Socket Programming - Text Messages + Encryption and Decryption

Computer Networks

UE21CS252B

**4th Semester, Academic Year 2022-2023**

|  |  |  |
| --- | --- | --- |
| Name: Atharva Menkudle  Name: Aravilli Atchuta Ram | SRN: PES2UG21CS104  SRN: PES2UG21CS104 | Section B |

Server Side Code:

import socket

def decrypt(msg, key):

    encrypted\_msg = ''

    for c in msg:

        encrypted\_msg += chr(ord(c) ^ key)

    return encrypted\_msg

def encrypt(msg, key):

    decrypted\_msg = ''

    for c in msg:

        decrypted\_msg += chr(ord(c) ^ key)

    return decrypted\_msg

HOST = '192.168.160.189'

PORT = 1552

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

    s.bind((HOST, PORT))

    s.listen()

    print('Server listening on', (HOST, PORT))

    while True:

        conn, addr = s.accept()

        print('Connected by', addr)

        with conn:

            # Receive key from client

            key = int(conn.recv(1024).decode('utf-8'))

            ###print('Received key:', key)

            while True:

                data = conn.recv(1024)

                if not data:

                    break

                msg = data.decode('utf-8')

                print('\*\*\*\*\*\*\*\*\*\*\*')

                print('Received:', msg)

                decrypted\_msg = decrypt(msg, key)

                print('\*\*\*\*\*\*\*\*\*\*\*\*\*')

                print('Decrypted message:', decrypted\_msg)

Client Side Code:

##Uni Directional Flow where client sends and server receives

##Server decrypts and prints the message

import socket

import random

def encrypt(msg, key):

    encrypted\_msg = ''

    for c in msg:

        encrypted\_msg += chr(ord(c) ^ key)

    return encrypted\_msg

def decrypt(msg, key):

    decrypted\_msg = ''

    for c in msg:

        decrypted\_msg += chr(ord(c) ^ key)

    return decrypted\_msg

HOST = '192.168.160.110'

PORT = 1552

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

    s.connect((HOST, PORT))

    # Generate random key

    key = random.randint(0, 255)

    print('Generated key:', key)

    # Send key to server

    s.sendall(str(key).encode('utf-8'))

    while True:

        msg = input('Enter message to send (type "quit" to exit): ')

        if msg == 'quit':

            break

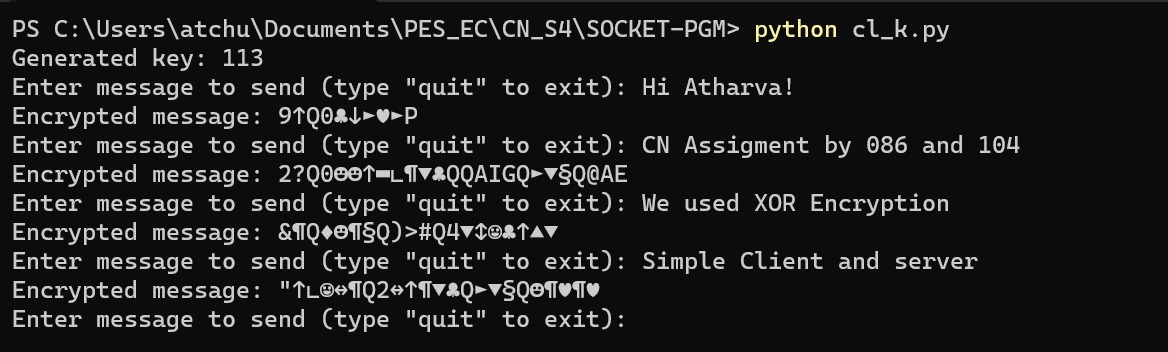
        encrypted\_msg = encrypt(msg, key)

        print('Encrypted message:', encrypted\_msg)

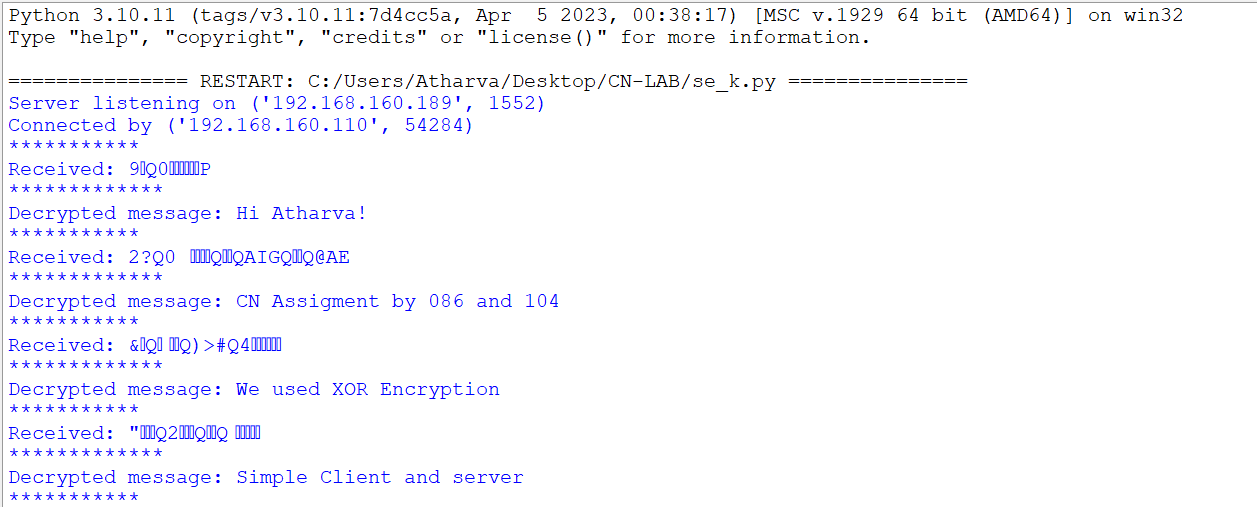
        s.sendall(encrypted\_msg.encode('utf-8'))

Output Screenshots:

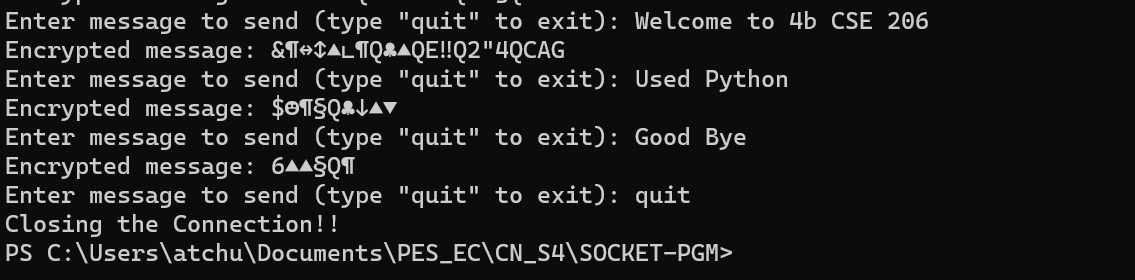
Client side:



Server Side:



Client Side:



Server Side:

