Blockchain Technology

Assignment – 04

Aim: **Program to verify Confidentiality using Cryptography(encryption-decryption).**

Description:

1. Cryptography package:

Cryptography is a package which provides cryptographic recipes and primitives to Python developers. Cryptography includes both high level recipes and low level interfaces to common cryptographic algorithms such as symmetric ciphers, message digests, and key derivation functions.

1. Fernet algorithm:

Fernet guarantees that a message encrypted using it cannot be manipulated or read without the key. Fernet is an implementation of symmetric (also known as “secret key”) authenticated cryptography.

1. Caesar algorithm:

Caesar Cipher Technique is the simple and easy method of encryption technique.

It is simple type of substitution cipher. Each letter of plain text is replaced by a letter with some fixed number of positions down with alphabet.

Code:

from cryptography.fernet import Fernet

def caesarEncrypt(text, numOfShifts):

    result = ""

    # transverse the plain text

    for i in range(len(text)):

      char = text[i]

      # Encrypt uppercase characters in plain text

      if (char.isupper()):

         result += chr((ord(char) + numOfShifts-65) % 26 + 65)

      else:

         result += chr((ord(char) + numOfShifts-97) % 26 + 97)

    return result

def caesarDecrypt(text, numOfShifts):

    result = ""

    # transverse the plain text

    for i in range(len(text)):

      char = text[i]

      # Encrypt uppercase characters in plain text

      if (char.isupper()):

         result += chr((ord(char) - numOfShifts-65) % 26 + 65)

      # Encrypt lowercase characters in plain text

      else:

         result += chr((ord(char) - numOfShifts-97) % 26 + 97)

    return result

text = input("Enter the text: ")

encryptionAlgo = input("Enter encryption algorithm [caesar, fernet] : ")

if encryptionAlgo == 'fernet':

   key = Fernet.generate\_key()

   f = Fernet(key)

   token = f.encrypt(text.encode())

   print(f"\n---> Fernet cipher text: {token.decode()}")

   text = f.decrypt(token)

   print(f"---> Plain text: {text.decode()}")

elif encryptionAlgo == 'caesar':

   numOfShifts = int(input("\nEnter the number shifts: "))

   print(f"\n--->Shift pattern : {numOfShifts}")

   encryptedText = caesarEncrypt(text, numOfShifts)

   print(f"--->Caesar cipher text: {encryptedText}")

   decryptedText = caesarDecrypt(encryptedText, numOfShifts)

   print(f"--->Plain text: {decryptedText}")

Output:

**First output:**

Enter the text: Vishal

Enter encryption algorithm [caesar, fernet] : caesar

Enter the number shifts: 3

--->Shift pattern : 3

--->Caesar cipher text: Ylvkdo

--->Plain text: Vishal

**Second output:**

Enter the text: Vishal

Enter encryption algorithm [caesar, fernet] : fernet

---> Fernet cipher text: gAAAAABi66CHvmzg5DC\_xQxpnlimIhhP13ljsU9S1toPfBPj2xgMkFOHZa1qfatetmVkYHHd\_MADFCydy5qgPW6HSlEhGHdikw==

---> Plain text: Vishal