Cryptography And Network Security Lab

Assignment submission

PRN No: 2019BTECS00017

Full name: Muskan Raju Attar

Batch: B5

Assignment: 16

Title of assignment: SSL/TLS Handshake Analysis using Wireshark

Title:

SSL/TLS Handshake Analysis using Wireshark

Aim:

To observe SSL/TLS (Secure Sockets Layer/ Transport Layer Security)in action. SSL/TLS is used to secure TCP connections, and it is widely used as part of the secure web: HTTPS is SSL over HTTP

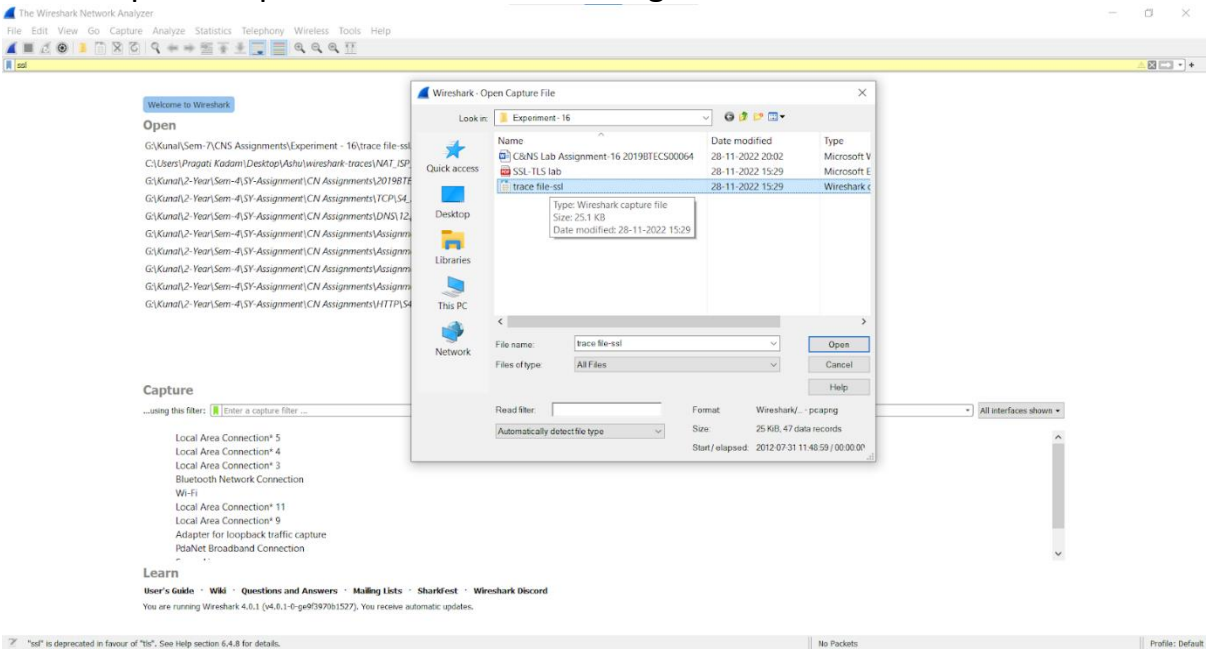
Theory:

- Wireshark is a free and open-source packet analyzer.
- It is used for network troubleshooting, analysis, software and communications protocol development, and education.
- Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues.
- Wireshark is cross-platform, using the Qt widget toolkit in current releases to implement its user interface, and using pcap to capture packets; it runs on Linux, macOS, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows.
- There is also a terminal-based (non-GUI) version called TShark. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of the GNU General Public License version 2 or any later version.

Use of Wireshark

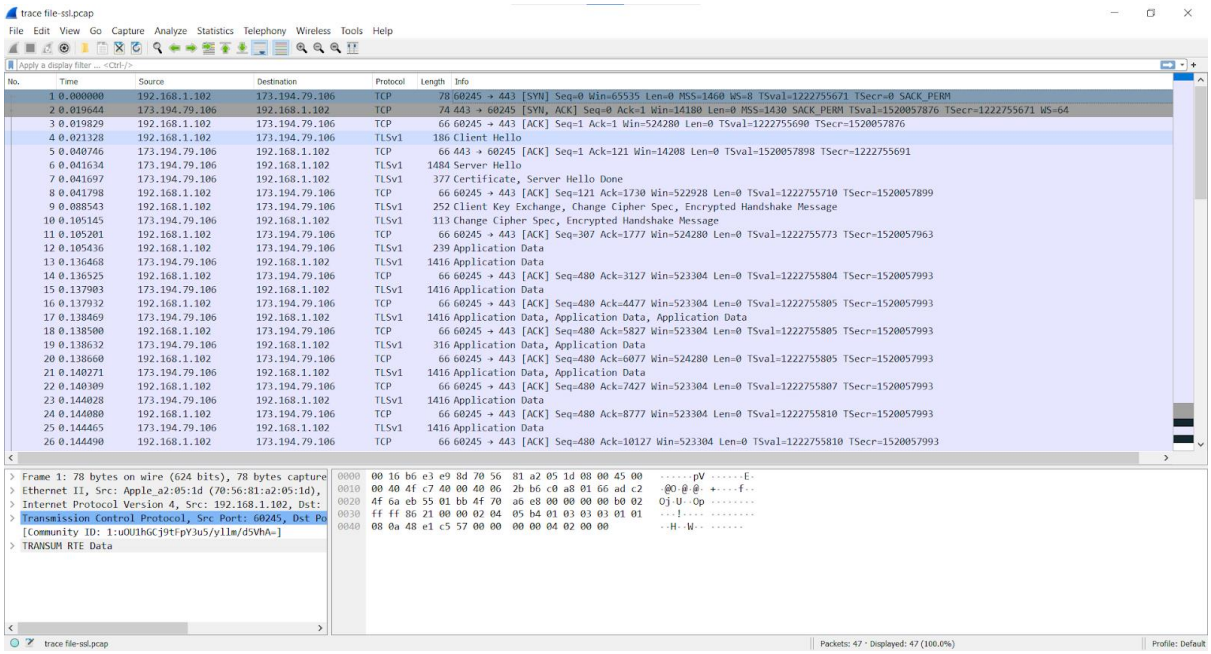
Step 1: Open a Trace you should use a supplied trace file trace-ssl.pcap.

File → Open → open from folder containing file

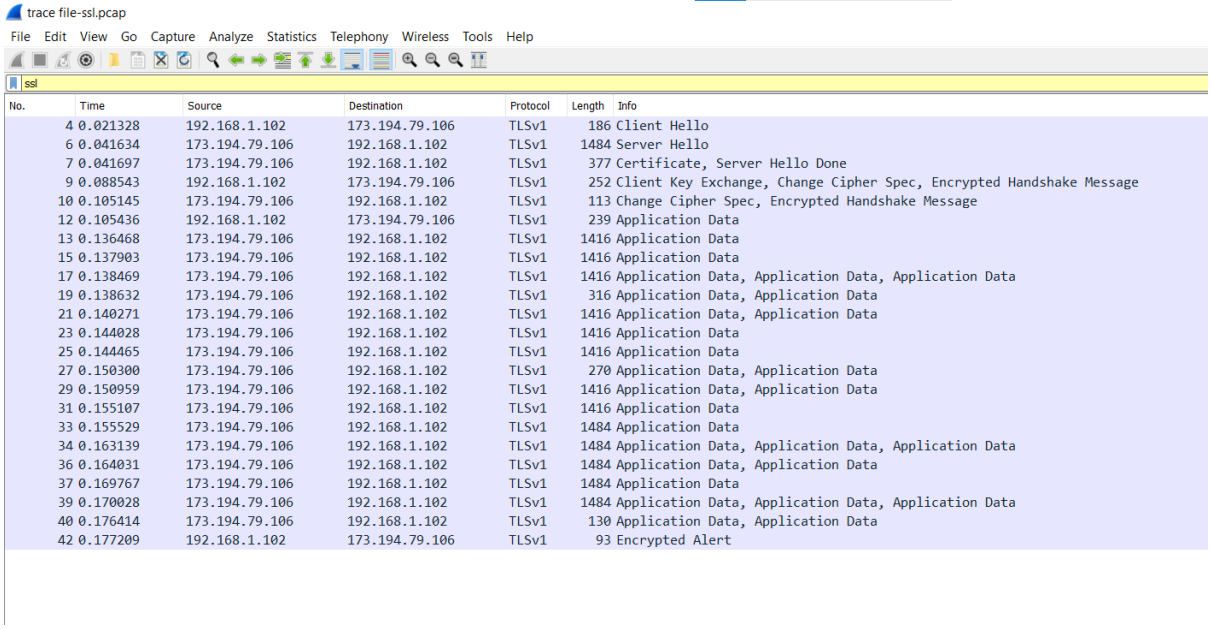


Step 2: Inspect the Trace

Now we are ready to look at the details of some SSL messages. To begin, enter and apply a display filter of ssl. This filter will help to simplify the display by showing only SSL and TLS messages. It will exclude other TCP segments that are part of the trace, such as Acks and connection open/close. Select a TLS message somewhere in the middle of your trace for which the Info field reads Application Data, and expand its Secure Sockets Layer block(by using triangular icon on left side). Application Data is a generic TLS message carrying contents for the application, such as the web page. It is a good place for us to start looking at TLS messages. Look for the following protocol blocks and fields in the message



Applying SSL Filter



- The lower layer protocol blocks are TCP and IP because SSL runs on top of TCP/IP.]

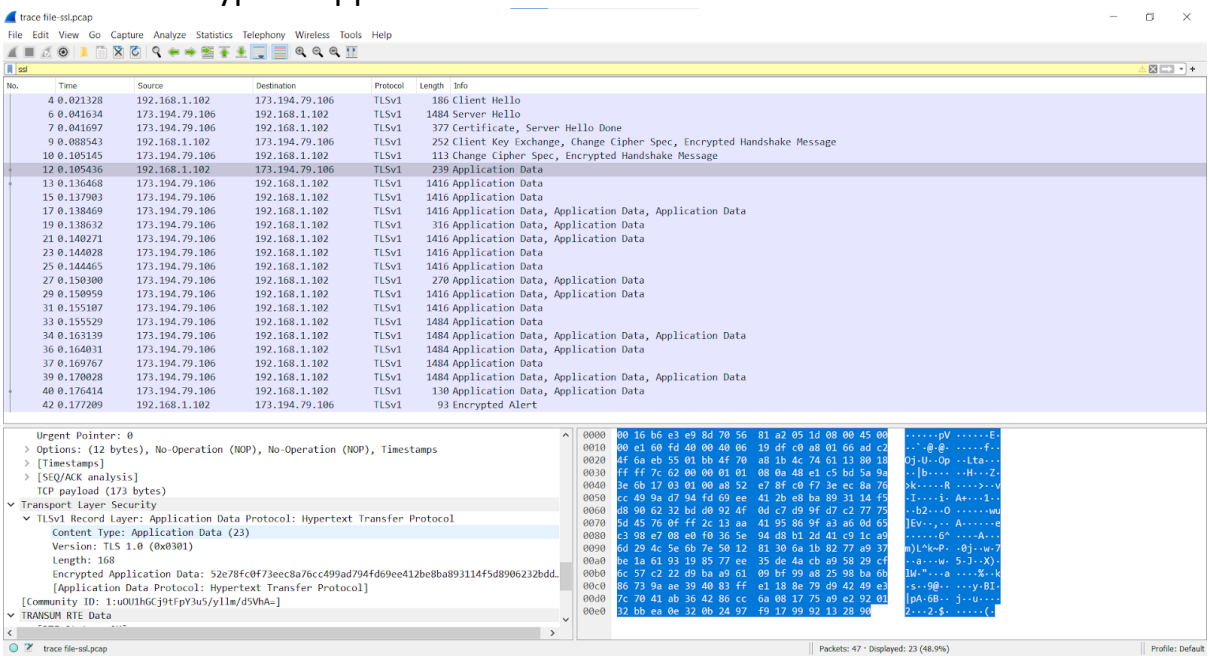
- The SSL layer contains a TLS Record Layer. This is the foundational sublayer for TLS. All messages contain records. Expand this block to see its details.
- Each record starts with a Content Type field. This tells us what is in the contents of the record. Then comes a Version identifier.It will be a constant value for the SSL connection.
- It is followed by a Length field giving the length of the record.
  Last comes the contents of the record. Application Data records are sent after SSL has secured the connection, so the contents will show up as encrypted data.

Note that, unlike other protocols we will see such as DNS, there may be multiple records in a single message. Each record will show up as its own block. Look at the Info column, and you will see messages with more than one block.

1. What is the Content Type for a record containing Application Data?

Ans:

The Content Type is Application Data.

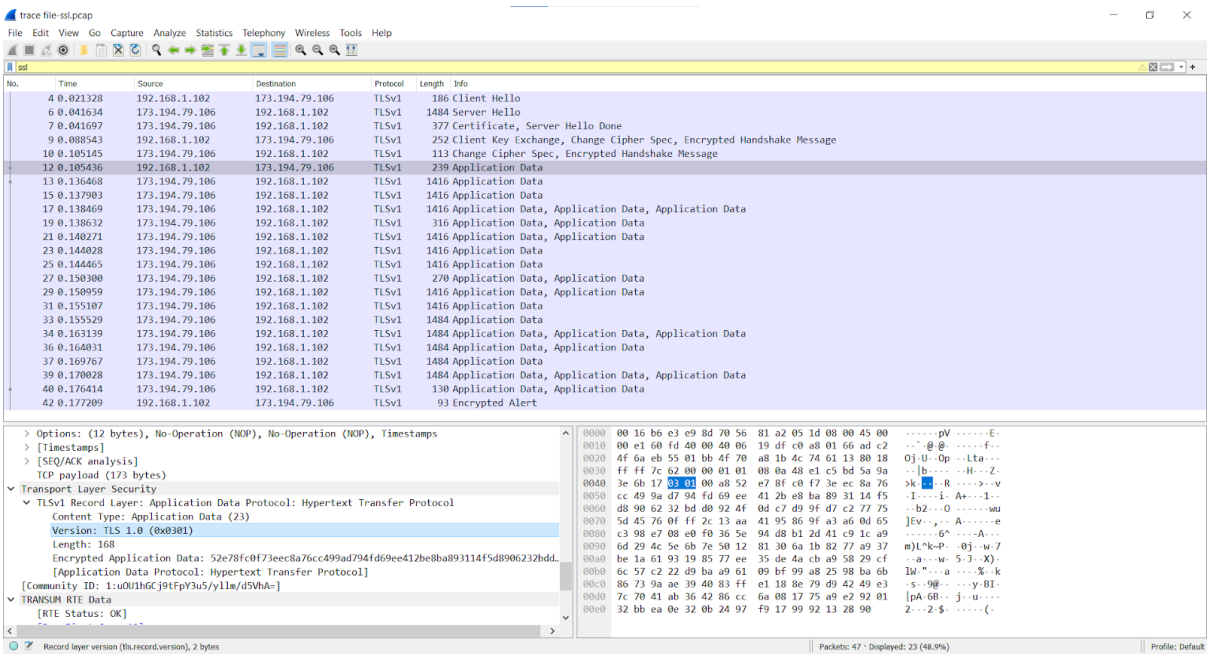


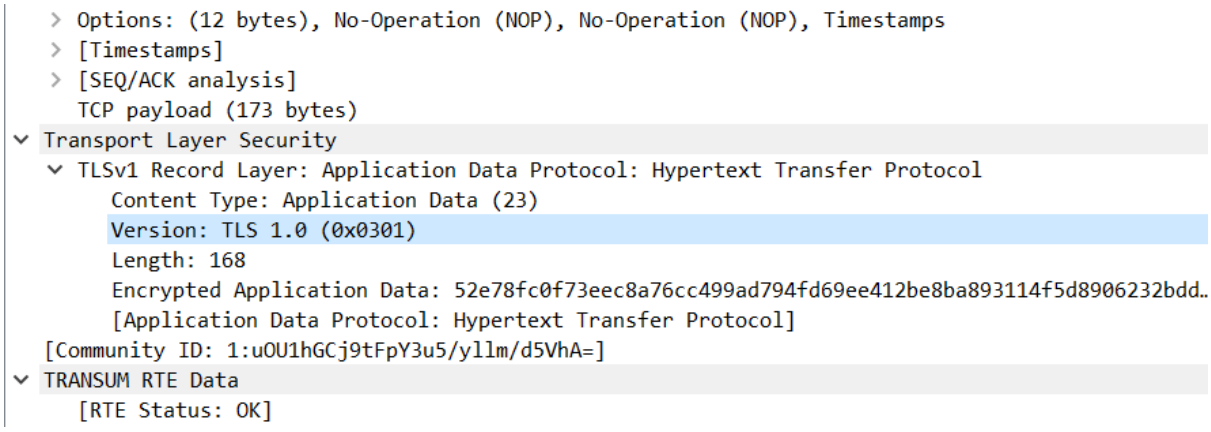
```
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (173 bytes)
v Transport Layer Security
  v TLSv1 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 168
      Encrypted Application Data: 52e78fc0f73eec8a76cc499ad794fd69ee412be8ba893114f5d8906232bdd…
      [Application Data Protocol: Hypertext Transfer Protocol]
  [Community ID: 1:uOU1hGCj9tFpY3u5/yllm/d5VhA=]
v TRANSUM RTE Data
```

2. What version constant is used in your trace, and which version of TLS does it represent?

Ans:

The version of TLS used is 1.0





Step 3: SSL Handshake

An important part of SSL is the initial handshake that establishes a secure connection. The handshake proceeds in several phases. There are slight differences for different versions of TLS and depending on the encryption scheme that is in use. The usual outline for a brand new connection is:

- Client (the browser) and Server(the web server) both send their Hellos
- Server sends its certificate to Client to authenticate (and optionally asks for Client Certificate)
- Client sends keying information and signals a switch to encrypted data.
- Server signals a switch to encrypted data.
- Both Client and Server send encrypted data.
- An Alert is used to tell the other party that the connection is closing. Note that there is also a mechanism to resume sessions for repeat connections between the same client and server to skip most of steps b and c.

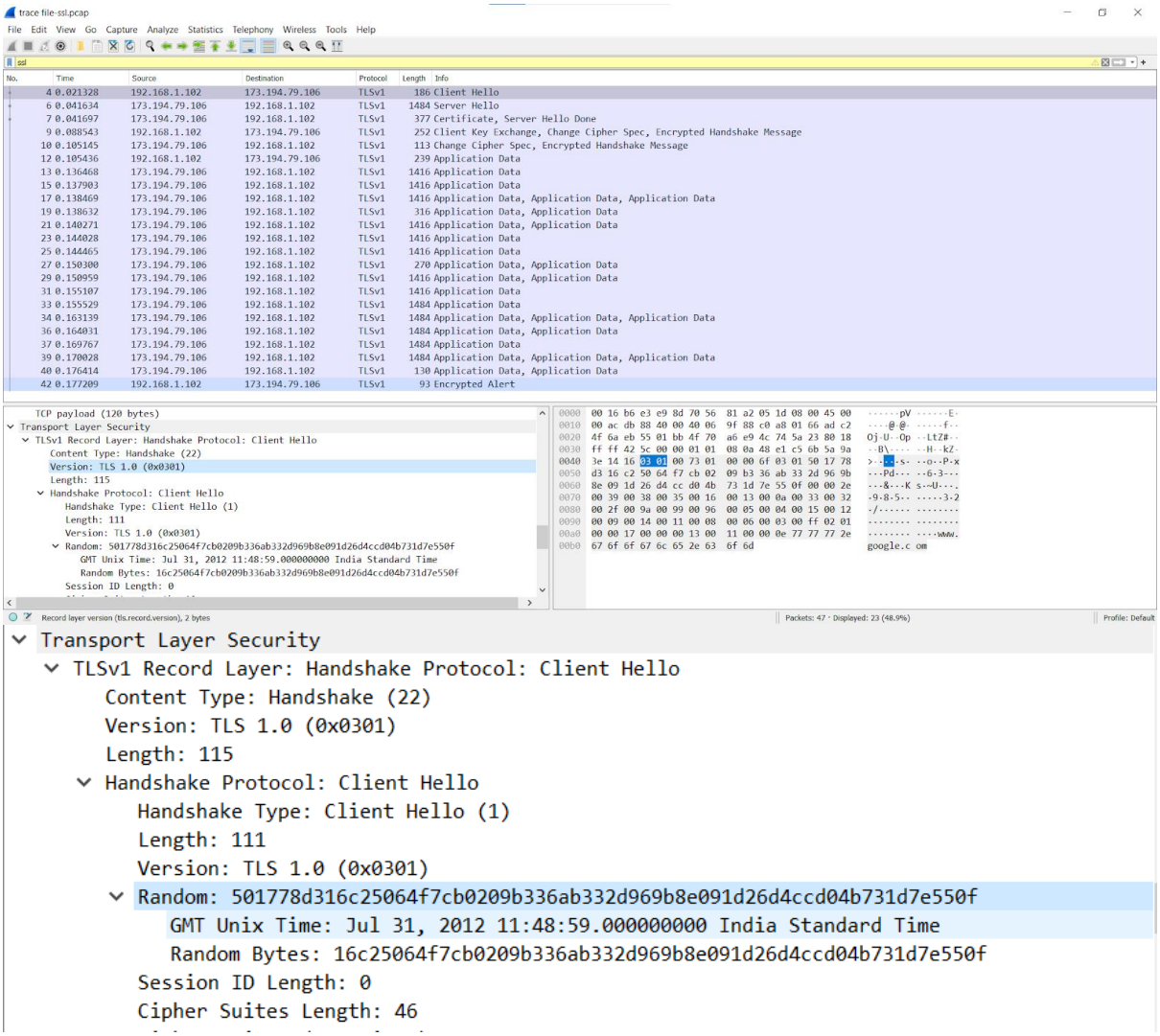
Hello Message

Find and inspect the details of the Client Hello and Server Hello messages, including expanding the Hand- shake protocol block within the TLS Record. For these initial messages, an encryption scheme is not yet established so the contents of the record are visible to us. They contain details of the secure connection setup in a Handshake protocol format.

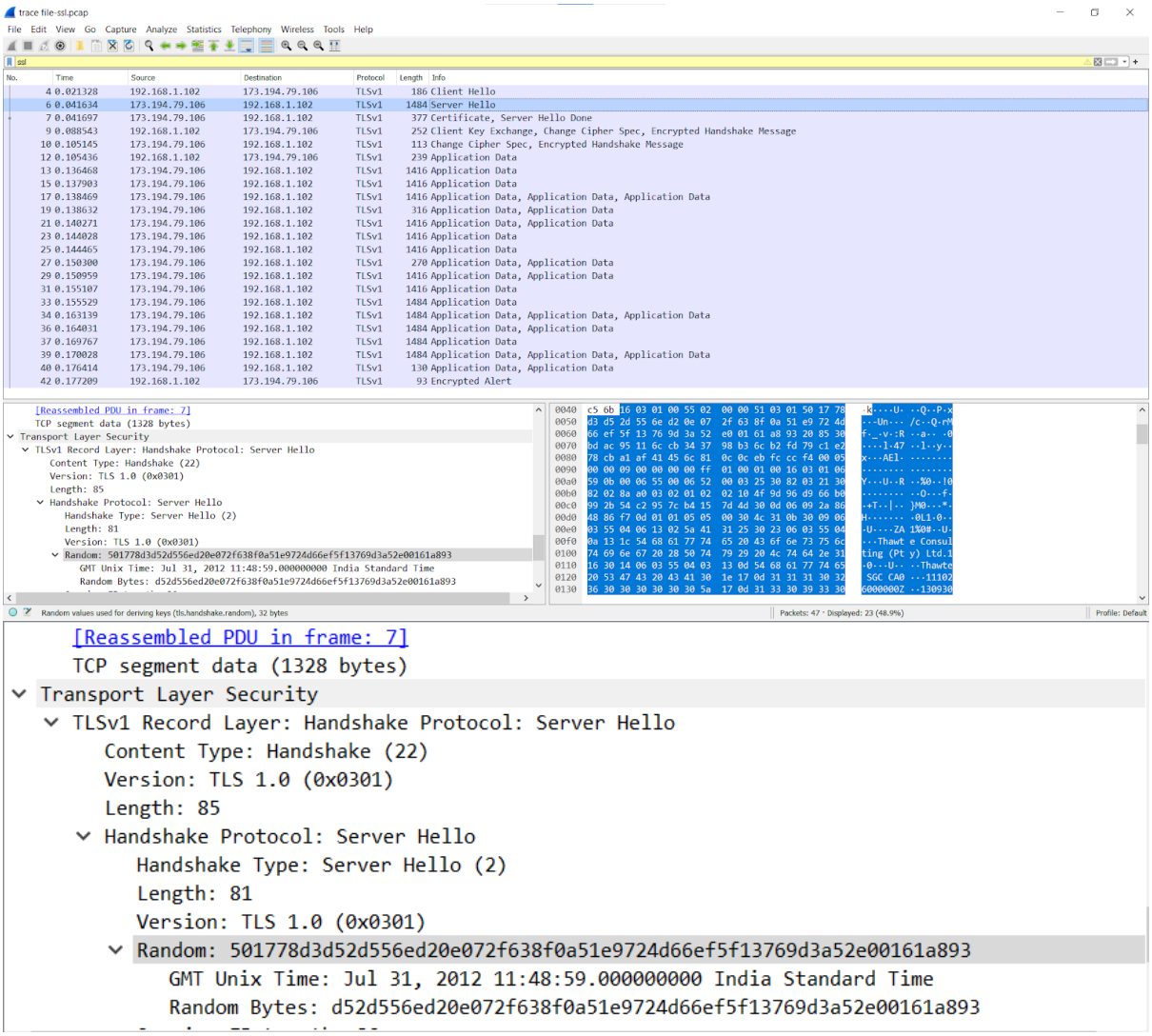
1. How long in bytes is the random data in the Hellos? Both the Client and Server include this random data (a nonce) to allow the establishment of session keys.

Ans:

Client:



Server:



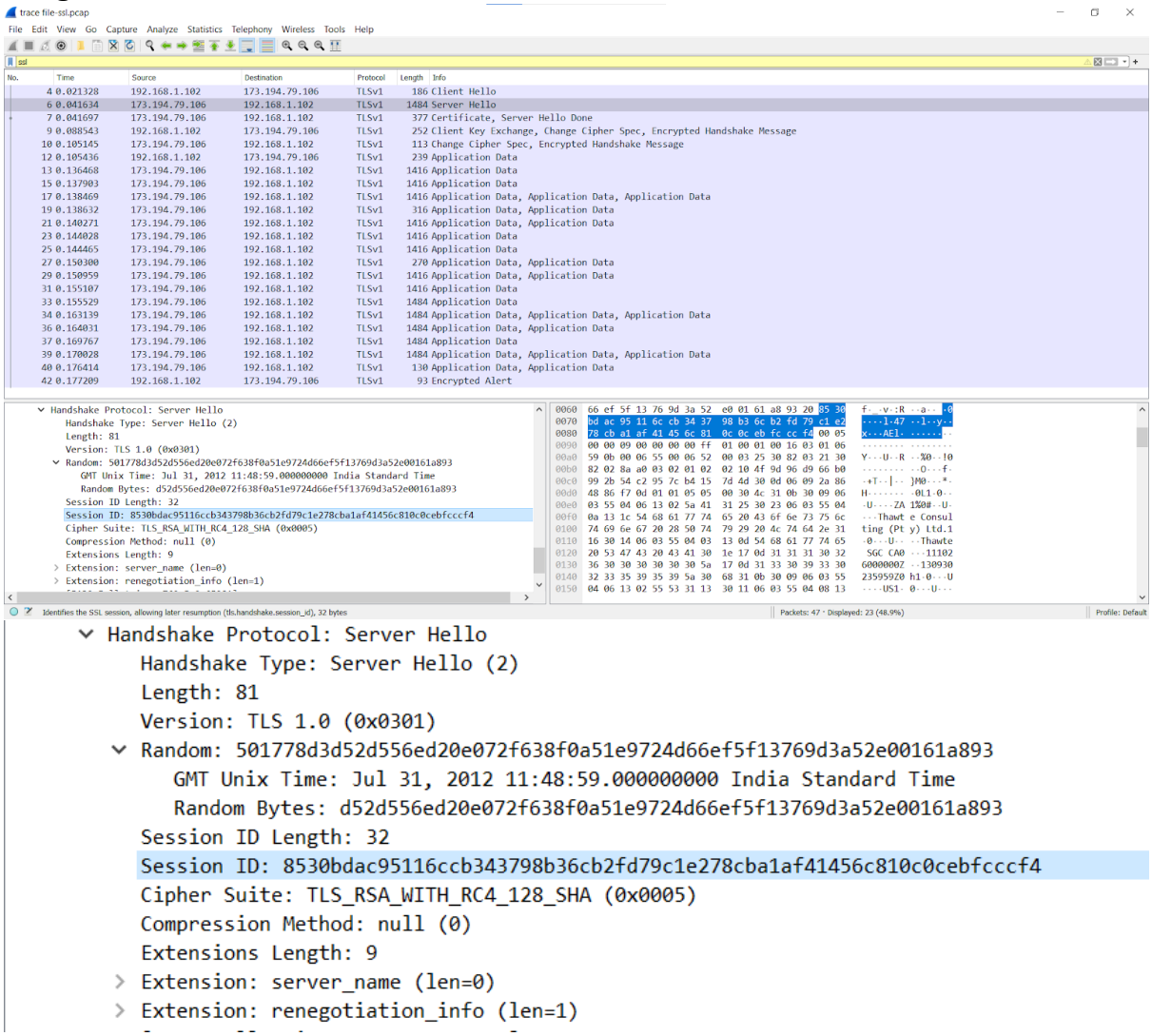
2. How long in bytes is the session identifier sent by the server?This identifier allows later resumption of the session with an abbreviated

handshake when both the client and server indicate the same value. In our case, the client likely sent no session ID as there was nothing to resume.

Ans:

Server:

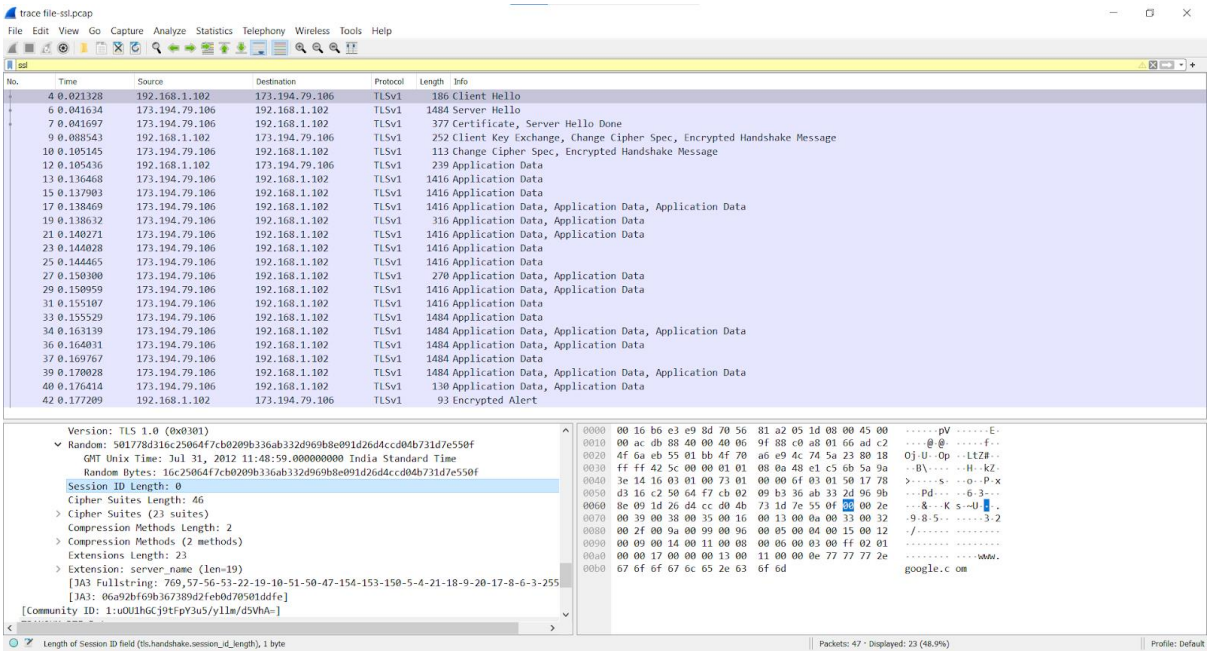
Length if Session ID is 32



Handshake Protocol: Server Hello
Handshake Type: Server Hello (2)
Length: 81
Version: TLS 1.0 (0x0301)
Random: 501778d3d52d556ed20e072f638f0a51e9724d66ef5f13769d3a52e00161a893
GMT Unix Time: Jul 31, 2012 11:48:59.000000000 India Standard Time
Random Bytes: d52d556ed20e072f638f0a51e9724d66ef5f13769d3a52e00161a893
Session ID Length: 32
Session ID: 8530bdac95116ccb343798b36cb2fd79c1e278cba1af41456c810c0cebfcccf4
Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
Compression Method: null (0)
Extensions Length: 9
Extension: server_name (len=0)
Extension: renegotiation_info (len=1)

Client:

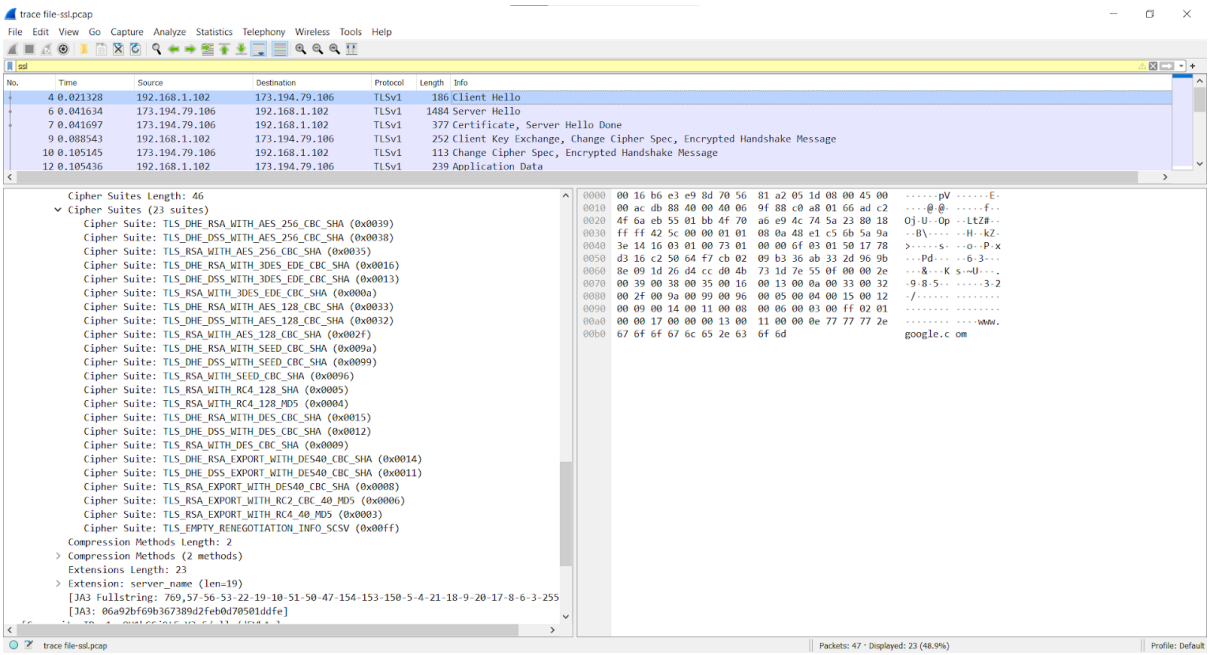
Length of Session ID is 0



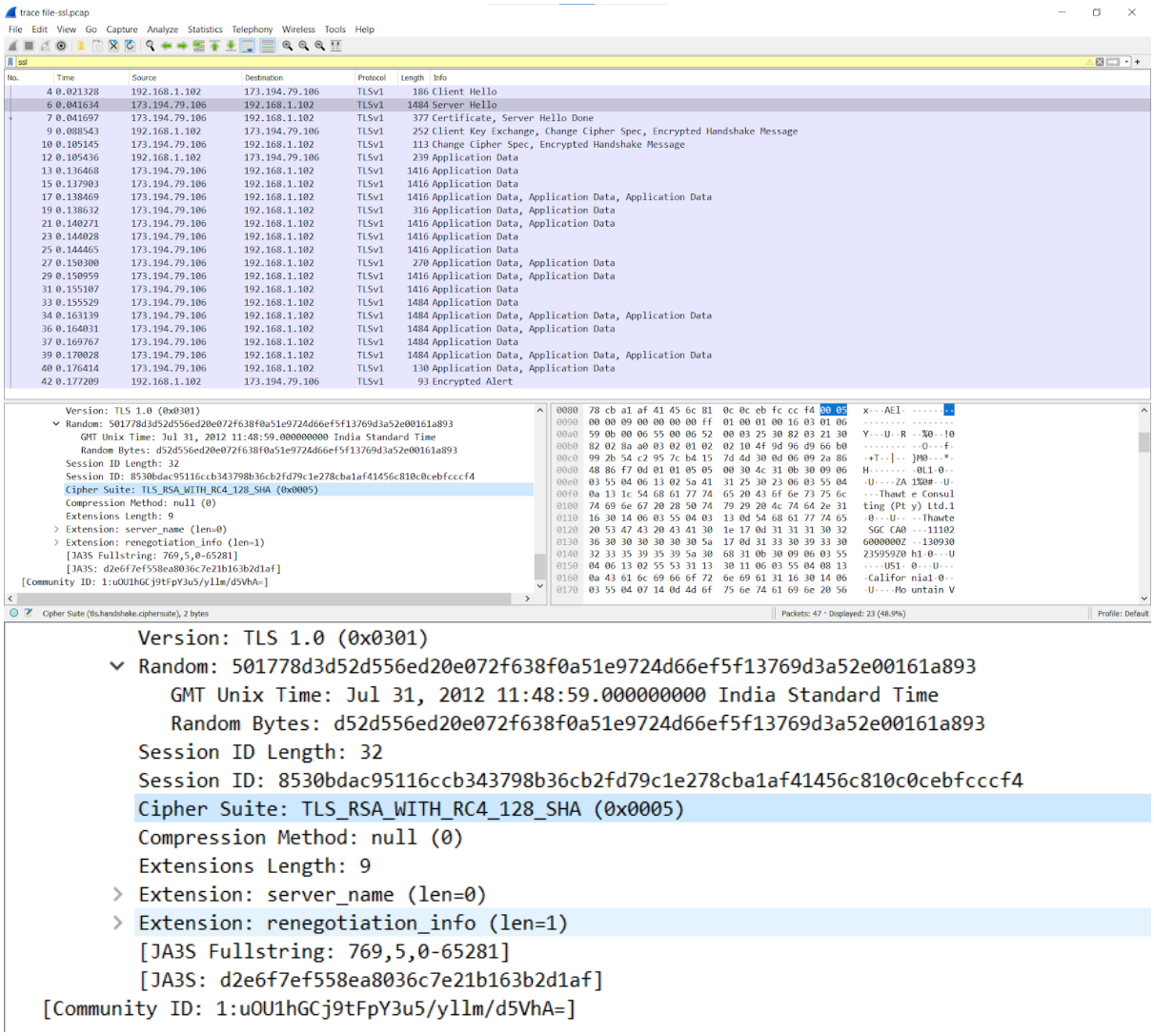
3. What Cipher suite is chosen by the Server? Give its name and value. The Client will list the different cipher methods it supports, and the Server will pick one of these methods to use.

Ans:

Client:



Server:



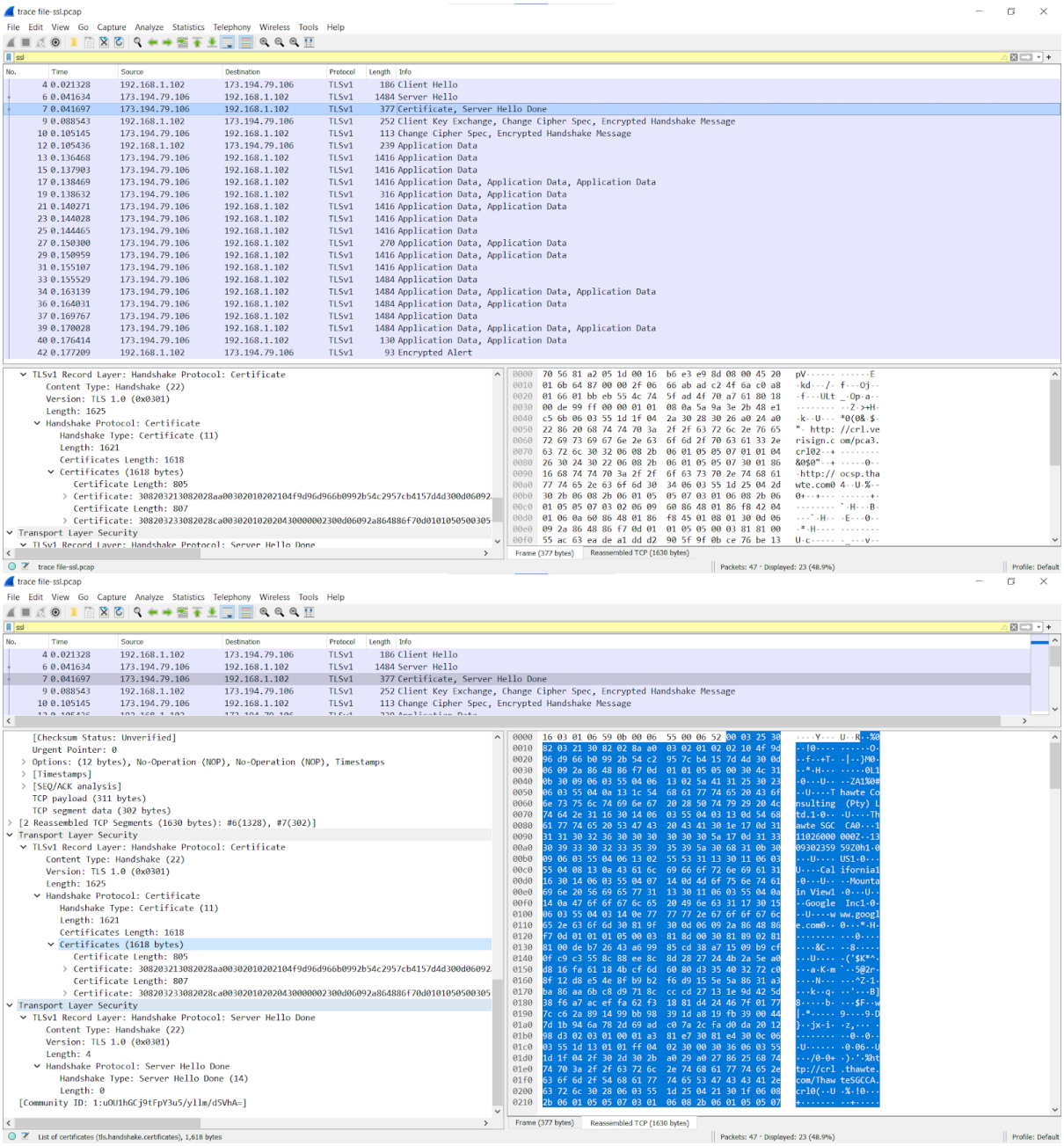
Certificate Messages:

Next, find and inspect the details of the Certificate message, including expanding the Handshake protocol block within the TLS Record. As with the Hellos, the contents of the Certificate message are visible because an encryption scheme is not yet established. It should come after the Hello messages.

1. Who sends the Certificate, the client, the server, or both? A certificate is sent by one party to let the other party authenticate that it is who it claims to be. Based on this usage, you should be able to guess who sends the certificate and check the messages in your trace.

Ans:

The Server sends Certificate to the client



A Certificate message will contain one or more certificates, as needed for one party to verify the identity of the other party from its roots of trust certificates. You can inspect those certificates in your browser.

Client Key Exchange and Change Cipher Messages

Find and inspect the details of the Client Key Exchange and Change Cipher messages, expanding their various details. The key exchange message is sent to pass keying information so that both sides will have the same secret session key. The change cipher message signal a switch to a new encryption scheme to

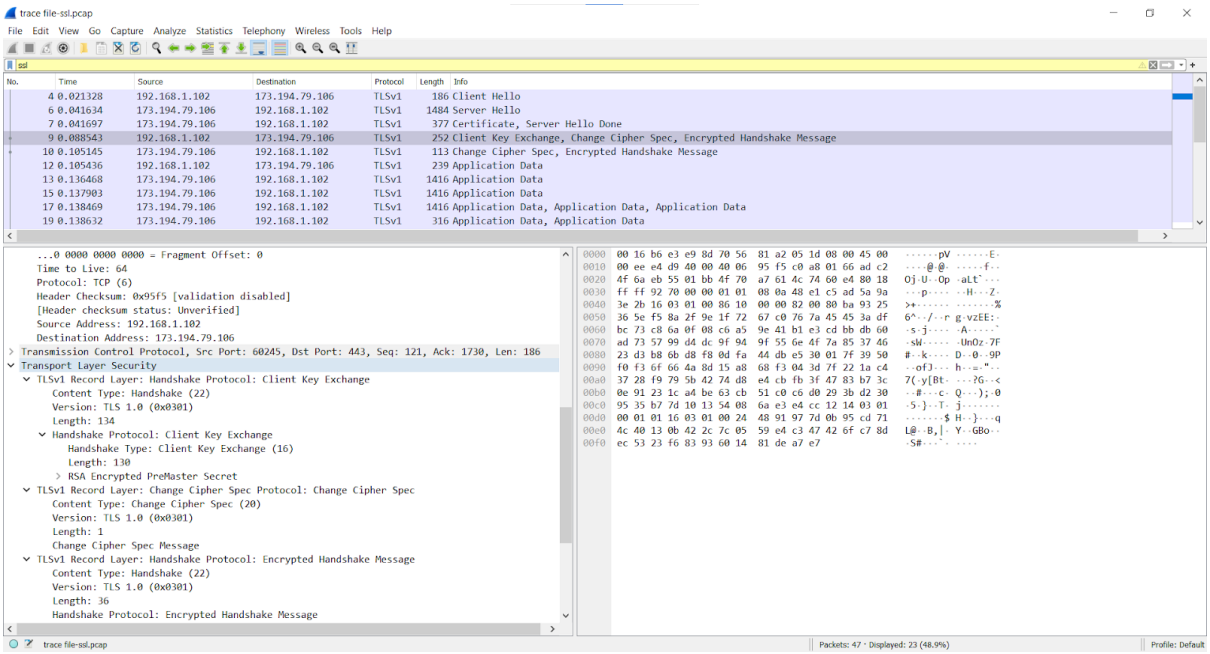
the other party. This means that it is the last unencrypted message sent by the party.

1. Who sends the Change Cipher Spec message, the client, the server, or both?

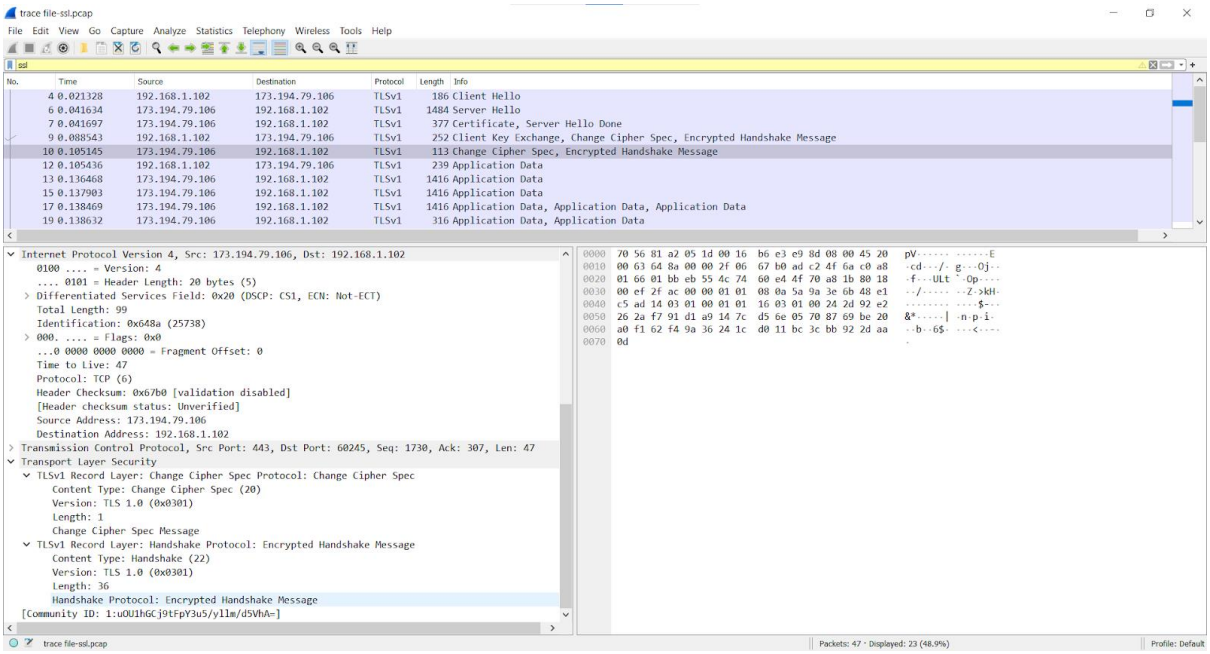
Ans:

Both the server and the client sends the Change Cipher Spec Message

Client:

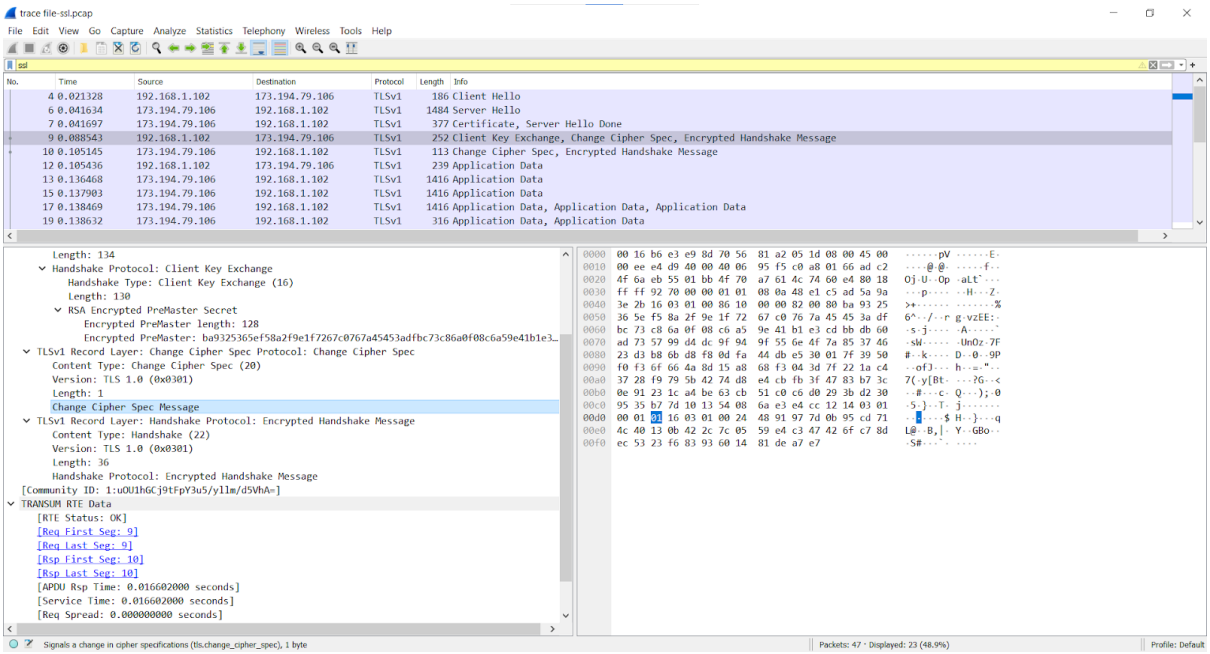


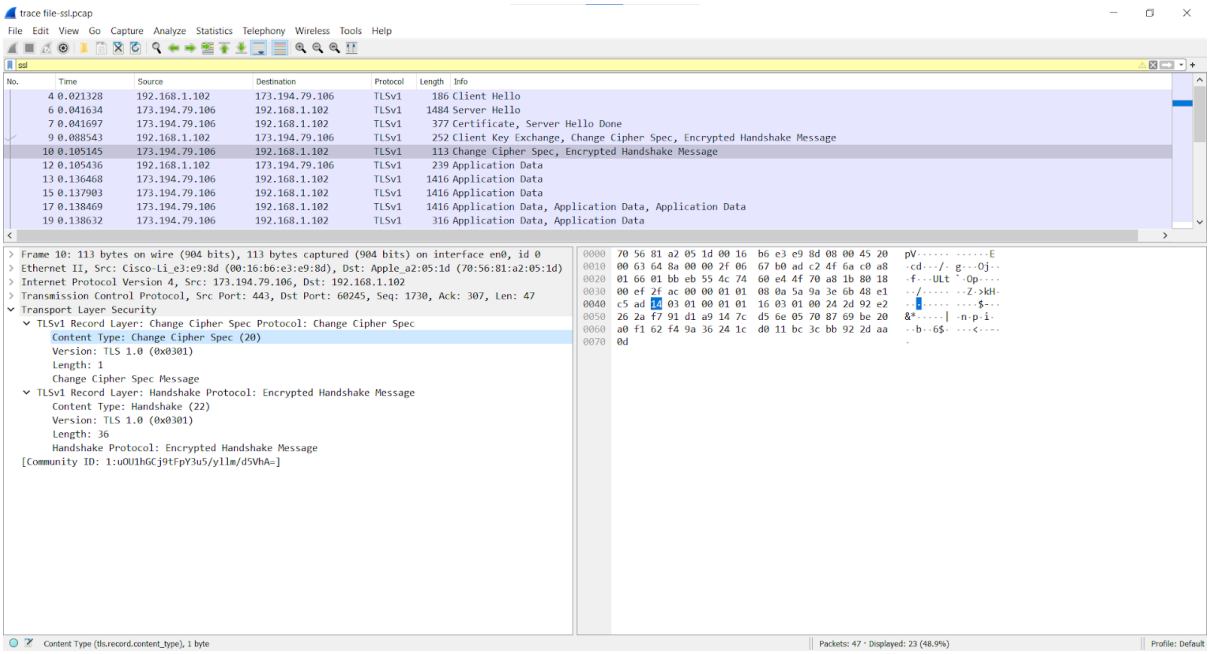
Server:



2. What are the contents carried inside the Change Cipher Spec message? Look past the Content Type and other headers to see the message itself.

Ans:





Conclusion:

Performed the experiment successfully.

Wireshark is used to analyse the packets of various protocols such as TCP, UDP, SSL, TLS, etc.