DATA HIDING AND STORING USING STEGANOGRAPHY AND VISUAL CRYPTOGRAPHY

A PROJECT REPORT

Submitted by

T.SNEHITHA, 20BCE0858
THALLA DEEPAK, 20BCE2061
ISHIKA GARG, 20BCE2793

Course Code: CSE 3501

Course Title: INFORMATION SECURITY ANALYSIS AND AUDIT

Under the guidance of

Dr. Kakelli Anil Kumar
Associate Professor
SCOPE, VIT, Vellore.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

November-2022

INDEX

Abstract

The goal of this project is to improve the current technique for embedding text and sending shared messages and keys as images. It consists of the installation of obfuscation-security together with the concealment of data as it travels via the network. The pixels of the pictures are changed so that it is challenging to catch the text hidden in the image. To put it plainly, our point is to upgrade muddling in pictures exposed to steganography when transmitted or put away.

As we know there are many ways to analyse steganography and it's also simple to get the result if the image is available. This lead to a lot of problems regarding the safety of information.To solve this problem we are using Visual cryptography on top of steganography. This leads to reduce in the MITM Attack or Spoofing attacks. Therefore this project helps in improving the security and reliability among the network and increasing its efficiency throughout the network.

1. Introduction

1.1.Theoretical Background

Steganography and cryptography are the best techniques that can be utilized in information security to hide the secret in communicated information. Cryptography is a conventional innovation which is still being used and will be used for achieving the security in our frameworks and to know from the brutal society. Compared to cryptography, steganography can be considered a more recent technology. Therefore, many more methods of protecting ourselves and our data have emerged from the concept of cryptography, gaining the attention of security enthusiasts.

The best solutions existing for implementing the above techniques are

a) Using LSB Steganography as one of the steganography methods to carry out our work .

b) The second option is encrypting the message with AES (or another algorithm) before sending the image and key. Here, we'll encrypt using AES. Steganography will be used to hide the message, and then visual cryptography will be used to encrypt the image using methods like AES.

1.2.Motivation

After learning about the steganography and visual cryptography technologies, we who are pursuing the information security course, got inspired by the topic of security-related concepts and sought to integrate communication with modified security. The problem statement and our desire to solve the problems they are now experiencing kept us motivated to conduct some study on it.

1.3. Issues Faced

Even without the help of encryption we can use steganography to recover the hidden information using a variety of methods. Therefore, in order to truly conceal the secret message, we need to find a technique to encrypt it.

While using AES to encrypt the message and then disguising it in an image will work, sharing the private key will be problematic, which will present another obstacle.

1.4. Contributions of the Project

This project aims to address some of the problems that pre-existing transmission and security methods have when used together. This research provides a wide range of potential benefits to society for improved communication between sender and receiver without outside interference. Any organisations that now use the technology of steganography can benefit from this effort.

2. Literature Survey

2.1. Research papers and Journals

Name of the paper	Authors	Source	Content
Design and Implementation of Visual Cryptography for Transmission of Secure Data	Guru Prasad M Bhat, Nayana G Bhat.	International Journal on Recent and Innovation Trends in Computing and Communication	This paper discusses the key concepts of visual cryptography, how Privacy is protected and steganography. The transmission of the steg-image, the conversion of image toste genographic image, rotation of pixels in the image are discussed.
Obscurity of Data Using Steganography with Encryption	Amreen Rahman	International Journal of Research in Engineering, Science and	This paper discusses the security implementation using obscurity and encryption. It discusses the types of

		Management (IJRESM)	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 the LSB method.
Combine use of Steganography and Visual Cryptography for Secured Data hiding inComputer Forensics	Ravindra Gupta, Akanksha Jain, Gajendra singh.	International Journal of Computer Science and Information Technologies (IJCSIT)	This paper made us 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 messages inside the shares is the new concept of idea that is involved in this research. This discusses some known algorithms and combines them with visual cryptography that makes the system more secure and robust.
High Embedding capacity data hiding technique based on EMSD	Sedar Solak	IEEE Access	The purpose of steganography is to obtain a good stego-image. This discusses

and LSBsubstitution algorithms			Enhanced modified Signed digit algorithm along with Least significant bit substitution. The Proposed algorithm discussed here was the various patterns 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

A Robust and Secured Image Steganography using LSB and RandomBit Substitution	Md. Ehasn Ali, Md. Sohrawordi, Md. Palash Uddin.	American Journal of Engineering Research (AJER)	Here, the concept of steganography has been classified into three ways, Pure, secret key and public key. Some metrics of measurements like Mean squared errors and Peak signal to noise ratio are discussed. The proposed method of image steganography hides the message it in random position and hides the references like LSB and others which are used in the process
Research on Various Cryptography Techniques	Yahia Alemami, Mohamad Afendee Mohamed, Saleh Atiewi	International Journal of Recent Technology and Engineering	Security plays a critical position in preserving information privacy and secrecy. Many encryption strategies are available to protect data during transmission or storage. These encryption methods vary in terms of strength, speed, and resource consumption (CPU usage, memory, and power). This study aims to present the most popular and interesting algorithms currently in

			use.
Concepts Of Cryptography And Cryptographic Hash Function	Dr. R.K Gupta	European Journal of Molecular & Clinical Medicine	We start with basic concepts of cryptography and move towards its history. Main concentration is on various algorithms including DES, RSA. Here we also discussed cryptographic hash functions- MD family, SHA family and RIPEMD, BLAKE and WHILPOOL families also and finally we wind up the paper comparison
A Review Paper on Cryptography	Abdalbasit Mohammed Qadir, Nurhayat Varol	2019 7th International Symposium on Digital Forensics	With the internet having reached a level that merges with our lives, growing explosively

		and Security	during the last several decades, data security has become a main concern for anyone connected to the web. Data security ensures that our data is only accessible by the intended receiver and prevents any modification or alteration of data.
Cryptography: from the ancient history to now, it's applications and a new complete numerical model	S. M. Naser	International Journal of Mathematics and Statistics Studies	In this computer and internet based modern era, people have to deal with private information in thousands way. Today's prime concern is to keep secure the information of different channel and medium related to one's and to communicate securely, and to do so cryptography method is the sole key to it. This research paper will briefly lighten on the history of cryptography, basic definitions related to cryptography and some basic theorems to build different types of cryptography models

A RESEARCH PAPER ON CRYPTOGRAPHY	Gurdeep Singh, Prateek Kumar, Nishant Taneja, Gurpreet Kaur	International Journal For Technological Research In Engineering	Data is any type of stored digital information. Security is about the protection of assets. Data security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, personal databases and websites. Cryptography is evergreen and developments
A Methodology Based on Steganography and Cryptography to Protect HighlySecureMessages	Rashad J Rasras, Ziad A AlQadi, Mutaz Rasmi Abu Sara	Engineering, Technology & Applied Science Research	This literature describes itself by dividing into three parts, steganography, cryptography, and message extraction (reverse of producing a stage-image and decrypting the message, in perspective of the project).This deals with colour images, the time taken for the process, LSB implementation, comparisons with histogram
Information hiding in images using	Anoop Kumar	EasyChair Print.	The different types of steganography and their classifications are

Steganography techniques.			mentioned. The Techniques including LSB and HideSeek 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.

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 public medium. 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 proposed work

The proposed solution is going to enhance the method of first encrypting the message and embed the image along with the 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 shared images (say n share images). No need to send a key, we will send shared images and a secret message can only be revealed if an individual has all the shared images.

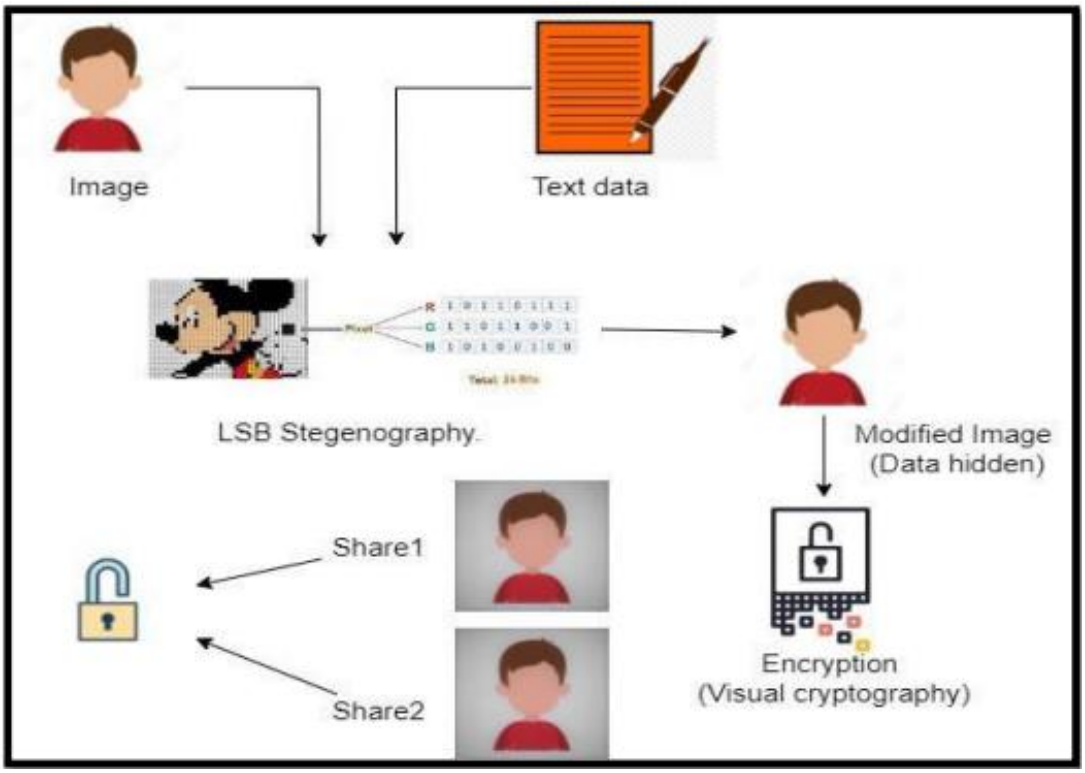
3. Overview of the Work

3.1. Problem description

When it comes to concealing sensitive information or data inside of photographs, steganography is extremely well-known. The challenge that we have discovered during our research is to communicate secret information/data by obfuscating it inside of an image.

The project's primary goal is to outline potential solutions for ensuring communication security. Implementation and research driven by a strong desire to understand the fundamentals and put them into practiseThe project lifetime was made entertaining by life as objectives.

3.2. Working model



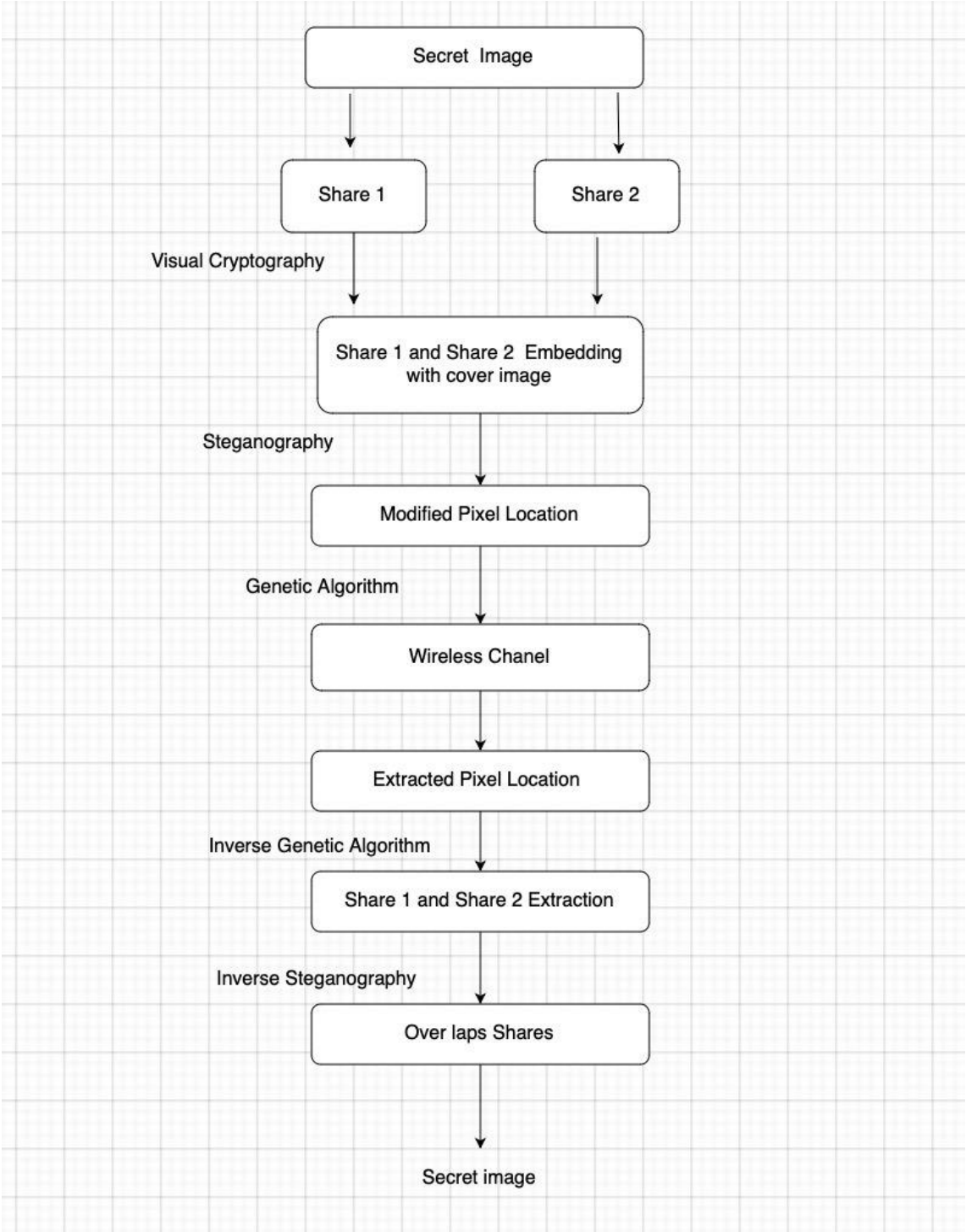
3.4. Algorithms

The creation of algorithms speeds up and simplifies the job flow. The following is the algorithm for our proposed concept:

Read the sender's text and image. Use any of the available steganography methods on them. The algorithm uses several random number generation, swapping, and generation processes. The chromosomes of the image can be altered in a few different ways throughout execution. Accordingly, correlation is also calculated. Finally, embedding is completed. Visual cryptography is then put into action, using the returned steg-image as input.

Extraction is completed at the receiver's side. Shares of the image are created after the image has gone through this module. The shares are created in a way that requires them to be combined in order to build the decrypted concealed message inside the shares.

3.5 Flowchart



4. Implementation

4.1. Description of Modules/Programs

Python is the primary language used in this project to create the themechanism. The following packages are utilised by the project:

- ✓ **numpy==1.14.3.** A well-liked library or module for manipulating arrays, matrices, and performing different linear algebra operations.
- ✓ **Pillow==6.2.2.** The Python interpreter now has image processing capabilities thanks to this package/library that is used for image manipulation.
- ✓ **streamlit==0.56.0.** A library or app framework to quickly construct stunning web apps with no need to care about the user interface.
- ✓ **Sys module.** This built-in module offers a number of functions and variables that are used to manipulate various Python components.

4.2. Software Requirements

The provided programme executes only in a Python environment. Also required for developing the code are a terminal on any operating system and a code editor.

The following Python tools and libraries are necessary for the application to run:

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

4.3. Hardware Requirements

We are thinking about the message that needs to be hidden inside an image utilising LSB steganography in order to be transferred between two parties. The shared photos are the data that must be exchanged between the parties engaged in communication (which contains message and keys).

The programme is simple to use and adaptable to the demands and specifications of the user. The technical requirements of at least 1GB RAM and any entry level processor can be met.

4.4. Source code

```
#################################################################
########################## main.py##############################
#################################################################
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"))
```

```
##########################################################################
############################## 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")
    img3 = 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)
    img3 = Image.fromarray(img3)
    img1.save("images/pic1.png", "PNG")
    img2.save("images/pic2.png", "PNG")
    img3.save("images/pic3.png", "PNG")


def compress_n_join_shares(img1="images/share1.png", img2="images/share2.png",
img3="images/share3.png"):
 # Read images
    img1 = np.asarray(Image.open(img1)).astype('int16')
    img2 = np.asarray(Image.open(img2)).astype('int16')
    img3 = np.asarray(Image.open(img3)).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] + img3[i, j, k]
# Save compressed image
    img = img.astype(np.dtype('u1'))
    img = Image.fromarray(img)
    img.save("images/compress.png", "PNG")
```

4.5. Test cases

All types of images, including grayscale, png, jpg, and jpeg, are examined under typical circumstances, providing they meet the hardware and software requirements outlined in the preceding sections.

The application has attained 100% accuracy for all input types examined (characters, digits, special characters, numerals, etc.). However, there is a tiny latency and delay when decrypting and obtaining the secret message.

This can be significantly reduced by using data structures to store the pixel values for computations.

Input Text	Input Image Type	Output Images(shares)	Output Text	Duration
Hi	Img.png and then Pic1.png, pic2.png	Share1.png, share2.png and then compress.png	Hi	0.1s
Prof. K Anil Kumar	Img.png and then Pic1.png, pic2.png	Share1.png, share2.png and then compress.png	Prof. K Anil Kumar	1.s
Information Security Analysis and Audit Prof. K Anil Kumar	Img.png and then Pic1.png, pic2.png	Share1.png, share2.png and then compress.png	Information Security Analysis and Audit Prof. K Anil Kumar	5s

4.6. Execution of the project

4.6.1 Execution snapshots

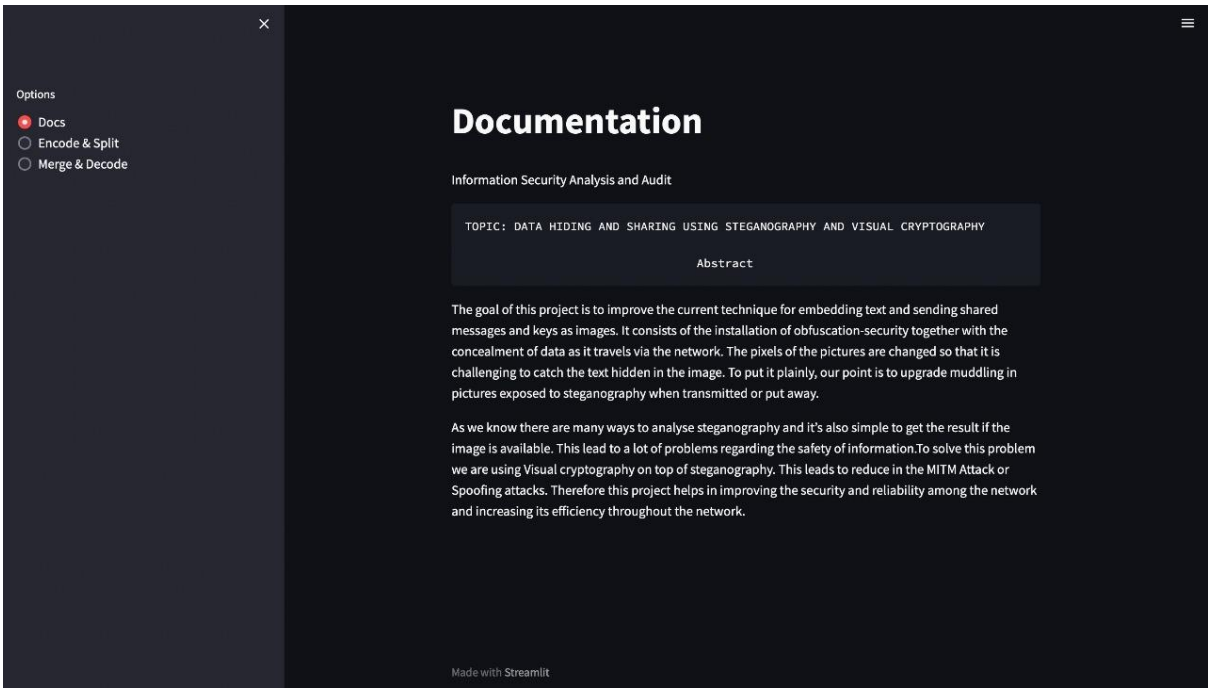


Fig 6.1.1

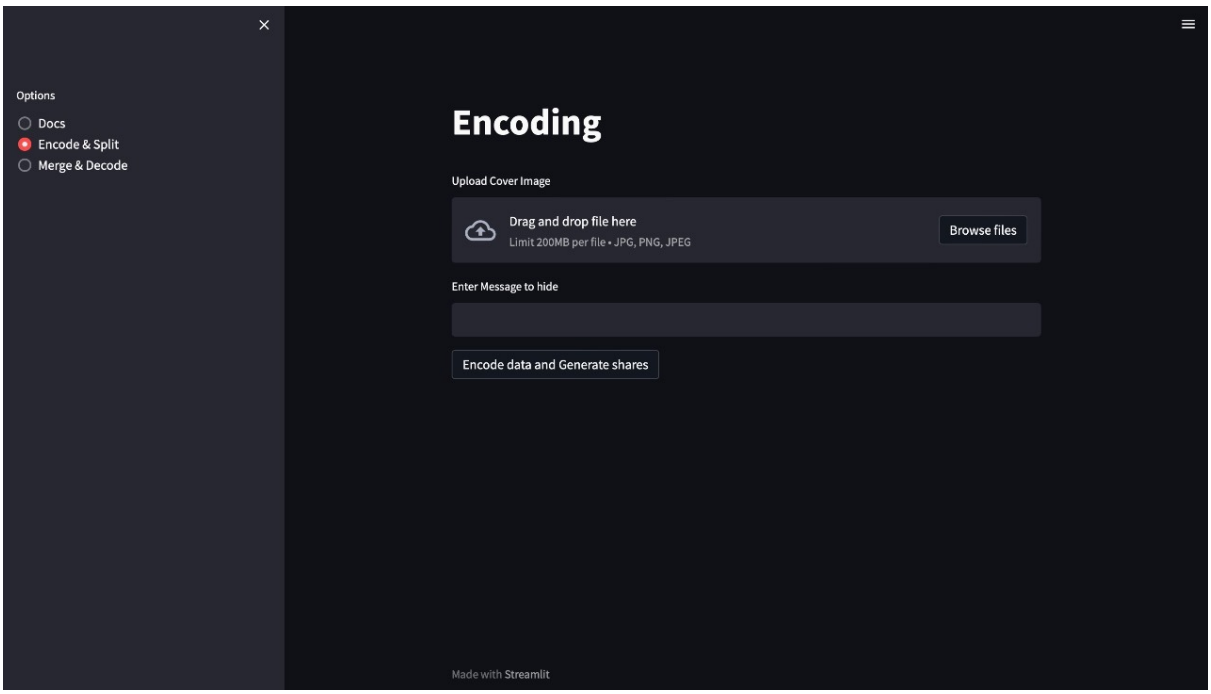


Fig 6.2.1

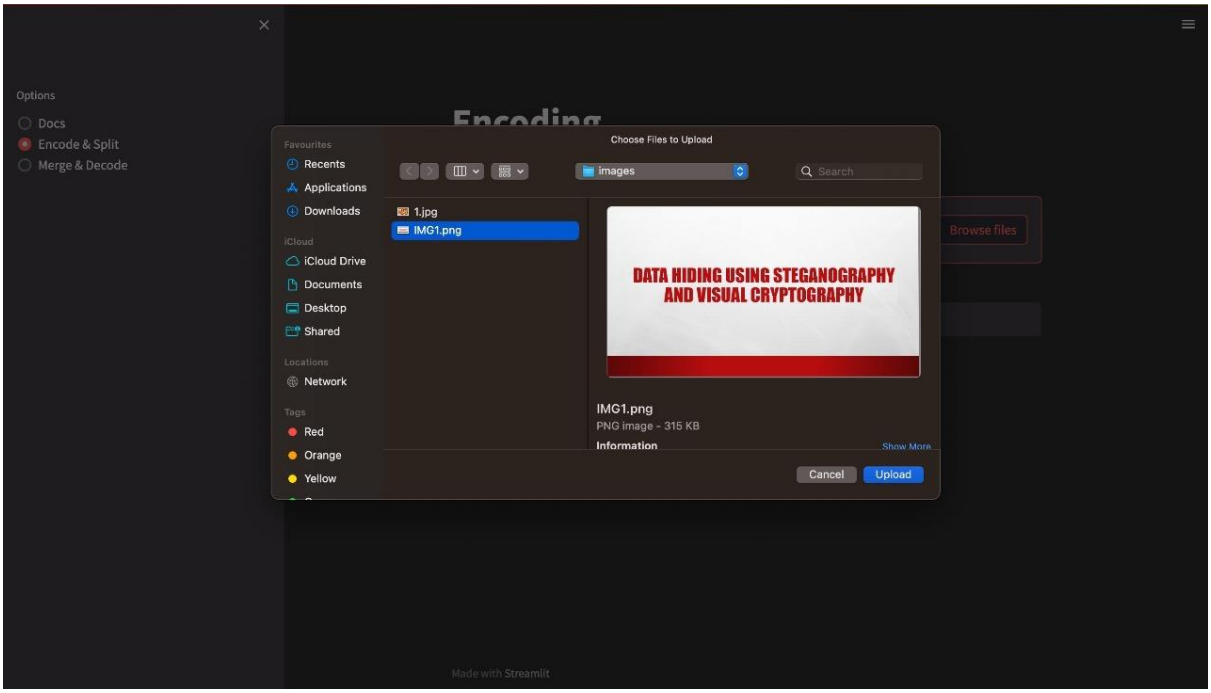


Fig 6.1.3

Image is splitted into two shares

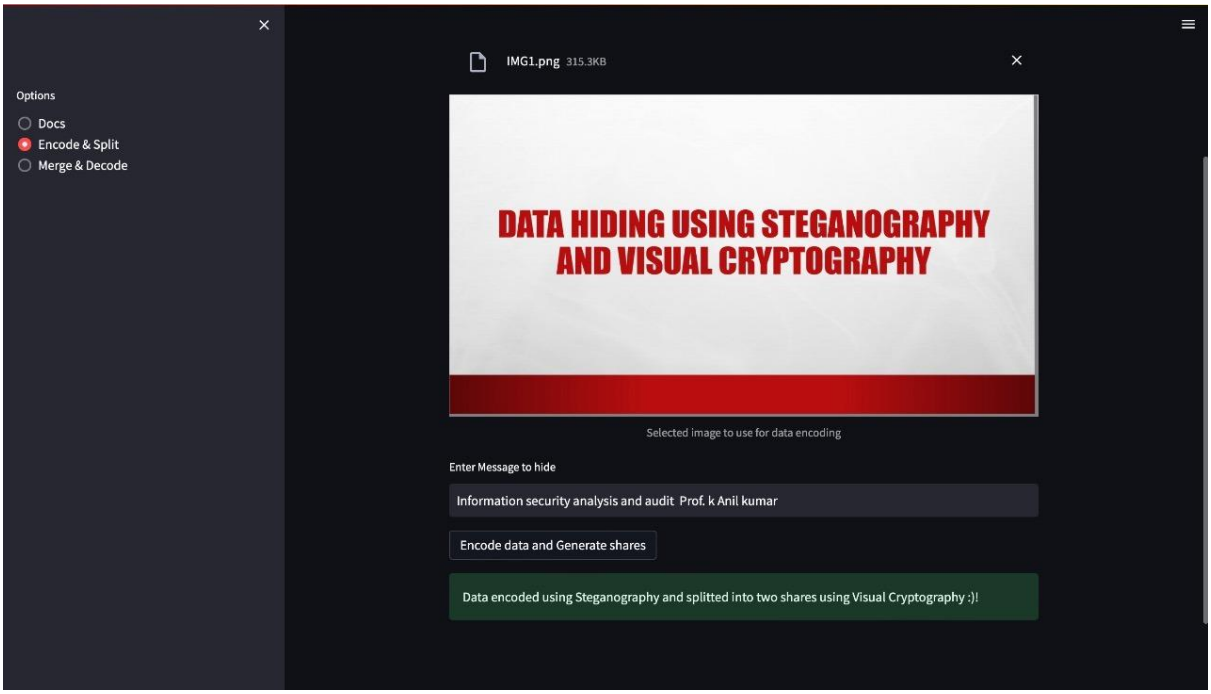


Fig 6.1.4

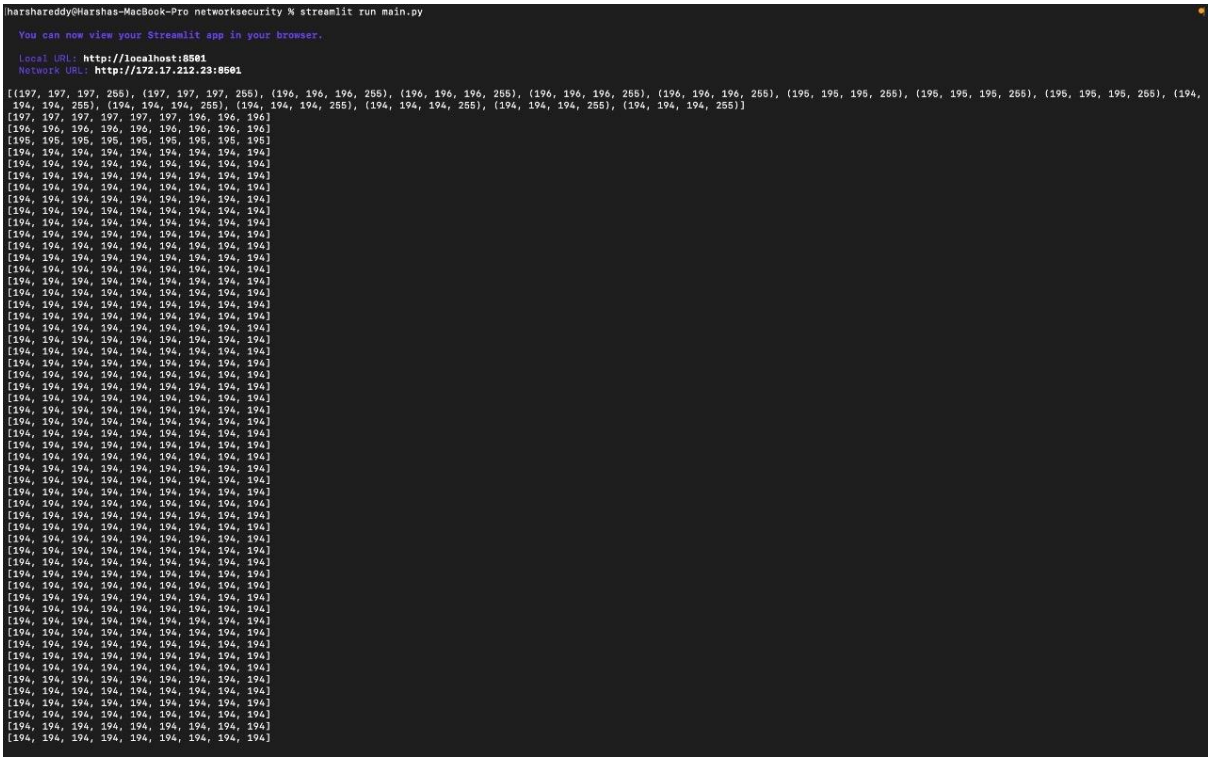


Fig 6.1.5

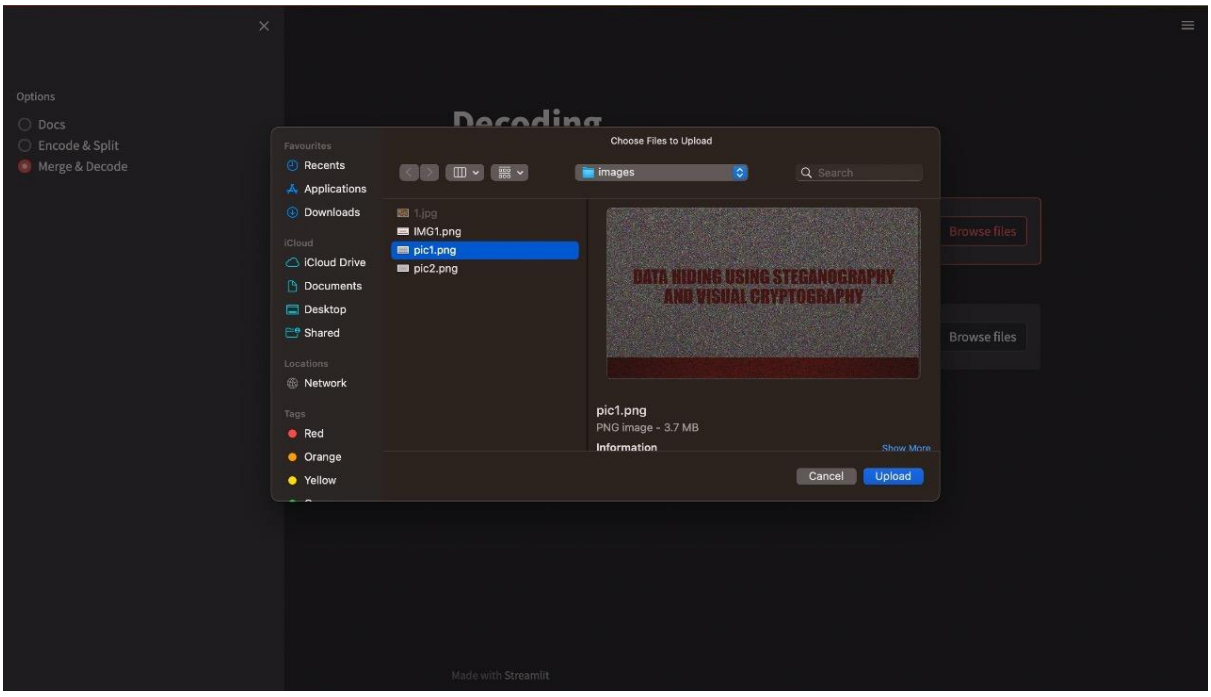


Fig 6.1.6

Message decoded and printed

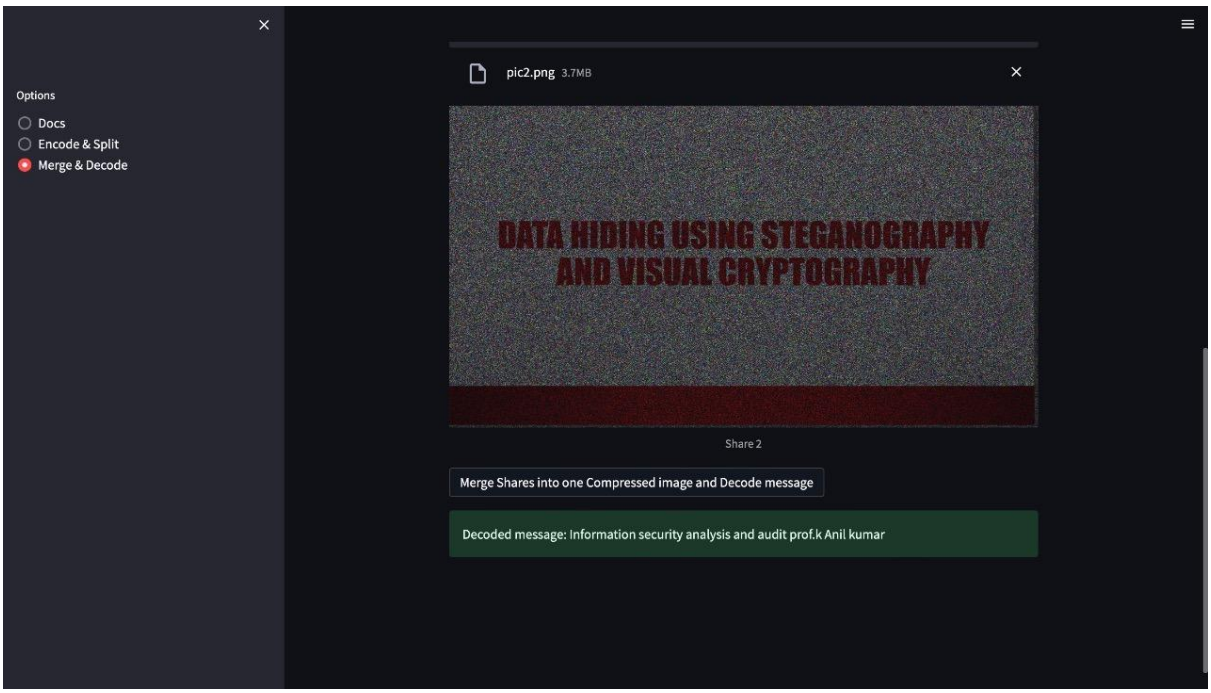


Fig 6.1.7

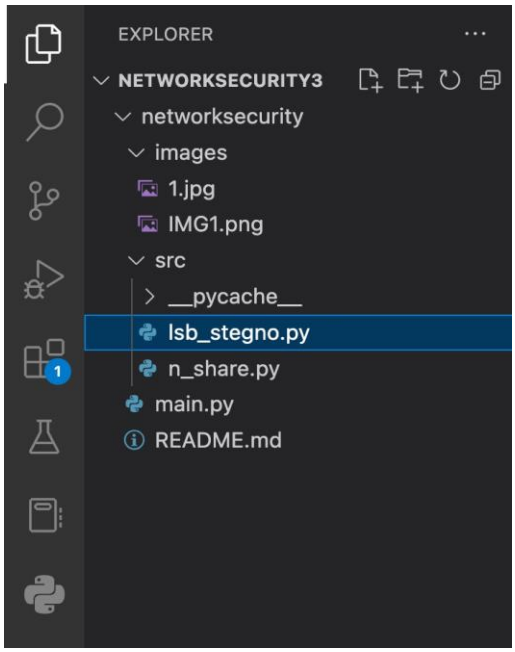


Fig 6.1.8

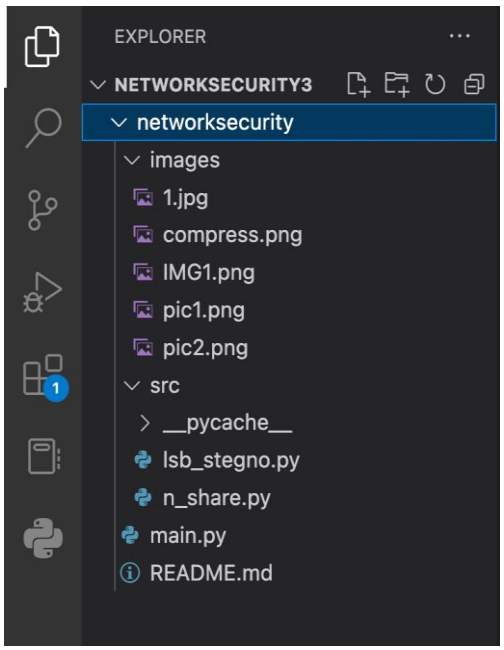


Fig 6.1.9

Image split into three shares

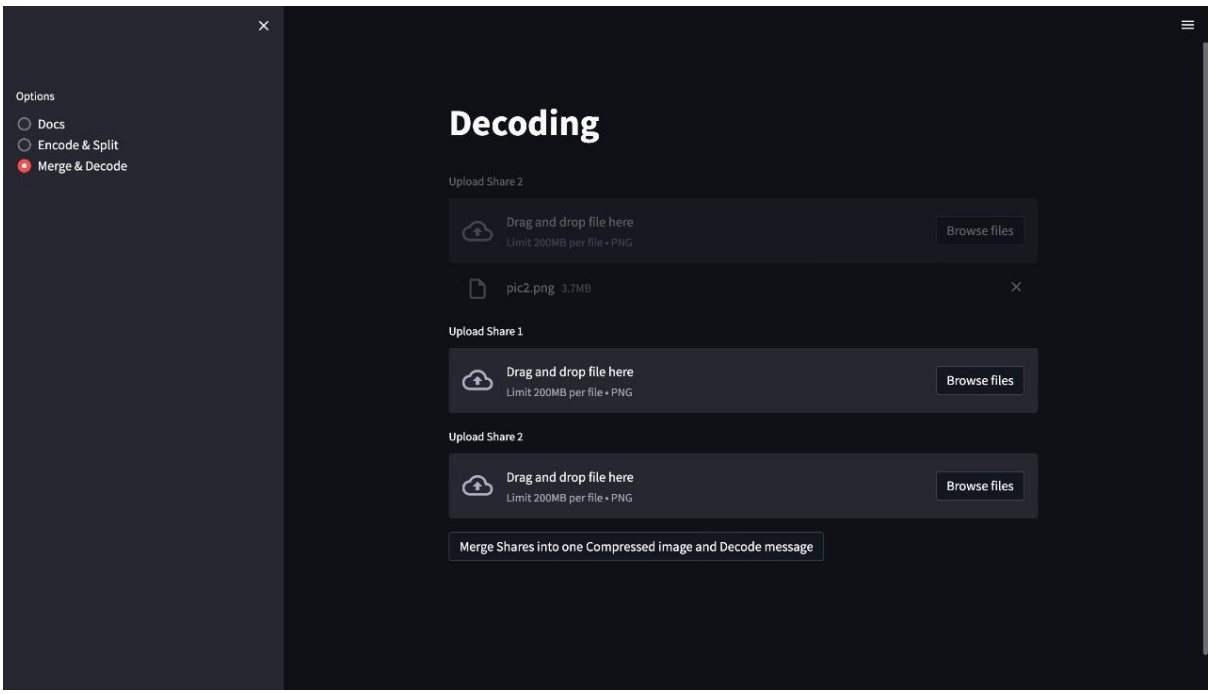


Fig 6.2.0

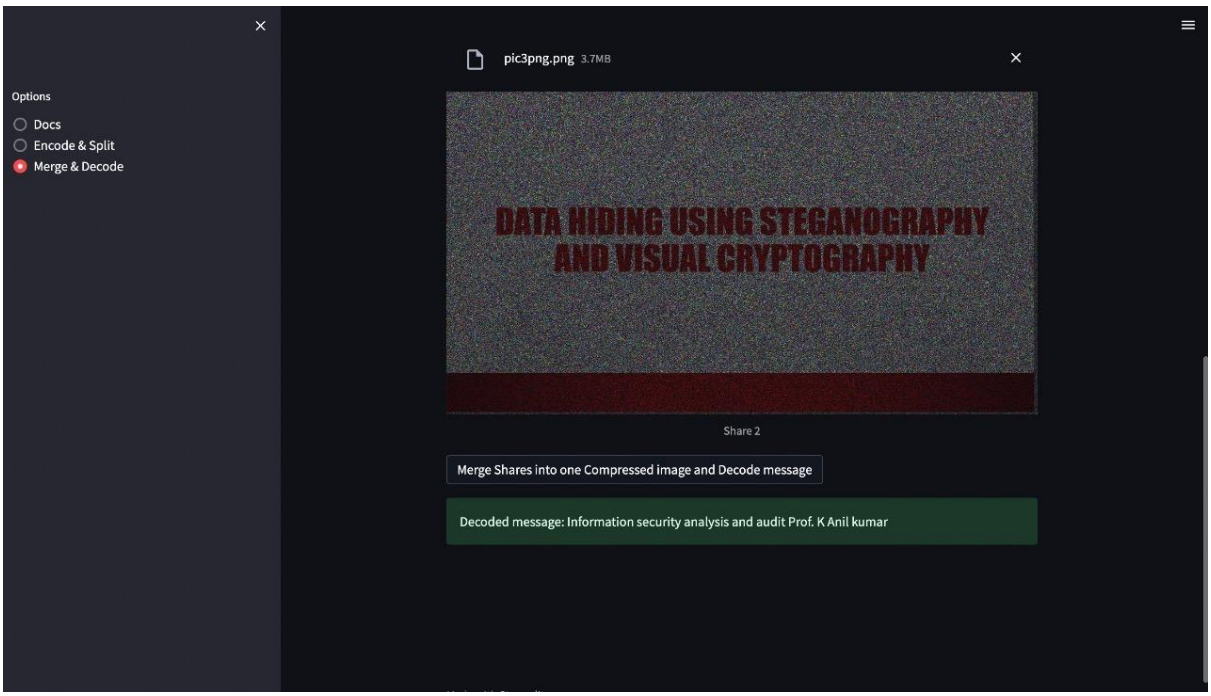


Fig 6.2.1

4.8 Output- in terms of performance metrics

The performance metrics graph is displayed below. The number of share photos that will be generated is shown on the X-axis in Figure 6.2.1. The time required to decrypt the secret message from the shares produced is shown on the Y-axis.

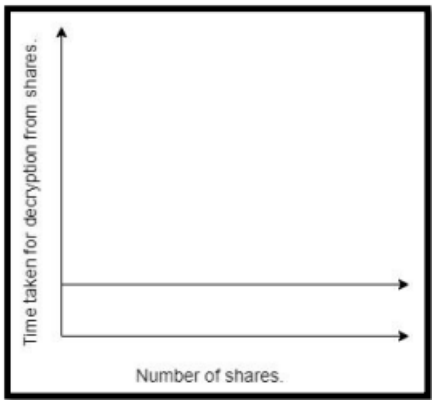


Fig 6.2.3

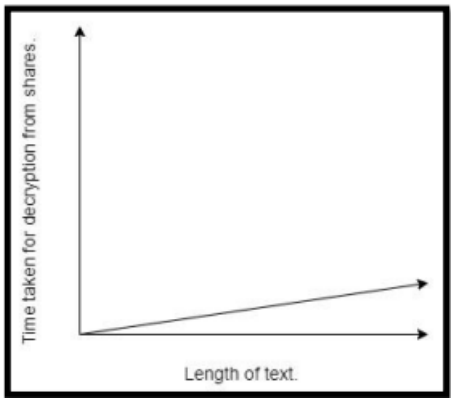


Fig 6.2.4

Graph 6.2.3 Legend: X - axis represents the length of the secret message that must be embedded by the user and concealed in an image utilising the application technique. Y- axis represents the time required to decrypt the secret message from the shares produced.

5. CONCLUSION AND FUTURE SCOPE

As our method doesn't rely on the encryption of the LSB of pixel values, it becomes resistant to RS assaults. This article introduces the idea of shares, therefore the decryption is much more complex than our imagination. This method works best for both coloured and grayscale photos.

If this kind of approach is integrated with neural networks for the creation of challenging shares of the encrypted image, the method may prove to be significantly more secure and resistant to attacks.

6. REFRENCES

1. Bhat, Guru Prasad M., and Nayana G. Bhat.(2013). Design and Implementation of Visual Cryptography System for Transmission of Secure Data: International Journal on Recent and Innovation Trends in Computing and Communication.(Vol. 5.7: 718-721).

2. Rahman, Amreen(2012). Obscurity of Data Using Steganography with Encryption: International Journal of Research in Engineering, Science and Management.(Vol 4.1: 133-136).

3. Gupta, Ravindra, Akanksha Jain, and Gajendra Singh.(2012). Combine use of steganographyand visual cryptography for secured data hiding in computer forensics: InternationalJournal of Computer Science and Information Technologies.

4. Solak, Serdar.(2020). High Embedding Capacity Data Hiding Technique Based on EMSD and LSBSubstitution Algorithms.(166513-166524).

5. Ali, U. A. M. E., Md Sohrawordi, and Md Palash Uddin.(2019). A Robust and Secured ImageSteganography using LSB and Random Bit Substitution: American Journal of Engineering Research (AJER).

6. Rasras, Rashad J., Ziad A. AlQadi, and Mutaz Rasmi Abu Sara.(2019) "A methodology based onsteganography and cryptography to protect highly secure messages." Engineering, Technology & Applied Science Research.(Vol 9.1: 3681-3684).