**LAB 3 TASKS**

**K20-1052 S.M.HASSAN ALI**

**K20-0244 Munir Abbasi**

**K20-1655 Ehtesham Zafar**

**Q1.**

import base64

import json

from Crypto.PublicKey import RSA

from Crypto.Cipher import PKCS1\_OAEP

key\_pair = RSA.generate(1024)

public\_key = key\_pair.publickey().exportKey()

private\_key = key\_pair.exportKey()

key = RSA.importKey(public\_key)

message = b'A message to secure'

cipher = PKCS1\_OAEP.new(key)

ciphertext = cipher.encrypt(message)

data = [

    {'private\_key': private\_key.decode('utf-8'),

     'ciphertext': base64.b64encode(ciphertext).decode('utf-8')

     }

]

try:

    with open('message.json', 'a') as f:

        json.dump(data, f)

        f.write('\n')

except FileNotFoundError:

    with open('message.json', 'w') as f:

        json.dump(data, f)

try:

    with open('message.json', 'r') as f:

        data = json.load(f)

except FileNotFoundError:

    with open('message.json', 'w') as f:

        json.dump(data, f)

dict = []

dict.append(data)

private\_key = dict[0][0]['private\_key'].encode('utf-8')

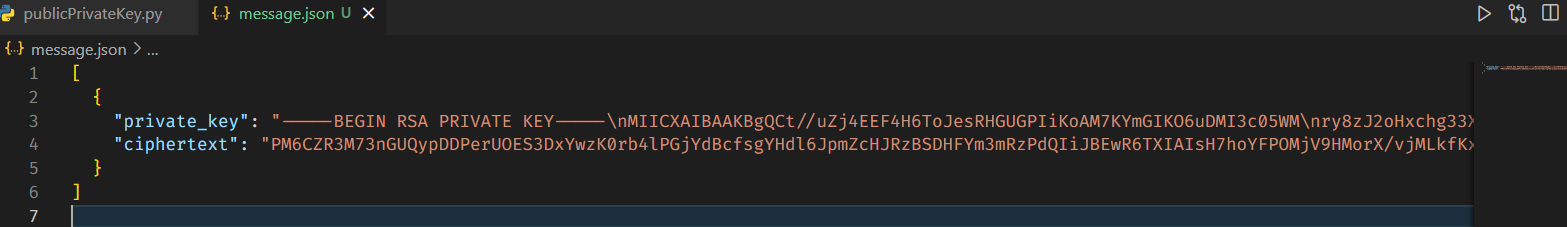
ciphertext = base64.b64decode(dict[0][0]['ciphertext'].encode('utf-8'))

key = RSA.importKey(private\_key)

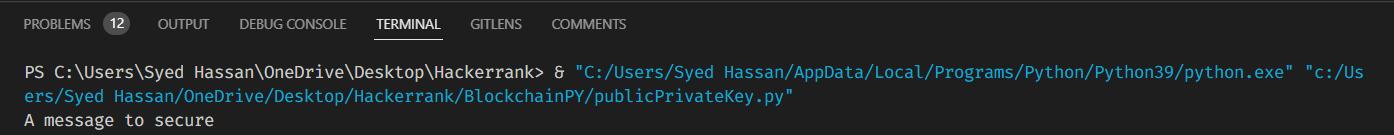
cipher = PKCS1\_OAEP.new(key)

ciphertext = cipher.decrypt(ciphertext)

print(ciphertext.decode('utf-8'))



**Output:**



**Q2.**

#Client Code

import base64

import json

from Crypto.PublicKey import RSA

from Crypto.Cipher import PKCS1\_OAEP

import socket

# Generate RSA key pair

key\_pair = RSA.generate(1024)

public\_key = key\_pair.publickey().exportKey()

private\_key = key\_pair.exportKey()

# Encrypt message using public key

key = RSA.importKey(public\_key)

message = b'This message will be received by server as decrypted cipher text'

cipher = PKCS1\_OAEP.new(key)

ciphertext = cipher.encrypt(message)

# Store private key and ciphertext in text file because of getting some errors of parsing

data = {

    'private\_key': private\_key.decode('utf-8'),

     'ciphertext': base64.b64encode(ciphertext).decode('utf-8')

     }

with open('message.txt', 'w') as f:

f.write(json.dumps(data))

# Send JSON file to server

s = socket.socket()

s.connect(("127.0.0.1", 12357))

with open('message.txt', 'r') as f:

    data = f.read()

    s.sendall(data.encode())

s.close()

# Server code

import base64

import json

from Crypto.PublicKey import RSA

from Crypto.Cipher import PKCS1\_OAEP

import socket

# Receive text file from client

s = socket.socket()

s.bind(("127.0.0.1", 12357))

s.listen(1)

print("[\*] Listening on 0.0.0.0:1234")

conn, addr = s.accept()

with open('message.txt', 'wb') as f:

    while True:

        data = conn.recv(1024)

        data = data.decode()

        if not data:

            break

        f.write(data)

conn.close()

# Read text file and retrieve private key and ciphertext

with open('message.txt', 'r') as f:

    data = json.loads(f.read().strip())

private\_key = data['private\_key'].encode('utf-8')

ciphertext = base64.b64decode(data['ciphertext'].encode('utf-8'))

# Decrypt ciphertext using private key

key = RSA.importKey(private\_key)

cipher = PKCS1\_OAEP.new(key)

ciphertext = cipher.decrypt(ciphertext)

# Print original message

print(ciphertext.decode('utf-8'))

**Output:**

