**Nom :** Medjahdi

**Prénom :** Lamia

**Promo :** M1 SIC

**TP Cryptographie appliquée – RSA**

**Objectif du tp :**

Apprendre à générer des clés RSA publiques et privées à partir des bibliotheques

Crypter/Decrypter des données (texte / fichier externe) avec les clés générées

**Outils utilisés :** Idle – langage : python

**Question 1 : (echauffement)**

*import* os

*import* math

*from* Crypto.Cipher *import* AES

*from* Crypto.Cipher *import* PKCS1\_OAEP

*from* Crypto.PublicKey *import* RSA

fichier=open("1\_crypte.txt","rb")

data=fichier.read()

#*importe la cle prive du fichier*

contenu\_cleprive = RSA.importKey(open("fichier\_cle\_prive.pem","rb").read())

#*cree un objet a partir de la cle permettant de chiffrer avec RSA*

objet\_cle\_rsa\_prive = PKCS1\_OAEP.new(contenu\_cleprive)

#*imprime le resultat du dechiffrement*

print(objet\_cle\_rsa\_prive.decrypt(data))

**Resultat : Nous avons importé la clé privé – créé un objet a partir de la clé et puis on a déchiffré le contenu du fichier 1\_crypte.txt**



**Question 2  :**

**On commence par générer la clé publique avec le fichier Listing 1 :**

#*importe le module RSA*

*from* Crypto.PublicKey *import* RSA

#*genere une cle publique/prive avec modulos (N) de 2048 bits*

cle= RSA.generate(2048)

#*recupere la cle prive*

cle\_prive = cle.exportKey()

#*ouvre un fichier en mode d'ecriture en octets*

Fichier = open("fichier\_cle\_prive.pem", "wb")

#*ecrit le contenu de la cle prive sur fichier*

Fichier.write(cle\_prive)

#*ferme le fichier*

Fichier.close()

#*recupere la cle publique*

cle\_publique = cle.publickey().exportKey()

#*ecrit le contenu de la cle publique sur fichier*

Fichier = open("fichier\_cle\_publique.pem", "wb")

Fichier.write(cle\_publique)

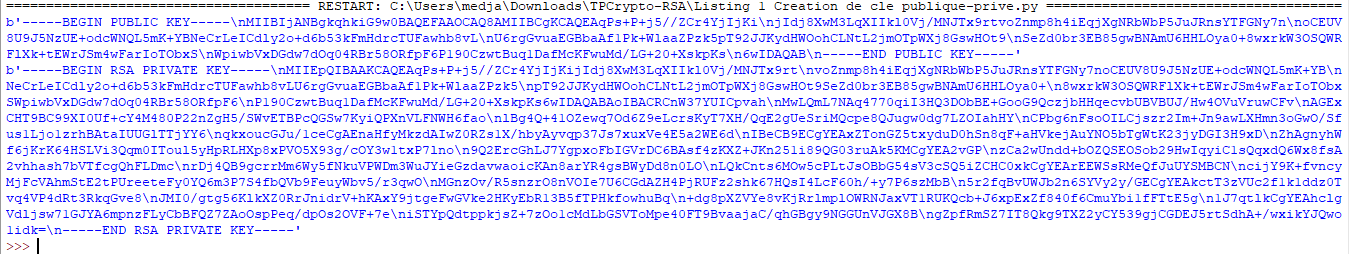
Fichier.close()

#*imprime la cle publique*

print(cle\_publique)

#*imprime la cle RSA*

print(cle\_prive)

**à l’execution ca affiche :**

**Et ca genere un fichier fichier\_cle\_publique.pem qui contient la clé publique**

**puis on envoie le contenu au serveur**

*import* socket

*import* \_thread *as* thread

*from* Crypto.PublicKey *import* RSA

*from* Crypto.Cipher *import*  PKCS1\_OAEP

def Traiter\_Connexion(connexion\_avec\_client,adresse\_client):

   global objet\_cle\_rsa\_prive,contenu\_clepublique

   MessageRec=b""

   print ("Connexion de la machine = ", adresse\_client)

   connexion\_avec\_client.send(contenu\_clepublique)

*try*:

*while* True:

          MessageRec=connexion\_avec\_client.recv(1024)

          MessageRec=objet\_cle\_rsa\_prive.decrypt(MessageRec)

*if* MessageRec==b"Fin":

*break*

          print("Client" ,adresse\_client," a dit :",MessageRec.decode())

*except*:

      print("Deconnexion")

   print("Deconnexion de :",adresse\_client)

*try*:

      connexion\_avec\_client.close()

*except*:

*pass*

SocketServeur = socket.socket()

host = socket.gethostname()

port = 9500

SocketServeur.bind(("127.0.0.1", port))

SocketServeur.listen(5)

contenu\_clepublique = open("fichier\_cle\_publique.pem","rb").read()

contenu\_cleprive = RSA.importKey(open("fichier\_cle\_prive.pem","rb").read())

#*cree un objet a partir de la cle permetant de chiffre avec RSA*

objet\_cle\_rsa\_prive = PKCS1\_OAEP.new(contenu\_cleprive)

print("Lancement serveur")

*while* True:

   ConnexionAUnClient, addrclient = SocketServeur.accept()

   thread.start\_new\_thread(Traiter\_Connexion,(ConnexionAUnClient,addrclient))

**Résultat :**

**Le serveur est lancé :**

**Question 3-4-5 :**

*import* socket

*from* Crypto.PublicKey *import* RSA

*from* Crypto.Cipher *import*  PKCS1\_OAEP

SocketClient = socket.socket()

host = socket.gethostname()

port = 9500

SocketClient.connect(("127.0.0.1", port))

Cle\_publique=b""

*while* True:

        recu=SocketClient.recv(1024)

        Cle\_publique+=recu

*if* b'-----END PUBLIC KEY-----' in Cle\_publique:

*break*

clepublique=RSA.importKey(Cle\_publique)

objet\_cle\_rsa\_publique = PKCS1\_OAEP.new(clepublique)

print("Lancement serveur")

*while* True:

   MessageATransmettre=input().encode()

   #*Chiffre le message lu du clavier (MessageATransmettre)*

   #*Utilisez la methode encrypt de objet\_cle\_rsa\_publique*

   #*Mettez le resultat dans resultat\_chiffre*

   #*(A FAIRE 2) ajoutez l'instruction dans cette ligne*

   SocketClient.send(objet\_cle\_rsa\_publique.encrypt(MessageATransmettre))

*if*(MessageATransmettre==b"Fin"):

*break*

      print( "Deconnexion de :",addrclient)

ConnexionAUnClient.close()