Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Experiment 6

Date of Performance : 13-03-2023
Date of Submission : 20-03-2023
SAP Id: 60004200107 **Name :** Kartik Jolapara
Div: B **Batch :** B1

AIM

Implement RSA cryptosystem. Demonstrate the application of RSA cryptosystem using multimedia data.

THEORY

RSA algorithm is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes that the Public Key is given to everyone and the Private key is kept private.
The idea of RSA is because it is difficult to factorize a large integer. The public key consists of two numbers where one number is a multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So, if somebody can factorize the large number, the private key is compromised. Therefore, encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken soon. But till now it seems to be an infeasible task.

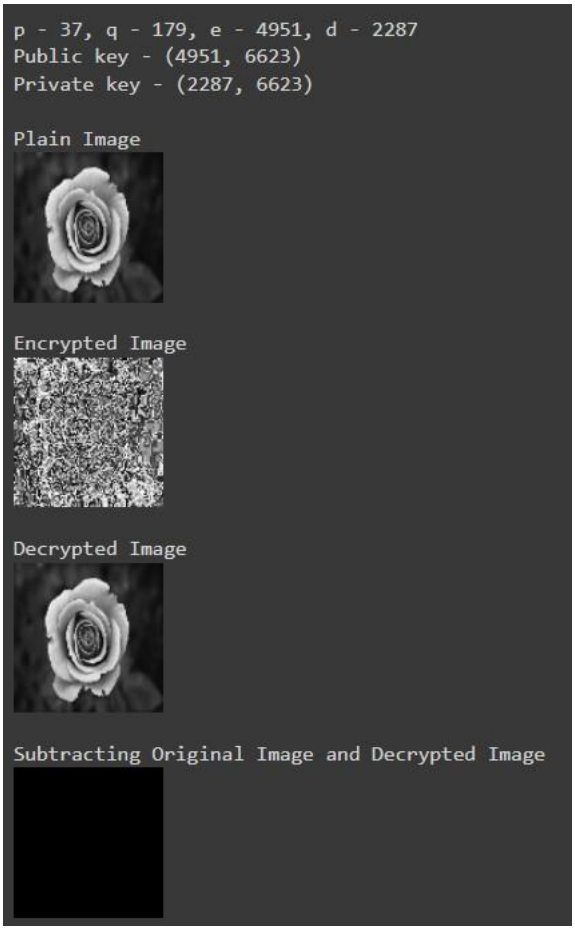
PROGRAM

```
import random as r from
PIL import Image from
numpy import array
import numpy as np

def modInverse(e, phin):        for d in range(1, phin):
        if (((e % phin) * (d % phin)) % phin == 1):
                        return d
        return -1

def prime(a):
count = 0   for i in
range(2, int(a/2)):
if a % i == 0:
count += 1   if
count == 0:
```

```
return True   else:
return False

cond = 1 count
= 0 while
cond:
p         = r.randint(2,
255)   if prime(p):
break

cond = 1
while cond:
q         = r.randint(2,
255)   if prime(q) and
q != p:
   break

n = (p * q) phin =
(p - 1)*(q - 1)

cond = 1
while cond:
  e = r.randint(2, phin)   if
prime(e) and e != p and e != q:
   break

d = modInverse(e, phin)

while d == -1:
cond = 1
  temp = e
while cond:
e = r.randint(2, phin)     if prime(e) and e != p and e != q and e != temp:
     break   d =
modInverse(e, phin)
print(f"p - {p}, q - {q}, e
- {e}, d - {d}")

publicKey = (e, n)
privateKey = (d, n)

print(f"Public key - {publicKey}")
print(f"Private key - {privateKey}")
```

```
im = Image.open(r"/content/download (1).jpg")
ar = array(im) list(ar)

result = np.zeros((100,100,3))
for i in range(100):   for j in
range(100):     for k in
range(3):
     result[i][j][k] = (int(ar[i][j][k])**e)%n

print("\nPlain Image") im =
Image.fromarray(np.uint8(ar))
im.show() print("\nEncrypted Image")
im = Image.fromarray(np.uint8(result))
im.show()

result_dec = np.zeros((100,100,3))
for i in range(100):   for j in
range(100):     for k in range(3):
     result_dec[i][j][k] = (int(result[i][j][k])**d)%n

print("\nDecrypted Image") im =
Image.fromarray(np.uint8(result_dec))
im.show()

check =
np.zeros((100,100,3)) for i in
range(100):   for j in
range(100):     for k in
range(3):
     check[i][j][k] = int(ar[i][j][k])-int(result_dec[i][j][k]) print("\nSubtracting Original Image
and Decrypted Image") r = Image.fromarray(np.uint8(check)) r.show()
```

INPUT AND OUTPUT



CONCLUSION

Thus, we have successfully implemented RSA cryptosystem.