VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Data Hiding in Images using Steganography and obscure transfer/storage using Visual Cryptography

PROJECT REPORT

Submitted by:

SI NO	REG.NO	NAME
1	20BKT0144	CHHAVI JAIN
2	20BCI0141	AKSHAT AGARWAL
3	20BCE0718	ARYAN RAJESH

Prepared for:

Information Security management

(CSE 3502)

PROJECT COMPONENT

Under the guidance of:

Dr. KUMARESAN A

VIT, VELLORE.

School of Information Technology

INDEX:

ABSTRACT

The purpose of this project is to enhance the existing method of embedding text and transferring message and keys (shared) in the form of images. It is the combination of data hiding when passed through the network along with the implementation of obfuscation-security. The pixels ofthe images are modified in such a way that it is difficult to apprehend the text hidden in the image. In short, our aim is to enhance obfuscation in images subjected to steganography when transmitted or stored.

Steganography is easy to analyze and get the result if we've the Image (there is no encryption). To counter this given problem, we've used Visual Cryptography on top of Steganography to make MITM Attack or spoofing more difficult. This project improves security, reliability among the network, efficiency throughout the network.

1. INTRODUCTION

1.1. Theoretical Background

Steganography and cryptography are very important techniques used in data security to hide the secret in transmitted data. Cryptography is a traditional technology which is still in use and is used for attaining the security in our systems and to be aware from the harsh society. Steganography can be stated as a recent technology as compared to cryptography. So, from the concept of cryptography, many other forms of securing ourselves and our data have made their place into the security enthusiasts. The existing solutions that implement either of the technologies are:

a) The use of one of the steganography techniques like LSB Steganography to do our job.

b) Second solution (given in Reference Paper) is suggesting to first encrypt the message using AES (or some other algorithm) and then sending image along with key.

Here we will use AES for encryption. We will embed the message using steganography and then apply visual cryptographic techniques, via encryption mechanisms like AES, on the image.

1.2. Motivation

We as students under the Network Information Security course, were motivated by the subject of security related concepts and wanted to integrate the communication with modified security, after getting to know about the technology of steganography and visual cryptography. The problem statement and the curiosity to eliminate the issues that are faced by them currently, made us motivated all the time to make some research out of it.

1.3. Issues Faced

a) We can use steganography but there is no encryption, we can extract the concealed information by a lot of means. Hence, we somehow need a way to encrypt it in order to truly conceal the secret message.

b) Encrypting the message with AES and then hiding in image will work but there will be a problem with sharing the private key, which will give us another challenge of sharing private keys.

1.4. Contributions of the Project

This project is an attempt to overcome some of the issues of pre-existing techniques of transmission along with security, to go hand in hand. This project has a great scope of contributions to the society for a better understanding between sender and receiver without intervention of invaders. This project can be helpful to any organizations that currently use the steganography technology.

2. LITERATURE SURVEY.

2.1. Survey of the Existing Models/Work

Literature Survey 1:

Design and Implementation of Visual Cryptography for Transmission of Secure Data

Authors: Guru Prasad M Bhat, Nayana G Bhat.

Source: International Journal on Recent and Innovation Trends in Computing and Communication

Content: This paper discusses about the key concepts of visual cryptography, how privacy is protected and steganography. The transmission of the steg-image, the conversion of image to steganographic image, rotation of pixels in the image are discussed.

Literature Survey 2:

Obscurity of Data Using Steganography with Encryption

Authors: Amreen Rahman.

Source: International Journal of Research in Engineering, Science and Management (IJRESM)

Content: This paper discusses about the security implementation using obscurity and encryption. It discusses about the types of steganography and methods involved in each type of steganography like Phase coding, LSB coding, spread spectrum, and echo hiding. It describes the steganography method using LSB method.

Literature Survey 3:

Combine use of Steganography and Visual Cryptography for Secured Data hiding in Computer Forensics

Authors: Ravindra Gupta, Akanksha Jain, Gajendra singh.

Source: International Journal of Computer Science and Information Technologies (IJCSIT)

Content: This paper made us to understand our problem statement much better and paved the way to the solution to the current existing problems of the methodology. A shared image combination for reveal of hidden message inside the shares is the new concept of idea that is involved in this research. This discusses about some known algorithms and combining them with visual cryptography that made the system more secure and robust.

Literature Survey 4:

High Embedding capacity data hiding technique based on EMSD and LSB substitution algorithms

Authors: Sedar Solak

Source: IEEE Access

Content: The purpose of steganography is to obtain a good stego-image. This discusses about Enhanced modified Signed digit algorithm along with Least significant bit substitution. The proposed algorithm discussed here was the various pattern of combination of different types of algorithms present in the steganography mechanism. Discusses about the basic features to be present in the mechanism. High embedding capacity and image quality are to be maintained in the overall transmission.

Literature Survey 5:

A Robust and Secured Image Steganography using LSB and Random Bit Substitution

Authors: Md. Ehasn Ali, Md. Sohrawordi, Md. Palash Uddin.

Source: American Journal of Engineering Research (AJER)

Content: Here, the concept of steganography has been classified into three ways, Pure, secret key and public key. Some metric of measurements like Mean squared errors and Peak signal to noise ratio are discussed. The proposed method of image steganography hides the message bit in random position and hides the references like LSB and others which are used in the process.

Literature Survey 6:

A Methodology Based on Steganography and Cryptography to Protect Highly Secure Messages

Authors: Rashad J Rasras, Ziad A AlQadi, Mutaz Rasmi Abu Sara

Source: Engineering, Technology &Applied Science Research

Content: This literature describes itself by dividing into three parts, steganography, cryptography, and message extraction (reverse of producing a steg-image and decrypting the message, in perspective of the project). This deals with color images, the time taken for the process, LSB implementation, comparisons with histogram.

Literature Survey 7:

Information hiding in images using Steganography techniques.

Authors: Anoop Kumar

Source: EasyChair Print

Content: The different types of steganography and their classifications are mentioned. The techniques including LSB and Hide Seek are well described for the readers. It mentions the use of different existing tools and their drawbacks to which the proposed system has to overcome. The provision of security, reliability, feasibility and maintainability of the steganography mechanism are given a place in the paper.

Literature Survey 8:

High capacity reversible and secured data hiding in images using interpolation and difference expansion technique

Authors: Pratap Chandra Mandal, Imon Mukherjee, Biswa Nath Chatterji

Source: Springer Link

Content: A new 2-layer secure, high-capacity reversible data hiding technique has been proposed, using the concept of interpolation-based data hiding and difference expansion method. This technique offers high-capacity data hiding, by considering the maximum difference between neighboring pixels without compromising on quality of the images.

Literature Survey 9:

Advanced 3-Bit LSB Based on Data Hiding Using Steganography

Authors: Kinjal Patani, Dushyantsinh Rathod

Source: Springer link

Content: In this paper, authors have used a 3-bit least significant bits technique for embedding secrete image into the cover image, which is called as Stego image. ECC algorithm is used to kept data more secure and secrete which makes very hard to get secrete message even if the secrete message is disclose.

Literature Survey 10:

Information hiding in images using Discrete Cosine Transform

Authors: G Emmanuel1, G G Hungil1, J Maiga1 and A J Santoso

Source: IOP conference series: Material Science and Engineering

Content: This paper presents the study which hides text messages into the image file by using Discrete Cosine Transform (DCT) steganography method. The method continues to keep the quality of the cover image which transfers the information to reduce suspicion.

2.2. Problem definition:

The issues that are observed while researching that in the existing methodologies and mechanisms, the data is encrypted and put into the image using cryptography and steganography respectively. So here, the key has to be shared in another secret medium through the publicmedium. Instead of having a key, the idea of removing the key is the problem definition here.The concept of shares, the mechanism of embedding the key within the image, etc. were some of the ideas that are empowered during the process of defining the problem.

The issues that are currently faced by the existing users made our problem definition stronger and foundational.

2.3. Aim of the Proposed Work:

The proposed solution is going to enhance the method of first encrypting the message and embed in image along with key. We won't encrypt text and then embed it using steganography. We will directly embed text using steganography (like LSB steganography) and then we will use Visual Cryptography to encrypt the image and produce the share images (say n share images). No need to send key, we will send share images and secret message can only be revealed if an individual has all the share images.

3. OVERVIEW OF WORK

3.1 Objectives of the Project:

Steganography is really famous when it comes to hiding secret data/information inside images. The problem that we have found during the research is to achieve sending secret information/data by hiding inside an image with obscurity. The main objective of the project is to present the ways to avoid the problem for achieving security in communication. The implementation and research with a keen desire on learning the principles and to apply in reallife as objectives made the project lifecycle exciting.

3.1. Software Requirements

Requires a python Environment to execute the proposed program. Also, a terminal of any operating system and a code editor to develop the code.

Python tools and modules/libraries required for the application to run:

- ✓ numpy==1.14.3
- ✓ Pillow==6.2.2
- ✓ streamlit==0.56.0

3.2. Hardware Requirements

We are here considering the message to be sent between two parties, must be put inside the image using LSB steganography. The data that has to be shared in between the parties involvedin the communication are the shared images (which contains message and keys). The applicationis easy to use and user-friendly to modify to their own needs and requirements. It is feasible to fulfill the technical requirements of at least 1GB RAM, any entry level processor.

4. SYSTEM DESIGN

4.1 Introduction and Related Concepts:

An overview:

There are tools and methods of encrypting the image in today's world but there are not enough tools to provide a robust security to that encrypted image. Here by the method of static analyzing the pixel values of the image, we can get the hidden message inside the image after applying Stereographic manipulation. This is done by one of the most famous algorithms, RS analysis. With the use of LSB replace, it was easy to detect the hidden message. Digital watermarking wasalso one of the popular conventional methods in this field of securing the data to be conveyed. Steganalysis must be conducted for every image that is to be made for conveying the secret data. RS Steganalysis was far better than LSB Steganalysis as it suffered from many attacks in the

recent times.

This proposed system will have a message and an image as cover to this message. Message will be stored in the LSB of the image. The pixel characteristics and properties have to be maintained after the embedding of secret message. This image is now called as “steg-image”. Here in LSB, the binary representation of the secret message replaces the least significant bits in the cover image to for steg-image. According to Kerchoff’s principle of assumption, only key is kept secretbut not the algorithm. This key serves the purpose of finding the location of pixel of the secret message.

Bitmap images are used generally for this. The simple implementation of the LSB technique is called as BlindHide. As this is more prone to attack, an application is used to randomize the location of the bits. Here in this application, HideSeek, an advanced of version of BlindHide is used. But the changes made by these two mechanisms are visible to naked eye, which is not the idea to implement. So, Laplace formula is used here. This project consists of two important algorithms, Genetic algorithm steganography and Visual cryptography. The steg-image here is the input for encryption using visual cryptography.

4.2 Algorithm:

The process of making algorithm makes the work flow faster and easier. The algorithm for our proposed concept is as follows:

Read image and text from sender. Implement any of the present steganography techniques on them. There are many random number generations, swapping and many in the algorithm. There are some ways in the execution which modify the chromosomes of the image. Correlation is also calculated accordingly. And then embedding is done. After that visual cryptography comes into the execution which takes in the returned steg-image as input. And at receiver’s side, extractionis done. After the image pass through this module, shares of the image are produced. The shares are generated in such a way that they have to be merged for the construction of the decrypted hidden message inside the shares all together. In case of decoding the message, pixel expansion effect is eliminated because of implementing the random number generation and their allocation on the cover image.

4.3. Proposed System Architecture:

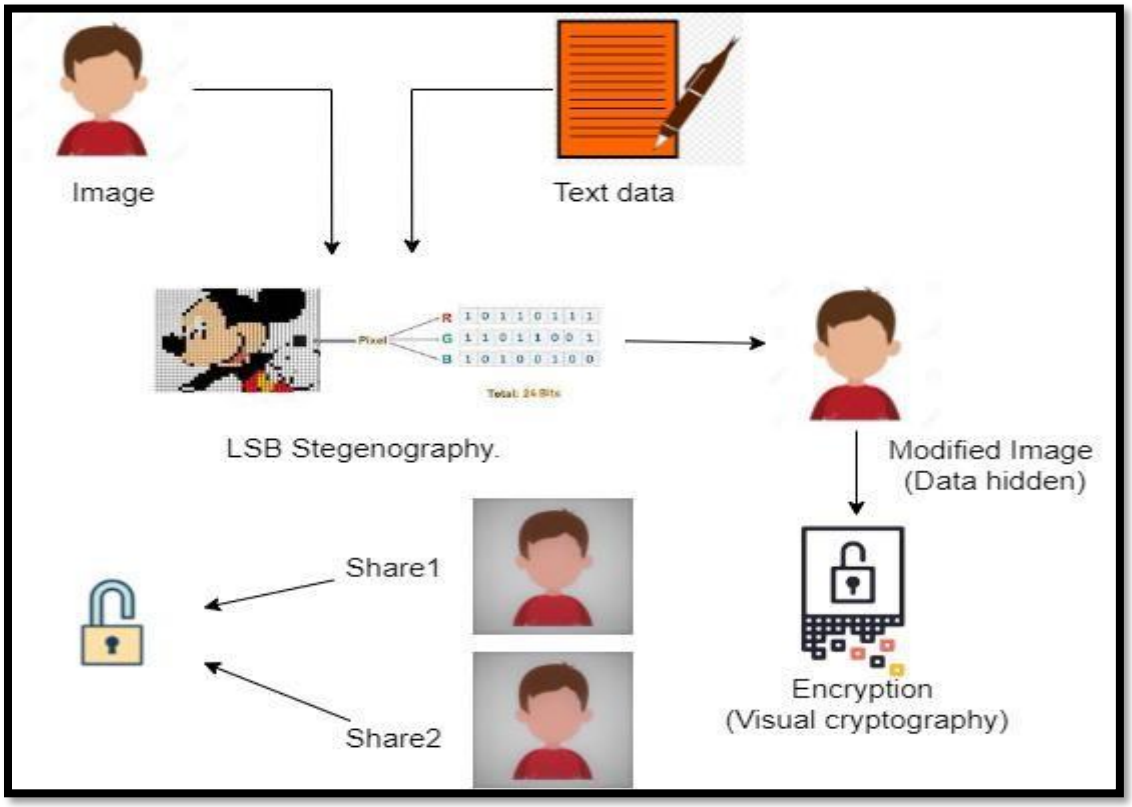

Fig 4.3.1

5. IMPLEMENTATION

5.1. Description of modules/libraries used:

This project uses python programming language as the language for core development of the mechanism. The packages that are used in the project are:

- ✓ numpy==1.14.3.: - A popular library/module used for array/matrix manipulations and various linear algebra operations.
- ✓ Pillow==6.2.2.: - This package/library is used for image manipulations and adds imageprocessing capabilities to the python interpreter.
- ✓ streamlit==0.56.0.: - A library/app framework to create beautiful web apps in a short timeso that nothing to worry about the user interface.
- ✓ sys module: - This built-in module provides various functions and variables that are used tomanipulate different parts of the Python runtime environment.

5.2. Proposed Source/System Code:

Lsb_stegno.py

```
import numpy as np
from PIL import Image

# Convert encoding msg into 8-bit binary
# form using ASCII value of characters


def charToBinList(msg):
    # list of binary codes
    # of given msg
    l = []
    for i in msg:
        l.append(format(ord(i), "08b"))
    return l


# Pixels are modified according to the
# 8-bit binary msg and finally returned


def modPix(pix, msg):
    datalist = charToBinList(msg)
    lendata = len(datalist)
    img_data = iter(pix)
    for i in range(lendata):
        # Extracting 3 pixels at a time
        pix = [
            value
            for value in img_data.__next__()[:3]
            + img_data.__next__()[:3]
            + img_data.__next__()[:3]
        ]
        print(pix)
        # 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):
                pix[j] -= 1
        # Ninth pixel of every set tells
        # whether to stop or to read further.
        # 0 means keep reading; 1 means the
        # message is over.
        if i == lendata - 1:
            if pix[-1] % 2 == 0:
```

```
                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(new_img, msg):
    w = new_img.size[0]
    (x, y) = (0, 0)
    # print(list(new_img.getdata()))
    print(list(new_img.getdata())[:15])
    for pixel in modPix(new_img.getdata(), msg):
        # Putting modified pixels in the new image
        new_img.putpixel((x, y), tuple(pixel))
        if x == w - 1:
            x = 0
            y += 1
        else:
            x += 1
    # print(list(new_img.getdata())[:15])


# Encode msg into image


def lsb_encode(msg):
    try:
        image = Image.open("images/img.jpg", "r")
    except:
        image = Image.open("images/img.png", "r")
    new_img = image.copy()
    encode_enc(new_img, msg)
    return new_img


def lsb_decode(file_name):
    image = Image.open(file_name, "r")
    msg = ""
    imgdata = iter(image.getdata())
    while True:
        pixels = [
            value
            for value in imgdata.__next__()[:3]
            + imgdata.__next__()[:3]
            + imgdata.__next__()[:3]
        ]
```

```
        # string of binary msg
        binstr = ""
        for i in pixels[:8]:
            if i % 2 == 0:
                binstr += "0"
            else:
                binstr += "1"
        msg = msg + chr(int(binstr, 2))
        if pixels[-1] % 2 != 0:
            return msg
```

n_share.py

```
import numpy as np
from PIL import Image
def generate_shares(data, share=2):
    data = np.array(data, dtype='u1')
    # Generate image of same size
    img1 = np.zeros(data.shape).astype("u1")
    img2 = np.zeros(data.shape).astype("u1")
    # Set random factor
    for i in range(data.shape[0]):
        for j in range(data.shape[1]):
            for k in range(data.shape[2]):
                n = int(np.random.randint(data[i, j, k] + 1))
                img1[i, j, k] = n
                img2[i, j, k] = data[i, j, k] - n
    img1 = Image.fromarray(img1)
    img2 = Image.fromarray(img2)
    img1.save("images/pic1.png", "PNG")
    img2.save("images/pic2.png", "PNG")


def compress_n_join_shares(img1="images/share1.png", img2="images/share2.png"):
 # Read images
    img1 = np.asarray(Image.open(img1)).astype('int16')
    img2 = np.asarray(Image.open(img2)).astype('int16')
    img = np.zeros(img1.shape)
    # Fit to range
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            for k in range(img.shape[2]):
                img[i, j, k] = img1[i, j, k] + img2[i, j, k]
 # Save compressed image
    img = img.astype(np.dtype('u1'))
    img = Image.fromarray(img)
    img.save("images/compress.png", "PNG")
```

main.py

```python
from src.n_share import generate_shares, compress_n_join_shares
from src.lsb_stegno import lsb_encode, lsb_decode
from PIL import Image
import streamlit as UI
import os
import sys

sys.path.insert(0, "./src")

option = UI.sidebar.radio("Options", ["Docs", "Encode & Split", "Merge & Decode"])
if option == "Docs":
    UI.title("Documentation")
    with open("README.md", "r") as f:
        docs = f.read()
    UI.markdown(docs, unsafe_allow_html=True)
elif option == "Encode & Split":
    UI.title("Encoding")
    # Image
    img = UI.file_uploader("Upload Cover Image", type=["jpg", "png", "jpeg"])
    if img is not None:
        img = Image.open(img)
        try:
            img.save("images/img.jpg")
        except:
            img.save("images/img.png")
        UI.image(
            img,
            caption="Selected image to use for data encoding",
            use_column_width=True,
        )
    # Data
    txt = UI.text_input("Enter Message to hide")
    # Encode message
    if UI.button("Encode data and Generate shares"):
        # Checks
        if len(txt) == 0:
            UI.warning("No data to hide")
        elif img is None:
            UI.warning("No image file selected")
        # Generate splits
        else:
            generate_shares(lsb_encode(txt))
            try:
                os.remove("images/img.jpg")
            except FileNotFoundError:
```

```
                os.remove("images/img.png")
            UI.success(
                "Data encoded using Steganography and splitted into two shares using Visual
Cryptography :)!"
            )
elif option == "Merge & Decode":
    UI.title("Decoding")
    # Share 1
    img1 = UI.file_uploader("Upload Share 1", type=["png"])
    if img1 is not None:
        img1 = Image.open(img1)
        img1.save("images/share1.png")
        UI.image(img1, caption="Share 1", use_column_width=True)
    # Share 2
    img2 = UI.file_uploader("Upload Share 2", type=["png"])
    if img2 is not None:
        img2 = Image.open(img2)
        img2.save("images/share2.png")
        UI.image(img2, caption="Share 2", use_column_width=True)
    # Decode message
    if UI.button("Merge Shares into one Compressed image and Decode message"):
        # Check
        if img1 is None or img2 is None:
            UI.warning("Upload both shares")
        # Compress shares
        else:
            compress_n_join_shares()
            os.remove("images/share1.png")
            os.remove("images/share2.png")
            UI.success("Decoded message: " + lsb_decode("images/compress.png"))
```

5.3. Test cases:

All kinds of images (greyscale, png, jpg, jpeg) are tested under the normal conditions which satisfy the hardware and software requirements as mentioned in the previous sections. All kinds of input (characters, digits, special characters, numbers, etc.…) are also tested under the same andthe application has achieved a 100% accuracy. But there is a slight latency and delay in case if decryption and getting the secret message and it can be made much more efficient with the use ofdata structures to store the pixel values for the purpose of calculations.

Input text	**Input image type**	**Output images (shares)**	**Output text.**	**Duration.**
Hello	img.png and then pic1.png, pic2.png	share1.png, share2.png, and then compress.png	Hello	0.3s
Networking Information Security	img.png and then pic1.png, pic2.png	share1.png, share2.png, and then compress.png	Networking and Security.	1s
dfvdgdfgdfgdfsdfdsfsdypojkypjofvsdfsdfsfsdfsdfsdfsdffs	img.png and then pic1.png, pic2.png	share1.png, share2.png, and then compress.png	dfvdgdfgdfgdfsdfdsfsdypojkypjofvsdfsdfsfsdfsdfsdfsdffs	3s

Table 5.3.1

6. OUTPUT AND PERFORMANCE ANALYSIS

6.1 Execution snapshots:

Encoding :

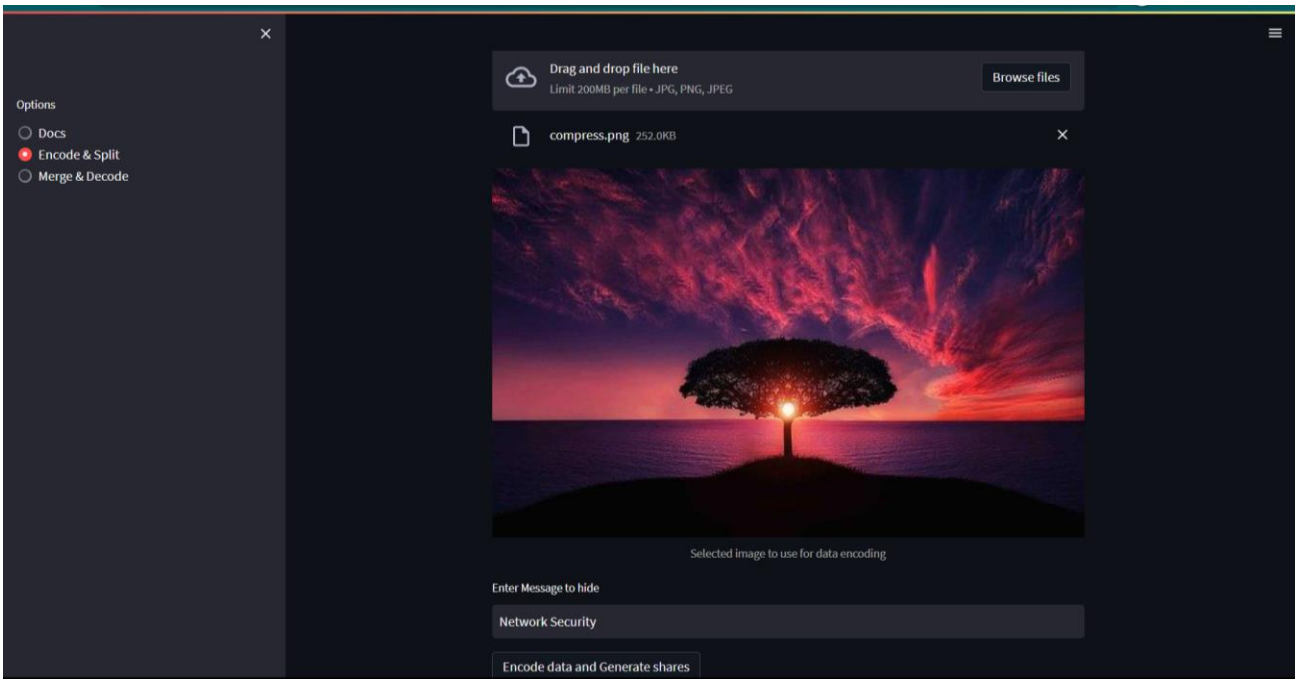

Fig 6.1.1

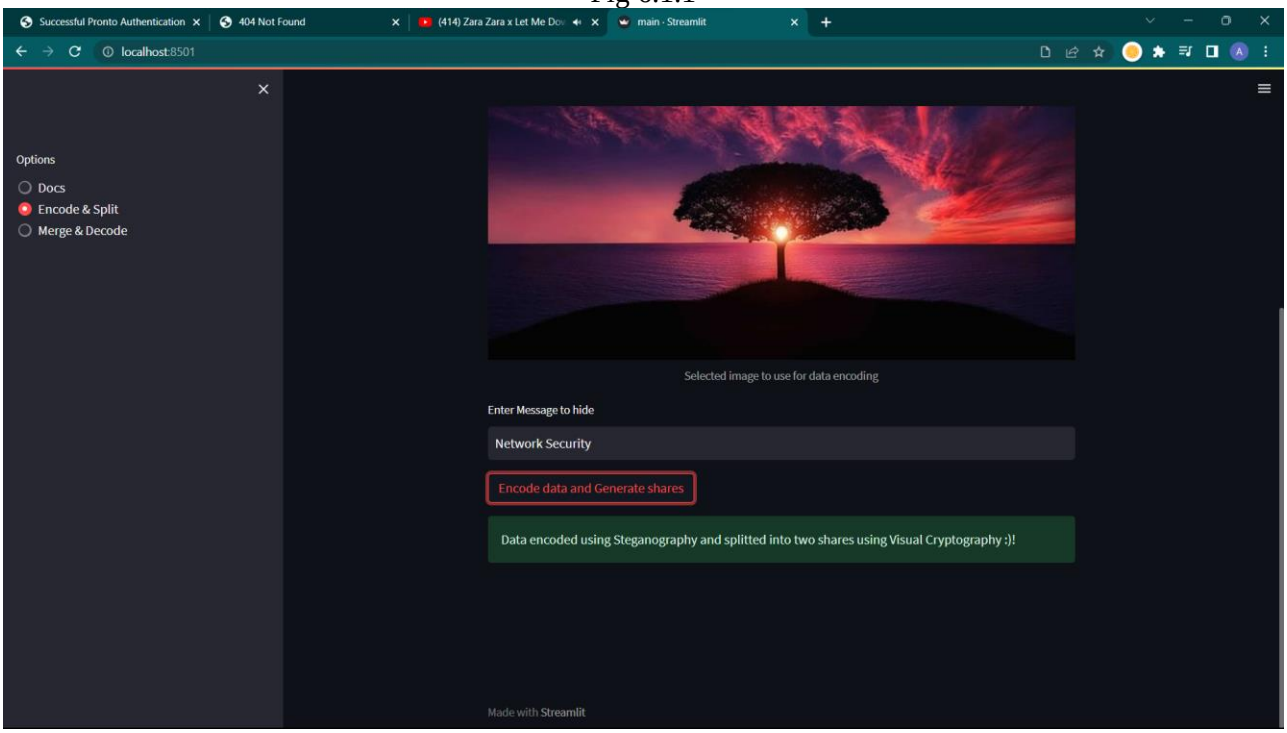

Fig 6.1.2

Decoding :

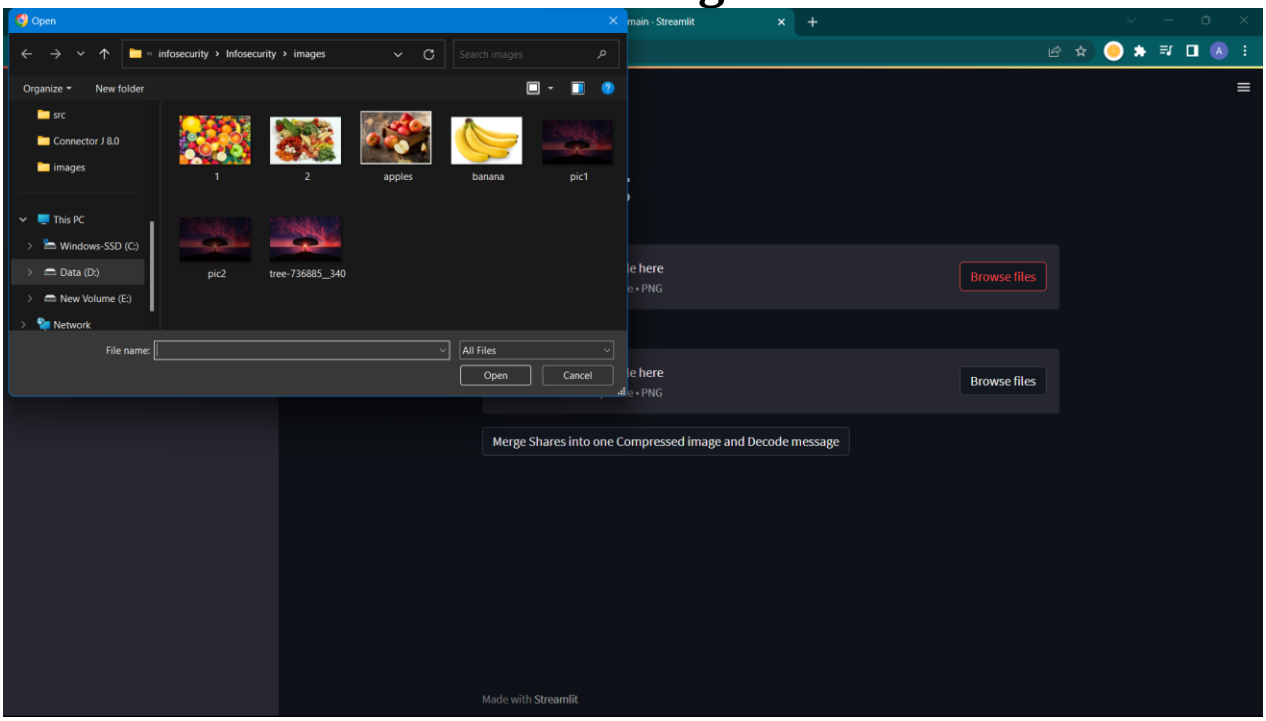

Fig6.1.3

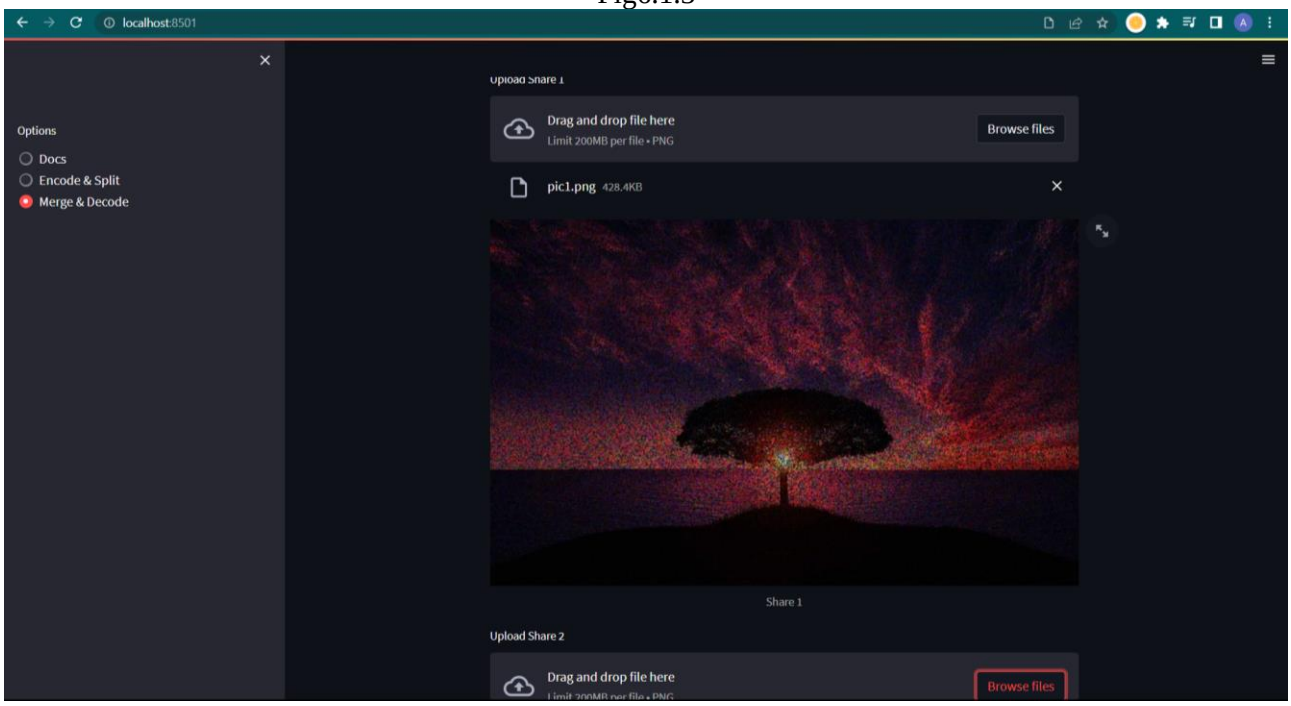

Fig 6.1.4

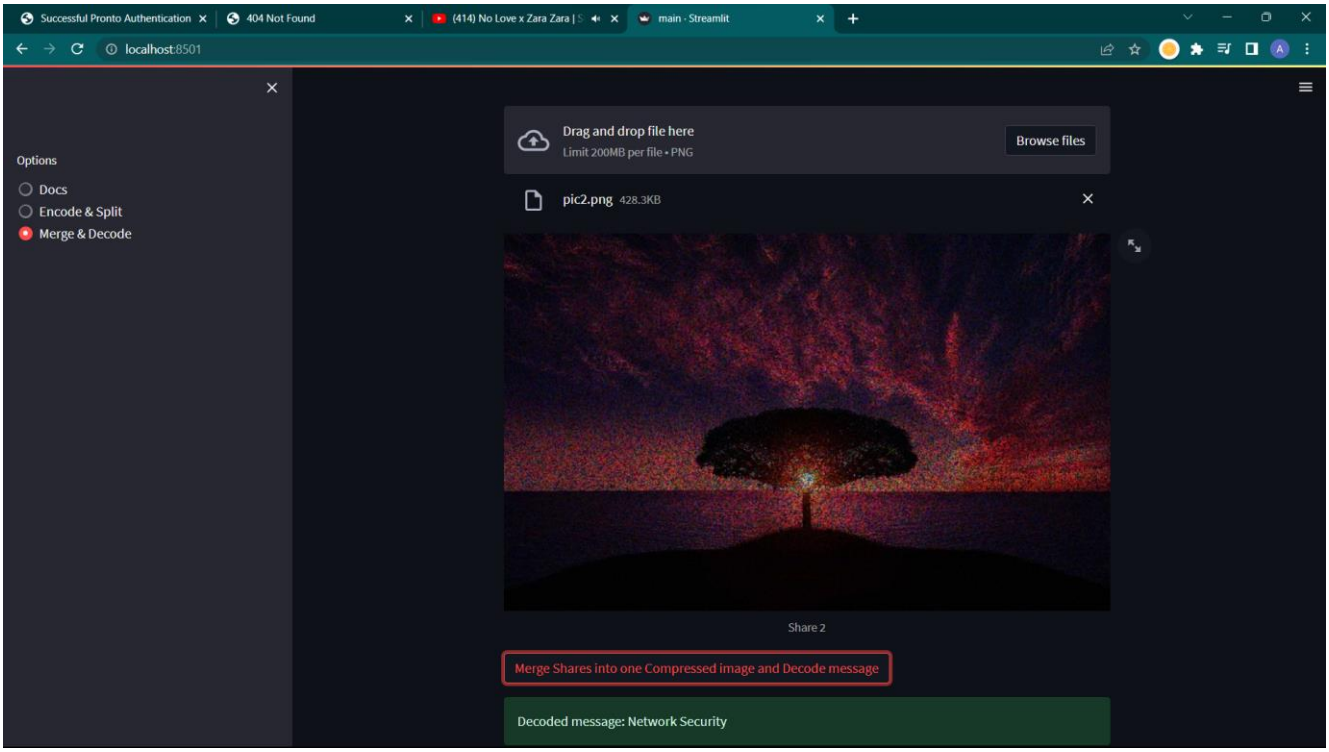
Successful Pronto Authentication
404 Not Found
(414) No Love x Zara Zara |
main - Streamlit
localhost:8501
Options
Docs
Encode & Split
Merge & Decode
Drag and drop file here
Limit 200MB per file • PNG
Browse files
pic2.png 428.3KB
Share 2
Merge Shares into one Compressed image and Decode message
Decoded message: Network Security

Fig 6.1.5

6.2 Output – in terms of performance metrics:

The performance metrics graph is as shown below:

Figure 6.2.1 Legend: X-axis represents the count of share images to be generated. Y-axis represents the time taken for decrypting the secret message from the shares generated.

Figure 6.2.2 Legend: X-axis represents length of the secret message that is to be embedded by the user to hide in image using the application mechanism. Y-axis represents the time taken for decrypting the secret message from the shares generated.

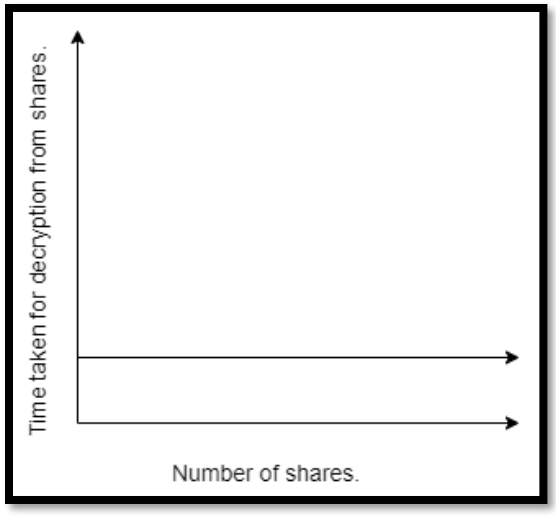

Fig 6.2.1

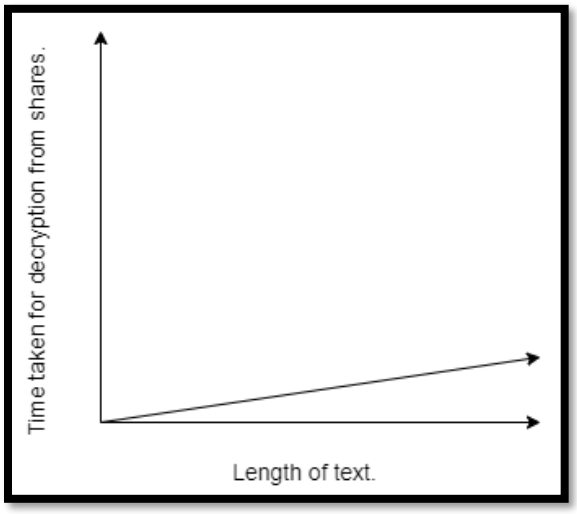

Fig 6.2.2

6.3 Performance comparison with existing works:

The existing works have a different workflow as compared to this methodology. But while comparing some of the basic processes like inputting the message, embedding into image, etc. are up to the mark in terms of performance. This project can go hand in hand with the existing projects and will not hamper the performance of the parent process/mechanism.

7.CONCLUSION AND FUTURE DIRECTIONS

This technique is resistant to RS attacks because this technique doesn't depend on the encryption of LSB of pixel values. As the concept of shares is introduced here, the decryption is very difficult than to imagine. This mechanism is optimal for both grey-scale and colored images.

This kind of approach may prove this system to be much more secure and robust against attacks if integrated with neural networks for generation of difficult shares of the encrypted image.

8.REFERENCES.

Image references:

1. 4.3.1 – Architecture. (Self-Made using app.diagrams.net)
2. 6.1.1, 6.1.2, 6.1.3, 6.1.4, 6.1.5 – Output screenshots
3. 6.2.1, 6.2.2 – Performance metrics.

Table references:

1. 5.3.1 – Test cases.

Papers/Journal references:

1. Bhat, Guru Prasad M., and Nayana G. Bhat. "Design and Implementation of Visual Cryptography System for Transmission of Secure Data." International Journal on Recent and Innovation Trends in Computing and Communication 5.7: 718-721.
2. Rahman, Amreen. "Obscurity of Data Using Steganography with Encryption." International Journal of Research in Engineering, Science and Management 4.1 (2021): 133-136.
3. Gupta, Ravindra, Akanksha Jain, and Gajendra Singh. "Combine use of steganography and visual cryptography for secured data hiding in computer forensics." International Journal of Computer Science and Information Technologies 3.3 (2012): 4366-4370.
4. Solak, Serdar. "High Embedding Capacity Data Hiding Technique Based on EMSD and LSB Substitution Algorithms." IEEE Access 8 (2020): 166513-166524.
5. Ali, U. A. M. E., Md Sohrawordi, and Md Palash Uddin. "A Robust and Secured Image Steganography using LSB and Random Bit Substitution." American Journal of Engineering Research (AJER) 8.2 (2019): 39-44.
6. Rasras, Rashad J., Ziad A. AlQadi, and Mutaz Rasmi Abu Sara. "A methodology based on steganography and cryptography to protect highly secure messages." Engineering, Technology & Applied Science Research 9.1 (2019): 3681-3684.
7. G G Hungil, J Maiga and A J Santoso, ‘‘Information hiding in images using Discrete Cosine Transform’’, IOPP Publishing Ltd IOP Conference Series: Materials Science and Engineering, Volume 1098, Environmental Engineering (2021) IOP Conf. Ser.: Mater. Sci. Eng. 1098 052083
8. Pratap Chandra Mandal, Imon Mukherjee, Biswa Nath Chatterji, ‘‘High capacity reversible and secured data hiding in images using interpolation and difference expansion technique’’, Multimedia Tools and Applications (2021)
9. Kinjal Patani, Dushyantsinh Rathod, ‘‘Advanced 3-Bit LSB Based on Data Hiding Using Steganography’’, Conference Paper, Springer Link (2020)
10. Yadav R, Nigam S (2018) An improve color image steganography using ECC algorithm. IJRRETAS 4(4). ISSN: 2455-4723

For details see Section 2.1 [Literature survey]