VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

A MINI PROJECT REPORT

on

"Image Steganography using LSB Algorithm"

Submitted by

A M Akhilesh	4SF19IS001
Jackson Lobo	4SF19IS031

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

Under the Guidance of

Ms. Jayapadmini Kanchan,

Assistant Professor,

Department of ISE,

at

SAHYADRI

College of Engineering and Management

Adyar, Mangaluru - 575 007

2021 - 22

SAHYADRI

College of Engineering and Management

Adyar, Mangaluru - 575 007

Department of Information Science & Engineering

CERTIFICATE

This is to certify that the mini project entitled **"Image Steganography using LSB Algorithm"** has been carried out by **A M Akhilesh (4SF19IS001) and Jackson Lobo (4SF19IS031)** the bonafide students of Sahyadri College of Engineering and Management, Bachelor of Engineering in Information Science & Engineering of Visvesvaraya Technological University, Belagavi during the year 2021-22. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed in File Structures Laboratory with Mini Project(18ISL67) for the said degree in sixth semester.

Signature of the Guide1	**Signature of the Guide2**	**Signature of the HOD**
Ms. Jayapadmini Kanchan	Mrs. Harinakshi C	Dr. Shamanth Rai

External Viva:

Examiner's Name	Signature with Date
1.	.....
2.	.....

SAHYADRI

College of Engineering and Management

Adyar, Mangaluru - 575 007

Department of Information Science & Engineering

DECLARATION

We hereby declare that the entire work embodied in this Mini Project Report titled **"Image Steganography using LSB Algorithm"** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Ms. Jayapadmini Kanchan,** for **Bachelor of Engineering** in **Information Science & Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

A M Akhilesh (4SF19IS001)

Jackson Lobo (4SF19IS031)

Dept. of ISE, SCEM, Mangaluru

Abstract

Least Significant Bit replacement, a spatial domain algorithm, is the most popular and widely used technique in image steganography due to its simplicity and effectiveness. Different methods of data hiding in spatial domain have been proposed and continue to be improved upon. Among them, the LSB replacement method is quite simple and the most popular. However, because the LSB replacement method is quite simple,compared to the other methods, some of its security issues must be improved upon. This project proposes a highly secured data hiding technique in the spatial domain of image steganography. The proposed scheme takes the message bit and performs XOR operation with the 7th bit of every RGB component and, after then, the produced output is embedded within the 8th bit of each component of RGB. The embedding procedure is done in a way that there will be no sign of original message inside the cover object and, obviously, without using any outside key. A detailed study of the proposed LSB replacement algorithm including PSNR- and MSE-based investigations has been made. Experimental results shows a very good peak signal-to-noise ratio (PSNR) (55.90 dB for 65,536 bits of message within a 256x256 pixel cover image) and mean square error (MSE) value which indicates to less imperceptibility and more security. The comparative results prove that the proposed technique provides more security to secret information sharing, compared to other related techniques.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on **"Image Steganography using LSB Algorithm"**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi for the award of Bachelor of Engineering in Information Science & Engineering.

We are profoundly indebted to our guide, **Ms. Jayapadmini Kanchan**, Assistant Professor, Department of Information Science & Engineering for innumerable acts of timely advice, encouragement and We sincerely express our gratitude.

We express our sincere gratitude to **Dr. Shamanth Rai**, Head and Associate Professor, Department of Information Science & Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Rajesha S**, Principal, Sahyadri College of Engineering and Management and **Dr. D. L. Prabhakara**, Director, Sahyadri Educational Institutions, who have always been a great source of inspiration.

Finally, yet importantly, we express our heartfelt thanks to our family and friends for their wishes and encouragement throughout the work.

A M Akhilesh (4SF19IS001)

Jackson Lobo (4SF19IS031)

Table of Contents

List of Figures

Chapter 1

Introduction

Information is an asset. Sensitive information or message is not only an asset but such can also become a threat if it loses its confidentiality. The growth of digital communication enables us to transmit messages or information over the internet within a very short time. While digital communication made our life much easier concerns exist over the security of data transmission. For a secure and reliable data transfer process, cryptography has remained the most used solution over the last few decades.

Since a very longtime, steganography has been used as a solution to maintain data confidentiality. The word steganography refers to hiding the presence of a message or information within a cover. Image steganography is the most popular technique because it is lightweight. The most common and popular method of modern day steganography is to make use of LSB of picture's pixel information. This technique works best when the file is longer than the message file and if image is gray-scale.

1.1 Purpose

The main purpose of this project is to hide messages inside an image by replacing Least Significant Bit of image with the bits of message to be hidden. By modifying only the first most right bit of an image we can insert our secret message and it also make the picture unnoticeable, but if our message is too large it will start modifying the second right most bit.

1.2 Scope

Steganography can be a solution which makes it possible to send news and information without being censored and without the fear of the messages being intercepted and traced back to us. With LSB methods of inserting data, simply compressing the image using lossy compression is enough to disable or remove the hidden message

1.3 Overview

Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. Information Security is important. As we're moving toward more digitalized world, the impact that data breaches have on our life grows. This application uses LSB algorithm for image steganography, to encrypt the messages into the images and simultaneously decrpyt the message from the same image.

Chapter 2

Requirements Specification

2.1 Hardware Specification

- Processor : Intel(R) Core(TM) i3-8145U CPU @ 2.30GHz
- RAM : 8GB
- Hard Disk : 1TB
- Input Device : Standard Keyboard and Mouse
- Output Device : Monitor

2.2 Software Specification

- Programming Language : C++
- IDE :Visual Studio Code

Chapter 3

System Design

3.1 Architecture Diagram

The user is displayed with two options. If the user chooses encryption, the last bit called as Least Significant Bit in each pixel of the image is replaced with the data bit. The resultant image obtained is called as stego image.

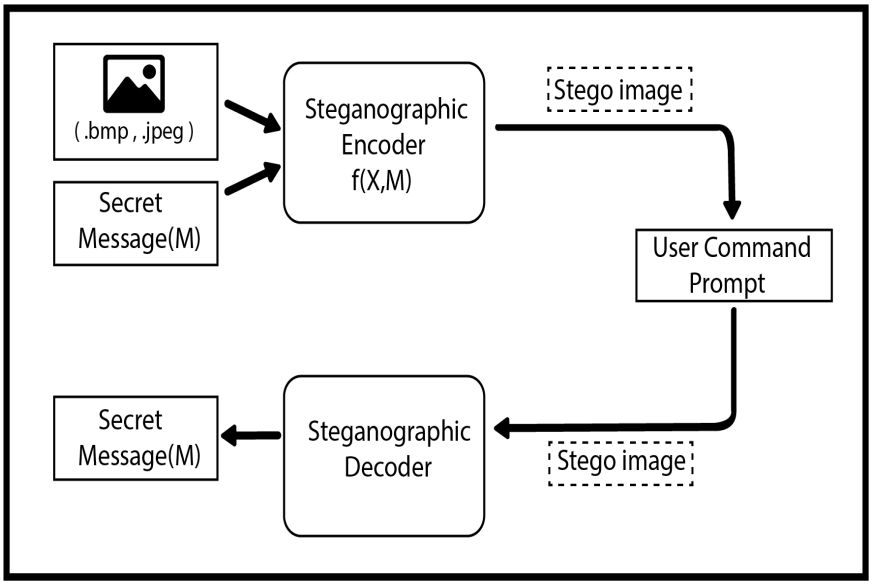

Figure 3.1: System Architecture Diagram

When applying LSB techniques to each byte of a 24-bit image, three bits can be encoded into each pixel.

Pixels : (00100111 11101001 11001000)
(00100111 11001000 11101001)
(10001000 00100111 11101001)

Figure 3.2: Bits Before Encoding

To hide the letter A whose binary value of ASCII code is 01000001, we would replace the LSB of these pixels to have the following new values:

Result : (00100110 11101001 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)

Figure 3.3: Bits After Encoding

If the user chooses decryption. The secret message is retrieved from the stego image using the same LSB algorithm.

Chapter 4

Implementation

The encode function takes in the image and text input file path, and reads the entire file path. Every pixel of image is converted to its 8-bit binary code. Simultaneously each character from the text file is converted to its binary value using their ASCII values. Now pixels are read from left to right in a group of 3 containing a total of 9 values. The first 8-values are used to store binary data. Finally the encoded image is stored in the specified output path.

```
void encode(string img, string txt, string out) {
    Mat image = imread(img);
    if (image.empty()) {
        cout << "\nImage Error";
        exit(-1);
    }
    ifstream file(txt);
    if (!file.is_open()) {
        cout << "\nText File Error";
        exit(-1);
    }
    char ch;
    file.get(ch);
    int bit_count = 0;
    bool last_null_char = false;
    bool encoded = false;
    for (int row = 0; row < image.rows; row++) {
        for (int col = 0; col < image.cols; col++) {
            for (int color = 0; color < 3; color++) {
                Vec3b pixel = image.at<Vec3b>(Point(row, col));
                if (isBitSet(ch, 7 - bit_count))
                    pixel.val[color] |= 1;
                else
                    pixel.val[color] &= ~1;

                image.at<Vec3b>(Point(row, col)) = pixel;
                bit_count++;
                if (last_null_char && bit_count == 8) {
                    encoded = true;
                    goto OUT;
                }
                if (bit_count == 8) {
                    bit_count = 0;
                    file.get(ch);
                    if (file.eof()) {
                        last_null_char = true;
```

Figure 4.1: Pseudo Code of Encryption Code

The decode function takes in the encrypted image file path. The function starts to read left bits from each pixel until 8th bit in encountered. All the 8 bits are grouped together to form the ASCII values finally decoding the character.

```
void decode(string out_img) {
    Mat image = imread(out_img);
    if (image.empty()) {
        cout << "\nImage Error";
        exit(-1);
    }
    char ch = 0;
    int bit_count = 0;
    cout<<"\nThe decoded message is : \n";
    for (int row = 0; row < image.rows; row++) {
        for (int col = 0; col < image.cols; col++) {
            for (int color = 0; color < 3; color++) {
                Vec3b pixel = image.at<Vec3b>(Point(row, col));
                if (isBitSet(pixel.val[color], 0))
                    ch |= 1;
                bit_count++;
                if (bit_count == 8) {
                    if (ch == '\0')
                        goto OUT;
                    bit_count = 0;
                    cout << ch;
                    ch = 0;
                }
                else
                    ch = ch << 1;
            }
        }
    }
OUT:;
}
```

Figure 4.2: Pseudo Code of Decryption Code

Chapter 5

Results and Discussion

Initially, menu based user interface will be displayed asking the user to select any one option. Once user selects the encryption option the application will ask user to enter the file path. Once the file path is pasted, the encryption process will begin. Once the encryption process is complete, the application will save the encrypted image in the output file path.

```
To Encode press 1
To Decode press 2
Enter your choice:1

---ENCODING---
Enter the image path(jpg,png or bmp):C:\\Users\\jacks\\OneDrive\\Documents\\MiniProject\\sample.bmp
Enter the text file path:C:\\Users\\jacks\\OneDrive\\Documents\\MiniProject\\text.txt
Enter the output image path:C:\\Users\\jacks\\OneDrive\\Documents\\MiniProject\\output.bmp

Hurray!Image is encoded
```

Figure 5.1: Encryption Summary

At the end, user will have option to select decode and also encode. User can decrypt the stego image. If they select the decrypt option then user has to add the path of encrypted image file. Soon after this the decrypting process starts and displays the encoded text to the user.

```
To Encode press 1
To Decode press 2
Enter your choice:2

---DECODING---
Enter the image path(jpg,png or bmp):C:\\Users\\jacks\\OneDrive\\Documents\\MiniProject\\output.bmp

The decoded message is :
Hello. Our FS Mini Project is about Image Stegnography.
```

Figure 5.2: Decryption Summary

Chapter 6

Conclusion

Steganography can protect data by hiding it but using it alone may not guarantee total protection. It is possible that by using a stegano encryption technique, enemy detects presence of text message in the image file and then the enemy may succeed in extracting information from the picture, which can be disastrous in real life situations. This is same for plain encryption. In this case by seeing the meaningless appearing sequence of bits enemy can detect that some illegal message is being sent and we may land–up in a problematic situation.However, if one uses both methods, this will lead to 'security in depth'. The message should first be encoded using a strong encryption algorithm and then embedded into a carrier.

References

[1] Michael J. Folk, Bill Zoellick, Greg Riccardi: File Structures-An Object Oriented Approach with C++, 3rd Edition, Pearson Education, 1998.

[2] K.R. Venugopal, K.G. Srinivas, P.M. Krishnaraj: File Structures Using C++, Tata McGraw-Hill, 2008.

[3] Scot Robert Ladd: C++ Components and Algorithms, BPB Publications, 1993.

[4] Raghu Ramakrishan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw Hill, 2003.

[5] Stefan Katzenbeisser: Information Hiding Techniques for Steganography and Digital Watermarking, Artech House Publishers, 2000.

[6] Abhijeet Shridhar: LSB based Image steganography, Geeks For Geeks, 2021.