RSA ENCRYPTION ASSIGNMENT

Server.py

import socket

import math

import random

from string import ascii\_letters

s=socket.socket()

s.bind(('localhost',9999))

s.listen(1)

print('Server listening...')

prime = set()

def prime\_filler():

prime\_check=[True]\*250

prime\_check[0]=False

prime\_check[1]=False

for i in range(2,256):

for j in range(i\*2,250,i):

prime\_check[j]=False

for i in range(len(prime\_check)):

if prime\_check[i]:

prime.add(i)

def randomprime():

rand = random.randint(0,len(prime)-1)

it = iter(prime)

for \_ in range(rand):

next(it)

res= next(it)

prime.remove(res)

return res

def set\_primes():

p=randomprime()

q=randomprime()

# print(p)

# print(q)

global n

n= p\*q

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

global e

e=2

while True:

if math.gcd(e,phi)==1:

break

e+=1

global d

d=0

while True:

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

break

d+=1

# print(e)

# print(d)

def encrypt(val):

encry\_text= pow(val,e)%n

return encry\_text

def encoder(message):

enc=[]

for char in message:

enc.append(encrypt(ord(char)))

return enc

while True:

conn,addr = s.accept()

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

print('Received plain\_text:',msg)

prime\_filler()

set\_primes()

code = encoder(msg)

conn.send(bytes(str(code),'UTF-8'))

d\_bytes = d.to\_bytes(4, byteorder='big') # 4 bytes for a 32-bit integer

n\_bytes = n.to\_bytes(4, byteorder='big')

conn.send(d\_bytes)

conn.send(n\_bytes)

conn.close()

break

s.close()

Client.py

import socket

from string import ascii\_letters

cipher\_letters='nzghqkcdmyfoialxevtswrupjbNZGHQKCDMYFOIALXEVTSWRUPJB'

c=socket.socket()

c.connect(('localhost',9999))

print('Client connected')

pt=input('Enter plain text:')

c.send(bytes(pt,'UTF-8'))

def decrypt(val\_1):

decry=pow(val\_1,d)%n

return chr(decry)

def decoder(code):

s=""

for i in code:

s+=decrypt(int(i))

return s

while True:

received\_cipher=c.recv(1024).decode()

received\_d\_bytes = c.recv(4)

received\_n\_bytes = c.recv(4)

# Convert bytes to integers

d = int.from\_bytes(received\_d\_bytes, byteorder='big')

n = int.from\_bytes(received\_n\_bytes, byteorder='big')

# print(f'd = {d}')

# print(f'n = {n}')

new\_received\_cipher = received\_cipher.replace(",","").replace(" ","").replace("[","").replace("]","")

print('\nReceived cipher: ')

print(new\_received\_cipher)

val = list(received\_cipher.replace("[","").replace("]","").replace(" ","").split(","))

print("\nDecoded message: ")

print(''.join(str(p) for p in decoder(val)))

break

c.close()



