Blockchain Technology

Assignment – 02

Aim: Program to verify integrity .

Description:

1. Hashlib module:

This module implements a common interface to many different secure hash and message digest algorithms. Included are the FIPS secure hash algorithms SHA1, SHA224, SHA256, SHA384, and SHA512 (defined in FIPS 180-2) as well as RSA’s MD5 algorithm (defined in internet RFC 1321). The terms “secure hash” and “message digest” are interchangeable. Older algorithms were called message digests. The modern term is secure hash.

1. Socket module:

Socket programming shows how to use socket APIs to establish communication links between remote and local processes. The processes that use a socket can reside on the same system or different systems on different networks. Sockets are useful for both stand-alone and network applications.

Code:

Server side:

import socket

import hashlib

a=input("Enter the message:")

sha256hex=hashlib.sha256(a.encode()).hexdigest()

s=socket.socket()

print("\nSocket successfully created")

port=9999

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

print("\nSocket binded to %s"%(port))

s.listen(1)

print("\nSocket is listening......\n")

while True:

    c,addr=s.accept()

    print('Got connection from',addr)

    c.send(a.encode())

    c.send(sha256hex.encode())

    print("Message has been sent")

    c.close()

    break

Output:

Enter the message:hi

Socket successfully created

Socket binded to 9999

Socket is listening......

Got connection from ('127.0.0.1', 49533)

Message has been sent

Client side:

import socket

import hashlib

c=socket.socket()

port=9999

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

a=c.recv(1024).decode()

data=c.recv(1024).decode()

print("\nWelcome to the network")

c.close()

hexcode=hashlib.sha256(a.encode()).hexdigest()

if hexcode==data:

    print("Integrity of data is maintained")

    print(f"message:{a}")

else:

    print("Integrity of data is lost")

Output:

Welcome to the network

Integrity of data is maintained

message:hi