Wireshark Experiment – 04

Transport Layer

Name: Pratik P Patil
USN: 02FE21BEC063
Roll No: 53
Div: A

Transport Layer

The Transport Layer is the fourth layer of the OSI (Open Systems Interconnection) model and plays a crucial role in end-to-end communication between devices in a network. Its primary purpose is to provide reliable data transfer, error recovery, and flow control, ensuring that data is delivered accurately and efficiently between applications on different devices.

Key Functions of the Transport Layer

- Breaks down large data streams into smaller segments for transmission.
- Provides direct communication between applications on different devices.
- stablishes a reliable connection before data transfer and ensures data integrity.
- Ensures data integrity by detecting and correcting errors during transmission.
- Regulates the data flow to prevent the sender from overwhelming the receiver.
- Allows multiple applications to use the network simultaneously by distinguishing between them using port numbers.

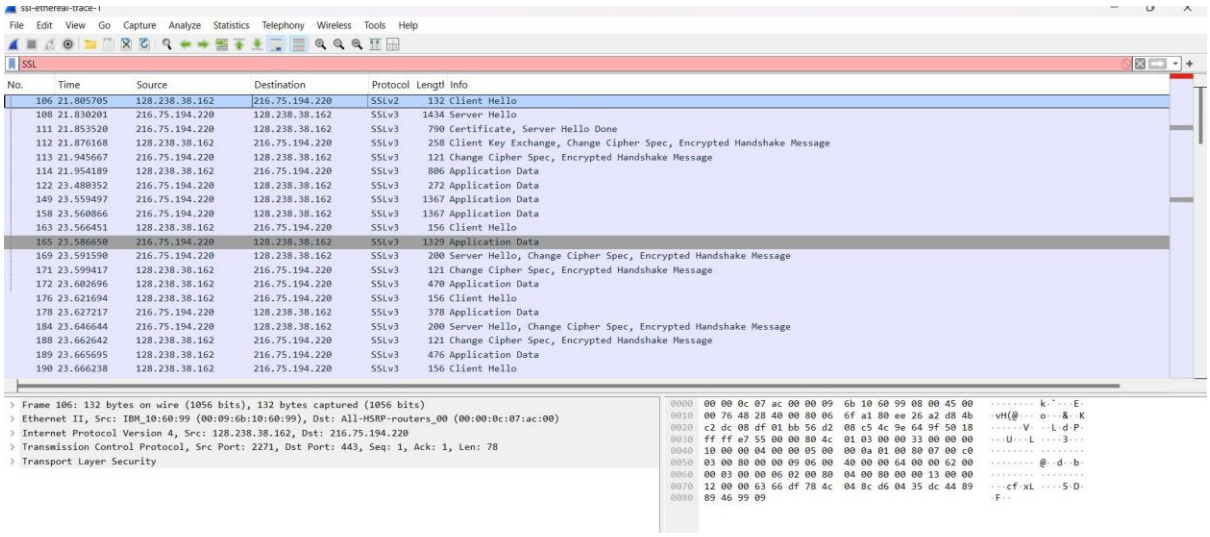
Protocols in the Transport Layer

1. **Transmission Control Protocol (TCP)**:
    - Connection-oriented protocol.
    - Provides reliable data delivery, error checking, and retransmission.
    - Used in applications like web browsing (HTTP), email (SMTP), and file transfers (FTP).

2. **User Datagram Protocol (UDP)**:
    - Connectionless protocol.
    - Focuses on low latency and minimal overhead.
    - Commonly used in applications like video streaming, online gaming, and VoIP.

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.

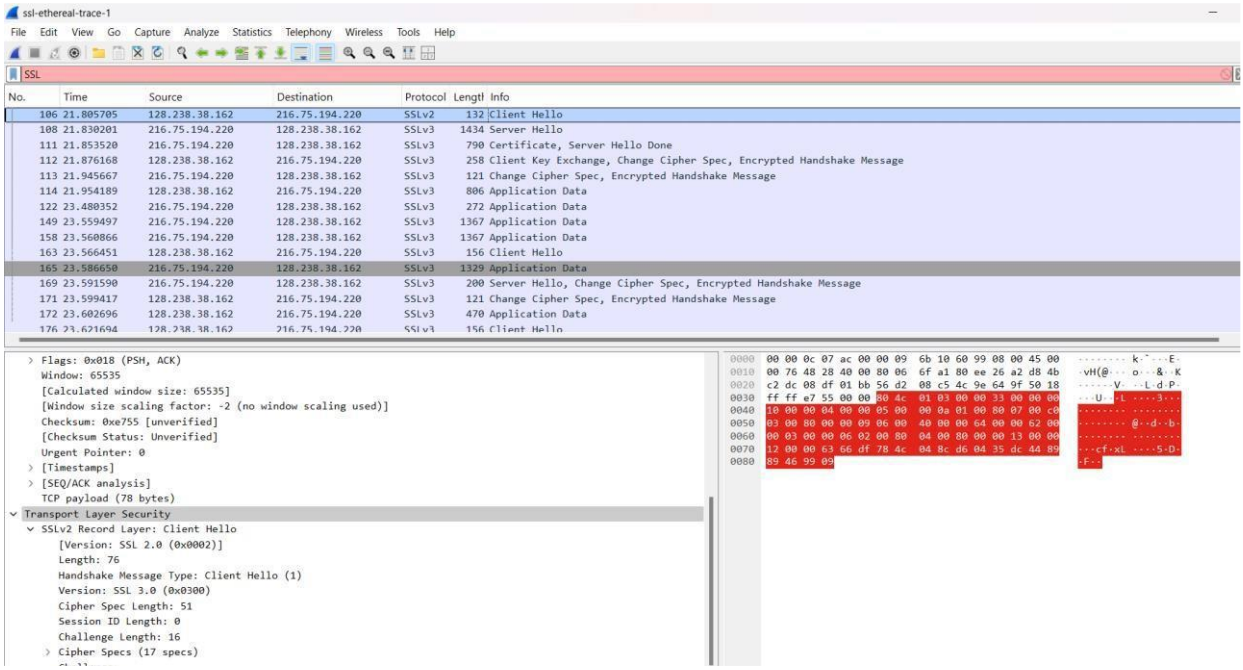
ANS:

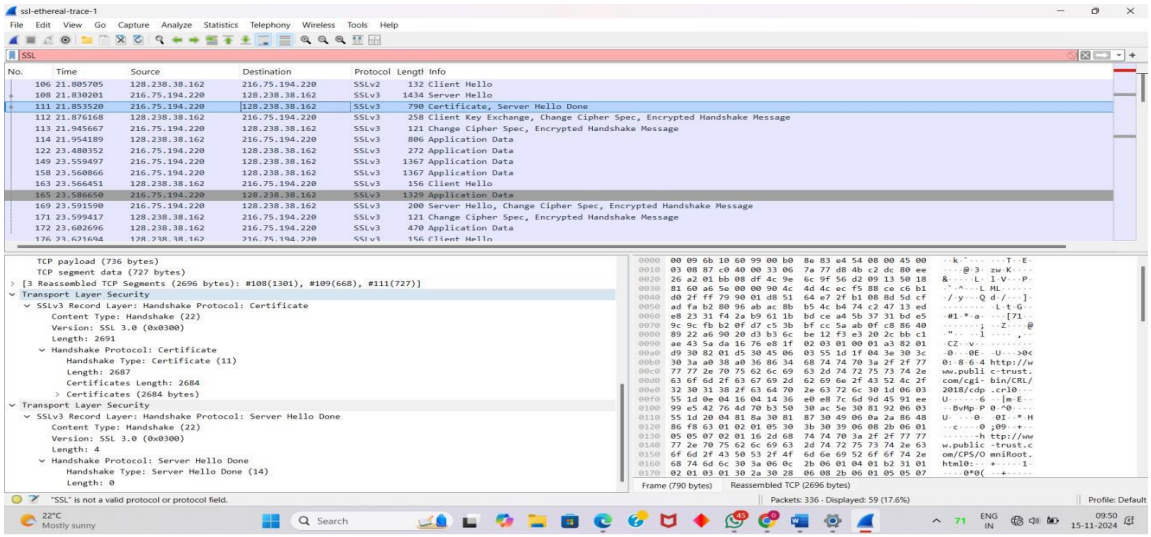


First 8 Frames

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.

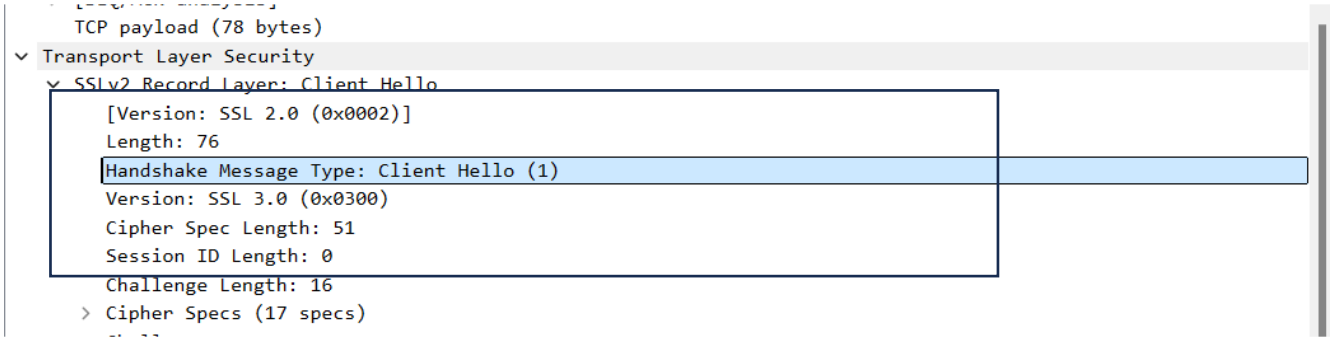
ANS:





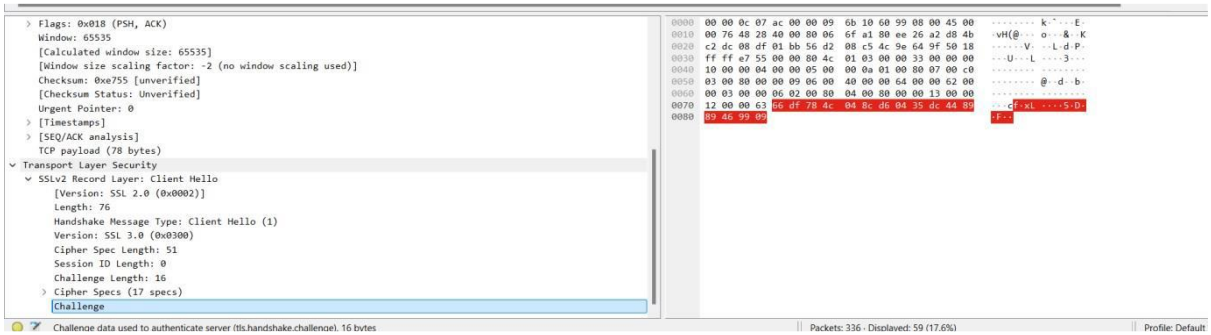
3.Expand the Client Hello record. (If your trace contains multiple ClientHello records, expand the frame that contains the first one.) What is the value of the content type?

ANS:



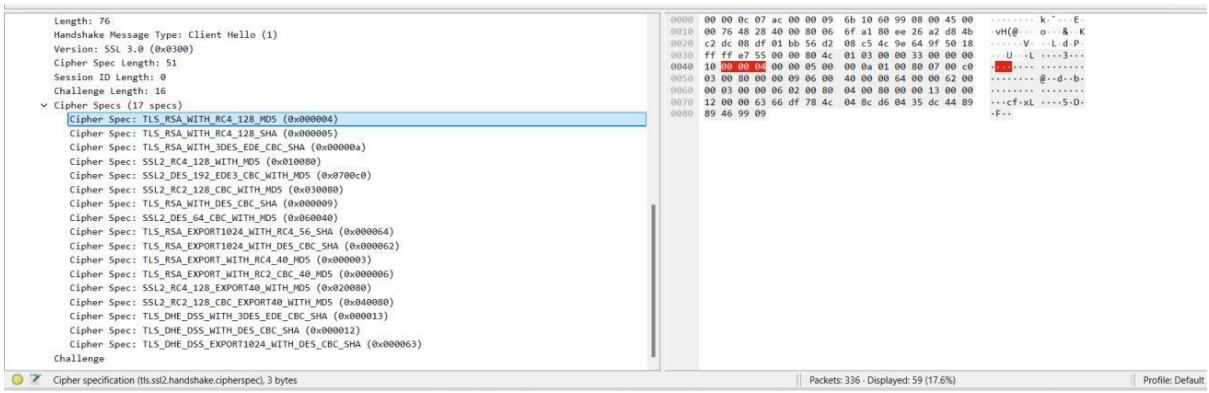
4.Does the Client Hello record contain a nonce (also known as a “challenge”)? If so, what is the value of the challenge in hexadecimal notation?

ANS:



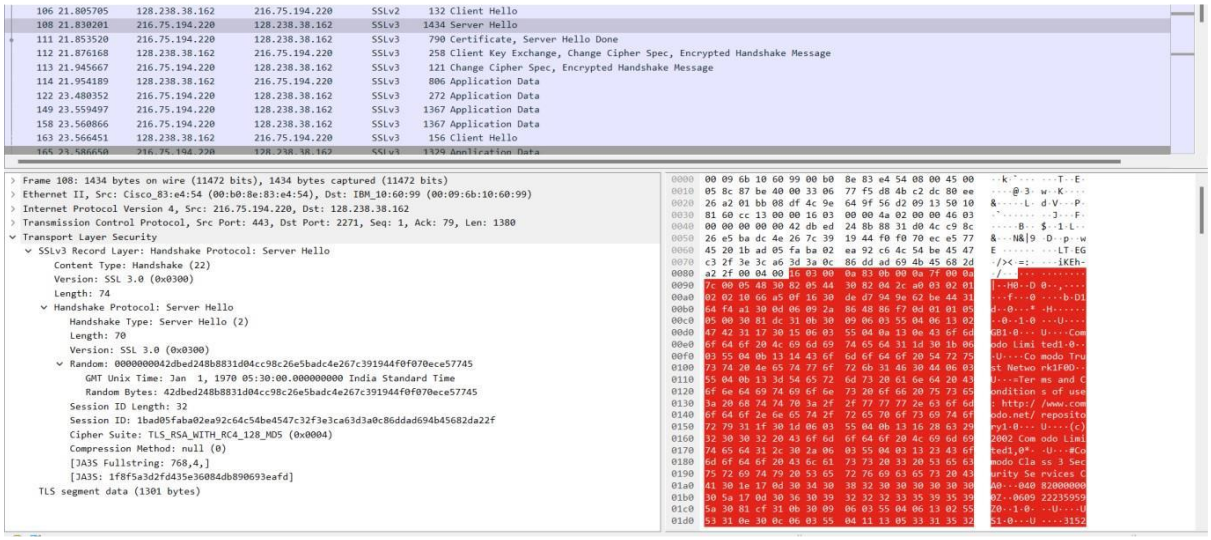
5.Does the Client Hello record advertise the cyber suites it supports? If so, in the first listed suite, what are the public-key algorithm, the symmetric-key algorithm, and the hash algorithm?

ANS:



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

ANS:



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 includes a nonce, as known as Random.

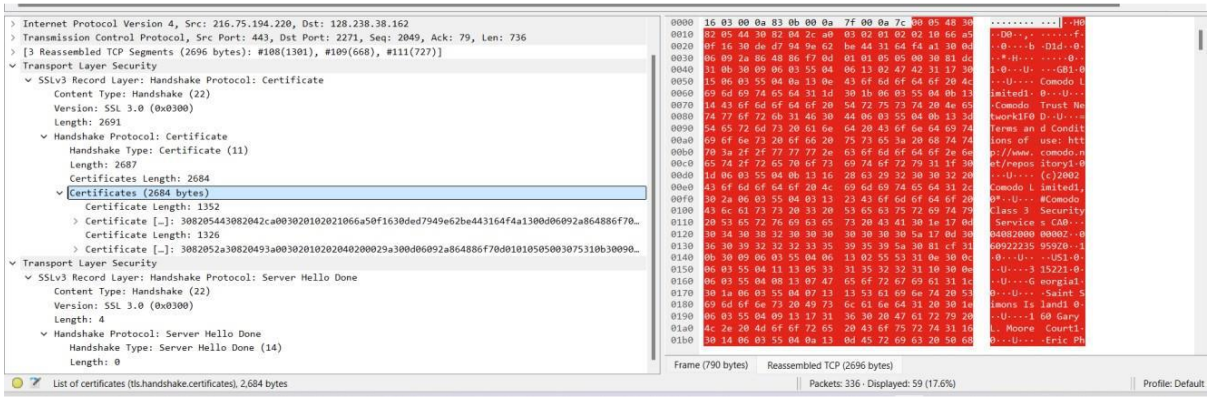
bytes, and it is 28 bytes long (as highlighted above). The purpose of the client and server nonces in SSL is to prevent attacker from replaying or reordering records.

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

ANS:Yes, this record includes a Session ID which is 32-bytes long. Its purpose is to allow session resumption, which can significantly reduce the number of time-consuming server handshake to crease a new session ID. In the Client Hello record, a nonzero session ID means that the client to resume its previously established session; and a zero session ID means that the client wishes to establish a new session with the server.

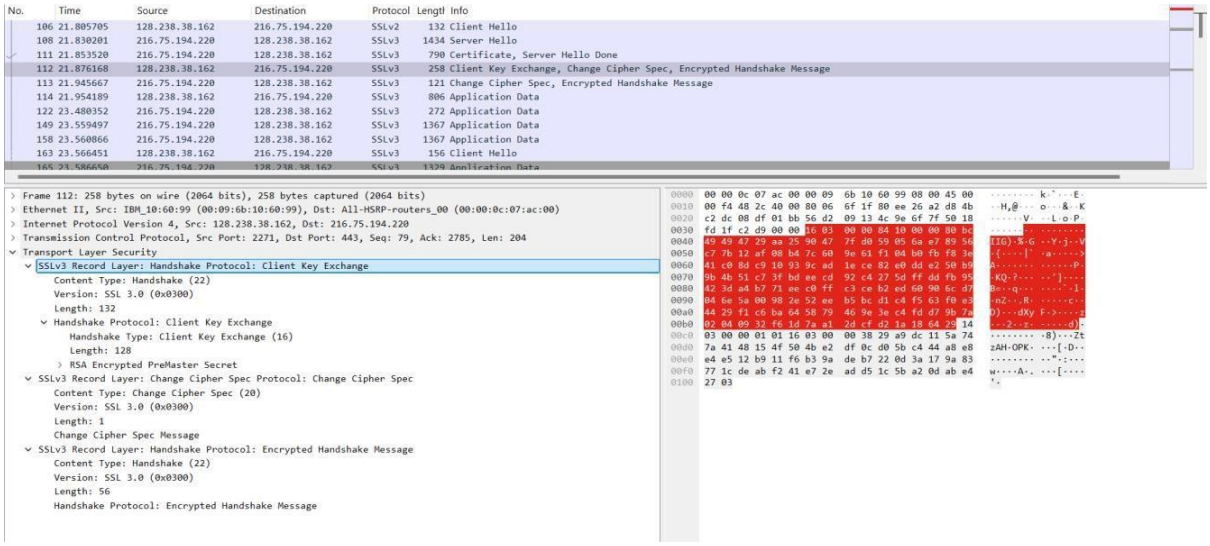
9.Does this record contain a certificate, or is the certificate included in a separate record. Does the certificate fit into a single Ethernet frame?

Yes, this record contains a certificate. The certificate is 2684 bytes long, thus it can 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? Change Cipher Spec Record (sent by client) and Encrypted

ANS:Yes, this record contains a pre-master secret (highlighted above). This encrypted pre-master secret is decrypted at the server side and is used to produce a master secret. Then this master secret is used to produces “key block”, which is then sliced and diced into client MAC key, server MAC key, client encryption key, server encryption key, client IV and serve IV. The secret is encrypted using server’s public key. The encrypted secret is 128 byte long.



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

ANS: The purpose of Change Cipher Spec is to indicate change in encryption and authentication algorithms and to update the cipher suite to be used on this connection. This record is only 1 byte long in my trace.

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

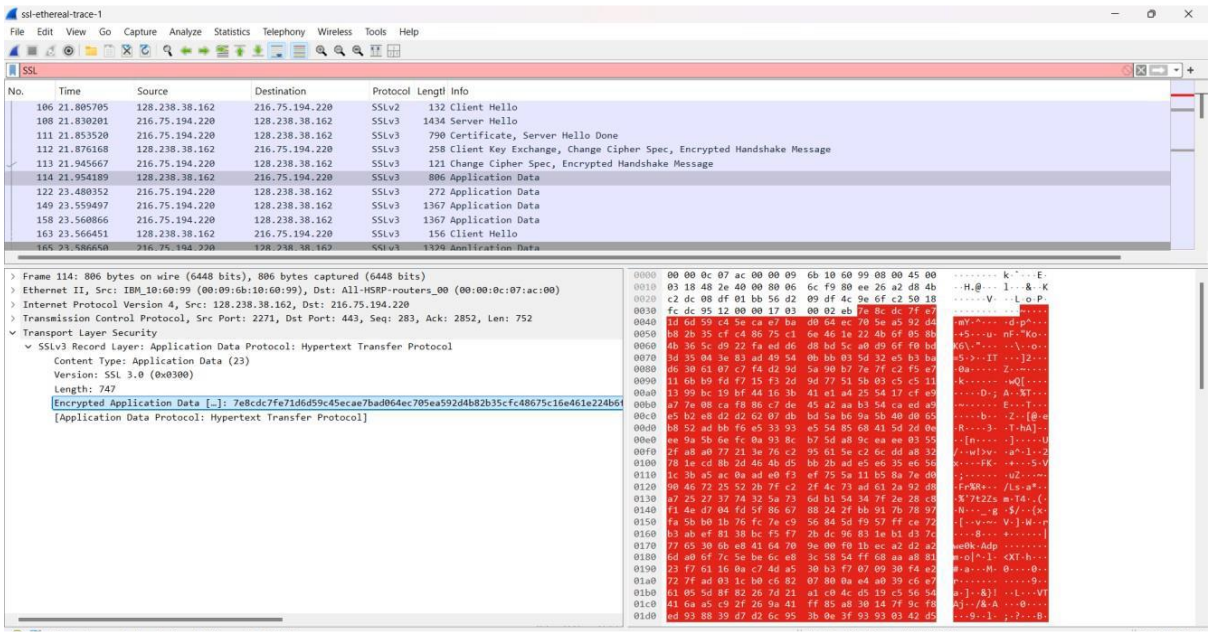
ANS:The sender of this Encrypted Handshake Records and all handshake messages up to but not including this message are encrypted in record. This information is concatenated and hashed using two hash algorithms, MD5 and SHA. The content of this record is the concatenation of these two hash values. The Encrypted Handshake Record is used to verify that key exchange and authentication processes were successful.

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? Application Data

Yes, the server also sends its own Change Cipher Spec and Encrypted Handshake records. The only difference is the sender of this record; the sender is now the server while the sender was the client in previous message.

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?

ANS:



The application data is encrypted using the specified algorithms in the chosen cipher suite; in my case, RSA (public-key), 256-bit CBC AES (symmetric)