

Laboratoire 4 : PKI et TLS

Nathan Füllemann

15 juin 2025

Signatures des emails

Génère une CSR, clé privée pour la CA TLS intermédiaire

```
openssl req -new -config config.conf -out Fullemann-Email.csr -keyout Fullemann-Email
```

- new : création d'une nouvelle demande de certificat
- config config.conf : utilise le fichier de configuration OpenSSL fourni
- out Fullemann-TLS.csr : sauvegarde la CSR dans ce fichier
- keyout Fullemann-TLS.key : sauvegarde la clé privée générée dans ce fichier

Puis faire signer le certificat par la CA racine ### Génère une CSR et une clé privée pour le certificat email utilisateur

```
openssl req -new -config config-email.conf -out nathan.fulleman@hes-so.ch.csr -keyout
```

Signe la CSR utilisateur avec la CA intermédiaire email

```
openssl ca -config config-email.conf -in ../mail_client/nathan.fulleman@hes-so.ch.csr
```

Concatène le certificat de la CA intermédiaire et celui de la racine

```
cat ../mail_CA/Fullemann-Email.crt ../HEIG-VDRoot.crt > CERTIF_CHAIN.crt
```

Exporte la clé privée, le certificat utilisateur et la chaîne dans un fichier PKCS#12

```
openssl pkcs12 -export -name fullemann_p12 \  
-inkey nathan.fulleman@hes-so.ch.key \  
-in nathan.fulleman@hes-so.ch.crt \  
-certfile CERTIF_CHAIN.crt \  
-out nathan.fulleman@hes-so.ch.p12
```

Configuration d'un serveur TLS

Génère une CSR et une clé privée pour la CA TLS intermédiaire

```
openssl req -new -config config.conf -out Fullemann-TLS.csr -keyout Fullemann-TLS.key
```

Génère une CSR et une clé privée pour le certificat du serveur TLS

```
openssl req -new -config config-client.conf -out ../TLS_client/IP.csr -keyout ../TLS_client/IP.key
```

Signe la CSR du serveur TLS avec la CA intermédiaire TLS

```
openssl ca -config config.conf -in ../TLS_client/IP.csr -out ../TLS_client/IP.crt -ext
```

Concatène le certificat serveur et celui de la CA intermédiaire

```
cat ../TLS_client/IP.crt ../TLS_CA/Fullemann-TLS.crt ../HEIG-VDRoot.crt > ../TLS_client/chain.crt
```

Configuration nginx

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;

    # Chemins absolus vers les certificats
    ssl_certificate /etc/nginx/CERTIF_CHAIN.pem;
    ssl_certificate_key /etc/nginx/IP.key;
    ssl_trusted_certificate /etc/nginx/Fullemann-TLS.crt;
    # Protocoles supportés
    ssl_protocols TLSv1.2 TLSv1.3;

    # Ciphersuites
    ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-
    ssl_prefer_server_ciphers on;

    ssl_session_timeout 1d;
    ssl_session_cache shared:MozSSL:10m; # about 40000 sessions
    ssl_session_tickets off;

    # HSTS (ngx_http_headers_module is required) (63072000 seconds)
    add_header Strict-Transport-Security "max-age=63072000" always;

    access_log /var/www/labo-crypto.com/logs/access.log;
    error_log /var/www/labo-crypto.com/logs/error.log;
    root /var/www/labo-crypto.com/public/;

    location / {
        index index.html;
    }

    error_page 405 =200 $uri;
}

server {
    listen 80 default_server;
    listen [::]:80 default_server;
```

```
    return 301 https://$host$request_uri;
}
```

Passage de fichier au serveur distant

```
scp -P 2212 ../TLS_client/CERTIF_CHAIN.pem labo@secuctfd.iict-heig-vd.in:~
```

```
ssh -p 2212 labo@secuctfd.iict-heig-vd.in
```

```
sudo mv ~/CERTIF_CHAIN.pem /etc/nginx/
```

Acces au serveur web

HTTP : <http://secuctfd.iict-heig-vd.in:8012> HTTPS : <https://secuctfd.iict-heig-vd.in:44312>

Réponses aux questions

1. Pourquoi devons-nous transmettre une chaîne de certificats dans les deux applications (email et TLS) ? > La chaîne de certificats permet au client de checker si le certificat présenté a bien été émis par une autorité de confiance. Comme le certificat utilisateur ou serveur est signé par une CA intermédiaire, il faut ducoup aussi que dans la chaine la CA intermédiaire soit incluse pour que le client puisse remonter jusqu'à la racine de confiance (HEIG-VDRoot).
2. Comment avez-vous configuré nginx ? Donnez votre fichier de configuration. > La configuration est metnionner si dessus. Par rapport à la configuration de <https://ssl-config.mozilla.org/> pour nginx 1.14 j'ai désactivé l'OCSP car les consignes nous le dise. TLS 1.2 est activé aussi. A part ça j'ai modifié les chemin d'accès de ma chaine de certificat (CERTIF_CHAIN.crt) et et de la clé privée du certificat TLS client (IP.key)
3. Fournissez le résultat du scan de testssl sur votre serveur ainsi que des commentaires, si nécessaire.

```
#####
testssl.sh version 3.2.0 from https://testssl.sh/
```

This program is free software. Distribution and modification under
GPLv2 permitted. USAGE w/o ANY WARRANTY. USE IT AT YOUR OWN RISK!

Please file bugs @ <https://testssl.sh/bugs/>

```
#####
```

Using OpenSSL 3.5.0 (Apr 8 2025) [~96 ciphers]
on MacBook-Pro-de-Nathan:/opt/homebrew/opt/openssl@3/bin/openssl

Start 2025-06-15 17:13:41 -->> 10.190.133.59:44312 (secuctfd.iict-heig-vd.in)

```
rDNS (10.190.133.59):  --
Service detected:    HTTP
```

Testing protocols via sockets except NPN+ALPN

```

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      not offered
TLS 1.1    not offered
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)

```

Testing cipher categories

```

NULL ciphers (no encryption)                not offered (OK)
Anonymous NULL Ciphers (no authentication)   not offered (OK)
Export ciphers (w/o ADH+NULL)                not offered (OK)
LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA                    not offered
Obsoleted CBC ciphers (AES, ARIA etc.)        not offered
Strong encryption (AEAD ciphers) with no FS  not offered
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)

```

Testing server's cipher preferences

Hexcode	Cipher Suite Name (OpenSSL)	KeyExch.	Encryption	Bits	Cipher Suite

SSLv2					
-					
SSLv3					
-					
TLSv1					
-					
TLSv1.1					
-					
TLSv1.2 (no server order, thus listed by strength)					
xc030	ECDHE-RSA-AES256-GCM-SHA384	ECDH 253	AESGCM	256	TLS_ECDHE_RSA_AES256_GCM_SHA384
xcca8	ECDHE-RSA-CHACHA20-POLY1305	ECDH 253	ChaCha20	256	TLS_ECDHE_RSA_CHACHA20_POLY1305_SHA256
xc02f	ECDHE-RSA-AES128-GCM-SHA256	ECDH 253	AESGCM	128	TLS_ECDHE_RSA_AES128_GCM_SHA256
TLSv1.3 (no server order, thus listed by strength)					
x1302	TLS_AES_256_GCM_SHA384	ECDH 253	AESGCM	256	TLS_AES_256_GCM_SHA384
x1303	TLS_CHACHA20_POLY1305_SHA256	ECDH 253	ChaCha20	256	TLS_CHACHA20_POLY1305_SHA256
x1301	TLS_AES_128_GCM_SHA256	ECDH 253	AESGCM	128	TLS_AES_128_GCM_SHA256

```

Has server cipher order?    no
(limited sense as client will pick)

```

Testing robust forward secrecy (FS) -- omitting Null Authentication/Encryption, 3DES

```
FS is offered (OK)          TLS_AES_256_GCM_SHA384
                             TLS_CHACHA20_POLY1305_SHA256
                             ECDHE-RSA-AES256-GCM-SHA384
                             ECDHE-RSA-CHACHA20-POLY1305
                             TLS_AES_128_GCM_SHA256
                             ECDHE-RSA-AES128-GCM-SHA256
Elliptic curves offered:    prime256v1 secp384r1 secp521r1 X25519 X448
TLS 1.2 sig_algs offered:   RSA-PSS-RSAE+SHA512 RSA-PSS-RSAE+SHA384
                             RSA-PSS-RSAE+SHA256 RSA+SHA512 RSA+SHA384
                             RSA+SHA256 RSA+SHA224 RSA+SHA1
TLS 1.3 sig_algs offered:   RSA-PSS-RSAE+SHA512 RSA-PSS-RSAE+SHA384
                             RSA-PSS-RSAE+SHA256
```

Testing server defaults (Server Hello)

```
TLS extensions (standard)  "max fragment length/#1" "EC point formats/#11"
                             "application layer protocol negotiation/#16"
                             "extended master secret/#23"
                             "supported versions/#43" "key share/#51"
                             "next protocol/#13172"
                             "renegotiation info/#65281"
Session Ticket RFC 5077 hint no -- no lifetime advertised
SSL Session ID support     yes
Session Resumption         Tickets no, ID: yes
TLS clock skew             Random values, no fingerprinting possible
Certificate Compression     none
Client Authentication       none
Signature Algorithm         SHA256 with RSA
Server key size             RSA 2048 bits (exponent is 65537)
Server key usage            Digital Signature, Key Encipherment
Server extended key usage   TLS Web Server Authentication, TLS Web Client Authentication
Serial                     OC NOT ok: length should be >= 64 bits entropy (is: 1
Fingerprints               SHA1 00FE6170DBEB889047260DC7A50BC874BFC13C0A
                             SHA256 A1793865726D45B5639ECCBEE502CABC795013244BEFD4D0
Common Name (CN)           secuctfd.iict-heig-vd.in
subjectAltName (SAN)       secuctfd.iict-heig-vd.in
Trust (hostname)           Ok via SAN and CN (same w/o SNI)
Chain of trust             NOT ok (self signed CA in chain)
EV cert (experimental)     no
Certificate Validity (UTC)  99 >= 60 days (2025-06-15 11:15 --> 2025-09-23 11:15)
ETS/"eTLS", visibility info not present
Certificate Revocation List --
OCSP URI                  --
                             NOT ok -- neither CRL nor OCSP URI provided
OCSP stapling              not offered
OCSP must staple extension --
```

```

DNS CAA RR (experimental)    not offered
Certificate Transparency      --
Certificates provided         3
Issuer                       Fullemann-TLS (HEIG-VD from CH)
Intermediate cert validity    #1: ok > 40 days (2035-06-15 11:15). Fullemann-TLS <-- HEIG-VD
                              #2: ok > 40 days (2030-12-31 23:59). HEIG-VDRoot <-- HEIG-VD
Intermediate Bad OCSP (exp.) Ok

```

Testing HTTP header response @ "/"

```

HTTP Status Code             200 OK
HTTP clock skew               0 sec from localtime
Strict Transport Security      730 days=63072000 s, just this domain
Public Key Pinning            --
Server banner                 nginx/1.14.0 (Ubuntu)
Application banner            --
Cookie(s)                     (none issued at "/")
Security headers              --
Reverse Proxy banner          --

```

Testing vulnerabilities

```

Heartbleed (CVE-2014-0160)    not vulnerable (OK), no heartbeat extension
CCS (CVE-2014-0224)          not vulnerable (OK)
Ticketbleed (CVE-2016-9244), experiment. not vulnerable (OK), no session ticket extension
ROBOT                         Server does not support any cipher suites that require TLS 1.2
Secure Renegotiation (RFC 5746) supported (OK)
Secure Client-Initiated Renegotiation not vulnerable (OK)
CRIME, TLS (CVE-2012-4929)    not vulnerable (OK)
BREACH (CVE-2013-3587)       potentially NOT ok, "gzip" HTTP compression
                              Can be ignored for static pages or if no compression
POODLE, SSL (CVE-2014-3566)   not vulnerable (OK), no SSLv3 support
TLS_FALLBACK_SCSV (RFC 7507) No fallback possible (OK), no protocol below TLS 1.2
SWEET32 (CVE-2016-2183, CVE-2016-6329) not vulnerable (OK)
FREAK (CVE-2015-0204)        not vulnerable (OK)
DROWN (CVE-2016-0800, CVE-2016-0703) not vulnerable on this host and port (OK)
                              make sure you don't use this certificate elsewhere
                              https://search.censys.io/search?resource=hosts
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK): no DH EXPORT ciphers,
BEAST (CVE-2011-3389)        not vulnerable (OK), no SSL3 or TLS1
LUCKY13 (CVE-2013-0169), experimental not vulnerable (OK)
Winshock (CVE-2014-6321), experimental not vulnerable (OK)
RC4 (CVE-2013-2566, CVE-2015-2808) no RC4 ciphers detected (OK)

```

Running client simulations (HTTP) via sockets

Browser	Protocol	Cipher Suite Name (OpenSSL)	Forward Sec
Android 7.0 (native)	TLSv1.2	ECDHE-RSA-AES128-GCM-SHA256	256 bit ECDH
Android 8.1 (native)	TLSv1.2	ECDHE-RSA-AES128-GCM-SHA256	253 bit ECDH
Android 9.0 (native)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Android 10.0 (native)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Android 11/12 (native)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Android 13/14 (native)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Chrome 101 (Win 10)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Chromium 137 (Win 11)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Firefox 100 (Win 10)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Firefox 137 (Win 11)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
IE 8 Win 7	No connection		
IE 11 Win 7	No connection		
IE 11 Win 8.1	No connection		
IE 11 Win Phone 8.1	No connection		
IE 11 Win 10	TLSv1.2	ECDHE-RSA-AES256-GCM-SHA384	256 bit ECDH
Edge 15 Win 10	TLSv1.2	ECDHE-RSA-AES256-GCM-SHA384	253 bit ECDH
Edge 101 Win 10 21H2	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Edge 133 Win 11 23H2	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Safari 18.4 (iOS 18.4)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Safari 15.4 (macOS 12.3.1)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Safari 18.4 (macOS 15.4)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
Java 7u25	No connection		
Java 8u442 (OpenJDK)	TLSv1.3	TLS_AES_256_GCM_SHA384	253 bit ECDH
Java 11.0.2 (OpenJDK)	TLSv1.3	TLS_AES_128_GCM_SHA256	256 bit ECDH
Java 17.0.3 (OpenJDK)	TLSv1.3	TLS_AES_256_GCM_SHA384	253 bit ECDH
Java 21.0.6 (OpenJDK)	TLSv1.3	TLS_AES_256_GCM_SHA384	253 bit ECDH
go 1.17.8	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH
LibreSSL 3.3.6 (macOS)	TLSv1.3	TLS_CHACHA20_POLY1305_SHA256	253 bit ECDH
OpenSSL 1.0.2e	TLSv1.2	ECDHE-RSA-AES256-GCM-SHA384	256 bit ECDH
OpenSSL 1.1.1d (Debian)	TLSv1.3	TLS_AES_256_GCM_SHA384	253 bit ECDH
OpenSSL 3.0.15 (Debian)	TLSv1.3	TLS_AES_256_GCM_SHA384	253 bit ECDH
OpenSSL 3.5.0 (git)	TLSv1.3	TLS_AES_256_GCM_SHA384	253 bit ECDH
Apple Mail (16.0)	TLSv1.2	ECDHE-RSA-AES256-GCM-SHA384	256 bit ECDH
Thunderbird (91.9)	TLSv1.3	TLS_AES_128_GCM_SHA256	253 bit ECDH

Rating (experimental)

Rating specs (not complete)	SSL Labs's 'SSL Server Rating Guide' (version 2009q from
Specification documentation	https://github.com/ssllabs/research/wiki/SSL-Server-Rat
Protocol Support (weighted)	0 (0)
Key Exchange (weighted)	0 (0)
Cipher Strength (weighted)	0 (0)
Final Score	0
Overall Grade	T

Grade cap reasons

Grade capped to T. Issues with the chain of trust (self signed CA in chain)

Done 2025-06-15 17:14:33 [0053s] --> 10.190.133.59:44312 (secuctfd.iict-heig-vd.in)

Le scan testssl.sh montre que TLS 1.2 et TLS 1.3 sont activés. Le grade T est normal car la chaîne de confiance (CERTIF_CHAIN.crt) contient le certificat de l'HEIG qui auto signé ce qui n'est pas accepté par testssl.

1. Quelle durée de validité avez-vous choisie pour le certificat du serveur TLS ? Pourquoi ? Comment cette durée va-t-elle évoluer dans un futur proche ?

J'ai choisi une durée de 100 jours car jusqu'au 15 mars 2026 la durée max d'un certificat TLS est de 398 jours puis de 200 jours puis 100 jours pour 2027 puis 47 jours pour 2029. Donc on va vers des certificats avec des durées toujours plus courtes pour le futur à fin de garantir toujours plus de sécurité si un certificat était compromis. Donc j'ai choisi cette durée pour prendre de l'avance jusqu'à 2027 afin de garantir de la sécurité jusqu'à cette date et si c'était en production on automatiserait le renouvellement des certificats.

Problème rencontré

Lors de la première partie sur l'envoi de mail signé un erreur est apparu car après plusieurs essais je ne peux toujours pas envoyé de mail signé. Malgré des tests avec l'adresse mail hes-so et heig