Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

Sécurité des usages TIC

TP n°4

Tiers de confiance, PKI & Certificat

L'ANSSI, « Agence nationale de la sécurité des systèmes d'information »

1 – Allez sur le site de l'ANSSI, `https://cyber.gouv.fr/` et `https://messervices.cyber.gouv.fr` consultez les documents suivants :

- « Zero Trust » :
  - `https://messervices.cyber.gouv.fr/documents-guides/anssi_essentiels_zero_trust_1.0.pdf`
- « Mécanismes cryptographiques : Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques » (version 3.00 du 20 mars 2026) :
  - `https://messervices.cyber.gouv.fr/documents-guides/anssi-guide-mecanismes-crypto-3.00.pdf`
  - Quelle est la taille minimale et recommandée des modules RSA pour le chiffrement asymétrique selon ce guide v3.00 ?
    Regardez en particulier, l'importance donnée à la Cryptographie Post Quantique.
- « Typologie de la menace »
  - `https://www.cert.ssi.gouv.fr/uploads/CERTFR-2026-CTI-002.pdf`
  - *pour référence*

Certificats : chaîne et constitution

2 – Regardez les certificats présents dans votre système d'exploitation ou le navigateur Firefox sous Linux : Existe-t-il des AC françaises ?

3 – a. Allez sur l'infrastructure de gestion de la confiance de l'administration, dite « IGC/A », sur `https://cyber.gouv.fr/igca/`
Quelle taille de clés sont proposées pour le certificat racine ?
Pourquoi en existe-t-il deux versions ?
Existe-t-il d'autre(s) différence(s) concernant les opérations cryptographiques ?

b. Allez sur le lien « *modalités de vérification* » : comment vérifier que vous avez le « *bon* » certificat racine ?
Quel serait le risque sinon ?

4 – Regardez le certificat protégeant l'accès au site sécurisé `https://www.unilim.fr/`.

a. quel algorithme est utilisé pour réaliser la signature électronique ?
b. quel algorithme est utilisé pour calculer l'empreinte du certificat ?
c. quelle est la date d'expiration du certificat ?
d. donnez la chaîne de confiance du site ;
e. quelle est la CA, « *Certificate Authority* », de ce certificat ?
f. allez sur `https://www.amazon.fr/`, est-ce que la chaîne de certificat est la même ?

5 – a. Allez sur `https://www.certinomis.fr/produit/certinomis-decideur`
et sur `https://www.certinomis.fr/produit/certinomis-executif`
Quels sont les usages ?

b. Allez sur
`https://www.certinomis.fr/certinomis-operateur-de-confiance-numerique-2/service-dhorodatage-electronique`
Qu'est-ce que « l'horodataga » ?
Comment est-il sécurisé ?

c. Allez voir sur `https://www.certinomis.fr/notre-metier/les-services-de-confiance-eidas`
quels sont les cinq services de confiance ?
Qu'apporte la nouvelle norme eIDAS 2.0 (European Digital Identity Wallets) ?
Allez sur `https://www.certinomis.fr/nos-oids-sha1`
Qu'est-ce quun OID ? À quoi sert-il ?

d. Allez sur `https://www.certinomis.fr/certinomis-operateur-de-confiance-numerique-2/certinomis-sc`
Quels sont les tarifs ?
Pourquoi un code PIN ou une clé cryptographique ?

e. Allez sur `https://www.certigna.com/famille-de-services/certificat-personne-morale/`
Qu'est-ce qu'un « *Certigna Cachet* » ?
Regardez l'offre de Certigna.

Manipulation d'une AC

Ici, on utilisera RSA, mais en entreprise, on privilégie ECC (P-256/P-384).

6 – Création de l'Autorité de certification

a. en « ligne de commande » :

- créez un répertoire « `MY_CA` » ;
- allez dans ce répertoire ;
- créez un fichier `index.txt` et `serial.txt`, ainsi qu'un sous-répertoire « `newcerts` » :

```
xterm
touch index.txt
echo 01 > serial.txt
mkdir newcerts
```

- Créez le fichier « `root-ca.cnf` » avec le contenu suivant :

```
[ req ]
default_bits = 4096
default_keyfile = ca.key
distinguished_name = req_distinguished_name
x509_extensions = v3_ca
string_mask = nombstr
req_extensions = v3_req

[ req_distinguished_name ]
countryName = Country Name (2 letter code)
countryName_default = FR
countryName_min = 2
countryName_max = 2
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = FRANCE
localityName = Locality Name (eg, city)
localityName_default = Limoges
0.organizationName = Organization Name (eg, company)
0.organizationName_default = Master 1
organizationalUnitName = Organizational Unit Name (eg, section)
organizationalUnitName_default = Service de Certification
commonName = Common Name (eg, Mon Autorite de Certification)
commonName_default = Mon AC du Master 1
commonName_max = 64
emailAddress = Email Address (eg, celle du responsable)
emailAddress_max = 40

[ v3_ca ]
basicConstraints = critical,CA:true
subjectKeyIdentifier = hash

[ v3_req ]
nsCertType = email,server
```

- créez la bi-clé de l'Autorité de Certification :

```
xterm
openssl genrsa -aes128 -out ca.key 4096
```

Mémorisez bien le mot de passe d'accès à la clé privée de l'AC.

- Réalisez l'« **auto-signature** » du certificat de l'AC :

```
xterm
openssl req -new -x509 -days 7300 -config root-ca.cnf -key ca.key -out ca.crt
```

Vous rentrerez différentes informations aux questions posées (le choix par défaut est suffisant).

7 – Création d'un certificat utilisateur

a. Toujours dans le répertoire « `MY_CA` » :

◇ Créez le fichier « `user-cert.cnf` » avec le contenu suivant :

```
[ req ]
default_bits = 2048
default_keyfile = user.key
distinguished_name = req_distinguished_name
string_mask = nombstr
req_extensions = v3_req

[ req_distinguished_name ]
commonName = Common Name (eg, Jean DUPONT)
commonName_max = 64
emailAddress = Email Address (eg, prenom.nom@etu.unilim.fr)
emailAddress_max = 40

[ v3_req ]
nsCertType = client, email
basicConstraints = critical,CA:false
```

◇ Créez votre bi-clé et la « **demande de certificat** » :

xterm
```
openssl genrsa -out user.key 2048
openssl req -new -config user-cert.cnf -key user.key -out user.csr
```

◇ Créez le fichier « `ca-sign.cnf` » avec le contenu suivant :

```
[ ca ]
default_ca = default_CA

[ default_CA ]
dir = .
certs = $dir
new_certs_dir = $dir/newcerts
database = index.txt
serial = serial.txt
RANDFILE = seed.rnd
certificate = ca.crt
private_key = ca.key
default_days = 3650
default_crl_days = 30
default_md = sha256
preserve = yes
x509_extensions = user_cert
policy = policy_anything

[ policy_anything ]
commonName = supplied
emailAddress = supplied

[ user_cert ]
subjectAltName = email:copy
basicConstraints = critical,CA:false
authorityKeyIdentifier = keyid:always
subjectKeyIdentifier    = hash
extendedKeyUsage = clientAuth,emailProtection
```

◇ Faites signer par l'AC, le certificat de l'utilisateur :

xterm
```
openssl ca -config ca-sign.cnf -out user.crt -batch -infiles user.csr
```

◇ Vérifiez le contenu de ce certificat :

xterm
```
openssl x509 -in user.crt -text -noout
```

8 – Utilisez le certificat pour faire de l'authentification pour un serveur Web :

a. lancez openssl en « émulation » de serveur web avec connexion sécurisée SSL utilisant le certificat et la clé privée créés précédemment :

xterm
```
openssl s_server -cert user.crt -key user.key -www
```

b. connectez vous sur ce serveur depuis votre navigateur Web à l'aide de l'url :
`https://localhost:4433/`
- ◇ Qu'est-ce que vous indique le navigateur ?
- ◇ Comment corriger le(s) problème(s) ?

Utilisation de Let's Encrypt

9 – Let's Encrypt vous permet d'obtenir un certificat gratuit pour la sécurité des connexions vers votre site Web. Regardez quelles sont les garanties qu'il vous fournit et comment vous pouvez le récupérer sur votre machine.

Les nouveautés en 2026 :
- ▷ Certificats Domain Validated (DV) gratuits.
- ▷ Validité en forte diminution (transition CA/B Forum : 199 jours max dès fév. 2026 pour certains CA, puis vers 45 jours d'ici 2028).
- ▷ Nouveautés : certificats short-lived (6 jours), nouvelle hiérarchie « Generation Y », suppression par défaut de l'Extended Key Usage « Client Authentication ».

Comment le récupérer sur votre machine ? (certbot ou ACME).

Étude de ED25519

10 – Étude de DSA.

a. Qu'est-ce que veut dire *DSA* ?

b. Pourquoi « *OpenSSH* » a rendu obsolète les clés DSA ?
Des infos sont disponibles à `http://www.openssh.com/legacy.html`

11 –

a. Qu'est-ce que Ed25519 ?
Sur quelle courbe repose-t-il ?
Quel est son algorithme parent ?
Pourquoi l'ANSSI, le CA/B Forum et l'industrie le recommandent-ils fortement par rapport à RSA 4096, ECDSA P-256 et DSA ?

b. Quels sont ses avantages majeurs (citez en au moins 4).

c. Créez votre paire de clés Ed25519 (dans le répertoire MY_CA) :

xterm
```
$ openssl genpkey -algorithm Ed25519 -out ed25519.key
$ openssl pkey -in ed25519.key -pubout -out ed25519.pub
```

d. Signature et vérification d'un document :

xterm
```
$ echo "TP Sécurité TIC - Message confidentiel à signer" > message.txt

# Signature avec Ed25519 (sortie en binaire)
$ openssl pkeyutl -sign -inkey ed25519.key -rawin -in message.txt -out
signature.bin

# Vérification
$ openssl pkeyutl -verify -pubin -inkey ed25519.pub -rawin -in message.txt
-sigfile signature.bin
```

e. Questions :
- ⋆ Quelle est la taille de la clé privée/publique et de la signature Ed25519 ?
- ⋆ Pourquoi Ed25519 est-il plus sûr que DSA/ECDSA mal implémenté ?
- ⋆ Dans quels contextes est-il déjà largement déployé en 2026 ?
- ⋆ Existe-t-il une variante encore plus sécurisée ?

En 2026 : Ed25519 (ou Ed448) est fortement recommandé pour toutes les nouvelles implémentations de signature numérique. RSA 3072+ et P-384 restent acceptés uniquement pour la retro-compatibilité.

12 – a. Qu'est-ce que la « *Logjam Attack* » ?
Vous pourrez aller sur `https://weakdh.org`

b. Pour tester si un serveur Web est vulnérable à cette attaque :

xterm
```
$ openssl s_client -connect www.unilim.fr:443 -cipher "EDH" | grep "Server
Temp Key"
```

Ctrl-c pour arrêter la communication...

xterm
```
$ openssl s_client -connect www.unilim.fr:443 -cipher "EXP"
```