

# Cryptography

DAT159

By Sondre Gjellestad and Arne Molland

## Task 1

After setting up the certificate, the connection works.

```
Message to TCPServer: Message from TC
P SSLClient
Response from Server: HTTP/1.1 200 OK
```

Run | Debug

```
public static void main(String[] args) throws InvalidKeyException, NoSuchAlgorithmException,
    NoSuchPaddingException, UnrecoverableKeyException, KeyStoreException {
    // Set the truststore dynamically using the system property

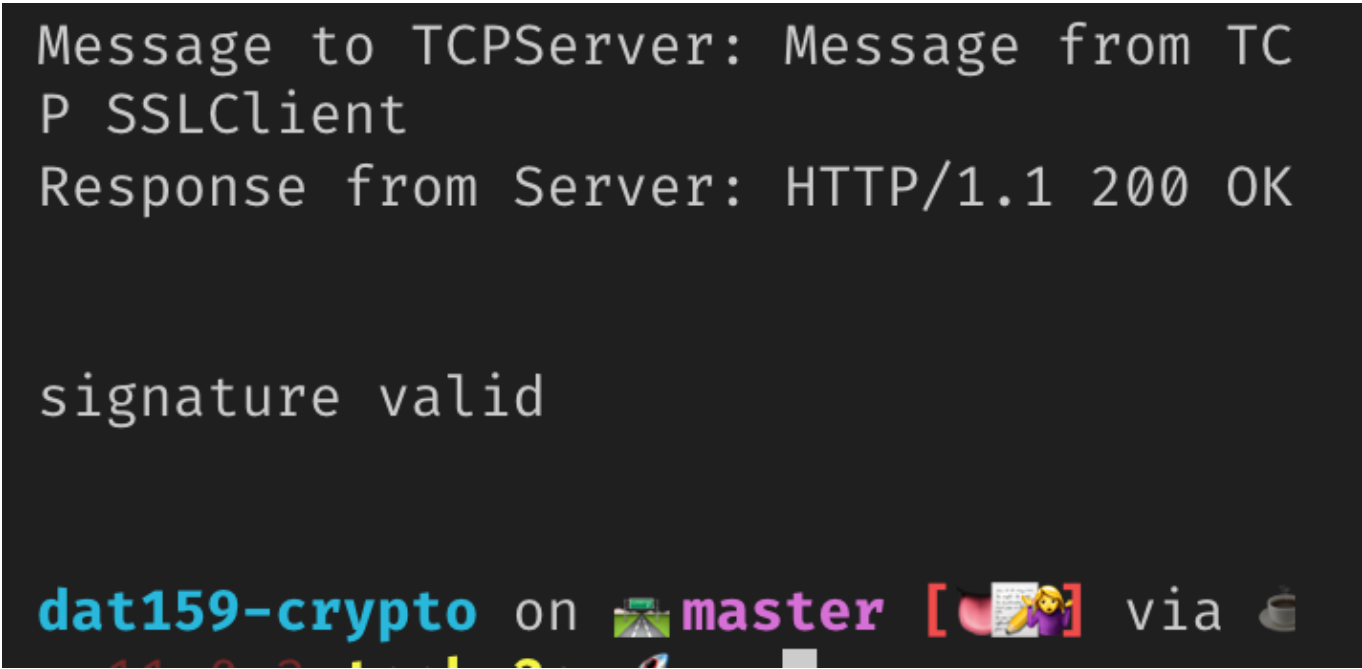
    System.setProperty("javax.net.ssl.trustStore", "keys/tcp_truststore");
    System.setProperty("javax.net.ssl.trustStorePassword", "abcdef");

    String message = "Message from TCP SSLClient";
    TCPClientSSLRSA c = new TCPClientSSLRSA(ServerConfig.SERVER, ServerConfig.PORT);
    c.clientProcess(message);
}
```

3972	99.315022	localhost	localhost	TLSv1.3	481	Client Hello
3973	99.315047	localhost	localhost	TCP	56	ibm-rsyscon(9085) → 53127 [ACK] Seq=1 Ack=426 Win=407872 Len=0 TSval=114092558
3974	99.370974	localhost	localhost	TLSv1.3	216	Server Hello
3975	99.371001	localhost	localhost	TCP	56	53127 → ibm-rsyscon(9085) [ACK] Seq=426 Ack=161 Win=408128 Len=0 TSval=114092558
3976	99.384994	localhost	localhost	TLSv1.3	62	Change Cipher Spec
3977	99.385018	localhost	localhost	TCP	56	53127 → ibm-rsyscon(9085) [ACK] Seq=426 Ack=167 Win=408128 Len=0 TSval=114092558
3978	99.387538	localhost	localhost	TLSv1.3	62	Change Cipher Spec
3979	99.387555	localhost	localhost	TCP	56	ibm-rsyscon(9085) → 53127 [ACK] Seq=167 Ack=432 Win=407808 Len=0 TSval=114092558
3980	99.400944	localhost	localhost	TLSv1.3	136	Application Data
3981	99.400962	localhost	localhost	TCP	56	53127 → ibm-rsyscon(9085) [ACK] Seq=432 Ack=247 Win=408000 Len=0 TSval=114092558
3982	99.403963	localhost	localhost	TLSv1.3	1004	Application Data
3983	99.403978	localhost	localhost	TCP	56	53127 → ibm-rsyscon(9085) [ACK] Seq=432 Ack=1195 Win=407104 Len=0 TSval=114092558
3984	99.464908	localhost	localhost	TLSv1.3	358	Application Data
3985	99.464934	localhost	localhost	TCP	56	53127 → ibm-rsyscon(9085) [ACK] Seq=432 Ack=1497 Win=406784 Len=0 TSval=114092558

## Task 3

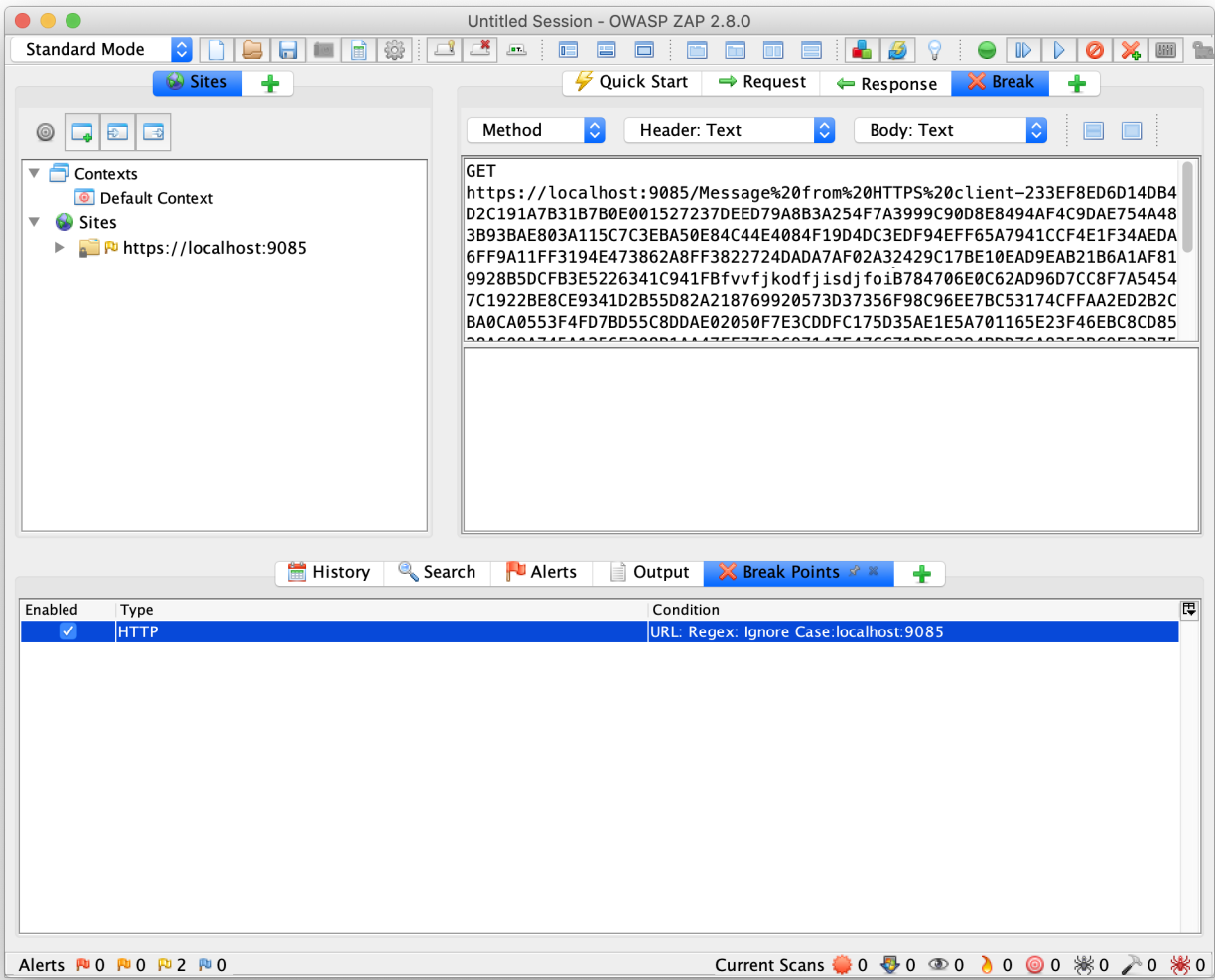
Here is the result in WireShark.



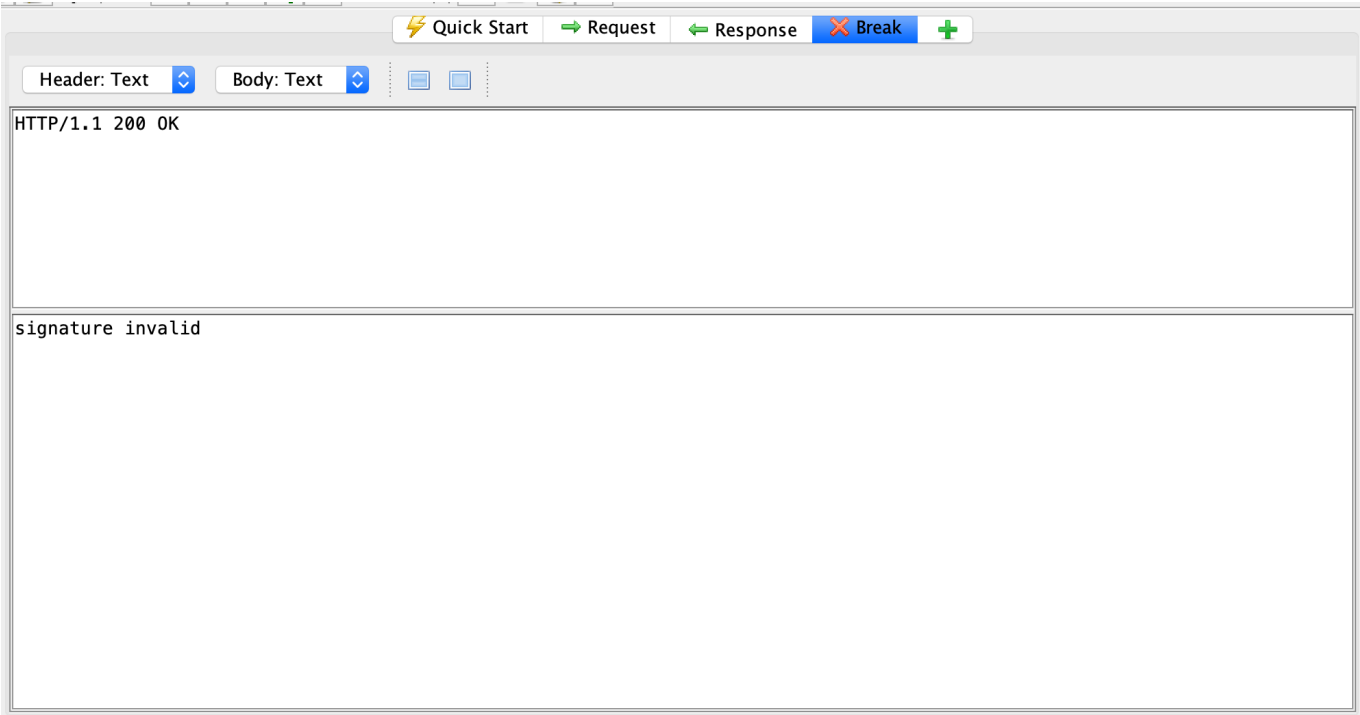
157	23.730437	localhost	localhost	TCP	56	62153 → ibm-rsyscon(9085) [ACK] Seq=1 Ack=1 Win=408256 Len=0 TSval=104359
158	23.730445	localhost	localhost	TCP	56	[TCP Window Update] ibm-rsyscon(9085) → 62153 [ACK] Seq=1 Ack=1 Win=408256
159	23.826555	localhost	localhost	TLSv1.3	481	Client Hello
160	23.826602	localhost	localhost	TCP	56	ibm-rsyscon(9085) → 62153 [ACK] Seq=1 Ack=426 Win=407872 Len=0 TSval=104359
161	23.862079	localhost	localhost	TLSv1.3	216	Server Hello
162	23.862100	localhost	localhost	TCP	56	62153 → ibm-rsyscon(9085) [ACK] Seq=426 Ack=161 Win=408128 Len=0 TSval=104359
163	23.876571	localhost	localhost	TLSv1.3	62	Change Cipher Spec
164	23.876594	localhost	localhost	TCP	56	62153 → ibm-rsyscon(9085) [ACK] Seq=426 Ack=167 Win=408128 Len=0 TSval=104359
165	23.879214	localhost	localhost	TLSv1.3	62	Change Cipher Spec
166	23.879235	localhost	localhost	TCP	56	ibm-rsyscon(9085) → 62153 [ACK] Seq=167 Ack=432 Win=407808 Len=0 TSval=104359
167	23.895409	localhost	localhost	TLSv1.3	136	Application Data
168	23.895427	localhost	localhost	TCP	56	62153 → ibm-rsyscon(9085) [ACK] Seq=432 Ack=247 Win=408000 Len=0 TSval=104359
169	23.896998	localhost	localhost	TLSv1.3	96	Application Data
170	23.897017	localhost	localhost	TCP	56	62153 → ibm-rsyscon(9085) [ACK] Seq=432 Ack=287 Win=408000 Len=0 TSval=104359
171	23.897780	localhost	localhost	TCP	44	ibm-rsyscon(9085) → 62153 [RST, ACK] Seq=287 Ack=432 Win=407808 Len=0

## Task 4 | Tamper with the Message (Integrity)

After setting up the proxy certificates, we've set a breakpoint like this:



After tampering with the message this is the response from the server.



Question

Can we impersonate `HttpsClientProxySSLRSA` in this setup? If so, we would need it's private key in order to sign messages correctly.