**INFORMATION SECURITY LAB – 18IT580**

**Name : Reshma A Reg No: 19IT077**

1. **RSA: CODE:**

#RSA

# p 1312961745803

# q 1000000999999

# e 1284916147

import math print("\*\*\*\*\*\*RSA\*\*\*\*\*\*")

p = int(input("Enter the prime no p :")) q = int(input("Enter the prime no q :")) print("p = ",p)

print("q = ",q) #n = p\*q

n= p\*q print("n = ",n)

#phi\_n = (p-1)\*(q-1)

phi\_n = (p-1)\*(q-1) print("Euler's totient = ",phi\_n) #calculate public key (e)

for i in range(2,phi\_n): if(math.gcd(i,phi\_n)==1):

e=i break

else:

continue print("public key (e) = ",e) #calculate private key (d) for i in range(1,phi\_n):

if(((i\*e)%phi\_n)==1): d=i

break

else:

continue print("private key (e) = ",d)

def encrypt(M,e,n): c=pow(M,e,n) #ct = txt[c] return c

def decrypt(C,d,n): p=pow(C,d,n) pt = txt[p] return pt

txt = "abcdefghijklmnopqrstuvwxyz"

choice = int(input("Enter 1-Encryption 2-Decryption 0-Exit")) while(choice!=0):

if(choice==1):

m = str(input("Enter the text to encrypt :")) M = txt.index(m)

cipher = encrypt(M,e,n) print("Encrypted cipher text = ",cipher)

elif(choice==2):

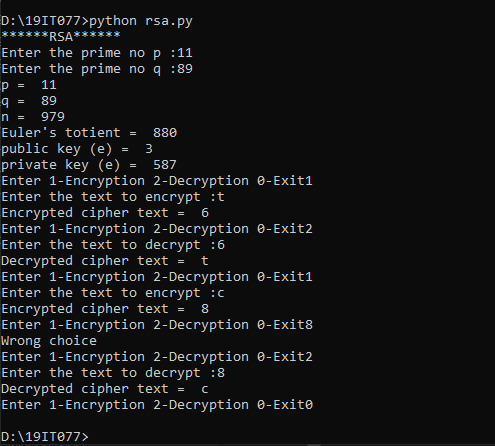
C = int(input("Enter the text to decrypt :")) plain = decrypt(C,d,n)

print("Decrypted cipher text = ",plain) else:

print("Wrong choice")

choice = int(input("Enter 1-Encryption 2-Decryption 0-Exit"))

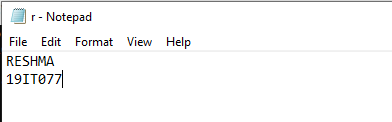
**OUTPUT:**



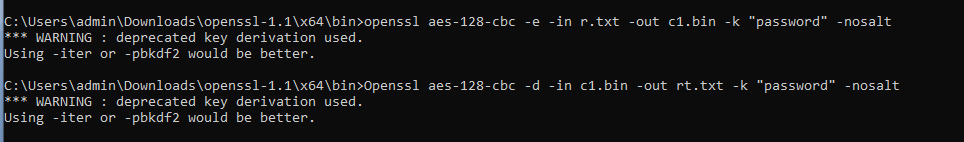
1. **Openssl :**

**Demonstrate the usage of encryption/ decryption using AES, RSA, SHA and Digital Signature in openssl**

**Creating a txt file r.txt**



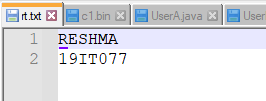
**AES**



Openssl aes-128-cbc -e -in r.txt -out c1.bin -k “password” -nosalt

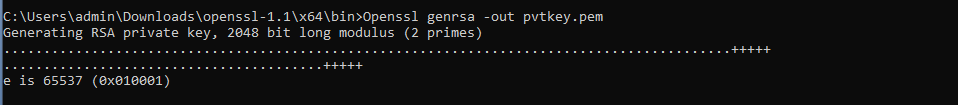


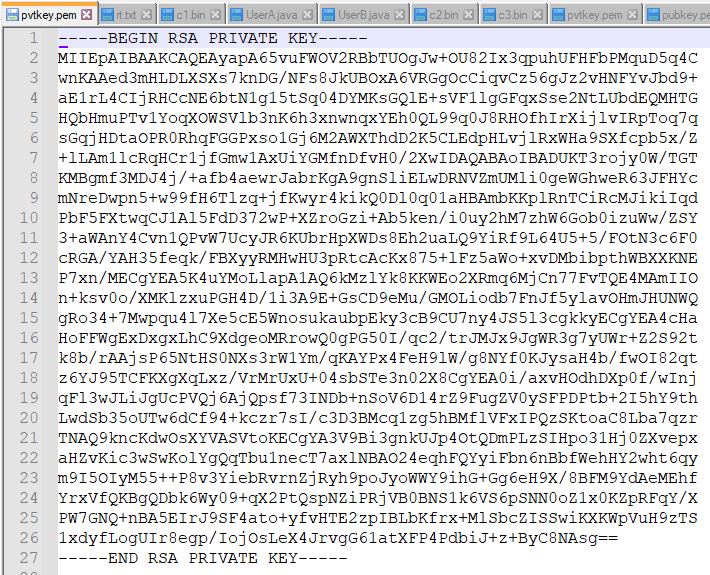
Openssl aes-128-cbc -d -in c1.bin -out rt.txt -k “password” -nosalt



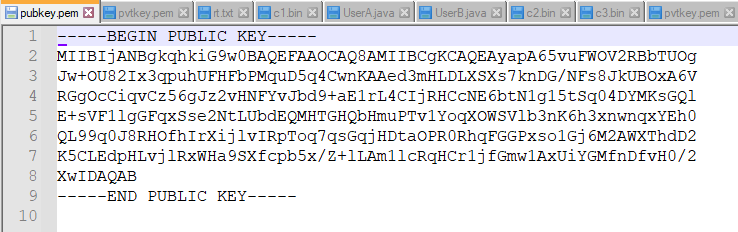
#rsa

Openssl genrsa -out pvtkey.pem

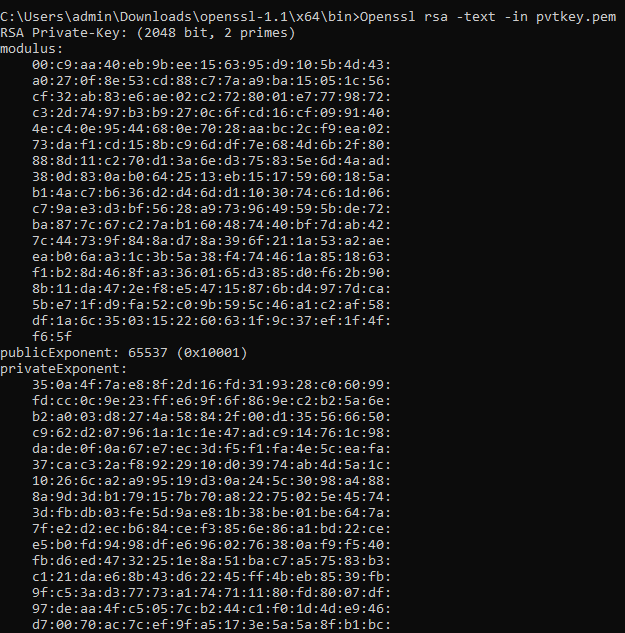


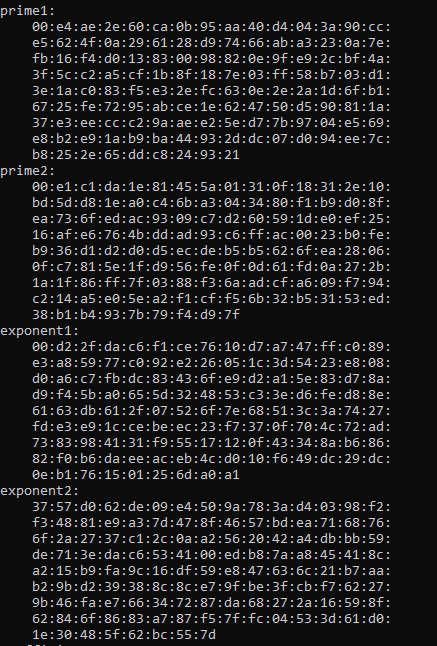


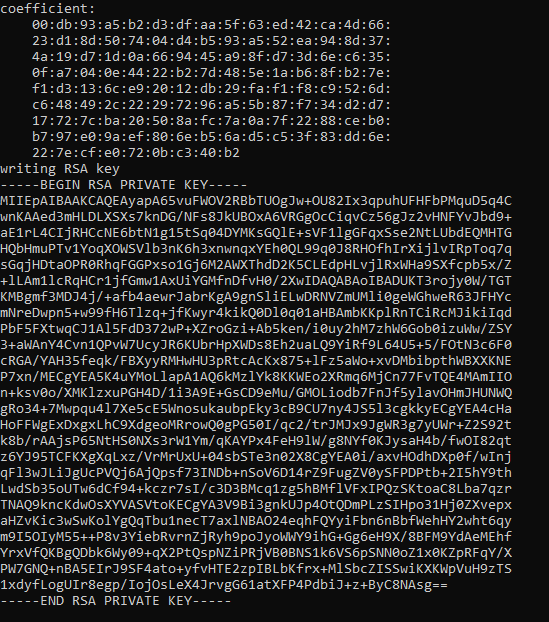
Openssl rsa -pubout -in pvtkey.pem -out pubkey.pem



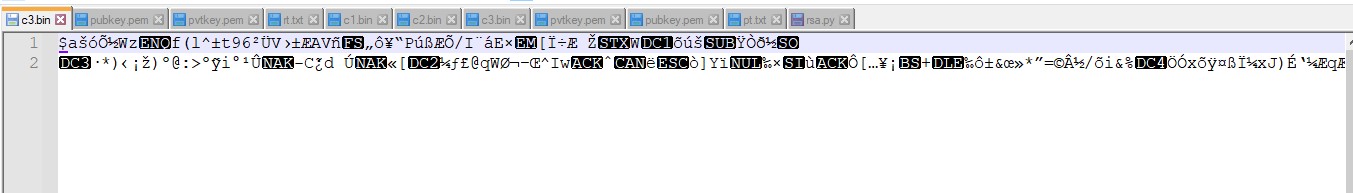
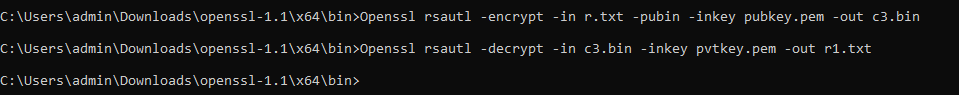
Openssl rsa -text -in pvtkey.pem



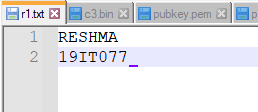




Openssl rsautl -encrypt -in plain.txt -pubin -inkey pubkey.pem -out c3.bin



Openssl rsautl -decrypt -in c3.bin -inkey pvtkey.pem -out r1.txt



#SHA

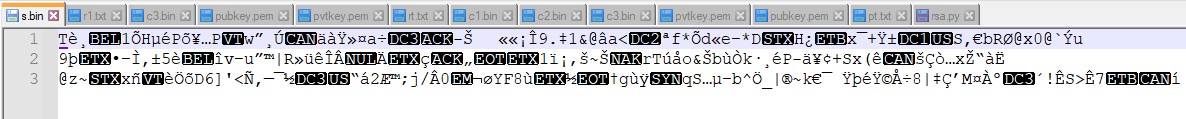
Openssl md5 r.txt



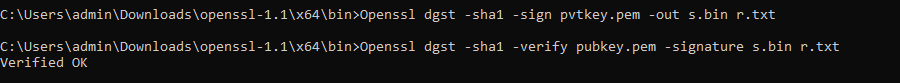
Openssl SHA256 r.txt



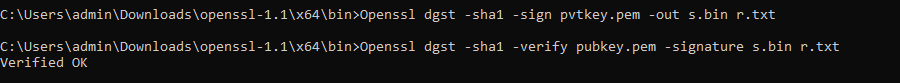
#digital signature



Openssl dgst -sha1 -sign pvtkey.pem -out s.bin r.txt



Openssl dgst -sha1 -verify pubkey.pem -signature s.bin r.txt



**Result:**

**Thus the usage of encryption/ decryption using AES, RSA, SHA and Digital Signature in openssl is demonstrated.**