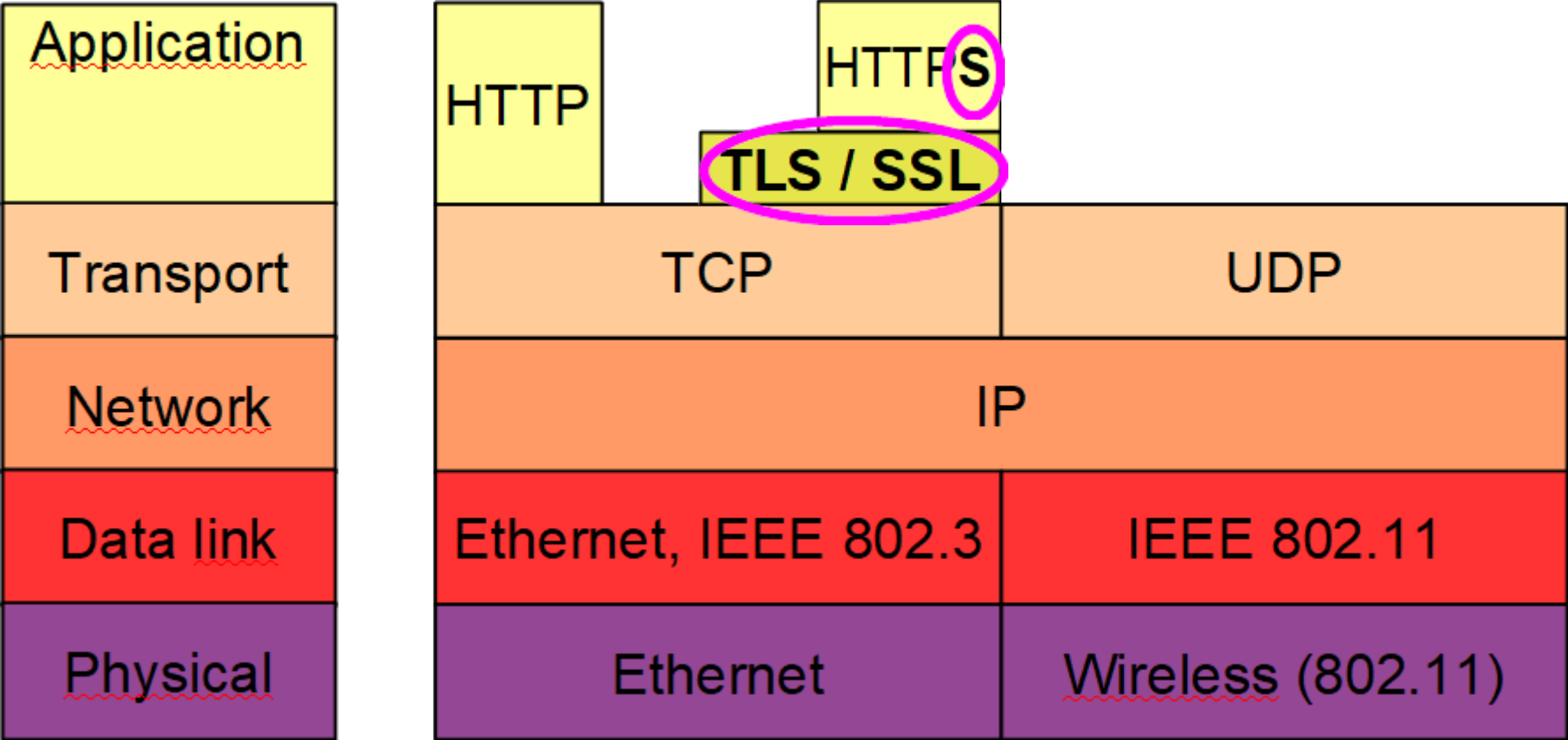


Network Security

Transport Layer Security (TLS)

Prof. Dr. Stefan Heiss



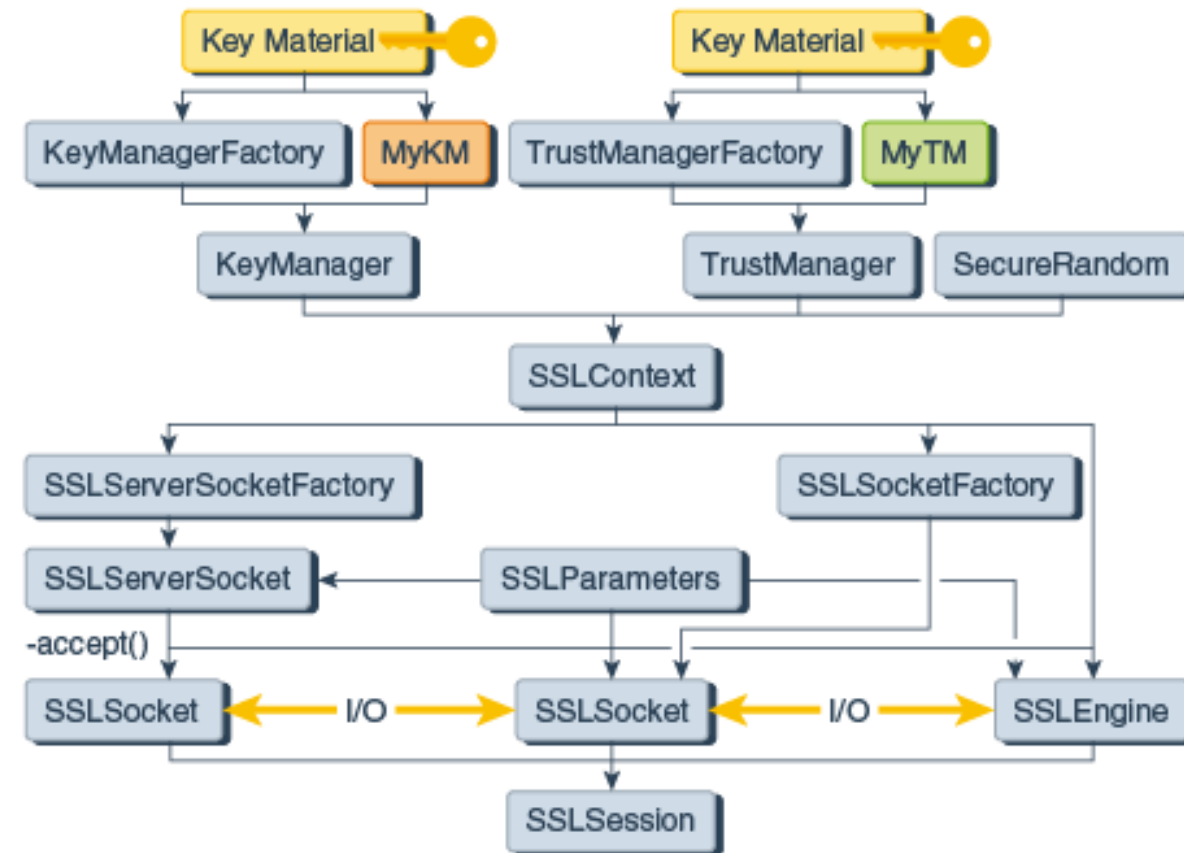
- **Secure Sockets Layer Protocols**

- **SSL 1.0:** Netscape Communications Corp., 1994
- **SSL 2.0:** The SSL Protocol, Netscape Communications Corp., Feb. 1995
- **SSL 3.0:** The SSL 3.0 Protocol, Netscape Communications Corp., Nov. 1996

- **Transport Layer Security Protocols**

- **TLS 1.0 = SSL 3.1:** The TLS Protocol - Version 1.0, [RFC 2246](#), Jan. 1999
- **TLS 1.1:** The Transport Layer Security (TLS) Protocol - Version 1.1, [RFC 4346](#), April 2006
- **TLS 1.2:** The Transport Layer Security (TLS) Protocol - Version 1.2, [RFC 5246](#), August 2008
- **TLS 1.3:** The Transport Layer Security (TLS) Protocol - Version 1.3, [RFC 8446](#), August 2018

- Java Secure Socket Extension (JSSE)
- [JSSE Reference Guide](#) (Chapter 8 in [Security Developer's Guide](#))
- TLS 1.3 support since Java 11



- **TLS Record Protocol**

- Privacy: symmetric encryption
- Integrity: MAC's or AEAD methods
- Keys are negotiated per connection with the help of the TLS Handshake Protocol

- **TLS Handshake Protocol**

- Authentication of peer's identities
- Negotiation of secret session keys

- **TLS v ≤ 1.2:** <Record Protocol Algos.> **_WITH_** <Record Protocol Algos.>
 - Addition of Kerberos Cipher Suites to Transport Layer Security (TLS) , [RFC 2712](#)
 - Addition of SEED Cipher Suites to Transport Layer Security (TLS), [RFC 4162](#)
 - Pre-Shared Key Ciphersuites for Transport Layer Security (TLS), [RFC 4279](#)
 - **The Transport Layer Security (TLS) Protocol - Version 1.1**, [RFC 4346](#)
 - Pre-Shared Key (PSK) Ciphersuites with NULL Encryption for Transport Layer Security (TLS), [RFC 4785](#)
 - Using the Secure Remote Password (SRP) Protocol for TLS Authentication , [RFC 5054](#)
 - **The Transport Layer Security (TLS) Protocol - Version 1.2**, [RFC 5246](#)
 - AES Galois Counter Mode (GCM) Cipher Suites for TLS, [RFC 5288](#)
 - TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM) , [RFC 5289](#)
 - DES and IDEA Cipher Suites for Transport Layer Security (TLS), [RFC 5469](#)
 - Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode, [RFC 5487](#)
 - ECDHE_PSK Cipher Suites for Transport Layer Security (TLS) , [RFC 5489](#)
 - Camellia Cipher Suites for TLS, [RFC 5932](#)
 - Addition of the ARIA Cipher Suites to Transport Layer Security (TLS), [RFC 6209](#)
 - Datagram Transport Layer Security Version 1.2, [RFC 6347](#)
 - Addition of the Camellia Cipher Suites to Transport Layer Security (TLS), [RFC 6367](#)
 - AES-CCM Cipher Suites for Transport Layer Security (TLS), [RFC 6655](#)
 - AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS, [RFC 7251](#)
 - ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS), [RFC 7905](#)
 - Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier, [RFC 8422](#)
 - GOST Cipher Suites for Transport Layer Security (TLS) Protocol Version 1.2, [draft-smyshlyaev-tls12-gost-suites](#)

- **TLS v = 1.3:** <Record Protocol Algos.>
 - The Transport Layer Security (TLS) Protocol Version 1.3, [RFC 8446](#)
 - Secure Password Ciphersuites for Transport Layer Security (TLS), [RFC 8492](#)
 - ShangMi (SM) Cipher Suites for TLS 1.3, [RFC 8998](#)
 - TLS 1.3 Authentication and Integrity only Cipher Suites, [draft-camwinget-tls-ts13-macciphersuites](#)
 - GOST Cipher Suites for Transport Layer Security (TLS) Protocol Version 1.3, [draft-smyshlyaev-tls13-gost-suites](#)

- Ciphersuite identifiers: [Transport Layer Security \(TLS\) Parameters](#)

A symmetric cipher suite defines the pair of the AEAD algorithm and hash algorithm to be used with HKDF. Cipher suite names follow the naming convention:

```
CipherSuite TLS_AEAD_HASH = VALUE;
```

Component	Contents
TLS	The string "TLS"
AEAD	The AEAD algorithm used for record protection
HASH	The hash algorithm used with HKDF
VALUE	The two-byte ID assigned for this cipher suite

This specification defines the following cipher suites for use with TLS 1.3.

Description	Value
TLS_AES_128_GCM_SHA256	{0x13,0x01}
TLS_AES_256_GCM_SHA384	{0x13,0x02}
TLS_CHACHA20_POLY1305_SHA256	{0x13,0x03}
TLS_AES_128_CCM_SHA256	{0x13,0x04}
TLS_AES_128_CCM_8_SHA256	{0x13,0x05}

« Voransicht als Mitglied schließen

Voransicht als Mitglied

Network Security

Aktionen

Inhalt

Timeline

Info

Lernfortschritt

FOREN

NWS Forum 2021

Platform for all information, hints and discussions in the context of the NWS lecture of the summer term 2021

Beiträge (Ungelesen): 6 (0)

Letzter Beitrag: Q1: A detailed description of the fi... von Stefan Heiss (sheiss), Gestern, 13:37

Lectures

NWS 2021 (Labs, Lecture Notes,...)

Exams

Kalender

Mai 2021

KW Mo Di Mi Do Fr Sa So

17 1 2

18 3 4 5 6 7 8 9

19 10 11 12 13 14 15 16

20 17 18 19 20 21 22 23

21 24 25 26 27 28 29 30

22 31

*WLAN

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tls

No.	Time	Source	Destination	Protocol	Length	Info
2	0.237892	193.174.116.155	192.168.178.61	TLSv1.2	100	Application Data
11	2.944917	192.168.178.61	193.174.116.151	TLSv1.2	571	Client Hello
15	2.992311	193.174.116.151	192.168.178.61	TLSv1.2	1506	Server Hello
19	3.034612	193.174.116.151	192.168.178.61	TLSv1.2	1360	Certificate, Server Key Exchange, Server
21	3.046468	192.168.178.61	193.174.116.151	TLSv1.2	180	Client Key Exchange, Change Cipher Spec,
22	3.048256	192.168.178.61	193.174.116.151	TLSv1.2	500	Application Data
24	3.087243	193.174.116.151	192.168.178.61	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake
26	3.121316	193.174.116.151	192.168.178.61	TCP	457	443 → 63561 [ACK] Seq=5714 Ack=1090 Win=
32	3.123269	193.174.116.151	192.168.178.61	TLSv1.2	72	Application Data
33	3.123269	193.174.116.151	192.168.178.61	TCP	68	443 → 63561 [ACK] Seq=10096 Ack=1090 Win=
45	3.169291	193.174.116.151	192.168.178.61	TLSv1.2	84	Application Data [TCP segment of a reass
61	3.206796	193.174.116.151	192.168.178.61	TLSv1.2	535	Application Data [TCP segment of a reass
62	3.207428	193.174.116.151	192.168.178.61	TLSv1.2	1075	Application Data [TCP segment of a reass
74	3.217301	193.174.116.151	192.168.178.61	TLSv1.2	84	Application Data [TCP segment of a reass
77	3.217301	193.174.116.151	192.168.178.61	TLSv1.2	84	Application Data [TCP segment of a reass
99	3.247803	193.174.116.151	192.168.178.61	TLSv1.2	535	Application Data [TCP segment of a reass
105	3.251676	192.168.178.61	216.58.213.238	TLSv1.3	571	Client Hello
110	3.253351	193.174.116.151	192.168.178.61	TLSv1.2	119	Application Data [TCP segment of a reass

Transmission Control Protocol, Src Port: 63561, Dst Port: 443, Seq: 1, Ack: 517

Transport Layer Security

TLSv1.2 Record Layer: Handshake Protocol: Client Hello

Content Type: Handshake (22)

Version: TLS 1.0 (0x0301)

Length: 512

Handshake Protocol: Client Hello

Handshake Type: Client Hello (1)

Length: 508

Version: TLS 1.2 (0x0303)

Random: 87bef8b8bcb3af5865975beb14d5ad3d186af94c493d29105559377ca178be83

GMT Unix Time: Mar 3, 2042 07:07:52.000000000 Mittteleuropäische Zeit

Random Bytes: bcb3af5865975beb14d5ad3d186af94c493d29105559377ca178be83

0040 03 87 be f8 b8 bc b3 af 58 65 97 5b eb 14 d5 ad Xe-[....

0050 3d 18 6a f9 4c 49 3d 29 10 55 59 37 7c a1 78 be ..j-LI=) -UY7]x

0060 83 20 e1 9f 97 4f 0b be 6d c5 2a c1 1c 07 1b f0m*.....

0070 9d da f2 5c 55 a1 ea 27 50 60 b5 0e 45 c1 62 da ...U...P...E-b

0080 98 1b 00 24 13 01 13 03 13 02 c0 2b c0 2f cc a9 ...\$.+./..

0090 cc a8 c0 2c c0 30 c0 0a c0 09 c0 13 c0 14 00 9c ...,.0.....

Random values used for deriving keys (tls.handshake.random), 32 bytes

Packets: 582 · Displayed: 184 (31.6%)

Profile: Default

Prof. Heiss, 17.06.2022

NWS - TLS

11

- Use a web browser that allows key logging, see: [NSS Key Log Format](#)
- Enable key logging by setting the environment variable SSLKEYLOGFILE to point to a file.
- Start Wireshark to capture a TLS session and filter for “tls”
- To decrypt captured TLS packets right click on a TLS packet and choose
 - Protocol Preferences -> Transport Layer Security -> Open Transport Layer Security preferences...
 - Add the name (complete path) of the file with the key logs at: “(Pre)-Master-Secret log filename”

Decrypting a captured TLS session

The screenshot shows the Wireshark interface with a packet capture of a TLS session. The packet list on the left shows several packets, with packet 44 selected. A context menu is open over packet 44, and the option 'Open Transport Layer Security preferences...' is highlighted. The packet details pane on the right shows the structure of the selected packet, including the Transport Layer Security section.

No.	Time	Source	Destination	Protocol	Length	Info
42	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	85	Encrypted Alert
44	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	571	Client Hello
48	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	1506	Server Hello
52	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	1360	Certificate, Server Key Exchange, Server Hello Done
54	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake
55	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	1548	Application Data
57	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
64	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	93	Application Data
65	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	93	Application Data
66	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	93	Application Data
67	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	93	Application Data
70	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	96	Application Data
72	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	96	Application Data
74	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	96	Application Data
75	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	96	Application Data
78	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	571	Client Hello
84	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	1506	Server Hello
88	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	1360	Certificate, Server Key Exchange, Server Hello Done
90	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake
91	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	967	Application Data
93	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
95	3.014785	192.168.178.61	193.174.116.151	TLSv1.2	526	Application Data

Transport Layer Security: Protocol

Packets: 916 · Displayed: 326 (35.6%) · Dropped: 0 (0.0%) · Profile: Default

The screenshot shows the Wireshark Preferences dialog box, specifically the Transport Layer Security tab. The left pane lists various protocols, with TLS selected. The right pane shows the configuration options for TLS, including the RSA keys list, TLS debug file, and checkboxes for reassembling TLS records and application data.

Wireshark · Preferences

TCPROS
TDMoE
TDMoP
TDS
TeamSpeak2
TECMP
TELNET
Teredo
TETRA
TFP
TFTP
Thread
Thrift
Tibia
TIME
TIPC
TiVoConnect
TLS
TNS

Transport Layer Security

RSA keys list

TLS debug file

☒ Reassemble TLS records spanning multiple TCP segments

☒ Reassemble TLS Application Data spanning multiple TLS records

☐ Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared Key

(Pre)-Master-Secret log filename

The image shows a Wireshark capture of a TLS 1.3 Client Hello packet. The packet list at the top shows the following entries:

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212....	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Cipher Spec, Encrypted Extensions
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] Seq=1251 Ack=518 Win=501 Len=1250 [TCP segment of a re...
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate Verify, Finished
5 0.027442	212....	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027697	212....	23.57.29.178	HTTP	568	GET /en/java/javase/13/docs/api/index.html HTTP/1.1

The packet details pane shows the following structure for the selected packet:

- ▼ TLSv1.3 Record Layer: Handshake Protocol: Client Hello
 - Content Type: Handshake (22)
 - Version: TLS 1.0 (0x0301)
 - Length: 512
 - ▼ Handshake Protocol: Client Hello
 - Handshake Type: Client Hello (1)
 - Length: 508
 - Version: TLS 1.2 (0x0303)
 - Random: eefb571fc25208d233e8697990e4bdbb239b57c42f0398f964344559eaf11e35
 - Session ID Length: 32
 - Session ID: 48439ab97bcdd5a7dac22ba43418d590b34ea569989b332c90c409624c3d5b90
 - Cipher Suites Length: 34
 - > Cipher Suites (17 suites)
 - Compression Methods Length: 1
 - > Compression Methods (1 method)
 - Extensions Length: 401
 - > Extension: server name (len=20)

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0030 02 00 6c 2d 00 00 16 03 01 02 00 01 00 01 fc 03  ..1-..-...
0040 03 ee fb 57 1f c2 52 08 d2 33 e8 69 79 90 e4 bd  ...W..R..3.iy..
0050 bb 23 9b 57 c4 2f 03 98 f9 64 34 45 59 ea f1 1e  .#.W./..d4EY..
0060 35 20 48 43 9a b9 7b cd d5 a7 da c2 2b a4 34 18  5 HC..{. ....+4.
0070 d5 90 b3 4e a5 69 98 9b 33 2c 90 c4 09 62 4c 3d  ...N.i.. 3,...bL=
0080 5b 90 00 22 13 01 13 03 13 02 c0 2b c0 2f cc a9  [...". ....+./..
```

The status bar at the bottom indicates: Record Layer (tls.record), 517 Bytes | Pakete: 34 · Angezeigt: 34 (100.0%) | Profil: Default

The image shows a Wireshark capture of a TLS 1.3 Server Hello message. The packet list at the top shows a sequence of events: Client Hello, Server Hello, TCP segment, Certificate, Change Cipher Spec, and an HTTP GET request. The selected packet is the Server Hello (packet 2), which is expanded to show its structure: Content Type (Handshake), Version (TLS 1.2), Length (122), Handshake Type (Server Hello), and various fields including Random, Session ID, Cipher Suite, and Extensions. The packet details pane shows the raw bytes of the message, with the first 35 bytes highlighted in blue. The status bar at the bottom indicates that the packet is 1304 bytes long and contains 35 bytes of decrypted TLS data.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212....	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Cipher Spec, Encrypted Extensions
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] Seq=1251 Ack=518 Win=501 Len=1250 [TCP segment of a re...
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate Verify, Finished
5 0.027442	212....	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027697	212....	23.57.29.178	HTTP	568	GET /en/java/javase/13/docs/api/index.html HTTP/1.1

▼ TLSv1.3 Record Layer: Handshake Protocol: Server Hello
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 122
▼ Handshake Protocol: Server Hello
Handshake Type: Server Hello (2)
Length: 118
Version: TLS 1.2 (0x0303)
Random: 73b3eeab9ddfc1cbab9b166871ed9bf8fa8235d03edc4094e432d46785a56765
Session ID Length: 32
Session ID: 48439ab97bcdd5a7dac22ba43418d590b34ea569989b332c90c409624c3d5b90
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
Compression Method: null (0)
Extensions Length: 46
► Extension: supported_versions (len=2)
► Extension: key_share (len=36)
TLS Fullstring: 771 4866 43-511

0030 01 f5 9c bf 00 00 16 03 03 00 7a 02 00 00 76 03z...v.
0040 03 73 b3 ee ab 9d df c1 cb ab 9b 16 68 71 ed 9b .s.....hq..
0050 f8 fa 82 35 d0 3e dc 40 94 e4 32 d4 67 85 a5 67 ...5>.@..2.g.g
0060 65 20 48 43 9a b9 7b cd d5 a7 da c2 2b a4 34 18 e HC..{.+4.

Frame (1304 bytes) Decrypted TLS (35 bytes)
Record Layer (tls.record), 127 Bytes || Pakete: 34 · Angezeigt: 34 (100.0%) || Profil: Default

TLS 1.3 Capture – (Server) Change Cipher Spec

The image shows a Wireshark capture of a TLS 1.3 handshake. The packet list on the left shows six packets. Packet 5, at time 0.027442, is a TLSv1.3 Change Cipher Spec, Finished message from 212.100.100.100 to 23.57.29.178. The packet details pane on the right shows the structure of this message: TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec, Content Type: Change Cipher Spec (20), Version: TLS 1.2 (0x0303), Length: 1, Change Cipher Spec Message. The packet bytes pane at the bottom shows the raw data, with the first byte of the message (0x03) highlighted in blue.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212.100.100.100	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.57.29.178	212.100.100.100	TLSv1.3	1304	Server Hello, Change Cipher Spec, Encrypted Extensions
3 0.009679	23.57.29.178	212.100.100.100	TCP	1304	443 → 56798 [PSH, ACK] Seq=1251 Ack=518 Win=501 Len=1250 [TCP segment of a re...
4 0.009679	23.57.29.178	212.100.100.100	TLSv1.3	1139	Certificate, Certificate Verify, Finished
5 0.027442	212.100.100.100	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027607	212.100.100.100	23.57.29.178	HTTP	568	GET /en/java/javase/13/docs/api/index.html HTTP/1.1

Compression Method: null (0)
Extensions Length: 46
Extension: supported_versions (len=2)
Type: supported_versions (43)
Length: 2
Supported Version: TLS 1.3 (0x0304)
Extension: key_share (len=36)
Type: key_share (51)
Length: 36
Key Share extension
[JA3S Fullstring: 771,4866,43-51]
[JA3S: 15af977ce25de452b96affa2addb1036]
TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
Content Type: Change Cipher Spec (20)
Version: TLS 1.2 (0x0303)
Length: 1
Change Cipher Spec Message
TLSv1.3 Record Layer: Handshake Protocol: Encrypted Extensions

00b0 d2 34 fc 25 5f 14 03 03 00 01 01 17 03 03 00 34 -4-%4
00c0 3f 8d 96 68 2c c8 8a 54 ae a7 b0 7d dd fd d9 d3 ?-h,-T ...}....
00d0 6b 7f 27 66 12 87 0a 30 8f a4 f1 26 c6 88 a7 ca k-'f...0 ...&....

Frame (1304 bytes) Decrypted TLS (35 bytes)
Record Layer (tls.record), 6 Bytes || Pakete: 34 · Angezeigt: 34 (100.0%) || Profil: Default

TLS 1.3 Capture – Encrypted Extensions

The image shows a Wireshark capture of a TLS 1.3 handshake. The packet list on the left shows the following packets:

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212....	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Cipher Spec, Encrypted Extensions
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] Seq=1251 Ack=518 Win=501 Len=1250 [TCP segment of a re...
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate Verify, Finished
5 0.027442	212....	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027697	212....	23.57.29.178	HTTP	568	GET /en/java/javase/13/docs/api/index.html HTTP/1.1

The packet details pane on the left shows the following structure for the selected packet (2):

- TLSv1.3 Record Layer: Handshake Protocol: Server Hello
- TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
- TLSv1.3 Record Layer: Handshake Protocol: Encrypted Extensions
 - Opaque Type: Application Data (23)
 - Version: TLS 1.2 (0x0303)
 - Length: 52
 - [Content Type: Handshake (22)]
 - Handshake Protocol: Encrypted Extensions
 - Handshake Type: Encrypted Extensions (8)
 - Length: 31
 - Extensions Length: 29
 - Extension: server_name (len=0)
 - Extension: supported_groups (len=6)
 - Extension: application_layer_protocol_negotiation (len=11)

The packet bytes pane on the left shows the raw data for the selected packet (2):

```
00b0 d2 34 fc 25 5f 14 03 03 00 01 01 17 03 03 00 34  .4.%... ..4
00c0 3f 8d 96 68 2c c8 8a 54 ae a7 b0 7d dd fd d9 d3  ?..h,..T ...}....
00d0 6b 7f 27 66 12 87 0a 30 8f a4 f1 26 c6 88 a7 ca  k.'f...0 ...&....
00e0 30 86 37 df 5d 95 38 e2 6d 47 85 15 fe 5a bd 55  0.7.]~8. mG...Z.U
00f0 bb 24 ce 07 17 03 03 0c 8f 98 06 df 8b ef dd d7  .$.~... ..
0100 62 4a 97 7d 30 b5 c8 a3 15 03 4a 07 7e 6e a9 a6  bJ.}0... ..J~n..
0110 bc 1f 2f a7 fa 90 45 a4 5b 29 2f e4 5e 85 62 9b  ../...E. [)/.^~b.
```

The packet list on the right shows the following packets:

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212....	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Cipher Spec, Encrypted Extensions
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] Seq=1251 Ack=518 Win=501 Len=1250 [TCP segment of a re...
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate Verify, Finished
5 0.027442	212....	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027697	212....	23.57.29.178	HTTP	568	GET /en/java/javase/13/docs/api/index.html HTTP/1.1

The packet details pane on the right shows the following structure for the selected packet (2):

- TLSv1.3 Record Layer: Handshake Protocol: Server Hello
- TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
- TLSv1.3 Record Layer: Handshake Protocol: Encrypted Extensions
 - Opaque Type: Application Data (23)
 - Version: TLS 1.2 (0x0303)
 - Length: 52
 - [Content Type: Handshake (22)]
 - Handshake Protocol: Encrypted Extensions
 - Handshake Type: Encrypted Extensions (8)
 - Length: 31
 - Extensions Length: 29
 - Extension: server_name (len=0)
 - Extension: supported_groups (len=6)
 - Extension: application_layer_protocol_negotiation (len=11)

The packet bytes pane on the right shows the raw data for the selected packet (2):

```
00b0 d2 34 fc 25 5f 14 03 03 00 01 01 17 03 03 00 34  .4.%... ..4
00c0 3f 8d 96 68 2c c8 8a 54 ae a7 b0 7d dd fd d9 d3  ?..h,..T ...}....
00d0 6b 7f 27 66 12 87 0a 30 8f a4 f1 26 c6 88 a7 ca  k.'f...0 ...&....
00e0 30 86 37 df 5d 95 38 e2 6d 47 85 15 fe 5a bd 55  0.7.]~8. mG...Z.U
00f0 bb 24 ce 07 17 03 03 0c 8f 98 06 df 8b ef dd d7  .$.~... ..
0100 62 4a 97 7d 30 b5 c8 a3 15 03 4a 07 7e 6e a9 a6  bJ.}0... ..J~n..
0110 bc 1f 2f a7 fa 90 45 a4 5b 29 2f e4 5e 85 62 9b  ../...E. [)/.^~b.
```


The image shows a Wireshark capture of a TLS 1.3 handshake. The packet list at the top shows six packets. Packet 4, at time 0.009679, is a TLSv1.3 Certificate message (length 1139 bytes) from 212.57.29.178 to 23.57.29.178. The packet details pane shows the structure of the TLSv1.3 Record and the Handshake Protocol: Certificate message. The Handshake Protocol: Certificate message has a length of 3194 bytes and contains a Certificate Request Context Length of 0, a Certificates Length of 3190, and a list of certificates. The first certificate is from 308205f1308204d9a003020102021001887bd706f662005734fbfd6c302135300d06092a... (id-at-commonName=www-ww.oracle...). The second certificate is from 308204943082037ca003020102021001fda3eb6eca75c888438b724bcfb91300d06092a... (id-at-commonName=DigiCert SHA2). The packet bytes pane shows the raw data of the handshake message.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212.57.29.178	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.57.29.178	212.201.96.89	TLSv1.3	1304	Server Hello, Change Cipher Spec, Encrypted Extensions
3 0.009679	23.57.29.178	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] Seq=1251 Ack=518 Win=501 Len=1250 [TCP segment of a re...
4 0.009679	23.57.29.178	212.201.96.89	TLSv1.3	1139	Certificate, Certificate Verify, Finished
5 0.027442	212.57.29.178	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027607	212.57.29.178	23.57.29.178	HTTP	568	GET /en/java/javase/13/docs/api/index.html HTTP/1.1

▼ TLSv1.3 Record Layer: Handshake Protocol: Certificate
Opaque Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 3215
[Content Type: Handshake (22)]
▼ Handshake Protocol: Certificate
Handshake Type: Certificate (11)
Length: 3194
Certificate Request Context Length: 0
Certificates Length: 3190
▼ Certificates (3190 bytes)
Certificate Length: 1525
> Certificate: 308205f1308204d9a003020102021001887bd706f662005734fbfd6c302135300d06092a... (id-at-commonName=www-ww.oracle...)
Extensions Length: 479
> Extension: status_request (len=475)
Certificate Length: 1176
> Certificate: 308204943082037ca003020102021001fda3eb6eca75c888438b724bcfb91300d06092a... (id-at-commonName=DigiCert SHA2...)
Extensions Length: 0
> Transport Layer Security

0000 0b 00 0c 7a 00 00 0c 76 00 05 f5 30 82 05 f1 30 ...z...v ...0...0
0010 82 04 d9 a0 03 02 01 02 02 10 01 88 7b d7 06 f6{...
0020 62 00 57 34 fb fd 6c 30 21 35 30 0d 06 09 2a 86 b.W4..l0 !50...*

Frame (1139 bytes) Reassembled TCP (3220 bytes) Decrypted TLS (3198 bytes) Decrypted TLS (79 bytes) Decrypted TLS (52 bytes)
Handshake protocol message (tls.handshake), 3.198 Bytes || Package: 34 · Anzeigt: 34 (100.0%) || Profil: Default

The image shows a Wireshark capture of a TLS 1.3 handshake. The packet list on the left shows six packets. Packet 4, at time 0.009679, is a TLSv1.3 record containing a Certificate Verify message. The packet details pane on the right shows the structure of this message: it is a TLSv1.3 Record Layer Handshake Protocol Certificate Verify with an opaque type of Application Data (23), version TLS 1.2 (0x0303), and a length of 96 bytes. The content type is Handshake (22). The handshake type is Certificate Verify (15) with a length of 75 bytes. The signature algorithm is ecdsa_secp256r1_sha256 (0x0403), using a SHA256 hash algorithm (4 bytes) and an ECDSA signature (3 bytes). The signature length is 71 bytes, and the signature itself is a long hexadecimal string. The packet bytes pane at the bottom shows the raw data of the handshake protocol message, with hex and ASCII representations.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212....	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Cipher Spec, Encrypted Extensions
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] Seq=1251 Ack=518 Win=501 Len=1250 [TCP segment of a re...
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate Verify, Finished
5 0.027442	212....	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027607	212...	23.57.29.178	HTTP	568	GET /en/java/javase/13/docs/api/index.html HTTP/1.1

Transmission Control Protocol, Src Port: 443, Dst Port: 56798, Seq: 2501, Ack: 518, Len: 1085
> [3 Reassembled TCP Segments (3220 bytes): #2(1060), #3(1250), #4(910)]
> Transport Layer Security
▼ Transport Layer Security
 ▼ TLSv1.3 Record Layer: Handshake Protocol: Certificate Verify
 Opaque Type: Application Data (23)
 Version: TLS 1.2 (0x0303)
 Length: 96
 [Content Type: Handshake (22)]
 ▼ Handshake Protocol: Certificate Verify
 Handshake Type: Certificate Verify (15)
 Length: 75
 ▼ Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
 Signature Hash Algorithm Hash: SHA256 (4)
 Signature Hash Algorithm Signature: ECDSA (3)
 Signature length: 71
 Signature: 304502210090eaa694f72bbb012e4219acba3703d98d6da0eb1b89e26b0ce6032e1bd3c0...
 > TLSv1.3 Record Layer: Handshake Protocol: Finished

0000 0f 00 00 4b 04 03 00 47 30 45 02 21 00 90 ea a6 ...K...G 0E.!...
0010 94 f7 2b bb 01 2e 42 19 ac ba 37 03 d9 8d 6d a0 ...+...B...7...m...
0020 eb 1b 89 e2 6b 0c e6 03 2e 1b d3 c0 f9 02 20 72k...

Frame (1139 bytes) | Reassembled TCP (3220 bytes) | Decrypted TLS (3198 bytes) | Decrypted TLS (79 bytes) | Decrypted TLS (52 bytes)
Handshake protocol message (tls.handshake), 79 Bytes | Pakete: 34 · Angezeigt: 34 (100.0%) | Profil: Default

TLS 1.3 Capture – (Server) Finished

The image shows a Wireshark capture of a TLS 1.3 handshake. The packet list on the left shows six packets. Packet 4, at time 0.009679, is a TLSv1.3 'Certificate, Certificate Verify, Finished' message from 23.57.29.178 to 212.201.96.89. The packet details pane on the right shows the structure of this message: a Handshake Protocol: Certificate Verify (15) with a length of 75, followed by a Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403) with a length of 71. The Signature is 304502210090eaa694f72bbb012e4219acba3703d98d6da0eb1b89e26b0ce6032e1bd3c0... Below this is the TLSv1.3 Record Layer: Handshake Protocol: Finished (23) with a length of 69. The details pane also shows the Handshake Protocol: Finished (20) with a length of 48 and Verify Data. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII. The status bar at the bottom indicates the selected packet is a Handshake protocol message (tls.handshake), 52 Bytes, and shows the packet count and profile.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212....	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Cipher Spec, Encrypted Extensions
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] Seq=1251 Ack=518 Win=501 Len=1250 [TCP segment of a re...
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate Verify, Finished
5 0.027442	212....	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027607	212...	23.57.29.178	HTTP	568	GET /en/java/javase/13/docs/api/index.html HTTP/1.1

[Content Type: Handshake (22)]

- Handshake Protocol: Certificate Verify (15)
 - Handshake Type: Certificate Verify (15)
 - Length: 75
 - Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
 - Signature Hash Algorithm Hash: SHA256 (4)
 - Signature Hash Algorithm Signature: ECDSA (3)
 - Signature length: 71
 - Signature: 304502210090eaa694f72bbb012e4219acba3703d98d6da0eb1b89e26b0ce6032e1bd3c0...
- TLSv1.3 Record Layer: Handshake Protocol: Finished (23)
 - Opaque Type: Application Data (23)
 - Version: TLS 1.2 (0x0303)
 - Length: 69
 - [Content Type: Handshake (22)]
 - Handshake Protocol: Finished (20)
 - Handshake Type: Finished (20)
 - Length: 48
 - Verify Data

0000 14 00 00 30 98 48 be da c4 6f a6 dc f5 04 85 5d ...0.H...o....]

0010 04 77 ce 24 d8 6c 88 8a fe 27 96 4b 2e c1 df c0 .w\$.l..'.K....

0020 b7 03 ad 08 af 13 ed fc 0e 3a 2e 23 45 c5 47 2f:.#E.G/

Frame (1139 bytes) Reassembled TCP (3220 bytes) Decrypted TLS (3198 bytes) Decrypted TLS (79 bytes) Decrypted TLS (52 bytes)

Handshake protocol message (tls.handshake), 52 Bytes || Pakete: 34 · Angezeigt: 34 (100.0%) || Profil: Default

TLS 1.3 Capture – (Client) Change Cipher Spec

The image shows a Wireshark packet capture of a TLS 1.3 session. The packet list on the left shows a 'Change Cipher Spec, Finished' message (packet 5) from the client to the server. The packet details pane on the right shows the structure of this message, including the 'Change Cipher Spec Message' and the 'Handshake Protocol: Finished' message. The packet bytes pane at the bottom shows the raw data of the message, including the 'Change Cipher Spec Message' and the 'Handshake Protocol: Finished' message.

Packet 5: 0.027442 212.100.100.100 23.57.29.178 TLSv1.3 134 Change Cipher Spec, Finished

Packet Details:

- Transmission Control Protocol, Src Port: 56798, Dst Port: 443, Seq: 518, Ack: 3586, Len: 80
- Transport Layer Security
 - TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
 - Content Type: Change Cipher Spec (20)
 - Version: TLS 1.2 (0x0303)
 - Length: 1
 - Change Cipher Spec Message
 - TLSv1.3 Record Layer: Handshake Protocol: Finished
 - Opaque Type: Application Data (23)
 - Version: TLS 1.2 (0x0303)
 - Length: 69
 - [Content Type: Handshake (22)]
 - Handshake Protocol: Finished
 - Handshake Type: Finished (20)
 - Length: 48
 - Verify Data

Packet Bytes:

0030 02 00 6a 78 00 00 14 03 03 00 01 01 17 03 03 00 ..jx...

0040 45 8f 65 35 b4 ad 41 a3 84 13 52 49 7b e8 74 78 E-e5-A-..RI{.tx

0050 32 19 95 e8 dc 0e 38 3a e6 66 0f 54 88 03 99 cb 2-...8: -f.T...

0060 4f 2d 28 00 07 4b 17 75 c5 83 ea 57 35 50 b6 b0 0-(..K.u ...W5P..

0070 fc fe d5 80 af a7 97 b4 51 1b 44 06 d9 ef 55 19 Q.D...U..

0080 57 29 cd a9 b9 b1 W)....

Frame (134 bytes) Decrypted TLS (52 bytes)

Record Layer (tls.record), 6 Bytes

Pakete: 34 · Angezeigt: 34 (100.0%) Profil: Default

TLS 1.3 Capture – (Client) Finished

The image shows a Wireshark packet capture of a TLS 1.3 handshake. The packet list on the left shows six packets. Packet 5, at time 0.027442, is a TLSv1.3 Change Cipher Spec, Finished, from 212.12.201.96.89 to 23.57.29.178. Packet 6, at time 0.027607, is an HTTP GET request for /en/java/javase/13/docs/api/index.html. The packet details pane for packet 6 shows the Transport Layer Security section expanded, revealing the TLSv1.3 Record Layer: Handshake Protocol: Finished. The handshake type is Finished (20) with a length of 48 bytes. The verify data is shown as a5 78 87 11. The packet bytes pane shows the raw data of the handshake message, which is 52 bytes long. The status bar at the bottom indicates that the packet is a Handshake protocol message (tls.handshake), 52 Bytes long, and that 34 packets are displayed (100.0%).

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212.12.201.96.89	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.57.29.178	212.12.201.96.89	TLSv1.3	1304	Server Hello, Change Cipher Spec, Encrypted Extensions
3 0.009679	23.57.29.178	212.12.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] Seq=1251 Ack=518 Win=501 Len=1250 [TCP segment of a re...
4 0.009679	23.57.29.178	212.12.201.96.89	TLSv1.3	1139	Certificate, Certificate Verify, Finished
5 0.027442	212.12.201.96.89	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027607	212.12.201.96.89	23.57.29.178	HTTP	568	GET /en/java/javase/13/docs/api/index.html HTTP/1.1

Transmission Control Protocol, Src Port: 56798, Dst Port: 443, Seq: 518, Ack: 3586, Len: 80

Transport Layer Security

- TLsv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
 - Content Type: Change Cipher Spec (20)
 - Version: TLS 1.2 (0x0303)
 - Length: 1
 - Change Cipher Spec Message
- TLsv1.3 Record Layer: Handshake Protocol: Finished
 - Opaque Type: Application Data (23)
 - Version: TLS 1.2 (0x0303)
 - Length: 69
 - [Content Type: Handshake (22)]
- Handshake Protocol: Finished
 - Handshake Type: Finished (20)
 - Length: 48
 - Verify Data

0000 14 00 00 30 f6 cc e8 e8 ac d5 52 bc 78 7d 30 cf ...0... ..R.x}0.

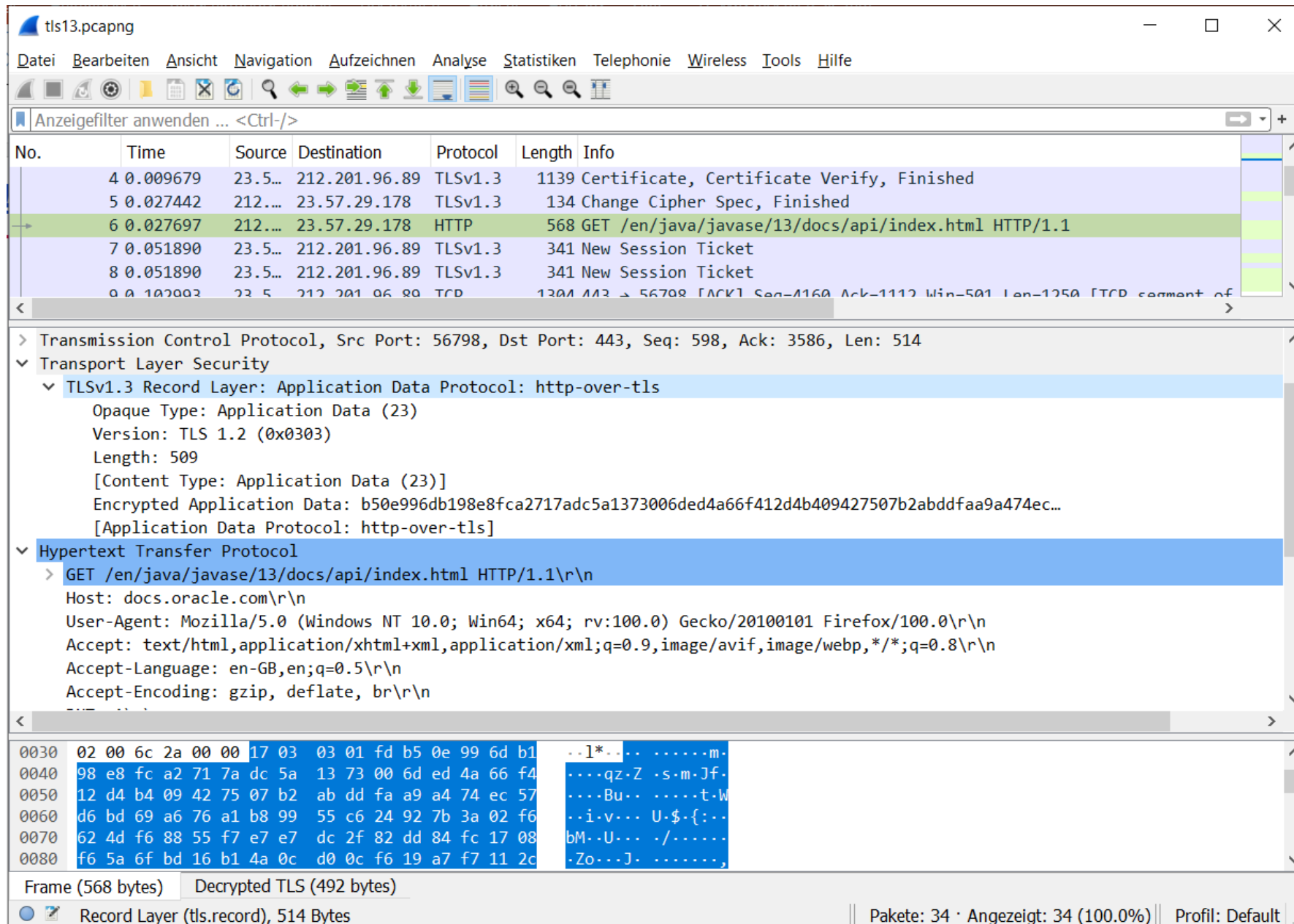
0010 1e d1 d4 50 59 11 fd 51 7f 77 80 e9 c6 01 2e b4 ...PY..Q .w.....

0020 6f eb f2 e3 61 e7 92 6a e1 71 e1 de 54 ce 7c 6d o...a..j .q..T..m

0030 a5 78 87 11 ..x..

Frame (134 bytes) Decrypted TLS (52 bytes)

Handshake protocol message (tls.handshake), 52 Bytes || Pakete: 34 · Angezeigt: 34 (100.0%) || Profil: Default



The image shows a Wireshark capture of a TLS 1.3 session. The packet list on the left shows a GET request for /en/java/javase/13/docs/api/index.html. The packet details pane on the right shows the TLS record layer and the application data protocol. The packet bytes pane at the bottom shows the raw data of the application data protocol.

Frame 6 (568 bytes) on interface (eth0) (23.57.29.178) contains:

- Transmission Control Protocol, Src Port: 56798, Dst Port: 443, Seq: 598, Ack: 3586, Len: 514
- Transport Layer Security
 - TLSv1.3 Record Layer: Application Data Protocol: http-over-tls
 - Opaque Type: Application Data (23)
 - Version: TLS 1.2 (0x0303)
 - Length: 509
 - [Content Type: Application Data (23)]
 - Encrypted Application Data: b50e996db198e8fca2717adc5a1373006ded4a66f412d4b409427507b2abddfaa9a474ec...
 - [Application Data Protocol: http-over-tls]
 - Hypertext Transfer Protocol
 - GET /en/java/javase/13/docs/api/index.html HTTP/1.1\r\n
 - Host: docs.oracle.com\r\n
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:100.0) Gecko/20100101 Firefox/100.0\r\n
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
 - Accept-Language: en-GB,en;q=0.5\r\n
 - Accept-Encoding: gzip, deflate, br\r\n

Frame (568 bytes) | Decrypted TLS (492 bytes)

Record Layer (tls.record), 514 Bytes | Pakete: 34 · Angezeigt: 34 (100.0%) | Profil: Default

The image shows a Wireshark packet capture of a TLS 1.3 handshake. The packet list at the top shows several packets, with packet 7 (0.051890) selected, which is a TLSv1.3 New Session Ticket. The packet details pane shows the structure of the New Session Ticket, including the Handshake Protocol, TLS Session Ticket, and various fields like Session Ticket Lifetime Hint, Session Ticket Age Add, Session Ticket Nonce, and Session Ticket. The packet bytes pane shows the raw data of the packet, with the first 265 bytes being the decrypted TLS message.

Time	Source	Destination	Protocol	Length	Info
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate Verify, Finished
5 0.027442	212....	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027697	212....	23.57.29.178	HTTP	568	GET /en/java/javase/13/docs/api/index.html HTTP/1.1
7 0.051890	23.5...	212.201.96.89	TLSv1.3	341	New Session Ticket
8 0.051890	23.5...	212.201.96.89	TLSv1.3	341	New Session Ticket

▼ TLSv1.3 Record Layer: Handshake Protocol: New Session Ticket
Opaque Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 282
[Content Type: Handshake (22)]

▼ Handshake Protocol: New Session Ticket
Handshake Type: New Session Ticket (4)
Length: 261

▼ TLS Session Ticket
Session Ticket Lifetime Hint: 83100 seconds (23 hours, 5 minutes)
Session Ticket Age Add: 471552146
Session Ticket Nonce Length: 8
Session Ticket Nonce: 0000000000000000
Session Ticket Length: 240
Session Ticket: 0000af5aa2c69fc63e32a423afbec7eb7f80f99e1e35d5810e9b6e616c164b2282faac6...
Extensions Length: 0

0000 04 00 01 05 00 01 44 9c 1c 1b 50 92 08 00 00 00D. ..P.....
0010 00 00 00 00 00 00 f0 00 00 0a f5 aa 2c 69 fc 63 ,i.c
0020 e3 2a 42 3a fb ec 7e b7 f8 0f 99 e1 e3 5d 58 10 .*B:~.]X.
0030 e9 b6 e6 16 c1 64 b2 28 2f aa c6 a0 69 3c 3b 13d.(/...i<;.
0040 3d 07 d2 0d 41 0a c0 45 eb 66 ba bd e9 1c b6 37 =...A..E .f.....7
0050 74 5d 83 c3 51 97 da b8 65 d8 0f f2 86 97 48 87 t]..Q... e.....H.

Frame (341 bytes) Decrypted TLS (265 bytes)

Handshake protocol message (tls.handshake), 265 Bytes || Pakete: 34 · Angezeigt: 34 (100.0%) || Profil: Default

- **TLSP Plaintext** records
- **length** $\leq 2^{14}$



0x14	change_cipher_spec
------	--------------------

0x15	alert
------	-------

0x16	handshake
------	-----------

0x17	application_data
------	------------------

0x03	0x00	SSL 3.0
------	------	----------------

0x03	0x01	TLS 1.0
------	------	----------------

0x03	0x02	TLS 1.1
------	------	----------------

0x03	0x03	TLS 1.2
------	------	----------------

0x03	0x04	TLS 1.3
------	------	----------------

■ TLSPplaintext records

```
enum {  
    invalid(0),  
    change_cipher_spec(20),  
    alert(21),  
    handshake(22),  
    application_data(23),  
    (255)  
} ContentType;
```

```
struct {  
    ContentType type;  
    ProtocolVersion legacy_record_version;  
    uint16 length;  
    opaque fragment[TLSPplaintext.length];  
} TLSPplaintext;
```



0x14	change_cipher_spec
------	--------------------

0x15	alert
------	-------

0x16	handshake
------	-----------

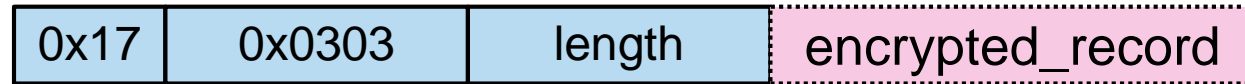
0x17	application_data
------	------------------

0x03	0x03	TLS 1.2
------	------	----------------

0x03	0x01	TLS 1.0
------	------	----------------

■ TLS 1.3: legacy_record_version: 0x0303 / 0x0301 (initial ClientHello)

- **TLSCiphertext** records



- In TLS 1.3, as opposed to previous versions of TLS, all ciphers are modeled as "Authenticated Encryption with Associated Data" (AEAD) [RFC5116].

```
struct {  
    opaque content[TLSPlaintext.length];  
    ContentType type;  
    uint8 zeros[length_of_padding];  
} TLSInnerPlaintext;
```

```
struct {  
    ContentType opaque_type = application_data; /* 23 */  
    ProtocolVersion legacy_record_version = 0x0303; /* TLS v1.2 */  
    uint16 length;  
    opaque encrypted_record[TLSCiphertext.length];  
} TLSCiphertext;
```



■ HandshakeType

0x00	hello_request
0x01	client_hello
0x02	server_hello
0x0B	certificate
0x0C	server_key_exchange
0x0D	certificate_request
0x0E	server_hello_done
0x0F	certificate_verify
0x10	client_key_exchange
0x14	finished



■ HandshakeType

0x01 client_hello

0x02 server_hello

0x0B certificate

0x0D certificate_request

0x0F certificate_verify

0x14 finished



■ HandshakeType

0x01	client_hello
0x02	server_hello
0x04	new_session_ticket
0x05	end_of_early_data
0x08	encrypted_extensions
0x0B	certificate
0x0D	certificate_request
0x0F	certificate_verify
0x14	finished
0x18	key_update
0xFE	message_hash



■ HandshakeType

0x01	client_hello
0x02	server_hello
0x04	new_session_ticket
0x05	end_of_early_data
0x08	encrypted_extensions
0x0B	certificate
0x0D	certificate_request
0x0F	certificate_verify
0x14	finished
0x18	key_update
0xFE	message_hash

0x16	legacy_ver	length	m_t	length	body [length]	
0x17	0x0303	length	m_t	length	body [length]	0x16

Client

0x01 client_hello

Server

0x02 server_hello

0x08 encrypted_extensions

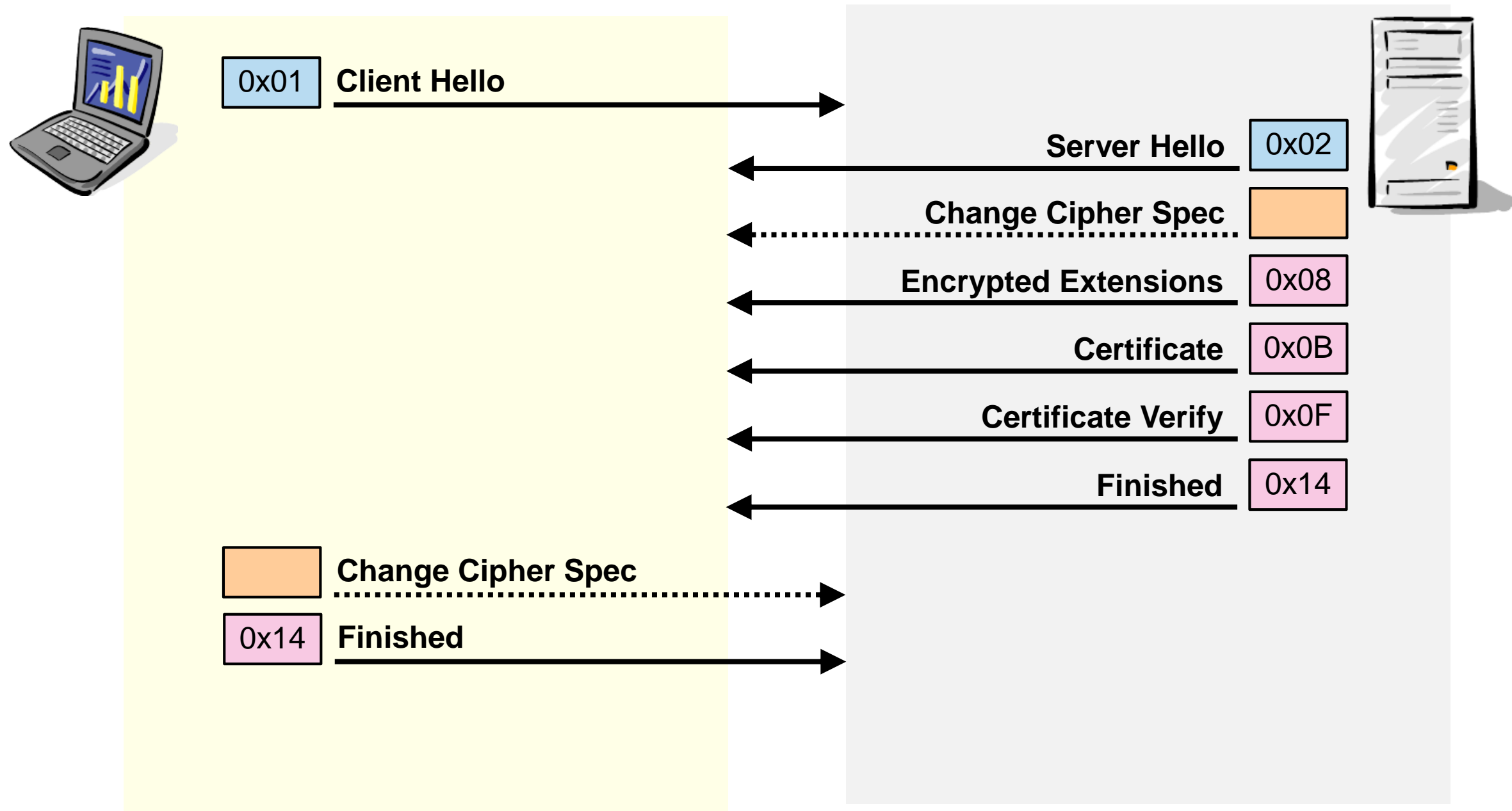
0x0B certificate

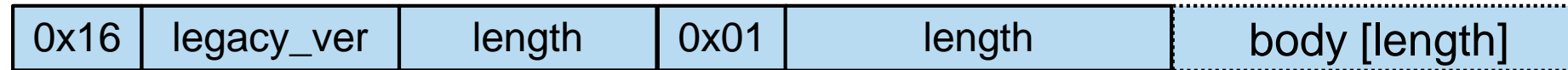
0x0F certificate_verify

0x14 finished

Client

0x14 finished





```
uint16 ProtocolVersion;
opaque Random[32];

uint8 CipherSuite[2];    /* Cryptographic suite selector */

struct {
    ProtocolVersion legacy_version = 0x0303;    /* TLS v1.2 */
    Random random;
    opaque legacy_session_id<0..32>;
    CipherSuite cipher_suites<2..2^16-2>;
    opaque legacy_compression_methods<1..2^8-1>;
    Extension extensions<8..2^16-1>;
} ClientHello;
```

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212....	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certific
5 0.027442	212....	23.57.29.178	TLSv1.3	134	Change Cipher Spec, F
6 0.027697	212....	23.57.29.178	HTTP	568	GET /en/java/javase/1

▼ TLSv1.3 Record Layer: Handshake Protocol: Client Hello

Content Type: Handshake (22)

Version: TLS 1.0 (0x0301)

Length: 512

▼ Handshake Protocol: Client Hello

Handshake Type: Client Hello (1)

Length: 508

Version: TLS 1.2 (0x0303)

Random: eefb571fc25208d233e8697990e4bdbb239b57c42f0398f964344

Session ID Length: 32

Session ID: 48439ab97bcdd5a7dac22ba43418d590b34ea569989b332c9

Cipher Suites Length: 34

> Cipher Suites (17 suites)

Compression Methods Length: 1

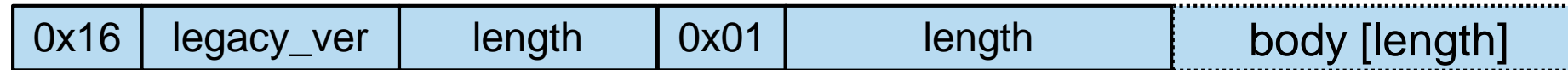
> Compression Methods (1 method)

Extensions Length: 401

> Extension: server_name (len=20)

0030	02 00 6c 2d 00 00 16 03 01 02 00 01 00 01 fc 03	..1-..
0040	03 ee fb 57 1f c2 52 08 d2 33 e8 69 79 90 e4 bd	...W..R..3.iy
0050	bb 23 9b 57 c4 2f 03 98 f9 64 34 45 59 ea f1 1e	..#..W./..d4EY
0060	35 20 48 43 9a b9 7b cd d5 a7 da c2 2b a4 34 18	5 HC..{.+
0070	d5 90 b3 4e a5 69 98 9b 33 2c 90 c4 09 62 4c 3d	...N.i.. 3,...
0080	5b 90 00 22 13 01 13 03 13 02 c0 2b c0 2f cc a9	[..".+..

Record Layer (tls.record), 517 Bytes



```
struct {
    ExtensionType extension_type;
    opaque extension_data<0..2^16-1>;
} Extension;
```

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212.1...	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certific
5 0.027442	212.1...	23.57.29.178	TLSv1.3	134	Change Cipher Spec, F
6 0.027697	212.1...	23.57.29.178	HTTP	568	GET /en/java/javase/1

▼ TLSv1.3 Record Layer: Handshake Protocol: Client Hello

Content Type: Handshake (22)

Version: TLS 1.0 (0x0301)

Length: 512

▼ Handshake Protocol: Client Hello

Handshake Type: Client Hello (1)

Length: 508

Version: TLS 1.2 (0x0303)

Random: eefb571fc25208d233e8697990e4bdbb239b57c42f0398f964344

Session ID Length: 32

Session ID: 48439ab97bcdd5a7dac22ba43418d590b34ea569989b332c9

Cipher Suites Length: 34

> Cipher Suites (17 suites)

Compression Methods Length: 1

> Compression Methods (1 method)

Extensions Length: 401

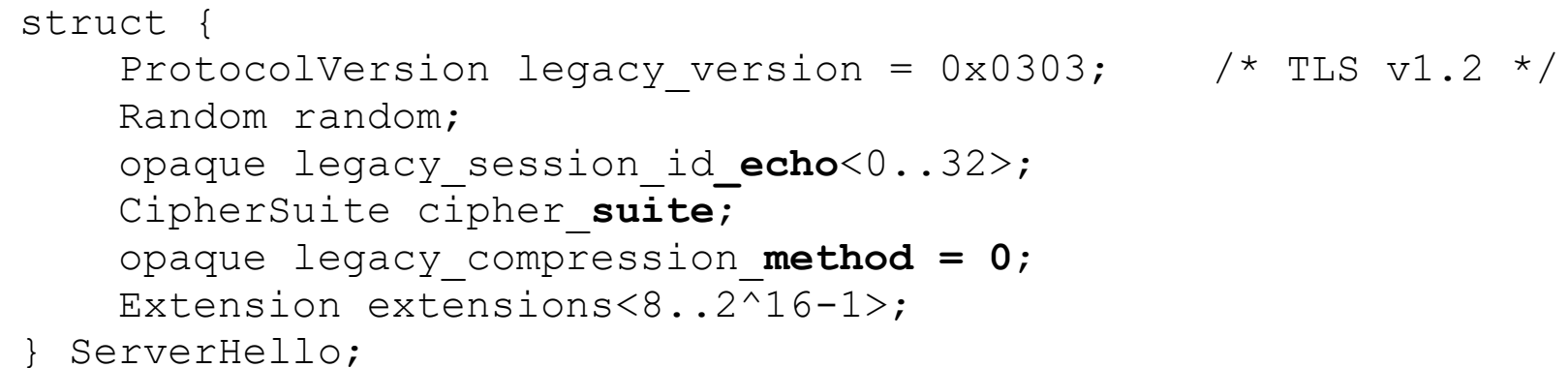
> Extension: server name (len=20)

0030	02 00 6c 2d 00 00 16 03 01 02 00 01 00 01 fc 03	..1-..
0040	03 ee fb 57 1f c2 52 08 d2 33 e8 69 79 90 e4 bd	...W..R..3.iy
0050	bb 23 9b 57 c4 2f 03 98 f9 64 34 45 59 ea f1 1e	..#W../..d4EY
0060	35 20 48 43 9a b9 7b cd d5 a7 da c2 2b a4 34 18	5 HC..{.+
0070	d5 90 b3 4e a5 69 98 9b 33 2c 90 c4 09 62 4c 3d	...N.i.. 3,...
0080	5b 90 00 22 13 01 13 03 13 02 c0 2b c0 2f cc a9	[..".+..

Record Layer (tls.record), 517 Bytes

```
enum {  
    server_name(0), /* RFC 6066 */  
    max_fragment_length(1), /* RFC 6066 */  
    status_request(5), /* RFC 6066 */  
    supported_groups(10), /* RFC 8422, 7919 */  
    signature_algorithms(13), /* RFC 8446 */  
    use_srtp(14), /* RFC 5764 */  
    heartbeat(15), /* RFC 6520 */  
    application_layer_protocol_negotiation(16), /* RFC 7301 */  
    signed_certificate_timestamp(18), /* RFC 6962 */  
    client_certificate_type(19), /* RFC 7250 */  
    server_certificate_type(20), /* RFC 7250 */  
    padding(21), /* RFC 7685 */  
    pre_shared_key(41), /* RFC 8446 */  
    early_data(42), /* RFC 8446 */  
    supported_versions(43), /* RFC 8446 */  
    cookie(44), /* RFC 8446 */  
    psk_key_exchange_modes(45), /* RFC 8446 */  
    certificate_authorities(47), /* RFC 8446 */  
    oid_filters(48), /* RFC 8446 */  
    post_handshake_auth(49), /* RFC 8446 */  
    signature_algorithms_cert(50), /* RFC 8446 */  
    key_share(51), /* RFC 8446 */  
    (65535)  
} ExtensionType;
```

```
enum {  
    server_name(0), /* RFC 6066 */  
    max_fragment_length(1), /* RFC 6066 */  
    status_request(5), /* RFC 6066 */  
    supported_groups(10), /* RFC 8422, 7919 */  
    signature_algorithms(13), /* RFC 8446 */  
    use_srtp(14), /* RFC 5764 */  
    heartbeat(15), /* RFC 6520 */  
    application_layer_protocol_negotiation(16), /* RFC 7301 */  
    signed_certificate_timestamp(18), /* RFC 6962 */  
    client_certificate_type(19), /* RFC 7250 */  
    server_certificate_type(20), /* RFC 7250 */  
    padding(21), /* RFC 7685 */  
    pre_shared_key(41), /* RFC 8446 */  
    early_data(42), /* RFC 8446 */  
    supported_versions(43), /* RFC 8446 */  
    cookie(44), /* RFC 8446 */  
    psk_key_exchange_modes(45), /* RFC 8446 */  
    certificate_authorities(47), /* RFC 8446 */  
    oid_filters(48), /* RFC 8446 */  
    post_handshake_auth(49), /* RFC 8446 */  
    signature_algorithms_cert(50), /* RFC 8446 */  
    key_share(51), /* RFC 8446 */  
    (65535)  
} ExtensionType;
```



0030	01 f5 9c bf 00 00	16 03 03 00 7a 02 00 00 76 03 z.z.v
0040	03 73 b3 ee ab 9d df c1	cb ab 9b 16 68 71 ed 9b	.s.....	...hq.
0050	f8 fa 82 35 d0 3e dc 40	94 e4 32 d4 67 85 a5 67	...5.>@	..2.g.g
0060	65 20 48 43 9a b9 7b cd	d5 a7 da c2 2b a4 34 18	e HC..{.+.4.

```

0
|
v
PSK -> HKDF-Extract = Early Secret
|
+-----> Derive-Secret(., "ext binder" | "res binder", "")
|               = binder_key
|
+-----> Derive-Secret(., "c e traffic", ClientHello)
|               = client_early_traffic_secret
|
+-----> Derive-Secret(., "e exp master", ClientHello)
|               = early_exporter_master_secret
|
v
Derive-Secret(., "derived", "")
|
v
(EC)DHE -> HKDF-Extract = Handshake Secret
|
+-----> Derive-Secret(., "c hs traffic",
|               ClientHello...ServerHello)
|               = client_handshake_traffic_secret
|
+-----> Derive-Secret(., "s hs traffic",
|               ClientHello...ServerHello)
|               = server_handshake_traffic_secret
|
v
Derive-Secret(., "derived", "")
|
v
0 -> HKDF-Extract = Master Secret
|
+-----> Derive-Secret(., "c ap traffic",
|               ClientHello...server Finished)
|               = client_application_traffic_secret_0
|
+-----> Derive-Secret(., "s ap traffic",
|               ClientHello...server Finished)
|               = server_application_traffic_secret_0
|
+-----> Derive-Secret(., "exp master",
|               ClientHello...server Finished)
|               = exporter_master_secret
|
+-----> Derive-Secret(., "res master",
|               ClientHello...client Finished)
|               = resumption_master_secret

```

The key derivation process makes use of the **HKDF-Extract** and **HKDF-Expand** functions as defined for **HKDF** [RFC5869], as well as the functions defined below:

HKDF-Expand-Label(Secret, Label, Context, Length) =
HKDF-Expand(Secret, HkdfLabel, Length)

```

struct {
    uint16 length = Length;
    opaque label<7..255> = "tls13 " + Label;
    opaque context<0..255> = Context;
} HkdfLabel;

```

Derive-Secret(Secret, Label, Messages) =
HKDF-Expand-Label(Secret, Label,
Transcript-Hash(Messages), Hash.length)

```

      0
      |
      v
PSK -> HKDF-Extract = Early Secret
      |
      +-----> Derive-Secret(., "ext binder" | "res binder", "")
                  = binder_key
      |
      +-----> Derive-Secret(., "c e traffic", ClientHello)
                  = client_early_traffic_secret
      |
      +-----> Derive-Secret(., "e exp master", ClientHello)
                  = early_exporter_master_secret
      |
      v
      Derive-Secret(., "derived", "")
      |
      v
(EC)DHE -> HKDF-Extract = Handshake Secret
      |
      +-----> Derive-Secret(., "c hs traffic",
                  ClientHello...ServerHello)
                  = client_handshake_traffic_secret
      |
      +-----> Derive-Secret(., "s hs traffic",
                  ClientHello...ServerHello)
                  = server_handshake_traffic_secret
      |
      v

```

[sender]_write_key = HKDF-Expand-Label(Secret, "key", "", key_length)

[sender]_write_iv = HKDF-Expand-Label(Secret, "iv", "", iv_length)

TLS 1.3 – Encrypted Extensions ([RFC 8446, 4.3.1](#))



```
struct {  
    Extension extensions<0..2^16-1>;  
} EncryptedExtensions;
```

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212....	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Cipher Spec
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK]
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate Verify
5 0.027442	212....	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Finished
6 0.027697	212....	23.57.29.178	HTTP	568	GET /en/java/javase/13/

- > TLSv1.3 Record Layer: Handshake Protocol: Server Hello
- > TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
- > TLSv1.3 Record Layer: Handshake Protocol: Encrypted Extensions
 - Opaque Type: Application Data (23)
 - Version: TLS 1.2 (0x0303)
 - Length: 52
 - [Content Type: Handshake (22)]
 - > Handshake Protocol: Encrypted Extensions
 - Handshake Type: Encrypted Extensions (8)
 - Length: 31
 - Extensions Length: 29
 - > Extension: server_name (len=0)
 - > Extension: supported_groups (len=6)
 - > Extension: application_layer_protocol_negotiation (len=11)

Offset	Hex	ASCII
00b0	d2 34 fc 25 5f 14 03 03 00 01 01 17 03 03 00 34	.4.%... ..
00c0	3f 8d 96 68 2c c8 8a 54 ae a7 b0 7d dd fd d9 d3	?..h,..T ...}
00d0	6b 7f 27 66 12 87 0a 30 8f a4 f1 26 c6 88 a7 ca	k.'f...0 ...&...
00e0	30 86 37 df 5d 95 38 e2 6d 47 85 15 fe 5a bd 55	0.7.]·8· mG...Z·
00f0	bb 24 ce 07 17 03 03 0c 8f 98 06 df 8b ef dd d7	·\$.~... ..
0100	62 4a 97 7d 30 b5 c8 a3 15 03 4a 07 7e 6e a9 a6	bJ·}0... ..J~n..
0110	bc 1f 2f a7 fa 90 45 a4 5b 29 2f e4 5e 85 62 9b	../.E· [)/·^·b·

Frame (1304 bytes) Decrypted TLS (35 bytes)

Record Layer (tls.record), 57 Bytes



Time	Source	Destination	Protocol	Length	Info
1 0.000000	212...	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Ciph
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK]
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate
5 0.027442	212...	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Fin
6 0.027607	212...	23.57.29.178	HTTP	568	GET /en/java/javaee/13/

▼ TLSv1.3 Record Layer: Handshake Protocol: Certificate

Opaque Type: Application Data (23)

Version: TLS 1.2 (0x0303)

Length: 3215

[Content Type: Handshake (22)]

▼ Handshake Protocol: Certificate

Handshake Type: Certificate (11)

Length: 3194

Certificate Request Context Length: 0

Certificates Length: 3190

▼ Certificates (3190 bytes)

Certificate Length: 1525

> Certificate: 308205f1308204d9a003020102021001887bd706f662005:

Extensions Length: 479

> Extension: status_request (len=475)

Certificate Length: 1176

> Certificate: 308204943082037ca003020102021001fda3eb6eca75c88:

Extensions Length: 0

> Transport Layer Security

<

0000	0b 00 0c 7a 00 00 0c 76 00 05 f5 30 82 05 f1 30	...z...v ...0...
0010	82 04 d9 a0 03 02 01 02 02 10 01 88 7b d7 06 f6{...
0020	62 00 57 34 fb fd 6c 30 21 35 30 0d 06 09 2a 86	b.W4..l0 !50...*

Frame (1139 bytes)	Reassembled TCP (3220 bytes)	Decrypted TLS (3198 bytes)
--------------------	------------------------------	----------------------------

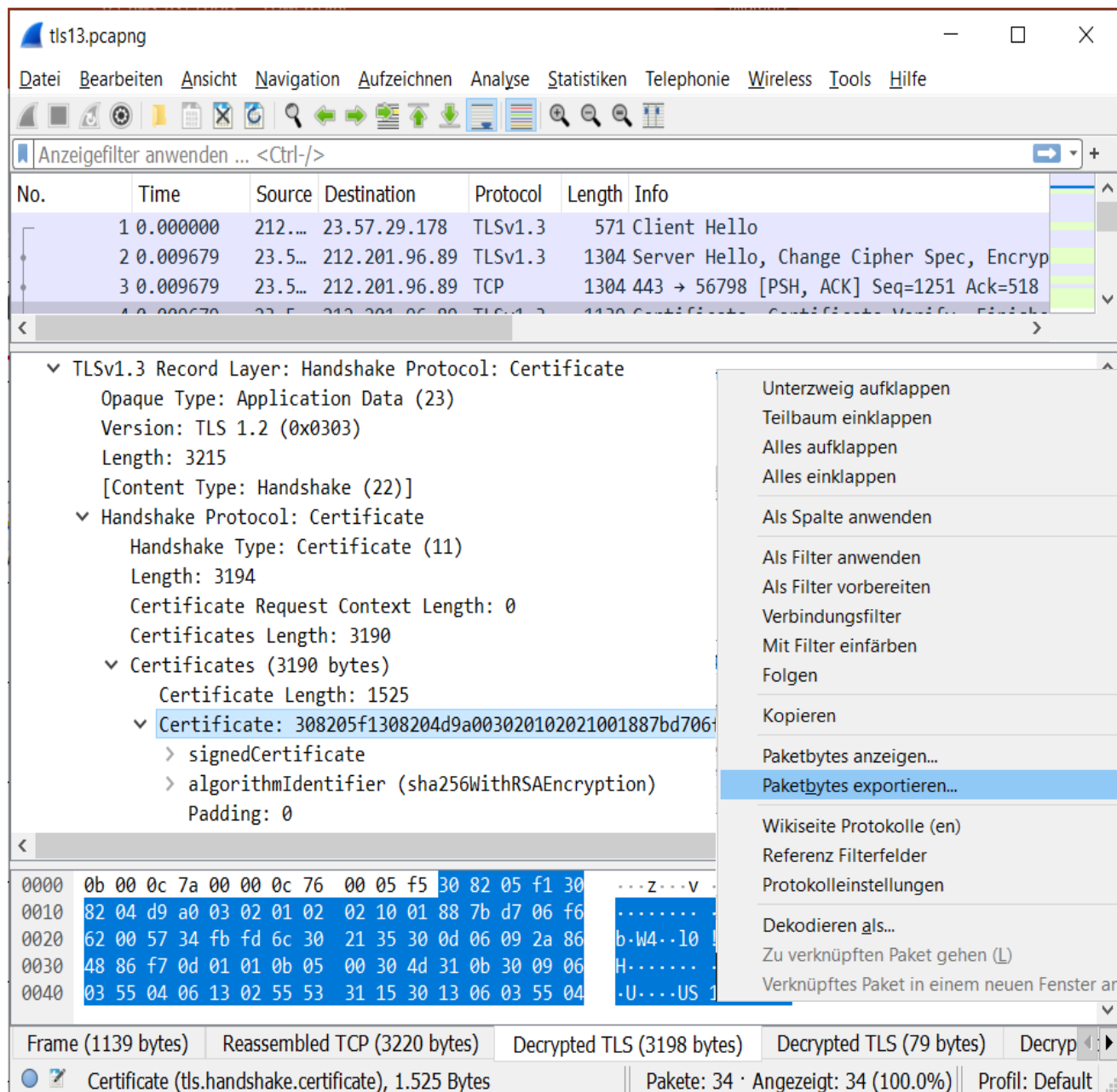
Handshake protocol message (tls.handshake), 3.198 Bytes

```
enum {
    X509(0),
    RawPublicKey(2),
    (255)
} CertificateType;

struct {
    select (certificate_type) {
        case RawPublicKey:
            /* From RFC 7250 ASN.1_subjectPublicKeyInfo */
            opaque ASN1_subjectPublicKeyInfo<1..2^24-1>;
        case X509:
            opaque cert_data<1..2^24-1>;
    };
    Extension extensions<0..2^16-1>;
} CertificateEntry;

struct {
    opaque certificate_request_context<0..2^8-1>;
    CertificateEntry certificate_list<0..2^24-1>;
} Certificate;
```

X.509 Certificates – Export from a Capture



The screenshot shows the Wireshark interface with a capture file named 'tls13.pcapng'. The packet list shows a TLSv1.3 handshake. The packet details pane is expanded to show the 'Handshake Protocol: Certificate' section. A context menu is open over the 'Certificate' field, with the 'Paketbytes exportieren...' option selected. The packet bytes pane shows the raw data of the certificate.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	212...	23.57.29.178	TLSv1.3	571	Client Hello
2	0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Cipher Spec, Encryp
3	0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] Seq=1251 Ack=518

▼ TLSv1.3 Record Layer: Handshake Protocol: Certificate
Opaque Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 3215
[Content Type: Handshake (22)]

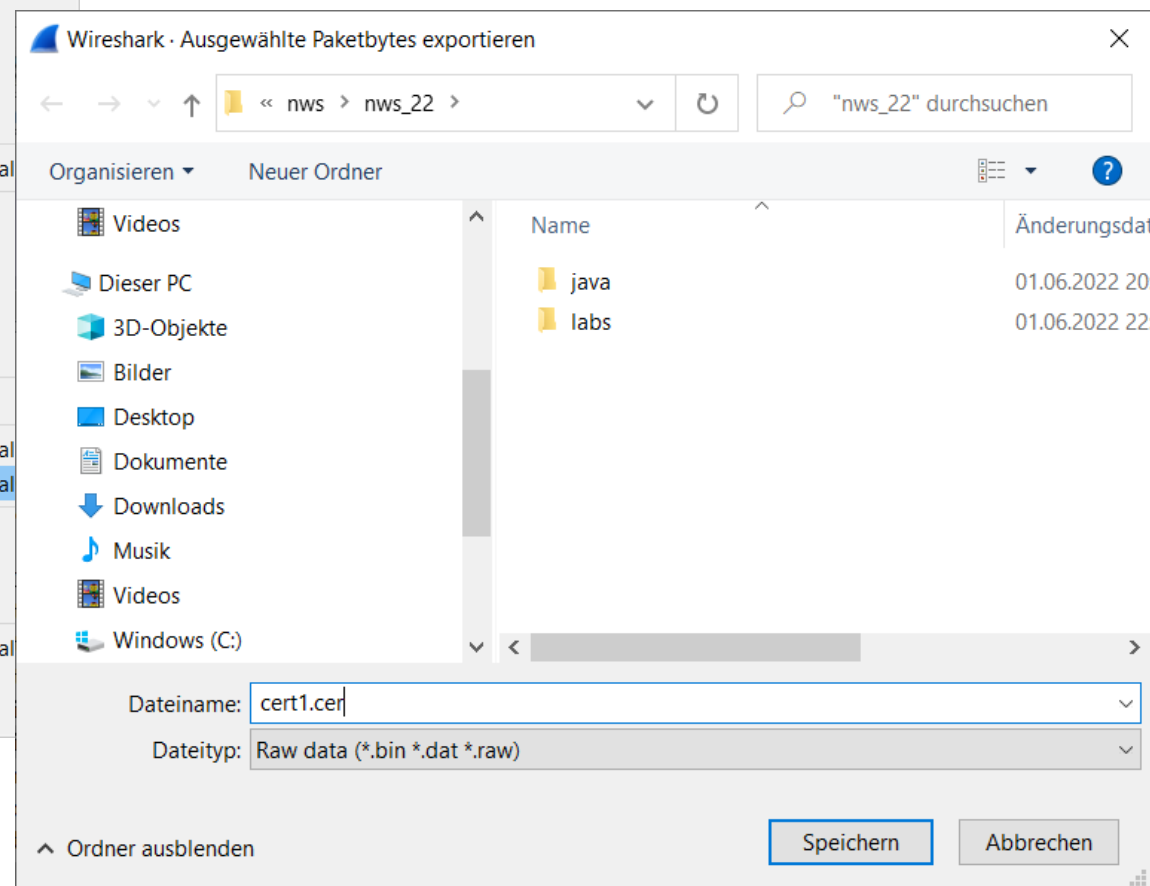
▼ Handshake Protocol: Certificate
Handshake Type: Certificate (11)
Length: 3194
Certificate Request Context Length: 0
Certificates Length: 3190

▼ Certificates (3190 bytes)
Certificate Length: 1525

▼ Certificate: 308205f1308204d9a003020102021001887bd706f
 > signedCertificate
 > algorithmIdentifier (sha256WithRSAEncryption)
 Padding: 0

0000 0b 00 0c 7a 00 00 0c 76 00 05 f5 30 82 05 f1 30 ...Z...V
0010 82 04 d9 a0 03 02 01 02 02 10 01 88 7b d7 06 f6
0020 62 00 57 34 fb fd 6c 30 21 35 30 0d 06 09 2a 86 b·W4·10
0030 48 86 f7 0d 01 01 0b 05 00 30 4d 31 0b 30 09 06 H.....
0040 03 55 04 06 13 02 55 53 31 15 30 13 06 03 55 04 .U....US

Frame (1139 bytes) Reassembled TCP (3220 bytes) Decrypted TLS (3198 bytes) Decrypted TLS (79 bytes) Decryp
Certificate (tls.handshake.certificate), 1.525 Bytes Pakete: 34 · Angezeigt: 34 (100.0%) Profil: Default



The screenshot shows the 'Wireshark · Ausgewählte Paketbytes exportieren' dialog box. The 'Organisieren' tab is selected, showing a tree view of the file system. The 'Dateiname' field is set to 'cert1.cer' and the 'Dateityp' is set to 'Raw data (*.bin *.dat *.raw)'. The 'Speichern' button is highlighted.

Wireshark · Ausgewählte Paketbytes exportieren

« nws » nws_22 » "nws_22" durchsuchen

Organisieren Neuer Ordner

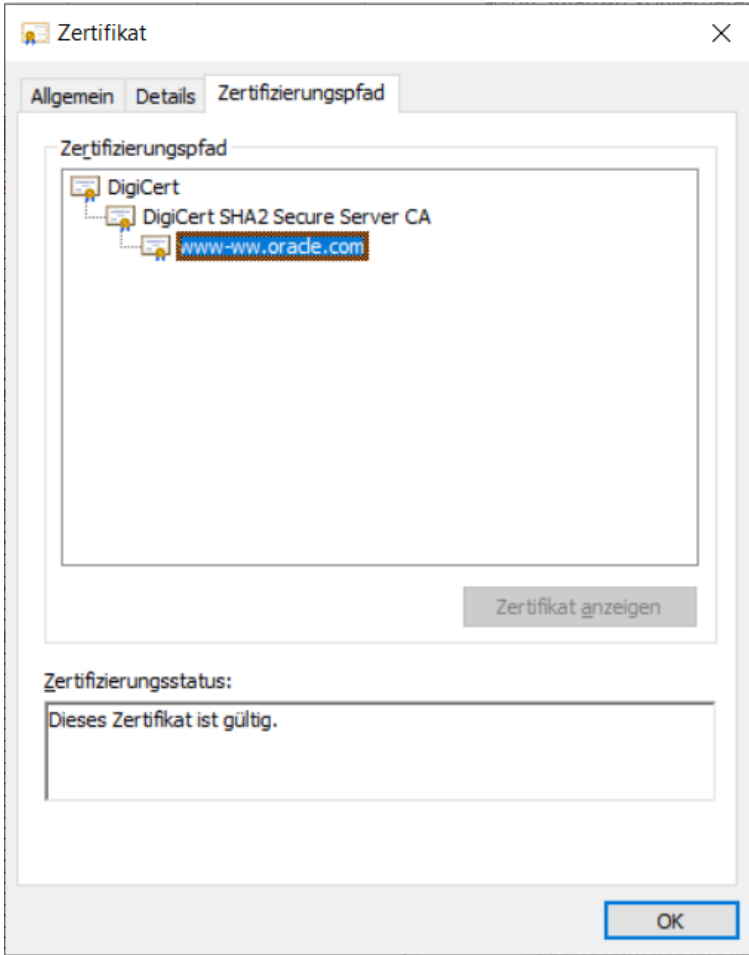
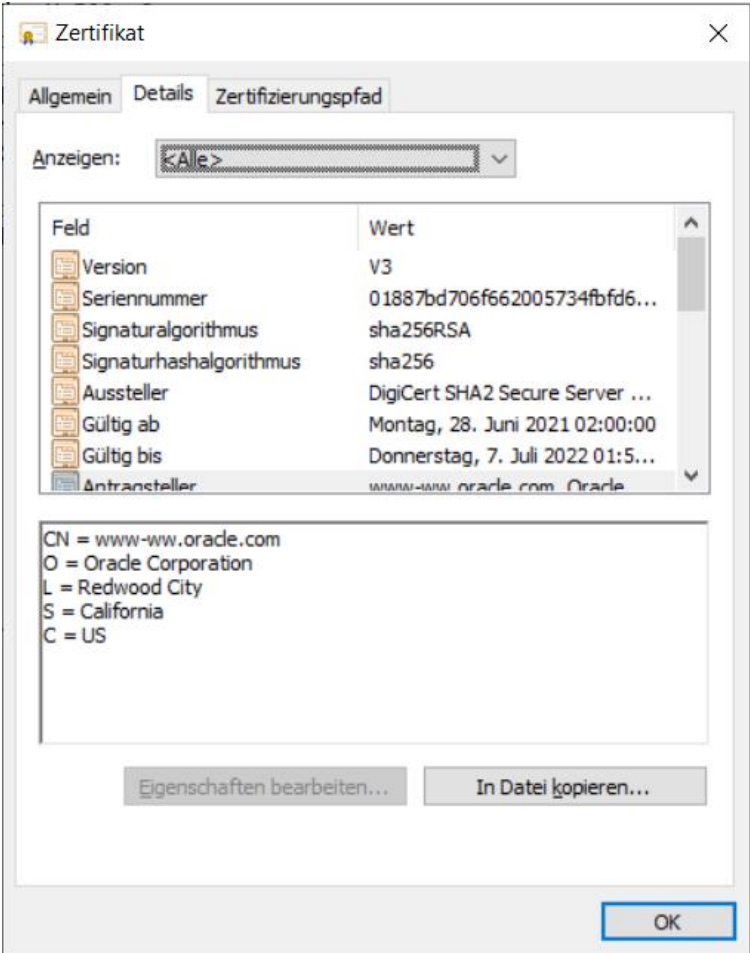
Videos
Dieser PC
3D-Objekte
Bilder
Desktop
Dokumente
Downloads
Musik
Videos
Windows (C:)

Name Änderungsdat
java 01.06.2022 20
labs 01.06.2022 22

Dateiname: cert1.cer
Dateityp: Raw data (*.bin *.dat *.raw)

Ordner ausblenden Speichern Abbrechen

X.509 Certificates – Displayed in MS Windows Certificate Tool

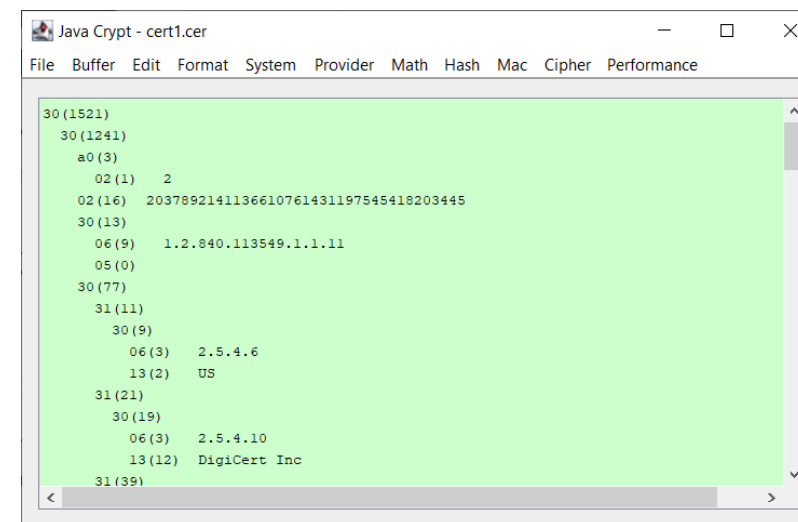


- **RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile**

- **RFC 5280, 4.1: Basic Certificate Fields**

The X.509 v3 certificate basic syntax is as follows. For signature calculation, the data that is to be signed is encoded using the ASN.1 distinguished encoding rules (DER) [X.690]. ASN.1 DER encoding is a tag, length, value encoding system for each element.

- Good Intro by Burton S. Kaliski Jr.: [A Layman's Guide to a Subset of ASN.1, BER, and DER](#)
- JavaCrypt – Tool for structured display:



■ RFC 5280, 4.1: Basic Certificate Fields

```
Certificate ::= SEQUENCE {  
    tbsCertificate      TBSCertificate,  
    signatureAlgorithm  AlgorithmIdentifier,  
    signatureValue      BIT STRING }
```

```
TBSCertificate ::= SEQUENCE {  
    version          [0] EXPLICIT Version DEFAULT v1,  
    serialNumber      CertificateSerialNumber,  
    signature         AlgorithmIdentifier,  
    issuer            Name,  
    validity          Validity,  
    subject           Name,  
    subjectPublicKeyInfo SubjectPublicKeyInfo,  
    issuerUniqueID    [1] IMPLICIT UniqueIdentifier OPTIONAL,  
                    -- If present, version MUST be v2 or v3  
    subjectUniqueID   [2] IMPLICIT UniqueIdentifier OPTIONAL,  
                    -- If present, version MUST be v2 or v3  
    extensions        [3] EXPLICIT Extensions OPTIONAL  
                    -- If present, version MUST be v3  
}
```




```
struct {
    SignatureScheme algorithm;
    opaque signature<0..2^16-1>;
} CertificateVerify;
```

The digital signature is then computed over the concatenation of:

- A string that consists of octet 32 (0x20) repeated 64 times
- The context string
- A single 0 byte which serves as the separator
- The content to be signed

Time	Source	Destination	Protocol	Length	Info
1 0.000000	212...	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Ci
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK]
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate
5 0.027442	212...	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Fin
6 0.027607	212...	23.57.29.178	HTTP	568	GET /en/java/javaee/13/

>	Transmission Control Protocol, Src Port: 443, Dst Port: 56798, Seq: 250
>	[3 Reassembled TCP Segments (3220 bytes): #2(1060), #3(1250), #4(910)]
>	Transport Layer Security
▼	Transport Layer Security
▼	TLSv1.3 Record Layer: Handshake Protocol: Certificate Verify
	Opaque Type: Application Data (23)
	Version: TLS 1.2 (0x0303)
	Length: 96
	[Content Type: Handshake (22)]
▼	Handshake Protocol: Certificate Verify
	Handshake Type: Certificate Verify (15)
	Length: 75
▼	Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
	Signature Hash Algorithm Hash: SHA256 (4)
	Signature Hash Algorithm Signature: ECDSA (3)
	Signature length: 71
	Signature: 304502210090eaa694f72bbb012e4219acba3703d98d6da0eb1b
>	TLSv1.3 Record Layer: Handshake Protocol: Finished

0000	0f 00 00 4b 04 03 00 47 30 45 02 21 00 90 ea a6	...K...G 0E!...
0010	94 f7 2b bb 01 2e 42 19 ac ba 37 03 d9 8d 6d a0	...+...B...7...m
0020	eb 1b 89 e2 6b 0c e6 03 2e 1b d3 c0 f9 02 20 72	...k...

Frame (1139 bytes)	Reassembled TCP (3220 bytes)	Decrypted TLS (3198 bytes)	D
--------------------	------------------------------	----------------------------	---

Handshake protocol message (tls.handshake), 79 Bytes



Time	Source	Destination	Protocol	Length	Info
1 0.000000	212....	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Ci
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] :
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificat
5 0.027442	212....	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Fin
6 0.027607	212....	23.57.29.178	HTTP	568	GET /en/java/javase/13/

<

- > Transmission Control Protocol, Src Port: 443, Dst Port: 56798, Seq: 250
- > [3 Reassembled TCP Segments (3220 bytes): #2(1060), #3(1250), #4(910)]
- > Transport Layer Security
- ▼ Transport Layer Security
 - ▼ TLSv1.3 Record Layer: Handshake Protocol: Certificate Verify
 - Opaque Type: Application Data (23)
 - Version: TLS 1.2 (0x0303)
 - Length: 96
 - [Content Type: Handshake (22)]
 - ▼ Handshake Protocol: Certificate Verify
 - Handshake Type: Certificate Verify (15)
 - Length: 75
 - ▼ Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
 - Signature Hash Algorithm Hash: SHA256 (4)
 - Signature Hash Algorithm Signature: ECDSA (3)
 - Signature length: 71
 - Signature: 304502210090eaa694f72bbb012e4219acba3703d98d6da0eb1b
 - > TLSv1.3 Record Layer: Handshake Protocol: Finished

<

0000	0f 00 00 4b 04 03 00 47 30 45 02 21 00 90 ea a6	...K...G 0E!...
0010	94 f7 2b bb 01 2e 42 19 ac ba 37 03 d9 8d 6d a0	...+...B...7...m
0020	eb 1b 89 e2 6b 0c e6 03 2e 1b d3 c0 f9 02 20 72	...k...

Frame (1139 bytes) Reassembled TCP (3220 bytes) Decrypted TLS (3198 bytes) D

Handshake protocol message (tls.handshake), 79 Bytes

```
struct {
    SignatureScheme algorithm;
    opaque signature<0..2^16-1>;
} CertificateVerify;
```

The digital signature is then computed over the concatenation of:

- A string that consists of octet 32 (0x20) repeated 64 times
20
- The context string
544c5320312e332c20736572766572204365727469666963617465566572696679 ("TLS 1.3, server CertificateVerify".getBytes())
- A single 0 byte which serves as the separator
00
- The content to be signed
Transcript-Hash(Handshake Context, Certificate)



Time	Source	Destination	Protocol	Length	Info
1 0.000000	212...	23.57.29.178	TLSv1.3	571	Client Hello
2 0.009679	23.5...	212.201.96.89	TLSv1.3	1304	Server Hello, Change Ciph
3 0.009679	23.5...	212.201.96.89	TCP	1304	443 → 56798 [PSH, ACK] S
4 0.009679	23.5...	212.201.96.89	TLSv1.3	1139	Certificate, Certificate
5 0.027442	212...	23.57.29.178	TLSv1.3	134	Change Cipher Spec, Fin
6 0.027607	212...	23.57.29.178	HTTP	568	GET /en/java/javase/13/

[Content Type: Handshake (22)]
Handshake Protocol: Certificate Verify
Handshake Type: Certificate Verify (15)
Length: 75
Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
Signature Hash Algorithm Hash: SHA256 (4)
Signature Hash Algorithm Signature: ECDSA (3)
Signature length: 71
Signature: 304502210090eaa694f72bbb012e4219acba3703d98d6da0eb1b
TLSv1.3 Record Layer: Handshake Protocol: Finished
Opaque Type: Application Data (23)
Version: TLS 1.2 (0x0303)
Length: 69
[Content Type: Handshake (22)]
Handshake Protocol: Finished
Handshake Type: Finished (20)
Length: 48
Verify Data

0000	14 00 00 30 98 48 be da	c4 6f a6 dc f5 04 85 5d	...0.H...o....
0010	04 77 ce 24 d8 6c 88 8a	fe 27 96 4b 2e c1 df c0	..w\$.l..'.K...
0020	b7 03 ad 08 af 13 ed fc	0e 3a 2e 23 45 c5 47 2f:..#E.G,

Frame (1139 bytes)	Reassembled TCP (3220 bytes)	Decrypted TLS (3198 bytes)	D
--------------------	------------------------------	----------------------------	---

Handshake protocol message (tls.handshake), 52 Bytes

```
struct {
    opaque verify_data[Hash.length];
} Finished;
```

```
verify_data =
    HMAC(finished_key,
         Transcript-Hash(Handshake Context,
                         Certificate*, CertificateVerify*))
```

* Only included if present.

```
finished_key =
    HKDF-Expand-Label(BaseKey, "finished", "", Hash.length)
```

BaseKey:

- server_handshake_traffic_secret (Server Finished)
- client_handshake_traffic_secret (Client Finished)