Abstact:

Image Steganography – As the name suggests, Image Steganography refers to the process of hiding data within an image file. The image selected for this purpose is called the cover-image and the image obtained after steganography is called the stego-image. So, steganography is the technique of encrypting data in a data carrier (an image, in this case) in such a manner that it is impossible for an outsider to identify that a message has been encrypted into the image, a novel data hiding technique based on the BPCS technique of digital images is presented in this report. Data hiding is one of the best topics in secret communication. A lossless data hiding technique using bits in images is presented in this project. The data hiding technique does not affect the visible properties of the image. Steganography is art and science of hiding the fact that communication is 3 taking place. Secrets can be hidden in all types of medium: text, audio, video and images.Steganography is the practice of hiding a secret message inside of (or even on top of) something that is not secret. That something can be just about anything you want.

The purpose of steganography is to conceal and deceive. It is a form of covert communication and can involve the use of any medium to hide messages. It's not a form of cryptography, because it doesn't involve scrambling data or using a key. Instead, it is a form of data hiding and can be executed in clever ways. Where cryptography is a science that largely enables privacy, steganography is a practice that enables secrecy and deceit.

AES cryptosystem is one of the well-known block ciphers used to secure the information and protect the communication by providing a difficulty to the attackers specifically in cryptanalysis. However, researchers have underscored some weakness points of block ciphers, for instance:

(a) all block ciphers apply the same key for the encipherment of multiple blocks.

(b) if an adversary can discover the key for one block, he can readily break the other blocks.

Keywords:

QR code, data transfer, encryption, subcarriers, differential phase shift keying,AES, LSB, MSB,Differencing GrayScale Image, information hiding, Pixel Mapping,Pixel value,Image steganography.

1. Introduction:

1.1 Theoretical Background

Usually, the data is concealed inside an innocuous cover such that even if hostile agents discover the cover, there is no suspicion about the presence of data in that cover. Steganography and Cryptography are cousins in the spy craft family. However, Cryptographic & Steganographic techniques differ from each other. In cryptography, the original message is scrambled i.e. its original structure is changed in order to make it meaningless. Thus, when an attacker discovers the message it is still difficult for him to get the original message back.

Steganography does not attempt to scramble the original message. The intention of both steganography & cryptography is to protect the original message from the attacker. Both are excellent means, but when used alone can be broken. As a result, several experts have suggested the idea of using both the techniques in order to provide an additional layer of security. Cryptographic technique encrypts the original message. This encrypted message is then hidden using steganographic technique. Now, even if the attacker defeats the steganographic technique, he still requires a cryptographic decoding key to decipher the encrypted message.

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix.Unlike DES, the number of rounds in AES is variable and depends on the length of the key.

1.2 Motivation

Imagine a situation where the data to be shared is first encrypted, then embedded in a carrier file (say, an image), and then this entire steganographic combination is encrypted again. So, it basically provides a multitude of layers of protection of the data, and not only is the data encrypted, if the hacker manages to break the first layer of encryption, the information won't make any sense, as it'll just be the carrier file. If he/she manages to somehow get the key for getting the hidden data, there's still one more layer of encryption to break. So, it is a sort of cascading protection.

3. Overview of the Proposed System :

3.1. Aim of the proposed Work

The increase in the number of cyber crimes is a major concern in society these days.Any product's QR code can be scanned with the help of a QR code scanner and the information contained in it can be read.As result of which, data integrity is at risk.The main aim of our project is to safeguard data integrity and confidentiality and increase the security in data transmission so that it can be immune from cyber attacks.Also our project is made such that it can easily be used by any person and the information can be encrypted using a QR code or an image.With the help of our model, data may be transferred very securely over the network and only the sender and the receiver will be able to encrypt/decrypt the data.

3.2. Objective(s) of the proposed work

1. Increasing data confidentiality and integrity
2. Increasing the security of the communication system so that it can be immune to fubar attacks.
3. Using various latest technologies for encryption, decryption and transmission of data so that efficiency can be maintained.

3.3. Introduction and Related Concepts

Stegnography:

Steganography is the practice of hiding a secret message inside of (or even on top of) something that is not secret. That something can be just about anything you want. These days, many examples of steganography involve embedding a secret piece of text inside of a picture. Or hiding a secret message or script inside of a Word or Excel document.The purpose of steganography is to conceal and deceive. It is a form of covert communication and can involve the use of any medium to hide messages. It’s not a form of cryptography, because it doesn’t involve scrambling data or using a key. Instead, it is a form of data hiding and can be executed in clever ways. Where cryptography is a science that largely enables privacy, steganography is a practice that enables secrecy – and deceit.

AES256 Mechanism:

1. **Key expansion**, which creates new keys, known as round keys, for each subsequent round of encryption, using Rijndael's key schedule.
2. **Round key addition**, during which the initial round key is added to the mix of data that has been divided.
3. **Byte substitution**, which substitutes every byte with a different byte based on the Rijndael S-box substitution box.
4. **Row shifting**, which moves every row of the divided data one space to the left for the second row, two spaces to the left for the third row, and three spaces to the left for the fourth row.
5. **Column mixing**, which uses a pre-established matrix to multiply the divided data's columns and create a new block of code.
6. **Round key addition**, during which another round key is added to the mixture of columns.

3.4. Framework, Architecture or Module for the Proposed System(with explanation)

Module Used:

- **Os , binascii , Crypto** , **base64 , numpy** , **qrcode** , **pyzbar** , **PIL**

1. binascii:

The binascii module contains a number of methods to convert between binary and various ASCII-encoded binary representations.

2.Crypto:

This is a collection of both secure hash functions (such as SHA256 and RIPEMD160), and various encryption algorithms (AES, DES, RSA, ElGamal, etc.).

3.base64:

In Python the base64 module is used to encode and decode data. First, the strings are converted into byte-like objects and then encoded using the base64 module. The below example shows the implementation of encoding strings isn't base64 characters.

4.numpy:

NumPy is a Python library used for working with arrays.
It also has functions for working in domain of linear algebra, fourier transform, and matrices.

5.qrcode:

pyqrcode module is a QR code generator. The module automates most of the building process for creating QR codes. This module attempts to follow the QR code standard as closely as possible. The terminology and the encodings used in pyqrcode come directly from the standard.

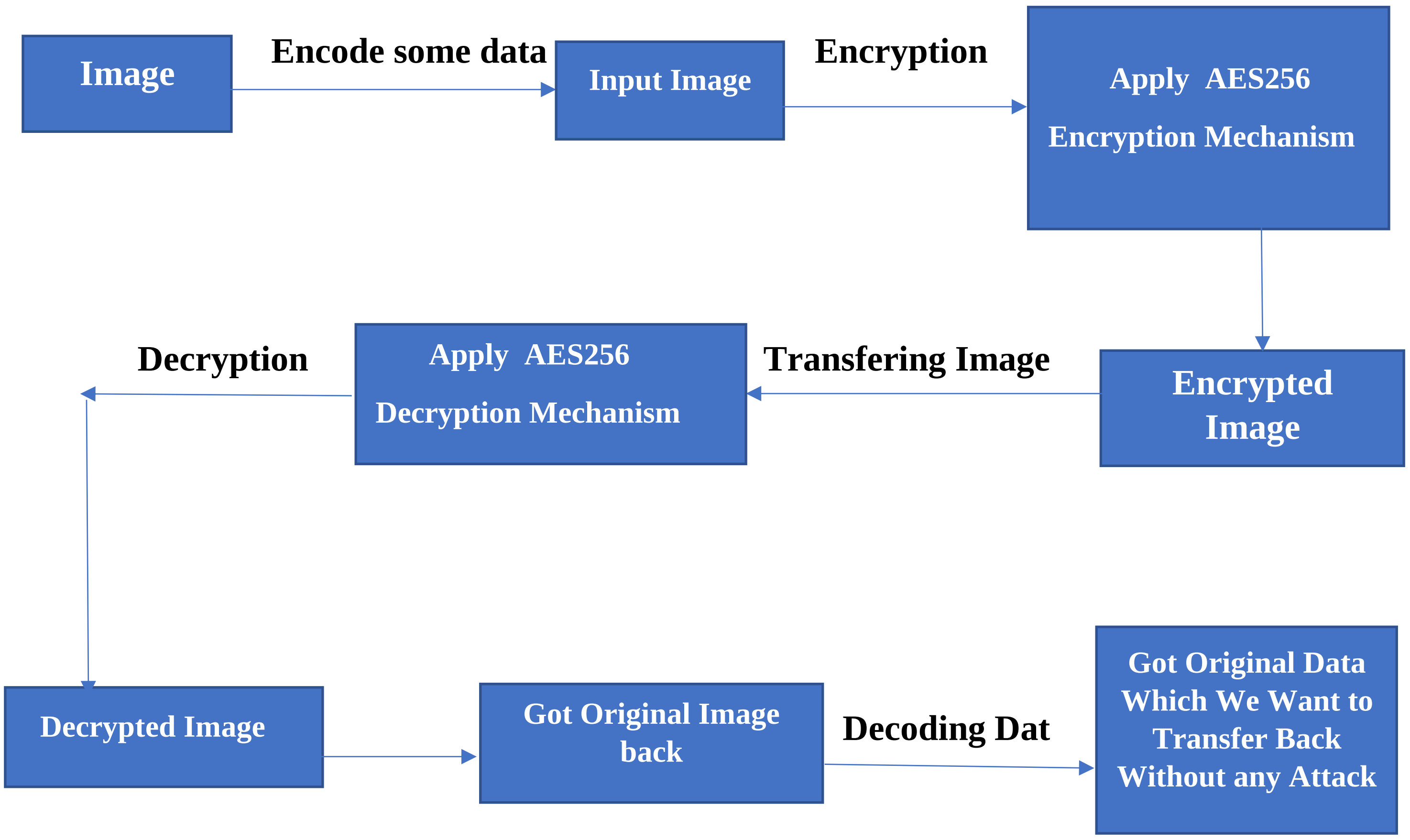
6.pyzbar:

The pyzbar module is a module that is responsible for reading and decoding 1-D barcodes or QR codes easily and it requires PIL module in order to function properly.

7.PIL:

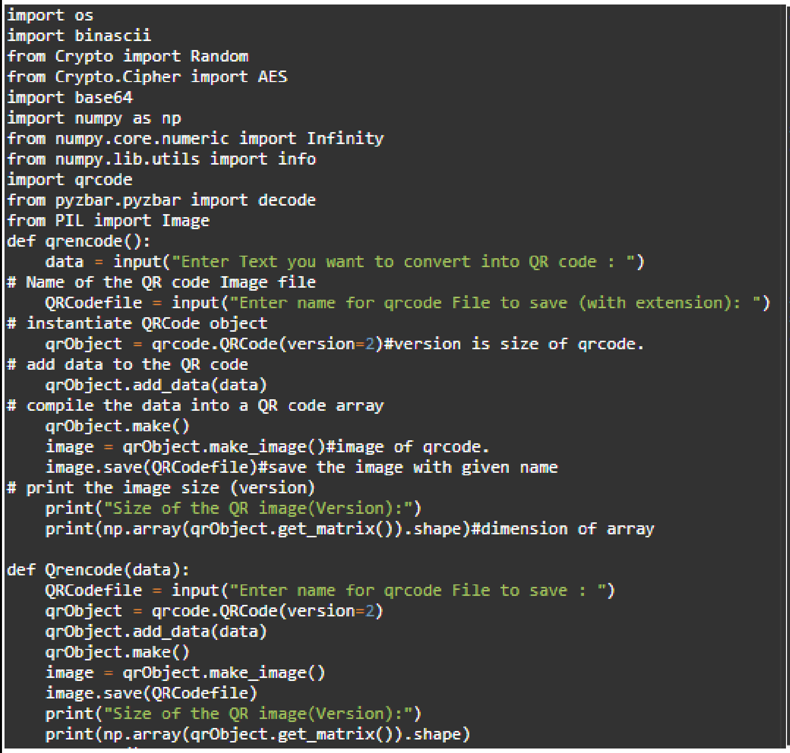
Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file format.

3.5. Proposed System Model(ER Diagram/UML Diagram/Mathematical Modeling)

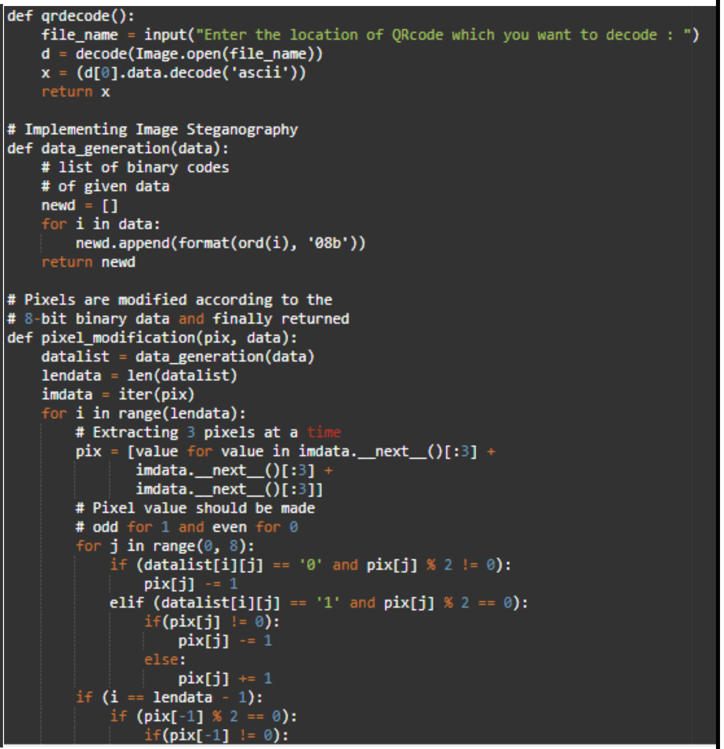


4. Proposed\ System Analysis and Design:

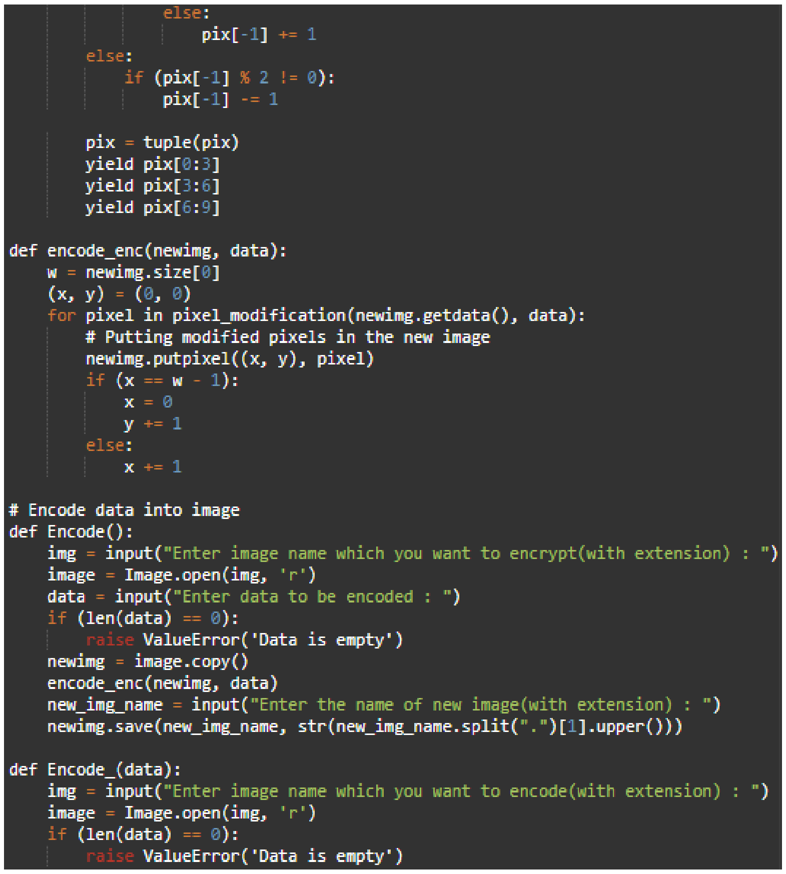
1. We have used AES-128 in our project. For this , CBC mode has been used. The use of CBC mode is done because of its acceptance in most of the fields and also because of its high speed encryption and security.
2. For the QR code encryption and decryption, the library pyzbar has been used.For this, the version 1 has been used and block size is 12. qrencode () function has been used to make the object of qr and embed the data into it.For the decryption part, qrdecode() function is used which will decode the text and returns the ascii format of the text.
3. Steganography is implemented using 4 main functions that are modpix(),gendata(),encode() and decode().
4. The gendata() function is used to generate the 8-bit ascii value of each character present in the given text.
5. modpix() function is used to modify the values of the pixels based on the concept of even/odd and comparing the 8-bit ascii value. If the pixel value is even and the corresponding binary value is 1, then it is made odd otherwise it is kept the same. If the pixel value is odd and the corresponding binary value is 0, then it is made even otherwise it is kept the same.At the end, after all the pixels have been modified, 3 pixels at a time are returned and each pixel corresponds to R,G and B.
6. encode() function has been used to encode the modified pixels received into the image.This function will return an image with data embedded in it in the form of pixels.
7. The decode function will get the pixels from the image, and convert it into original pixels and then decode them and will give us back the text encoded.
8. The rest of the code asks the user to input the text, the choices which user wants and the type of techniques the user wants to use.



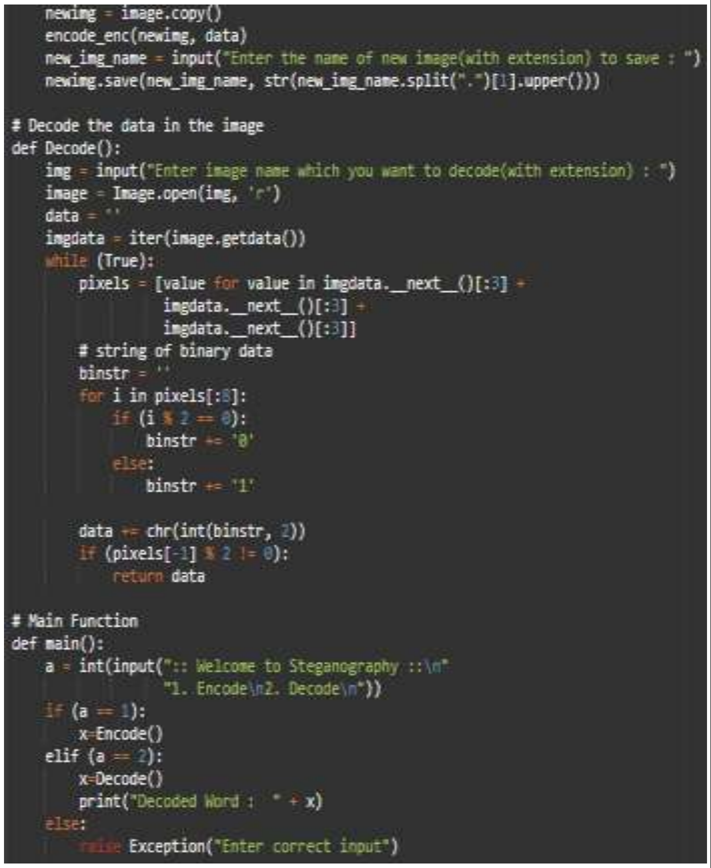
```python
import os
import binascii
from Crypto import Random
from Crypto.Cipher import AES
import base64
import numpy as np
from numpy.core.numeric import Infinity
from numpy.lib.utils import info
import qrcode
from pyzbar.pyzbar import decode
from PIL import Image
def qrencode():
    data = input("Enter Text you want to convert into QR code : ")
# Name of the QR code Image file
    QRCodefile = input("Enter name for qrcode File to save (with extension): ")
# instantiate QRCode object
    qrObject = qrcode.QRCode(version=2)#version is size of qrcode.
# add data to the QR code
    qrObject.add_data(data)
# compile the data into a QR code array
    qrObject.make()
    image = qrObject.make_image()#image of qrcode.
    image.save(QRCodefile)#save the image with given name
# print the image size (version)
    print("Size of the QR image(Version):")
    print(np.array(qrObject.get_matrix()).shape)#dimension of array

def Qrencode(data):
    QRCodefile = input("Enter name for qrcode File to save : ")
    qrObject = qrcode.QRCode(version=2)
    qrObject.add_data(data)
    qrObject.make()
    image = qrObject.make_image()
    image.save(QRCodefile)
    print("Size of the QR image(Version):")
    print(np.array(qrObject.get_matrix()).shape)
```



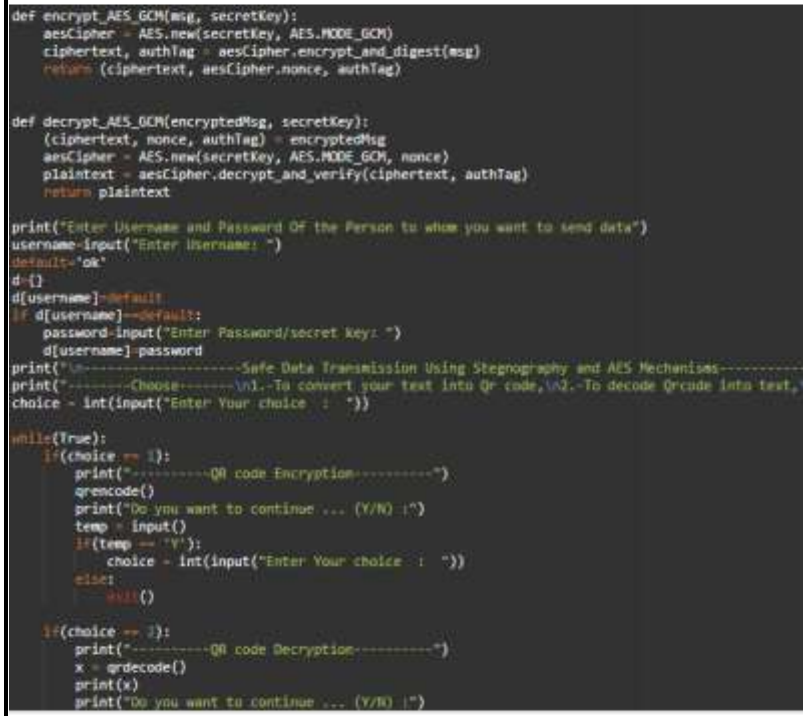
```python
def qrdecode():
    file_name = input("Enter the location of QRcode which you want to decode : ")
    d = decode(Image.open(file_name))
    x = (d[0].data.decode('ascii'))
    return x

# Implementing Image Steganography
def data_generation(data):
    # list of binary codes
    # of given data
    newd = []
    for i in data:
        newd.append(format(ord(i), '08b'))
    return newd

# Pixels are modified according to the
# 8-bit binary data and finally returned
def pixel_modification(pix, data):
    datalist = data_generation(data)
    lendata = len(datalist)
    imdata = iter(pix)
    for i in range(lendata):
        # Extracting 3 pixels at a time
        pix = [value for value in imdata.__next__()[:3] +
               imdata.__next__()[:3] +
               imdata.__next__()[:3]]
        # Pixel value should be made
        # odd for 1 and even for 0
        for j in range(0, 8):
            if (datalist[i][j] == '0' and pix[j] % 2 != 0):
                pix[j] -= 1
            elif (datalist[i][j] == '1' and pix[j] % 2 == 0):
                if(pix[j] != 0):
                    pix[j] -= 1
                else:
                    pix[j] += 1
        if (i == lendata - 1):
            if (pix[-1] % 2 == 0):
                if(pix[-1] != 0):
```



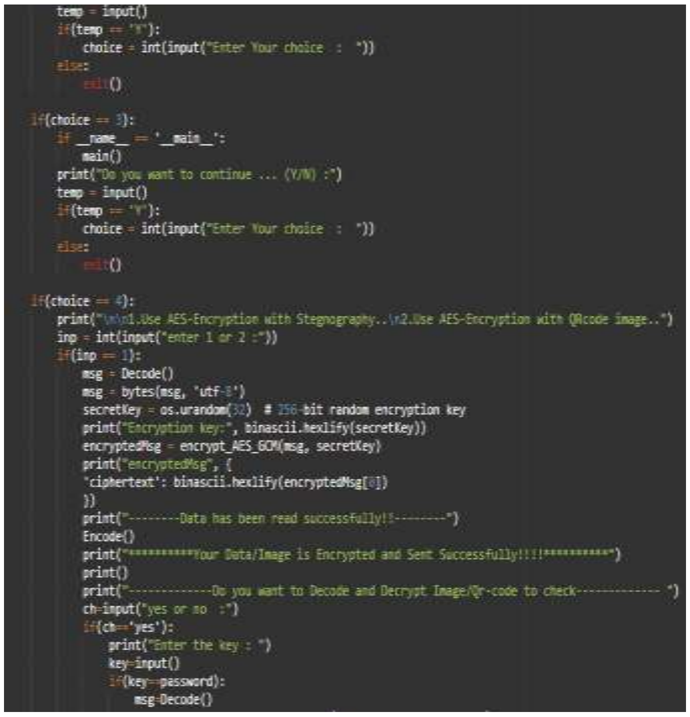
```python
                else:
                    pix[-1] += 1
            else:
                if (pix[-1] % 2 != 0):
                    pix[-1] -= 1

        pix = tuple(pix)
        yield pix[0:3]
        yield pix[3:6]
        yield pix[6:9]

def encode_enc(newimg, data):
    w = newimg.size[0]
    (x, y) = (0, 0)
    for pixel in pixel_modification(newimg.getdata(), data):
        # Putting modified pixels in the new image
        newimg.putpixel((x, y), pixel)
        if (x == w - 1):
            x = 0
            y += 1
        else:
            x += 1

# Encode data into image
def Encode():
    img = input("Enter image name which you want to encrypt(with extension) : ")
    image = Image.open(img, 'r')
    data = input("Enter data to be encoded : ")
    if (len(data) == 0):
        raise ValueError('Data is empty')
    newimg = image.copy()
    encode_enc(newimg, data)
    new_img_name = input("Enter the name of new image(with extension) : ")
    newimg.save(new_img_name, str(new_img_name.split(".")[1].upper()))

def Encode_(data):
    img = input("Enter image name which you want to encode(with extension) : ")
    image = Image.open(img, 'r')
    if (len(data) == 0):
        raise ValueError('Data is empty')
```



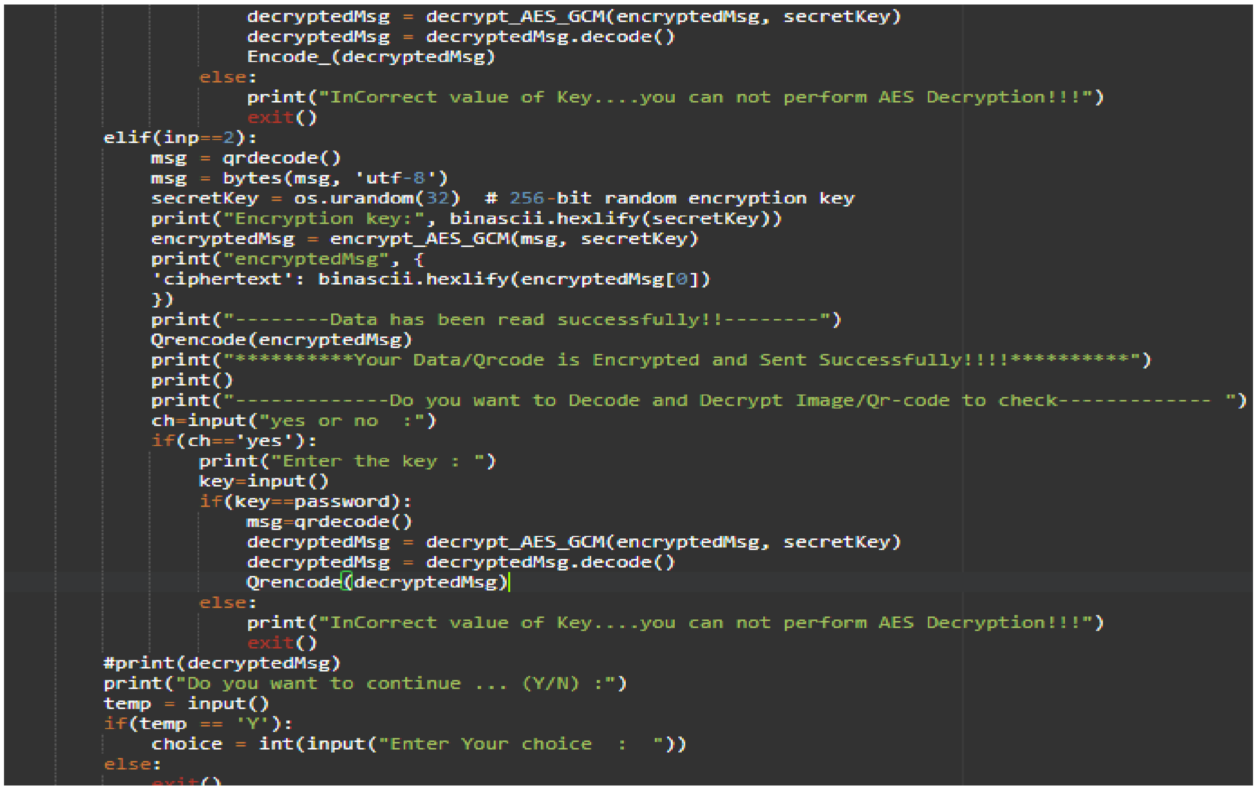
```python
    newimg = image.copy()
    encode_enc(newimg, data)
    new_img_name = input("Enter the name of new image(with extension) to save : ")
    newimg.save(new_img_name, str(new_img_name.split(".")[1].upper()))

# Decode the data in the image
def Decode():
    img = input("Enter image name which you want to decode(with extension) : ")
    image = Image.open(img, 'r')
    data = ''
    imgdata = iter(image.getdata())
    while (True):
        pixels = [value for value in imgdata.__next__()[:3] +
                  imgdata.__next__()[:3] +
                  imgdata.__next__()[:3]]
        # string of binary data
        binstr = ''
        for i in pixels[:8]:
            if (i % 2 == 0):
                binstr += '0'
            else:
                binstr += '1'

        data += chr(int(binstr, 2))
        if (pixels[-1] % 2 != 0):
            return data

# Main Function
def main():
    a = int(input(":: Welcome to Steganography ::\n"
                  "1. Encode\n2. Decode\n"))
    if (a == 1):
        x=Encode()
    elif (a == 2):
        x=Decode()
        print("Decoded Word :  " + x)
    else:
        raise Exception("Enter correct input")
```



```
def encrypt_AES_GCM(msg, secretKey):
    aesCipher = AES.new(secretKey, AES.MODE_GCM)
    ciphertext, authTag = aesCipher.encrypt_and_digest(msg)
    return (ciphertext, aesCipher.nonce, authTag)


def decrypt_AES_GCM(encryptedMsg, secretKey):
    (ciphertext, nonce, authTag) = encryptedMsg
    aesCipher = AES.new(secretKey, AES.MODE_GCM, nonce)
    plaintext = aesCipher.decrypt_and_verify(ciphertext, authTag)
    return plaintext

print("Enter Username and Password Of the Person to whom you want to send data")
username=input("Enter Username: ")
default='ok'
d={}
d[username]=default
if d[username]==default:
    password=input("Enter Password/secret key: ")
    d[username]=password
print("\n---------------------Safe Data Transmission Using Stegnography and AES Mechanisms----------
print("--------Choose-------\n1.-To convert your text into Qr code,\n2.-To decode Qrcode into text,
choice = int(input("Enter Your choice  :  "))

while(True):
    if(choice == 1):
        print("----------QR code Encryption----------")
        qrencode()
        print("Do you want to continue ... (Y/N) :")
        temp = input()
        if(temp == 'Y'):
            choice = int(input("Enter Your choice  :  "))
        else:
            exit()

    if(choice == 2):
        print("----------QR code Decryption----------")
        x = qrdecode()
        print(x)
        print("Do you want to continue ... (Y/N) :")
```



```
        temp = input()
        if(temp == 'Y'):
            choice = int(input("Enter Your choice  :  "))
        else:
            exit()

    if(choice == 3):
        if __name__ == '__main__':
            main()
        print("Do you want to continue ... (Y/N) :")
        temp = input()
        if(temp == 'Y'):
            choice = int(input("Enter Your choice  :  "))
        else:
            exit()

    if(choice == 4):
        print("\n\n1.Use AES-Encryption with Stegnography..\n2.Use AES-Encryption with QRcode image..")
        inp = int(input("enter 1 or 2 :"))
        if(inp == 1):
            msg = Decode()
            msg = bytes(msg, 'utf-8')
            secretKey = os.urandom(32)  # 256-bit random encryption key
            print("Encryption key:", binascii.hexlify(secretKey))
            encryptedMsg = encrypt_AES_GCM(msg, secretKey)
            print("encryptedMsg", {
            'ciphertext': binascii.hexlify(encryptedMsg[0])
            })
            print("--------Data has been read successfully!!--------")
            Encode()
            print("**********Your Data/Image is Encrypted and Sent Successfully!!!!**********")
            print()
            print("------------Do you want to Decode and Decrypt Image/Qr-code to check------------ ")
            ch=input("yes or no  :")
            if(ch=='yes'):
                print("Enter the key : ")
                key=input()
                if(key==password):
                    msg=Decode()
```



```
                    decryptedMsg = decrypt_AES_GCM(encryptedMsg, secretKey)
                    decryptedMsg = decryptedMsg.decode()
                    Encode_(decryptedMsg)
                else:
                    print("InCorrect value of Key....you can not perform AES Decryption!!!")
                    exit()
        elif(inp==2):
            msg = qrdecode()
            msg = bytes(msg, 'utf-8')
            secretKey = os.urandom(32)  # 256-bit random encryption key
            print("Encryption key:", binascii.hexlify(secretKey))
            encryptedMsg = encrypt_AES_GCM(msg, secretKey)
            print("encryptedMsg", {
            'ciphertext': binascii.hexlify(encryptedMsg[0])
            })
            print("--------Data has been read successfully!!--------")
            Qrencode(encryptedMsg)
            print("**********Your Data/Qrcode is Encrypted and Sent Successfully!!!!**********")
            print()
            print("-------------Do you want to Decode and Decrypt Image/Qr-code to check------------- ")
            ch=input("yes or no  :")
            if(ch=='yes'):
                print("Enter the key : ")
                key=input()
                if(key==password):
                    msg=qrdecode()
                    decryptedMsg = decrypt_AES_GCM(encryptedMsg, secretKey)
                    decryptedMsg = decryptedMsg.decode()
                    Qrencode(decryptedMsg)
                else:
                    print("InCorrect value of Key....you can not perform AES Decryption!!!")
                    exit()
        #print(decryptedMsg)
        print("Do you want to continue ... (Y/N) :")
        temp = input()
        if(temp == 'Y'):
            choice = int(input("Enter Your choice  :  "))
        else:
            exit()
```

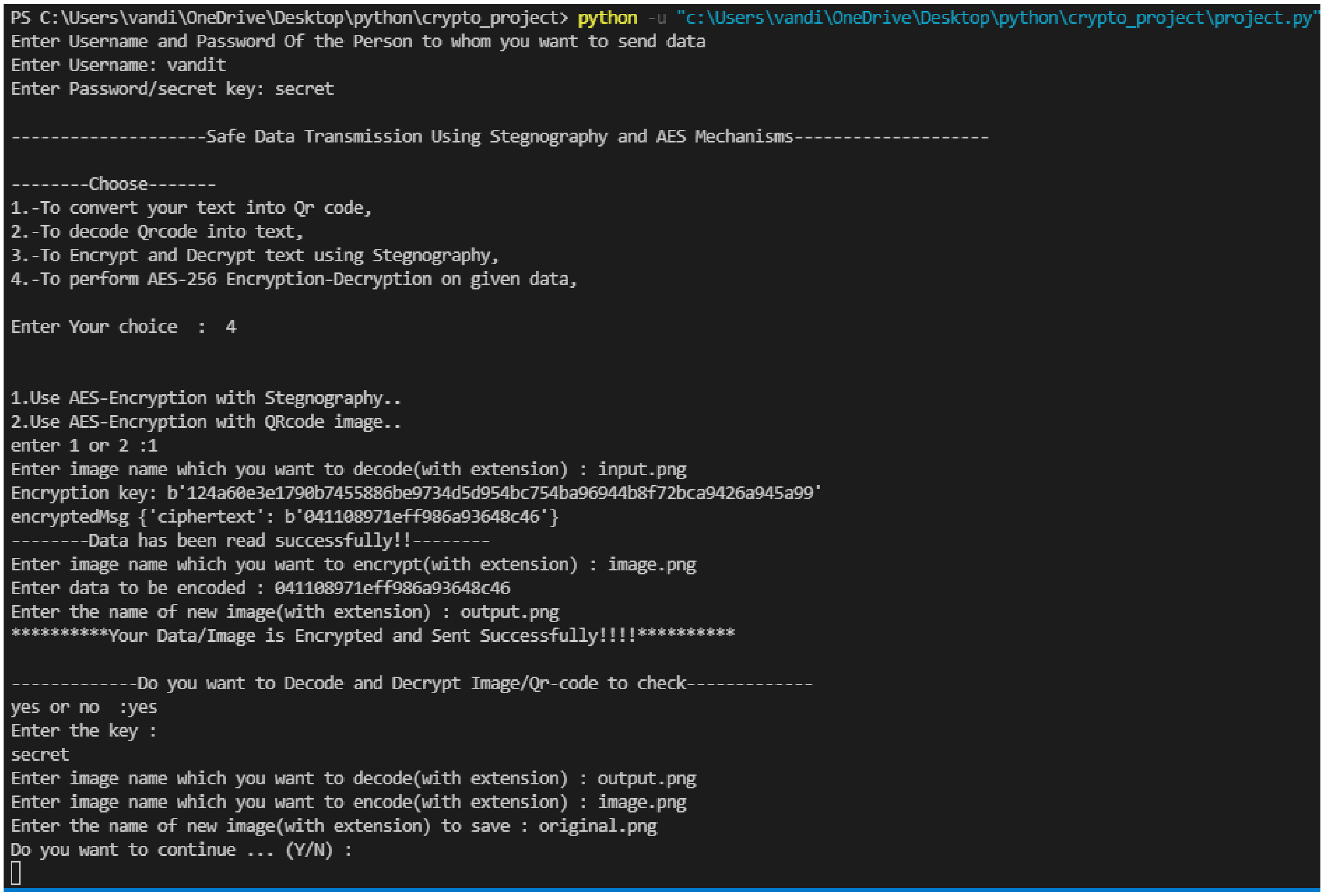
5. Results and Discussion:

Input image:

Data Security
Vs
Data Privacy
StealthLabs

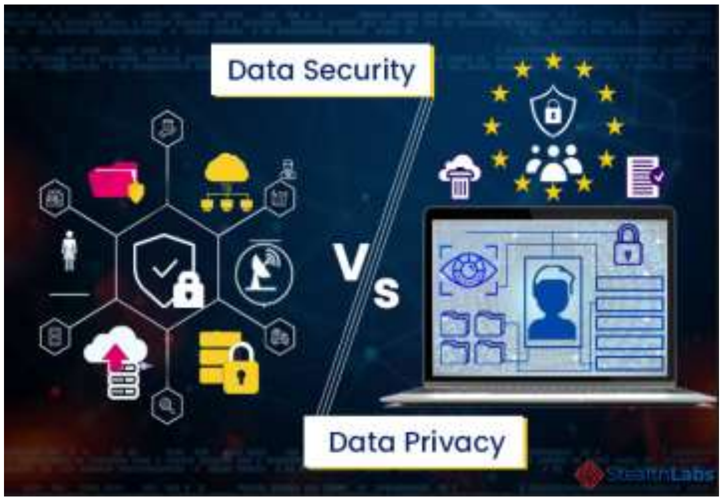
Input image with data:

Data Security
Vs
Data Privacy
StealthLabs

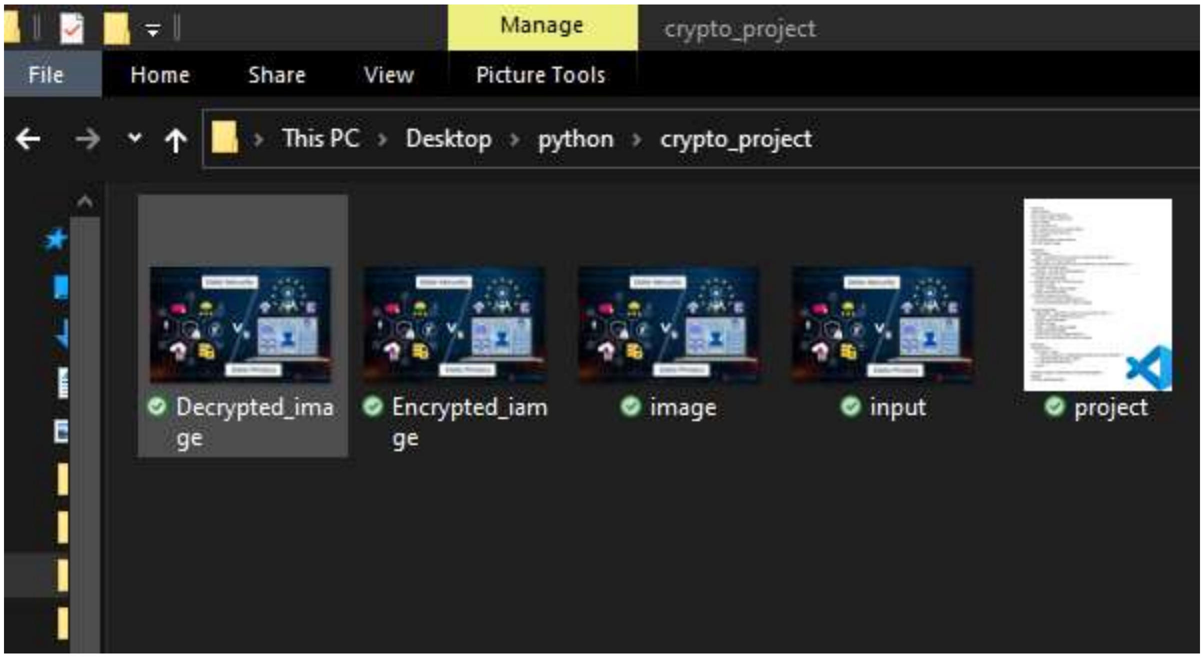


```
PS C:\Users\vandi\OneDrive\Desktop\python\crypto_project> python -u "c:\Users\vandi\OneDrive\Desktop\python\crypto_project\project.py"
Enter Username and Password Of the Person to whom you want to send data
Enter Username: vandit
Enter Password/secret key: secret

--------------------Safe Data Transmission Using Stegnography and AES Mechanisms--------------------

--------Choose-------
1.-To convert your text into Qr code,
2.-To decode Qrcode into text,
3.-To Encrypt and Decrypt text using Stegnography,
4.-To perform AES-256 Encryption-Decryption on given data,

Enter Your choice  :  4


1.Use AES-Encryption with Stegnography..
2.Use AES-Encryption with QRcode image..
enter 1 or 2 :1
Enter image name which you want to decode(with extension) : input.png
Encryption key: b'124a60e3e1790b7455886be9734d5d954bc754ba96944b8f72bca9426a945a99'
encryptedMsg {'ciphertext': b'041108971eff986a93648c46'}
--------Data has been read successfully!!--------
Enter image name which you want to encrypt(with extension) : image.png
Enter data to be encoded : 041108971eff986a93648c46
Enter the name of new image(with extension) : output.png
**********Your Data/Image is Encrypted and Sent Successfully!!!!**********

------------Do you want to Decode and Decrypt Image/Qr-code to check------------
yes or no  :yes
Enter the key :
secret
Enter image name which you want to decode(with extension) : output.png
Enter image name which you want to encode(with extension) : image.png
Enter the name of new image(with extension) to save : original.png
Do you want to continue ... (Y/N) :
```

Encrypted image: **Image After Decryption:**



Data Security
Vs
Data Privacy
StealthLabs



With the importance and uses of Internet only increasing every passing moment, the need for confidentiality and security for communication has only been rising in order to be safe from the issue of information threats as Sending and Receiving information via mail or over web browsers aren't as secure anymore, as confidential information like passwords and bank details are now commonly sent over such Platforms.

As of now, the two most widely used techniques for security and confidentiality of data are Cryptography and Steganography. Cryptography scrambles the information into a cipher so that a third person will not be able to access it without the key while Steganography hides the information within a media file (usually image or video) to prevent being identified by an outside party.
One of the most prominent issues with Cryptography is the possibility of the Key being compromised, which leads to the issue of safe transmission of keys which further requires another system for the transmission. While Public Key encryption can be a solution, it has yet to be proven to be unbreakable. It is fair to say that in the not-so-distant Future, Quantum Computing will be able to break and expose the relationship between Public and Private Keys. Thus, this is just a temporary fix of converting the issue into another form. The drawbacks can be summarized as:

- Hey-Distribution problem
- Mathematical vulnerabilities of Asymmetric Encryption
- Legal limitations by Government
- Cryptanalysis

Due to huge threat of Steganography being misused by terrorists and criminals, studies on breaking Steganography and finding its weaknesses and flaws are being heavily supported and pioneered all to prevent its illicit use. As of now, the methods are:

- Visual Analysis: This aims to check and expose the presence of information/data being hidden with the file through a naked eye or computer-aided inspection.
- Statistical Analysis: This aims to find small alterations in the cover medium and thereby unravel the statistical features in the Steganographic process used.

Therefore, it has been shown that while, Cryptography and Steganography individually provides security and confidentiality to the message/data, both still have quite a few drawbacks and vulnerabilities and are hence individually insufficient to guarantee security.
Hence, creating a system in which both the methods are combined and intertwined to create a completely new system would solve and balance out almost all of the vulnerabilities created by each strategy and will hence achieve a credible method to guarantee safe and secure transmission of data while also guaranteeing the confidentiality of the message.