OPENSSL 4

ETAPE 1 : COMMENÇONS PAR FAIRE LE TOUR DES ECC AVEC OPENSSL :

(a) En utilisant l'argument ecparam, lister toutes les courbes elliptiques supportées par la biblioth`eque openssl : "openssl ecparam -list_curves"

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl ecparam -list_curves
  secp112r1 : SECG/WTLS curve over a 112 bit prime field
  secp112r2 : SECG curve over a 112 bit prime field
  secp128r1 : SECG curve over a 128 bit prime field
  secp128r2 : SECG curve over a 128 bit prime field
  secp160k1 : SECG curve over a 160 bit prime field
  secp160r1 : SECG curve over a 160 bit prime field
```

(b) Générer votre paire de clefs key.pem en utilisant la courbe "prime256v1" : "openssl ecparam -genkey -name prime256v1 -out key.pem"

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl ecparam -genkey -name
prime256v1 -out key.pem
nathan@nathan-VirtualBox:~/Bureau/TP4$ ls
key.pem
```

(c) Protéger cette paire de clefs par un chiffrement symétrique DES3 : "openssl ec -in key.pem -des3 -out keyout.pem"

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl ec -in key.pem -des3 -
out keyout.pem
read EC key
writing EC key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
nathan@nathan-VirtualBox:~/Bureau/TP4$ ls
keyout.pem  key.pem
```

Pwd : EC_key

(d) Afficher votre paire de clefs : "openssl ec -in keyout.pem -text -noout"

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl ec -in keyout.pem -tex
t -noout
read EC key
Enter PEM pass phrase:
Private-Key: (256 bit)
priv:
    a4:35:9d:a3:d2:dd:9c:d3:cd:60:7d:d1:18:7a:be:
    fa:c6:c0:c0:ed:da:7e:f7:f5:b7:b0:85:c3:3c:d5:
    7e:8b
pub:
    04:ac:73:70:db:85:e4:07:b7:25:86:11:fc:67:1e:
    b1:f7:cd:04:78:44:59:66:d9:92:0c:cd:36:87:d0:
    8a:7d:8f:0c:8f:68:de:0c:11:7a:00:cd:1a:b2:d9:
    48:21:30:59:17:46:61:4f:28:fc:ac:cf:66:aa:45:
    e8:d2:37:40:67
ASN1 OID: prime256v1
NIST CURVE: P-256
nathan@nathan-VirtualBox:~/Bureau/TP4$
```

(e) Extraire la clef publique de cette paire de clefs : ”openssl ec -in keyout.pem - pubout -out pub”

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl ec -in keyout.pem -pub
out -out pub
read EC key
Enter PEM pass phrase:
writing EC key
nathan@nathan-VirtualBox:~/Bureau/TP4$ ls
keyout.pem  key.pem  pub
```

(f) Générer votre proper demande de certificate x509:”openssl req -new -key keyout.pem -out csr_pub.pem”

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl req -new -key keyout.p
em -out csr_pub.pem
Enter pass phrase for keyout.pem:
You are about to be asked to enter information that will be incorpora
ted
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) [AU]:TN
State or Province Name (full name) [Some-State]:Tunis
Locality Name (eg, city) []:Mutuelville
Organization Name (eg, company) [Internet Widgits Pty Ltd]:EC
Organizational Unit Name (eg, section) []:EC
Common Name (e.g. server FQDN or YOUR name) []:EC
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:EC_cert
An optional company name []:
nathan@nathan-VirtualBox:~/Bureau/TP4$
```

(g) Générer votre certificat X509 : ”openssl req -x509 -days 365 -key keyout.pem -in csr_pub.pem -out certificate.pem”

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl req -x509 -days 365 -k
ey keyout.pem -in csr_pub.pem -out certificate.pem
Enter pass phrase for keyout.pem:
nathan@nathan-VirtualBox:~/Bureau/TP4$ ls
certificate.pem  csr_pub.pem  keyout.pem  key.pem  pub
nathan@nathan-VirtualBox:~/Bureau/TP4$
```

(h) Afficher votre certificat : ”openssl x509 -in certificate.pem -text -noout”

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl x509 -in certificate.p
em -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            14:c6:8a:e0:03:62:0a:66:5e:02:bd:a4:b2:a3:23:a4:9c:81:ed:
```

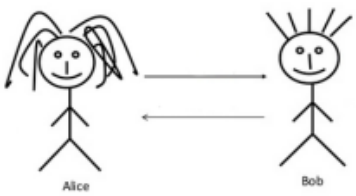
(i) Hacher et signer un fichier de votre choix en utilisant ecdsa avec sha1 :"openssl dgst -ecdsa-with-SHA1 -out mess_signe -sign keyout.pem test"

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl dgst -sha1 -sign keyout.pem test2 >
 mess_sign
Enter pass phrase for keyout.pem:
```

(j) Vérifier votre message signé : "openssl dgst -ecdsa-with-SHA1 -verify pub - signature mess_signe test"

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl dgst -sha1 -verify pub -signature m
ess_sign test2
Verified OK
```

ETAPE 2 : COMMUNICATION ENTRE ALICE ET BOB A` TRAVERS UNE CLEF COMMUNE :

i. Génération des clefs pour Alice. ii. Extraction de la clef publique de Alice.		i. Génération de clefs pour Bob. ii. Extraction de la clef publique de Bob.

(a) Génération des clefs pour alice :" openssl ecparam -name secp256k1 -genkey - noout -out alice_priv_key.pem"

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl ecparam -name secp256k1 -genkey -no
out -out alice_priv_key.pem
```

(b) Extraction de la clef publique de Alice : "openssl ec -in alice_priv_key.pem -pubout -out alice_pub_key.pem"

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl ec -in alice_priv_key.pem -pubout -
out alice_pub_key.pem
```

(c) Génération des clefs pour Bob : "openssl ecparam -name secp256k1 -genkey - noout -out bob_priv_key.pem"

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl ecparam -name secp256k1 -genkey -no
out -out bob_priv_key.pem
```

(d) Extraction de la clef publique de Bob : ”openssl ec -in bob_priv_key.pem - pubout -out bob_pub_key.pem”

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl ec -in bob_priv_key.pem -pubout -ou
t bob_pub_key.pem
read EC key
writing EC key
```

3. ETAPE 3 : LE PARTAGE D’UNE CLEF COMMUNE :

(a) Alice lance cette commande pour obtenir une clef commune :”openssl pkeyutl -derive -inkey alice_priv_key.pem -peerkey bob_pub_key.pem -out alice_shared_secret.bin”

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl pkeyutl -derive -inkey alice_priv_k
ey.pem -peerkey bob_pub_key.pem -out alice_shared_secret.bin
```

(b) Bob lance cette commande pour obtenir une clef commune :”openssl pkeyutl -derive -inkey bob_priv_key.pem -peerkey alice_pub_key.pem -out bob_shared_secret.bin”

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl pkeyutl -derive -inkey bob_priv_key
.pem -peerkey alice_pub_key.pem -out bob_shared_secret.bin
```

(c) Pour vérifier la véracité de la clef commune, nous allons la convertir en Base64 dans les deux machines respectives de Alice et Bob :”base64 alice_shared_secret.bin” et ”base64 bob_shared_secret.bin”

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ base64 alice_shared_secret.bin
6o9hxJ0RDeRQ7lnUEMl+6Oj0/X2Q6ykxxVdXy9u9Fvw=
nathan@nathan-VirtualBox:~/Bureau/TP4$ base64 bob_shared_secret.bin
6o9hxJ0RDeRQ7lnUEMl+6Oj0/X2Q6ykxxVdXy9u9Fvw=
nathan@nathan-VirtualBox:~/Bureau/TP4$ diff bob_shared_secret.bin alice_shared_sec
ret.bin
nathan@nathan-VirtualBox:~/Bureau/TP4$
```

ETAPE 4 : CHIFFREMENT ET DECHIFFREMENT D’UN FICHIER :

(a) Chiffrement d’un fichier Test :”openssl enc -aes256 -k $(base64 alice_shared_secret.bin) -e -in test - out cipher.txt”

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ touch test
nathan@nathan-VirtualBox:~/Bureau/TP4$ echo "yo les amis" > test
```

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl enc -aes256 -k $(base64 alice_share
d_secret.bin) -e -in test -out cipher.txt
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

(b) Déchiffrement d’un fichier cipher.txt :”openssl enc -aes256 -k $(base64 alice_shared_secret.bin) -d -in cipher.txt -out test1”

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ openssl enc -aes256 -k $(base64 alice_share
d_secret.bin) -d -in cipher.txt -out test1
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

(c) Comparer les deux fichiers test1 et test avec la commande diff.

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ diff test test1
nathan@nathan-VirtualBox:~/Bureau/TP4$
```

```
nathan@nathan-VirtualBox:~/Bureau/TP4$ cat test test1
yo les amis
yo les amis
```

Aucune différence