is larger than the ethernet frame.

**Question 9:**

Answer the following questions about encryption:

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. It is encrypted and has a length of 128 bytes. The pre master secret is created by the client from the cipher suite and is encrypted during the handshake. The server decryptes and obtains a shared master secret which will be used for symmetric encryption for the rest of communication as (RSA and asymmetric is expensive).

**Question 10:**

What is the purpose of the **Change Cipher Spec** **record** (sent by client)? How many bytes is the record in your trace?

The purpose is to authenticate the client by acknowledging the chosen cipher suite from the previous Server hello packet.

**Question 11:**

In the **Encrypted handshake record**, what is being encrypted? How?

The handshake is being encrypted. We know that it must be following the cipher suite: RSA\_WITH\_RC4\_128\_MD5 (0x0004)

**Question 12:**

Does the **Server** also send a **Change cipher record** and an **Encrypted hand shake record** to the client? How are those records different from those sent by the client?

Yes. The purpose of the Serve change cipher record is to define the re-negotiated cipher spec and keys that will be used for all the messages exchanged henceforth. The purpose of the client’s change cipher record is to acknowledge that it has updated its suite to match. (Double check this)

**Question 13:**

How is the Application Data being encrypted? Do the records containing application data include a MAC? Does TShark distinguish between the encrypted application data and the MAC?

The Data is being encrypted by a session key from the cipher suite. The record does indeed include a mac, however the mac is encrypted together with the data payload, hence Tshark does not distinguish between them.