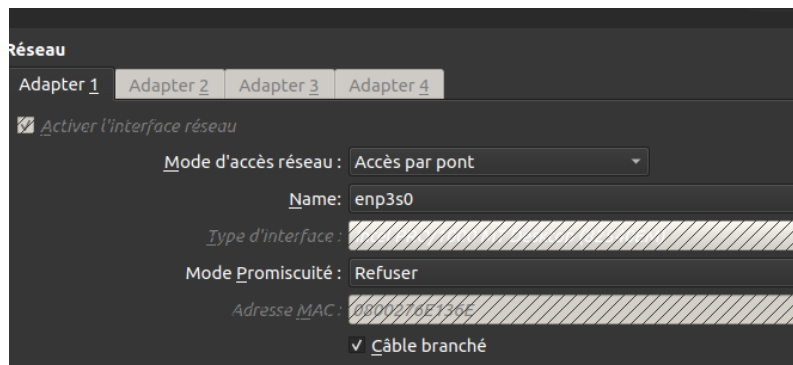


Contrôle crypto avancé : Horodatage

L'étape 1 : Signature du diplôme

D'abord on met notre VM en accès par pont



Voici l'IP donnée par le dhcp 192.168.1.87

```
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    link/ether 08:00:27:6e:13:6e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.87/24 brd 192.168.1.255 scope global dynamic noprefixroute
        eth0
        valid_lft 43155sec preferred_lft 43155sec
        inet6 2a01:e0a:250:10e0:376e:65c0:5f2b:a3e/64 scope global dynamic nopre
            ixroute
            valid_lft 86357sec preferred_lft 86357sec
            inet6 fe80::a00:2478:cfb6:603f/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

On crée le fichier Information.txt



Et on crée une paire de clé privé/publig pour HighUniversity avec la commande
`openssl genrsa -out HighUniversityPrivateKey.pem 2048`

```
Open HighUniversityPrivateKey.pem CPU us
1 -----BEGIN PRIVATE KEY-----
2 MIIIEvgIBA0ANBgkqhkiG9w0BAQEFAASCBAgEAAoIBAQDFeIAjbumVM1Du
3 KPzGe0x0XjTQLiUGNTzEtomj3QH/8wKLuJkao08xAWuPtFAoIUkm1hklmV4bYMR4
4 PuRYQjLTL15CpNldzmurLm9ZF8EJi0tFMfJhJF9PD6IfTox0ioZfIKDMeo87Wkxa
5 C5B7ysDbIZeVPVWv20FCoBBuxgQP5m/hYmRVBz/Eb0J3+whlsIdNBwQNV3N4sZSs
6 1sLHpI+h9u4YFIUzWXZM26SBDmKhtVA6h7h8Ctg0J1WErRV235rqns633Zwqe8le
7 GKQqgwCSRQtXIGomMf1paAA5eJEZzvEPAweV7W03f/QhocEpA3wj9igovg/E6yeH
8 aLE9qyzjAgMBAAECggEAV+FutVYw7WWbCMJ47CQHnXNl0Zdej5zhtHAA4rhzvRp
9 F6FJkc9H+Pk20zgW7uURVxVFD7SXavu/s/vZR2YBeQE+WS5gBh1NUVtS9oVZxVOW
10 cYEa60q7R8YfUmt0mpDjiAZPSXDpuXJQNGqdHfP5RYxHl4U6Uw3kAfCWH5ZETIZq
11 +ED6c4Z4Q8MCPaOt913uJdksfFLrKVMkFi1Ja06A947gToI3+xZrYclHslaPd09+
12 u4+Uc6PLKhCT/8vkwL9Qit6XrGwIfenrtOR8b31a8/r3eLLuhRMCAhtst4T+Cz7G
13 co7BLZ10/mmti1LeLr9CrLLDgq8FGALHRRfN7Rd74QKBgQDhdLJoQcVJm5g7mL9Z
14 NxToYASEuzs4B8qXDKedU/O6WJLTc5zsrECFK1sLHuDeHYUxCMltbZpWHXrCDH
15 mR2hMEuLyjf6m9r089+aUvVReFnrdrTKIV2m73wDYuTdEkWe/XrDMTxfLUYKfzL
16 DbVrNpWb7WdIa6mTbQ+3gH+lqwKBgQDg0TexmpQjC3oymFSZdEP5vYzy1G0VeEH
17 xNvU173tMg3ssa27G9cc99Lpr1cBjQIjOkednxZoWs1HWLSvoD7VfkoAMD2TkQe6
18 pJAb5+80Ut3jBbp1+mC3R5YqnmQkXkPI5YBVx9z7bZd4bbtirY9nxyHxB7VeI2L
19 Mw4dCJdtqQKBgQCSso7/fYQ0f/SPGrWvBEMrU3a9D5UoEDDXLESXSoPpRYs6mqDv
20 DtXktuizJENhJEbSicHj6KYF20A8qDhwB3YbEsS16PHJtC2sqUxeKwqKp0e7XnGt
21 HJOxWGC/Umd2YhzIb4I1Hgrro6sU3QU7uS7JgzABpQqmOb+OONGYMHlcnwKBgQC9
22 z9IU2E2LKRcl84xSEL81Hgi3qdt9ibe4su6Bln8iZ6ggAf2XEqdUj3qNdnBKEND0
23 alsD10PMLWnviEL+b0IhywnsB0prnG+Vkw/Q390gJYhOoc/KBc2P+drmcxLE8Mrt
24 hOt8cPtgURB7gxMvHaJom340dFLB1Y7FetBUxawjGQKBgBC3dxa5ZI3dMRge+gWC
25 FaPaxDMnLoC91EG6AGZM23ae6wGE0U29bFalkLS3zhEN+1w5UbvQYXZErC/2
26 2cr+rn/+h4KxhhqmnZtJmDtY0/Roxcqvd1nuWlxPv+b3R9mB3BuhAHL7kiJk2ZMZ
27 rw3lFyA0tInvM/MaymxCD1rv
28 -----END PRIVATE KEY-----
```

On extrait la clef publique et on la sauvegarde sous le nom : HighUniversityPublicKey.key avec la commande :

```
openssl rsa -in HighUniversityPrivateKey.pem -pubout -out HighUniversityPublicKey.key
```

et on signe le fichier **Information.txt** avec la clé privée :

```
(kali@kali)-[~/test]
$ openssl rsa -in HighUniversityPrivateKey.pem -pubout -out HighUniversityP
ublicKey.key

writing RSA key

(kali@kali)-[~/test]
$ openssl rsautl -sign -in Information.txt -inkey HighUniversityPrivateKey.
pem -out Signature.txt

The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.

(kali@kali)-[~/test]
```

L'étape 2 : Horodatage certifié

Ensuite on crée une requête d'horodatage avec la commande

```
openssl ts -query -data Signature.txt -sha256 -out Signature.txt.tsq
```

On envoie la requête au serveur [www.freetsa.org](https://freetsa.org) et on récupère la réponse :

```
curl -H "Content-Type: application/timestamp-query" --data-binary
'@Signature.txt.tsq' https://freetsa.org/tsr > Signature.tsr
```

Afficher la date de la signature avec la commande suivante:

```
#Openssl ts -reply -in Signature.tsr -text
```



```
(kali㉿kali)-[~/test]
└─$ steghide info Diplome.jpg
"Diplome.jpg":
  format: jpeg
  capacity: 3.7 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "info.zip":
    size: 1.9 KB
    encrypted: rijndael-128, cbc
    compressed: yes
```

L'étape 4 : Vérification de la date de la signature

On a télécharger les certificat venant de freetsa.org

wget <https://freetsa.org/files/cacert.pem>

wget <https://freetsa.org/files/tsa.crt>

Puis on extrait les données cachées de l'image Diplome.jpg , décompresse le fichier info.zip , et on vérifie l'horodatage avec OpenSSL

```
(kali㉿kali)-[~/test/New Folder]
└─$ steghide extract -sf Diplome.jpg
Enter passphrase:
wrote extracted data to "info.zip".

(kali㉿kali)-[~/test/New Folder]
└─$ unzip info.zip
Archive: info.zip
  extracting: Signature.txt
  extracting: Signature.txt.tsq
  inflating: Signature.tsr

(kali㉿kali)-[~/test/New Folder]
└─$ openssl ts -verify -in Signature.tsr -queryfile Signature.txt.tsq -CAfile cacert.pem -untrusted tsa.crt
Using configuration from /usr/lib/ssl/openssl.cnf
Warning: certificate from 'tsa.crt' with subject '/O=Free TSA/OU=TSA/description=This certificate digitally signs documents and time stamp
ail.com/L=Wuerzburg/C=DE/ST=Bayern' is not a CA cert
Verification: OK

(kali㉿kali)-[~/test/New Folder]
└─$
```

L'étape 5 : vérification de la signature de HighUniversity

Enfin on vérifie la signature avec la clé publique avec la commande

openssl rsautl -verify -in Signature.txt -pubin -inkey

HighUniversityPublicKey.key -out hashSignature.txt

Et comme on peut voir hashSignature.txt est bien pareil que Information.txt

```
File Actions Edit View Help
(kali㉿kali)-[~/test]
└─$ openssl rsautl -verify -in Signature.txt -pubin -inkey HighUniversityPublicKey.key -out hashSignature.txt

The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.

Open Infor... Save hashSignature.txt ~/test
1 Nom : Dupont
2 Prénom : Frédéric
3 Date de naissance : 01/01/1998
4 Spécialité : Informatique
5 Année universitaire : 2020/2021

1 Nom : Dupont
2 Prénom : Frédéric
3 Date de naissance : 01/01/1998
4 Spécialité : Informatique
5 Année universitaire : 2020/2021
```

