SSL / TLS

- **SSL** stands for **S**ecure **S**ocket **L**ayer
- **TLS** stands for **T**ransport **L**ayer **S**ecurity
- SSL is an encryption-based internet security protocol, developed by Netscape in 1995.
- In 1999 the Internet Engineering Task Force (IETF) proposed an update to SSL to enhance the security and changed the name as TLS.
- Some people still use SSL to refer to TLS, others use the term "SSL/TLS encryption" because SSL still has so much name recognition.
- This protocol helps to establish a secured, authenticated & encrypted links between the communicating devices

Advantages of SSL / TLS:

• Generally, data on the Web was transmitted in plaintext that anyone could read if they intercepted the message. In order to provide a high degree of **privacy**, SSL encrypts data that is transmitted across the web. This means that anyone who tries to intercept this data will only see a garbled mix of characters that is nearly impossible to decrypt.

• SSL initiates an **authentication** process called a **handshake** between two communicating devices to ensure that both devices are really who they claim to be.

• SSL also digitally signs data in order to provide **data integrity**, verifying that the data is not tampered with before reaching its intended recipient.

SSL/TLS Handshake

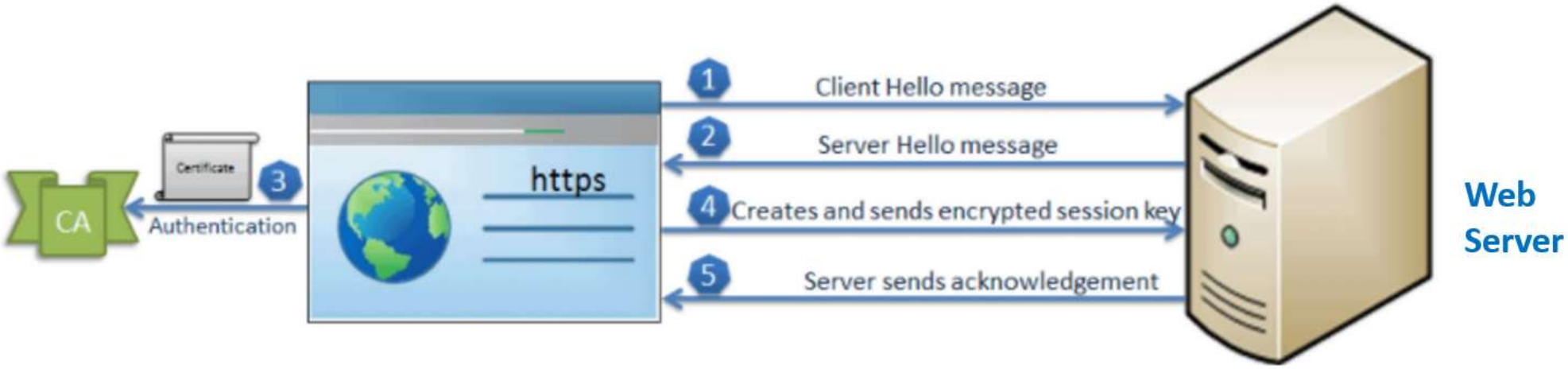
1.The client sends a "**client hello**" message. This includes the client's SSL version number, cipher settings, session-specific data and other information that the server needs to communicate with the client using SSL.

2.The server responds with a "**server hello**" message. This includes the server's SSL version number, cipher settings, session-specific data, an SSL certificate with a public key and other information that the client needs to communicate with the server over SSL.

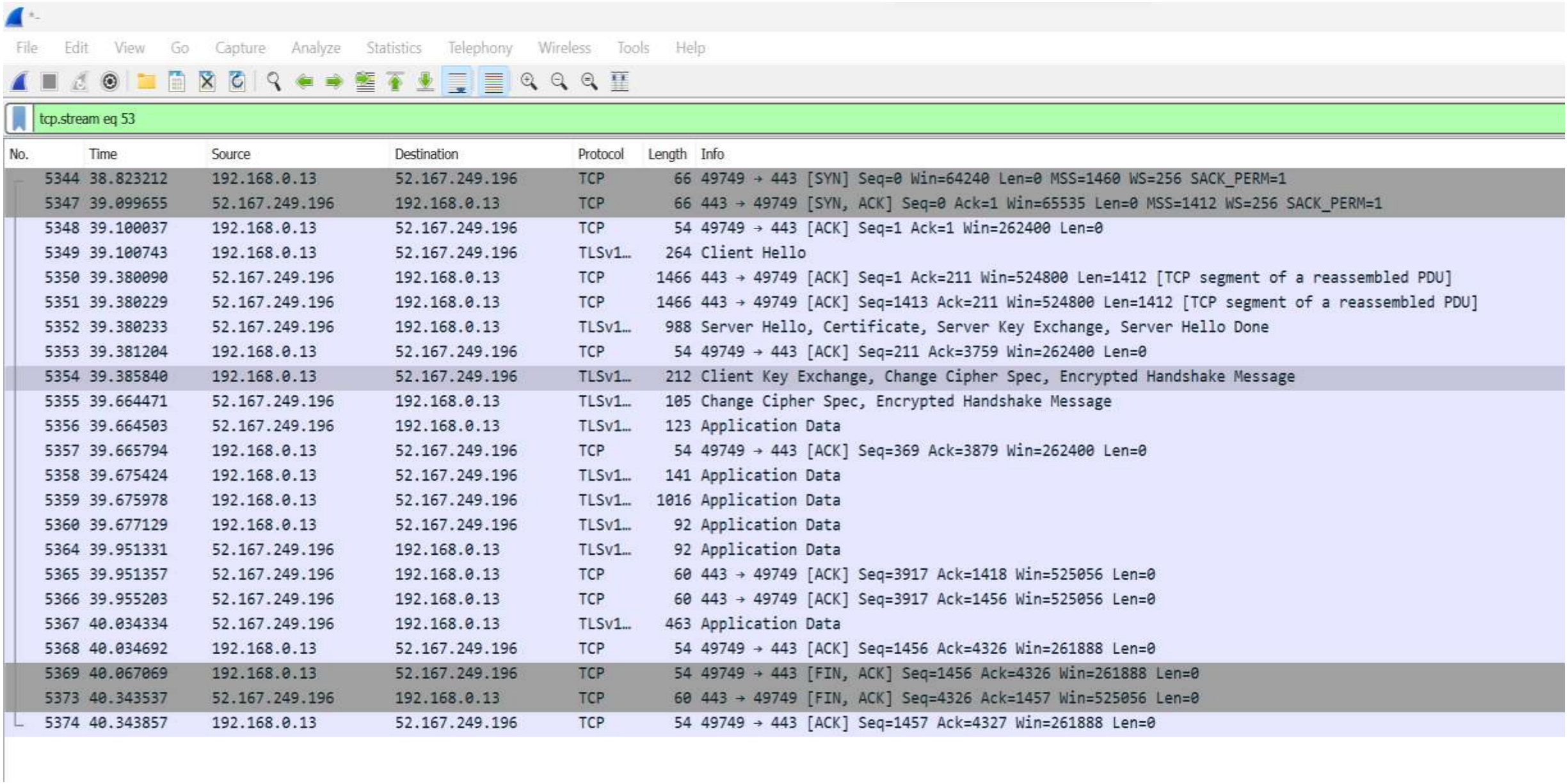
3.The client verifies the server's SSL certificate from list of Root **CA** (Certificate Authority) stored in its browser and authenticates the server. If the authentication fails, then the client refuses the connection and throws an exception. If the authentication succeeds, then proceed to step 4.

4.The client creates a **session key**, encrypts it with the server's public key and sends it to the server. If the server has requested client authentication (mostly in server to server communication), then the client sends his own certificate to the server.

5.The server decrypts the session key with its private key and sends the acknowledgement to the client encrypted with the session key.

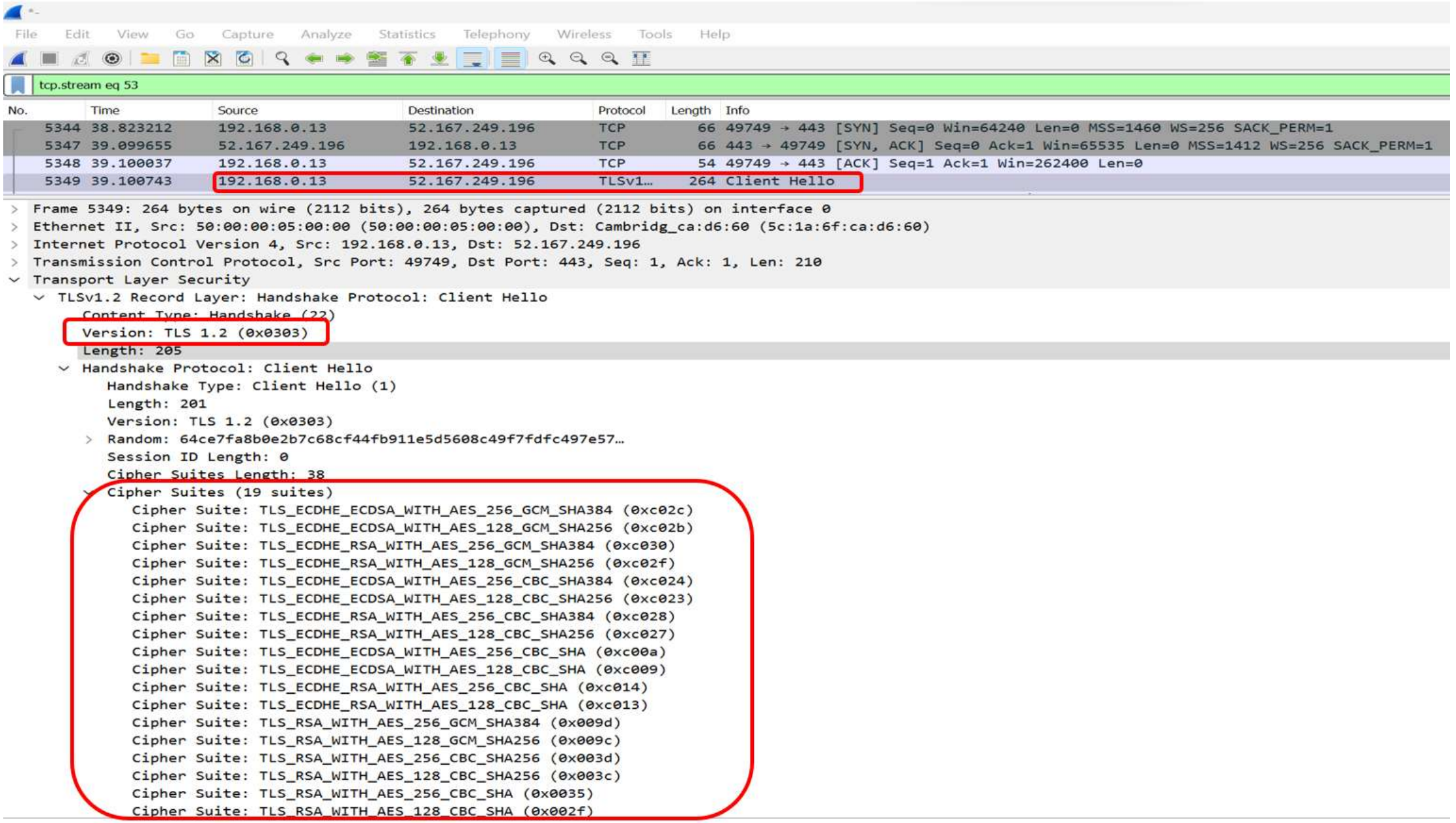


SSL/TLS Wireshark capture

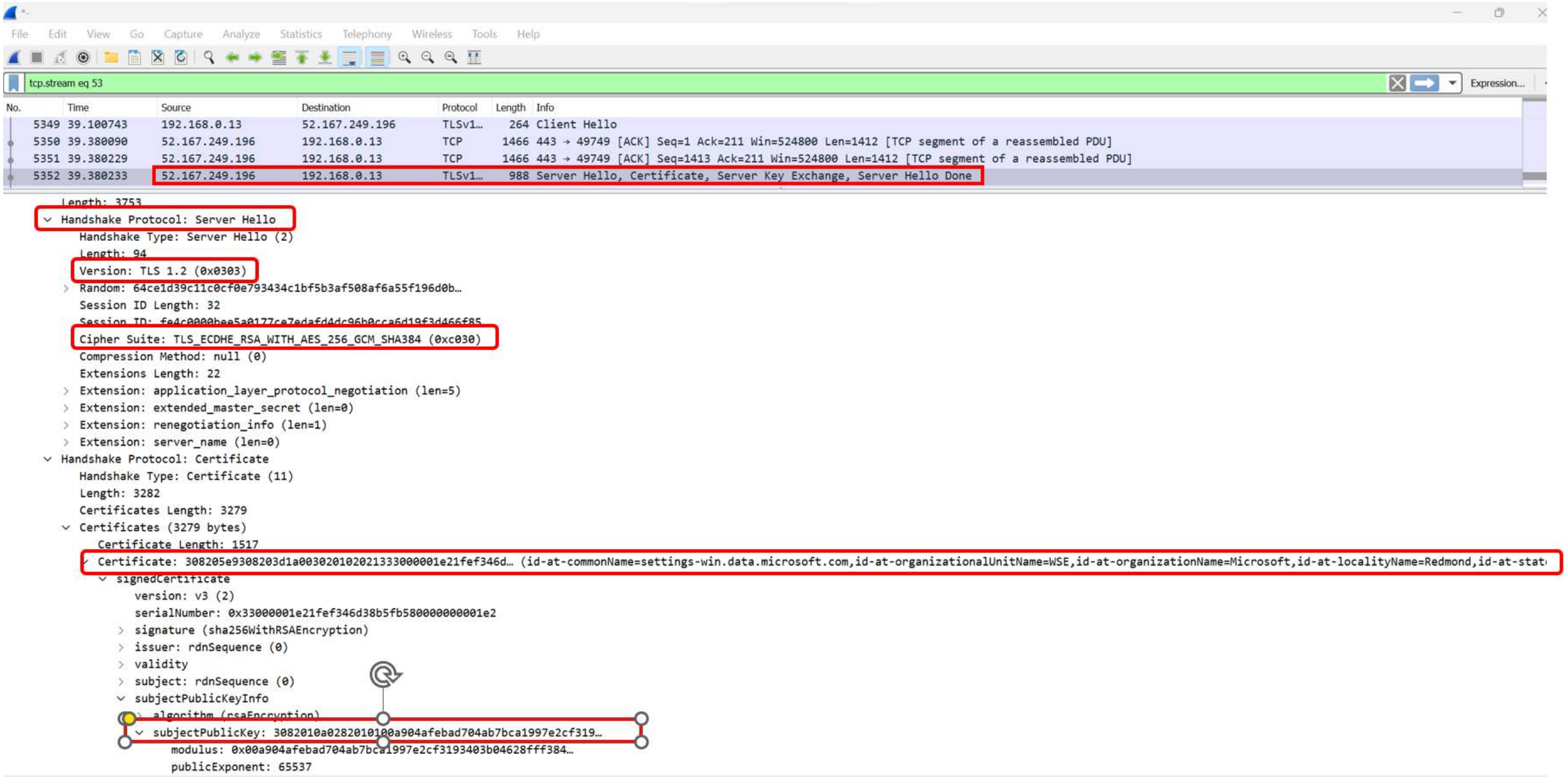


No.	Time	Source	Destination	Protocol	Length	Info
5344	38.823212	192.168.0.13	52.167.249.196	TCP	66	49749 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
5347	39.099655	52.167.249.196	192.168.0.13	TCP	66	443 → 49749 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 WS=256 SACK_PERM=1
5348	39.100037	192.168.0.13	52.167.249.196	TCP	54	49749 → 443 [ACK] Seq=1 Ack=1 Win=262400 Len=0
5349	39.100743	192.168.0.13	52.167.249.196	TLSv1…	264	Client Hello
5350	39.380090	52.167.249.196	192.168.0.13	TCP	1466	443 → 49749 [ACK] Seq=1 Ack=211 Win=524800 Len=1412 [TCP segment of a reassembled PDU]
5351	39.380229	52.167.249.196	192.168.0.13	TCP	1466	443 → 49749 [ACK] Seq=1413 Ack=211 Win=524800 Len=1412 [TCP segment of a reassembled PDU]
5352	39.380233	52.167.249.196	192.168.0.13	TLSv1…	988	Server Hello, Certificate, Server Key Exchange, Server Hello Done
5353	39.381204	192.168.0.13	52.167.249.196	TCP	54	49749 → 443 [ACK] Seq=211 Ack=3759 Win=262400 Len=0
5354	39.385840	192.168.0.13	52.167.249.196	TLSv1…	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
5355	39.664471	52.167.249.196	192.168.0.13	TLSv1…	105	Change Cipher Spec, Encrypted Handshake Message
5356	39.664503	52.167.249.196	192.168.0.13	TLSv1…	123	Application Data
5357	39.665794	192.168.0.13	52.167.249.196	TCP	54	49749 → 443 [ACK] Seq=369 Ack=3879 Win=262400 Len=0
5358	39.675424	192.168.0.13	52.167.249.196	TLSv1…	141	Application Data
5359	39.675978	192.168.0.13	52.167.249.196	TLSv1…	1016	Application Data
5360	39.677129	192.168.0.13	52.167.249.196	TLSv1…	92	Application Data
5364	39.951331	52.167.249.196	192.168.0.13	TLSv1…	92	Application Data
5365	39.951357	52.167.249.196	192.168.0.13	TCP	60	443 → 49749 [ACK] Seq=3917 Ack=1418 Win=525056 Len=0
5366	39.955203	52.167.249.196	192.168.0.13	TCP	60	443 → 49749 [ACK] Seq=3917 Ack=1456 Win=525056 Len=0
5367	40.034334	52.167.249.196	192.168.0.13	TLSv1…	463	Application Data
5368	40.034692	192.168.0.13	52.167.249.196	TCP	54	49749 → 443 [ACK] Seq=1456 Ack=4326 Win=261888 Len=0
5369	40.067069	192.168.0.13	52.167.249.196	TCP	54	49749 → 443 [FIN, ACK] Seq=1456 Ack=4326 Win=261888 Len=0
5373	40.343537	52.167.249.196	192.168.0.13	TCP	60	443 → 49749 [FIN, ACK] Seq=4326 Ack=1457 Win=525056 Len=0
5374	40.343857	192.168.0.13	52.167.249.196	TCP	54	49749 → 443 [ACK] Seq=1457 Ack=4327 Win=261888 Len=0

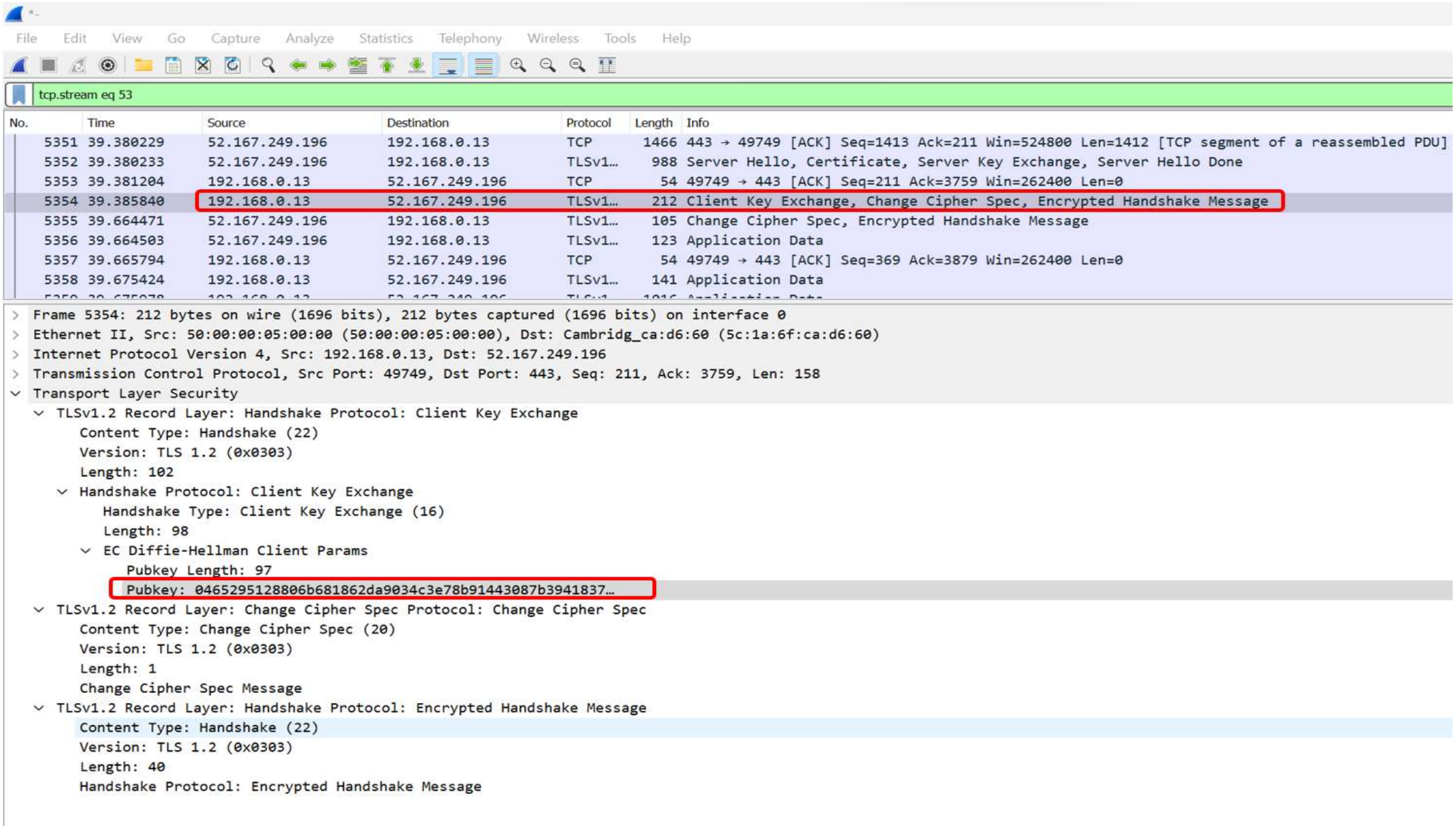
Client Hello Message



Server Hello message



Client Key Exchange



Encrypted data transactions: