Oppgave 2:

For å lage nøkler for at server og klient skal kunne kommunisere brukte jeg kommandoen:

keytool -genkey -alias signFiles -keystore examplestore

Må bruke enten Java 8 versjonen av Keytools eller endre keystore hashingen til RSA, for at keystoren skal fungere som nøkler (Brukte Java 8).

Koden er lagt til i besvarelsen.

Brukte Java 8 når jeg kompilerte filene.

Setter keyStoren til serveren og trustStoren til klienten ved bruk av disse kommandoene. Må være absolute path til keyStoren jeg laget med forrige kommando. Kunne brukt ./examplestore også tror jeg.

```
javac TLSServer.java
java -Djavax.net.ssl.keyStore=/mnt/c/Users/dider/NettProg/Oving2/examplestore -Djavax.net.ssl.keyStorePassword=password TLSServer
```

```
javac TLSClient.java
java -Djavax.net.ssl.trustStore=/mnt/c/Users/dider/NettProg/Oving2/examplestore -Djavax.net.ssl.trustStorePassword=password TLSClient
```

Klient:

```
Enter something:
Hello World
Hello World
Enter something:
```

Tjener:

```
SSL server running on port: 4545...
[SSL: ServerSocket[addr=0.0.0.0/0.0.0.0,localport=4545]]
Server connected
Hello World
```

Ved Wireshark får vi:

1	0.000000	127.0.0.1	127.0.0.1	TCP	56	58254 → 4545 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
2	0.000026	127.0.0.1	127.0.0.1	TCP	56	4545 → 58254 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
3	0.000107	127.0.0.1	127.0.0.1	TCP	44	58254 → 4545 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
4	4.617690	127.0.0.1	127.0.0.1	TLSv1.2	311	Client Hello
5	4.617729	127.0.0.1	127.0.0.1	TCP	44	4545 → 58254 [ACK] Seq=1 Ack=268 Win=2619392 Len=0
6	4.655419	127.0.0.1	127.0.0.1	TLSv1.2	134	Server Hello
7	4.655436	127.0.0.1	127.0.0.1	TCP	44	58254 → 4545 [ACK] Seq=268 Ack=91 Win=2619648 Len=0
8	4.655641	127.0.0.1	127.0.0.1	TLSv1.2	1276	Certificate
9	4.655654	127.0.0.1	127.0.0.1	TCP	44	58254 → 4545 [ACK] Seq=268 Ack=1323 Win=2618368 Len=0
10	4.661931	127.0.0.1	127.0.0.1	TLSv1.2	639	Server Key Exchange
11	4.661954	127.0.0.1	127.0.0.1	TCP	44	58254 → 4545 [ACK] Seq=268 Ack=1918 Win=2617856 Len=0
12	4.662101	127.0.0.1	127.0.0.1	TLSv1.2	53	Server Hello Done
13	4.662111	127.0.0.1	127.0.0.1	TCP	44	58254 → 4545 [ACK] Seq=268 Ack=1927 Win=2617856 Len=0
14	4.691296	127.0.0.1	127.0.0.1	TLSv1.2	311	Client Key Exchange
15	4.691331	127.0.0.1	127.0.0.1	TCP	44	4545 → 58254 [ACK] Seq=1927 Ack=535 Win=2619136 Len=0
16	4.706711	127.0.0.1	127.0.0.1	TLSv1.2	50	Change Cipher Spec
17	4.706750	127.0.0.1	127.0.0.1	TCP	44	4545 → 58254 [ACK] Seq=1927 Ack=541 Win=2619136 Len=0
18	4.751059	127.0.0.1	127.0.0.1	TLSv1.2	89	Encrypted Handshake Message
19	4.751077	127.0.0.1	127.0.0.1	TCP	44	4545 → 58254 [ACK] Seq=1927 Ack=586 Win=2619136 Len=0
20	4.796743	127.0.0.1	127.0.0.1	TLSv1.2	50	Change Cipher Spec
21	4.796760	127.0.0.1	127.0.0.1	TCP	44	58254 → 4545 [ACK] Seq=586 Ack=1933 Win=2617856 Len=0
22	4.797001	127.0.0.1	127.0.0.1	TLSv1.2	89	Encrypted Handshake Message
23	4.797011	127.0.0.1	127.0.0.1	TCP	44	58254 → 4545 [ACK] Seq=586 Ack=1978 Win=2617600 Len=0
24	4.798147	127.0.0.1	127.0.0.1	TLSv1.2	79	Application Data
25	4.798157	127.0.0.1	127.0.0.1	TCP	44	4545 → 58254 [ACK] Seq=1978 Ack=621 Win=2619136 Len=0
26	4.798536	127.0.0.1	127.0.0.1	TLSv1.2	79	Application Data
27	4.798545	127.0.0.1	127.0.0.1	TCP	44	58254 → 4545 [ACK] Seq=621 Ack=2013 Win=2617600 Len=0

Her ser vi at vi får en TCP med TLS handshake sammenkobling.

4	4.617690	127.0.0.1	127.0.0.1	TLSv1.2	311	Client Hello
6	4.655419	127.0.0.1	127.0.0.1	TLSv1.2	134	Server Hello
8	4.655641	127.0.0.1	127.0.0.1	TLSv1.2	1276	Certificate
10	4.661931	127.0.0.1	127.0.0.1	TLSv1.2	639	Server Key Exchange
12	4.662101	127.0.0.1	127.0.0.1	TLSv1.2	53	Server Hello Done
14	4.691296	127.0.0.1	127.0.0.1	TLSv1.2	311	Client Key Exchange
16	4.706711	127.0.0.1	127.0.0.1	TLSv1.2	50	Change Cipher Spec
18	4.751059	127.0.0.1	127.0.0.1	TLSv1.2	89	Encrypted Handshake Message
20	4.796743	127.0.0.1	127.0.0.1	TLSv1.2	50	Change Cipher Spec
22	4.797001	127.0.0.1	127.0.0.1	TLSv1.2	89	Encrypted Handshake Message
24	4.798147	127.0.0.1	127.0.0.1	TLSv1.2	79	Application Data
26	4.798536	127.0.0.1	127.0.0.1	TLSv1.2	79	Application Data

Starter med en tjener og klient «hello», deretter blir det et bytte av nøklene som skal være like ettersom vi satte nøklene til samme keystore. Disse godkjennes før noe som helst av dataen som klienten sendte blir sendt over. Dette skjer etter en kryptert handshake som godkjenner at klienten kan kobles til serveren. Dataen som her er meldingen «Hello» blir sendt i Application dataen, først fra klienten til serveren, deretter fra serveren som svarer med samme melding til klienten. Disse meldingene er krypterte:

```
▾ Transport Layer Security
  ▾ TLSv1.2 Record Layer: Application Data Protocol: Application Data
        Content Type: Application Data (23)
        Version: TLS 1.2 (0x0303)
        Length: 30
        Encrypted Application Data: 00000000000000001693f2c895e8a5b916cc35ceb12aa44ae4e82bd1e3b92
```

Fra klienten til server

```
Source Port: 58254
Destination Port: 4545
```

Fra server til klienten:

```
Source Port: 4545
Destination Port: 58254
```