A Report on

Hiding Message in Coloured Images at LSB by Double Encryption

Submitted in partial fulfillment for award of

Bachelor of Technology

Degree in

Computer Science & Engineering

By

M.SAI NEHA(Y17ACS485) **M.RAJIV GANDHI(Y17ACS482)**

S.SRAVAN KUMAR (L18ACS584) **S.JOHN PAUL(Y13ACS535)**

Under the guidance of

Mr.P.Nanda Kishore,M.Tech

Department of Computer Science and Engineering

Bapatla Engineering College

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

BAPATLA – 522 102, Andhra Pradesh, INDIA

2020-2021

Department of
Computer Science & Engineering

CERTIFICATE

This is to certify that the project report entitled **Guidelines for the Preparation of Project Thesis Department of CSE** that is being submitted by M.Sai Neha(Y17ACS485),M.Rajiv Gandhi(Y17ACS482), S.Sravan Kumar (L18ACS584), S.John Paul(Y13ACS535) in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

Signature of the Guide
P.NandaKishore
Asst.Professor

Signature of the HOD
Dr.Sk.Nazeer
Prof. & Head

DECLARATION

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

M.Sai Neha(Y17ACS485)

M.Rajiv Gandhi(Y17ACS482)

S.Sravan Kumar (L18ACS584)

S.John Paul(Y13ACS535)

Acknowledgement

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **[Asst.Prof P.Nanda Kishore, M.Tech(CSE)],** Department of CSE, for his/her valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. Sk.Nazeer**, Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal [**V. Damodara Naidu]** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **Dr .N.Sudhakar ,** Prof. Dept. of CSE for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

M.Sai Neha(Y17ACS485)
M.Rajiv Gandhi(Y17ACS482)
S.Sravan Kumar (L18ACS584)
S.John Paul(Y13ACS535)

Contents

List of Tables

List of Figures

Abstract

This is one of the most important feature ,Even if attacker is successful in destroying the steganographic technique then tamper resistance makes it difficult for the attacker to alter or damage the original data .While you can think of it as last step that as a sender we can do to protect our data from other people

The steganography word is derived from the combination of the steganos, which means "covered, hidden or protected", and the graphein, which means "writing". The main purpose of steganography studies is to prevent the hidden data from being to obtained by unauthorized persons. In order to realize the basic purpose of the steganography methods, there should be minimal change in the pattern file.

In this study, LSB method which is one of the methods of concealing digital image information is examined. In this study, a new method of data hiding is proposed in order to minimize the changes occurring in the cover file while hiding the data with LSB method and to create the most appropriate mask to make difficult to obtain hidden data.

1 Introduction

Today, with the rapid development of technology and the widespread use of the Internet, data sharing on the Internet has increased considerably .This network, which makes our life easier, comes with very serious security holes. Today, due to these security vulnerabilities, information security is a problem that needs to be addressed carefully.

Communicated messages can easily be obtained and modified by third parties .There are two important techniques used to ensure the confidentiality of the communication and to transmit the message in a secure way .They are cryptography and steganography.

1.1 Information Security

Information security [1], sometimes abbreviated to info sec, is a set of practices intended to keep data secure from unauthorized access or alterations, both when it's being stored and when it's being transmitted from one machine or physical location to another. You might sometimes see it referred to as data security. As knowledge has become one of the 21st century's most important assets, efforts to keep information secure have correspondingly become increasingly important.

The basic components of information security are most often summed up by the so-called CIA triad: confidentiality, integrity, and availability.

1.1.1 Confidentiality

Confidentiality is perhaps the element of the triad that most immediately comes to mind when you think of information security. Data is confidential when only those people who are authorized to access it can do so; to ensure confidentiality, you need to be able to identify who is trying to access data and block attempts by those without authorization. Passwords, encryption, authentication, and defense against penetration attacks are all techniques designed to ensure confidentiality.

1.1.2 Integrity

Integrity means maintaining data in its correct state and preventing it from being improperly modified, either by accident or maliciously. Many of the techniques that ensure confidentiality will also protect data integrity—after all, a hacker can't change data they can't access—but there

are other tools that help provide a defense of integrity in depth: checksums can help you verify data integrity, for instance, and version control software and frequent backups can help you restore data to a correct state if need be. Integrity also covers the concept of non-repudiation: you must be able to prove that you've maintained the integrity of your data, especially in legal contexts.

1.1.3 Availability

Availability is the mirror image of confidentiality: while you need to make sure that your data can't be accessed by unauthorized users, you also need to ensure that it can be accessed by those who have the proper permissions. Ensuring data availability means matching network and computing resources to the volume of data access you expect and implementing a good backup policy for disaster recovery purposes.

1.2 Cryptography

Cryptography is technique of securing information and communications through use of codes so that only that person for whom the information is intended can understand it and process it. Thus, preventing unauthorized access to secret message. The prefix “crypt” means “hidden” and suffix graphy means “writing”

Cryptography is the science of keeping information secure by transforming it into form that unintended recipients cannot understand. In cryptography, an original human readable message, referred to as plaintext, is changed by means of an algorithm, or series of mathematical operations, into something that to an uninformed observer would look like gibberish; this gibberish is called cipher text.

The process used for the conversion of the plain text into cipher text by using the symmetric key and this process is known as the encryption .One important aspect of the encryption process is that it almost always involves both an algorithm and a *key*. A key is just another piece of information, almost always a number that specifies how the algorithm is applied to the plaintext in order to encrypt it .Even if you know the method by which some message is encrypted, it's difficult or impossible to decrypt without that key. The sender takes the plain text or message and using a key ,it is converted into cipher text .The key is shared among the sender and receiver

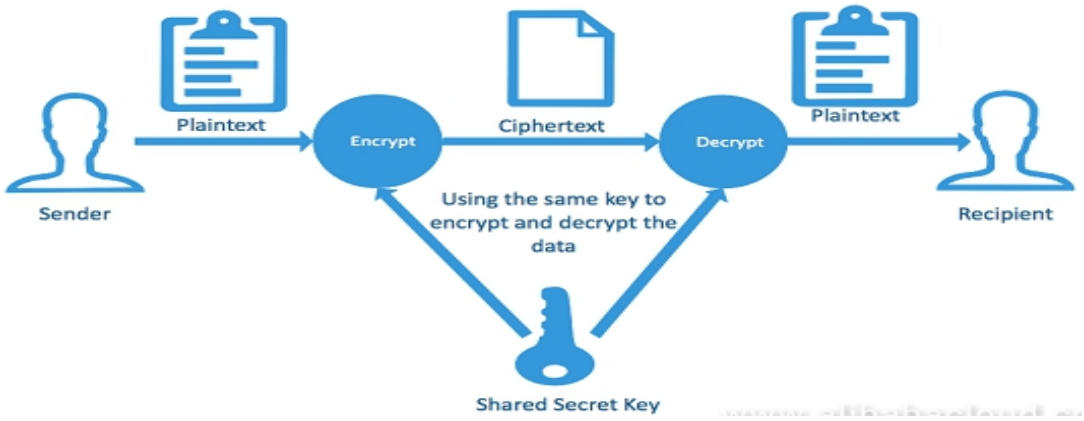

Figure 1.1 Cryptographic Model

. The receiver takes the cipher text, secret key and decrypt it to get the original message .Cryptography ensures the information security to some extent .But it doesn't hide the fact that information is being sent .This is where steganography comes into picture .The goal in steganography is to create the stego object and to transmit it to the buyer by placing important information in such a way that it does not appear in the ordinary cover object.

1.3 Steganography

The word "Steganography" comes from the Greek steganos (covered or secret) and graphy (writing or drawing) and thus means, literally, covered writing. It is a data hiding techniques, which aims at transmitting a message on a channel where some other kind of information is already being transmitted.

The goal of steganography is to hide messages inside the images in such a way that does not allow any "enemy" to even detect that there is a secret message present in the image. Steganography attempts to hide the existence of communication .Steganography [2] can be applied to many types of data, including audio, video, and images and can hide any kind of digital information.

Steganography relies on hiding message in unsuspected multimedia data and is generally used in secret communication between acknowledged parties .The technique replaces unused or insignificant bits of the digital media with the secret data. The concept is to embed the hidden

object into a significantly larger object so that the change is undetectable by the human eye. All digital file formats can be used for steganography, but the formats those are with a high degree of redundancy are more suitable.

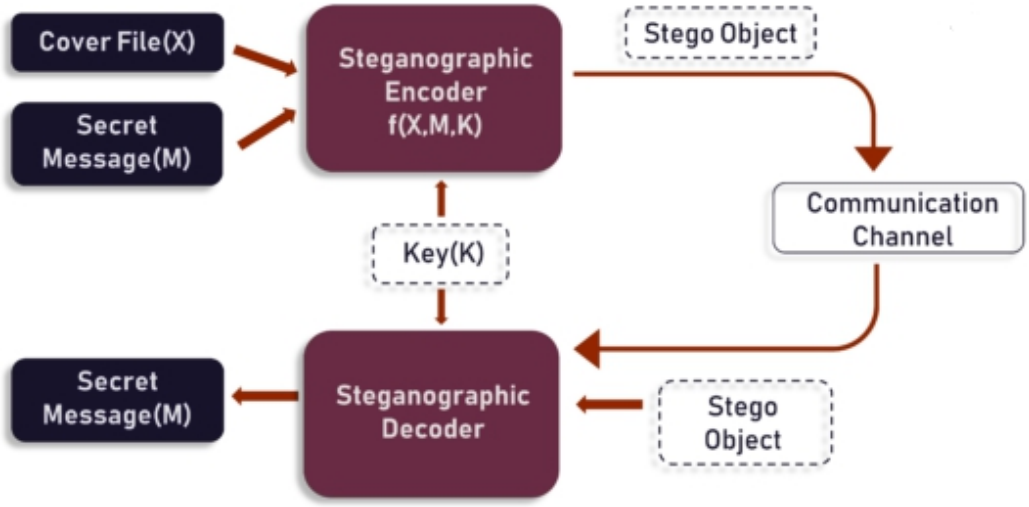

Figure 1.2 Basic Steganographic Model

As the Figure 1.2 depicts, both cover file(X) and secret message (M) are fed into steganographic encoder as input. Steganographic Encoder function, f(X, M, and K) embeds the secret message into a cover file. Resulting Stego Object looks very similar to your cover file, with no visible changes. This completes encoding. To retrieve the secret message, Stego Object is fed into Steganographic Decoder.

1.3.1 Characteristics of Steganography

There are three main characteristics of steganography .They are transparency, robustness, tamper resistance

Transparency

Transparency means the closeness of property between the stego and reconstructed files and the original cover and secret files, respectively Robustness [3] is the significant parameter in watermarking while transparency and hiding capacity are the most important for steganography. It's an important feature .Each cover media can be image or audio or video has certain information hiding capacity .If more information or data is hidden in the cover then, it results in degradation of cover media .

As you can see the stego image or final image Figure 1.4 after hiding the data inside our cover image is not proper or exactly similar to original image Figure 1.3.So there is some sort of distortion. If attacker identifies this distortion then our steganographic model fails and message is extracted and damaged by attacker

Figure 1.3 a lake before secret message insertion

Figure 1.4 a lake after secret message insertion

Robustness

Robustness is the ability of hidden message to go undamaged even if stego object undergoes some sort of transformation like cropping or scaling and blurring or linear or nonlinear filtering or some sort of hindrance .We have to make sure that technique in anyway doesn't affect our secret message.You can only study or describe the robustness relative to a certain context, i.e. assuming a specific application scenario and a specific attack scenario. To find robustness [4], aspects of the context includes

1. Choice of cover data.

2. Size of the cover data and size of the payload. Note that the problem does not scale linearly, as you will see from Andrew Ker's work on the square root law of steganography.

3. Steganalysis algorithm used by the attacker.

Tamper resistance

This is one of the most important features, Even if attacker is successful in destroying the steganographic technique then tamper resistance makes it difficult for the attacker to alter or damage the original data. While you can think of it as last step that as a sender we can do to protect our data from other people.

In the end, any application of strong steganography must ensure that the above features are satisfied, in other words they must ensure better perceptual transparency, robustness and tamper–resistance so that the integrity of the original work is maintained.

1.3.2 Types of Steganography

There are mainly 2 types of steganography methods. They are Linguistic steganography and Technical steganography.Linguistics steganography is a text based steganography.Technical steganography is steganography applied to pictures, sounds and videos. In this paper, we are going to take a short look at different steganography methods. Figure 1.5 below shows the different categories of file formats that can be used for steganography techniques [5]

Text Steganography

In text steganography formatting or by changing certain characteristics of textual elements can be changed. It consists of line-shift coding, word-shift coding and feature coding.

Image Steganography

Image is commonly used cover file. There are different file formats are available for digital images and for these file formats different algorithms are exist such as least significant bit insertion, Masking and filtering, Redundant Pattern Encoding, Encrypt and Scatter, Algorithms and transformations.

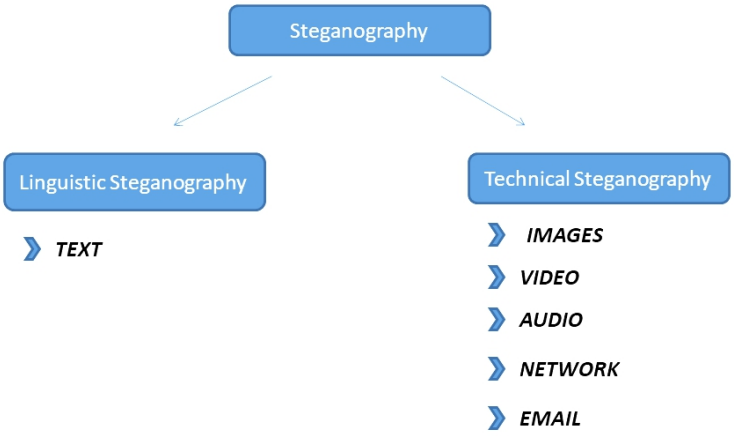

Figure 1.5 Types Of Steganography

Video Steganography

Video files consist of assortment of images and sounds, so most of the presented techniques on images and audio can be applied to video files too. An advantage of using video steganography is that large amount of data that can be hidden inside the cover file and it is the fact that it is flow of images and sounds

Audio Steganography

Secret message is embedded into digitized audio signal which result slender shifting of binary sequence of the equivalent audio file. There are a number of methods like LSB coding, Phase coding, spread spectrum, Echo hiding which are used for audio steganography.

Network Steganography

It is also called as protocol steganography .Protocol steganography embeds the information using network control protocol like http, ftp, tcp, Ssh, udp etc. Secret information is embedded in voice-over IP. Protocol steganography is an advance dimension of steganography and more secure than other dimensions.

Email Steganography

Email-based steganography is one of the most popular techniques for the text steganography. This technique hides the secret message within the email body and email addresses. Email addresses are used to hide the secret data, regardless they are valid addresses or not**.**

2 Literature Review

In Steganography, The most well-known techniques to data hiding in images are least significant bit (LSB) substitution, and masking & filtering techniques. LSB is a simple approach to embedding information in an image. But image manipulation can destroy the hidden information in this image. Applying LSB technique to each byte of a 24-bit image, three bits can be encoded into each pixel, as each pixel is represented by three bytes. Applying LSB technique to each byte of an 8-bit image, only one bit can be encoded into each pixel, as each pixel is represented by one byte

2.1 LSB technique

Steganography is the embedding of a data into another data. The LSB method [6] is the most commonly used method in image steganography. In the LSB method, each bit of the message to be hidden is written to the last bit of each byte of the image file. The data hiding process with the LSB method is as shown in Figure 2.1

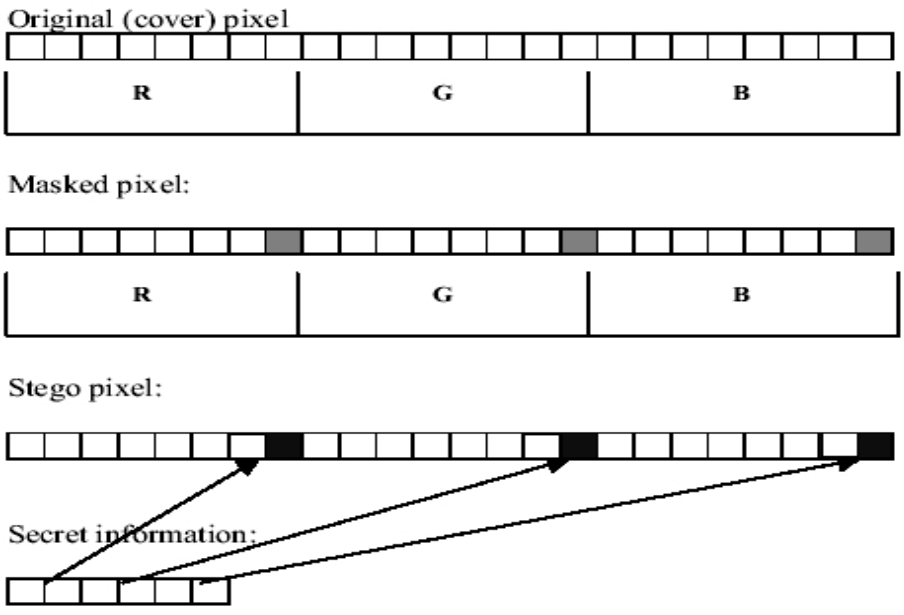

Figure 2.1 Data Hiding by LSB Method

When data is hidden, there is no obligation to follow a linear sequence starting from the first byte until the last byte. In the process of hiding the data, the order of the bytes to be used is arranged by means of a key called stego-key, and data can be hidden in completely mixed order.

In the Lsb method, adding to the last two bits rather than adding to the last bit doubles the amount of data that can be hidden. When the last two bite data hiding process is performed, the change on the cover object is still invisible, but it is more likely to be determined by steganalysis method

LSB Technique with an example

In LSB steganography, the least significant bits of the cover media's digital data are used to conceal the message. The simplest of the LSB steganography techniques is LSB replacement. LSB replacement steganography changes the last bit of each of the pixel values to reflect the message that needs to be hidden.

Consider an 8- bit gray scale bitmap image where each pixel is stored as a byte representing a gray scale color value. Suppose the first eight pixels of the original image have the following gray color values:

01010010	01001010	10010111	11001100	11010101	01010111	00100110	01000011

To hide the letter Z whose binary value of ASCII [7] code is 10110101; we would replace the LSBs of these pixels to have the following new values:

01010011	01001010	10010111	11001101	11010100	01010111	00100110	01000011

2.2 Hiding Message in colored images using double decryption

Images [8] are often referred to as RGB images that use channels. Channels are a very simple idea that is easy to use.If we took look at our computer monitor's display with a magnifying lens we would see that it consists of a very large number of triplets of red, green and blue dots. there always will be triplets of red, green and blue elements. In most monitors the dots are quite large and easy to see even with a weak lens.

Images on the computer display are formed when the monitor precisely varies the brightness of the red, green and blue elements in each triplet. Because the dots are close together

the human eye will fuse the three red, green and blue dots of varying brightness into a single dot that appears to be the color combination of the three levels of red, green and blue color amazing, but true! From here on in we'll refer to the Red, Green and Blue elements of images as "R", "G" and "B"

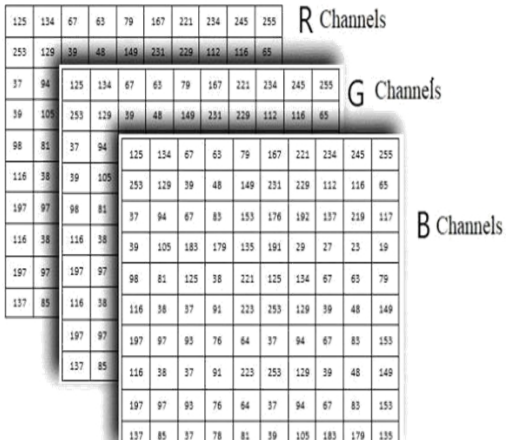

Figure 2.2 Color Channels of a Picture

Images on computers are usually nothing more than a series of number triplets where each triplet of three numbers is intended to control a single triplet of R, G and B dots on the monitor. Each triplet of numbers is a single pixel. A number triplet such as 67, 228, and 180 indicates to turn the R element up to 67 brightness and the G element to 228 brightness and the B element to 180 brightness. The result will be seen by humans as a pretty shade of green-blue color. The RGB numbers are called color values

2.2.1 Implementation Overview

In the proposed system we concentrate on finding some algorithm to hide the data inside images using steganography technique. An algorithm is designed to hide all the data inputted within the image to protect the privacy of the data. Then, the system is developed based on the new steganography algorithm.

This proposed system provides the user with two options encrypt and decrypt the data, in encryption the secret information is hiding in with image file, and on the other side the decryption is getting the hidden information from the stego image file, and also the user can

show the image size after and before the encryption. The processes of encryption and decryption of the data file consists of:

- Providing security for the data to be transmitted through network using steganography.
- Proposing an approach for hiding the data within an image using a steganographic algorithm which provides better accuracy and quality of hiding.

Figure 2.3 shows the data hiding algorithm block diagram. It starts with selecting a cover image then the secret message to be inserted is taken. Then we have to analyze the image. If data size is larger than the size of the image file then we have to take a bigger size cover image to fit the secret message. If the data size is less than the image size then we can insert the message into the cover file.

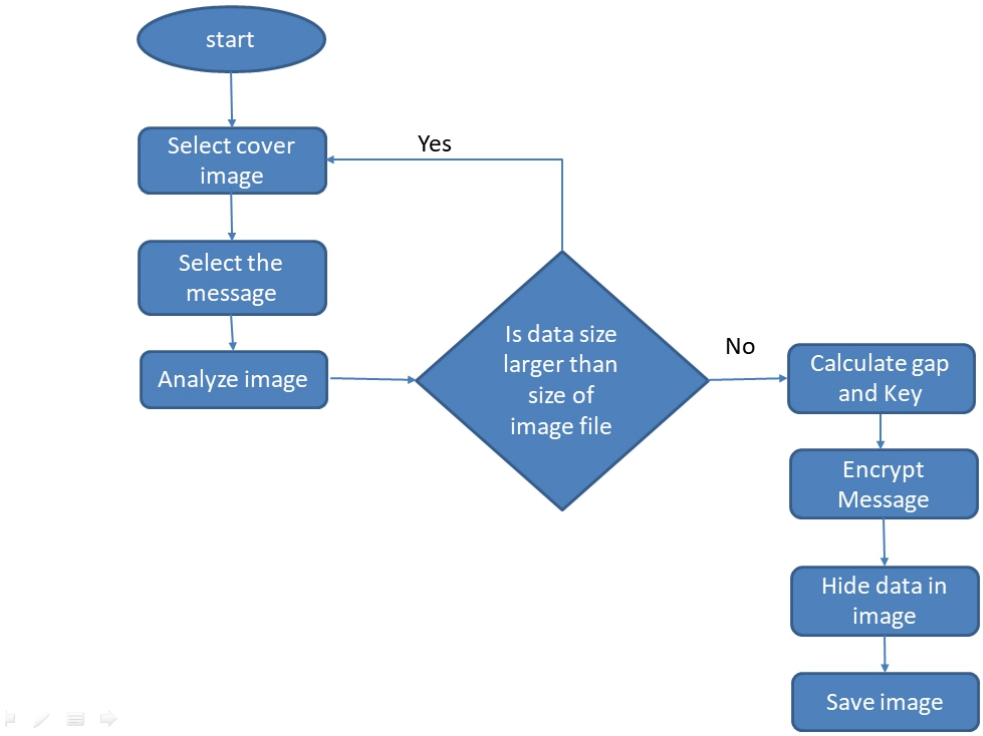

Figure 2.3Data Hiding Algorithm Block Diagram

The gap and key are calculated. The key is the first digit in the total number of pixels of the image. This key is converted to binary and inserted in the Red color channel of the image.

The message is then encrypted using cryptographic technique. The cryptographic technique used here is additive cipher.

After encrypting the message, the cipher text is converted into ASCII values and then the ASCII value is converted into binary value since the colour component of a pixel in an image is represented by binary number. The binary values obtained are hidden in the image lsb positions.The binary bits of the message are embedded in the image starting from first pixel to last pixel.

Suppose we have 10 frames and 4 bits, we have to fill each frame with only 1 bit. Bits are filled throughout the frames. Between the target frames (bits to be inserted), there is a gap which is constantly maintained throughout the image.

Figure 2.4 Original Image

Figure 2.5 Encoded Image

This gap is also converted into binary and inserted in Green colour channel. After the bits are inserted in the targeted frames in the image, the image is sent to receiver. Receiver obtains the key from Red colour channel, gap from Green colour channel and uses them to obtain the message.

During Decoding, gap is extracted from the green channel. By using gap, message is retrieved from the blue channel in the encrypted format. Key shift is extracted from the red channel. The encrypted text obtained is converted is converted into plain text using key.The original image Figure 2.4 is similar to encoded image Figure 2.5.But the encoded image has a secret image “This is a hidden message” inserted in it.

Double Encryption here says that the without the key and gap, we cannot extract the original secret message. Even if intruder fails to extract at least one of key or gap ,then the secret message cannot be obtained. Moreover, Even if the intruder is able to extract the secret message, he have to break the cryptographic technique. Hence, this process is much safer than classical lsb and gives maximum results

2.2.2 Results and Discussion

.The fact that the size of the file does not change reduces the visibility of the hidden data. This method aims to hide the data in the text file so that there is minimal change in the digital image file. This method differs from the LSB method by encrypting the data using the mask to provide least modification, and by hiding the data by splitting the images.

In the LSB method, the addition of data is done only on the last bit and in sequential order. Data hiding by the LSB method is easily discerned and data can be easily obtained. In this method, the data cannot be easily solved even if the data is obtained because it is hidden both in frames and without encryption.

Table 2- 1 Concealed data obtained by improved lsb method

	Original Text	The Resulting Text
LSB method	This is a hidden message	Uijitjtijeefonfttbhf

2.2.3 Conclusion

In the experimental study, the classical LSB method was investigated and two of the disadvantages of the LSB method were tried to be improved. The first of these disadvantages is the double encryption, and the second is the easy acquisition of the hidden data is not possible. Both key and gap is used for encrypting the data and providing at least bit change while concealing the data.

One of the disadvantages of the proposed method is that it is limited by the size of the cover file to be hidden as it is in the classical LSB method. This disadvantage has been attempted to be increased by three times the amount of data to be concealed using three of the RGB color channels of the cover file.

Another drawback of the improved lsb method is that size of data to be hidden is reduced since the bits of the cover object use to hide the key and gap. When the security of the issue is addressed, this disadvantage can be ignored since it is difficult to obtain the hidden value while the capacity is reduced. It is possible to develop this method with a more complex encryption method and when the data is hidden by the proposed method

3 Code

```
[ ]  import cv2
```

Open CV is a free open source library used in real-time image processing. It's used to process images, videos, and even live streams, but in this tutorial, we will process images only as a first step.

```
[ ]  def get_binary(n):
         return f'{n:08b}'
```

This method converts the integer value to binary format i.e., 8bits either 0's or 1's

- {} places a variable into a string
- 'n' is the integer to be converted to binary
- : adds formatting options for this variable
- 08 formats the number to eight digits zero-padded on the left
- b converts the number to its binary representation
- 'f'-F-strings provide a concise, readable way to include the value of Python expressions inside strings.

```
[ ]  def get_decimal(bin):
         bin = '0b' + bin
         return int(bin, 2)
```

This method converts the binary number to decimal

- 0b converts the number to its binary representation
- Int(bin,2) converts the binary to integer and returns an integer

```
[4]  def msg_to_bin(msg):
         ascii = []
         binmsg = ""
         for i in msg:
             ascii.append(ord(i))  # returns ascii value of 'i'
         for i in ascii:
             binmsg += get_binary(i)
         return binmsg
```

This method converts the message to binary for storing in the image

- Msg: It is the message to be converted to binary
- Ascii.append(ord(i)): It is an in-built method that returns the ascii value of a character

```
[5]  def first_digit(n):
         while n >= 10:
             n = n / 10
         return int(n)
```

This method returns the first digit of the integer.

```
def encrypt(string, shift):
    cipher = ''
    for char in string:
        if char == ' ':
            cipher += char
        elif char.isupper():
            cipher += chr((ord(char) + shift - 65) % 26 + 65)
        else:
            cipher += chr((ord(char) + shift - 97) % 26 + 97)
    return cipher
```

This method is used to convert the plane text into cipher text. This is a cryptographic technique used for information security. It is called additive cipher

The simplest mono-alphabetic cipher is additive cipher. It is also referred to as ‘Shift Cipher’ or ‘Caesar Cipher’. As the name suggests, ‘addition modulus 2’ operation is performed on the plain-text to obtain a cipher-text.

C = (M + k) mod n where,

C -> cipher-text, Here it is referred as cipher

M -> message/plain-text, Here it is referred as char

k -> key, Here key is referred as shift

```
def encode_img(img, msg):
    msglen = len(msg) * 8
    k = idx = ix = count = space = 0
    shaper = img.shape
    r = shaper[0]
    c = shaper[1]
    pixels = r * c
    gap = int(pixels / msglen)
    gapstr = str(gap) + "$"
    bingap = msg_to_bin(gapstr)
    gap = int(gap + 1)
    key = first_digit(pixels)
    print("KEY is ", key)
    msg = encrypt(msg, key)
    keystr = str(key) + "$"
    binkey = msg_to_bin(keystr)
    binmsg = msg_to_bin(msg)
```

This method takes the stego object (image) and secret message to insert the secret message in it.Here, required objects and variables are initialised.gap and key are calculated. The key is the first digit in the total number of pixels of the image. This key is converted to binary and inserted in the Red color channel of the image.

After encrypting the message using the encrypt method in the above figure, the cipher text is converted into ASCII values and then the ASCII value is converted into binary value since the colour component of a pixel in an image is represented by binary number. The binary values obtained are hidden in the image lsb positions. The binary bits of the message are embedded in the image starting from first pixel to last pixel.

```
for i in range(0, r):
    for j in range(0, c):
        if ix == len(bingap):
            break
        gpx = img[i, j, 1]  # taking GREEN from rgb
        gbinary = get_binary(gpx)  # converting px to binary
        gbinary = gbinary[0:-1]  # exclude last bit
        gbinary += bingap[ix]  # adding bit from msg at last position
        img[i, j, 1] = get_decimal(gbinary)  # convert binary value to decimal and replace it in pixel
        ix += 1
```

Here, GAP is converted into binary and each bit is inserted into GREENCHANNEL.GAP is the distance between frames in which bits of message are to be hidden.Forexample, 3 bits need to be filled in 10 frames.Here GAP is 3(i.e., 10/3) and frames needed are 1,5,9

```
for i in range(0, r):
    for j in range(0, c):
        if k == len(binkey):
            break
        rpx = img[i, j, 0]  # taking RED from rgb
        rbinary = get_binary(rpx)  # converting px to binary
        rbinary = rbinary[0:-1]  # exclude last bit
        rbinary += binkey[k]  # adding bit from msg at last position
        img[i, j, 0] = get_decimal(rbinary)  # convert binary value to decimal and replace it in pixel
        k += 1
```

Here, KEY(SHIFT) is converted into binary and each bit is inserted into RED CHANNEL.The first digit in the total no of pixels in the image is taken as KEY.For example, if the total no of pixels in the image of resolution 1080x1920 are 2,073,600. Then 2 is taken as KEY

```
for i in range(0, r):
    for j in range(0, c):
        count += 1
        if count == space or count == 1:
            if idx == len(binmsg):
                return
            bpx = img[i, j, 2]  # taking BLUE from rgb
            bbinary = get_binary(bpx)  # converting px to binary
            bbinary = bbinary[0:-1]  # exclude last bit
            bbinary += binmsg[idx]  # adding bit from msg at last position
            img[i, j, 2] = get_decimal(bbinary)  # convert binary value to decimal and replace it in pixel
            idx += 1
            space += gap
print("GAP is ", gap)
```

Here, CIPHER TEXT is converted into binary and each bit is inserted into BLUE CHANNEL at frames after GAP.For example, 4 bits need to be filled in 10 frames. Here GAP is 2(i.e., 10/4) and frames needed are 1,4,7,10.

```
[16] img = cv2.imread('/content/dog.jpg')
     msg = input("enter message:")
     encode_img(img, msg)
     f=cv2.imwrite('encodedIMG.jpg', img)
     if f==True:
       print("Message Inserted Successfully")
     else:
       print("Message is not Inserted")
```

```
enter message:"my password is 1216@"
KEY is  1
GAP is  689
Message Inserted Successfully
```

cv2.imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix. The image should be in the working directory or a full path of image should be given.

Syntax: cv2.imread(path, flag)

path: A string representing the path of the image to be read.

flag: It specifies the way in which image should be read. It's default value is cv2.IMREAD_COLOR.Return Value: This method returns an image that is loaded from the specified file.

The image inserted is Figure 2.4.The secret message is "my password is 1216@" .The message is inserted successfully. The output image generated is encoded image with the secret information as shown in the Figure 2.5 .

Decoding the secret Information:

```
[ ] import cv2
```

Open CV is a free open source library used in real-time image processing. It's used to process images, videos, and even live streams, but in this tutorial, we will process images only as a first step.

```
[ ]  def get_binary(n):
         return f'{n:08b}'
```

This method converts the integer value to binary format i.e., 8bits either 0's or 1's

- {} places a variable into a string
- 'n' is the integer to be converted to binary
- : adds formatting options for this variable
- 08 formats the number to eight digits zero-padded on the left
- b converts the number to its binary representation
- 'f'-F-strings provide a concise, readable way to include the value of Python expressions inside strings.

```
[ ]  def get_decimal(bin):
         bin = '0b' + bin
         return int(bin, 2)
```

This method converts the binary number to decimal

- 0b converts the number to its binary representation
- Int(bin,2) converts the binary to integer and returns an integer

```
[4]  def decrypt(string, shift):
         decipher = ''
         for char in string:
             if char == ' ':
                 decipher += char
             elif char.isupper():
                 decipher += chr((ord(char) - shift - 65) % 26 + 65)
             else:
                 decipher += chr((ord(char) - shift - 97) % 26 + 97)
         return decipher
```

This method is used to convert the cipher text into plane text. This process is called decryption

M = (C – k) mod n where,

C -> cipher-text, Here it is referred as cipher

M -> message/plain-text, Here it is referred as char

k -> key, Here key is referred as shift

```
def get_key(img):
    shaper = img.shape
    r = shaper[0]
    c = shaper[1]
    key = tmp = ""
    for i in range(0, r):
        for j in range(0, c):
            rpx = img[i, j, 0]  # taking RED from rgb
            rbinary = get_binary(rpx)  # converting px to binary
            tmp += rbinary[-1]  # getting last bit
            if len(tmp) == 8:
                ascii = get_decimal(tmp)  # converting binary to ascii
                char = chr(ascii)  # converting ascii to character
                tmp = ""
                if char == "$":
                    print("KEY is", key)
                    return key
                else:
                    key += char
```

KEY(SHIFT) is extracted from the REDCHANNEL. We have to take red channel from rgb values and then convert them into binary. Then extract the last bit from all the pixels. Then after a count of 8 bits ,we have to convert the binary number into integer which is a ascii value. Then we have to identify the character for the ascii value obtained.

```
def get_gap(img):
    shaper = img.shape
    r = shaper[0]
    c = shaper[1]
    gap = tmp = ""
    for i in range(0, r):
        for j in range(0, c):
            gpx = img[i, j, 1]  # taking GREEN from rgb
            gbinary = get_binary(gpx)  # converting px to binary
            tmp += gbinary[-1]  # getting last bit
            if len(tmp) == 8:
                ascii = get_decimal(tmp)  # converting binary to ascii
                char = chr(ascii)  # converting ascii to character
                tmp = ""
                if char == "$":
                    return gap
                else:
                    gap += char
```

Gap is extracted from green channel. We have to take the green channel from rgb values of the image. Then we have to convert the pixel values to binary. Then we have to extract the last bit. After the count of 8 bits, convert the binary value to ascii. The ascii values gives the gap value.

```
def decode_img(img):
    shaper = img.shape
    r = shaper[0]
    c = shaper[1]
    msg = tmp = ""
    space = count = 0
    gap = int(get_gap(img)) + 1
    for i in range(0, r):
        for j in range(0, c):
            count += 1
            if count == space or count == 1:
                bpx = img[i, j, 2]  # taking BLUE from rgb
                bbinary = get_binary(bpx)  # converting px to binary
                tmp += bbinary[-1]  # getting last bit
                if len(tmp) == 8:
                    ascii = get_decimal(tmp)  # converting binary to ascii
                    char = chr(ascii)  # converting ascii to character
                    tmp = ""
                    msg += char
                space += gap
    print("GAP is ", gap)
    return msg
```

By using GAP, MESSAGE is retrieved from BLUE CHANNEL in encrypted image. We have to take blue channel from rgb values and then convert them into binary. Then extract the last bit from all the pixels. Then after a count of 8 bits ,we have to convert the binary number

into integer which is a ascii value. Then we have to identify the character for the ascii value obtained.

```
img = cv2.imread('encoded.png')
print(decrypt(decode_img(img), int(get_key(img))))
```

```
GAP is  632
KEY is 1
this is a hidden message
```

The extracted message is “this is a hidden message”. Thus through decoding we extract the hidden message.

4 Bibliography

[1] Josh Fruhlinger. (2020, March) CSO. [Online]. https://www.csoonline.com/article/3513899/what-is-information-security-definition-principles-and-jobs.html

[2] Yildiray YIGIT Bitlis Eren University Murat KARABATAK Firat University Faculty of Technology, "Developing LSB Method in colored images,"*LSB* , vol. 1, p. 6, march 2018.

[3] Archana Choudary. (2020, November 20) edurekha. [Online]. https://www.edureka.co/blog/steganography-tutorial

[4] IETEC, "A Robust Image Steganography on Resisting JPEG Compression with No Side Information,"*New Adavnces of Privacy Protection and Multimedia Content Security of Big Data and Cloud Computing*, vol. 35, p. 250, 11 jun 2018.

[5] N.Thinaharan and M.Vasanthi B.Chitradevi, "Data Hiding Using Least Significant Bit Steganography in Digital Images,"*Statistical Approaches on Multidisciplinary Research*, vol. 1, p. 8, may 2017.

[6] Dalia Nashat & Loay Mamdouh, "An efficient steganographic technique for hiding data,"*Journal of the Egyptian Mathematical Society 27, Article number: 57 (2019)*, vol. 2, p. 45, December30 2019.

[7] American Standards Association's. (1961, May) wikipedia. [Online]. https://en.wikipedia.org/wiki/ASCII

[8] Manifold Software Limited. (1993-2017, december) manifold.net. [Online]. www.manifold.net