Øving 2 oppgave 2

Først og fremst opprettet jeg et keystore med algoritmen RSA i C:\Users\Java\jdk-11.0.6\bin. Deretter kopierte jeg javaklassene fra nettsiden og jaret de. Etter dette er alt man må gjøre for å få det å fungere å starte programmet via kommandovinduet sammen med keystoret som du opprettet.

```
Microsoft Windows [Version 10.0.19042.746]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Lars>java -jar -Djavax.net.ssl.KeyStore=C:\Java\jdk-11.0.6\bin\examplestore -Djavax.net.ssl.keyStorePassword=pa
ssword "C:\Users\Lars\Desktop\JavaSSLServer.jar"
SSL ServerSocket started
[SSL: ServerSocket[addr=0.0.0.0/0.0.0.0,localport=8000]]
ServerSocket accepted
```

```
C:\Users\Lars>java -jar -Djavax.net.ssl.trustStore=C:\Java\jdk-11.0.6\bin\examplestore -Djavax.net.ssl.trustStorePasswor
d=password "C:\users\lars\desktop\JavaSSLClient.jar"
Enter something:
08-Feb-21
```

De to programmene vil nå kommunisere kryptert. For å kunne se nærmere på dette kan man bruke wireshark. Velger loopback filteret i wireshark, og filtrerer på port 8000.

Når klienten kobler seg til serveren:

1	0.000000	127.0.0.1	127.0.0.1	TCP	56 55515 → 9212 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
2	0.000028	127.0.0.1	127.0.0.1	TCP	56 9212 → 55515 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
3	0.000073	127.0.0.1	127.0.0.1	TCP	44 55515 → 9212 [ACK] Seq=1 Ack=1 Win=2619648 Len=0

Når man sender første melding vil klienten initiere en Client Hello. Denne initierer en sesjon med serveren. Deretter vil serveren svare med en Server Hello som inneholder informasjon om serveren, og deretter vil sertifikatet bli sendt. Change Cipher Spec betyr rett og slett at serveren blir gjort oppmerksom på at fremtidig kommunikasjon med denne klienten vil bli gjort med nøkkelen og algoritmen som nettopp ble brukt.

4	55.222089	127.0.0.1	127.0.0.1	TLSv1.2	440 Client Hello
5	55.222107	127.0.0.1	127.0.0.1	TCP	44 9212 → 55515 [ACK] Seq=1 Ack=397 Win=2619648 Len=0
6	55.246958	127.0.0.1	127.0.0.1	TLSv1.3	204 Server Hello
7	55.246981	127.0.0.1	127.0.0.1	TCP	44 55515 → 9212 [ACK] Seq=397 Ack=161 Win=2619392 Len=0
8	55.260995	127.0.0.1	127.0.0.1	TLSv1.3	50 Change Cipher Spec
9	55.261012	127.0.0.1	127.0.0.1	TCP	44 9212 → 55515 [ACK] Seq=161 Ack=403 Win=2619648 Len=0
10	55.262202	127.0.0.1	127.0.0.1	TLSv1.3	50 Change Cipher Spec
11	55.262222	127.0.0.1	127.0.0.1	TCP	44 55515 → 9212 [ACK] Seq=403 Ack=167 Win=2619392 Len=0
12	55.268128	127.0.0.1	127.0.0.1	TLSv1.3	110 Application Data
13	55.268150	127.0.0.1	127.0.0.1	TCP	44 55515 → 9212 [ACK] Seq=403 Ack=233 Win=2619392 Len=0
14	55.269863	127.0.0.1	127.0.0.1	TLSv1.3	966 Application Data
15	55.269874	127.0.0.1	127.0.0.1	TCP	44 55515 → 9212 [ACK] Seq=403 Ack=1155 Win=2618624 Len=0
16	55.292442	127.0.0.1	127.0.0.1	TLSv1.3	346 Application Data
17	55.292465	127.0.0.1	127.0.0.1	TCP	44 55515 → 9212 [ACK] Seq=403 Ack=1457 Win=2618112 Len=0

Når denne handshaken er ferdig, vil enhver melding sendt med klientprogrammet resultere i fire linjer i wireshark. En for å sende, en for acknowledgement, for hvert av programmene.

Når du kutter kommunikasjonen, dukker denne meldingen opp: