IMPLEMENTATION OF IMAGE ENCRYPTION AND DECRYPTION

A PROJECT REPORT

Submitted by
SANJAY.V
(201061022)
&
VIKNESH.K
(201061027)

in partial fulfillment for the award of the degree
of
BACHELOR OF TECHNOLOGY
In
INFORMATION TECHNOLOGY

IFET COLLEGE OF ENGINEERING
(AN AUTONOMOUS INSTITUTION)
VILLUPURAM 605108
NOVEMBER 2021

IFET COLLEGE OF ENGINEERING
(An Autonomous Institution)

BONAFIDE CERTIFICATE

Certified that this report titled "IMPLEMENTATION OF IMAGE ENCRYPTION AND DECRYPTION" is the bonafide work of **SANJAY.V (201061022) & VIKNESH.K (201061027)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Mrs.J.JAYACHITRA M.E., (Ph.D).,
HEAD OF THE DEPARTMENT
Associate Professor,
Department of IT,
IFET College of Engineering,
Villupuram - 605108

Mrs.J.SUGANTHI M.Tech.,
SUPERVISOR
Assistant Professor,
Department of IT,
IFET College of Engineering,
Villupuram - 605108

CERTIFICATE OF EVALUTION

College name : IFET College of Engineering, Villupuram.

Branch : B.Tech– IT

Month & Year : November 2021

Sub. Code & Name : 19UITMP301/MINI PROJECT-I

Name of the Student	Register Number	Title of the Project	Name of the Supervisor with Designation
V.SANJAY & K.VIKNESH	**201061022 & 201061027**	IMPLEMENTATION OF IMAGE ENCRYPTION AND DECRYPTION	Mrs.J.SUGANTHI Assistant Professor

The report for the Mini project-I submitted for the fulfillment of the award of the degree of Bachelor of Technology in Information Technology of IFET College of Engineering (Autonomous), permanently affiliated to Anna University was evaluated and confirmed to be the work done by the above student.

SUPERVISOR HEAD OF THE DEPARTMENT

Submitted for the End Semester examination held on _____

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We thank the almighty, for the blessings that have been showered upon me to bring forth the success of the project. We would like to express my sincere gratitude to our Chairman **Mr.K.V.Raja**, our Secretary **Mr.K.Shivram Alva** for providing us with an excellent infrastructure and necessary resources to carry out this project and we extend our gratitude to our principal **Dr.G.Mahendran**, for his constant support to our work.

We also take this opportunity to express our sincere thanks to our Vice-Principal **Dr.S.Matilda** who has provided all the needful help in executing the project successfully.

We wish to express our thanks to our Head of the Department, **Mrs.J.Jayachithra**, Associate Professor for her persistent encouragement and support to complete this project. We express my heartfelt gratitude to my guide **Mrs.J.Suganthi,** Assistant professor Department of Information Technology for her priceless guidance and motivation which helped us to bring this project to a perfect shape.

And we thank our faculty advisor **Mrs.C.Kavitha,** Assistant Professor, Department of Information Technology who encouraged us in each and every step of this project to complete it successfully.

We also thank our lab technicians and all the staff members of our department for their support and assistance.

Last but not the least, we wholeheartedly thanks our family and friends for their moral support in tough times and their constructive criticism which made me succeed in our work.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

DES	**- D**ata **E**ncryption **S**tandard
AES	**-A**dvanced **E**ncryption **S**tandard
PNG	**- P**ortable **N**etwork **G**raphics
JPEG	**- J**oint **P**hotographic **E**xperts **G**roup
RSA	**-R**ivest **S**hamir **A**dleman
IDEA	**- I**nternational **D**ata **E**ncryption **A**lgorithm

LIST OF FIGURES

ABSTRACT

In today's world almost all digital services like internet communication, medical and military imaging systems, multimedia system needs a high-level security. There is a need for security level in order to safely store and transmit digital images containing critical information. Many methods are used to provide the security, integrity, confidentiality and to prevent unauthorized access of sensitive information such as Cryptography. In this project we use Python program in order to hide image. Such Encryption technique helps to avoid intrusion attacks. To solve this problem, we are using AES algorithm for encrypting and decrypting image. This encrypted data is unreadable to the unauthorized user. This encrypted data can be sent over network and can be decrypted using AES at the receiving end. Hence it ensures secure transmission of image.

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Now a day the use of devices such as computer, mobile and many more other devices for communication as well as for data storage and transmission has increased. As a result, there is increase in number of users. Along with these users, there is also increase in number of unauthorized users which are trying to access a data by unfair means. This arises the problem of data security. Images are sent over an insecure transmission channel from different sources, some image data contains secret data, some images itself are highly confidential hence, securing them from any attack is essentially required. To solve this problem, we are using AES algorithm for encrypting and decrypting image. This encrypted data is unreadable to the unauthorized user. This encrypted data can be sent over network and can be decrypted using AES at the receiving end. Hence it ensures secure transmission of image.

1.2 DOMAIN OVERVIEW

Image encryption can be defined in such a way that it is the process of encoding secret image with the help of some encryption algorithm in such a way that unauthorized users can't access it. It has applications in internet communication, multimedia systems, medical and military imaging systems. Several cryptographic algorithms have been proposed up to now like AES, DES, RSA, IDEA etc.

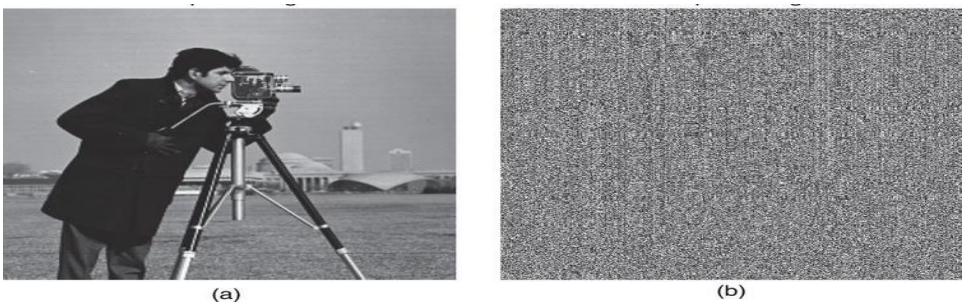


Fig.1: Sample Encrypted Image

CHAPTER 2
EXISTING SYSTEM

2.1 EXISTING SYSTEM

Data Encryption Standard (DES) is a block cipher algorithm that takes plain text in blocks of 64 bits and converts them to ciphertext using keys of 48 bits. DES was once the go-to, symmetric key algorithm for the encryption of electronic data, but it has been superseded by the more secure Advanced Encryption Standard algorithm.

2.2 DISADVANTAGES OF PROPOSED SYSTEM

- The main disadvantage to DES is that it is broken using brute-force, because it is use 48 bits to encrypt the data.
- However, using 3DES mitigates this issue at the cost of increasing execution time.
- DES is also vulnerable to attacks using linear cryptanalysis.
- The number of rounds in DES increases the complexity of the algorithm.

CHAPTER 3
PROPOSED SYSTEM

3.1 PROPOSED SYSTEM

The Advanced Encryption Standard (AES) algorithm is a symmetric block cipher that processes image which is of blocks size 128 bits using three different cipher key size of lengths 128,192 or 256 bits. The image can only be viewed by the receiver as the image is encrypted using AES and the key is only known to the sender and receiver. Since the image is encrypted using AES, it is more secure than the DES and triple DES.

3.2 ADVANTAGES OF PROPOSED SYSTEM

- AES data encryption is a more mathematically efficient and elegant cryptographic algorithm, but its main strength rests in the option for various key lengths.
- It allows you to choose a 128-bit, 192-bit or 256-bit key, making it exponentially stronger than the 56-bit key of DES.
- It is not cracked till now.
- Faster than the DES algorithm.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS

- Python latest Interpreter
- PyCharm
- Crypto. Cipher [Module]
- Windows operating system

4.2 HARDWARE REQUIREMENTS

- Sender Computer: for seeing the original and encrypted image
- Receiver Computer: for seeing the decrypted image

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

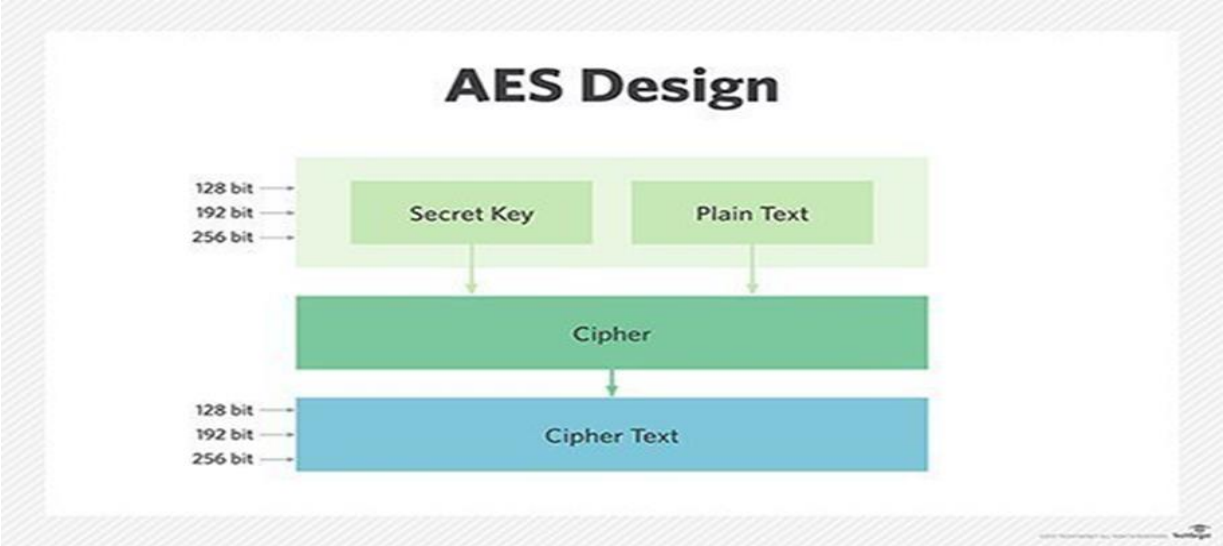


Fig.2: System Architecture

5.2 MODULES

1) Design and Implementation constraints

2) Functional Requirements

3) Non-Functional Requirements

5.2.1 Module -1: Design and Implementation constraints

- Python Interpreter must latest version
- Encryption and Decryption should be done using AES algorithm
- Original Image must be in .jpeg/.png format.

5.2.2 Module – 2:Functional Requirements

- The system shall encrypt the given image to an unreadable format. This is done using AES encryption function.

- The system shall decrypt the received encrypted image to a readable format. This is done using AES decryption function.

5.2.3 Module – 3:Non-Functional Requirements

Performance Requirements:

- For smooth & efficient encryption, image size must be less than 5MB.
- Decryption should not take more than 10 seconds.

Safety and Security Requirements:

- If the decryption takes more than 10 seconds, then discard the message (because the message might have been corrupted during transmission) and ask sender to re-send it.
- Encryption is done using encryption key. Decryption will happen only when same encryption key is used at the receiver side.

5.3 AES Image Encryption:

Conversion of original image i.e plain image into encrypted image i.e cipher image is known as image encryption.

The round consists of the following stages for image encryption :

- SubstituteBytes
- ShiftRow
- MixColumns
- AddRoundKey

SubstituteBytes:

The SubBytes transformation includes non-linear byte substitution, operating on each of the state bytes independently. This is done by using a once-precalculated substitution table called S-box. S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values.

ShiftRow:

ShiftRows transformation includes, the rows of the state are cyclically left shifted. Row 0 remain unchange; row 1 does shift of one byte to the left; row 2 does shift of two bytes to the left and row 3 does shift of three bytes to the left.

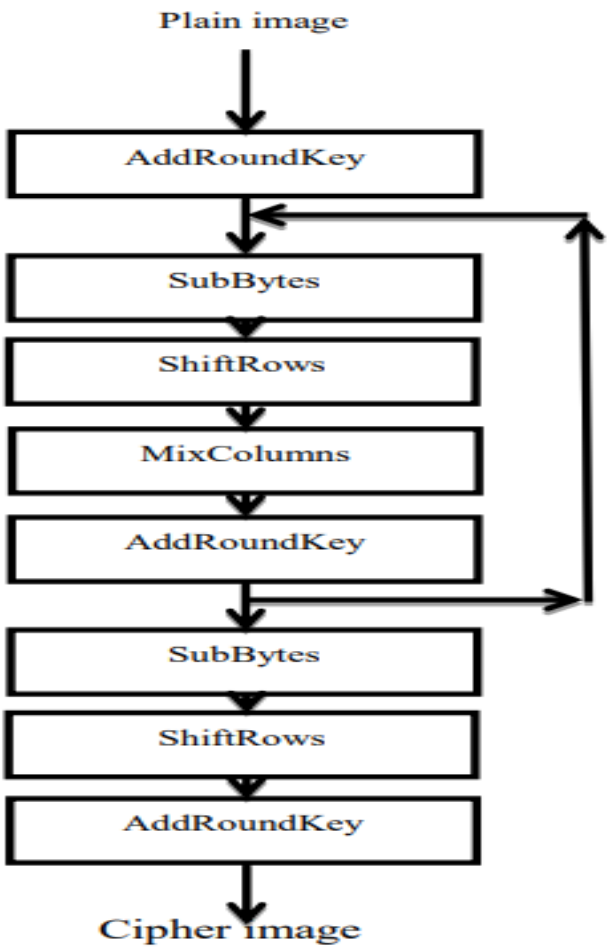


Fig.3:Image Encryption Process

MixColumns:

In MixColumns transformation, the columns of the state are considered as polynomials over GF ( ) and multiplied by modulo + 1 with a fixed polynomial .

AddRoundKey:

In the AddRoundKey transformation, a Round Key is added to the State resulted from the operation of the MixColumns transformation by a simple bitwise XOR operation. The RoundKey of each round is derived from the main key using the KeyExpansion algorithm. The encryption and decryption algorithm needs fourteen 256-bit RoundKey.

5.4 AES Image Decryption:

Reverse of encryption is called decryption. It means conversion of cipher image into plain image. The round consists of the following stage for image decryption.

- AddRoundKey
- InverseShiftRow
- InverseSubstituteByte
- InverseMixColumns

AddRoundKey:

AddRoundKey is its own inverse function because the XOR function is its own inverse. The round keys have to be selected in reverse order.

InverseShiftRow:

InvShiftRows exactly functions the same as ShiftRows, only in the opposite direction. The first row is not shifted, while the second, third and fourth rows are shifted right by one, two and three bytes respectively.

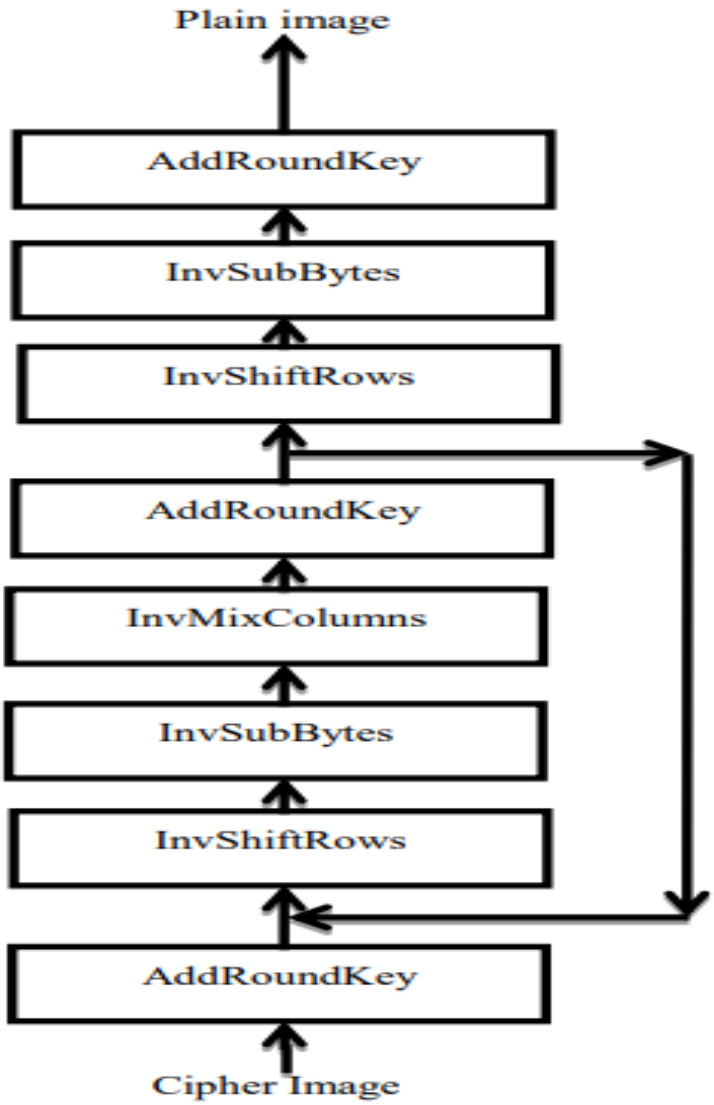


Fig.4: Image Decryption Process

InverseSubstituteByte:

The InvSubBytes transformation is done using a onceprecalculated substitution table called InvS-box. That InvSbox table contains 256 numbers (from 0 to 255) and their corresponding values.

InverseMixColumns:

In the InvMixColumns transformation, the polynomials of degree less than 4 over GF(28), which coefficients are the elements in the columns of the state, are multiplied modulo (x4 + 1) by a fixed polynomial d(x) = {0B}x3 + {0D}x2 + {09}x + {0E}, where {0B}, {0D}; {09}, {0E} denote hexadecimal values

CHAPTER 6
CONCLUSION

6.1. CONCLUSION AND FUTURE ENCHANCEMENT

We have successfully developed a program that encrypts and decrypts the image files accurately. This will help in minimizing the problem of data theft and leaks of other sensitive information. The file that we obtained after encryption is very safe and no one can steal data from this file. So, this file can be sent on a network without worrying. Our developed solution is a small contribution that can be very helpful for military or medical fields in future times.

In Future we are going to build a website that convert image using AES algorithm that Encrypt and Decrypt the user's image. Since image encryption is done using AES, this system provides security from intrusion attacks and the usage of AES technique allows the encryption and decryption process to be more secure and faster. Thus, this system provides security in storage and transmission of digital images.

APPENDIX -I (SOURCE CODE)

```
from Crypto import Random
fromCrypto.Cipher import AES
importos
importos.path
fromos import listdir
fromos.path import isfile, join
import time


classEncryptor:
def __init__(self, key):
self.key = key

def pad(self, s):
return s + b"\0" * (AES.block_size - len(s) % AES.block_size)

def encrypt(self, message, key, key_size=256):
message = self.pad(message)
iv = Random.new().read(AES.block_size)
cipher = AES.new(key, AES.MODE_CBC, iv)
return iv + cipher.encrypt(message)

defencrypt_file(self, file_name):
with open(file_name, 'rb') as fo:
plaintext = fo.read()
enc = self.encrypt(plaintext, self.key)
with open(file_name + ".enc", 'wb') as fo:
fo.write(enc)
os.remove(file_name)

def decrypt(self, ciphertext, key):
iv = ciphertext[:AES.block_size]
cipher = AES.new(key, AES.MODE_CBC, iv)
plaintext = cipher.decrypt(ciphertext[AES.block_size:])
returnplaintext.rstrip(b"\0")

defdecrypt_file(self, file_name):
with open(file_name, 'rb') as fo:
ciphertext = fo.read()
```

```
dec = self.decrypt(ciphertext, self.key)
with open(file_name[:-4], 'wb') as fo:
fo.write(dec)
os.remove(file_name)

defgetAllFiles(self):
dir_path = os.path.dirname(os.path.realpath(__file__))
dirs = []
fordirName, subdirList, fileList in os.walk(dir_path):
forfname in fileList:
if (fname != 'script.py' and fname != 'data.txt.enc'):
dirs.append(dirName + "\\" + fname)
returndirs
defencrypt_all_files(self):
dirs = self.getAllFiles()
forfile_name in dirs:
self.encrypt_file(file_name)
defdecrypt_all_files(self):
dirs = self.getAllFiles()
forfile_name in dirs:
self.decrypt_file(file_name)
key=
b'[EX\xc8\xd5\xbfI{\xa2$\x05(\xd5\x18\xbf\xc0\x85)\x10nc\x94\x02)j\xdf\xcb\xc4\
x94\x9d(\x9e'
enc = Encryptor(key)
clear = lambda: os.system('cls')
ifos.path.isfile('data.txt.enc'):
while True:
password = str(input("Enter password: "))
enc.decrypt_file("data.txt.enc")
p = " “
with open("data.txt", "r") as f:
p = f.readlines()
if p[0] == password:
enc.encrypt_file("data.txt")
break
while True:
clear()
choice = int(input(
```

```
"1. Press '1' to encrypt file.\n2. Press '2' to decrypt file.\n3. Press '3' to Encrypt all files in the directory.\n4. Press '4' to decrypt all files in the directory.\n5. Press '5' to exit.\n"))
clear()
if choice == 1:



enc.encrypt_file(str(input("Enter name of file to encrypt: ")))
elif choice == 2:
enc.decrypt_file(str(input("Enter name of file to decrypt: ")))
elif choice == 3:
enc.encrypt_all_files()
elif choice == 4:
enc.decrypt_all_files()
elif choice == 5:
exit()
else:
print("Please select a valid option!")

else:
while True:
clear()
password = str(input("Setting up stuff. Enter a password that will be used for decryption: "))
repassword = str(input("Confirm password: "))
if password == repassword:
break
else:
print("Passwords Mismatched!")
    f = open("data.txt", "w+")
f.write(password)
f.close()
enc.encrypt_file("data.txt")
print("Please restart the program to complete the setup")
time.sleep(15)
```

APPENDIX-II (SNAP SHOTS)

1. NORMAL IMAGE

Screenshot 1: Normal Image

2. PROGRAM SNIP 1

/home/kali/Desktop/Finalcode.py - Mousepad

File Edit Search View Document Help

```
#!/usr/bin/python3

from Crypto import Random
from Crypto.Cipher import AES
import os
import os.path
from os import listdir
from os.path import isfile, join
import time


class Encryptor:
    def __init__(self, key):
        self.key = key

    def pad(self, s):
        return s + b"\0" * (AES.block_size - len(s) % AES.block_size)

    def encrypt(self, message, key, key_size=256):
        message = self.pad(message)
        iv = Random.new().read(AES.block_size)
        cipher = AES.new(key, AES.MODE_CBC, iv)
        return iv + cipher.encrypt(message)

    def encrypt_file(self, file_name):
        with open(file_name, 'rb') as fo:
            plaintext = fo.read()
        enc = self.encrypt(plaintext, self.key)
        with open(file_name + ".enc", 'wb') as fo:
            fo.write(enc)
        os.remove(file_name)

    def decrypt(self, ciphertext, key):
        iv = ciphertext[:AES.block_size]
        cipher = AES.new(key, AES.MODE_CBC, iv)
        plaintext = cipher.decrypt(ciphertext[AES.block_size:])
        return plaintext.rstrip(b"\0")

    def decrypt_file(self, file_name):
        with open(file_name, 'rb') as fo:
            ciphertext = fo.read()
        dec = self.decrypt(ciphertext, self.key)
        with open(file_name[:-4], 'wb') as fo:
            fo.write(dec)
        os.remove(file_name)

    def getAllFiles(self):
        dir_path = os.path.dirname(os.path.realpath(__file__))
        dirs = []
```

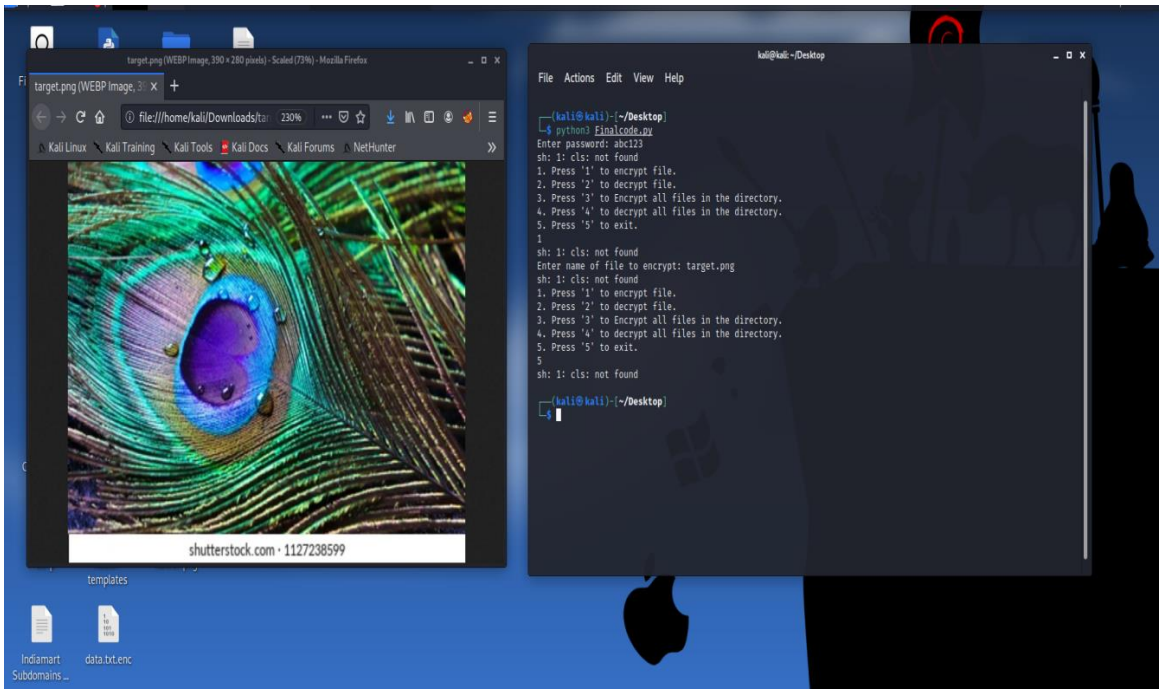
Screenshot 2:Program Snip 1

3. PROGRAM SNIP 2

```
        dirs = []
        for dirName, subdirList, fileList in os.walk(dir_path):
            for fname in fileList:
                if (fname != 'script.py' and fname != 'data.txt.enc'):
                    dirs.append(dirName + "\\" + fname)
        return dirs

    def encrypt_all_files(self):
        dirs = self.getAllFiles()
        for file_name in dirs:
            self.encrypt_file(file_name)

    def decrypt_all_files(self):
        dirs = self.getAllFiles()
        for file_name in dirs:
            self.decrypt_file(file_name)


key = b'[EX\xc8\xd5\xbfI{\xaz$\x85(\xd5\x18\xbf\xc0\x85)\x10nc\x94\x02}j\xdf\xcb\xc4\x94\x9d(\x9e'
enc = Encryptor(key)
clear = lambda: os.system('cls')

if os.path.isfile('data.txt.enc'):
    while True:
        password = str(input("Enter password: "))
        enc.decrypt_file("data.txt.enc")
        p = ''
        with open("data.txt", "r") as f:
            p = f.readlines()
        if p[0] == password:
            enc.encrypt_file("data.txt")
            break

    while True:
        clear()
        choice = int(input(
            "1. Press '1' to encrypt file.\n2. Press '2' to decrypt file.\n3. Press '3' to Encrypt all files in the directory.\n4. Press '4' to decrypt all files in the directory.\n5. Press '5' to exit.\n"))
        clear()
        if choice == 1:
            enc.encrypt_file(str(input("Enter name of file to encrypt: ")))
        elif choice == 2:
            enc.decrypt_file(str(input("Enter name of file to decrypt: ")))
        elif choice == 3:
            enc.encrypt_all_files()
        elif choice == 4:
            enc.decrypt_all_files()
        elif choice == 5:
            exit()
```

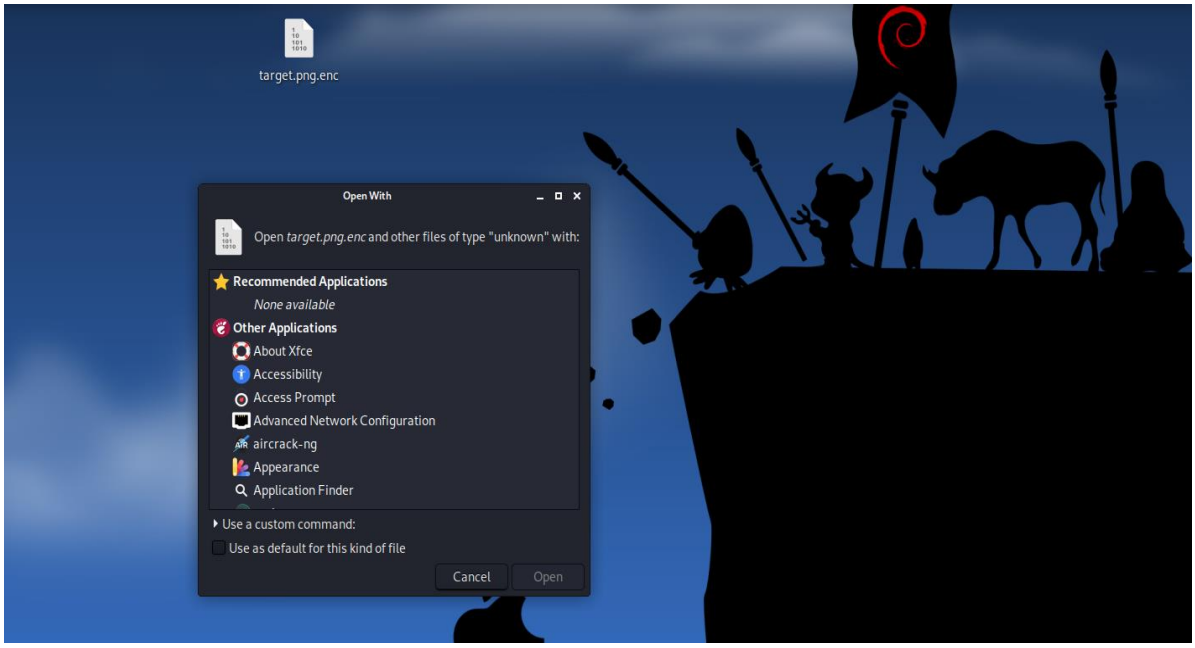
Screenshot 3: Program Snip 2

4. IMAGE ENCRYPTION



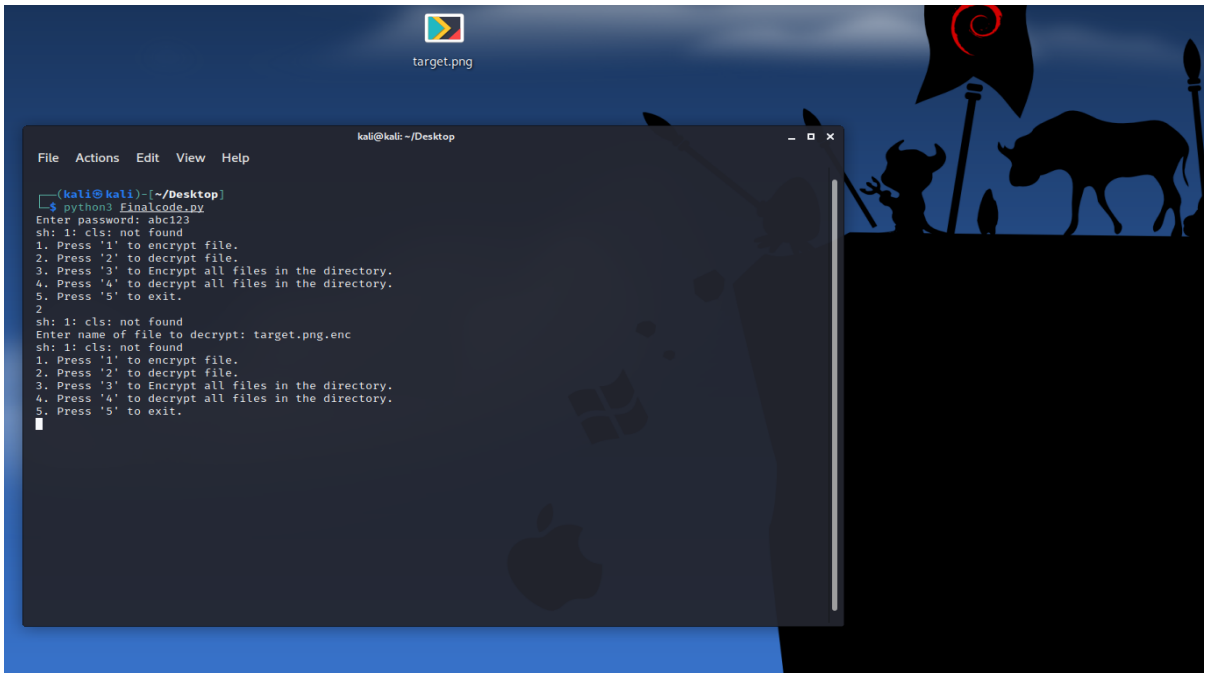
Screenshot 4: Image Encryption

5. AFTER ENCRYPTION



Screenshot 5:After Encryption

6. IMAGE DECRYPTION



Screenshot 6:After Decryption

7.AFTER DECRYPTION

Screenshot 7: After Decryption

REFERENCES

[1] Norouz B et al (2014) A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process. Multimedia Tools Appl 71(3):1469–1497

[2] Kumari M, Gupta S, Sardana P (2017) A survey of image encryption algorithms. 3D Res 8(4):37

[3] Adolph C., Amano K., Bang-Jensen B., Fullman N., Wilkerson J. *medRxiv.* 2020

[4] Partheeban P, Kavitha V (2018) Dynamic key dependent AES S-box generation with optimized quality analysis. Cluster Computer. https://doi.org/10.1007/s10586-018-2386-6

[5] ElBadawy ESAM, El-Masry WA, Mokhtar A, Hafez AES (2010) A new chaos advanced encryption standard (AES) algorithm for data security.

[6] El Maraghy M, Hesham S and Abd El Ghany M.A, "Real-time Efficient FPGA Implementation of AES Algorithm", IEEE International SOC Conference (SOCC), page 203-208, Sept 2013.

[7] M.Sambasiva Reddy and Mr.Y.Amar Babu, "Evaluation Of Microblaze and Implementation Of AES Algorithm using Spartan-3E", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 7, page 3341-3347, July 2013.

[8] Hoang Trang and Nguyen Van Loi, "An Efficient FPGA Implementation of The Advanced Encryption Standard algorithm", IEEE International Conference on Computing and Communication Technology, page 1-4, Ho Chi Minh city, 2012.

[9] Kamali S.H, Shakerian R, Hedayati M and Rahmani M, "A new modied version of Advanced Encryption Standard based algorithm for image encryption", (ICEIE) International Conference On Electronics and Information Engineering, volume 1, page 1250-1255, Aug 2010.

[10] Ahmad N, Hasan R and Jubadi W.M, "Design of AES Sbox using combinational logic optimization", IEEE Symposium on Industrial Electronics & Applications (ISIEA), page 512-517, Oct 2010.

[11] M. Zeghid, M. Machhout, L. Khriji, A. Baganne, and R. Tourki, “A Modified AES Based Algorithm for Image Encryption”, International Journal of Computer, Information, Systems and Control Engineering Vol:1 No:3, page 726-731, 2007.

[12] FIPS 197, "Advanced Encryption Standard (AES)," November 26, 2001.

[13] National Institute of Standards and Technology (NIST), "Data Encryption Standard (DES)," National Technical InformationService VA 22161, 1999.