CYBER *project* SECURITY

Steganography for Secure File Transmission using LSB algorithm and MD5Sum Hashing Method

Presentation by GROUP 50

CYBER SECURITY

Meet our team

Member 1
Details

Member 2
Details

Member 3
Details

Member 4
Details

Guided by: Name of Teacher

Assistant Professor Senior

Index

WHAT IS FILE ENCRYPTION?

- Companies work with files every day.
- Cybercrime affects about 32% of companies every year .
- To combat this risk, you need powerful security features that integrate well with your existing platforms and fit into your company budget.
- File encryption is a way of encoding files, including the sensitive data they contain, to send them securely.
- The encoding prevents unauthorized access and tampering by malicious methods.

PRESENT ISSUE

- A used public channel can be deliberately monitored by an intruder during the transmission process, which is an attempt to try and carry out a random assault.
- Our method makes use of encryption algorithm to encrypt the file .
- This helps to drastically increase the scope of establishing a secure communication line.

OBJECTIVE

To introduce a new mode of secure communication over the internet by using a steganographic and personal encryption methods that integrates an innocuous-looking image which is password protected and will be used to hide the text file containing the sensitive data.

Literature Review

STEGANOS means to hide something

GRAYPFIA means writing

This project aims to explain the basic steps of file security using image as a cover and encryption for the hidden file

Literature Review (Cont..)

- ENCRYPTION IS ONE OF THE MOST EFFECTIVE APPROACHES TO ACHIEVING DATA SECURITY AND PRIVACY.
- THE ORIGINAL INFORMATION IS RECOVERED ONLY THROUGH USING A PASSWORD, WHICH IS KNOWN AS THE DECRYPTION PROCESS.
- THE OBJECTIVE OF THE ENCRYPTION IS TO SECURE OR PROTECT DATA FROM UNAUTHORIZED ACCESS IN TERMS OF VIEWING OR MODIFYING THE DATA.
- ENCRYPTION CAN BE IMPLEMENTED BY USING SOME SUBSTITUTE TECHNIQUE, SHIFTING TECHNIQUE, OR MATHEMATICAL OPERATIONS.

- SEVERAL SYMMETRIC KEY BASE ALGORITHMS HAVE BEEN DEVELOPED IN THE PAST YEAR. IN THE PAPER AN EFFICIENT RELIABLE SYMMETRIC KEY-BASED ALGORITHM TO ENCRYPT AND DECRYPT THE TEXT DATA HAS BEEN PROPOSED.

- THE METHOD USES 8 BIT CODE VALUE OF THE ALPHABET AND PERFORM SOME SIMPLE CALCULATION LIKE LOGICAL NOT AND SIMPLE BINARY DIVISION TO PRODUCE. THE PROPOSED METHOD IS EASY TO UNDERSTAND AND EASY TO IMPLEMENT.

Proposed Work

- The conception is that it's a cover object(image) that's used to hide the initial file containing sensitive data that needs to be encrypted.

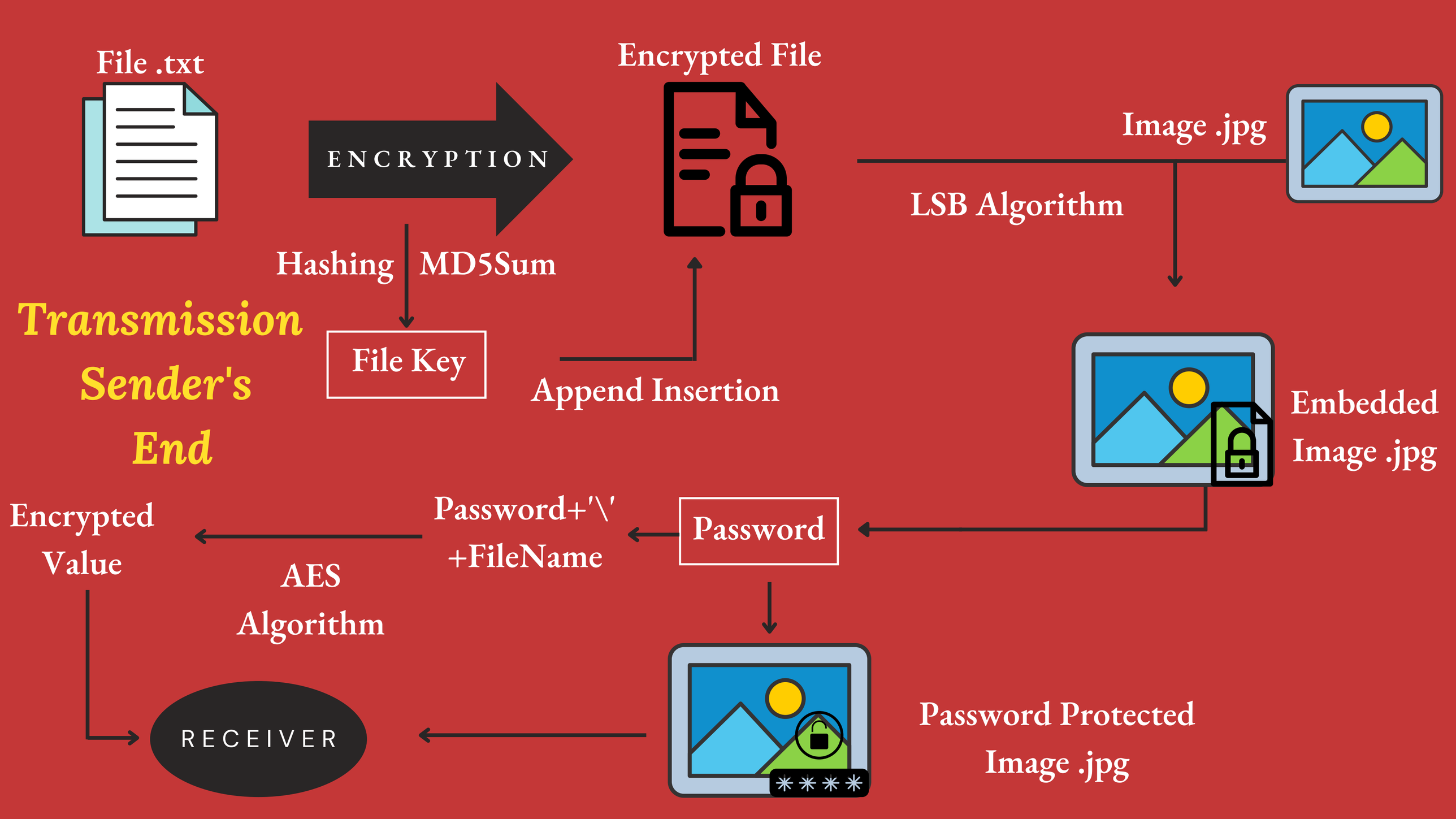
- The image file will be protected using a password and the file within the image file will be encrypted, thereby providing two levels of protection.

- One algorithm will be used for encrypting the file contents from plain text to a unique encrypted value. So, if a third-party member were to find the contents in the text file it would still be uninterpretable to the intruder.

- A file key is generated using MD5 hashing and it is added to the end of file using the append insertion method.

Proposed Work (Cont...)

- The encrypted text file is embedded into the image file using LSB Algorithm.
- The image will be password protected and to identify the password for the respective image , filename is added to the password string after a backslash('\').
- The password generated is encrypted using AES Algorithm.
- The password protected image and the encrypted password is sent to the receiver
- The receiver will decode the password and open the image file to view the text file in it.


File .txt
ENCRYPTION
Encrypted File
Image .jpg
LSB Algorithm
Hashing
MD5Sum
Transmission
Sender's
End
File Key
Append Insertion
Embedded
Image .jpg
Encrypted
Value
Password+'\'
+FileName
Password
AES
Algorithm
RECEIVER
Password Protected
Image .jpg

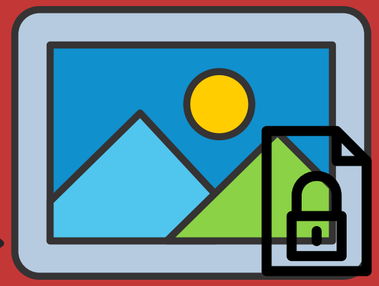
FileName

Encrypted Value

Decryption
of AES
Algorithm

Password+'\'
+FileName

Password



Embedded
Image .jpg

Decryption of LSB
Algorithm

Transmission Receiver's End

Encrypted File

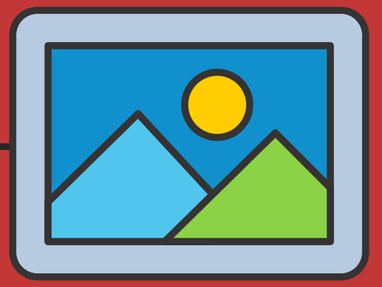


Image .jpg

DECRYPTION

File .txt

Hashing

MD5Sum

Removing the file key

File Key

File Key

MD5Sum

- Md5sum is a hashing method that calculates and verifies 128-bit hash.
- Md5sum is used to verify the integrity of files, as virtually any change to a file will cause its MD5 hash to change.
- Most commonly, md5sum is used to verify that a file has not changed as a result of a faulty file transfer or non-malicious meddling.

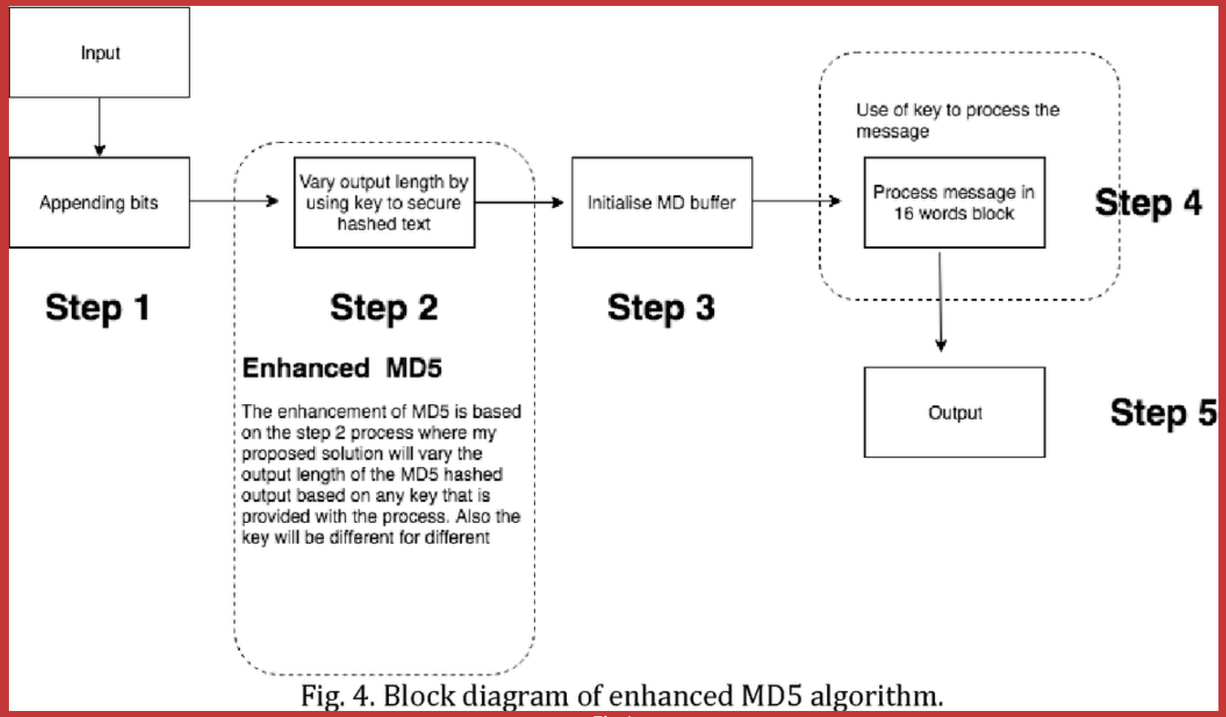


Fig. 4. Block diagram of enhanced MD5 algorithm.

Fig 1
Courtesy:https://imgs.search.brave.com/xFeBH1zztf9ylS_68X2NvT7kyNwLzM0QdxEHzmjiELQ/rs:fit:474:225:1/g:ce/aHR0cHM6Ly90c2Uy/Lm1tLmJpbmcubmV0/L3RoP2lkPU9JUC40/cEVGbGZHbDRfWjJm/eko0U0MtbHJ3SGFI/YSZwaWQ9QXBp

LSB Algorithm

- Least significant bit (LSB) insertion is a common and simple method to embed data in an image file.

- This technique operate well for image steganography. For hiding data within the images, the LSB (Least Significant Byte) approach is generally used.

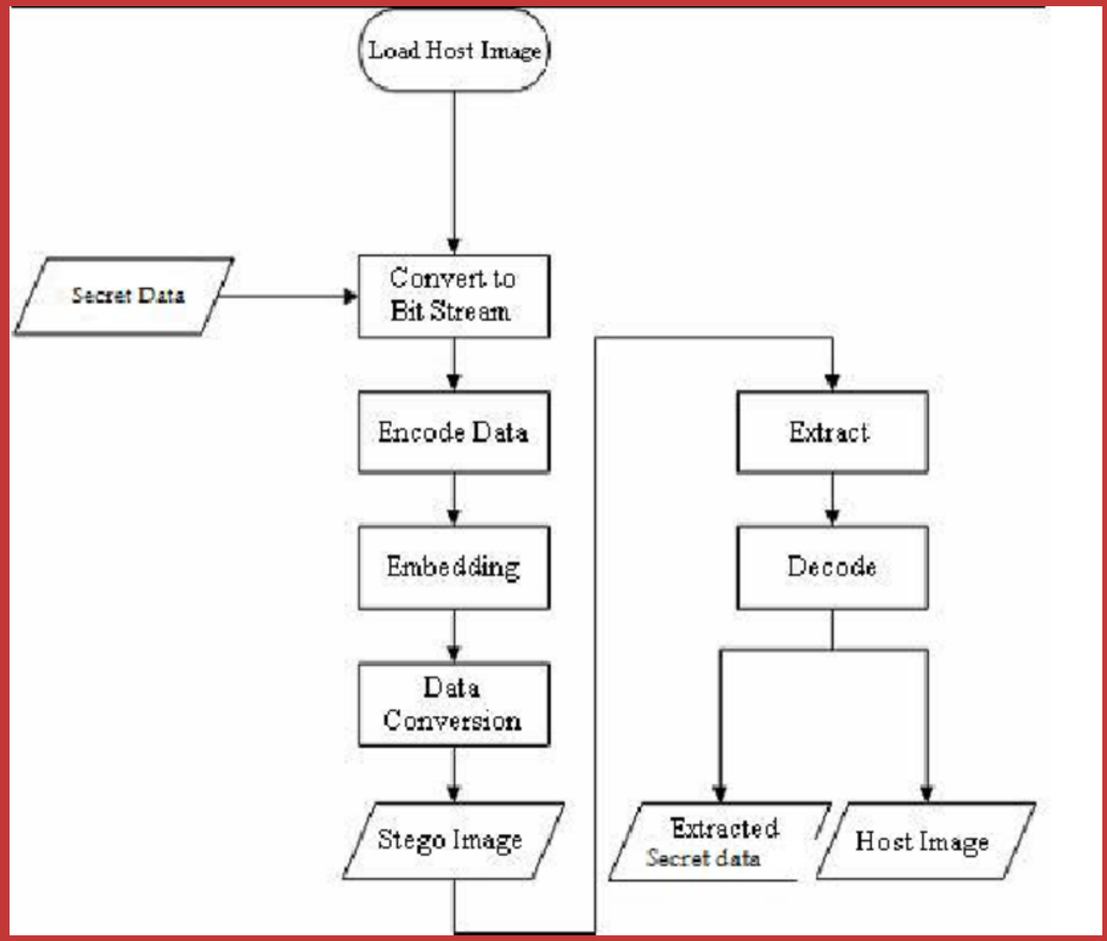


Fig 1

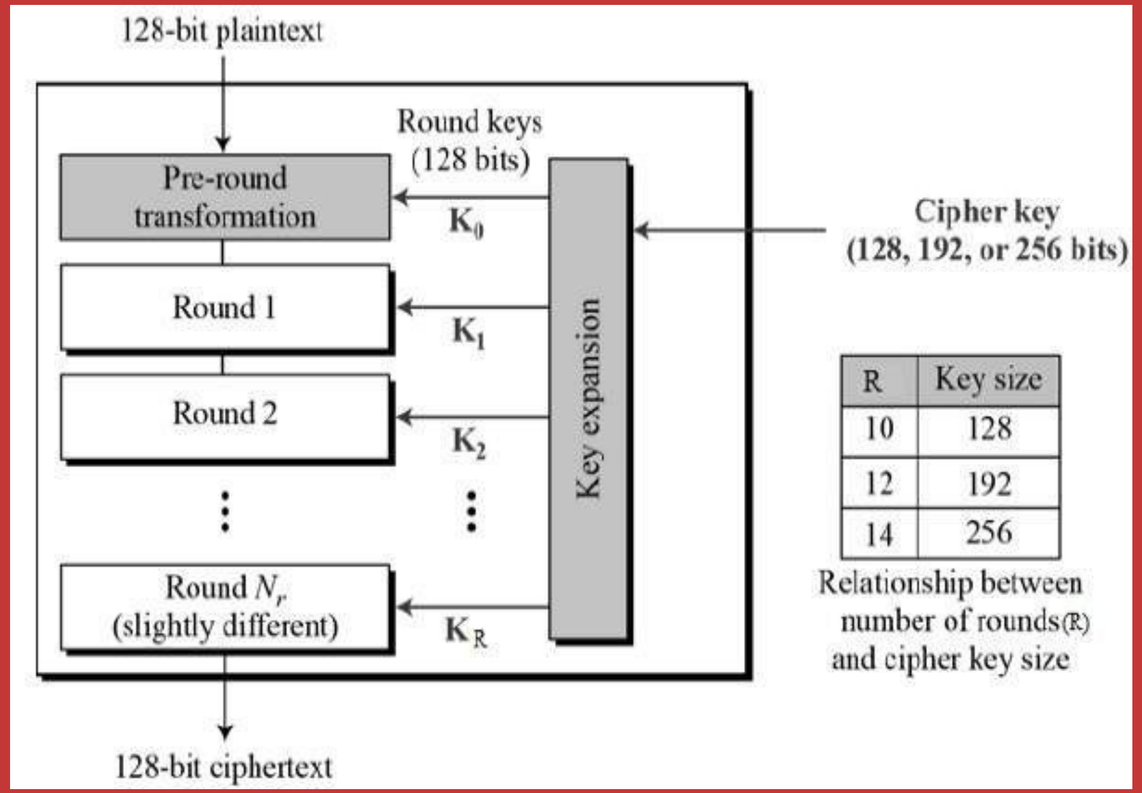
Courtesy:https://imgs.search.brave.com/xFeBH1zztf9yIS_68X2NvT7kyNwLzM0QdxEHzmjiELQ/rs:fit:474:225:1/g:ce/aHR0cHM6Ly90c2Uy/Lm1tLmJpbmcubmV0/L3RoP2lkPU9JUC40/cEVGbGZHbDRfWjJm/ekoOUOMtbHJ3SGFI/YSZwaWQ9QXBp

- LSB-STEGANOGRAPHY IS A STEGANOGRAPHY TECHNIQUE IN WHICH WE HIDE MESSAGES INSIDE AN IMAGE BY REPLACING LEAST SIGNIFICANT BIT OF IMAGE WITH THE BITS OF MESSAGE TO BE HIDDEN.

- BY MODIFYING ONLY THE FIRST MOST RIGHT BIT OF AN IMAGE WE CAN INSERT OUR SECRET MESSAGE AND IT ALSO MAKE THE PICTURE UNNOTICEABLE, BUT IF OUR MESSAGE IS TOO LARGE IT WILL START MODIFYING THE SECOND RIGHT MOST BIT AND SO ON .

AES Algortithm

- It is based on a ‘substitution–permutation network’.
- It comprises a series of linked operations, some of which involve replacing inputs with specific outputs (substitutions) and others shuffling bits around (permutations).
- AES performs all its computations on bytes rather than bits.



R	Key size
10	128
12	192
14	256

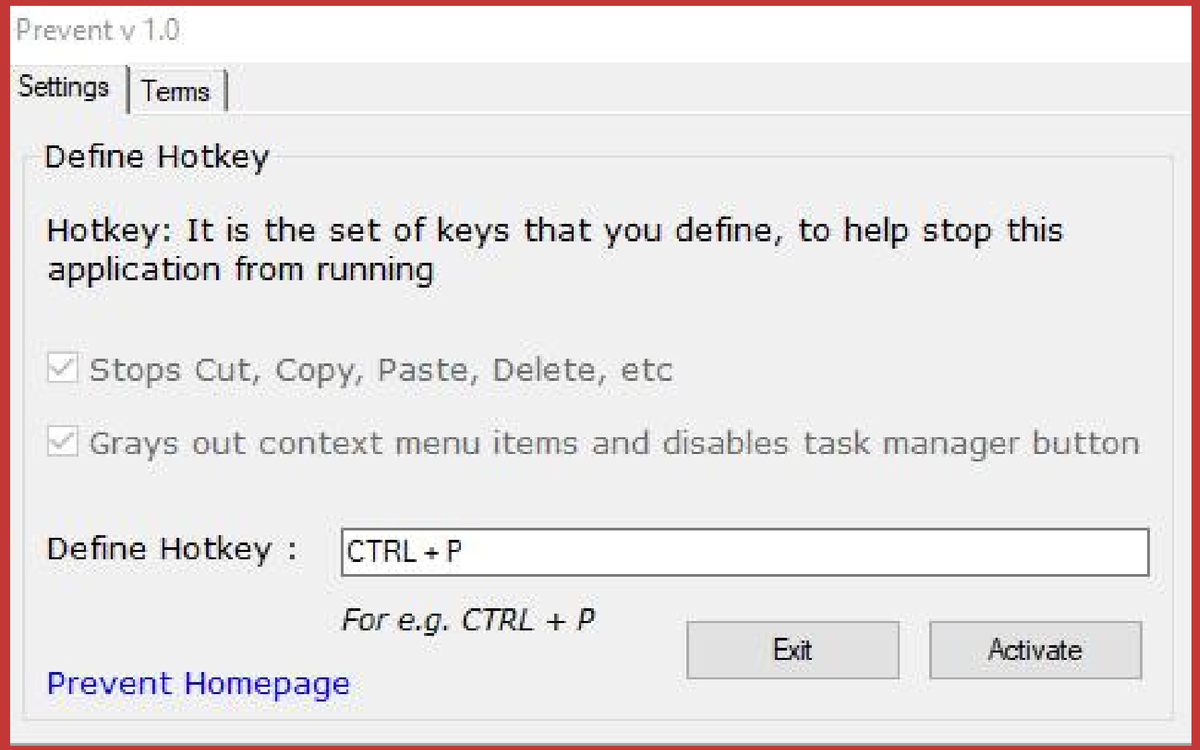
Relationship between number of rounds(R) and cipher key size

Fig 2

Courtesy:https://imgs.search.brave.com/xFeBH1zztf9yIS_68X2NvT7kyNwLzM0QdxEHzmjiELQ/rs:fit:474:225:1/g:ce/aHR0cHM6Ly90c2Uy/Lm1tLmJpbmcubmV0/L3RoP2lkPU9JUC40/cEVGbGZHbDRfWjJm/eko0U0MtbHJ3SGFI/YSZwaWQ9QXBp

How to prevent the file from being renamed?

- A user is able to delete or rename your files only because they get the option to do so in File Explorer.
- Prevent is a small application that allows you to disable certain options in File Explorer on your computer.
- It lets you disable options like Rename, Delete, Cut, and Copy so no one can touch or modify your chosen files.

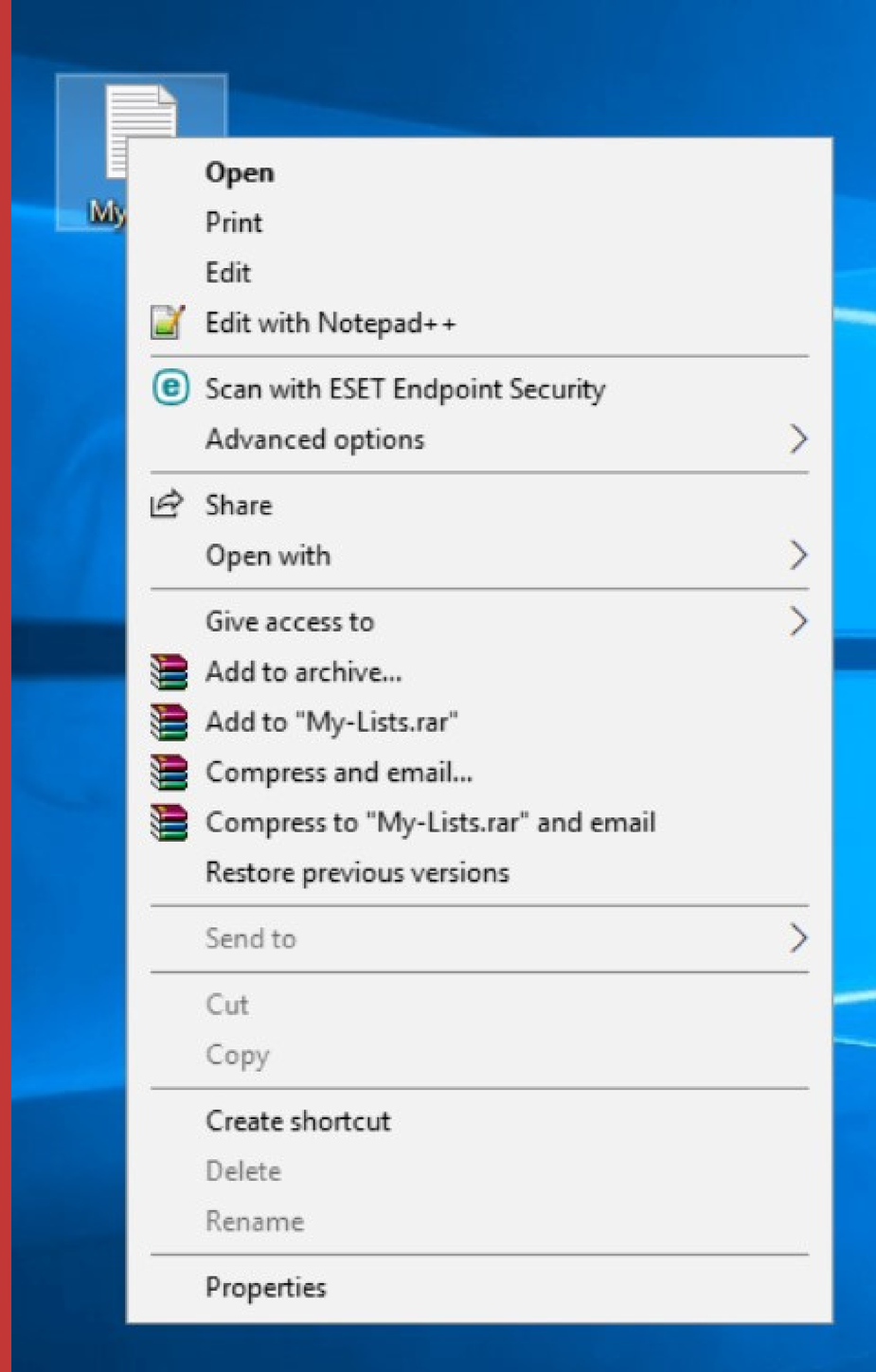


- Download, install, and launch the Prevent app on your Windows PC.

- When the app launches, you'll see that there's only one option you can configure.

- It's called Define Hotkey and it allows you to specify a keyboard shortcut that stops the app from running. Use any of the available keyboard shortcuts and then click on Activate.

- The app will start running, and when you right-click on your file, you'll find that the options mentioned above are grayed out. You can't click or use them. The app disables the physical buttons for those actions as well.



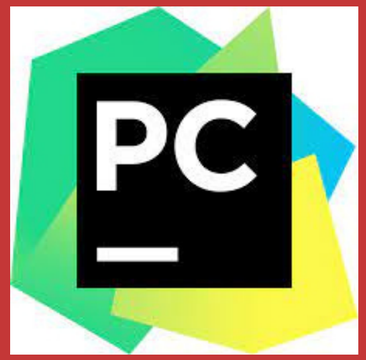
real time usage

Real time usage of File Encryption

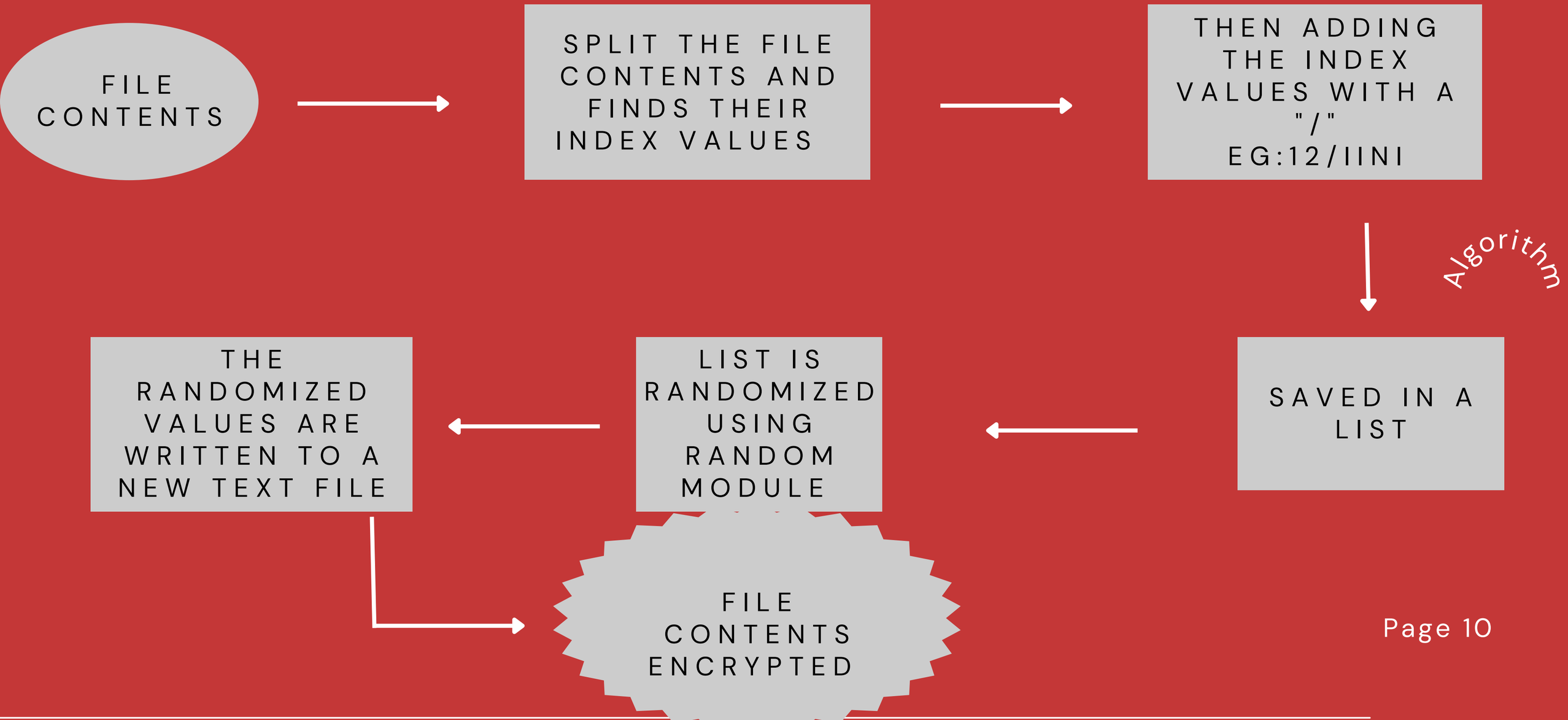
- File encryption and decryption has applications in internet communication, multimedia systems, medical imaging, telemedicine, military communication.
- Since these files may carry highly confidential information.And to provide such security and privacy to the user, File encryption is essential to protect from unauthorized user access.
- This file encryption allows transfers of data from one country to another from agencys such as RAW, NIA and FBI.

Softwares

- Python
- Kali Linux
- Visual Studio Code
- PyCharm
- IDLE Python
- Prevent



Algorithm

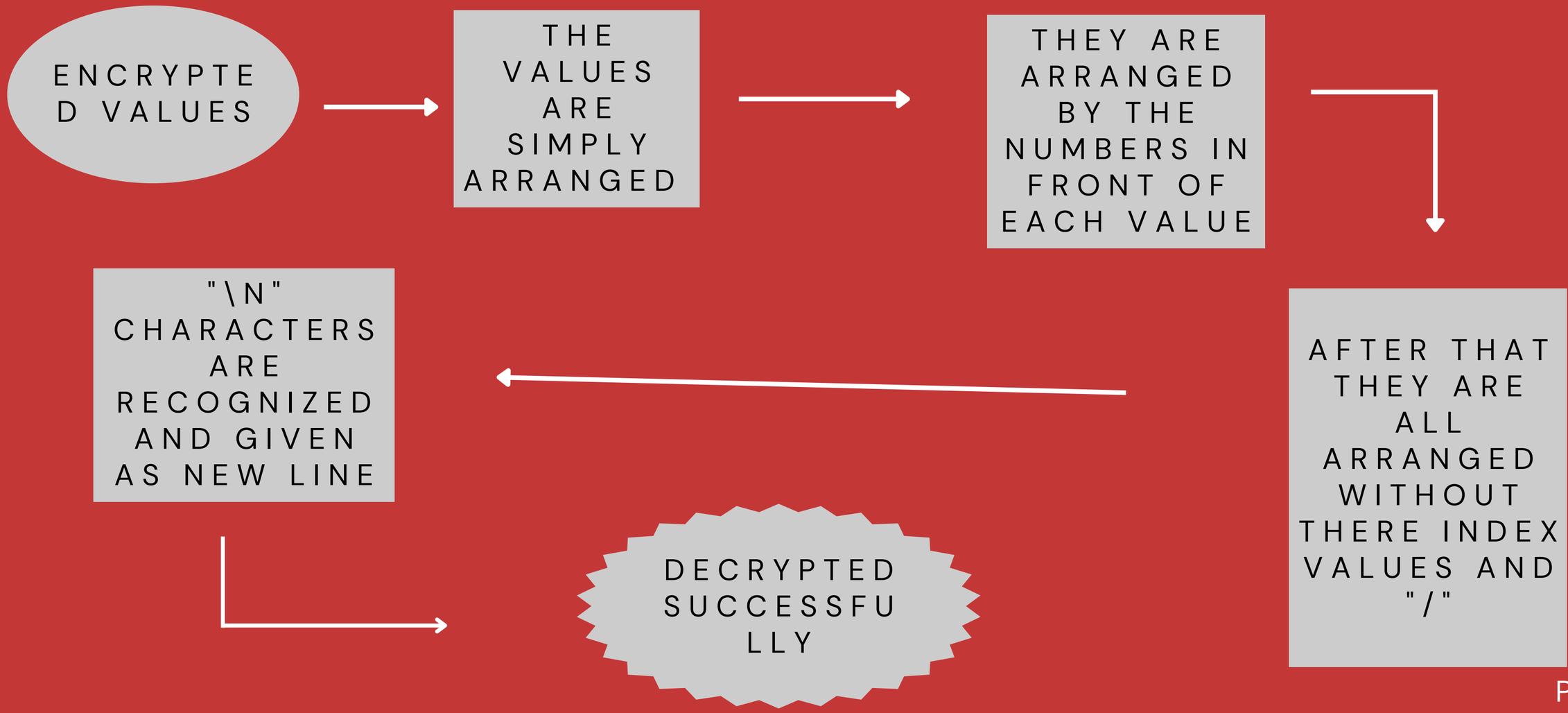

FILE CONTENTS
SPLIT THE FILE CONTENTS AND FINDS THEIR INDEX VALUES
THEN ADDING THE INDEX VALUES WITH A "/" EG:12/IINI
Algorithm
SAVED IN A LIST
LIST IS RANDOMIZED USING RANDOM MODULE
THE RANDOMIZED VALUES ARE WRITTEN TO A NEW TEXT FILE
FILE CONTENTS ENCRYPTED

Python Code 1

```
#encrytion code
import random
file1 = open("t.txt","r")
a=file1.read()
x=0
#a='ben 123_'
j=['1','2','3','4','5','6','7','8','9','0']
o=''#indexing'
e=[]#split with index numbers
r=[]
for i in a:
    q=str(i)
    s=str(x)
    x=x+1
    p=s+"/"+q
    e.append(p)
    r.append(p)

random.shuffle(e)
print(str(e))
```

This is a basic Encrytion algorithm in which we convert the file contents into random numbers and letters.

Algorithm



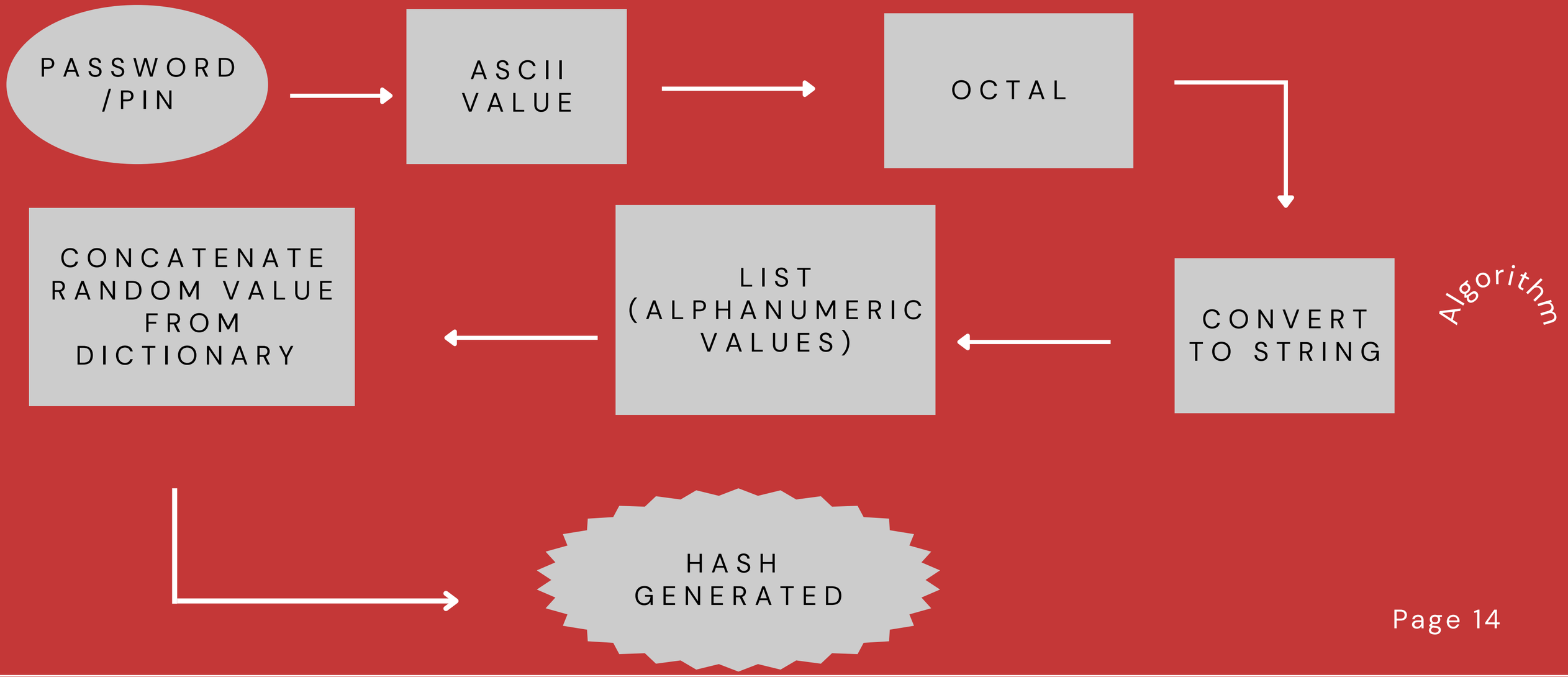
Algorithm

DECRYPTION Code


```
#decrytion code

a=eval (input("enter ur encryted values"))

b=len(a)

c=[]
f=[]
z=[]
r=''
for i in a:
    s=i.split("/")
    c.append(s)
    f.append(s[0])

l=len(f)
for i in range(l):
    for j in c:
        if j[0]==str(i):
            z.append(j[1])


for i in z:
    if i=="\n":
        print("\n")
    else:
        print(i,end='')
```

In this decryption algorithm, the encrypted contents are converted back to the normal text in their original forms using simple techniques.

Algorithm 3



Algorithm

Python Code 3

This hashing algorithm converts the password into alphanumeric values and octal numbers.

initial3.py - C:\Users\agent\OneDrive\Desktop\Project\initial3.py (3.10.5)

File Edit Format Run Options Window Help

```
from LSBSteg1 import *
from cryptography.fernet import Fernet
import cv2
def encode():
    fi=input("Enter the name of file to be Encrypted:>> ")
    global e
    global p
    with open(fi,'r+') as f:
        rea=f.read()
    def encrypt():
        #encrytion code
        import random
        a=rea
        x=0
        #a='ben 123_'
        j=['1','2','3','4','5','6','7','8','9','0']
        o=''#indexing'
        global e
        e=[]#split with index numbers
        r=[]
        for i in a:
            q=str(i)
            s=str(x)
            x=x+1
            p=s+"/"+q
            e.append(p)
            r.append(p)

        random.shuffle(e)
        #print(str(e))
    def writefile():
        global w
        w=input("Enter the name of File:- ")
        with open(w,'w+') as x:
            x.write(str(e))
            #y=x.read()
            #print(y)
    def hash():
        # Python program to find MD5 hash value of a file
        import hashlib

        filename = fi
        with open(filename,"rb") as f:
            bytes = f.read() # read file as bytes
            readable_hash = hashlib.md5(bytes).hexdigest();
                #print(readable_hash)
            return readable_hash
```

Hashing and Appending

initial3.py - C:\Users\agent\OneDrive\Desktop\Project\initial3.py (3.10.5)

File Edit Format Run Options Window Help

```
    global h
    h=hash()
    def append():
        with open(w,'a') as x:
            ha='\n'+h
            f=x.write(ha)
            x.close()
    def encode2():
    #encoding
        with open(w,'r') as f:
            rea1=f.read()
        x=input("Enter Image Name:- ")
        steg = LSBSteg(cv2.imread(x))
        #print(type(steg))
        img_encoded = steg.encode_text(rea1)
        global y
        y=input("Enter New Image Name:- ")
        cv2.imwrite(y, img_encoded)
    global p
    global fernet
    def password():
        global p
        p=input("Enter Password:- ")
        filename = y
        finalpassword = p+"|"+filename


        # generate a key for encryption and decryption
        # You can use fernet to generate
        # the key or use random key generator
        # here I'm using fernet to generate key

        key = Fernet.generate_key()

        # Instance the Fernet class with the key

        global fernet
        fernet = Fernet(key)

        # then use the Fernet class instance
        # to encrypt the string string must
        # be encoded to byte string before encryption
        encMessage = fernet.encrypt(finalpassword.encode())

        print("Original String: ", finalpassword)
        print("Encrypted String: ", encMessage)
    hash()
```

LSB and AES

initial3.py - C:\Users\agent\OneDrive\Desktop\Project\initial3.py (3.10.5)

File Edit Format Run Options Window Help

```
    encrypt()
    writefile()
    append()
    encode2()
    password()
#print('&&',e)
def decode():
    fid='file.txt'
    def decrypt():
    #decrytion code
        with open(fid,'r+') as ab:
            r=ab.read()
            #print('%%%%',r)
        a=e
        #print('$$',a)
        b=len(a)
        c=[]
        f=[]
        z=[]
        r=''
        for i in a:
            s=i.split("/")
            c.append(s)
            f.append(s[0])
        #print(f)
        l=len(f)
        for i in range(l):
            for j in c:
                if j[0]==str(i):
                    z.append(j[1])

        #print(z)
        with open(fid,'w+') as de:
            de.truncate()
            for i in z:
                if i=="\n":
                    de.write("\n")
                else:
                    de.write(i)
        #for i in c:
         #   d=i[1]
          #  f.append(d)

        #for i in f:
         #   r=r+i
        #print(r)
    def removelastline1():
```

Decryption Process

initial3.py - C:\Users\agent\OneDrive\Desktop\Project\initial3.py (3.10.5)

File Edit Format Run Options Window Help

```
    def removelastline1():
    #remove last line from a text line in python
        fd=open(fid,"r")
        d=fd.read()
        #print('*****',d)
        fd.close()
        m=d.split("\n")
        #print('^^^^',m)
        global ha
        ha=m[-1]
        s="\n".join(m[:-1])
        #print(s)
        fd=open(fid,"w+")
        for i in range(len(s)):
            fd.write(s[i])
        fd.close()
    def hash1():
        # Python program to find MD5 hash value of a file
        import hashlib

        filename = 'sample.txt'
        with open(filename,"rb") as f:
            bytes = f.read() # read file as bytes
            readable_hash = hashlib.md5(bytes).hexdigest();
                #print(readable_hash)
            return readable_hash
    global h1
    h1=hash1()
    def verify():
        if ha==h1:
            print("\n \nHash Verified")
        else:
            print("Hash Not verified")
    def decode2():
    #decoding
        im = cv2.imread("new.png")
        steg = LSBSteg(im)
        global t
        t=steg.decode_text()
        #print("Text value:",t)
        with open(fid,'w+') as f:
            f.write(t)
    with open('new.txt','r+') as f:
        r=f.read()
    def password1():
        import matplotlib.pyplot as plt
        import matplotlib.image as mpimg
```

Verifying the hash

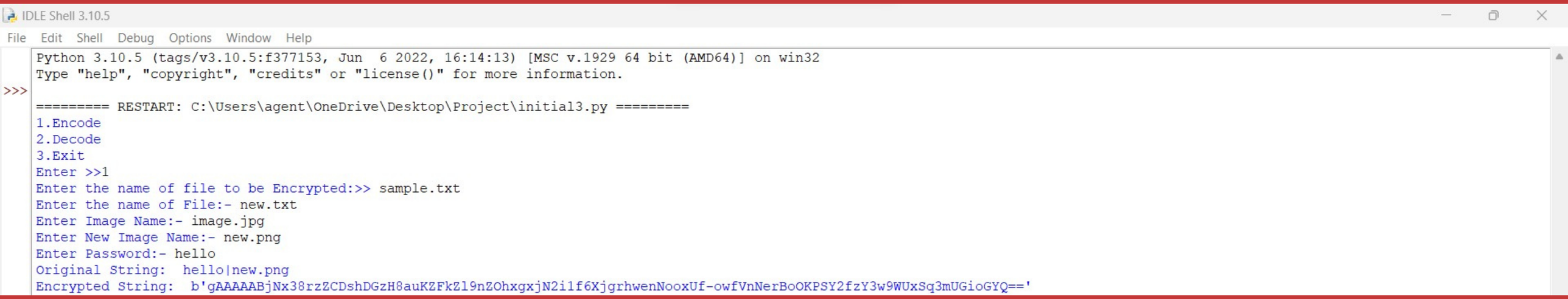
initial3.py - C:\Users\agent\OneDrive\Desktop\Project\initial3.py (3.10.5)

File Edit Format Run Options Window Help

```
        im = cv2.imread("new.png")
        steg = LSBSteg(im)
        global t
        t=steg.decode_text()
        #print("Text value:",t)
        with open(fid,'w+') as f:
            f.write(t)
    with open('new.txt','r+') as f:
        r=f.read()
    def password1():
        import matplotlib.pyplot as plt
        import matplotlib.image as mpimg
        setpassword = p
        userpassword = input("Enter Password:")
        if setpassword==userpassword:
            img = mpimg.imread('image.jpg')
            imgplot = plt.imshow(img)
            plt.show()
            decode2()
            removelastline1()
            decrypt()
            hash1()
            verify()
        else:
            print("Invalid Password")
    def aesdecrypt():
        f=input("Enter value:")
        decMessage = fernet.decrypt(f).decode()
        d=decMessage.split("|")
        print("decrypted string: ",decMessage)
        print("Password ",d[0])
        print("Filename ",d[1])
    aesdecrypt()
    password1()
def main():
    while True:
        a=input("Enter >>")
        if a=='1':
            encode()
        elif a=='2':
            decode()
        elif a=='0':
            break
        else:
            print("Wrong value")
main()
```

Decrypting LSB

Encryption Process



Decryption Process - I

IDLE Shell 3.10.5

File Edit Shell Debug Options Window Help

```
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Users\agent\OneDrive\Desktop\Project\initial3.py =========
1.Encode
2.Decode
3.Exit
Enter >>1
Enter the name of file to be Encrypted:>> sample.txt
Enter the name of File:- new.txt
Enter Image Name:- image.jpg
Enter New Image Name:- new.png
Enter Password:- hello
Original String:  hello|new.png
Encrypted String:  b'gAAAAABjNx38rzZCDshDGzH8auKZFkZl9nZOhxgxjN2i1f6XjgrhwenNooxUf-
1.Encode
2.Decode
3.Exit
Enter >>2
Enter value:gAAAAABjNx38rzZCDshDGzH8auKZFkZl9nZOhxgxjN2i1f6XjgrhwenNooxUf-owfVnNerB
decrypted string:  hello|new.png
Password  hello
Filename  new.png
Enter Password:hello
```

Figure 1

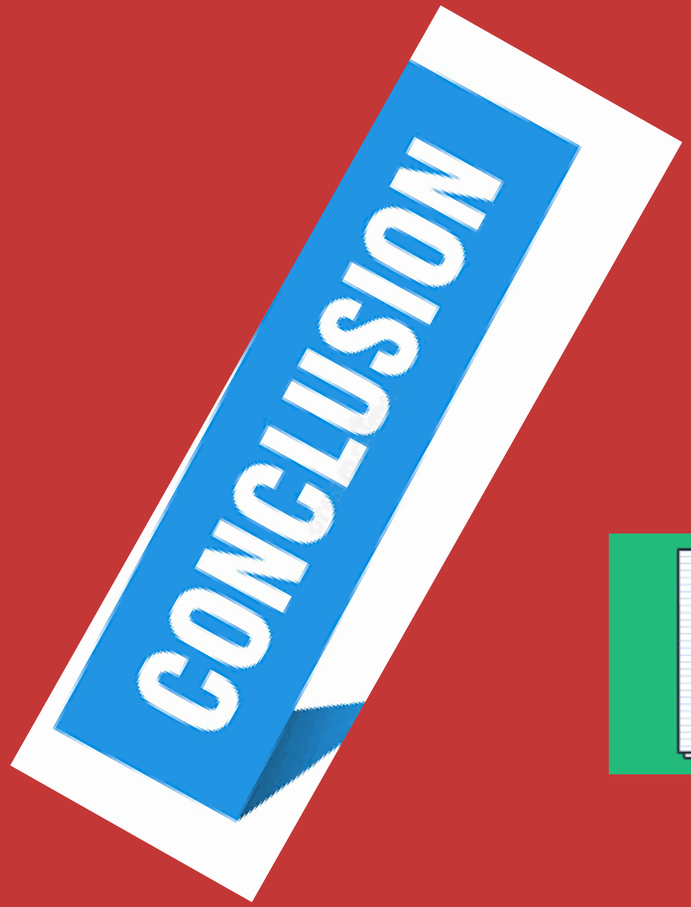
Decryption Process - II

```
1.Encode
2.Decode
3.Exit
Enter >>2
Enter value:gAAAAABjNx38rzZCDshDGzH8auKZFkZl9nZOhxgxjN2i1f6XjgrhwenNooxUf-owfVnNerBoOKPSY2fzY3w9WUxSq3mUGioGYQ==
decrypted string:  hello|new.png
Password  hello
Filename  new.png
Enter Password:hello
******************


Hash Verified
******************
```

Conclusion

conclusion

Thus, the project entitled "Steganography for Secure File Transmission using LSB algorithm and MD5Sum Hashing Method."was completed. By the end of the project, we have gained valuable skills including a grounding in how to interact with the operating system, file handling in python, optimizing algorithms, calculating the efficiencies, and learning how to form and manipulate images.



References 

- https://link.springer.com/chapter/10.1007/978-3-642-27948-5_23
- https://www.engpaper.net/data-encryption.html
- https://www.researchgate.net/publication/320149845_A_Secure_and_Fast_Approach_for_Encryption_and_Decryption_of_Message_Communication
- https://stackoverflow.com/questions/18569784/python-password-protection
- https://roytuts.com/how-to-encrypt-pdf-as-password-protected-file-in-python/

THANK YOU

presentation by

Group 50