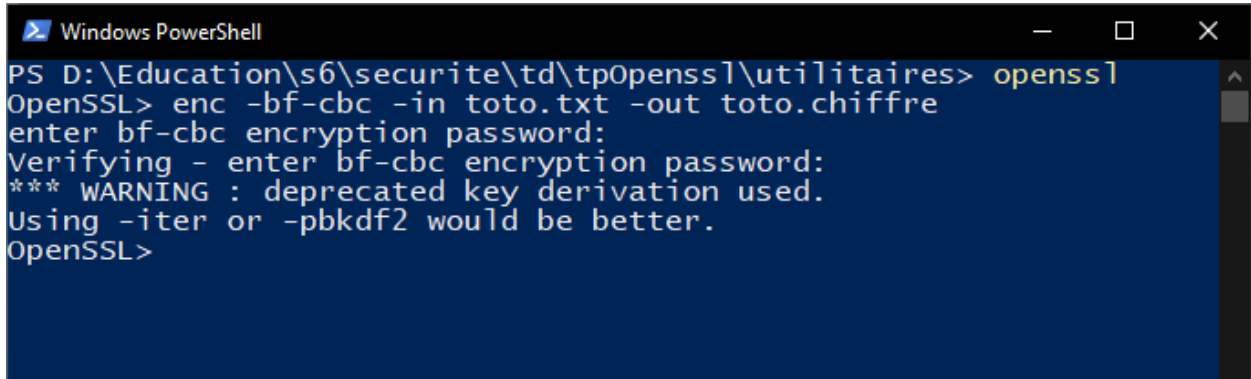


Exercise 1:

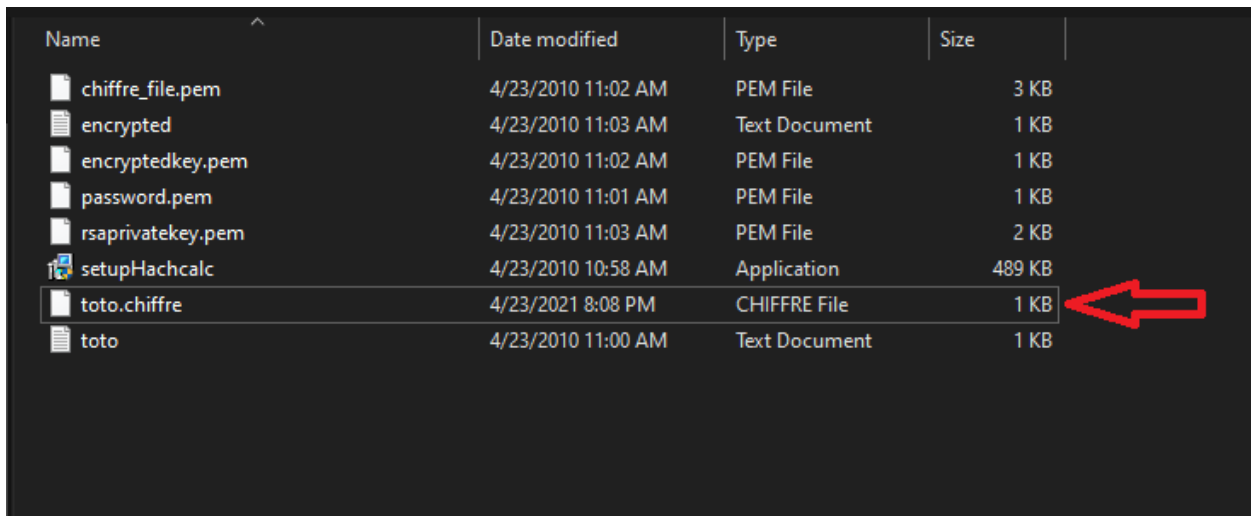
- Chiffrer toto.txt avec blowfish cbc en utilise la commande:

enc -bf-cbc -in toto.txt -out toto.chiffre



```
Windows PowerShell
PS D:\Education\s6\securite\td\tp0penssl\utilitaires> openssl
OpenSSL> enc -bf-cbc -in toto.txt -out toto.chiffre
enter bf-cbc encryption password:
Verifying - enter bf-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
OpenSSL>
```

Figure 1



Name	Date modified	Type	Size
chiffre_file.pem	4/23/2010 11:02 AM	PEM File	3 KB
encrypted	4/23/2010 11:03 AM	Text Document	1 KB
encryptedkey.pem	4/23/2010 11:02 AM	PEM File	1 KB
password.pem	4/23/2010 11:01 AM	PEM File	1 KB
rsaprivatekey.pem	4/23/2010 11:03 AM	PEM File	2 KB
setupHachcalc	4/23/2010 10:58 AM	Application	489 KB
toto.chiffre	4/23/2021 8:08 PM	CHIFFRE File	1 KB
toto	4/23/2010 11:00 AM	Text Document	1 KB

Figure 2

- Visualisation de contenu du fichier toto.chiffre

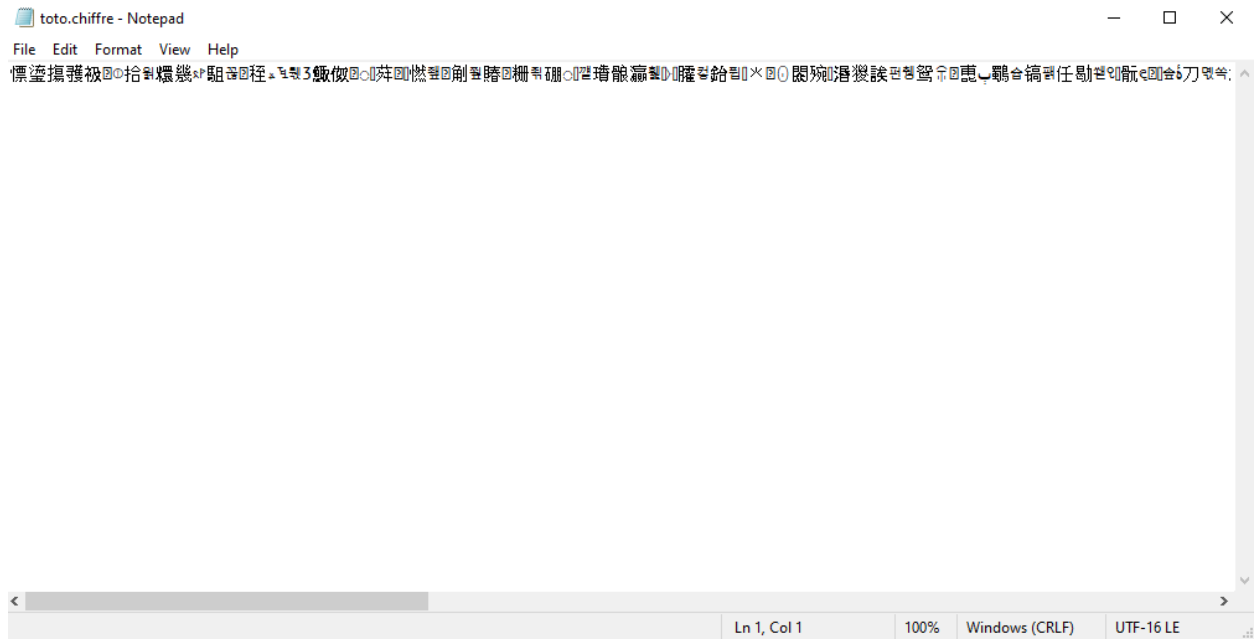


Figure 3

- Ce codage est : UTF-16 LE

Chiffrer toto.txt avec blowfish cbc et base64 en utilisant la commande:
`enc -bf-cbc -in toto.txt -base64 -out toto.pem`

```

Windows PowerShell
PS D:\Education\s6\securite\td\tp0penss1\utilitaires> openssl
OpenSSL> enc -bf-cbc -in toto.txt -base64 -out toto.pem
enter bf-cbc encryption password:
Verifying - enter bf-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
OpenSSL>

```

Figure 4

Name	Date modified	Type	Size	
chiffre_file.pem	4/23/2010 11:02 AM	PEM File	3 KB	
encrypted.txt	4/23/2010 11:03 AM	Text Document	1 KB	
encryptedkey.pem	4/23/2010 11:02 AM	PEM File	1 KB	
password.pem	4/23/2010 11:01 AM	PEM File	1 KB	
rsaprivatekey.pem	4/23/2010 11:03 AM	PEM File	2 KB	
setupHachcalc.exe	4/23/2010 10:58 AM	Application	489 KB	
toto.chiffre	4/23/2021 8:08 PM	CHIFFRE File	1 KB	
toto.pem	4/23/2021 10:23 PM	PEM File	1 KB	→
toto.txt	4/23/2010 11:00 AM	Text Document	1 KB	

Figure 5

- Visualisation de contenu du fichier toto.pem

toto.pem - Notepad

File Edit Format View Help

```
J2FsdGVkX18sQM/QPS1gBdntaEtEv9+bTVi1UJRUwrhVajvIfuWPvDRb9yQT4oFU
sctYE12sSVQnA8f5YhBeAtKhWEQrdcEBJnWjFdhd6CxnR2ug2vzB3F1oCtUXv7jG
WEs7au1gbPNQj2zItL/xRQdsyxqduxPqxb6aSJH61c54mykMwJfUpSMvvCN4YmsU
wn1u4aJ1kKLoBPMYGxradVEvAzF3kF9z7G3twNmiZg0XbnUmndz1e1zY5E75f/3
VrYS01NUCohK9HGUTIDVxHNozSsUUos+RtdLT33k8yJ/V71nXWq9kcp1Q/6fNo4/
/5Vyp/tDgMvbaSjiMN0LAnD/L6hZxj/r5Fn2Ugb5Na/I3t76RuFHBWlvYVurfDj
L+dKYUR0NIpbS0X09WTtzzYwLR3io/WJZkhf5ku1NqqcQ9B5G/EdOL/ddb1x+FLU
bI6N8AGng6uaJ5nAGaIdrdXpzun/Jn/68HrDBKHk5S9k8x1byb+mw04xDVsB9uEh
T20gtuBootG0usVq7enmSD0s3fJmyL1mUY0TUcNAKc2YkbbYFGktOKudnY64MZM4
xhuSv51Txpud5sztN0AK0uOHddnt705myA7RWnz5s1q3/CQv5J/VFxsKggPVMXe
vQVfyrkV0WMNQ4N6w/cuVsYjPb1asxHrJyEaYdT8A1M=
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

Figure 6

- Ce codage est : UTF-8

- Déchiffrement de fichier toto.chiffre to totoDechif.txt

```

Windows PowerShell
PS D:\Education\s6\securite\td\tpOpenssl\utilitaires> openssl
OpenSSL> enc -d -bf-cbc -in toto.chiffre -out totoDechif.txt
enter bf-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
OpenSSL>

```

Figure 7

- Après le déchiffrement en Remarque le même contenu :

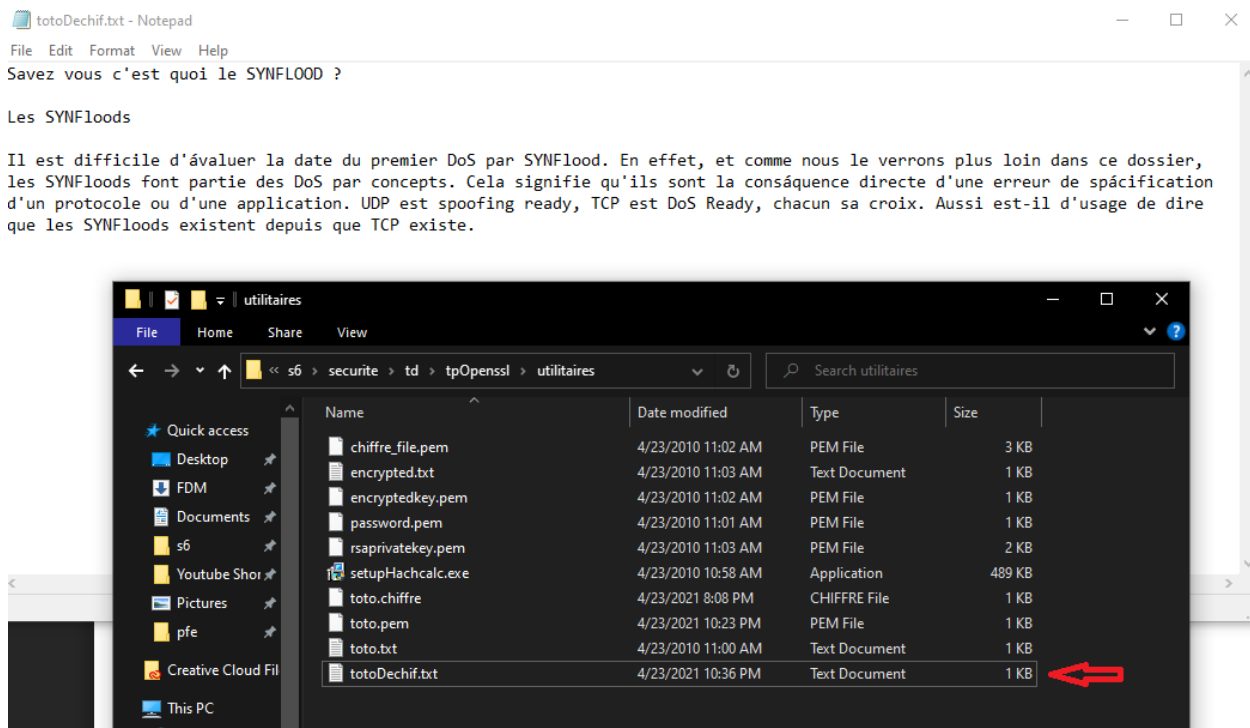


Figure 8

- La Tentative de déchiffrer le fichier toto.chiffre avec un mauvais mot de passe :

```

OpenSSL> enc -d -bf-cbc -in toto.chiffre -out totoDechif2.txt
enter bf-cbc decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
bad decrypt
8008:error:06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt:crypto\evp\evp_enc.c:610:
error in enc
OpenSSL>

```

Figure 9

openssl réagit avec un message “bad decrypt” et output file n’est pas correct :

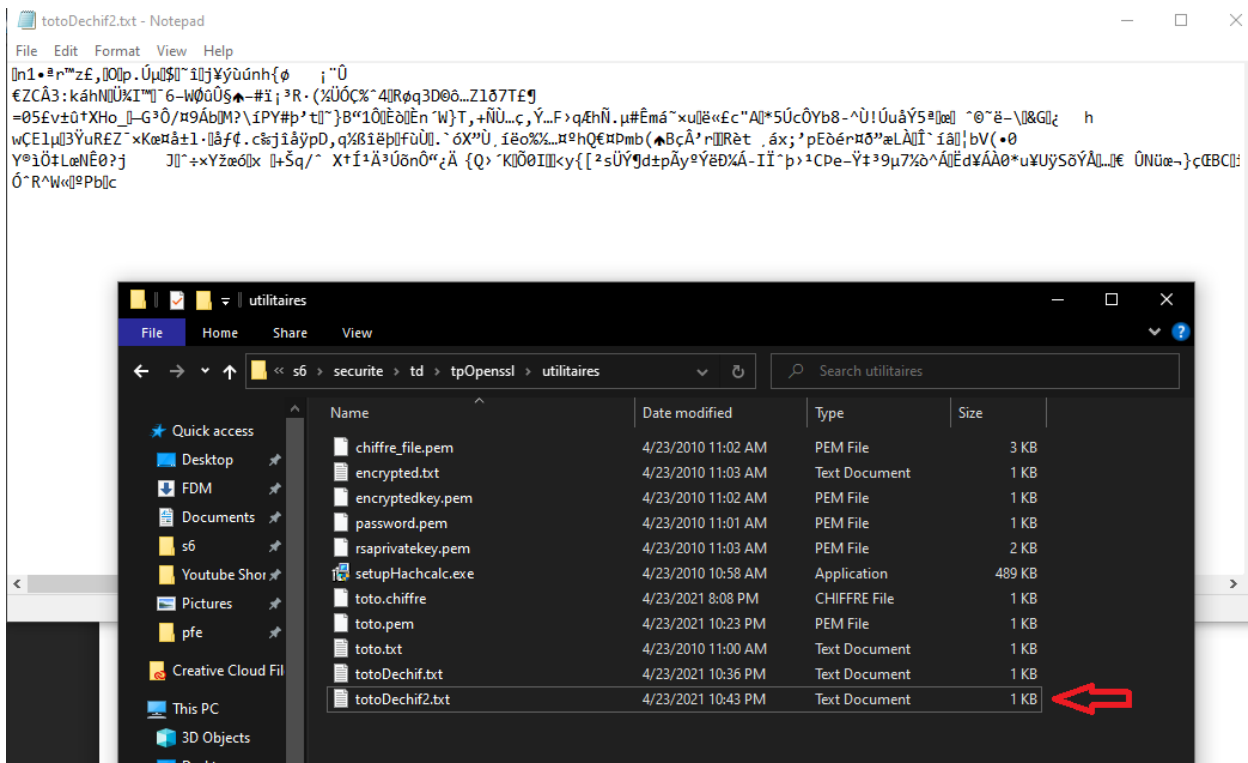


Figure 10

Exercice 02:

- Oui il est possible de décoder avec command suivant :
base64 -d -in password.pem -out password2.txt

Ou avec :

base64 -d -in password.pem

- La clé est : denial of service

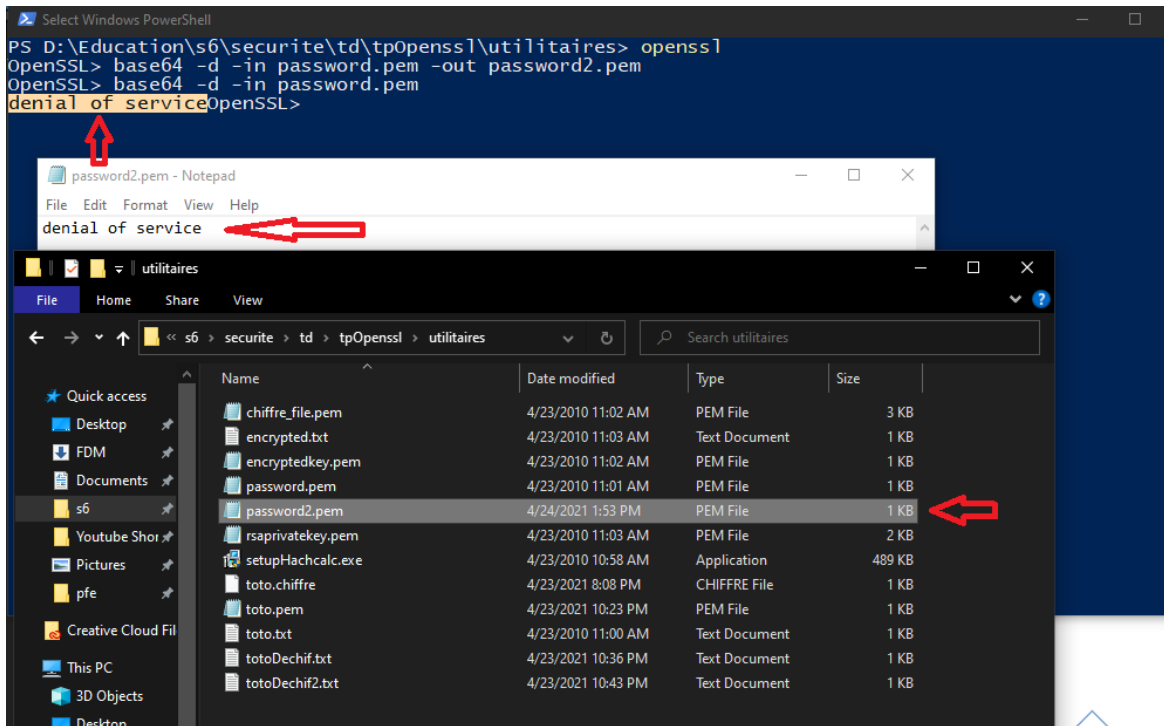


Figure 11

- Après décoder la clé j'utilise la commande suivant pour le déchiffrement de chiffre_file.pem :

enc -aes-256-cbc -salt -d -a -in chiffre_file.pem -out chiffre_fileD.txt -pass
file:password2.txt

Mais Ilya un problème dans le déchiffrement la clé n'est pas correct :

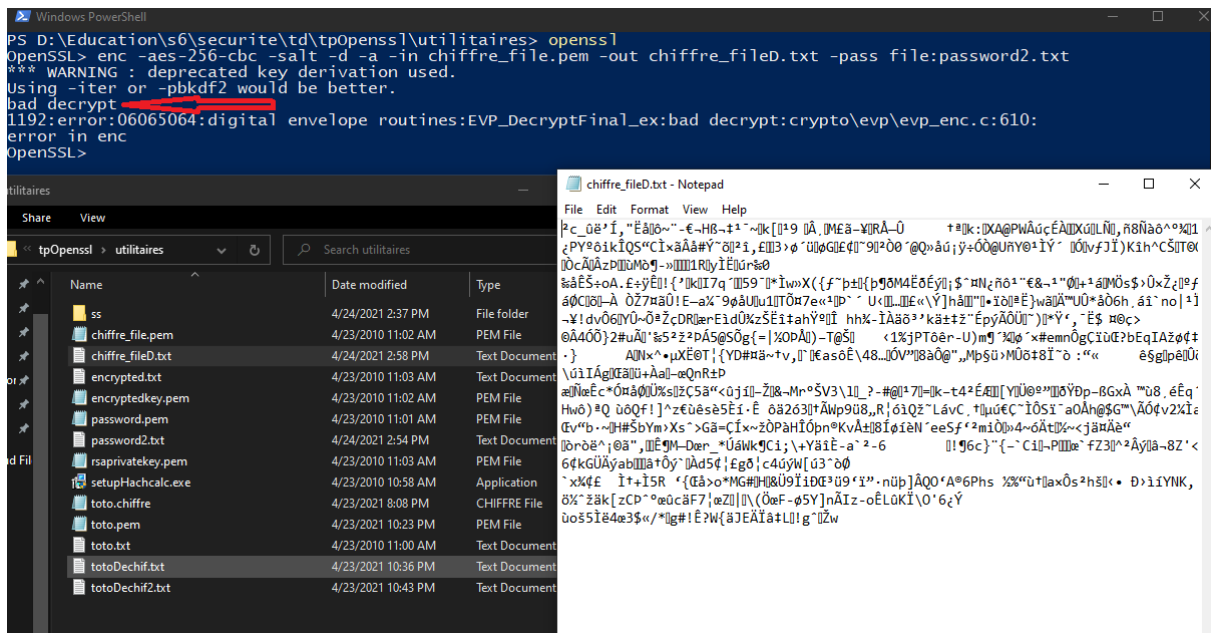


Figure 12

Exercice 03:

- Création de rsa clés avec la commande :

`genrsa -des3 -out myRsa.pem`

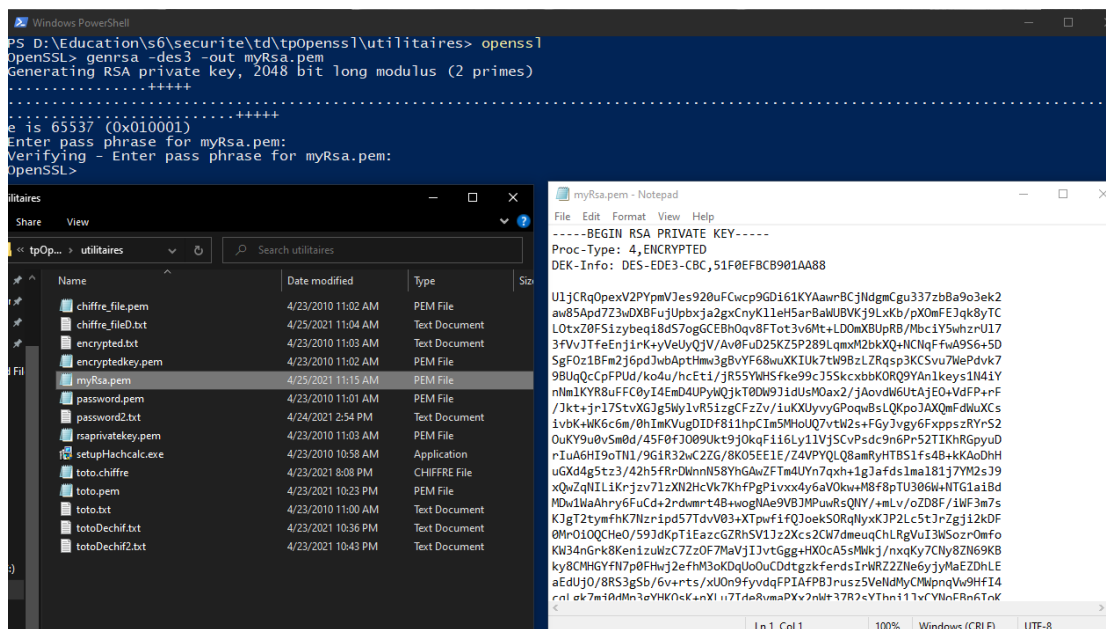


Figure 13

- Visualiser avec la commande : `rsa -in myRsa.pem -text -noout`

```

EnterSSL> rsa -in myRsa.pem -text -noout
OpenSSL RSA Private Key: (2048 bit, 2 primes)
modulus:
00:bc:80:92:f5:3a:0f:9c:e3:6c:14:4a:9a:ab:d7:
b6:a1:31:d1:8a:38:c5:97:09:59:62:8a:b9:d3:
62:12:12:17:9c:9d:b2:66:d7:0c:4:cf:04:92:a9:
13:fd:b9:ee:59:11:0d:c3:22:f6:ec:3f:16:94:07:
2a:fb:b7:81:a3:45:9a:49:b9:41:a6:d9:79:b2:71:
d8:b5:af:30:c2:88:9f:60:67:9e:4b:19:1c:ab:2e:
11:01:04:65:b2:42:33:8f:69:02:31:85:09:02:
82:76:b7:02:b2:f0:35:07:05:19:73:ee:e6:02:03:
2b:1d:76:72:37:1a:cb:2c:b8:1d:d6:ac:ff:54:43:
0e:65:10:0d:59:23:1e:e0:dd:33:80:a3:e2:a8:41:
71:0e:52:8f:9e:61:da:52:86:73:56:04:4b:aa:34:
ba:18:90:91:14:60:78:ae:45:09:54:de:1f:00:00:
35:cf:34:6a:72:02:2b:da:05:89:10:53:05:02:
69:f4:05:00:86:d1:70:ac:21:5e:b9:6a:b4:20:21:
76:6b:56:c9:6d:06:c5:45:03:06:c0:5d:59:08:
80:30:7d:e9:fe:1a:4e:7e:df:b0:5d:6f:cd:7b:1b:
b5:90:b7:43:10:ac:12:8f:1f:40:cd:43:55:e2:
9c:6f:
publicExponent: 65537 (0x10001)
privateExponent:
70:2f:a5:46:c5:53:5d:70:0c:76:d7:1f:ce:7e:07:
77:50:0b:01:54:18:3e:eb:10:e9:5d:b8:6a:77:df:
0c:a5:6f:e0:0a:0a:8f:22:8a:f9:b2:ab:6a:0d:ae:
5f:fb:6c:ae:58:9e:10:45:14:28:6b:9b:98:d0:
7e:4a:9b:80:eb:f7:51:26:34:b2:48:ad:de:d2:
be:ee:dd:2e:b5:cd:04:b7:26:82:53:63:b8:8d:2e:
70:f5:0e:03:eb:60:a0:b2:81:94:7a:70:71:ae:f9:
76:19:af:45:87:a8:47:91:45:60:b6:1e:5c:b6:be:
06:8e:0f:9f:b8:16:8b:7d:5f:dc:2b:7a:2d:3e:e1:
e1:59:86:53:fd:6e:2b:29:21:20:f7:02:
54:67:32:f3:2b:ec:67:e4:02:55:ae:27:03:26:b6:
bc:4e:80:82:6c:ad:9d:da:af:f2:66:69:cf:e9:0d:
c5:63:af:f2:baf0:b0:bc:6d:53:ec:21:fc:01:
6a:63:f5:b6:3c:d9:00:60:87:f6:77:da:8c:92:de:
b7:8d:0f:0e:cd:b5:7d:14:c9:04:e1:25:32:fd:13:
82:b6:9e:00:72:b6:88:6c:69:6d:12:eb:
47:b3:38:9a:81:e3:85:2d:35:22:08:c2:0f:81:a2:
c1
prime1:
00:fa:55:68:19:53:dc:73:69:b2:e2:79:f2:0b:1e:
2b:04:07:7b:73:96:b4:f5:b2:4b:55:c6:ed:f5:22:
71:10:e6:7c:2d:3d:ee:c7:2d:32:fb:80:96:96:cb:
ed:82:99:49:48:dd:31:3d:ee:07:73:77:98:a9:17:dc:
6c:69:4b:35:a5:05:a0:18:82:ac:68:0d:54:51:e5:
da:0f:1f:86:23:02:c0:16:42:86:35:15:99:57:
10:01:d1:36:af:20:2b:17:f4:46:8a:10:20:94:0:
30e:80:28:75:f7:64:14:e5:f7:96:b6:f4:9e:13:
e:21:3a:50:15:d7:d8:a6:99
prime2:
00:c4:c0:54:07:6b:64:37:26:f7:f9:eb:2d:f0:b2:
85:80:45:02:6a:b8:55:c8:37:14:11:79:28:21:3b:
31:4b:7a:a1:6d:ed:ab:65:4f:3b:ab:32:04:68:b1:
30:90:ad:bf:f8:b9:92:92:3:0a:da:4b:df:22:f5:fc:
62:6f:65:fa:8b:22:57:a9:31:75:9a:f4:43:6e:7f:
ba:76:31:1e:72:3b:51:1d:53:57:76:39:17:9a:ae:
51:ea:7c:d3:0b:07:34:3c:71:91:b6:c2:53:1c:3a:
94:d5:cc:ac:a6:ad:f7:5e:8b:33:9b:90:1f:bd:24:
df:9f:22:49:58:a6:3a:ff
exponent1:
00:ae:1f:53:ee:32:0f:a7:d3:80:92:d9:41:8b:8e:
2b:5d:7d:1a:d8:76:9d:f5:ee:5d:52:47:b1:62:8d:
d7:5f:af:a2:74:9b:7a:ec:65:b5:08:a3:56:3a:83:
9a:2b:fa:79:51:9e:3d:a9:16:ca:70:c0:2d:f5:a3:
c3:60:85:e8:c5:65:19:4b:93:30:fa:9b:37:14:ba:
2d:ae:a2:20:ac:ba:el:af:d7:a8:f8:2f:29:98:4e:
1f:05:24:49:41:b5:dd:49:15:56:47:82:33:3a:
36:02:23:42:81:73:57:90:5d:b8:9e:83:al:b8:86:
54:78:3c:8a:71:fe:ff:c0:19
exponent2:
0f:0a:0d:1d:10:12:51:76:fd:47:8c:0e:0f:ff:79:
0c:a8:48:48:bd:32:ec:18:ef:f6:e9:05:e9:79:cd:
c1:b3:62:93:6a:7a:2d:f3:3c:ce:49:2d:2e:95:
86:66:fa:35:6c:82:2e:1e:0b:65:47:c5:e5:2e:12:
8a:ef:4f:d9:b5:07:33:cb:09:04:ac:68:bc:fc:39:
16:e9:84:1f:93:63:27:c9:0d:f0:ca:49:7e:2b:93:
d2:80:84:92:92:60:47:9a:e8:ac:62:82:cb:6c:02:
2a:99:6a:44:c2:c3:cd:7b:5e:42:a3:47:45:6e:
30:3e:2e:ae:55:01:3b:0f
coefficient:
29:43:37:33:3f:6b:52:cf:c8:98:7b:21:47:1c:7b:
9c:68:08:0c:32:05:1d:b7:3a:8d:e7:24:9d:f0:a3:
70:af:b7:1d:cl:6b:17:74:2a:21:41:cl:5d:13:cf:
aa:64:8e:f3:ed:6d:40:81:b7:f6:99:bd:1e:a0:a3:
a2:fd:69:c2:38:f5:28:50:49:86:7d:18:65:3a:78:
7d:11:90:f8:96:aa:ef:63:5f:4da:10:93:1d:19:8a:
60:15:ee:a7:fe:bb:ce:2e:07:54:8f:70:a6:cd:1a:
f7:32:c4:1e:83:7b:e2:0b:36:74:ac:ec:1d:8b:
0d:77:eb:16:db:cl:c5:1e
OpenSSL>

```

Figure 14

Chiffrée les clés avec la commande :

```
enc -des3 -in myRsa.pem -out myRsaCryp.pem (Pour crypter l'intégralité du
fichier créé)
```

(Ou possible de chiffrée avec :

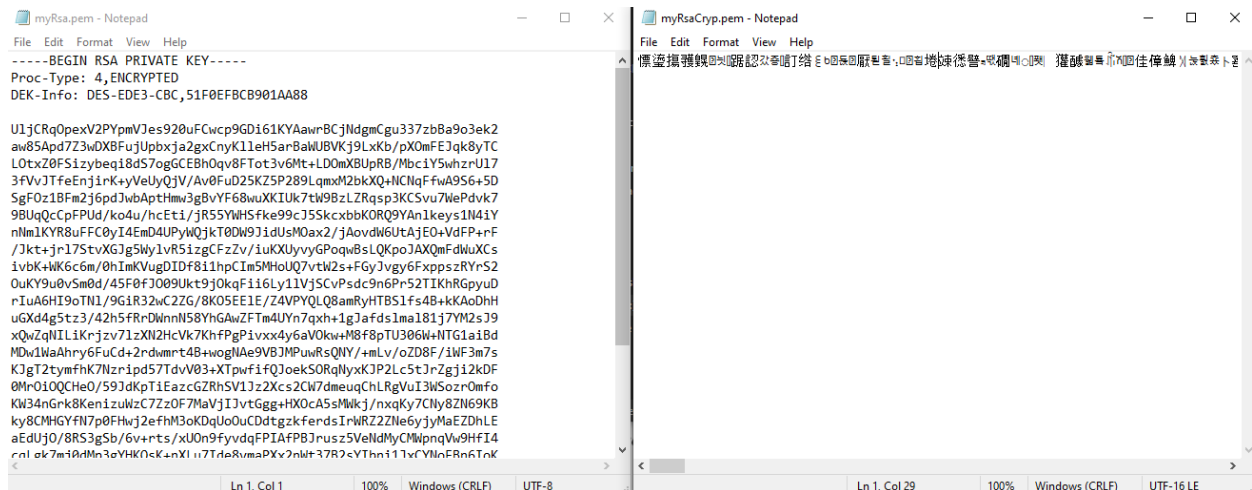
```
rsa -in myRsaPri.pem -des3 -out myRsaPriCryp.pem
```

Mais après d'extraire la clé privée comme dans les figures 22 et 23)

Figure 15


```
PS D:\Education\s6\secureit\td\tpopenssl\utilitaires> cat myRsaCryp.pem
Salted__eVz0;?K^AZUS!~oEoS~H^L_e9^c-sUNTx(C(0)XawezD8l<_oWuyt+,.03b~su!^oA(0m=[yãsoP%PLãH      zïOEI!/'zôAxS+LñveËÜ.oö.Y.iâu
"A*mo,QYUqsÖZaofofb   c AIA ey=of-mdeKCS3n>d$§¡vOçf-@Jj!0ü7-üãoRAAi#÷òom&f IzúJtIQ!ó0üUo_÷#ð@te.s`aaüy-v"Nk!t,aÜRe`)
ux!AO
filvb;!°SiD)>'°c£iia.p-2"!!UXUAäYÄYiisQoëI!iY^++w«N='R!°•-NNTYUX`aE·Av<?&°♦♦AuL9Ke-Üâë¶µma)öE"(¶-dëø3
S!@xEdxmE     !l+ 8i-äuhsEv!EP`H!H5Heß;p;E!E1„I,i,9!>û{on!au%10ogDeqz`xON97%“I76™E-övnöyyü,ö²lvoo/,`a!)xø9.ø-uGUm-z:S!uf
Q(xat?) Zgk0°C-2!üJCüff0K(ø.H)æ0=v-U<[U8+wPN4(      Z`!°÷OEa;ÄOIAXöçf)AGlõ:RODgm..wü.,t.18A!!10-[p"YÜ ö ÖI<qiqFV¥yINlXk`ö?i
p'k2XjbpuSLr~ ue"AY+i3y5Z>
AAYÜB`3ScYeßn!
Yovx~M`I55-u<&t`PKÅjzx">t`rENöY%„Y2?`D%æyüvZ>!hvÄil_NÇ_)rAA=f>äuöie603XC>d<x<4èäYü+SjCOëe|ö±.&iA5ääeo0C`+G
6E`ÄM°G Xyl3%j7~
=-EUel;e!ai €fux|c<c.Jdüwüá„!Yu6°!äOæo°@_-!Yüþ  ¢t°>iLæOS.U-Çic@~øÐÄOKI!ÜÜi eT0Y!ü"Ü²CY'S :
ö0%öIL!E`Y„AapWhzyN4U 9ösx°
et4!`Y0:"0003ñUEI°°ce f4ABYA°æKEEYV-o
üvuV!Z„æaooqÄf4ml=þ` NWJX ,6-éZarö.Øi$#!g&i/EÁ-c!""e0;5#AKl «Xxöö-e.ä      yôT_êps0!V~su."æü0e0C!°="8A#_ãA;AV~!öÇ!$!ÆEtL
!_EHc    7=L45s,¢Z+jzy`Hnq°cfEc .i:#0AU=äoSçgyiv&-&z oob/!YtøSAJyu;♠`<C!lgää UE U µ "ph80)<va
57NA!~L-°s„!ICCC!çzf=y! :t@ãq9L9LA!leq(e! „l.LNm TUWOIT!IS,Sükü!##%)0!AdSc6,f+2!È=Aø,øAw3cøheZ÷z M
-U°OX[CoIdX! T!üDl +e!ßT„!a!<öFo)!♦ ->ü3!öPöa+ tvSöVe!v# °Xyéb9!æARW-Y(MA#üA æo'pd'u0>,"°P-AE/V!WE.]\"O`Jau9!0ëa)-<+
ésQ!tëd5>epöfä`04!t
A0A§I!Z„æqaaY'bse:'9.øñq-'uq'y>aEDë!VD!§!` ÄA°ÆAxuh!Ü.huf:Ay,eEKrlÜN°>Lj`jYÜAEISEco°=æYÄ°®ESV°?7i!P'ÜäYa,?
!`!F106L:
!`muzu!p!hsAk„!9-a!>iqçb!v&VCLf.ÄëYö!ü!`!p qenhñ!`*XYA:s-SD!ßS00P°H`EEc°290b!H`æc°xvüAJ'e:eö°Nä-Bä0`X"!öDA` |b
BA, t2b!`_ei IE53!ø`a'p'!l!ü`d!ac<b!eJdz:Äo4!` `°#>!ÆA-A Pø»JTLtör-ÄDp:...
BA, öämcy!ö`e°N,gzöAAO
PS D:\Education\s6\secureit\td\tpopenssl\utilitaires>
```

le fichier crypté est plus sécurisé personne ne saura quelles sont vos clés car il est si difficile de savoir quel est le contenu du fichier est-ce des clés RSA ou quoi



- Extraction de clé public avec la commande :

Amir FARES L3 Informatique Groupe 02

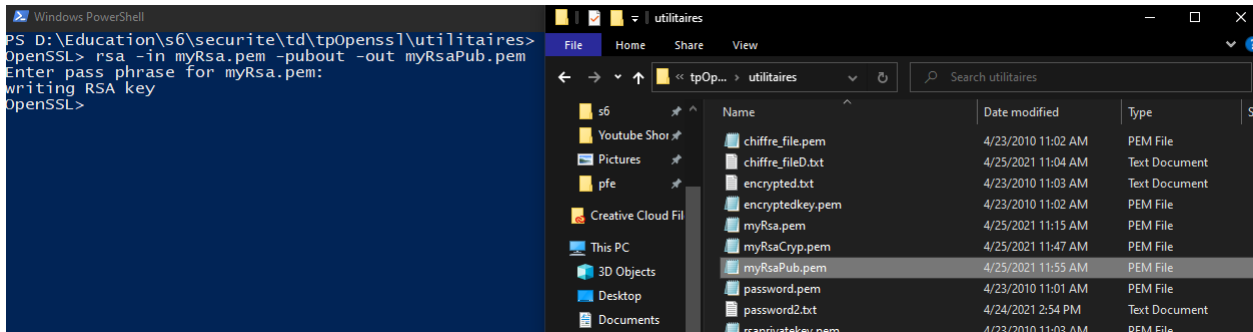


Figure 18

- Visualisation de clé public avec la commande :

Rsa -pubin -in myRsaPub.pem -text -noout

```

OpenSSL> rsa -pubin -in myRsaPub.pem -text -noout
RSA Public-Key: (2048 bit)
Modulus:
 00:bc:80:92:f5:3a:0f:9c:e3:6c:14:4a:9a:ab:d7:
 b6:a1:31:d1:8a:38:4c:5d:97:09:59:62:8a:b9:d3:
 62:1d:12:17:9c:9d:b2:66:d7:d1:c4:cf:04:92:a9:
 13:fd:b9:e6:59:11:0d:c3:22:f6:ec:f3:16:94:07:
 2a:f8:bf:81:a3:44:9a:49:b9:41:a6:d9:79:b2:71:
 d8:b5:af:b0:cd:88:9f:60:67:9e:4b:19:1c:ab:2e:
 1f:05:09:93:6b:c7:d7:d1:b4:85:ba:d2:53:18:69:
 82:76:37:02:b2:f0:35:07:05:19:73:ee:e0:62:03:
 2b:1d:fb:72:37:1a:cb:2c:be:18:d6:ac:ff:54:d3:
 0e:65:10:0d:59:23:1e:e0:dd:33:80:a3:e2:a8:41:
 71:0e:52:8f:9e:61:da:52:86:73:56:04:4b:aa:34:
 ba:18:99:91:14:60:78:ea:45:d9:54:d6:1f:00:9b:
 35:d0:c4:6a:75:cf:3a:f5:d2:2b:da:05:89:10:50:
 69:f4:05:00:86:d1:70:ac:21:5e:b9:6a:b4:20:21:
 7e:6b:56:c9:6d:06:c5:45:e3:06:c0:d5:5d:59:08:
 80:30:d7:e9:fe:1a:4e:7e:df:b0:5d:6f:cd:7b:1b:
 b5:90:b7:43:10:ac:02:d8:12:86:1f:40:43:55:e2:
 9c:67
Exponent: 65537 (0x10001)
OpenSSL>

```

Figure 19

- Création de document texte :

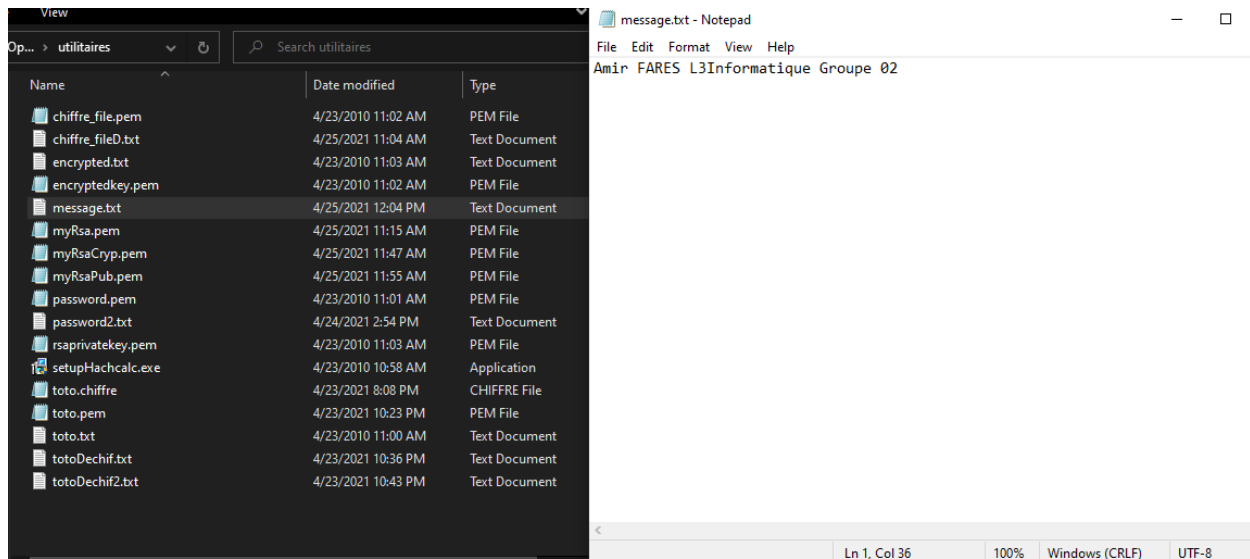


Figure 20

Chiffrée le fichier message.txt avec ma clé public on utilise la commande :

```
rsautl -encrypt -pubin -inkey myRsaPub.pem -in message.txt -out messageCryp.bin
```

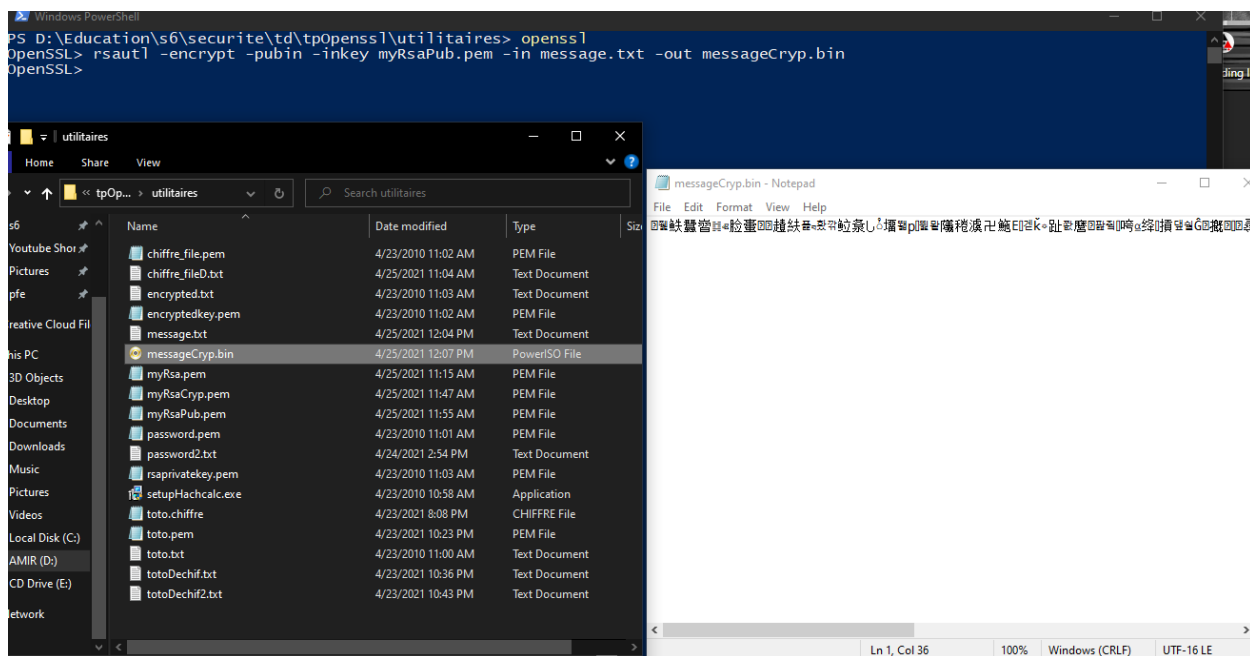


Figure 21

Extraire la clé privée dans un fichier spécifique avec cette commande:

rsa -in myRsa.pem -out myRsaPri.pem

(Nous pouvons crypter la clé privée à des fins de sécurité)

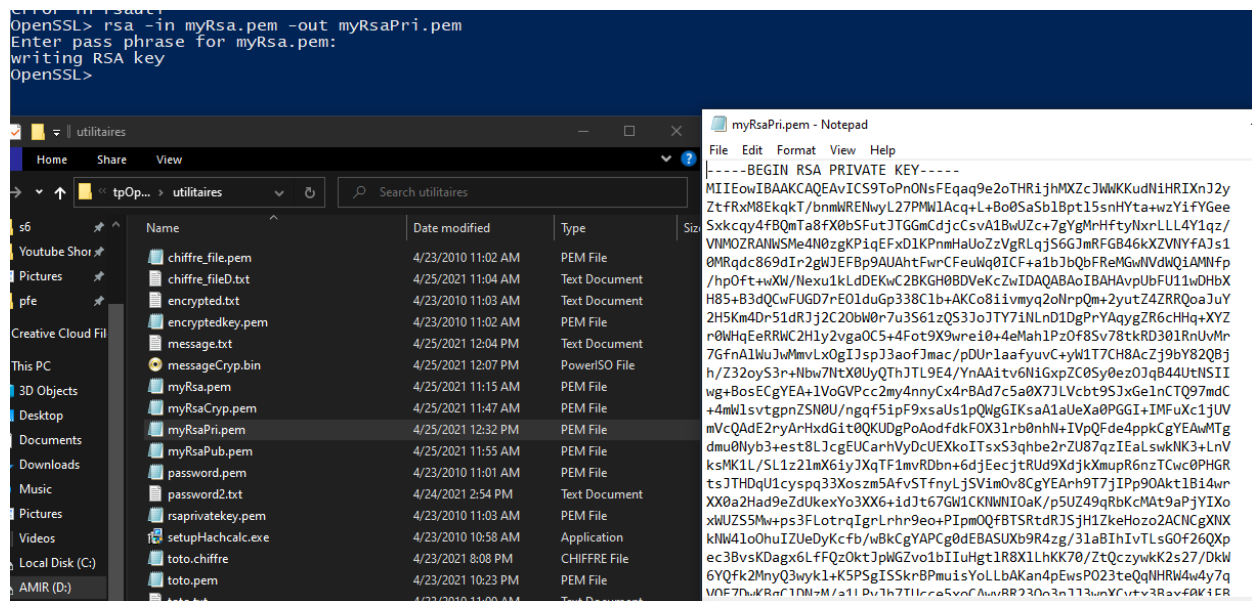


Figure 22

Maintenant, nous pouvons déchiffrer le message avec la clé privée en utilisant cette commande:

rsautl -decrypt -inkey myRsaPri.pem -in messageCryp.bin -out messageUnicr.txt

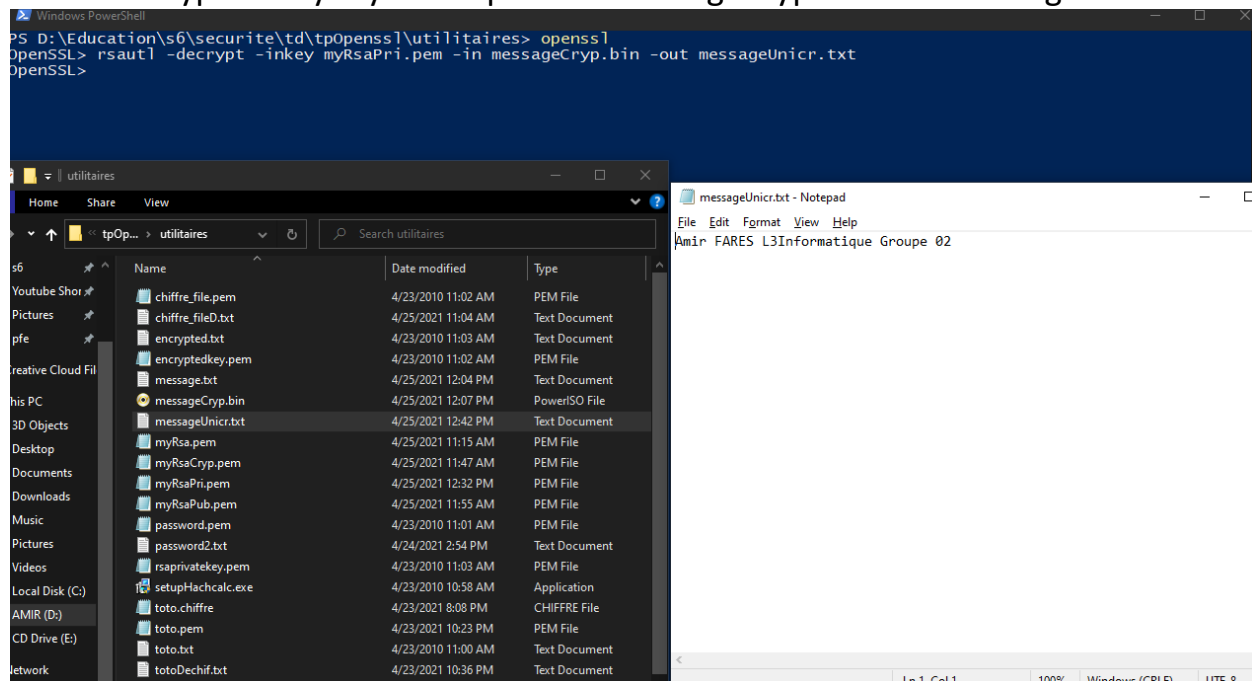


Figure 23