**RSA ECRYPTION ALGORITHM**

**BY-**

**AAKRITI MEHROTRA**

**22BCE1954**

**CODE**

**SERVER.PY**

import socket

def power(base, expo, m):

res = 1

base = base % m

while expo > 0:

if expo & 1:

res = (res \* base) % m

base = (base \* base) % m

expo = expo // 2

return res

def modInverse(e, phi):

for d in range(2, phi):

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

return d

return -1

def gcd(a, b):

while b != 0:

a, b = b, a % b

return a

def generateKeys():

p = 7919

q = 1009

n = p \* q

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

e = 0

for e in range(2, phi):

if gcd(e, phi) == 1:

break

d = modInverse(e, phi)

return e, d, n

def decrypt(c, d, n):

return power(c, d, n)

def receive\_data(host, port):

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_socket.bind((host, port))

server\_socket.listen(1)

print(f"Server listening on {host}:{port}...")

conn, addr = server\_socket.accept()

data = conn.recv(1024).decode()

conn.close()

return int(data)

if \_\_name\_\_ == "\_\_main\_\_":

e, d, n = generateKeys()

print(f"Public Key (e, n): ({e}, {n})")

print(f"Private Key (d, n): ({d}, {n})")

encrypted\_message = receive\_data("127.0.0.1", 12345)

print(f"Received Encrypted Message: {encrypted\_message}")

decrypted\_message = decrypt(encrypted\_message, d, n)

print(f"Decrypted Message: {decrypted\_message}")

**CLIENT.PY**

import socket

def power(base, expo, m):

res = 1

base = base % m

while expo > 0:

if expo & 1:

res = (res \* base) % m

base = (base \* base) % m

expo = expo // 2

return res

def encrypt(m, e, n):

return power(m, e, n)

def send\_data(host, port, data):

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client\_socket.connect((host, port))

client\_socket.send(str(data).encode())

client\_socket.close()

if \_\_name\_\_ == "\_\_main\_\_":

e = 5

n = 7990271

M = 1234

print(f"Original Message: {M}")

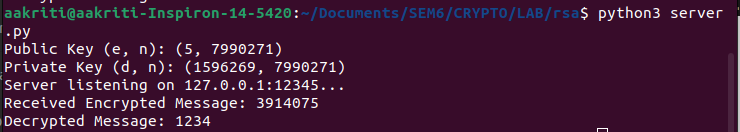
C = encrypt(M, e, n)

print(f"Encrypted Message: {C}")

send\_data("127.0.0.1", 12345, C)

**OUTPUT**

**SERVER**

****

**CLIENT**

