Project Guide: Dr. S. Alagumuthu Krishnan

Project Title: Steganography applications for a smart system

Team Members:

Anil-17R01A0533

Janakiram - 17R01A05E5

Manoj-17R01A05C2

Shreya Indukuri -17R01A05D4

Index

Introduction

- Innovation of technology and having fast Internet make information to distribute over the world easily and economically. This is made people to worry about their privacy and works.
- This is when the concept of encryption has come into picture. As one of the most popular encryption methods cryptography was a method where the messages from the sender could be encrypted using a key, which could later be decrypted by the receiver using the same or a different key.
- One breakthrough in the evolution of encryption that would overcome this problem of unwanted attention would be Steganography.
- Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. It is a technique that prevents unauthorized users to have access to the important data and provide methods that users can hide and mix their information within other information that make them difficult to recognize by attackers.

Problem Statement

.

- Cryptography has always been controversial, that encrypted messages would draw suspicion from others. This method would trigger an attempt to decrypt the ciphertext, thereby indirectly promoting the chance of the message being viewed by a third party except for sender and viewer.
- The advantage of steganography over cryptography alone is that the messages do not attract attention to themselves.
- So, by implementing Steganography it is possible to hide or embed the text file into an image, it is hard to suspect if it contains any secret message or not as the processed image looks similar to the original image for the naked eye.
- This approach will make the message more secure and let the sender and receiver only learn the text behind the image.

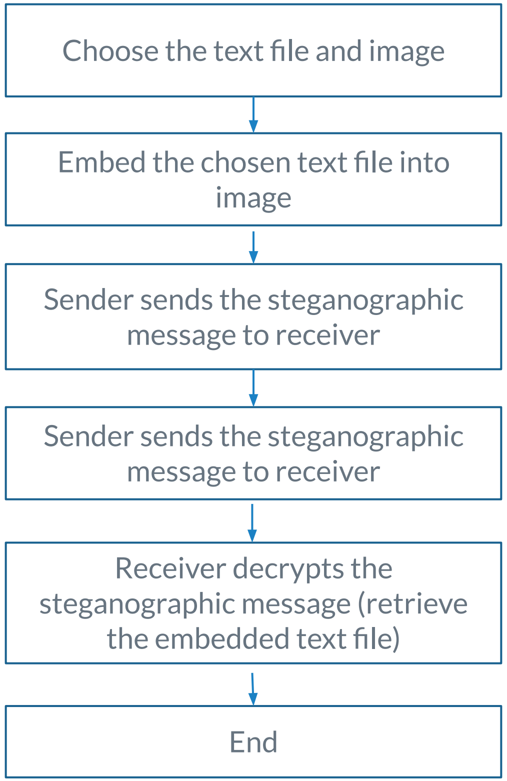
Objective

This Mini Project of ours is objected to build an end-to-end python application that could carry out simple steganography in images .The key functionalities of our project are to:

- Accept a user provided text and a user provided image.
- An Encryption function to produce a new encrypted image(replica) by embedding the text into the image using Steganography.
- A Decrypted function to restore the stored text from the encrypted image.

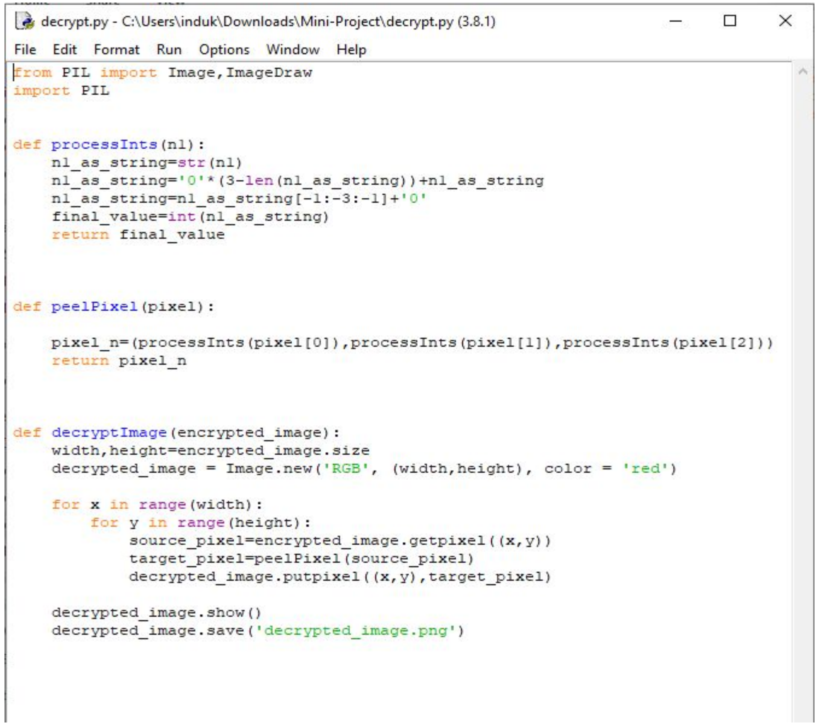
Methodology

Overall Flow Chart of the steganographic message

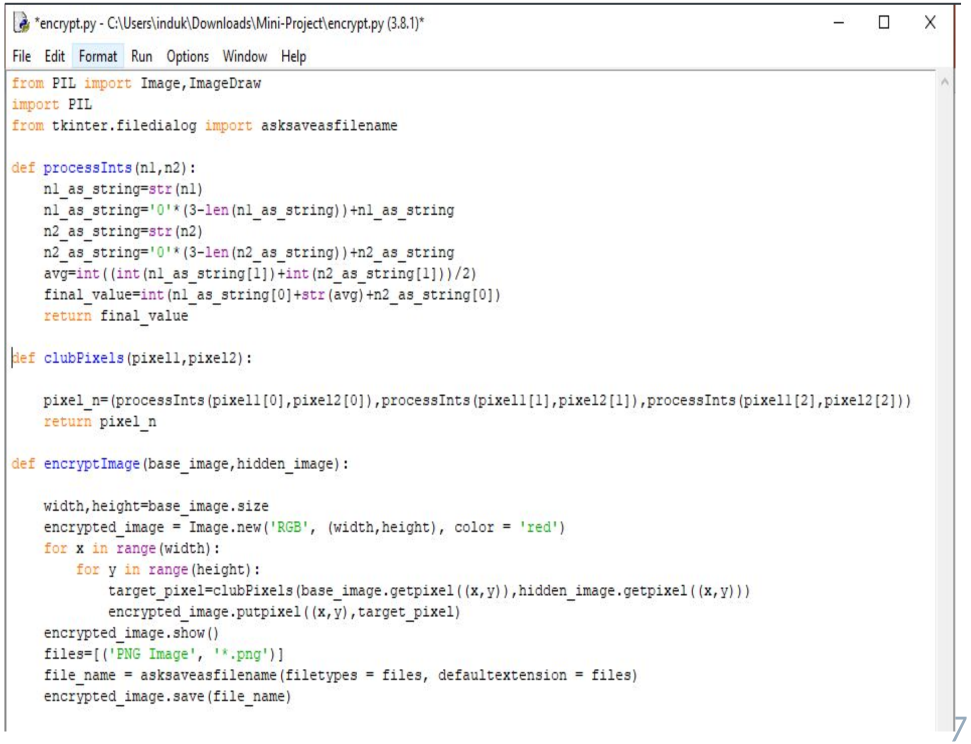

Choose the text file and image
Embed the chosen text file into image
Sender sends the steganographic message to receiver
Sender sends the steganographic message to receiver
Receiver decrypts the steganographic message (retrieve the embedded text file)
End

Implementation

The most important implementations are the encryption and decryption methods:

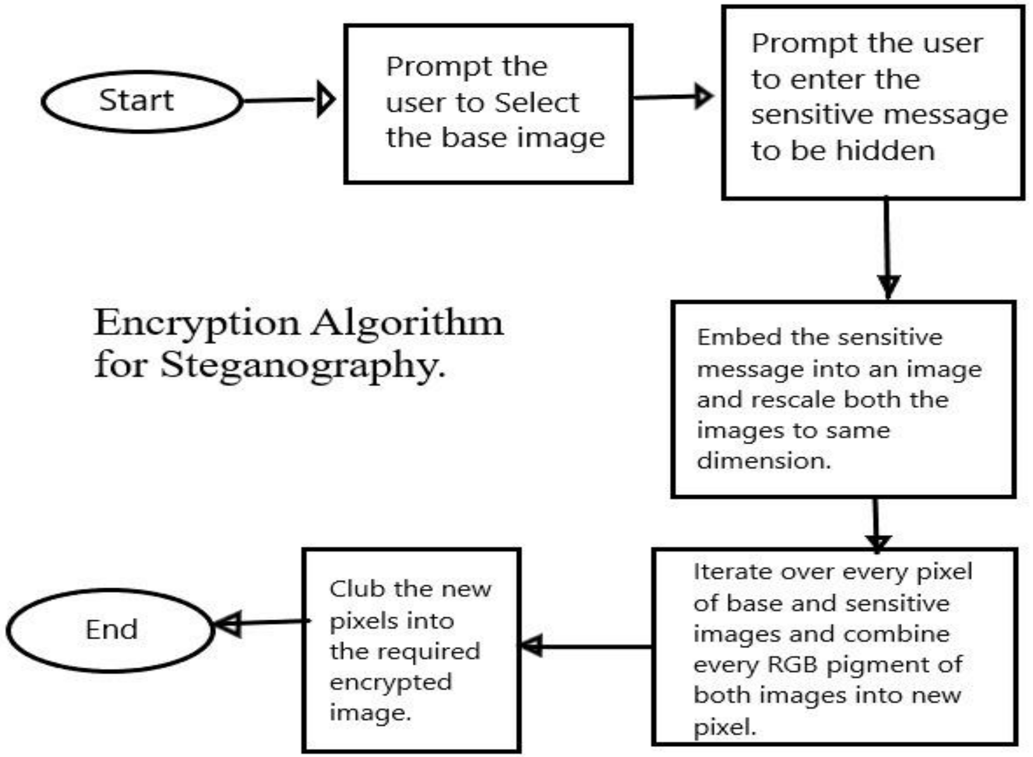

decrypt.py - C:\Users\induk\Downloads\Mini-Project\decrypt.py (3.8.1)
File Edit Format Run Options Window Help

```python
from PIL import Image,ImageDraw
import PIL


def processInts(n1):
    n1_as_string=str(n1)
    n1_as_string='0'*(3-len(n1_as_string))+n1_as_string
    n1_as_string=n1_as_string[-1:-3:-1]+'0'
    final_value=int(n1_as_string)
    return final_value



def peelPixel(pixel):

    pixel_n=(processInts(pixel[0]),processInts(pixel[1]),processInts(pixel[2]))
    return pixel_n



def decryptImage(encrypted_image):
    width,height=encrypted_image.size
    decrypted_image = Image.new('RGB', (width,height), color = 'red')

    for x in range(width):
        for y in range(height):
            source_pixel=encrypted_image.getpixel((x,y))
            target_pixel=peelPixel(source_pixel)
            decrypted_image.putpixel((x,y),target_pixel)

    decrypted_image.show()
    decrypted_image.save('decrypted_image.png')
```


encrypt.py - C:\Users\induk\Downloads\Mini-Project\encrypt.py (3.8.1)
File Edit Format Run Options Window Help

```python
from PIL import Image,ImageDraw
import PIL
from tkinter.filedialog import asksaveasfilename

def processInts(n1,n2):
    n1_as_string=str(n1)
    n1_as_string='0'*(3-len(n1_as_string))+n1_as_string
    n2_as_string=str(n2)
    n2_as_string='0'*(3-len(n2_as_string))+n2_as_string
    avg=int((int(n1_as_string[1])+int(n2_as_string[1]))/2)
    final_value=int(n1_as_string[0]+str(avg)+n2_as_string[0])
    return final_value

def clubPixels(pixel1,pixel2):

    pixel_n=(processInts(pixel1[0],pixel2[0]),processInts(pixel1[1],pixel2[1]),processInts(pixel1[2],pixel2[2]))
    return pixel_n

def encryptImage(base_image,hidden_image):

    width,height=base_image.size
    encrypted_image = Image.new('RGB', (width,height), color = 'red')
    for x in range(width):
        for y in range(height):
            target_pixel=clubPixels(base_image.getpixel((x,y)),hidden_image.getpixel((x,y)))
            encrypted_image.putpixel((x,y),target_pixel)
    encrypted_image.show()
    files=[('PNG Image', '*.png')]
    file_name = asksaveasfilename(filetypes = files, defaultextension = files)
    encrypted_image.save(file_name)
```

Algorithm



Start
Prompt the user to Select the base image
Prompt the user to enter the sensitive message to be hidden
Embed the sensitive message into an image and rescale both the images to same dimension.
Iterate over every pixel of base and sensitive images and combine every RGB pigment of both images into new pixel.
Club the new pixels into the required encrypted image.
End

Encryption Algorithm for Steganography.

Libraries Used:

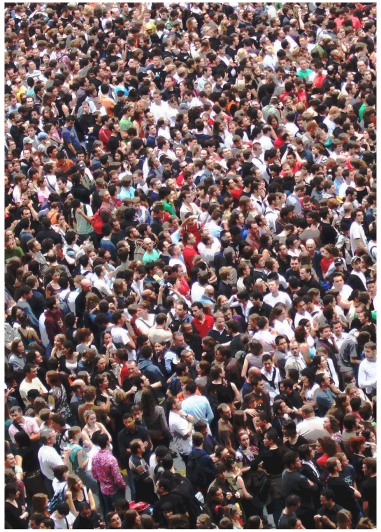
Pre-Defined:

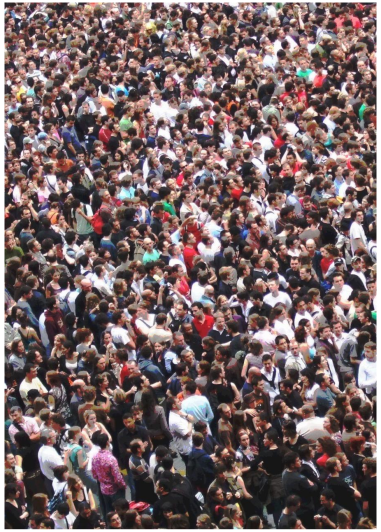
- Tkinter
- PIL
- textwrap
- Image
- ImageFont
- ImageDraw
- Askopenfilename
- asksaveasfilename

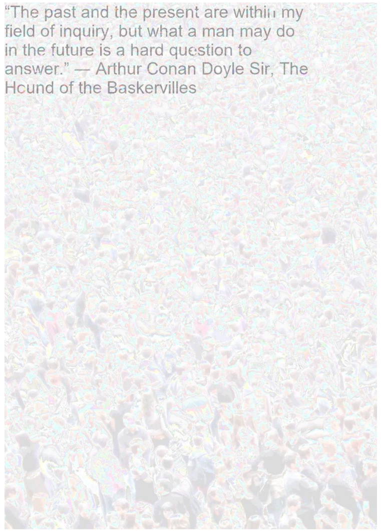
Custom-Designed Modules:

- ProcessInts
- ClubPixels
- EncryptImage
- PeelPixel
- DecryptImage
- Generate_image
- EncryptionController
- DecryptionController

Example


Original Image


Steganography Image


“The past and the present are within my field of inquiry, but what a man may do in the future is a hard question to answer.” — Arthur Conan Doyle Sir, The Hound of the Baskervilles

Decrypted Image

Thanks!

Any questions?