Hugo BEHERAY Zoé CROUZET Group 1

Lab 2 : Nouvelles Technologies et Société

1. Exercise ”Symmetric cryptography”

2.

```
sh-3.2# touch AuthDataBob
sh-3.2# ls
AuthDataBob    BobPrivateKey    BobPublicKey    BobSignature
sh-3.2# openssl dgst -sha256 -sign BobPrivateKey -out BobSignature AuthDataBob
sh-3.2# cd ..
sh-3.2# ls
Alice    Bob
sh-3.2# cd Alcie
sh: cd: Alcie: No such file or directory
sh-3.2# cd Alice
sh-3.2# ls
AlicePrivateKey    AlicePublicKey    AuthDataBob    BobPublicKey    BobSignature
sh-3.2# openssl dgst -sha256 -verify BobPublicKey -signature BobSignature
example.txt
example.txt: No such file or directory
sh-3.2# openssl dgst -sha256 -verify BobPublicKey -signature BobSignature
AuthDataBob
Verified OK
```

We notice that verification is OK.

3.

```
sh-3.2# touch AuthDataAlice
sh-3.2# ls
AlicePrivateKey    AlicePublicKey    AuthDataAlice    AuthDataBob    BobPublicKey
   BobSignature
sh-3.2# vim AuthData
AuthDataAlice  AuthDataBob
sh-3.2# vim AuthData
AuthDataAlice  AuthDataBob
sh-3.2# vim AuthDataAlice
sh-3.2# vim AuthDataAlice

[No write since last change]
/bin/sh: quit: command not found

shell returned 127

Press ENTER or type command to continue
sh-3.2#
```

```
sh-3.2# openssl dgst -sha256 -sign AlicePrivateKey -out AliceSignature
AuthDataAlice
sh-3.2# ls
AlicePrivateKey    AlicePublicKey    AliceSignature    AuthDataAlice
AuthDataBob    BobPublicKey    BobSignature
sh-3.2# cd ..
sh-3.2# cd Bob
sh-3.2# openssl dgst -sha256 -verify AlicePublicKey -signature AliceSignature
AuthDataAlice
Verified OK
sh-3.2#
```

We notice that verification is OK.

4.

```
sh-3.2# openssl rand -hex -out SymKey 64
sh-3.2# ls
AlicePublicKey    AliceSignature    AuthDataAlice    AuthDataBob    BobPrivateKey
   BobPublicKey    BobSignature    SymKey
sh-3.2# openssl rsautl -encrypt -inkey AlicePublicKey  -pubin -in SymKey -out
SymKeyEncrypted
sh-3.2# ls
AlicePublicKey    AliceSignature    AuthDataAlice    AuthDataBob    BobPrivateKey
   BobPublicKey    BobSignature    SymKey        SymKeyEncrypted
sh-3.2# cd ..
sh-3.2# ls
Alice    Bob
sh-3.2# cd Alice
sh-3.2# ls
AlicePrivateKey    AlicePublicKey    AliceSignature    AuthDataAlice
AuthDataBob    BobPublicKey    BobSignature    SymKeyEncrypted
sh-3.2# openssl rsautl -decrypt -inkey AlicePrivateKey -in SymKeyEncrypted -out
SymKey
sh-3.2#
```

5.

```
openssl enc -aes-128-cbc -kfile SymKey -in DataAlice -out DataAliceEncrypted
openssl enc -d -aes-128-cbc -kfile SymKey -in DataAliceEncrypted -out DataAlice
sh-3.2# cat DataAlice
douze douze
sh-3.2# openssl enc -d -aes-128-cbc -kfile SymKey -in DataAliceEncrypted -out
DataAlice
sh-3.2# cat DataAlice
douze douze
sh-3.2# openssl enc -aes-128-cbc -kfile SymKey -in DataBob -out DataBobEncrypted
sh-3.2# cd ..
sh-3.2# cd Alice
sh-3.2# ls
AlicePrivateKey        AliceSignature        AuthDataBob        BobSignature
   DataAliceEncrypted    SymKey
AlicePublicKey        AuthDataAlice        BobPublicKey        DataAlice
DataBobEncrypted    SymKeyEncrypted
sh-3.2# openssl enc -d -aes-128-cbc -kfile SymKey -in DataBobEncrypted -out
DataBob
sh-3.2# cat DataBob
```

```
Message to Alice
sh-3.2#
```

2. Exercise ”Certificates X509”

2.

```
sh-3.2# openssl s_client -connect www.lcl.fr:443 > CertificateLCL
depth=3 C = GB, ST = Greater Manchester, L = Salford, O = Comodo CA Limited, CN =
AAA Certificate Services
verify return:1
depth=2 C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST Network, CN =
USERTrust RSA Certification Authority
verify return:1
depth=1 C = GB, ST = Greater Manchester, L = Salford, O = Sectigo Limited, CN =
Sectigo RSA Organization Validation Secure Server CA
verify return:1
depth=0 C = FR, ST = Auvergne-Rh\C3\B4ne-Alpes, O = CREDIT LYONNAIS SA, CN =
lcl.fr
verify return:1
```

3.

```
openssl x509 -noout -in CertificateLCL  -issuer
issuer= /C=GB/ST=Greater Manchester/L=Salford/O=Sectigo Limited/CN=Sectigo RSA
Organization Validation Secure Server CA
```

5.

```
openssl x509 -noout -in CertificateLCL  -dates
notBefore=Dec 21 00:00:00 2021 GMT
notAfter=Dec 21 23:59:59 2022 GMT
```

6.

```
openssl x509 -noout -in CertificateLCL -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            aa:d2:50:3b:b3:81:e0:4f:59:37:ad:9c:2f:13:3d:25
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited,
CN=Sectigo RSA Organization Validation Secure Server CA
        Validity
            Not Before: Dec 21 00:00:00 2021 GMT
            Not After : Dec 21 23:59:59 2022 GMT
        Subject: C=FR, ST=Auvergne-Rh\xC3\xB4ne-Alpes, O=CREDIT LYONNAIS SA,
CN=lcl.fr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ba:72:be:e8:ef:59:d3:39:7a:41:0a:3b:d8:7a:
```

```
                    c9:e7:23:bd:73:e0:94:b1:8d:eb:61:18:22:bf:83:
                    10:ca:22:0e:87:0e:46:6a:0c:8c:24:03:26:75:9c:
                    bf:7c:9d:98:0b:15:c1:b5:91:fe:59:e9:bd:bf:cf:
                    8e:26:d4:28:65:11:ec:3a:de:e0:62:4e:50:8c:bf:
                    3a:8e:2f:95:49:5c:25:a3:d0:cd:83:eb:6e:68:fe:
                    e9:6a:68:07:66:74:e1:e8:cd:7a:84:47:99:a8:cc:
                    65:ef:9c:87:6c:02:91:24:d7:f1:8b:a4:f6:2c:9f:
                    cb:fa:2e:cf:2c:18:7d:75:61:b3:16:0f:fb:6b:74:
                    ae:ce:9d:05:fd:36:82:ee:a8:93:72:47:c2:5c:b4:
                    ae:00:f0:33:b0:fd:6f:3d:c7:dd:37:bd:70:6b:e4:
                    69:c3:a4:70:1e:f0:06:d1:46:41:60:12:93:bb:25:
                    e5:86:bb:2d:1c:38:3d:ad:19:b6:79:0a:13:ae:c6:
                    63:db:ed:26:a9:d7:47:d0:32:12:d0:ea:93:18:eb:
                    11:9a:ea:7e:6a:12:ae:61:d7:1c:45:d7:3a:d3:a9:
                    18:66:8a:e6:00:ec:9e:a8:1f:40:98:7c:f9:14:4a:
                    bf:6d:55:2a:2f:12:14:52:62:9c:ac:61:f2:e9:27:
                    a2:55
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Authority Key Identifier:

keyid:17:D9:D6:25:27:67:F9:31:C2:49:43:D9:30:36:44:8C:6C:A9:4F:EB

            X509v3 Subject Key Identifier:
                0F:6A:9B:2F:A8:45:FB:E2:65:75:2C:CA:A8:5A:D3:76:A9:0A:FD:5C
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Certificate Policies:
                Policy: 1.3.6.1.4.1.6449.1.2.1.3.4
                  CPS: https://sectigo.com/CPS
                Policy: 2.23.140.1.2.2

            X509v3 CRL Distribution Points:

                Full Name:

URI:http://crl.sectigo.com/SectigoRSAOrganizationValidationSecureServerCA.crl

            Authority Information Access:
                CA Issuers -
URI:http://crt.sectigo.com/SectigoRSAOrganizationValidationSecureServerCA.crt
                OCSP - URI:http://ocsp.sectigo.com

            1.3.6.1.4.1.11129.2.4.2:
                ...i.g.v.F.U.u..
O...i..}.,At..I.....p.mG...}.mC......GOE.!.....6b6..`..N...2>....Z.j........
>..s...4..G...d)...^X....&.....
{.v.A...."FJ...:.B.^N1.....K.h..b......}.mC......GOE. B...2...3....
i|....Q..B.J...y...!....5.W{L.u..Hgp..pr.._#..jZf.].#.u.)y...99!.Vs.c.w..W}.`
...,.6I.... O..?.#@.Q..R...W.%,`.q.'&.do2D.6 ..u.
            X509v3 Subject Alternative Name:
                DNS:lcl.fr, DNS:www.lcl.fr
    Signature Algorithm: sha256WithRSAEncryption
         4a:6a:ad:01:4b:6c:a0:9d:43:5f:71:03:f4:6a:b4:49:f9:67:
```

```
         08:8c:89:22:03:0a:fb:81:99:af:4b:5d:93:fd:e2:d2:36:2e:
         e4:da:b5:00:a6:08:31:d4:85:f8:63:00:69:fe:f5:eb:1a:62:
         cf:ee:84:52:11:31:93:ef:f8:8b:c3:ed:e1:2f:b7:e7:8b:d9:
         2b:bc:d9:5e:06:59:46:b8:d5:e5:33:72:a3:c0:17:a2:b2:88:
         20:64:fe:c1:ac:7e:5d:2f:e5:f2:3b:52:0b:53:34:cf:63:dd:
         48:cb:18:96:79:86:c2:a5:46:70:33:dd:db:93:c2:3f:af:fa:
         7c:df:fb:7f:96:57:14:ef:44:83:44:d3:f5:2f:cb:9b:80:2b:
         6f:10:1a:5a:79:d8:e6:56:16:ab:b4:bb:a4:8d:64:d3:59:e1:
         01:5a:da:6b:c7:f4:ff:94:53:da:1b:50:fd:d3:38:64:3f:36:
         12:a0:b3:11:86:e0:2d:70:0e:33:62:a4:c0:6f:42:b2:b0:e6:
         f3:43:d8:fd:f0:26:7d:44:80:89:55:67:33:fe:aa:ff:a2:cd:
         5f:19:f3:66:6b:d4:41:fc:9a:fb:16:b7:59:fa:bf:3d:38:6e:
         9c:a6:3f:47:a2:02:46:e6:c0:8c:86:29:0f:ce:bf:1c:d3:0c:
         2a:63:37:e8
```

7.

```
sh-3.2# openssl x509 -noout -in CertificateLCL  -serial
serial=AAD2503BB381E04F5937AD9C2F133D25
```

8.

```
openssl x509 -noout -in CertificateLCL  -pubkey
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAunK+6O9Z0zl6QQo72HrJ
5yO9c+CUsY3rYRgiv4MQyiIOhw5GagyMJAMmdZy/fJ2YCxXBtZH+Wem9v8+OJtQo
ZRHsOt7gYk5QjL86ji+VSVwlo9DNg+tuaP7pamgHZnTh6M16hEeZqMxl75yHbAKR
JNfxi6T2LJ/L+i7PLBh9dWGzFg/7a3Suzp0F/TaC7qiTckfCXLSuAPAzsP1vPcfd
N71wa+Rpw6RwHvAG0UZBYBKTuyXlhrstHDg9rRm2eQoTrsZj2+0mqddHODISOOqT
GOsRmup+ahKuYdccRdc6O6kYZormAOyeqB9AmHz5FEq/bVUqLxIUUmKcrGHy6Sei
VQIDAQAB
-----END PUBLIC KEY-----
```

9.

- *Generate a key pair (2048 bits) for a server protected by a password of your choice*

```
sh-3.2# openssl genrsa -out ServerKeyPair -passout pass:password 2048
Generating RSA private key, 2048 bit long modulus
......................+++
................................................................................
.............+++
e is 65537 (0x10001)
```

- *Extract the public key of the server ServerPublicKey as we have seen in the previous session*

```
sh-3.2# openssl rsa -in ServerKeyPair -pubout -out ServerPublicKey
writing RSA key
```

- *Rename ServerKeyPair by entering the UNIX command: mv ServerKeyPair ServerPrivateKey*

```
sh-3.2# mv ServerKeyPair ServerPrivateKey
sh-3.2# ls
Alice            Bob            CertificateLCL        ServerPrivateKey
ServerPublicKey
sh-3.2#
```

- *Generate the certificate request for the server ServerRequest.csr*

```
sh-3.2# openssl req -new -key ServerPrivateKey -out ServerRequest.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:FR
State or Province Name (full name) []:Ile de France
Locality Name (eg, city) []:Paris
Organization Name (eg, company) []:ECE
Organizational Unit Name (eg, section) []:4th year
Common Name (eg, fully qualified host name) []:myserver.fr
Email Address []:youremail

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
```

- *Generate a key (4096 bits) pair CAKeyPair for a CA protected by a password of your choice*

```
sh-3.2# openssl genrsa -out CaKeyPair -passout pass:password 4096
Generating RSA private key, 4096 bit long modulus
................................................................................
..................................++
..................++
e is 65537 (0x10001)
```

- *Generate a self-signed certificate CACertificate.crt for the CA*

```
sh-3.2# openssl req -x509 -new -key CAKeyPair -out CACertificate.crt -days 500
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:FR
State or Province Name (full name) []:Ile de france
Locality Name (eg, city) []:Paris
```

```
Organization Name (eg, company) []:ECE
Organizational Unit Name (eg, section) []:ECE
Common Name (eg, fully qualified host name) []:ECE
Email Address []:CAemail
```

- *Generate the certificate of the server ServerCertificate.crt by using the certificate and keys of the CA*

```
sh-3.2# openssl x509 -req -in ServerRequest.csr -CA CACertificate.crt -CAkey
CAKeyPair -CAcreateserial -out ServerCertificate.crt -days 500 -sha256
Signature ok
subject=/C=FR/ST=Ile de France/L=Paris/O=ECE/OU=4th
year/CN=myserver.fr/emailAddress=youremail
Getting CA Private Key
```

- *You can verify the certificate of the server by entering the OpenSSL command*

```
sh-3.2# openssl verify -CAfile CACertificate.crt ServerCertificate.crt
ServerCertificate.crt: OK
sh-3.2#
```

We notice that the certificate is verified.