DISTRIBUTED SYSTEMS AND COMPUTING
ITD01
LAB

SUBMITTED BY
SHIV KUMAR
2016UIT2563

Question: Implement to send a secure message over a network to a remote site.

Code : Server

import socket
import pickle
from Crypto.Cipher import AES

HOST = '127.0.0.1' # Standard loopback interface address (localhost) PORT = 65432 # Port to listen on (non-privileged ports are > 1023)

def decrypt(data, key):
data = pickle.loads(data)
cipher = AES.new(key, AES.MODE_EAX, nonce=data[2]) plaintext = cipher.decrypt(data[0])
try:

cipher.verify(data[1])

print("\nThe message is authentic!\n\nDecrypted message:", plaintext.decode(), "\n")

except ValueError:
print("Key incorrect or message corrupted!")

key = input("Enter 16-byte AES decryption key: ").encode()

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s: s.bind((HOST, PORT))
s.listen()
conn, addr = s.accept()

with conn:
print(f"Connection from {addr} has been established.") while True:

data = conn.recv(1024) if not data:

break

decrypt(data, key) conn.sendall(b'ACK')

Client:

import socket
import pickle
from Crypto.Cipher import AES

HOST = '127.0.0.1' # The server's hostname or IP address PORT = 65432 # The port used by the server

def encrypt(message, key):
cipher = AES.new(key, AES.MODE_EAX)
nonce = cipher.nonce
ciphertext, tag = cipher.encrypt_and_digest(message) return [ciphertext, tag, nonce]

```
message = input("Enter message to be encrypted & sent: ").encode() key = input("Enter 16-byte AES encryption key: ").encode()

msg = pickle.dumps(encrypt(message, key))

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s: s.connect((HOST, PORT)) s.sendall(msg)
print('Encrypted message sent successfully!') #s.sendall(b'Hello, world')

data = s.recv(1024) print('Received', data.decode())
```