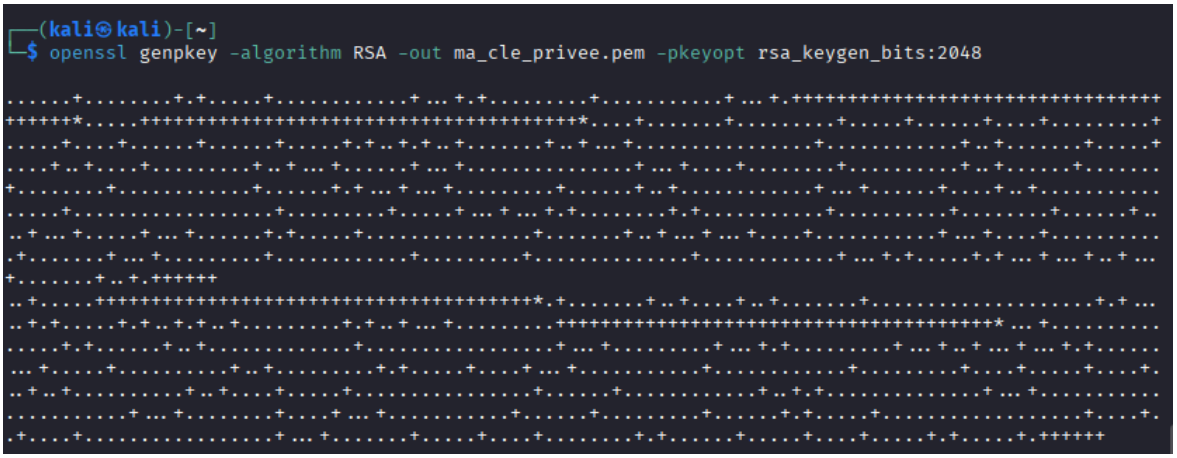
Rendu TP1 RSA

- ## TÂCHE 1: Génération et inspection de clés RSA

1. génération de clé privée et public(intégré)

```
┌──(kali㉿kali)-[~]
└─$ openssl genpkey -algorithm RSA -out ma_cle_privee.pem -pkeyopt rsa_keygen_bits:2048
```



2. extraire la clé publique

```
┌──(kali㉿kali)-[~]
└─$ openssl rsa -in ma_cle_privee.pem -pubout -out ma_cle_publique.pem
writing RSA key
```

3. structure de clé privé **RSA** (voir explication_rsa.txt pour plus de détails)

les 2 nombres premiers p et q représentent prime1 et prime2
n représente le modulus n=p*q
e représente l'exposant public
d représente l'exposant privé

4. Rôle (voir fichier explication_rsa.txt)

- ## TACHE2: chiffrement RSA d'un message texte

1. création d'un message texte

```
┌──(kali㉿kali)-[~]
└─$ echo " crypto asymetrique " >> message.txt
```

2. chiffrement de fichier

padding ou bourrage est l'ajout ou remplissage des données aléatoires pour avoir la taille accepter par rsa 256 bytes
avec le padding de pkcs#1 1.5

```
┌──(kali㉿kali)-[~]
└─$ openssl pkeyutl -encrypt -inkey ma_cle_publique.pem -pubin -in message.txt -out message.chiffre_pkc
s.bin
```

padding oaep:

```
┌──(kali㉿kali)-[~]
└─$ openssl pkeyutl -encrypt -inkey ma_cle_publique.pem -pubin -in message.txt -out message.chiffre_oae
p.bin -pkeyopt rsa_padding_mode:oaep
```

3. comparison dans les deux types de chiffrement:

la taille est identique 256bytes, contenu différent

```
┌──(kali㉿kali)-[~]
└─$ wc -c message.txt
21 message.txt

┌──(kali㉿kali)-[~]
└─$ wc -c message.chiffre_pkcs.bin
256 message.chiffre_pkcs.bin

┌──(kali㉿kali)-[~]
└─$ wc -c message.chiffre_oaep.bin
256 message.chiffre_oaep.bin
```

4. **déchiffrement:**

```
┌──(kali㉿kali)-[~]
└─$ openssl pkeyutl -decrypt -in message.chiffre_pkcs.bin -out message_dechiffre.pkcs -inkey ma_cle_pri
vee.pem -pkeyopt rsa_padding_mode:pkcs1

┌──(kali㉿kali)-[~]
└─$ cat message_dechiffre.pkcs
 crypto asymetrique
```

```
┌──(kali㉿kali)-[~]
└─$ openssl pkeyutl -decrypt -in message.chiffre_oaep.bin -out message_dechiffre.oaep -inkey ma_cle_pri
vee.pem -pkeyopt rsa_padding_mode:oaep

┌──(kali㉿kali)-[~]
└─$ cat message_dechiffre.oaep
crypto asymetrique
```

signature

```
┌──(kali㉿kali)-[~]
└─$ openssl pkeyutl -sign -inkey ma_cle_privee.pem -in message.txt -out signature.bin

┌──(kali㉿kali)-[~]
└─$ openssl pkeyutl -verify -inkey ma_cle_publique.pem -pubin -in message.txt -sigfile signature.bin
Signature Verified Successfully
```

Question: Pourquoi le RSA ne permet-il pas de chiffrer un gros fichier directement ? à cause de la taille c'est pour ca on recourt à AES si taille dépasse la taille accepter par rsa

- **TÂCHE 3: Export, conversion et compatibilité**

```
┌──(kali㉿kali)-[~]
└─$ openssl rsa -in ma_cle_privee.pem -outform DER -out ma_cle_privee.der

writing RSA key

┌──(kali㉿kali)-[~]
└─$ cat ma_cle_privee.der
◆◆0◆◆◆◆◆Ll´◆◆^◆◆◆
◆ε=q◆S*)dg◆◆◆Z◆◆◆◆◆◆cVa▒o◆◆)◆◆<▒◆@◆◆H◆/◆◆◆◆◆◆◆À◆◆◆◆V◆◆1◆◆na◆%◆◆h3f◆◆e◆*q◆
:`◆sKe◆ķ3◆δW◆3<m$8◆FQ◆◆UC◆へ
                              ◆◆%◆V:◆a)^D1P◆}◆,◆◆◆◆◆q{◆W◆{◆I}_◆◆m-{◆r◆◆◆◆!◆◆◆◆lẽz◆◆◆◆
                                                                                    ◆į◆◆r◆+}◆q!◆◆2ƒH◆M◆
8◆W◆g◆◆◆A◆◆◆C◆wdm0&K+CQ
Y◆◆◆◆◆◆g◆DZ◆Xw<w◆◆◆◆S◆0U◆◆m◆:◆▒◆E,◆◆◆◆◆◆◆A◆◆◆=-◆◆(◆!4L◆◆◆`L◆ṻ/◆Y◆)◆?◆◆◆I◆5Z◆[%▒◆◆%4◆P◆◆▒K◆◆◆◆◆◆9[.bN◆◆◆
◆◆,◆b◆◆◆o
 ◆◆▒Y◆g▒◆◆◆◆s◆u>◆@◆?◆"C◆◆?▒◆◆◆G!◆j◆◆h:◆Y5r◆◆◆k8◆◆◆|◆8^H◆◆x◆◆◆◆?×◆"4M◆◆Oll@◆◆◆<◆rBfm◆◆◆◆◆Lı◆e◆◆◆+◆1AS◆
                                                                                    5◆ऽ@
◆◆◆◆g◆◆|◆E◆◆◆◆◆◆4}◆◆Ă7◆◆&wO◆◆◆◆◆J◆◆◆5◆◆◆◆◆m
Y       ◆▒◆4x◆◆◆◆◆◆'ڰ}◆(clU◆◆=BF◆◆&◆V◆r◆d◆◆,◆◆◆◆E◆Ÿ◆Ѧ◆◆3◆]◆^Y,◆N◆ÂOpL$▒◆◆C◆&◆◆◆◆Q◆◆f◆+v◆h◆`x◆◆Ӈ◆◆;◆◆d◆◆
Q◆◆◆@c]V◆4c◆▒l\J)◆44Bb]'◆◆R◆10+3E◆'◆◆◆◆◆] |◆◆c◆0>◆◆$=▒◆◆g◆J◆◆◆◆◆◆|+◆y◆ sKY_B
>◆◆◆◆◆▒◆◆9◆R◆;!x %▒◆t◆◆◆2◆#+ŨB◆2◆◆kH◆sE◆◆◆y◆M◆◆4/6◆◆◆◆◆^◆"fi^Q◆◆◆a◆◆}◆◆◆◆◆I2◆i◆M75cnU
j◆◆◆◆c◆◆-◆◆◆◆◆◆◆Y◆◆:◆◆◆A◆◆◆◆ho◆D◆'◆◆C◆◆w◆◆?◆
                                                dd
                                                  }-Y◆|◆◆◆◆◆M◆W)◆Uڰ+\◆◆◆◆◆~ ◆&◆◆◆;◆{]C◆7Iw◆◆◆]◆SAε◆◆◆◆Y◆◆◆◆
◆%◆
```

```
┌──(kali㉿kali)-[~]
└─$ openssl rsa -inform DER -in ma_cle_privee.der -out ma_cle_privee_convertie.pem
writing RSA key

┌──(kali㉿kali)-[~]
└─$ cat ma_cle_privee_convertie.pem
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQCzhkxs1ZvLF9sd
XpHsHeoMBqI5Ooak8ktsDe/OtT1xplMqKQNkZ+IA7KBaHYSs7Bm68utjVmEYbxao
5Sm9sTzevIkUQOaIFUjlfy+hrfOBwNbRyICjiv79VukP/TGa9tC7YfolsNBoMxtS
Zrz2ZdIqcaoKOmCzcxxLZegQxLczgc60V9MzPG0kOKJGUcTIVUPx77i/C5GrBSXL
VjqZF2EpEV5EMVC4fbYs/57Bscdxe9NXFYJ7hkl9EgFfwQScbS0We9FywReXl/2+
uwgh+aCq1GzIhXqglYiKC8jEr5Hm3qxyjSt9lnHVdAgIECEXiroyxpFI202KONtX
q2fb5O1BAgMBAAECggEAAbecA0OMd2RtMCZLK0NRClmu8bD4nQShDtaBikRaxlh3
PBx3/waJtehT8DBVkeVttjqsGh23RSymjoz0hOWVQcMGi4A9LRT8iiiMITRM2e6M
YEwPxti0L8jPk9Up5T8Qn/+iSc8VNQ9aygNbJRitxAIlNIhQ5Y7vlZ0ZS52qsoCg
4A45WxUuYk7mvsF03SwGl2Lc3MgBbw3hC2QECBG9xRhZ3mfujr+ej7rrD3OrdT4Z
tkC5P74iQ5WGPxkYrM7LRyGMagDktWg62FkONXKZGRT45Ws46vePfME4XkiMhXi9
sJrLPyoQmw8CIjRN/7JPHseBQMCDlTyqckJmbZyxEQKBgQD71oerZf+V6Cv0MUFT
Ga0LNd/KlkCgeGoPwYlC0xHTpKy4nTUXByQhW9d3keei4hbXfM79Uy2GSsGwAUv7
PWqVNbvBupXSbQ0uHY46o8twid1LZJqP+wnbkjY3Bi3eSdTzcvsPTNO3Be6IRLvO
6v3yDSeCe12WiYfVmHe7hHR0MQKBgQC2fdOprMSCN4m3JncQT5akHreDrh9KDeS6
/rweZ/CjfNxFvpat7qJ/CTQKWQmlGpU0eKqZkbuW8AUn2rx9yihjbFWknT1CRtoS
+Sb7E1a0cqFkf7HALP6QEJfxRReZ066/yLqFxTPMXaxeWSyTTtrIgk9wTCTqopqz
qkOpJr0WEQKBgQCIUbkAk2bFK3bXaO4WYHil79Si4bA77NJkkvuM25nzuVjxWc+K
Foh9MiyxAZ6kMcPxgxK8V5D193NepWUOq1ENUZe/pxNAY11WEBOUNGOEGGxcSinM
NDRC0aNdJ5AemlKWMTArOQgFM0XKJ4KN3+/0XSB85fBj8zA+85gkPe+XkQKBgBZn
kEruiRmrk97GfCumecogc0tZX0IKPqCukY3yGNsZgzmU0o78EzsheCAlGkgI4XSV
s+wytiMryJZCBo4y+Y5rSIBzRYgThsZ5jE2a7DQvNp1OCBad74rkXqMiZmleUb/9
9thhg6B9h6H1tMFJFxcy2GnjD003NWMFblUgCmrhAoGBAPhjze0BLefn/Ivv5olZ
nYh3CDq97LZBp/AG8f0VaG8GrEQX6ifeGaseQ4Gbd4rwHD+DDGQX1IEMfS1ZpHyr
j5KjkE3IVwcbBywPKaNV25ErXJa++8/mfn76JrCGqDuZe11DwjdJd9Hjxl2EUx1B
2aTvzIrD8/tZiBeAEvjxtyXT
-----END PRIVATE KEY-----
```

les clés sont identiques

Format PEM format lisible ascii , DER format illisible