**DIGITAL SIGNATURE ALGORITHM**

**BY-**

**AAKRIT MEHROTRA**

**22BCE1954**

**CODE**

**SERVER**

import socket

def euclid(m, n):

if n == 0:

return m

else:

return euclid(n, m % n)

def exteuclid(a, b):

r1, r2 = a, b

s1, s2 = 1, 0

t1, t2 = 0, 1

while r2 > 0:

q = r1 // r2

r1, r2 = r2, r1 - q \* r2

s1, s2 = s2, s1 - q \* s2

t1, t2 = t2, t1 - q \* t2

if t1 < 0:

t1 += a

return r1, t1

p, q = 823, 953

n = p \* q

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

e = 313

r, d = exteuclid(Pn, e)

def verify\_message(M, S):

M1 = pow(S, e, n)

return M == M1

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

server\_socket.bind(('localhost', 12345))

server\_socket.listen(1)

print("Server listening on port 12345...")

conn, addr = server\_socket.accept()

print("Connected by", addr)

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

M, S = map(int, data.split(','))

print(f"Received M: {M}, S: {S}")

if verify\_message(M, S):

response = "Valid Signature. Accepting the message."

else:

response = "Invalid Signature. Rejecting the message."

conn.send(response.encode())

conn.close()

server\_socket.close()

**CLIENT**

import socket

def exteuclid(a, b):

r1, r2 = a, b

s1, s2 = 1, 0

t1, t2 = 0, 1

while r2 > 0:

q = r1 // r2

r1, r2 = r2, r1 - q \* r2

s1, s2 = s2, s1 - q \* s2

t1, t2 = t2, t1 - q \* t2

if t1 < 0:

t1 += a

return r1, t1

# Private Key

p, q = 823, 953

n = p \* q

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

e = 313

r, d = exteuclid(Pn, e)

M = int(input())

S = pow(M, d, n)

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

client\_socket.connect(('localhost', 12345))

message = f"{M},{S}"

client\_socket.send(message.encode())

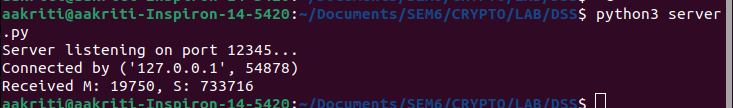
response = client\_socket.recv(1024).decode()

print("Server Response:", response)

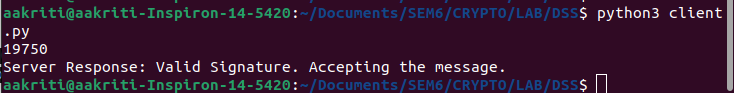
client\_socket.close()

**OUTPUT**

**SERVER**

****

**CLIENT**

****