VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA-590018

Internship Report on

“Implementation of Lsb Image Steganography”

Submitted in partial fulfillment for the award of the degree of

Bachelor of Engineering

in

Computer Science and Engineering

Submitted By

ANKITA S VAKIL
(1BG18CS013)
M SNAGAJYOTHI
(1BG18CS066)

Internship carried out at

Tequed Labs Private Limited

3, 1st Main Rd, Ittamadu, Banashankari 3rd Stage, Banashankari,
Bengaluru, Karnataka 560085

Vidyayāmruthamashnuthe

B.N.M. Institute of Technology

Approved by AICTE, Affiliated to VTU, Accredited as grade A Institution by NAAC.

All UG branches – CSE, ECE, EEE, ISE & Mech.E accredited by NBA for academic years 2018-19 to 2020-21 & valid upto 30.06.2021

Post box no. 7087, 27th cross, 12th Main, Banashankari 2nd Stage, Bengaluru- 560070,
INDIA Ph: 91-80- 26711780/81/82 Email: principal@bnmit.in, www.bnmit.org

Department of Computer Science and Engineering

2020-21

B.N.M. Institute of Technology

Approved by AICTE, Affiliated to VTU, Accredited as grade A Institution by NAAC.
All UG branches – CSE, ECE, EEE, ISE & Mech.E accredited by NBA for academic years 2018-19 to 2020-21 & valid upto 30.06.2021
Post box no. 7087, 27th cross, 12th Main, Banashankari 2nd Stage, Bengaluru- 560070, INDIA Ph: 91-80-
26711780/81/82 Email: principal@bnmit.in, www. bnmit.org

Department of Computer Science and Engineering

Vidyayāmruthamashnuthe

CERTIFICATE

This is to certify that the 7th Semester Internship work titled "Implementation of LSB Image Steganography" is a bonafide work carried out by **Ankita S Vakil(1BG18CS013) ,M S Nagajyothi(1BG18CS066)**, in partial fulfillment for the award of degree of **Bachelor of Engineering in Department of Computer Science and Engineering** of the Visvesvaraya Technological University, Belagavi, during the year 2020-21. The internship report has been approved as it satisfies the academic requirements with respect to the Internship work prescribed for Bachelor of Engineering Degree.

Signature of the Guide
Mr. Preetam B
Assistant Professor
Department of CSE
BNMIT, Bengaluru

Signature of the HOD
Dr. Sahana D. Gowda
Professor and HOD
Department of CSE
BNMIT, Bengaluru

Signature of the Principal
Dr.Krishnamurthy G N
Principal

Examiners

Internal Examiner External Examiner

Name:

Signature:

ACKNOWLEDGMENT

I sincerely owe my gratitude to all the persons who helped and guided me in completing this project.

I would like to thank **Shri. Narayan Rao R Maanay**, Secretary, BNMIT, Bengaluru for providing the excellent environment and infrastructure in the college.

I would like to sincerely thank **Prof. T J Rama Murthy**, Director, BNMIT, Bengaluru for having extended his constant support and encouragement during the course of this project.

I would like to sincerely thank **Dr. S Y Kulkarni** , Additional Director, BNMIT, Bengaluru for having extended his constant support and encouragement during the course of this project.

I would like to express my gratitude to **Prof. Eishwar N Maanay**, Dean, BNMIT, Bengaluru for his relentless support and encouragement.

I would like to thank **Dr. Krishnamurthy G N**, Principal, BNMIT, Bengaluru for his constant encouragement.

I would like to thank, **Dr. Sahana D. Gowda**, Professor & Head of the Department of Computer Science and Engineering for the encouragement and motivation she provides.

I would like to thank **Mr. Supreeth Y S**, CEO, Tequed Labs for having given me an opportunity to intern at Tequed Labs.

I also whole heartedly thank the **Cyber Security and Ethical Hacking team and the staff** at Tequed Labs, Bengaluru for their support and willingness to teach.

I would like to sincerely thank my guide **Mr. Preetam B,** *Assistant Professor***,** Department of Computer Science and Engineering, for providing relevant information, valuable guidance and encouragement to complete this project.

Ankita S vakil(1BG18CS013)
M S Nagajyothi(1BG18CS066)

ABSTRACT

Image Steganography is the art of hiding the fact that; communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the internet. For hiding secret information in images, there exists a large variety of Steganography techniques. Some are more complex than others and all of them have respective strong and weak points.

Different applications may require absolute invisibility of the secret information, while others require a large secret message to be hidden.

This project is developed for hiding information in any image file by changing the least significant bit of the image. The scope of the project is to implement steganography tools for hiding information in any type of file, (information file or image files) and to save the image in the path chosen by the user and at last, extruding the file.

CONTENTS

LIST OF FIGURES

CHAPTER 1

COMPANY PROFILE

1.1 Introduction

Tequed Labs Private Limited is a private company incorporated on 22 January 2018. It is classified as a non-government company and is registered at Registrar of Companies, Bangalore. Tequed Labs is a research and development center and educational institute based in Bangalore.

They are focused on providing quality education on latest technologies and develop products which are of great need to the society. They also involve in distribution and sales of latest electronic innovation products developed all over the globe to their customers.

They run a project consultancy where they undertake various projects from wide range of companies and assist them technically and build products and provide services to them. They are continuously involved in research about futuristic technologies and finding ways to simplify them for their clients.
They have developed a smart headgear which can give the location of accident when the rier has experienced threshold force.

This work was awarded state’s best innovation in IOT domain. This project was the world finalist in the international innovation challenge called MASTERPIECE in Dubai. It has been exhibited in NASSCOM Product Conclave and has received great appreciation from IT giants. This product has been patented bearing a patent number - 201741034208.

They have developed a women’s safety device which sends the location of the woman in distress to the nearby police station. This work was highly appreciated by the police department and the market ready product is going to be launched soon. This product won the best ICT category project award in a state level exhibit and was exhibited at NASSCOM PRODUCT CONCLVE 2017.

Their other research work includes development of a device for blind which can recognize objects and convert it into speech. This innovation has a lot of potential in helping the blind people.
Their other products include: -

- ➢ Automation of production line and remote quality control monitoring system.
- ➢ Development of mobile app and website for sales of artistic and antiqueproducts.
- ➢ Development of an energy conservation system for paper machineries.
- ➢ Development of an analytic tool for software based vehicle conditionanalysisfor resales.

1.2 Vision

To be a world-class research and development organization committed to enhancing stakeholder’s value.

1.3 Mission

To build best products which are socially innovative with high-quality attributes and provide excellent education to all.

1.4 Values

- Zeal to excel and zest for change.
- Integrity and fairness in all matters.
- Respect for dignity and potential of individuals
- Strict adherence to commitments.
- Ensure speed of response.
- Faster learning, creativity and team-work.
- Loyalty and pride in the company.

1.5 Quality Policy

In the quest to be world-class, TEQUED LABS pursues continual improvement in the quality of its products, services and performance leading to total customer satisfaction and business growth through dedication, commitment and team work of all employees.

1.6 Development Sectors of TEQUED LABS

- Software Development
- Embedded System Design
- Application Development
- IOT based home automation System
- Educational Services
- Product Research

CHAPTER 2

INTRODUCTION

2.1 Steganography

Data hiding is of importance in many applications. For hobbyists, secretive data transmission, for privacy of users etc. the basic methods are:
Steganography and Cryptography.

Steganography is a simple security method. Generally there are three different methods used for hiding information: Steganography, Cryptography and Watermarking.

In cryptography, the information to be hidden is encoded using certain techniques; this information is generally understood to be coded as the data appears nonsensical. Steganography is hiding information; this generally cannot be identified because the coded information doesn‟t appear to be abnormal i.e. its presence is undetectable by sight. Detection of steganography is called Steganalysis.

2.2 Types of Steganography

Steganography is of different types:

1. Text steganography
2. Image steganography
3. Audio steganography
4. Video steganography

In all of these methods, the basic principle of steganography is that a secret message is to be embedded in another cover object which may not be of any significance in such a way that the encrypted data would finally display only the cover data. So it cannot be detected easily to be containing hidden information unless proper decryption is used.

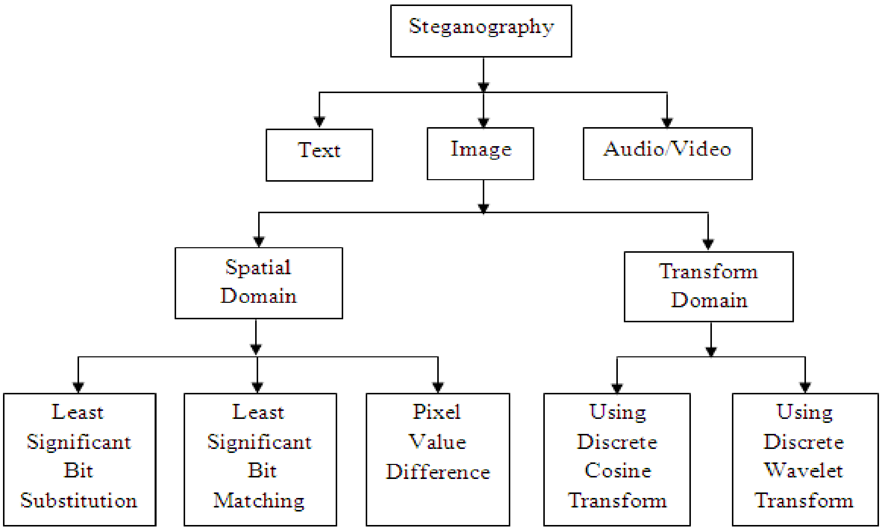

Steganography
Text
Image
Audio/Video
Spatial Domain
Transform Domain
Least Significant Bit Substitution
Least Significant Bit Matching
Pixel Value Difference
Using Discrete Cosine Transform
Using Discrete Wavelet Transform

Figure 2.1: Types of Steganography

s the above explanation goes, every steganography consists of three components:
1. Cover object
2. Message object
3. Resulting Steganographic object

In this project LSB substitution method is implemented and DCT method is discussedfor image steganography. MATLAB or Virtual Studio can be used for coding.

2.3 Basic Model of Steganography

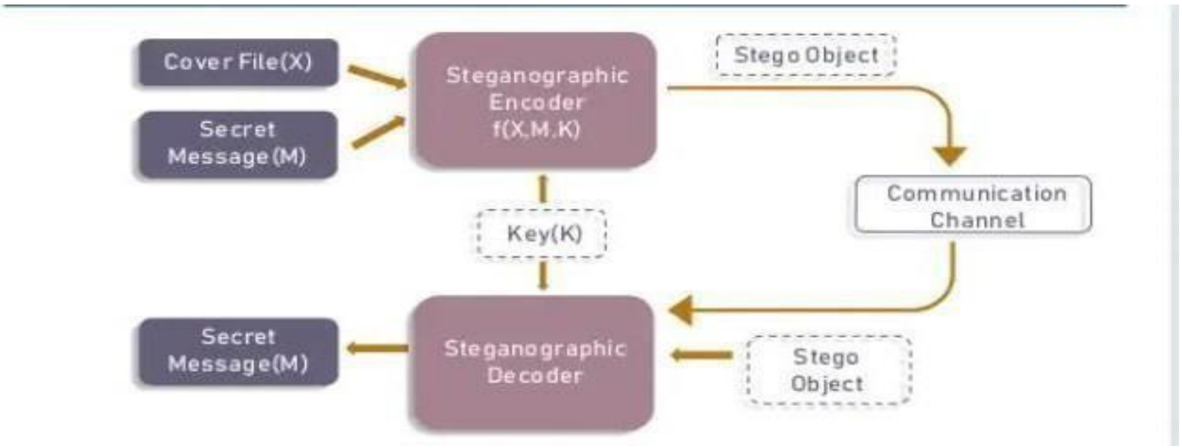


Figure 2.2: Basic Model for Steganography

Message is the data which the sender wishes to be confidential. It can be a plain text, ciphertext, other image, or anything that can be embedded in a bit stream such as a copyright mark, a covert communication, or a serial number.

Password is known as stego-key, which ensures that only recipient who knows the corresponding decoding key will be able to extract the message from a cover-object.

The cover-object with the secretly embedded message is then called the , Stego-object. Recovering message from a stego-object requires the cover-object itself and a corresponding decoding key if a stego-key was used during the encoding process. The original image may or may not be required in most applications to extract the message.

There are several suitable carriers below to be the cover-object*:*

- Network protocols such as TCP, IP and UDP
- Audio that using digital audio formats such as wav, midi, avi, mpeg, mpi and voc
- File and Disk that can hides and append files by using the slack space
- Text such as null characters, just alike morse code including html and java
- Images file such as bmp, gif and jpg, where they can be both color and gray-scale

In general, the information hiding process extracts redundant bits from cover-object. The process consists of two steps:

- Identification of redundant bits in a cover-object. Redundant bits are those bits that can be modified without corrupting the quality or destroying the integrity of the cover-object.
- Embedding process then selects the subset of the redundant bits to be replaced with data from a secret message. The stego-object is created by replacing the selected redundant bits with message bits

2.4 Image Steganography

Image Steganography refers to hiding information i.e. text, images or audio files in another image or video files.

Image Steganography and bitmap pictures:

Using bitmap pictures for hiding secret information is one of most popular choices for Steganography. Many types of software built for this purpose, some of these software use password protection to encrypting information on picture. To use these software you must have a „BMP' format of a pictures to use it, but using other type of pictures like "JPEG", "GIF" or any other types is rather or never used, because of algorithm of "BMP" pictures for Steganography is simple. Also we know that in the web most popular of image types are "JPEG" and other types not "BPM", so we should have a solution for this problem.This software provide the solution of this problem, it can accept any type of image to hide information file, but finally it give the only "BMP" image as an output that has hidden file inside it.

2.4.1 Bitmap Steganography:

Bitmap type is the simplest type of picture because that it doesn‟t have any technology for decreasing file size. Structure of these files is that a bitmap image created from pixels that any pixel created from three colors (red, green and blue said RGB) each color of a pixel is one byte information that shows the density of that color. Merging these three color makes every color that we see in these pictures. We know that every byte in computer science is created from 8 bit that first bit is Most- Significant-Bit (MSB) and last bit Least-Significant-Bit (LSB), the idea of using Steganography science is in this place; we use LSB bit for writing our security information inside BMP pictures. So if we just use last layer (8st layer) of information, we should change the last bit of pixels, in other hands we have 3 bits in each pixel so we have 3*height*width bits memory to write our information. But before writing our data we must write name of data(file), size of name of data & size of data. We can do this by assigning some first bits of memory (8st layer), using each 3 pixel of picture to save a byte of data.

(00101101 00011101 11011100)
(10100110 11000101 00001100)
(11010010 10101100 01100011)

Figure 2.3: Memory bits

There are two different methods for image steganography:

1. Spatial methods
2. Transform methods

In spatial method, the most common method used is LSB substitution method.

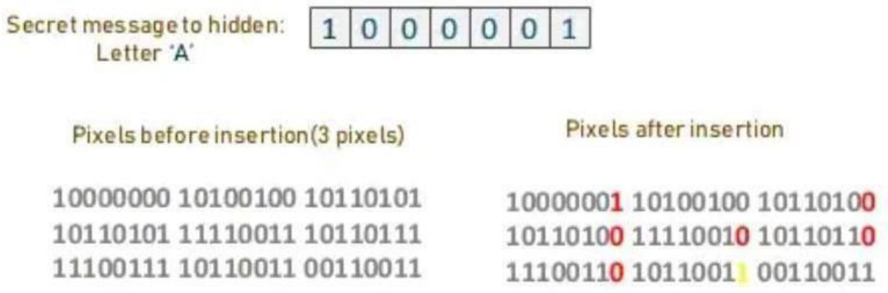
Least significant bit (LSB) method is a common, simple approach to embedding information in a cover file.

In steganography, LSB substitution method is used. I.e. since every image has three components (RGB). This pixel information is stored in encoded format in one byte. The first bits containing this information for every pixel can be modified to store the hidden text. For this, the preliminary condition is that the text to be stored has to be smaller or of equal size to the image used to hide the text.

LSB based method is a spatial domain method. But this is vulnerable to cropping and noise. In this method, the MSB (most significant bits) of the message image to be hidden are stored in the LSB (least significant bits) of the image used as the cover image.

It is known that the pixels in an image are stored in the form of bits. In a grayscale image, the intensity of each pixel is stored in 8 bits (1byte). Similarly for a colour (RGB-red, green, blue) image, each pixel requires 24 bits (8bits for each layer).

The Human visual system (HVS) cannot detect changes in the colour or intensity of a pixel when the LSB bit is modified. This is psycho-visual redundancy since this can e



Secret message to hidden: Letter 'A' | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

Pixels before insertion(3 pixels)

10000000 10100100 10110101
10110101 11110011 10110111
11100111 10110011 00110011

Pixels after insertion

10000001 10100100 10110100
10110100 11110010 10110110
11100110 10110011 00110011

be used as an advantage to store information in these bits and yet notice no major difference in the image.

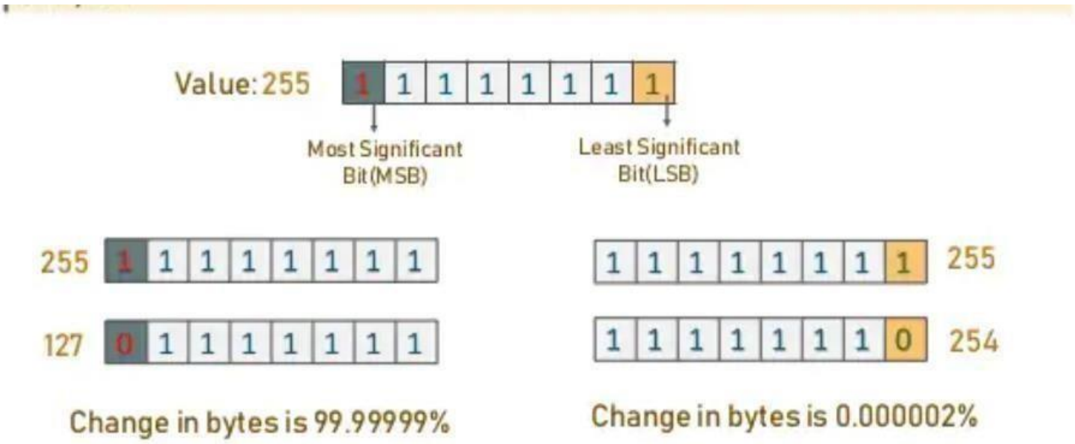

Value:255
1 1 1 1 1 1 1 1
Most Significant Bit(MSB)
Least Significant Bit(LSB)
255
1 1 1 1 1 1 1 1
127
0 1 1 1 1 1 1 1
Change in bytes is 99.99999%
1 1 1 1 1 1 1 1
255
1 1 1 1 1 1 1 0
254
Change in bytes is 0.000002%

Figure 2.4: LSB method

Steps used in LSB steganography:

a. Steps for hiding message image:

1. Read the image to be used as cover image. Noise is added to make it easier to disguise changes due to embedding the message image.
2. Read the image to be sued as message image.
3. Separate the bit planes of each image.

As it is known that the LSB (least significant bit) plane contains the least information associated with any image, and the MSB (most significant bit) plane contains most of the shape, colour information of an image.

It is generally ideal to replace up to 4 least bitplanes of the cover image, with the upper 4 bitplanes without revealing changes in the resultant image. Lesser number of bitplanes from the message image could be used, but the retrieved image would become distorted and loses information.

4. Replace the least 4 bitplanes of cover image with the 4 most significant bitplanes from message image.
5. Get the resultant Steganographic image by recombining these bitplanes.

b. Retrieving message image:

1. Read the Steganographic image.
2. Extract the required number of bitplanes of the image.
3. Recombining the lower four bitplanes would give the retrieved message image.

Discrete Cosine Transform (DCT) method:

When information is embedded in spatial domain, losses can occur such as when the image is cropped etc. To overcome this problem the information is embedded in frequency domain in such a way that we embed the secret information in the significant frequency values and omit the higher frequency part. First the required transformations are applied and then accordingly to hide the secret message, the transform coefficients are changed.

Like in other transforms, decorrelation of the image data is required after applying discrete cosine transform (DCT). And encoding can be then done independently for each coefficient. Hence, compression efficiency is not lost.

In blocking method, blocks of the image are considered and DCT (discrete cosine transform) is done in order to break them. Each block is then subdivided into 64 parts (DCT coefficients). These coefficients are modified i.e. the colour gets modified a little by storing some text or another image in it. Embedding the secret data in the carrier image is generally done for the DCT coefficients that are lower than the chosen threshold value. But embedding information in DCT coefficient value 0 is avoided as this may lead to visual distortion of the cover image.

Palette modification: In palette modification, the unused colours in an image‟s colour palette are replaced with colours to represent hidden message. Palette Modification replaces the unused colours within an image‟s colour palette with colours that represent the hidden message.

For example, we have an image containing 6shades of blue and 5 shades of brown. By modifying the bits, it is possible to generate a completely new palette of colours that were originally absent in the previous image. This changed colour palette may not be detected easily by human eye (HVS) and hence can be used to store other data or information.

- LSB technique can be used for BMP (bitmap) images. Since these involve lossless compression techniques.

  Blocking method – DCT and DWT – used for JPEG images. JPEG images have a lossy compression format, so spatial methods are u steganography. DCT can be used to perform steganography on these images as, they undergo 2 layers of compression. One is lossless and then Huffman coding is used. The encryption data can be placed between these two layers.
- Palette based method – used for GIF images. GIF images have a very limited color palette. Therefore palette modification method is more suitable.

In this project an encrypted image which has a hidden data or image inside it is decrypted through DCT and LSB substitution method. The encrypted data/image is decrypted using Visual Studio codes.

CHAPTER 3

ABOUT PROJECT

3.1 Objectives

The goal of steganography is covert communication. So, a fundamental requirement of this steganography system is that the hider message carried by stego-media should not be sensible to human beings.

The other goal of steganography is to avoid drawing suspicion to the existence of a hidden message. This approach of information hiding technique has recently became important in a number of application area.

This project has following objectives:

- To produce a security tool based on steganography techniques.
- To explore techniques of hiding data using encryption module of this project
- To extract techniques of getting secret data using decryption module.

Steganography sometimes is used when encryption is not permitted. Or, more commonly, steganography is used to supplement encryption. An encrypted file may still hide information using steganography, so even if the encrypted file is deciphered, the hidden message is not seen

3.2 Problem Statement

This project aims to use LSB Image steganography is to hide the file in an Image. This hidden information can be retrieved only through proper decoding technique

The former consists of linguistic or language forms of hidden writing. The later, such as invisible ink, try of hide messages physically. One disadvantage of linguistic steganography is that users must equip themselves to have a good knowledge of linguistry. In recent years, everything is trending toward digitization. And with the development of the internet technology, digital media can be transmitted conveniently over the network.

Therefore, messages can be secretly carried by digital media by using the steganography techniques, and then be transmitted through the internet rapidly Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the internet.

For hiding secret information in images, there exists a large variety of steganography techniques some are more complex than others and all of them have respective strong and weak points.

So we prepare this application, to make the information hiding more simple and user friendly.

3.3 System Requirements Specification

3.4.1 Software Requirements

- Linux Operating System
- Stego tools
- Ubuntu terminal

3.4.2 Hardware Requirements

- Processor: Intel Core Processor
- Main Memory: 128 MB RAM, 256 MB recommended.
- Hard Disk: 110 MB hard disk space required.
- Display: 800 x 600 or higher-resolution display with 256 colors.

3.4 Methodology

In this project an encrypted image which has a hidden data or image inside it is decrypted through DCT and LSB substitution method. The encrypted data/image is decrypted using Visual Studio codes and it is dealt with using data hiding.

User needs to run the application. The user has two tab options – encrypt and decrypt.

If user select encrypt, application give the screen to select image file, information file and option to save the image file.

If user select decrypt, application gives the screen to select only image file and ask path where user want to save the secrete file.

This project has two methods – Encrypt and Decrypt.

In encryption the secret information is hiding in with any type of image file. The message that has to be extracted is embedded in the image. Decryption is getting the secret information from image file by extracting the image using the LSB Steganography tool and the passkey to ensure tight security.The secret information is converted to binary bits and stored in the LSB of the image that is passed and later extracted. The image appears to be the same but there is a slight change in the LSB of the image.

.

CHAPTER 4

SYSTEM DESIGN AND ANALYSIS

4.1 System Design

Steganography system requires any type of image file and the information or message that is to be hidden. It has two modules encrypt and decrypt.

Microsoft .Net framework / Visual Studio prepares a huge amount of tool and options for programmers that they simplify programming. One of .Net tools for pictures and images is auto-converting most types of pictures to BMP format. We can use this tool in this software called “Steganography” that is written in C# , .Net language and you can use this software to hide your information in any type of pictures without any converting its format to BMP (software converts inside it).

The algorithm used for Encryption and Decryption in this application provides using several layers lieu of using only LSB layer of image. Writing data starts from last layer (8st or LSB layer); because significant of this layer is least and every upper layer has doubled significant from its down layer. So every step we go to upper layer image quality decreases and image retouching transpires.

The encrypt module is used to hide information into the image; no one can see that information or file. This module requires any type of image and message and gives the only one image file in destination.

The decrypt module is used to get the hidden information in an image file. It take the image file as an output, and give two file at destination folder, one is the same image file and another is the message file that is hidden it that.

Before encrypting file inside image we must save name and size of file in a definite place of image. We could save file name before file information in LSB layer and save file size and file name size in most right-down pixels of image. Writing this information is needed to retrieve file from encrypted image in decryption state.
The graphical representation of this system is as follows:

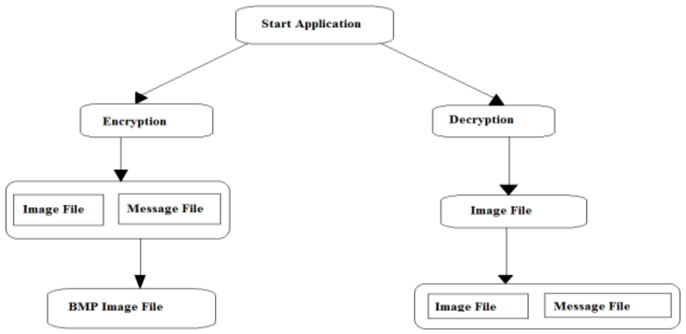
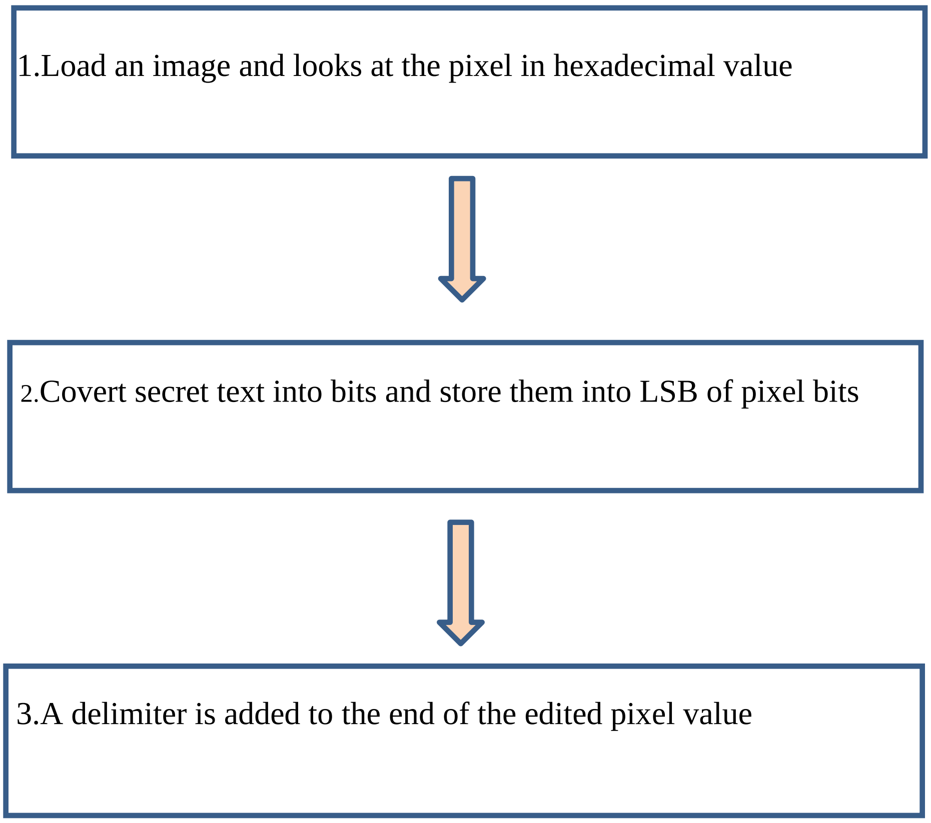


Figure 4.1: Graphical Representation of the Application

Figure 4.2: Encryption Process

To encrypt a secret message inside an image:

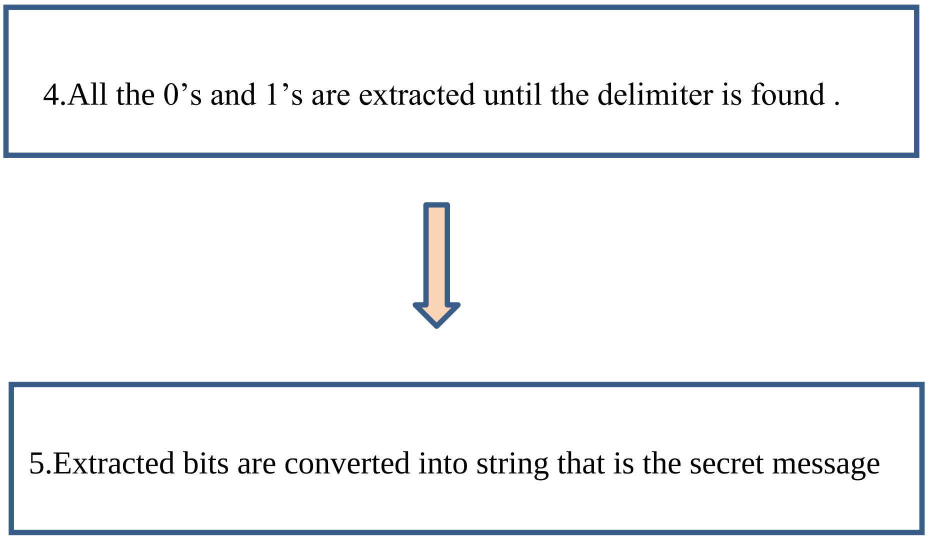


1.Load an image and looks at the pixel in hexadecimal value

2.Covert secret text into bits and store them into LSB of pixel bits

3.A delimiter is added to the end of the edited pixel value

Decryption happens at the receiver end, to extract the data the stego object is fed into the Steganographic decoder and the cover object and the secret message is received at the output. The process is made more secure by adding encryption that makes use of an encryption algorithm and decryption algorithm.



4.All the 0’s and 1’s are extracted until the delimiter is found .

5.Extracted bits are converted into string that is the secret message

Figure 4.3: Decryption Process

4.2 Implementation Code

```
import optparse
import PIL from image

def rgb2hex(r, g, b):
    return '#{:02x}{:02x}{:02x}'.format(r, g, b)

def hex2rgb(hexcode):
    return tuple(map(ord, hexcode[1:].decode('hex')))

def str2bin(message):
    binary = bin(int(binascii.hexlify(message), 16))
    return binary[2:]

def bin2str(binary):
    message = binascii.unhexlify('%x' % (int('0b' + binary, 2)))
    return message

def encode(hexcode, digit):
    if hexcode[-1] in ('0', '1', '2', '3', '4', '5'):
        hexcode = hexcode[:-1] + digit
        return hexcode
    else:
        return None

def decode(hexcode):
    if hexcode[-1] in ('0', '1'):
        return hexcode[-1]
    else:
        return None

def hide(filename, message):
    img = Image.open(filename)
    binary = str2bin(message) + '1111111111111110'
    if img.mode in ('RGBA'):
        img = img.convert('RGBA')
        datas = img.getdata()
        newData = []
        digit = 0
        temp = ''
```

```
for item in datas:
            if (digit < len(binary)):
                newpix = encode(rgb2hex(item[0],item[1],item[2]),binary[digit])
                if newpix == None:
                    newData.append(item)
                else:
                    r, g, b = hex2rgb(newpix)
                    newData.append((r,g,b,255))
                    digit += 1
            else:
                newData.append(item)
        img.putdata(newData)
        img.save(filename, "PNG")
        return "Completed!"
    return "Incorrect Image Mode, Couldn't Hide"

def retr(filename):
    img = Image.open(filename)
    binary = ''

    if img.mode in ('RGBA'):
        img = img.convert('RGBA')
        datas = img.getdata()

        for item in datas:
            digit = decode(rgb2hex(item[0], item[1], item[2]))
            if digit == None:
                pass
            else:
                binary = binary + digit
                if (binary[-16:] == '1111111111111110'):
                    print "Success"
                    return bin2str(binary[:-16])

        return bin2str(binary)
    return "Incorrect Image Mode, Couldn't Retrieve"

def Main():
    parser = optparse.OptionParser('usage %prog ' + '-e/-d <target file>')
    parser.add_option('-e', dest='hide', type='string', help='target picture path to hide
text')
```

```
  parser.add_option('-d', dest='retr', type='string', help='target picture path to retrieve
text')
    (options, args) = parser.parse_args()
    if (options.hide != None):
        text = raw_input("Enter a message to hide: ")
        print hide(options.hide, text)
    elif (options.retr != None):
        print retr(options.retr)


else:
        print parser.usage
        exit(0)

if__name__== '_main_':
    Main()
```

CHAPTER 5

DISCUSSIONS AND SNAPSHOTS

5.1 Brief discussion on how the project works

In this project, a secret message is hidden in an image and later extracted using the LSB Image Steganography Technique. The 2 images are cyber3.jpeg and dog.jpeg and the 2 text files are pattern.txt and test.txt respectively. The steganography folder has the input file and the output is saved in the folder called scan.

5.2 Snapshots

This is the initial picture with .jpeg extension in which a cover file is hidden The cyber3.jpeg image is stored in the folder Steganography.

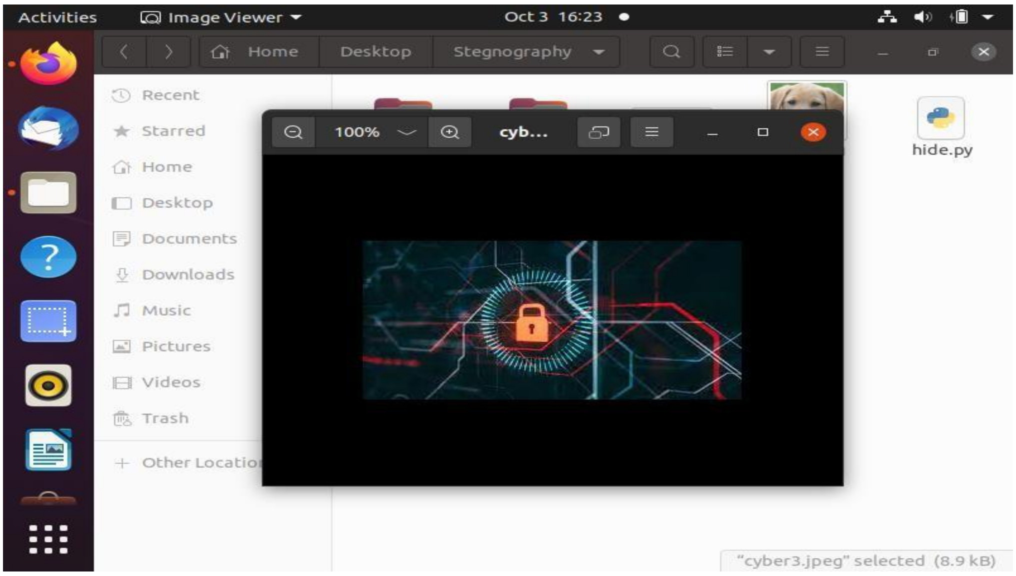
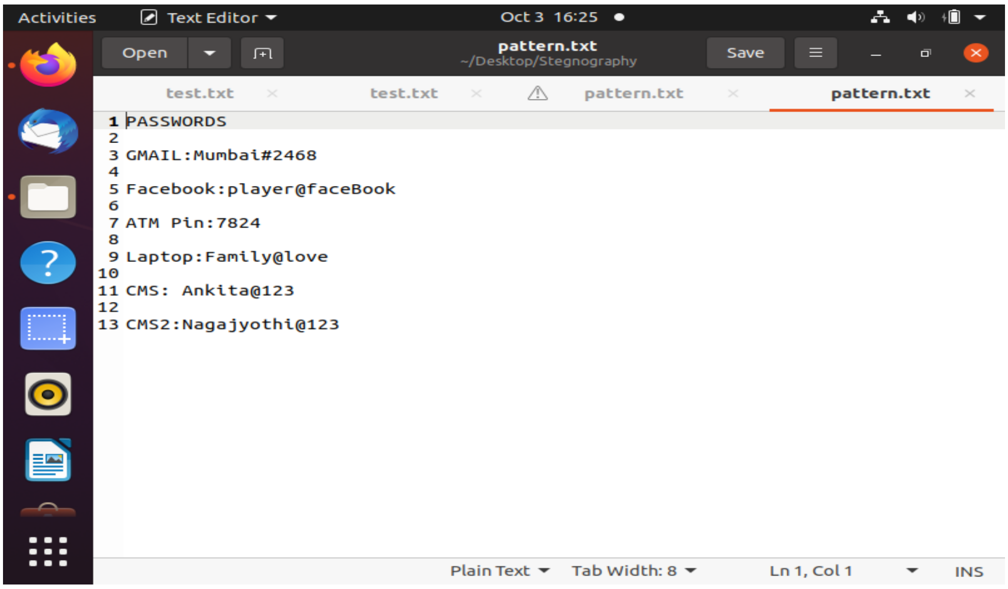


Figure 5.0: First Picture before LSB Steganography

The file that is called the cover file is sent along with the image. The file is of .txt extension and is called patter.txt. The file has all the essential passwords of a particular user which has to be hidden in the image using LSB Steganography technique and then extracted so as to avoid any kind of attack.



```
PASSWORDS

GMAIL:Mumbai#2468

Facebook:player@faceBook

ATM Pin:7824

Laptop:Family@love

CMS: Ankita@123

CMS2:Nagajyothi@123
```

Figure 5.1: First Cover File before LSB Steganography

Another image called dog.jpeg is used which is saved as dog.jpeg in the Steganography folder in which the user saves a secret message that has to be extracted using LSB Steganography.

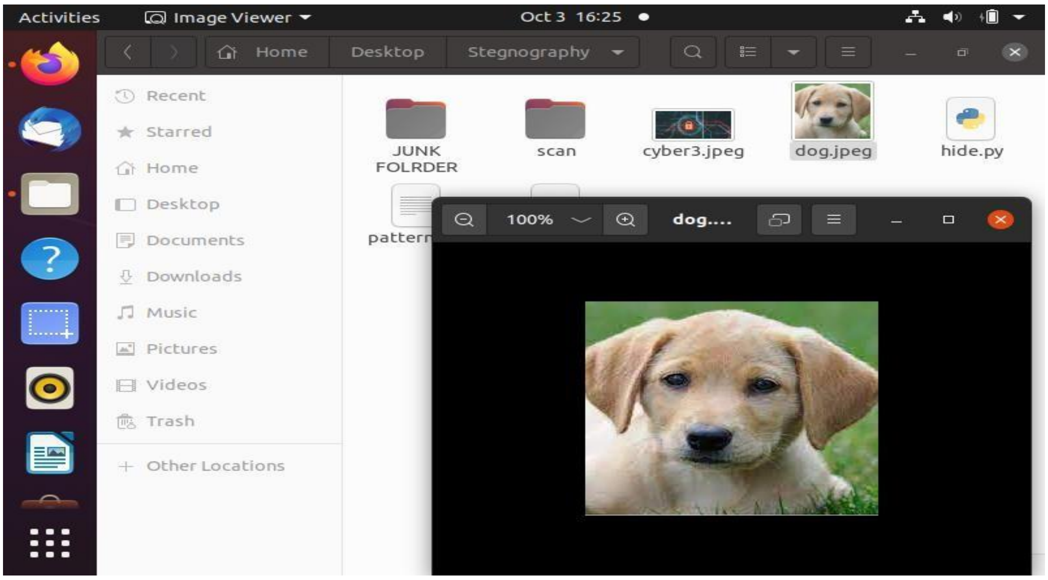
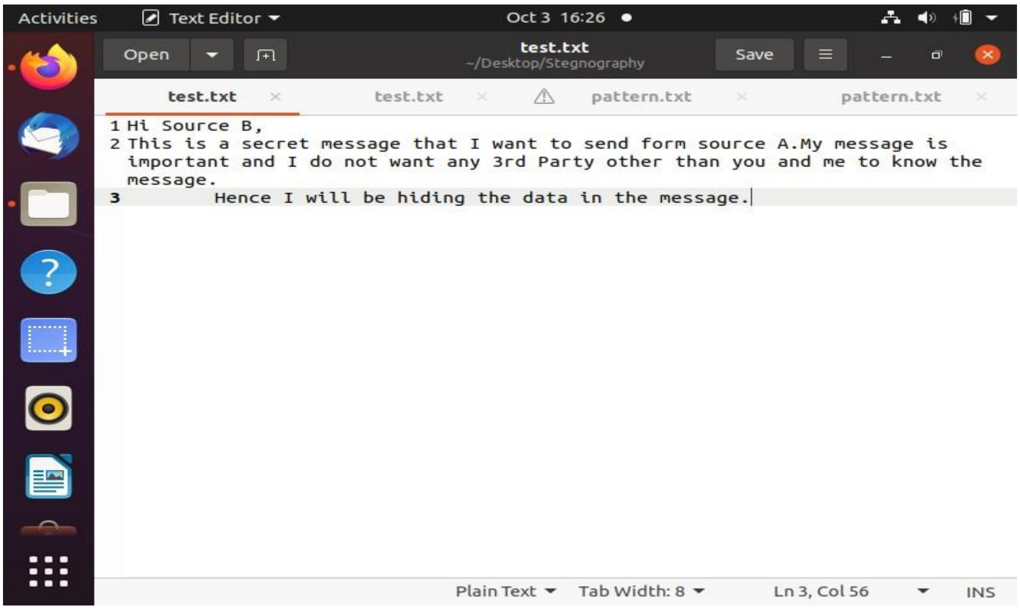


Figure 5.2: Second Picture before LSB Steganography

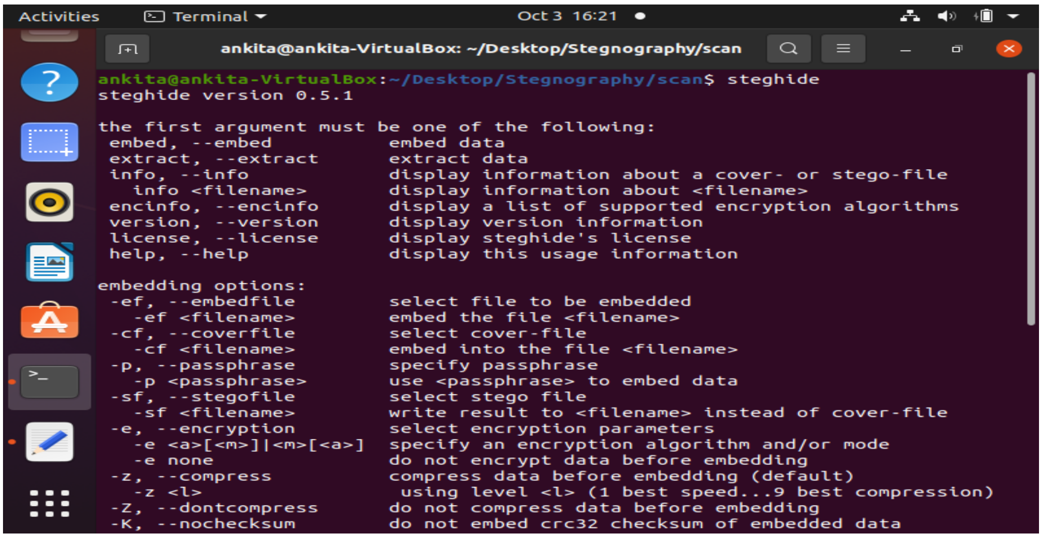
Another cover file that has to be hidden in the image is saved in the Steganography folder with the name test.txt which has to be extracted using the LSB Image Steganography.



```
Hi Source B,
This is a secret message that I want to send form source A.My message is important and I do not want any 3rd Party other than you and me to know the message.
        Hence I will be hiding the data in the message.
```

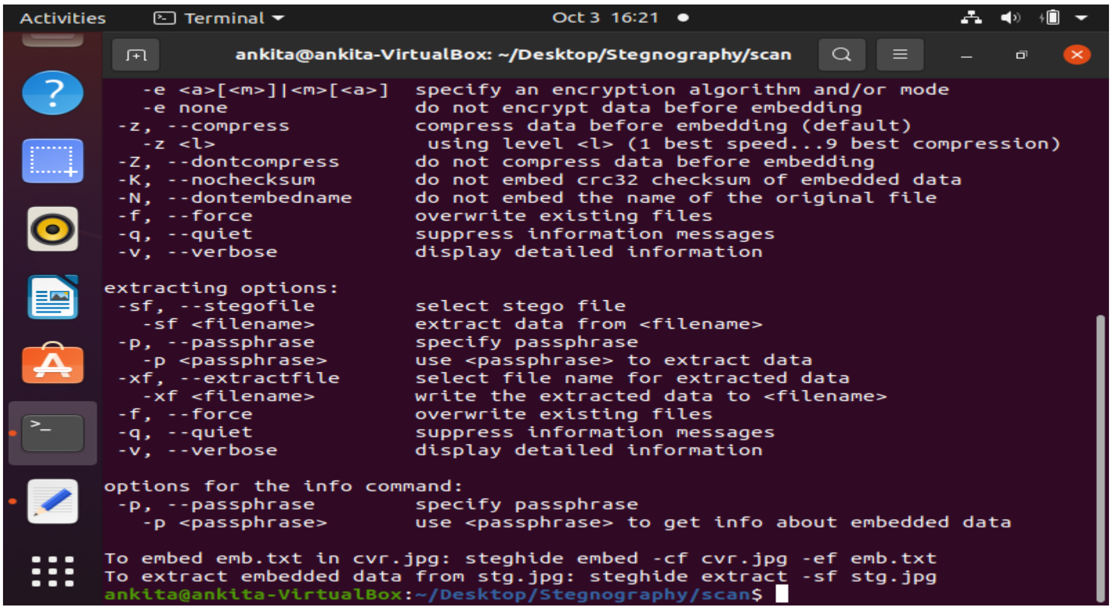
Figure 5.3: Second Cover File before LSB Steganography

In the output terminal, we make use of the steganography tools.Some of the steganography tools are given below which is used to extract the secret message from the cyber3.jpeg and Dog.jpeg image using the LSB Image Steganography technique.



```
ankita@ankita-VirtualBox:~/Desktop/Stegnography/scan$ steghide
steghide version 0.5.1

the first argument must be one of the following:
 embed, --embed          embed data
 extract, --extract      extract data
 info, --info            display information about a cover- or stego-file
   info <filename>       display information about <filename>
 encinfo, --encinfo      display a list of supported encryption algorithms
 version, --version      display version information
 license, --license      display steghide's license
 help, --help            display this usage information

embedding options:
 -ef, --embedfile        select file to be embedded
   -ef <filename>        embed the file <filename>
 -cf, --coverfile        select cover-file
   -cf <filename>        embed into the file <filename>
 -p, --passphrase        specify passphrase
   -p <passphrase>       use <passphrase> to embed data
 -sf, --stegofile        select stego file
   -sf <filename>        write result to <filename> instead of cover-file
 -e, --encryption        select encryption parameters
   -e <a>[<m>]|<m>[<a>]  specify an encryption algorithm and/or mode
   -e none               do not encrypt data before embedding
 -z, --compress          compress data before embedding (default)
   -z <l>                 using level <l> (1 best speed...9 best compression)
 -Z, --dontcompress      do not compress data before embedding
 -K, --nochecksum        do not embed crc32 checksum of embedded data
```

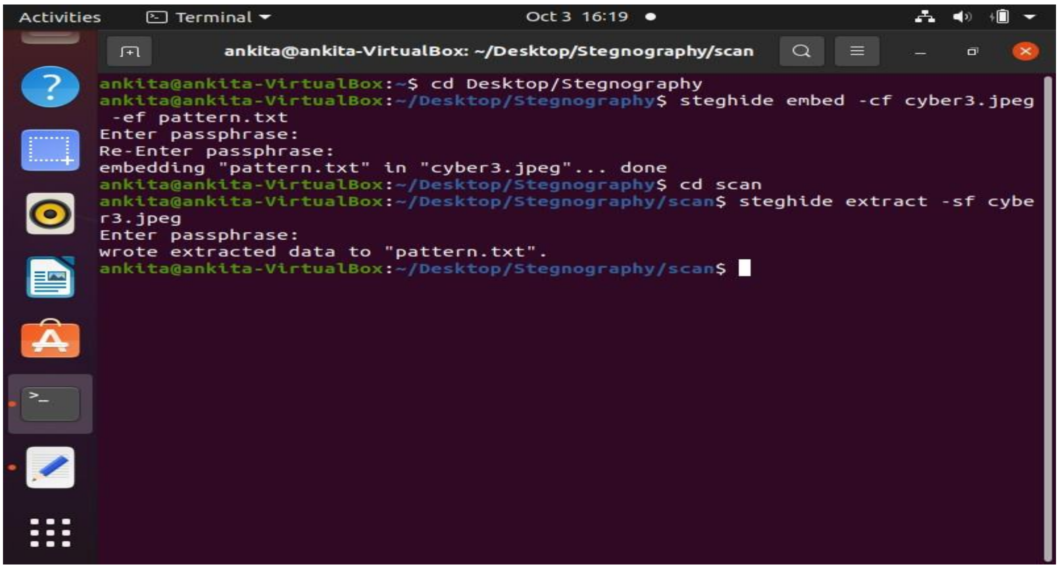
Figure 5.4: Steganography Tools



```
   -e <a>[<m>]|<m>[<a>]  specify an encryption algorithm and/or mode
   -e none               do not encrypt data before embedding
 -z, --compress          compress data before embedding (default)
   -z <l>                 using level <l> (1 best speed...9 best compression)
 -Z, --dontcompress      do not compress data before embedding
 -K, --nochecksum        do not embed crc32 checksum of embedded data
 -N, --dontembedname     do not embed the name of the original file
 -f, --force             overwrite existing files
 -q, --quiet             suppress information messages
 -v, --verbose           display detailed information

extracting options:
 -sf, --stegofile        select stego file
   -sf <filename>        extract data from <filename>
 -p, --passphrase        specify passphrase
   -p <passphrase>       use <passphrase> to extract data
 -xf, --extractfile      select file name for extracted data
   -xf <filename>        write the extracted data to <filename>
 -f, --force             overwrite existing files
 -q, --quiet             suppress information messages
 -v, --verbose           display detailed information

options for the info command:
 -p, --passphrase        specify passphrase
   -p <passphrase>       use <passphrase> to get info about embedded data

To embed emb.txt in cvr.jpg: steghide embed -cf cvr.jpg -ef emb.txt
To extract embedded data from stg.jpg: steghide extract -sf stg.jpg
ankita@ankita-VirtualBox:~/Desktop/Stegnography/scan$
```

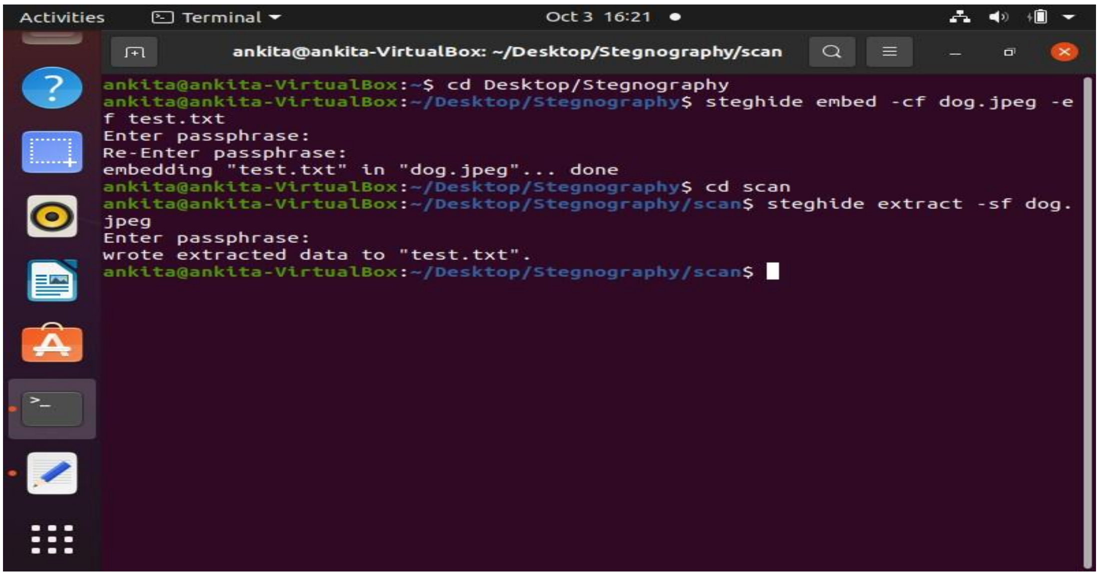
Figure 5.5: Steganography Tools

The steghide command is used to embed the patter.txt in the cyber3.jpeg and using a passkey the secret message is extracted.



```
ankita@ankita-VirtualBox:~$ cd Desktop/Stegnography
ankita@ankita-VirtualBox:~/Desktop/Stegnography$ steghide embed -cf cyber3.jpeg
 -ef pattern.txt
Enter passphrase:
Re-Enter passphrase:
embedding "pattern.txt" in "cyber3.jpeg"... done
ankita@ankita-VirtualBox:~/Desktop/Stegnography$ cd scan
ankita@ankita-VirtualBox:~/Desktop/Stegnography/scan$ steghide extract -sf cybe
r3.jpeg
Enter passphrase:
wrote extracted data to "pattern.txt".
ankita@ankita-VirtualBox:~/Desktop/Stegnography/scan$
```

Figure 5.6: Terminal command for First secret message



```
ankita@ankita-VirtualBox:~$ cd Desktop/Stegnography
ankita@ankita-VirtualBox:~/Desktop/Stegnography$ steghide embed -cf dog.jpeg -e
f test.txt
Enter passphrase:
Re-Enter passphrase:
embedding "test.txt" in "dog.jpeg"... done
ankita@ankita-VirtualBox:~/Desktop/Stegnography$ cd scan
ankita@ankita-VirtualBox:~/Desktop/Stegnography/scan$ steghide extract -sf dog.
jpeg
Enter passphrase:
wrote extracted data to "test.txt".
ankita@ankita-VirtualBox:~/Desktop/Stegnography/scan$ 
```

Figure 5.7: Terminal command for Second secret message

After using the appropriate stighide command, the image is passed to another folder called Scan where we find the secret message that was extracted from cyber3.jpeg image using LSB Image Steganography. The image looks the same as the original image but the LSB bits in the image have been varied.

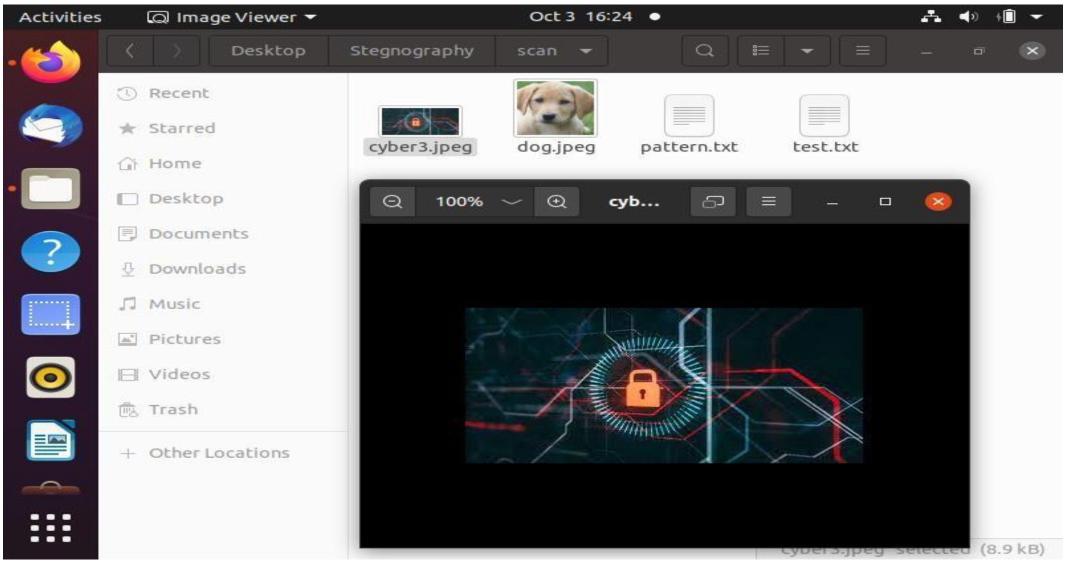
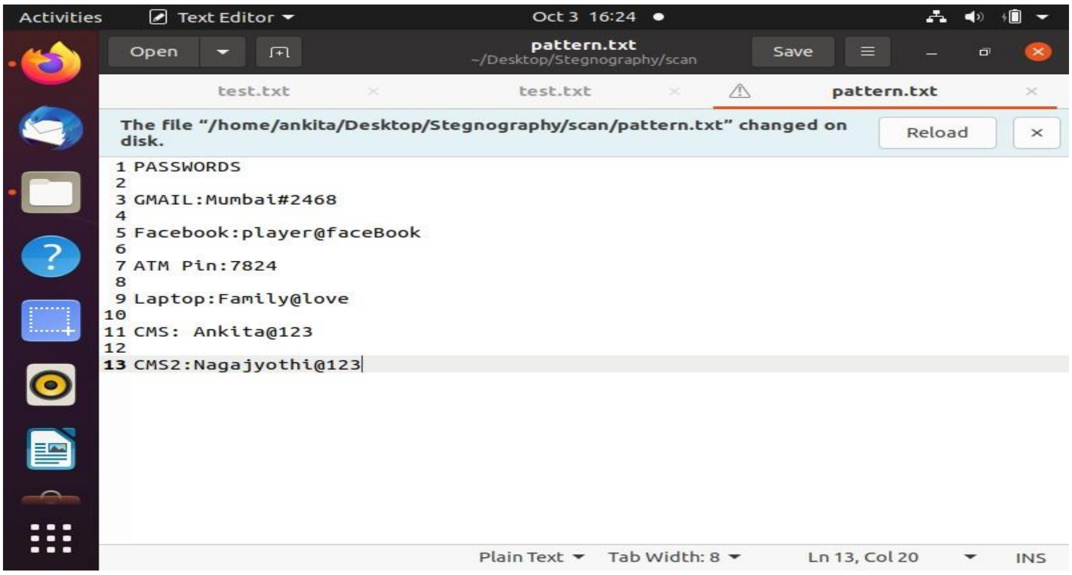


Figure 5.8: Replicated image of cyber3.jpeg after LSB Image Steganography



The file “/home/ankita/Desktop/Stegnography/scan/pattern.txt” changed on disk.

```
PASSWORDS

GMAIL:Mumbai#2468

Facebook:player@faceBook

ATM Pin:7824

Laptop:Family@love

CMS: Ankita@123

CMS2:Nagajyothi@123
```

Figure 5.9:Exctraction of secret message pattern.txt after LSB Image Steganography

After using the appropriate stighide command, the image is passed to another folder called Scan where we find the secret message that was extracted from dog.jpeg image using LSB Image Steganography. The image looks the same as the original image but the LSB bits in the image have been varied.

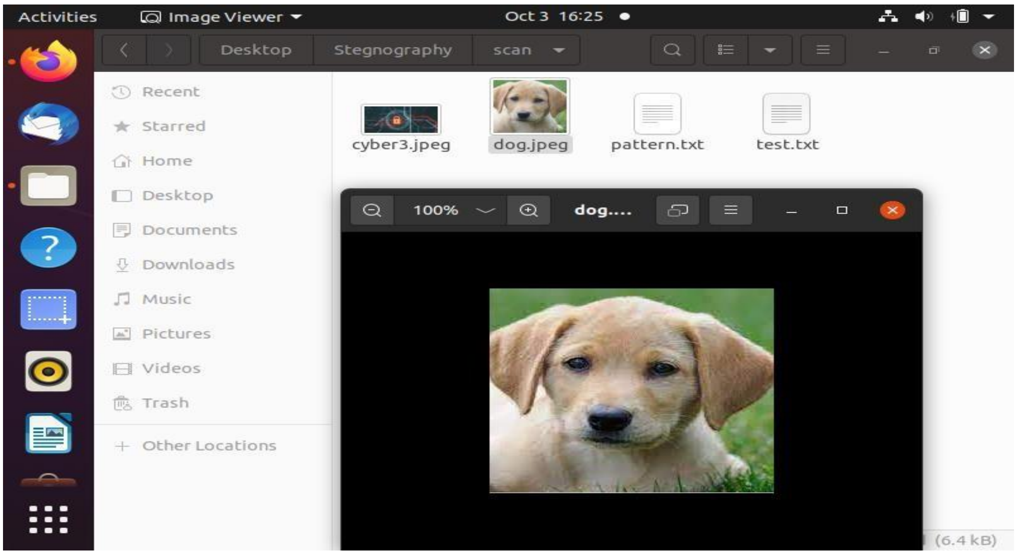
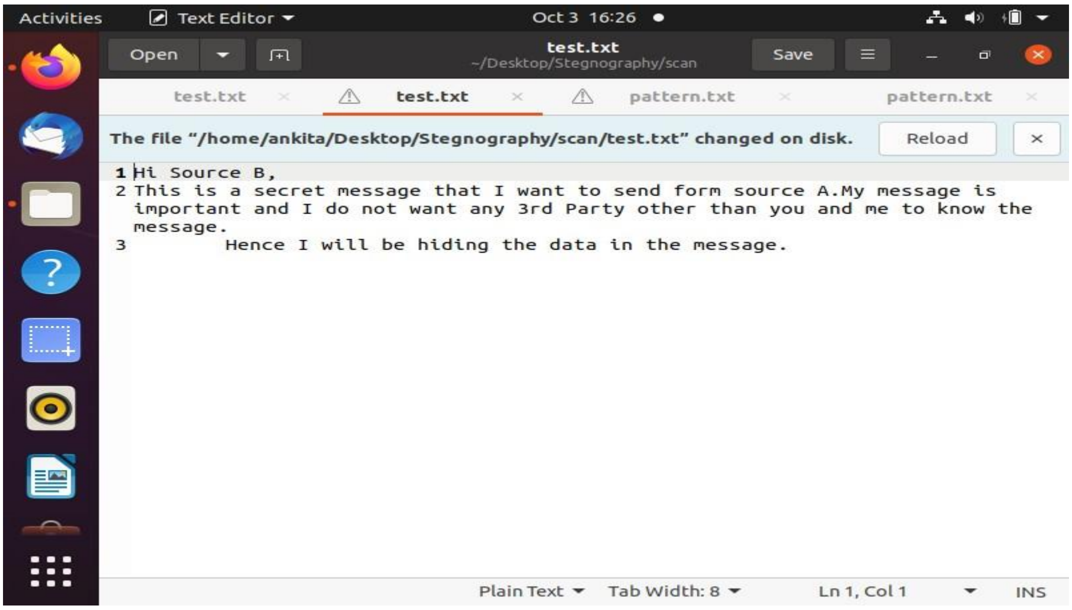


Figure 5.10: Replicated image of dog.jpeg after LSB Image Steganography



The file "/home/ankita/Desktop/Stegnography/scan/test.txt" changed on disk.

```
Hi Source B,
This is a secret message that I want to send form source A.My message is
important and I do not want any 3rd Party other than you and me to know the
message.
        Hence I will be hiding the data in the message.
```

Figure 5.11:Exctraction of secret message test.txt after lsb Image Steganography

General view of the original folder Steganography which consists of to images cyber3.jpeg and dog,jpeg and two secret messages that have to be send using LSB Image Steganography.

The path is given by Home/Desktop/Steganography.

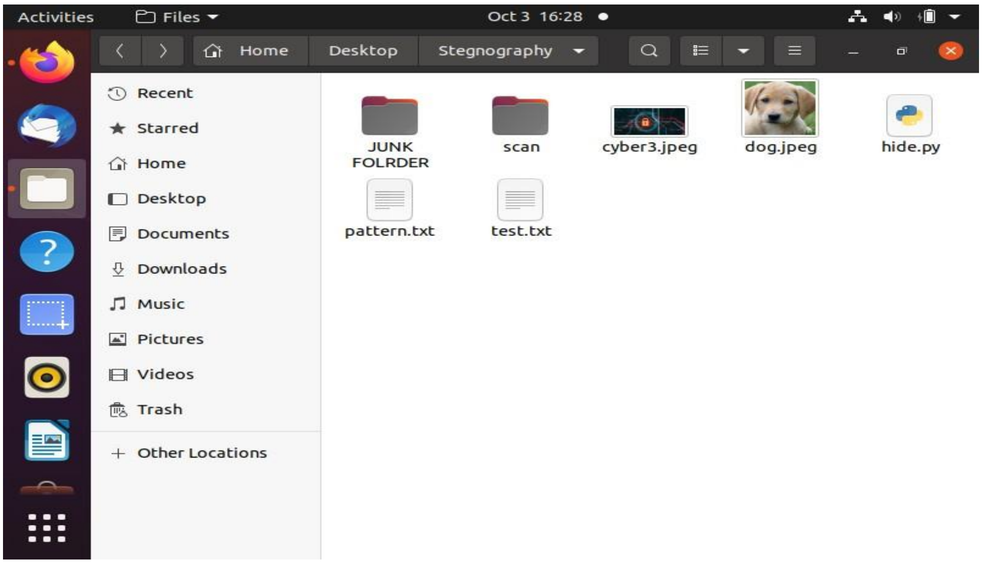


Figure 5.12:Steganography folder before LSB Image Steganography

After using the LSB Image Steganography, the secret message pattern.txt and test.txt is extracted from the image and saved in the folder called scan.
The path is given by Home/Desktop/Steganography/scan

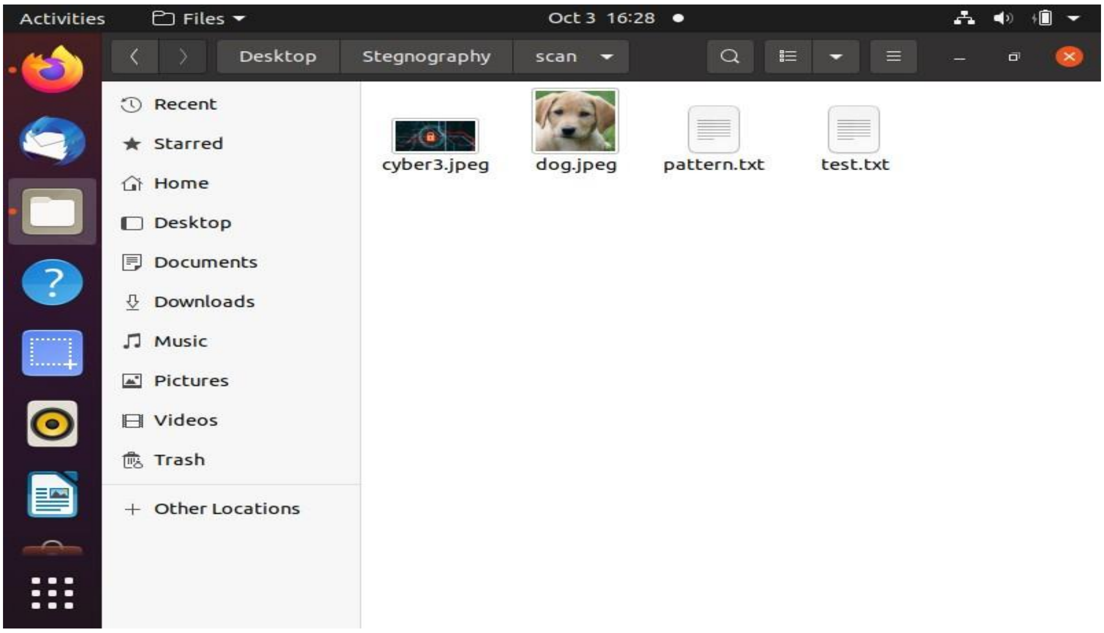


Figure 5.13:scan folder after LSB Image Steganography

CONCLUSION

Steganography is a really interesting subject and outside of the mainstream cryptography and system administration that most of us deal with day after day. Steganography can be used for hidden communication. We have explored the limits of steganography theory and practice. We printed out the enhancement of the image steganography system using LSB approach to provide a means of secure communication. A stego-key has been applied to the system during embedment of the message into the cover image.

This steganography application software provided for the purpose to how to use any type of image formats to hiding any type of files inside their. The master work of this application is in supporting any type of pictures without need to convert to bitmap, and lower limitation on file size to hide, because of using maximum memory space in pictures to hide the file.

Since ancient times, man has found a desire in the ability to communicate covertly. The recent explosion of research in watermarking to protect intellectual property is evidence that steganography is not just limited to military or espionage applications. Steganography, like cryptography, will play an increasing role in the future of secure communication in the “digital world”.

REFERENCES

.

1. http://www.google.com
2. http://www.microsoft.com
3. Niels Provos, and Peter Honeyman. “Hide and Seek: An Introduction to Steganography.” IEEE Security & Privacy Magazine, May-June 2013.
4. MCAD/MCSD Self-Paced Training Kit: Developing Web Applications with Microsoft® Visual Basic® .NET and Microsoft Visual C#®.NET, Second Edition
5. Shikha, and Vidhu Kiran Dutt. "International Journal of Advanced Research in Computer Science and Software Engineering." www.ijarcsse.com/ Sept.2014