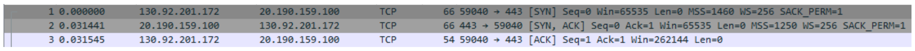
10.4 Question 4

10.4.A HTTPS operates over three connection levels – HTTP, TLS, and TCP. Have a look at the packet capture in *question4.pcap* and describe, which packets correspond to which of the three levels. Identify the TCP handshake, TLS handshake, HTTP data (specify packet numbers). (Recommendation: use Wireshark)

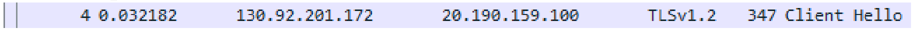
Packet 1-3 . . . . . .



```
1 0.000000   130.92.201.172   20.190.159.100   TCP   66 59040 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
2 0.031441   20.190.159.100   130.92.201.172   TCP   66 443 → 59040 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1250 WS=256 SACK_PERM=1
3 0.031545   130.92.201.172   20.190.159.100   TCP   54 59040 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
```

Port 443 is the standardized port for any HTTPS traffic. Here the connection with the server is established. These three packages are part of the TCP handshake using SYN and ACK packets.

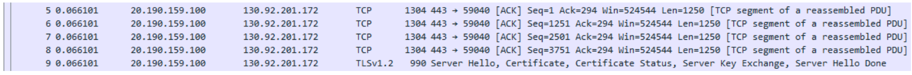
Packet 4 . . . .



```
4 0.032182   130.92.201.172   20.190.159.100   TLSv1.2   347 Client Hello
```

Packet 4 is the initialization of the TLS handshake by sending the *Client Hello* message.

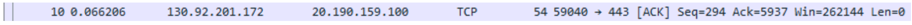
Packet 5-9 . . . . . .



```
5 0.066101   20.190.159.100   130.92.201.172   TCP       1304 443 → 59040 [ACK] Seq=1 Ack=294 Win=524544 Len=1250 [TCP segment of a reassembled PDU]
6 0.066101   20.190.159.100   130.92.201.172   TCP       1304 443 → 59040 [ACK] Seq=1251 Ack=294 Win=524544 Len=1250 [TCP segment of a reassembled PDU]
7 0.066101   20.190.159.100   130.92.201.172   TCP       1304 443 → 59040 [ACK] Seq=2501 Ack=294 Win=524544 Len=1250 [TCP segment of a reassembled PDU]
8 0.066101   20.190.159.100   130.92.201.172   TCP       1304 443 → 59040 [ACK] Seq=3751 Ack=294 Win=524544 Len=1250 [TCP segment of a reassembled PDU]
9 0.066101   20.190.159.100   130.92.201.172   TLSv1.2    990 Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
```

The marking *[TCP segment of a reassembled PDU]* implies that the packets 5-8 are part of a larger packet. These are used to collect multiple TCP segments for the first stage. Packet 9 contains the *Server Hello*, the certificate, its status, the server key exchange, and the *Server Hello Done*, hence being equivalent to stage 2 from the lecture.

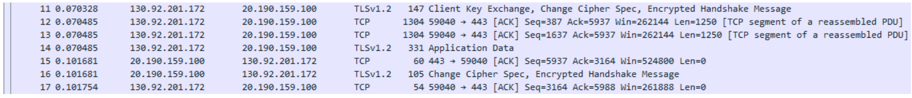
Packet 10 . . . . . .



```
10 0.066206   130.92.201.172   20.190.159.100   TCP   54 59040 → 443 [ACK] Seq=294 Ack=5937 Win=262144 Len=0
```

The client ackowledges that it received the packet.

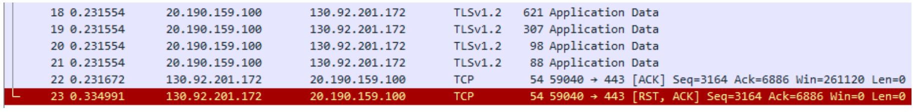
Packet 11-17 . . .



```
11 0.070328   130.92.201.172   20.190.159.100   TLSv1.2   147 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
12 0.070485   130.92.201.172   20.190.159.100   TCP      1304 59040 → 443 [ACK] Seq=387 Ack=5937 Win=262144 Len=1250 [TCP segment of a reassembled PDU]
13 0.070485   130.92.201.172   20.190.159.100   TCP      1304 59040 → 443 [ACK] Seq=1637 Ack=5937 Win=262144 Len=1250 [TCP segment of a reassembled PDU]
14 0.070485   130.92.201.172   20.190.159.100   TLSv1.2   331 Application Data
15 0.101681   20.190.159.100   130.92.201.172   TCP        60 443 → 59040 [ACK] Seq=5937 Ack=3164 Win=524800 Len=0
16 0.101681   20.190.159.100   130.92.201.172   TLSv1.2   105 Change Cipher Spec, Encrypted Handshake Message
17 0.101754   130.92.201.172   20.190.159.100   TCP        54 59040 → 443 [ACK] Seq=3164 Ack=5988 Win=261888 Len=0
```

The cipher suite is changed and in the end the handshake protocol is terminated.

Packet 18-23 . . .



```
18 0.231554   20.190.159.100   130.92.201.172   TLSv1.2   621 Application Data
19 0.231554   20.190.159.100   130.92.201.172   TLSv1.2   307 Application Data
20 0.231554   20.190.159.100   130.92.201.172   TLSv1.2    98 Application Data
21 0.231554   20.190.159.100   130.92.201.172   TLSv1.2    88 Application Data
22 0.231672   130.92.201.172   20.190.159.100   TCP        54 59040 → 443 [ACK] Seq=3164 Ack=6886 Win=261120 Len=0
23 0.334991   130.92.201.172   20.190.159.100   TCP        54 59040 → 443 [RST, ACK] Seq=3164 Ack=6886 Win=0 Len=0
```

In these packets HTTP data is sent which is said to be any *Application Data*.

With these information we can say that packet 4 until packet 17 is the TLS handshake.

10.4.B Is mutual authentication in place in the TLS handshake from question 4 A?

Phase 2 . . . . . As the server sends its own certificate it is obviously authenticated. However, it is particular that no *certificate_request message* was sent which requests a certificate from the client.

Phase 3 . . . . . As a baseline the book *Cryptography and Network Security Principles and Practice Seventh Edition page 560* is used: The server starts phase 3 by requesting a certificate from the client. As described above this is what is happening so both sides are authenticated to each other. However, it is ver particular that no *certificate_verify* message is sent in order to verify the certificates.