Name: Shrikant Pawar
Roll No.:57
Assignment No. 07

Title: Implement a client and a server on different computers using python. Perform the communication between these two entities by using RSA cryptosystem.

Client Code:

```
import socket
import time
import string
from diffie_hellman import keyGeneration, sharedKeyGeneration
from des import DES_Algorithm

serverPort = 8001
serverIP = "127.0.0.1"


def keyGenerationForDES(p, q, sharedKey):
    '''
    This is just a function to generate a key of sufficient length
    for the DES Algorithm to work using the shared key formed and the
    global parameters
    '''
    mapping = {}
    for index, letter in enumerate(string.ascii_letters):
        mapping[index] = letter

    val = str(sharedKey * p * q)

    finalKey = []
    for index in range(0, len(val), 2):
        finalKey.append(mapping[int(val[index:index + 1]) % len(mapping)])

    while len(finalKey) < 8:
        finalKey += finalKey

    return "".join(finalKey[:8])


def main():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Establishing the connection
    print("Establishing connection with client")
    client.connect((serverIP, serverPort))
    client.send("Connected!".encode())
    print("Connected!")
    # Getting the global parameters
    p = int(client.recv(4096).decode())
    q = int(client.recv(4096).decode())

    print(f"Large Prime number set to: {p}")
    print(f"Primitive Root is set to: {q}\n")

    # Generating the Public-Private Key Pair
    privateClient, publicClient = keyGeneration(p, q)
    time.sleep(2)

    # Recieving the Public Key of Server
    publicServer = int(client.recv(4096).decode())

    # Sending the Public Key
    client.send(str(publicClient).encode())

    time.sleep(2)

    # Getting the key to be used for DES
    key = int(str(sharedKeyGeneration(publicServer, privateClient, p)), 16)
    DES_key = keyGenerationForDES(p, q, key)

    while True:
        message_to_send = input("You: ")
        print("\n")
        encryptedMessage = DES_Algorithm(text=message_to_send, key=DES_key, encrypt=True).DES()
        client.send(encryptedMessage.encode())

        actual_message = client.recv(4096).decode()
        message = DES_Algorithm(text=actual_message, key=DES_key, encrypt=False).DES()
        if message != "exit":
            print("Peer says: " + message)
            print("The message recieved: {0}".format(actual_message))
            print("\n")
        else:
            client.close()


if __name__ == '__main__':
    main()
```