Safe and fast SSL/TLS-handshake

GS, FO

2024-10-11

Contents

Motivation

- unsecured channels subject to data exposure to third parties
- securing of data channel needed $\rightarrow$ SSL/TLS
- method of attributing identities to identity holders needed
- method of signaling cessation of identity usage needed
- x509/CRK (RFC 5280) and OCSP (RFC 6960) for identity handling/verification
- different methods of establishing secure channel

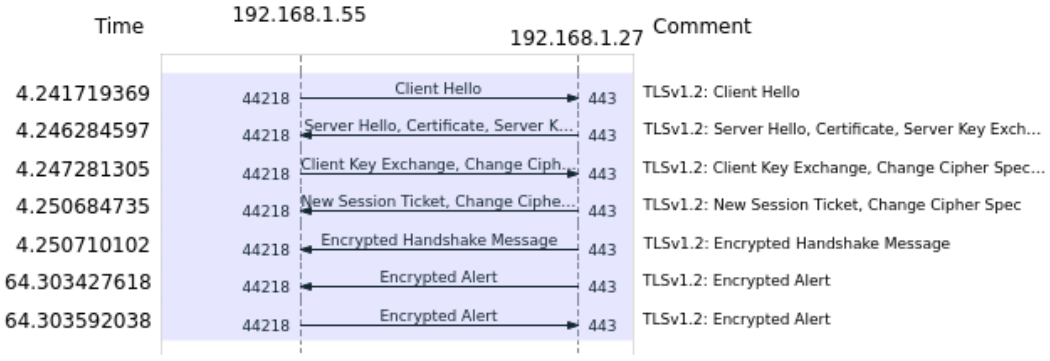
History

- 1994 SSL 1.0 concept
- 1995 SSL 2.0 first release (RFC 6101), MD5-hashing, one key for auth/enc, depr. 2011 (RFC 6176)
- 1996 SSL 3.0 (RFC 6101) subject to POODLE-attack (CVE-2014-3566[1]), depr. 2014 (RFC 7568)
- 1999 TLS 1.0 (RFC 2246), depr. 2021 (RFC 8996)
- 2006 TLS 1.1 (RFC 4346), protection against CBC-atacks, depr. 2022
- 2008 TLS 1.2 (RFC 5246), replaced MD5/SHA-1, algo selection mechanism
- 2018 TLS 1.3 (RFC 8446), defaults to AES256_GCM_SHA384, insecure algos removed, changed handshake/connection init

$\Rightarrow$ currently two productive versions

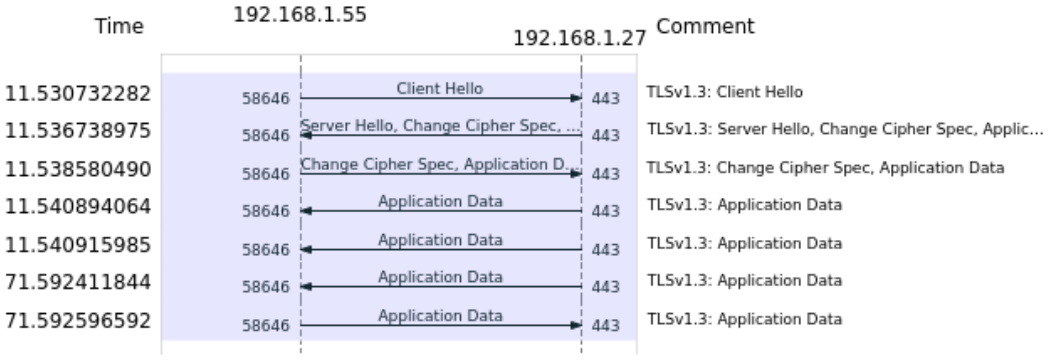
[1] https://security.googleblog.com/2014/10/this-poodle-bites-exploiting-ssl-30.html

Structure of TLS



TLSv1.2 handshake

Structure of TLS



TLSv1.3 handshake

TLSv1.3

- only AES and ChaCha20 as cipher
  $\longrightarrow$ in total 5 options
- only Diffie-Hellman (incl./excl. elliptic curves)
- no (downgrade)renegotiation anymore

Performance considerations

different ciphers

- software based, ChaCha20 is up to 9x faster

type	2 bytes	31 bytes	136 bytes	1024 bytes	8192 bytes	16384 bytes
AES-256-GCM (Software)	3912.58k	43681.83k	119433.57k	220805.46k	240091.14k	241401.86k
ChaCha20-Poly1305 (Software)	5406.06k	79034.59k	256344.41k	1439373.99k	2491817.98k	2634612.74k

Performance considerations

hardware acceleration

- AES with CPU-support performs best
- also faster than ChaCha20
- not available on all CPU/with all compilers/with all software

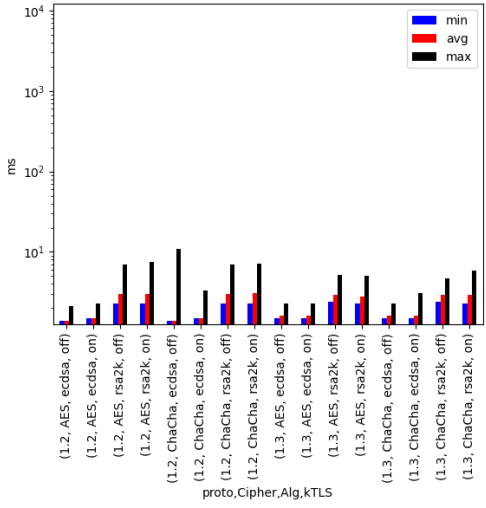
type	2 bytes	31 bytes	136 bytes	1024 bytes	8192 bytes	16384 bytes
AES-256-GCM (Software)	3912.58k	43681.83k	119433.57k	220805.46k	240091.14k	241401.86k
AES-256-GCM (CPU-instructions)	12352.14k	156771.30k	600536.56k	2364060.33k	3829205.67k	4008596.82k
ChaCha20-Poly1305 (Software)	5406.06k	79034.59k	256344.41k	1439373.99k	2491817.98k	2634612.74k

Performance considerations

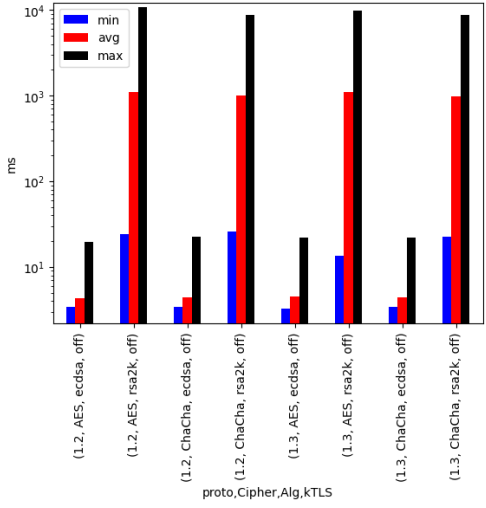
SSL offloading

- three options
  1. software based (user level)
  2. kTLS (kernel level)
  3. NIC-offload (fully managed by hardware)
- should improve throughput due to less context switches
- ! may introduce operational problems
  - ? possible issues with TCP-checksum on IPv6
  - \- size limits
  - \- issues with network-mounted resources
- impact questionable

Performance



Intel Xeon Gold 6150, 500 parallel connections

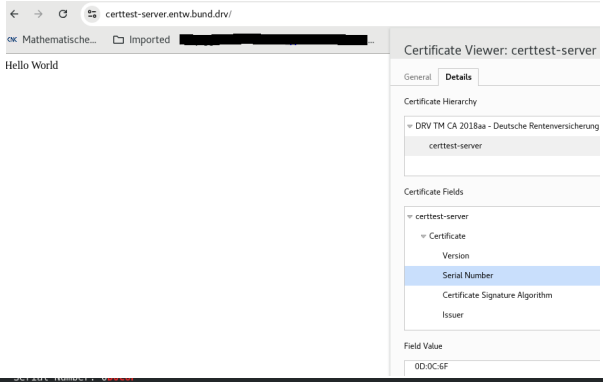


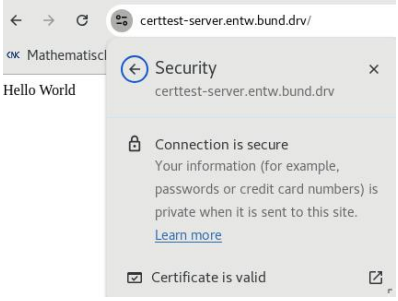
Raspberry Pi 4, 500 parallel connections

how to: fast and safe TLS handshake

- disable old ciphers
- ∼ OCSP
  - ! Let's Encrypt removes OCSP soon
- ∼ OCSP-Stapling
  - ! Chrome ignores CRL or OCSP, only knows of revoked certificate if stapled result
- use elliptic curve for keys
- use LARGE packages
- SSL offloading (e.g. Nvidia ConnectX-7 for 400G connections)

Caveats






```
(base) [roadrunner@rr027 Downloads]$ openssl ocsp -url http://ocsp-drv.tc.deutsche-rentenversicherung.de -CAfile drv_tm_ca_2018aa.cer -issuer drv_tm_ca_2018aa.cer -serial 855151
Response Verify Failure
140026174059712:error:27069065:OCSP routines:OCSP_basic_verify:certificate verify error:crypto/ocsp/ocsp_vfy.c:93:Verify error:unable to get issuer certificate
140026174059712:error:27069065:OCSP routines:OCSP_basic_verify:certificate verify error:crypto/ocsp/ocsp_vfy.c:93:Verify error:unable to get issuer certificate
855151: revoked
        This Update: Oct 10 17:21:18 2024 GMT
        Next Update: Oct 11 03:21:18 2024 GMT
        Reason: (UNKNOWN)
        Revocation Time: Nov 27 12:20:30 2021 GMT
(base) [roadrunner@rr027 Downloads]$ openssl crl -in "x.crl?dn=cn=706810,ou=NQ DRV TM CA,cn=Public,o=DRV,c=DE&attrname=certificateRevocationList" -inform der -noout -text|grep D0C6
F
    Serial Number: 0D0C6F
```