

## Travaux pratiques 1 : Crypto

Abdel-Malik FOFANA  
Ivan KRIVOKUCA

### Exercice 1:

- > chiffrement : `openssl enc -bf-cbc -in toto.txt -out toto.enc -k master1`
- > déchiffrement : `openssl enc -bf-cbc -d -in toto.enc -out toto.dec -k master1`

le fichier affiche ceci:

```
(maliki@Maliki-club)-[~/Cryptography/tp/TP4_cryptographie_moderne_openssl_annexe/OpensSL]
$ cat toto.enc
Salted__P*,00+TW*-Y *+}#h4(B!*+RG_#*+n@*+L*+ZG\*
{+4;+Ar++++8c+$*Z*+;+;+ji+U_+9*ü+<+
av*+1{+dM*+[C++++0+S*.S7+++F*+{+Y=+++++M*%amNI*,qo
+ji)*@7~oH*+tkRF*o
+Fi@:++P+ef+we_AI(s:RqbK>)pMg<+n>+U5+B+++9+B*+X!"+D+++.*+9+:*,+e++++!B*-G*+b4yy'\++S*+Z*+vK*^M*%46*+`]f+++B*+y*+*+51mSsP*+Yg*+*+{+ +2?+*+
+++++2++++<+A*|L+4+
Q*+"WH*+S{+[-++;FF+y+2*+<+r+u*+t+++++n+++++*+X*+G*+p*+*/=4
(maliki@Maliki-club)-[~/Cryptography/tp/TP4_cryptographie_moderne_openssl_annexe/OpensSL]
$
```

Le codage est : Blowfish en mode CBC ducoup hexadécimal (Cipher Block Chaining car on a mis "enc -bf-cbc" en option dans la commande)

- >chiffrement en base 64 : `openssl enc -bf-cbc -in toto.txt -out toto.b64 -k master1 -base64`
- Le codage est : base64

-> Voici ce qui se passe quand on déchiffre avec le mauvais et bon mot de passe

```
(maliki@Maliki-club)-[~/Cryptography/tp/TP4_cryptographie_moderne_openssl_annexe/OpensSL]
$ openssl enc -bf-cbc -d -in toto.enc -out toto.dec -k master2
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
bad decrypt
4067D4498B7F0000:error:1C800064:Provider routines:ossl_cipher_unpadblock:bad decrypt:../providers/implementations/ciphers/ciphercommon_block.c:124:
(maliki@Maliki-club)-[~/Cryptography/tp/TP4_cryptographie_moderne_openssl_annexe/OpensSL]
$ openssl enc -bf-cbc -d -in toto.enc -out toto.dec -k master1
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Exercice 2 :
Visualiser mot de passe base64:
(maliki@Maliki-club)-[~/Cryptography/tp/TP4_cryptographie_moderne_openssl_annexe/OpensSL] word.b64
$
```

quand le mot de passe est mauvais, le terminal nous dit que le déchiffrement est mauvais.

### Exercice 2 :

Visualiser mot de passe base64 et mettre le mot de passe en clair dans password.bin:

```
openssl enc -base64 -d -in password.b64 -out password.bin
```

Déchiffrer à encrypted\_file.b64 :

```
openssl enc -d -aes-256-cbc -in encrypted_file.b64 -out decrypted_file
-kfile password.bin
```

le mot de passe est : denial of service

cependant le mot de passe ne fonctionne pas voici la preuve :

```
(maliki@Maliki-club)-[~/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe/Openssl]
$ openssl enc -base64 -d -in password.b64 -out password.bin

(maliki@Maliki-club)-[~/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe/Openssl]
$ openssl enc -d -aes-256-cbc -in encrypted_file.b64 -out decrypted_file -kfile password.bin
bad magic number

(maliki@Maliki-club)-[~/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe/Openssl]
$ cat password.bin
denial of service

(maliki@Maliki-club)-[~/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe/Openssl]
$ openssl enc -aes-256-cbc -d -in encrypted_file.b64 -out decrypted_file -k 'denial of service'
bad magic number

(maliki@Maliki-club)-[~/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe/Openssl]
$
```

j'ai donc refait l'exercice avec un fichier bien crypter par mes soins avec le mot de passe "carottes" et cela marche :

```
(maliki@Maliki-club)-[~/cyber m1/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe (1)]
$ openssl enc -base64 -in password.txt -out password.b64

(maliki@Maliki-club)-[~/cyber m1/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe (1)]
$ openssl enc -base64 -d -in password.b64 -out password.bin

(maliki@Maliki-club)-[~/cyber m1/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe (1)]
$ cat password.bin
carottes

(maliki@Maliki-club)-[~/cyber m1/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe (1)]
$ openssl enc -aes-256-cbc -in test.txt -out encrypted_file.b64 -k carottes
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

(maliki@Maliki-club)-[~/cyber m1/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe (1)]
$ openssl enc -d -aes-256-cbc -in encrypted_file.b64 -out decrypted_file -kfile password.bin

*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

(maliki@Maliki-club)-[~/cyber m1/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe (1)]
$ cat decrypted_file
message chiffre avec aes en mode cbc avec clé de 256bit

(maliki@Maliki-club)-[~/cyber m1/Cryptographie/tp/TP4_cryptographie_moderne_openssl_annexe (1)]
$
```

Exercice 3 :

```
1. openssl genrsa -out rsaprivatekey.pem 1024
```

1. `openssl rsa -in rsaprivatekey.pem -text -noout` (pour avoir et ensuite visualiser le contenu du fichier avec `openssl rsa`.)

```
2. openssl rsa -in rsaprivatekey.pem -des3 -out  
rsaprivatekey.pem.des3 -passout pass:securite → protège la  
clé avec le mdp securité
```

Clé normale:

```
-----BEGIN PRIVATE KEY-----  
MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAMhKRTUAXOf04hR/  
bxSn1EsKWuYhVldedFQzy5IN3BU8x1LB3dWL4Fih1oIQT19nJuwQ1OuyCxbXaV72  
M0vHj+8itfb8vfoeJgEy+dekM0ZesDnDjoNabMuv7IsK30AvTFEhEHbuYvU7z6UH  
jOQZVF7fT0RQBwb57vJOe400OFz5AgMBAAECgYEAs8lAvajavnyurDoOLRvQz6C3  
I1qQGRS5F5H70NzSUtpiEV/qVjcgqlB2sgIUTw/Z+BeWqNRjWY1dkZsxR3kaMrt+  
MHGgI5UNZp9LSNqK1D2Qqgcnv7CWOFwnW6ZqBFreVusVrjJ53xKSaj3nibi3nPzd  
FUp3rVtqiEHetbXeLjUCQQDlZMjV0DQs9WMh1vXDJu0fUy7QfSSs6oeqjExEI3Zz  
XuWgMb3z1/8HD5pfsFAgSkafvsUEOAEuLvxErs2u4y5HAKEA34VV0b7HPO2/gymX  
Zg5IE1libMQXKrPGOBZy2M/uqhdDkzGJbWv5REEi+K+lWcKIPLnWMigGckobBh+7B  
RH16vwJAFTRNxEXsrMM6GKSvLwvoG18rEgaeV0UmqUMyWZGtn1iETVYlvICY2GQ+  
1t2LapCzGo0d1RgAM+6v7vfEh4gEfQJAJPlwdc7CNID7kh3aLoakQ5b/rP9RRIR8  
ws7mkPjLf3FLt8d5Zzb0f+YY1zj1FJKuj0Hgqp+Tn+VstCi5jbPYDwJABgRgT1YY  
CnvaabQWYw4TT+hWsoCXdYihv1LT2lh05718W7BQZrmyOgo63jSi6zZ0TB1EjGEh  
PXg8Nq+pYpmygw==  
-----END PRIVATE KEY-----
```

cat affiche :

Clé 3DES

```
-----BEGIN ENCRYPTED PRIVATE KEY-----  
MIIC1DBOBgkqhkiG9w0BBQ0wQTApBgkqhkiG9w0BBQwwHAQI/g+h9p0wZg0CAggA  
MAwGCCqGSIB3DQIJBQAwFAYIKoZIhvcNAwcECC12oXFh4Sv/BIICgOVd6rgMpd/k  
K2OkMjk4VoeHyOZbfXhr5xzEZ9lKmylwjeEvYXKJZ2BwIYgDLsAeVRNMxD7VDTSG  
XK8zFo8sOpHoCJ2XQ3Chn70wdrGf1Ctel/2ngz30X9t1taLDWOUQaM1ZjomEvyeG  
0YryHC1D3Zhp8ntZNgCnz//3+/Nk93A605ZQZPH+Bjn2OMHR7jFQzGdQwtCS+Cu5  
WXCKtJdk9/ewQmsQIRNk4usWXLzFlPkC3FCeVrQPgRXQG4JPhVQ/ZY8xWLMCMZJH  
k3z0gmmZzLw3GTE8bQcBwkhVojgs5PnLJ10eAgdKJWxyjNC4t9iEuZUCMFlfa7r  
gIswAgnj7ATE/OntK799deexTtdThOzXgSDtKDZlYIoTm6H9+JWccJ3E09pTv9Ka  
bPMNHKi3LOsY3GWg0o3tK6tUxS489pvlTaB2zSHBkcjr7dzV013ybSJdOWmsvxcw  
0In2MTLFWQu+hCU2do5dxBR0u46TraDneJdIIeEqTBRxWa6Hk2mMMiBP7c1GoyEf  
FNMo5iFyPQJ/dHQFl2G2ziMIkdb1RQxudZ5KtdiGBnObt+CJGcbq82NuH6LUtSVp  
s4VcnFSYCFjPihpSREiaxD6kk8K088jhZdv1pJR5PwmMX1Hg+Th1UhON1WlRHq/5  
mX0vGZY75LUNU/tNSrbrWX6r33r190gA6KqPfrqupKvquG6ntN2LsK6weWExba0R  
r9XdWy7/bcPlxXWLZecawBqY/Lw+4xGZxzgx+/atV+YON2WZNavwNv0XUtw4KKrw  
GOxt94DIM8yC1fkvBHGQq9kjVKo+fHoauQqsc8tflhDrQeikXgc0fAk2nbDBv582  
gmdIpjh6u4I=  
-----END ENCRYPTED PRIVATE KEY-----
```

Pour regarder les caractéristiques de la clé, on nous demande le mot de passe de la clé

```
~/Bureau/M1/UPC/crypto/OpenSSL > openssl rsa -in rsaprivatekey.pem.des3 -text -noout  
Enter pass phrase for rsaprivatekey.pem.des3:
```

(si le mdp rentrée est bon alors il affiche les caractéristiques de la clé, sinon il affiche une erreur)

3. `openssl rsa -in rsaprivatekey.pem -pubout -out rsapublickey.pem`  
`openssl rsa -pubin -in rsapublickey.pem -text -noout`
4. `echo "Ceci est un message classé confidentiel \!" > message.txt`
5. `openssl pkeyutl -encrypt -in message.txt -out message.enc -pubin -inkey rsapublickey.pem`

Exercice 4:

- 1.