Lab 2

1)2)
le fichier AuthDataBob à bien été crée.

```
[guillaume-alain.priso-totto@r05p03 Bob]$ touch AuthDataBob
[guillaume-alain.priso-totto@r05p03 Bob]$ echo > AuthDataBob "je suis bob le bricoleur"
[guillaume-alain.priso-totto@r05p03 Bob]$ ls
AuthDataBob  BobDocument  BobPrivateKey  BobPublicKey
[guillaume-alain.priso-totto@r05p03 Bob]$ cqt AuthDataBob
cqt: command not found
[guillaume-alain.priso-totto@r05p03 Bob]$ cat AuthDataBob
je suis bob le bricoleur
```

Ensuite lorsque l'on lance cette commande , Alice vérifie l'authenticité de la cléf public de bob a l'aide d'une signature crée par la cléf privé de Bob et de son fichier.

```
[guillaume-alain.priso-totto@r05p03 Alice]$ openssl dgst -sha256 -verify BobPublicKey -signature BobSignature AuthDataBob
Verified OK
```

3)
Donc ici on a bien signé le fichier AuthDataAlice grace a sa cléf privé . La signature est "AliceSignature.

```
[guillaume-alain.priso-totto@r05p03 Alice]$ touch AuthDataAlice
[guillaume-alain.priso-totto@r05p03 Alice]$ echo > AuthDataAlice "Slt c alice la fleuriste"
[guillaume-alain.priso-totto@r05p03 Alice]$ openssl dgst -sha256 -sign AlicePrivateKey -out AliceSignature AuthDataAlice
[guillaume-alain.priso-totto@r05p03 Alice]$ ls
AliceDoc  AlicePrivateKey  AlicePublicKey  AliceSignature  AuthDataAlice  AuthDataBob  BobPublicKey  BobSignature
[guillaume-alain.priso-totto@r05p03 Alice]$ cat AliceSignature
Á.ý²Ù/æ7íÍzʁÕj)JdiABÓ%DýRø/}9WäcSG²Ù+ÎÇ¿7ñ%wÒ[$ü$uí▯3@¶6¹··ÿ¯þ9¶êõ_
                    ]üó¿ó³/ëR¢8k{T uR:dæå{.fNÒÜÉ])ÆÂÍÙu,▯FFÙqô]=ò¹Þ
                                                                GH}³p½ûi2(KzÞÂZçÚ]ei)Íxd[guillaume-alain.priso-totto@r05p03 Alice]$ 1;2c
```

Ensuite, on a fait comme précédemment, Bob a vérifié l'authenticité de la clé publique d'Alice à l'aide de la signature qu'elle a fait avec son fichier et de sa cléf privé.

```
[guillaume-alain.priso-totto@r05p03 Bob]$  openssl dgst -sha256 -verify AlicePublicKey -signature AliceSignature AuthDataAlice
Verified OK
[guillaume-alain.priso-totto@r05p03 Bob]$
```

4)
On a généré une SymKey puis Bob l'a crypté avec la clé publique d'alice

```
[guillaume-alain.priso-totto@r05p03 Bob]$ openssl rand -hex-out SymKey 64
unknown option '-hex-out'
usage: rand [-base64 | -hex] [-out file] num
 -base64           Perform base64 encoding on output
 -hex              Hexadecimal output
 -out file         Write to the given file instead of standard output
[guillaume-alain.priso-totto@r05p03 Bob]$ openssl rand -hex -out SymKey 64
[guillaume-alain.priso-totto@r05p03 Bob]$ ls
AlicePublicKey  AliceSignature  AuthDataAlice  AuthDataBob  BobDocument  BobPrivateKey  BobPublicKey  BobSignature  SymKey
[guillaume-alain.priso-totto@r05p03 Bob]$ openssl rsautl -encrypt -in SymKey -pubin -inkey AlicePublicKey -out SymKeyEncrypted
[guillaume-alain.priso-totto@r05p03 Bob]$ ls
AlicePublicKey  AliceSignature  AuthDataAlice  AuthDataBob  BobDocument  BobPrivateKey  BobPublicKey  BobSignature  SymKey  SymKeyEncrypted
[guillaume-alain.priso-totto@r05p03 Bob]$ cat SymKeyEncrypted
Ý>Qç▯BłöPeñxé5ç¯²\
øž4,k³±}M¯!¶cWc·Á?Z[Î@¬˛îu
                        ?ixzāpG¡v±ý+w/˛´▯8iÍëß:èIã5)PwãÚ$LõÎ~^Ù°éÁkV²¦ìµ'r²ìua¬-R×G½«*¿ÍKj½§4Á
o!
ëbx-ffε'(Ī#=NñUŻs5®Å6UİOÍ?`wWúÔÍa}ÒÍy▯cç@>,åu6OCÔ◡)`a§
```

Alice a reçue SymKey crypté et envoyé par Bob et maintenant elle peut la décrypter avec sa cléf privé car SymKey à été crypté par sa cléf publique. Ils ont réussi à se partager une clé symétrique de manière sécurisé .

```
[guillaume-alain.priso-totto@r05p03 Alice]$ openssl rsautl -decrypt -in SymKeyEncrypted -inkey AlicePrivateKey -out SymKey
[guillaume-alain.priso-totto@r05p03 Alice]$ ls
AliceDoc  AlicePrivateKey  AlicePublicKey  AliceSignature  AuthDataAlice  AuthDataBob  BobPublicKey  BobSignature  SymKey  SymKeyEncrypted
[guillaume-alain.priso-totto@r05p03 Alice]$ cat SymKey
09ac2f839b6fd1891608c1ae5891edff1eca65f5c89228294f8290c937de2bb21af73b3c468dac542790f376a357e54b86e7237f748fafce12e98d7c38143f87
[guillaume-alain.priso-totto@r05p03 Alice]$
```

5)

ici on a fait comme pour Alice :

comme les 2 ont une cléf symétrique, alors Bob a crypté ses donnés à l'aide de sa cléf symétrique. Puis il l'a envoyé a Alice qui a décrypté les données crypté de Bob à l'aide de sa clé symétrique.

```
[guillaume-alain.priso-totto@r05p03 Alice]$ openssl enc -d -aes-128-cbc -salt -pbkdf2 -kfile SymKey -in DataAliceEncrypted -out DataAlice
[guillaume-alain.priso-totto@r05p03 Alice]$ openssl enc -d -aes-128-cbc -salt -pbkdf2 -kfile SymKey -in DataBobEncrypted -out DataBob
[guillaume-alain.priso-totto@r05p03 Alice]$ ls
AliceDoc  AlicePrivateKey  AlicePublicKey  AliceSignature  AuthDataAlice  AuthDataBob  BobPublicKey  BobSignature  DataAlice  DataAliceEncrypted  DataBob  DataBobEncrypted  SymKey  SymKeyEncrypted
[guillaume-alain.priso-totto@r05p03 Alice]$ cat DataBob
J,aime le MegaChevalier zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz
[guillaume-alain.priso-totto@r05p03 Alice]$
```

EXERCICE 2

1)

on a récupéré le certificat , puis avec les différente commande trouvé dans le guide, on a réussi à retrouver les informations suivantes :

```
guillaume-alain.priso-totto@r05p03 LAB2]$  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
guillaume-alain.priso-totto@r05p03 LAB2]$  openssl x509 -noout -in CertificateLCL -dates
notBefore=Dec 28 00:00:00 2020 GMT
notAfter=Dec 28 23:59:59 2021 GMT
guillaume-alain.priso-totto@r05p03 LAB2]$
```

```
[guillaume-alain.priso-totto@r05p03 LAB2]$  openssl x509 -noout -in CertificateLCL -serial
serial=AB6AD8276765FE51DC43493A4C7B6223
[guillaume-alain.priso-totto@r05p03 LAB2]$
```

```
guillaume-alain.priso-totto@r05p03 LAB2]$  openssl x509 -noout -in CertificateLCL -fingerprint
HA256 Fingerprint=DB:C0:36:51:5A:39:42:93:62:59:6E:7F:7A:C6:CE:B2:2A:E7:34:6A:13:7E:72:A8:12:EF:47:4E:6E:64:8B:8F
guillaume-alain.priso-totto@r05p03 LAB2]$
```

```
[guillaume-alain.priso-totto@r05p03 LAB2]$  openssl x509 -noout -in CertificateLCL -pubkey
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwWquvcQxcMCjfwYsgPbg
Zd4RVMhwtow3ONvViwyo7jO4CNebZT8esC5pP1GFtqp2opZMr6Wczi+yhNJ2AfCB
UKC23bAGiAzoDG9Z6HtOqbCiqb1Je9GmW03ZuygF6RxZHl5oi2nsQ5lSZ8Oas+NG
l/3aE6khRTQfR8mV8cL3iAfgVlfsOD6rx0SZNWz4WwQgAermd1cTRXwcr0VHjhbK
8yHmlLbLy7PUbya/kTiAsWh+eHTGvE1lbp9orHuxT+W6crRcyLhdmG+L+5NgAJsx
qJgFcjrjqAzrm5A/IBXIdUhCQRG55wTI/Y73ZMIrn1ujzHBT6YNP6RsB1BifDSYh
OQIDAQAB
-----END PUBLIC KEY-----
[guillaume-alain.priso-totto@r05p03 LAB2]$
```

ensuite , comme dans le LAB1 , on a récupéré la cléf publique du serveur :

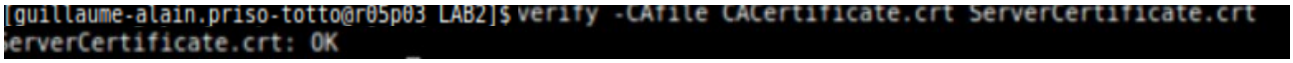
```
[guillaume-alain.priso-totto@r05p03 LAB2]$ openssl rsa -in ServerKeyPair -pubout -out ServerPublicKey
writing RSA key
[guillaume-alain.priso-totto@r05p03 LAB2]$ ls
 Alice   Bob   CertificateLCL  'Screen LAB2'   ServerKeyPair   ServerPublicKey
[guillaume-alain.priso-totto@r05p03 LAB2]$ cat ServerPublicKey
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAohiLZxF2b+etvOmoDHWv
1wCavs/MOBf6BaYZpb720IQ527mU/CdgVS9LKBkx9HR9U7DyMBA6WZJj/bYznItJ
Kdl6FWn6+vQcdHCNZqYgK4El0zvnIrNNRXWdswlA0OvXOIQgtYRjCKQ2scdu+mhS
79PEf/W5DzypXNbi87qLfWToCmRbwsgf1rUNY56EyuUsEvDDs35ZRkXE8LeUVRsA
TTIz/57++R72lijagOithzgfCFIKbKrQf1fsLH/g4vSV9XrhqcJX/zwkkbnEauC5
gGIXuEikgN6NGRJwaBUxlNwCIdAwfeMdq6mYR6JNIDn/ESQWn+ldYrwl/YgMeD38
FwIDAQAB
-----END PUBLIC KEY-----
[guillaume-alain.priso-totto@r05p03 LAB2]$
```

Puis, on a généré un certificat puis on l’a rempli comme demandé:

m

```
[guillaume-alain.priso-totto@r05p03 LAB2]$ openssl req -new -key CAKeyPair -out CACertificate.crt
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) []:MyCA
Organizational Unit Name (eg, section) []:MyCA
Common Name (eg, fully qualified host name) []:MyCA
Email Address []:CAemail

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
[guillaume-alain.priso-totto@r05p03 LAB2]$ openssl verify
```



Ici on a bien vérifié l’Authenticité du CAcertificat grace au ServeurCertificate .