Aide-Mémoire

Comme il s'agit d'une API assez vaste (mais cohérente), il faudra bien utiliser la documentation ci-dessous :

Système : station DORANCO ou VM Kali-Linux avec openssl

Ligne de commande : https://www.madboa.com/geek/openssl/

Guide pratique sur les certificats : https://guidespratiques.traduc.org/vf/SSL-Certificates-HOWTO.html

Syntaxe générale :

```
openssl commande -option -in fichierentree -out fichiersortie
```

Il est possible de faire travailler sur l'entrée standard (noter le -n pour ne pas avoir de \n rajouté)

```
echo -n "message à encoder" | openssl enc -base64
```

Algorithmes de base

Encodage

Encodez vos prénom et nom de famille en

Hexadécimal

```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n "ASR2-DORANCO" | xxd -p
415352322d444f52414e434f
```

Binaire

```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n "ASR2-DORANCO" | xxd -b
00000000: 01000001 01010011 01010010 00110010 00101101 01000100  ASR2-D
00000006: 01001111 01010010 01000001 01001110 01000011 01001111  ORANCO
```

Base64

```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n "ASR2-DORANCO" | openssl enc -base64
QVNSMi1ET1JBTkNP
```

Hachage

Hacher vos prénom et nom de famille en

- MD5

```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n ASR2-DORANCO | md5sum
67dec761a1e89841c6f54ed75e77d442  -
```

- SHA-1

```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n ASR2-DORANCO | sha1sum
5b4bb23dc67adad7e3c8af4949fe70554dfc2df4
```

- SHA-256

```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n ASR2-DORANCO | sha256sum
50dc952ae839fe9e06f8ef41fb3a01b6f64ff152033c6b35278c5b65ab1af5b5
```

- RIPE-MD160

```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo -n ASR2-DORANCO | openssl rmd160
(stdin)= 373b80eb82a9fd4fc956827a0fdd9abef69668a9
```

Certificats X509

Certificats CNRS et le Certificat de Godard

Effectuer les commandes suivantes.

```
openssl x509 -text -in CNRS2.pem
```

```
openssl verify -CAfile CNRS2.pem -purpose any CNRS2.pem
```

```
openssl x509 -text -in CNRS2-Standard.pem
```

```
openssl verify -CAfile CNRS2.pem -purpose any CNRS2-Standard.pem
```

```
openssl verify -CAfile CNRS2.pem -purpose any godard.pem
```

```
openssl verify -CAfile CNRS2-Standard.pem -purpose any godard.pem
```

```
cat CNRS2.pem CNRS2-Standard.pem > chaine.pem
```

```
openssl verify -CAfile chaine.pem -purpose any godard.pem
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl x509 -text -in CNRS2.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C = FR, O = CNRS, CN = CNRS2
        Validity
            Not Before: Jan 21 08:51:13 2009 GMT
            Not After : Jan 21 08:51:13 2029 GMT
        Subject: C = FR, O = CNRS, CN = CNRS2
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:b8:ba:77:ab:57:9d:12:25:19:15:44:8a:23:55:
                    da:63:0d:06:6d:59:eb:c1:04:a0:17:79:12:f3:27:
                    82:48:ea:8d:be:3c:cb:36:35:78:1b:04:d5:29:03:
                    b5:69:48:29:fb:a7:a7:6b:c4:dc:d2:29:10:f3:e2:
                    8a:bd:6b:1c:0c:33:1a:2e:ed:a6:84:84:36:26:cc:
                    a4:ef:fe:24:f9:1b:5f:ae:2d:59:03:50:2e:0f:08:
                    71:f5
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Subject Key Identifier:
                50:97:B6:0D:F7:AC:33:17:AF:F1:1D:46:3C:6B:3B:FF:00:A0:E5:E5
            X509v3 Authority Key Identifier:
                keyid:50:97:B6:0D:F7:AC:33:17:AF:F1:1D:46:3C:6B:3B:FF:00:A0:E5:E5
                DirName:/C=FR/O=CNRS/CN=CNRS2
                serial:00
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
    Signature Algorithm: sha1WithRSAEncryption
         34:31:6b:40:21:bf:87:f5:5e:7c:a1:a5:f5:ff:18:d2:24:e6:
         50:bc:67:98:ae:3b:0c:18:c7:88:08:ad:93:d1:06:09:90:6a:
         fe:a8:14:f1:90:0d:6b:48:70:d9:d0:67:7a:85:94:6a:9f:f0:
         69:9d:d6:78:ff:d7:b4:fc:ea:08:ed:ce:5a:08:5a:53:2e:af:
         f5:f2:de:3b:b2:99:39:fd:7d:b1:eb:cc:54:74:05:17:10:66:
         47:0c:ac:51:68:1c:2c:0f:7d:a7:d6:a0:d1:55:f5:3e:93:f4:
         aa:8d:19:cd
-----BEGIN CERTIFICATE-----
MIIDbTCCAlWgAwIBAgIBADANBgkqhkiG9w0BAQUFADAsMQswCQYDVQQGEwJGUjEN
MAsGA1UEChMEQ05SUzEOMAwGA1UEAxMFQ05SUzIwHhcNMDkwMTIxMDg1MTEzWhcN
MjkwMTIxMDg1MTEzWjAsMQswCQYDVQQGEwJGUjENMAsGA1UEChMEQ05SUzEOMAwG
ADQxa0Ahv4f1XnyhpfX/GNIk5lC8Z5iuOwwYx4gIrZPRBgmQav6oFPGQDWtlcNnQ
Z3qFlGqf8Gmd1nj/17T86gjtzloIWlMur/Xy3juymTn9fbHrzFR0BRcQZl8azjb1
S+Gfk7H4K4qIETH882qjjR3WCl3q+UzW4kLmPAsu7GJMaQ4sJp6mjec3r19qIQxF
6S/zSbcb9V8hUiAkCjDKYr0BgmZMQfLLQv1ir0R7KWkZFaWBROjD3nyUM2aeE49R
g89T+A3nrrxO+oHIHywpZN1rYyOcH8f3dzriMuppIfWdOcJ3+ga00gihkEcMrFFo
HCwPfafWoNFV9T6T9KqNGc0=
-----END CERTIFICATE----
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl verify -CAfile CNRS2.pem -purpose any CNRS2.pem
CNRS2.pem : OK
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl x509 -text -in CNRS2-Standard.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 3 (0x3)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C = FR, O = CNRS, CN = CNRS2
        Validity
            Not Before: Jan 21 09:03:52 2009 GMT
```

```
        Not After : Jan 20 09:03:52 2029 GMT
    Subject: C = FR, O = CNRS, CN = CNRS2-Standard
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
        RSA Public-Key: (2048 bit)
        Modulus:
          00:9c:a9:64:6a:b4:07:23:19:c3:be:08:08:c4:e2:
          2a:5d:ed:d4:28:de:fe:3f:a0:ed:92:44:6b:16:47:
          35:13:37:8d:74:a9:f5:4d:83:e4:05:1b:8f:08:61:
          4c:de:77:5b:d7:a9:ff:bb:65:9d:c9:ac:3c:0f:f1:
          51:85:eb:e0:94:a3:f9:97:08:11:8c:24:f0:0f:ba:
          38:b4:2c:d6:d7:6c:4e:63:fe:b4:31:9f:ef:8f:a6:
          5a:a5
        Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Basic Constraints: critical
        CA:TRUE
      X509v3 Subject Key Identifier:
        11:E3:D9:D1:52:47:1B:59:B1:3C:1B:78:66:6B:F4:A1:88:ED:0A:5B
      X509v3 Authority Key Identifier:
        keyid:50:97:B6:0D:F7:AC:33:17:AF:F1:1D:46:3C:6B:3B:FF:00:A0:E5:E5
        DirName:/C=FR/O=CNRS/CN=CNRS2
        serial:00
      X509v3 Key Usage: critical
        Certificate Sign, CRL Sign
      X509v3 CRL Distribution Points:
        Full Name:
          URI:http://crls.services.cnrs.fr/CNRS2/getder.crl
  Signature Algorithm: sha1WithRSAEncryption
      4f:e9:e3:17:e6:4c:27:f5:17:6a:50:55:ea:ee:cf:4c:aa:bd:
      ee:2c:dd:76:3e:59:a0:2f:0b:68:89:da:43:d0:1e:1d:2c:6b:
      12:f9:67:13:9e:65:2b:a4:84:d2:31:cf:82:a0:ca:ef:22:e8:
      10:6c:e9:cf:17:7c:ac:ad:87:85:4c:8a:e1:d5:7b:4c:e6:d9:
      0e:b2:a1:e4:29:9e:20:af:0f:3e:7b:b9:7f:24:4c:32:e0:88:
      54:bc:61:26:10:eb:91:d5:e9:f7:ee:d0:f6:0b:0b:fa:a6:90:
      dd:60:f4:40:9d:0b:a7:9c:f8:ce:2f:21:38:fe:ab:06:bf:da:
      0d:77:fa:67
-----BEGIN CERTIFICATE-----
MIIDtjCCAp6gAwIBAgIBAzANBgkqhkiG9w0BAQUFADAsMQswCQYDVQQGEwJGUjEN
MAsGA1UEChMEQ05SUzEOMAwGA1UEAxMFQ05SUzIwHhcNMDkwMTIxMDkwMzUyWhcN
MjkwMTIwMDkwMzUyWjA1MQswCQYDVQQGEwJGUjENMAsGA1UEChMEQ05SUzEXMBUG
A1UEAxMOQ05SUzItU3RhbmRhcmQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
R34szr/IMA0ARvYY5mw/MH6QbhCNHzlJz5Np68iewmdzr0qL9nxCKN2l3e2vyVUb
ZD9UvGEmEOuR1en37tD2Cwv6ppDdYPRAnQunnPjOLyE4/qsGv9oNd/pn
-----END CERTIFICATE-----
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl verify -CAfile CNRS2.pem -purpose any CNRS2-Standard.pem
CNRS2-Standard.pem: OK
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl verify -CAfile CNRS2.pem -purpose any Godard.pem
C = FR, O = CNRS, OU = UMR7279, CN = Emmanuel Godard, emailAddress = emmanuel.godard@lif.univ-mrs.fr
error 20 at 0 depth lookup: unable to get local issuer certificate
error Godard.pem: verification failed
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl verify -CAfile CNRS2-Standard.pem -purpose any Godard.pem
C = FR, O = CNRS, CN = CNRS2-Standard
error 2 at 1 depth lookup: unable to get issuer certificate
error Godard.pem: verification failed
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ cat CNRS2.pem CNRS2-Standard.pem > chaine.pem
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ openssl x509 -text -in Godard.pem
Data:
    Version: 3 (0x2)
    Serial Number: 48517 (0xbd85)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C = FR, O = CNRS, CN = CNRS2-Standard
    Validity
      Not Before: Oct 19 10:13:19 2016 GMT
      Not After : Oct 19 10:13:19 2018 GMT
    Subject: C = FR, O = CNRS, OU = UMR7279, CN = Emmanuel Godard, emailAddress = emmanuel.godard@lif.univ-mrs.fr
    Subject Public Key Info:
Public Key Algorithm: rsaEncryption
        RSA Public-Key: (2048 bit)
        Modulus:
          00:dc:12:58:3b:eb:fc:9d:5c:c8:b2:c1:c6:6e:93:
          89:59:e1:fc:c3:84:9a:a1:2c:0c:69:f3:6b:c3:60:
          80:1c:64:33:89:ea:7f:cd:2b:3f:b4:a0:bb:ed:9d:
          7f:b5:06:ee:d4:89:3a:23:83:00:d0:f7:7f:95:c8:
          55:37:b8:97:53:6b:de:1a:7b:bc:7e:ab:7c:8a:47:
          75:57:3a:4d:53:d1:da:1a:03:74:b5:7d:c7:03:c7:
          b0:1b
        Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Basic Constraints: critical
        CA:FALSE
      Netscape Cert Type:
        SSL Client, S/MIME, Object Signing
      X509v3 Key Usage: critical
        Digital Signature, Non Repudiation, Key Encipherment
      Netscape Comment:
        Certificat CNRS2-Standard. Pour toute information se reporter  http://igc.services.cnrs.fr/CNRS2-Standard/
      X509v3 Subject Key Identifier:
        AD:CE:85:51:8C:61:B8:FD:F7:FE:67:91:C4:81:67:34:51:9D:D5:D3
      X509v3 Authority Key Identifier:
        keyid:11:E3:D9:D1:52:47:1B:59:B1:3C:1B:78:66:6B:F4:A1:88:ED:0A:5B
        DirName:/C=FR/O=CNRS/CN=CNRS2
        serial:03
      X509v3 Subject Alternative Name:
        email:emmanuel.godard@lif.univ-mrs.fr
      X509v3 CRL Distribution Points:
        Full Name:
          URI:http://crls.services.cnrs.fr/CNRS2-Standard/getder.crl
  Signature Algorithm: sha1WithRSAEncryption
    16:5c:b4:4d:68:4f:b9:93:7c:18:4a:a9:29:89:58:71:54:f0:
    7b:b1:44:ff:f4:eb:06:06:78:9e:5b:e2:bd:76:aa:f3:93:3e:
    a4:aa:38:fc:f2:90:70:90:92:3b:ea:a2:f2:0d:49:07:02:2f:
    ec:8a:93:1d
-----BEGIN CERTIFICATE-----
MIIExzCCA6+gAwIBAgIDAL2FMA0GCSqGSIb3DQEBBQUAMDUxCzAJBgNVBAYTAkZS
MQ0wCwYDVQQKEwRDTlJTMRcwFQYDVQQDEw5DTlJTMi1TdGFuZGFyZDAeFw0xNjEw
UzItU3RhbmRhcmQuIFBvdXIgdG91dGUgaW5mb3JtYXRpb24gc2UgcmVwb3J0ZXIg
4CBodHRwOi8vaWdjLnNlcnZpY2VzLmNucnMuZnIvQ05SUzItU3RhbmRhcmQvMB0G
DmDn6Zf9yOyR0DmnJcCL4WbCfJNExp3ky1Y8As2j7QQHOSJvvvWhcaU6wfq1/EjG
N/DucOWkqjj88pBwkJI76qLyDUkHAi/sipMd
-----END CERTIFICATE-----
```

La clé publique de Godard est-elle authentique ?

Non elle n'est pas authentique. CA : FALSE