1. Introduction

Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. Sometimes it is unable to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement this, is called steganography. Steganography is the art and science of invisible communication. It helps to hide information in ways that prevent the detection of hidden messages. The word steganography comes from the Greek words "stegos" meaning "cover" and "grafia" meaning "writing" defining it as "covered writing". In image steganography the information is hidden exclusively in images. Today steganography is mostly used on computers with digital data communication through internet. Cryptography is used to hide secret information from other. Steganography differs from cryptography in the sense that where cryptography focuses on keeping the contents of a message secret, steganography focuses on keeping the existence of a message secret. Steganography and cryptography are both ways to protect information from third party access. Other technologies that are closely related to steganography is watermarking. In watermarking all of the instances of an object are "marked" in the same way. The kind of information hidden in objects when using watermarking is usually a signature to signify origin or ownership for the purpose of copyright protection. In watermarking the fact that information is hidden inside the files may be public knowledge – sometimes it may even be visible – while in steganography the imperceptibility of the information is crucial.

2. Background Study

Steganography become more important as more people communicate through internet around the world. To understand the project perfectly we have to know the structure of a BMP file and the knowledge of structured programming. A BMP file consists of either 3 or 4 parts as shown in the diagram. Header size is 14 byte, info header size 40 byte, padding size differs, rest of the byte are image date.

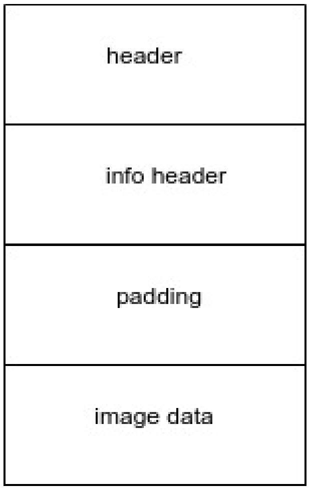


Figure 1: BMP file format.

Steganography include an array of secret communication methods that hide the message from being seen or discovered. In this modern time and also in the past we store the valuable object in the box or locker, so that no one can understand that something is hidden inside the box or locker. In image steganography we use this kind of technic. Watermarking is related to image steganography. So, we need to know the process of watermarking. We insert the secret message inside the image. For this reason we need to read every byte of the image pixel and know how to manipulate it and change the pixel value as least as possible so that the host image does not affect so much. The technique used in steganography is least significant bit (LSB) insertion and manipulation. We need to know how to implement LSB method. The main goal of steganography is to avoid detection of existing secret message inside the image.

3. Broad Domain

Image steganography is a technic of hiding information from other. In this process no one can understand that information is hidden inside the image. Steganography helps to make a secure communication system. For proper steganography we need to implement all the process correctly. Steganography is widely use around the world. The uses of the technic.

1. Secured data transfer through internet.
2. Military communication and transfer their plan.
3. Image steganography helps to reduce the copyright image and video.
4. Only owner can identify the secret message inside the image
5. Visible watermarking is used in satellite TV channels.

4. Challenges

In image steganography the main task is to hide message so that no one can understand that something is hidden initially. We use least significant bit (LSB) insertion method to insert the message. In this insertion process we can insert a single bit in the LSB of the RGB byte. Otherwise the RGB will change and the image also change forever. We have to convert the every byte of message data corresponding 8 bit binary when we insert the message. Otherwise it is impossible to decode the exact message which is encoded in the image.

5. Dependencies

Successfully run the image steganography program the required software and hardware requirements are:

5.1. Software

The project is programmed in "C++" language. I used Linux based gedit text editor to write the program of this project. To run the program we need a well developed "C++" environment and g++ compiler. User can use Windows or Linux operating system to run this project code.

5.2. Hardware

Since this program deals with image pixel and a bit calculations, it needs a good processor to run without any complication. I have used a computer with core i5 processor to develop this. But it works in core i3 processor too.

6. Methodology

Image steganography focus on hiding message inside the image. In this project we work with BMP image file. Using "fstream" header we read and write the BMP image. We usually open the BMP image file binary mode use file pointer.read() and file pointer.write() to read and write also. Store all the binary data in a character array to manipulate later implementation. We use 1D character array to store header and padding data and use 3D character array to store the pixel RGB(Red, Green, Blue) value. In this project we try to hide text and image data.

6.1 Text Hiding

First we insert an image which is used as a covered image. We use a text file (.txt) to insert inside the image pixel data. We read the text file and store it as a binary of the character byte. After that we insert the binary bit into LSB of the image pixel.

6.2 Image Hiding

First we insert an image which is used as a covered image. We use a image file (.bmp) to insert the image pixel data. We read the image file in which way we read and store the cover image. Without the header part of the image we convert the total byte as corresponding binary. After that we insert the binary bit into LSB of the image pixel. In this process the size ratio of the cover image and inserting image is 5 : 1.

7. Achievements

Through this project, I have learnt the file format of the BMP image and basic process of reading and writing BMP image, manipulate the image pixel RGB value. Using LSB method how to change the bit to insert message and extract the changing bit after that convert into message. How to reuse the method and make the program efficient and enrich the knowledge about structured(functional) programming.

8. Analysis and Design

We try to design this project as understandable and efficient for work. For this reason we try to divide the project as little as possible and create function to reuse it as much and try to access less memory. The design of the project described below:

1. First we divide the project into four parts. Such as: read and store cover image, text file, inserting image and decode the secret message.

2. After storing the text file and inserting image convert the byte data to binary data.

3. Insert the binary data in the Least Significant Bit (LSB) of the cover image pixel byte. We insert only two bit in every pixel of the cover image.

4. After inserting the secret data successfully create a stego image.

5. We insert a signature byte to denote that the hiding message is text or image.

6. Decode the secret message from the cover image if the image is encoded.

7. After decoding the cover image if the secret message is text then print the message in the terminal otherwise create the image.

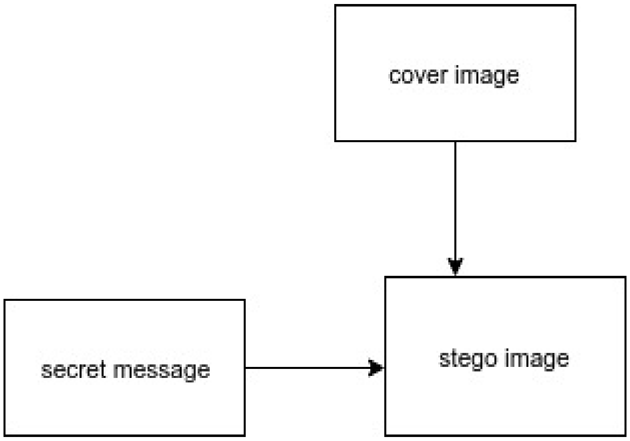


Figure 2: Create stego image.

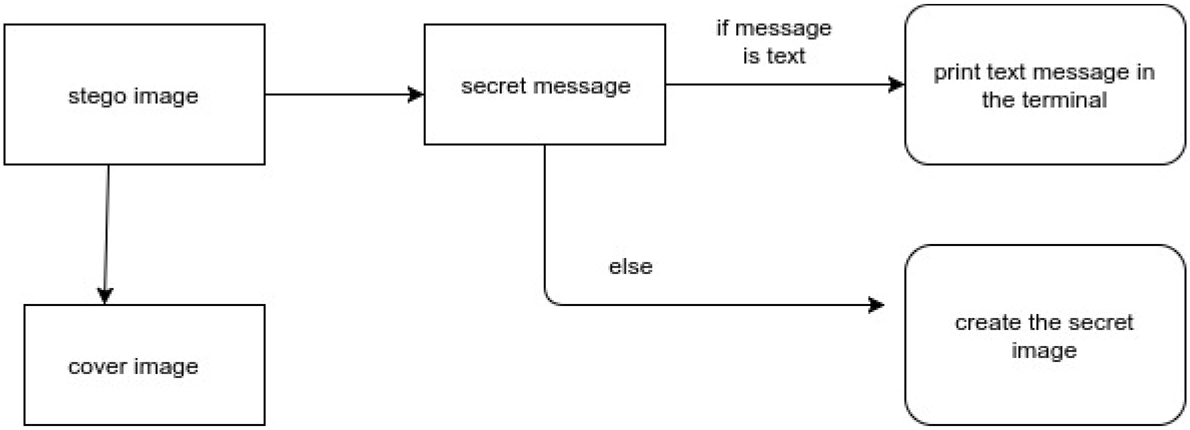


Figure 3: Decode the stego image.

9. Implementation and Testing

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image. When using a 24-bit image, a bit of red and blue color is changed. In other words, we can store 2 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 960,000 bits or 120,000 bytes of embedded data. For example a grid for 4 pixels of a 24-bit image can be as follows:

Pixel 1: (00101101 00011100 11011100)

Pixel 2: (10100110 11000100 00001100)

Pixel 3: (11010010 10101101 01100011)

Pixel 4: (10100110 11000100 01101011)

When a character 'A', which binary representation is 01000001, is embedded into the least significant bits of this part of the image, the resulting pixel is as follows:

Pixel 1: (0010110**0** 00011100 1101110**1**)

Pixel 2: (1010011**0** 11000100 0000110**0**)

Pixel 3: (1101001**0** 10101101 0110001**0**)

Pixel 4: (1010011**0** 11000100 0110101**1**)

10. Program Output

Input image:

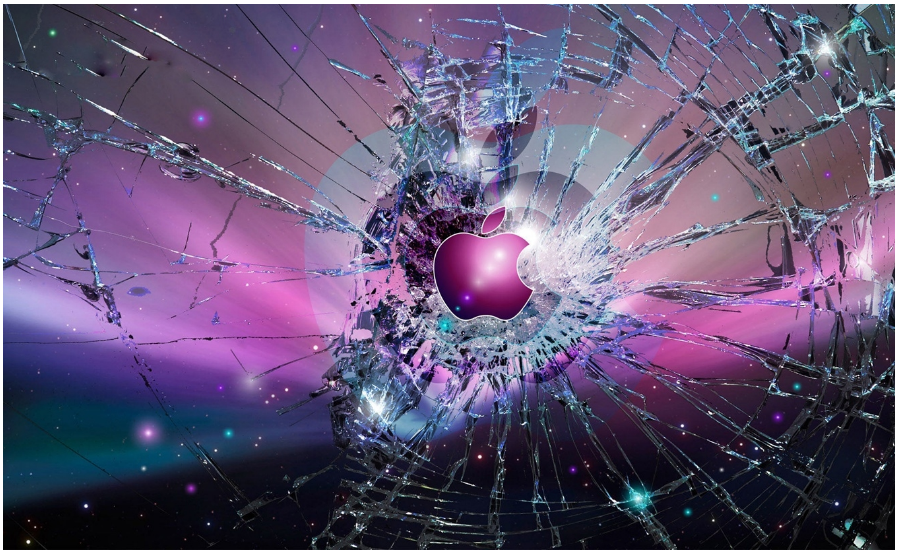
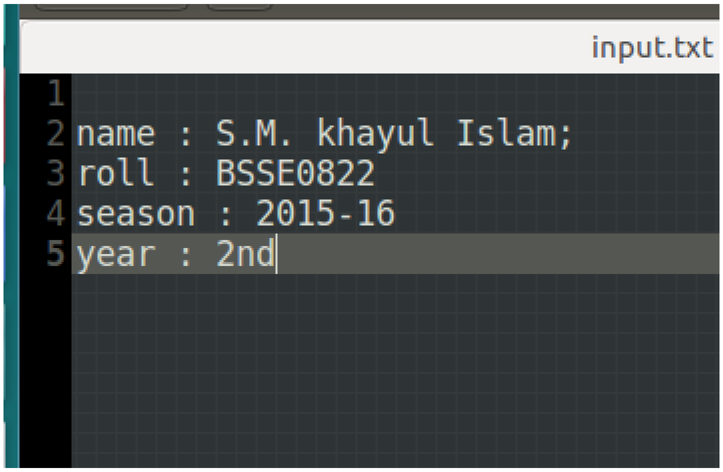


Figure 4: input image

Secret input text message screenshot:



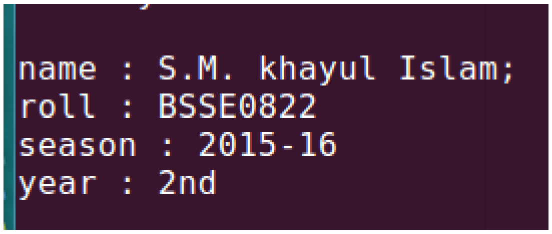
```

name : S.M. khayul Islam;
roll : BSSE0822
season : 2015-16
year : 2nd
```

Figure 5: secret text message screenshot.

Stego image with secret text message:

Figure 6: Stego image with secret text message

Secret text message output after decode:



```
name : S.M. khayul Islam;
roll : BSSE0822
season : 2015-16
year : 2nd
```

Figure 7: secret text output.

Secret image message input:

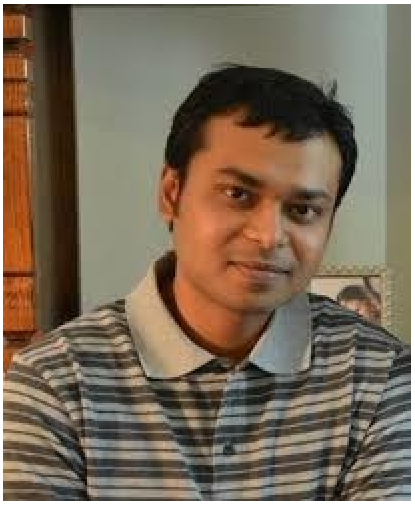


Figure 8: secret image

Stego image with secret image message:

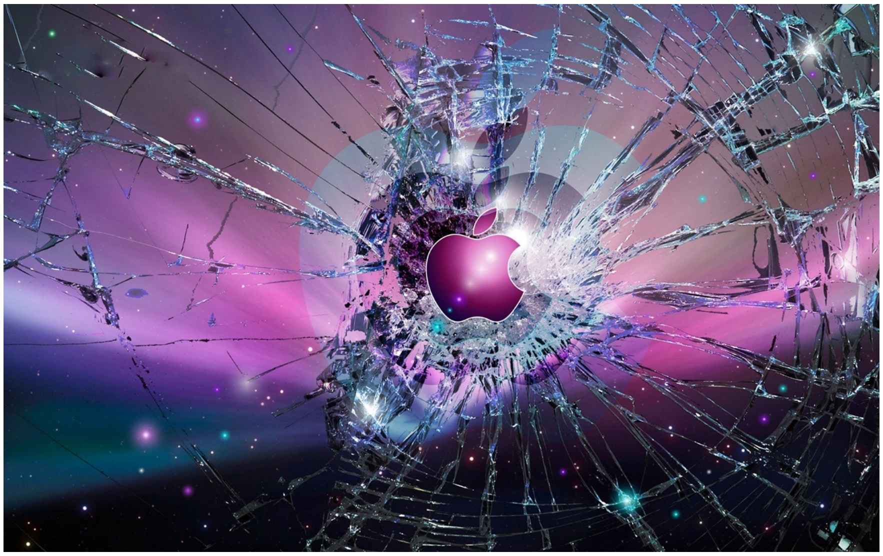


Figure 9: Stego image with secret image message.

Secret image message after decode:

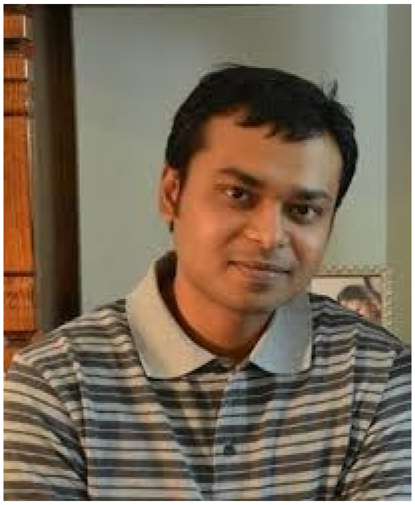


Figure 10: secret image output.

Stego image with visible image:

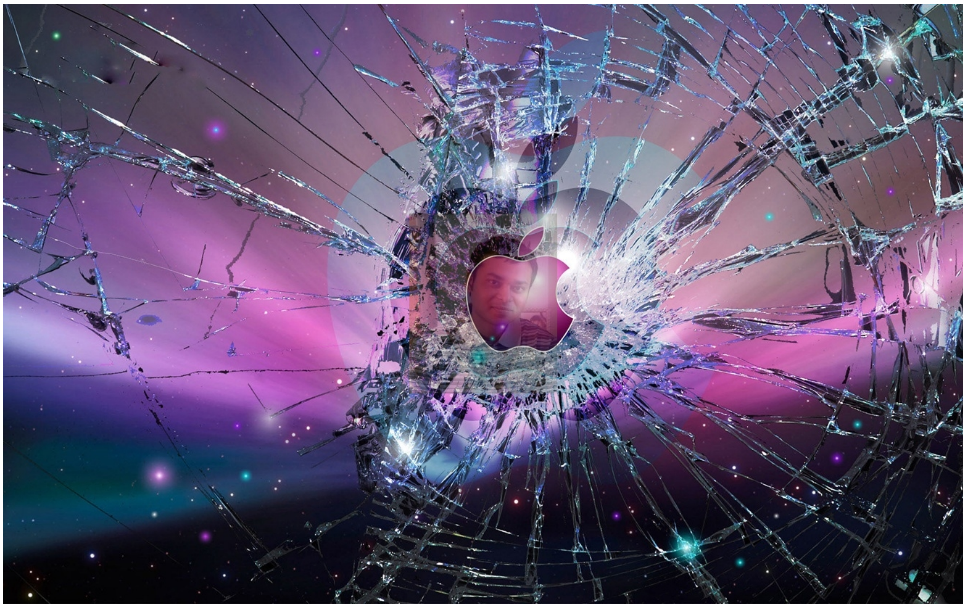
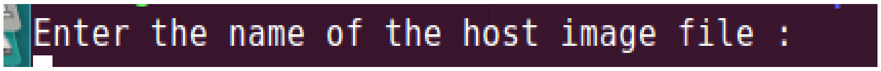


Figure 11: visible mark.

11. User Manual

Before execute the program it is good to create a folder and move the program, secret message, secret image into the folder and then run the program. In this case all the input and output are remain in the same folder and easy to understand.

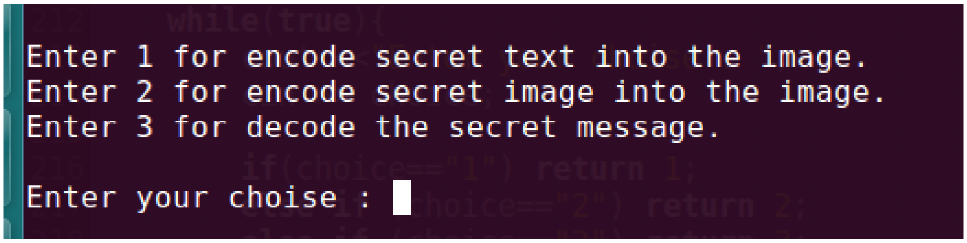
First user have to insert a bmp image file name.



```
Enter the name of the host image file :
```

Figure 12: input image file name.

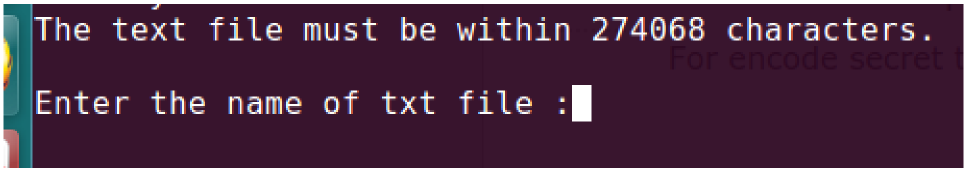
Then user choose a action to do.



```
Enter 1 for encode secret text into the image.
Enter 2 for encode secret image into the image.
Enter 3 for decode the secret message.

Enter your choise : 
```

Figure 13: user input action.

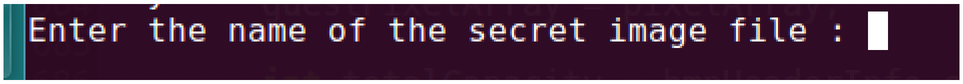
For encode secret text into the cover image enter the text file name.



```
The text file must be within 274068 characters.

Enter the name of txt file :
```

Figure 14: secret txt file name.

For encode secret image into the cover image enter the bmp file name.

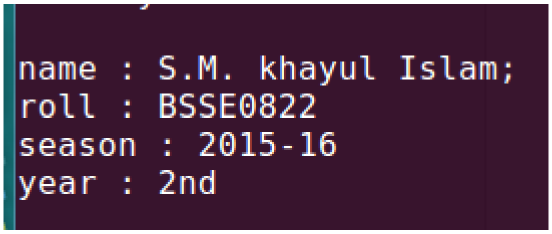


```
Enter the name of the secret image file : 
```

Figure 15: secret image file name.

For decode the secret message if the secret message is text format then show the message in the terminal. If the secret message is image then output the secret image as a bmp file in the same folder otherwise print a message nothing to decoded.

Secret text output.



```
name : S.M. khayul Islam;
roll : BSSE0822
season : 2015-16
year : 2nd
```

Figure 16: secret text output.

Secret image output.

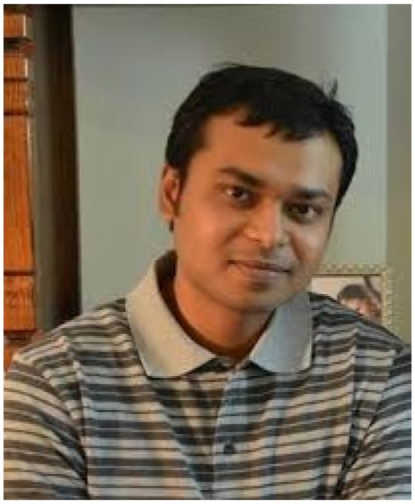
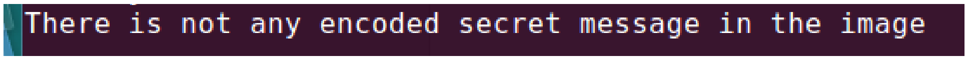


Figure 17: secret image output.

Empty stego image.



```
There is not any encoded secret message in the image
```

Figure 18: empty message.

12. Conclusion

In this project I learnt about binary file specifically BMP image file how to read byte by byte and write also. Image pixel value and the number of byte in a colour pixel. How to change the pixel value without changing the image. Learn the LSB method to implement steganograpy. In this project I handle huge size of pixel array through pointer for this reason my concept about pointer is clear and the implementation of pointer give me confidence about it. Learn how to reuse the function of the program and divide the work as small as possible so that it is easy to solve the problem.

References

1. http://searchsecurity.techtarget.com/definition/steganography.

2. http://repository.root-me.org/St%C3%A9ganographie/EN%20-%20Image %20Steganography%20Overview.pdf

3.https://pdfs.semanticscholar.org/55e0/0d76749f731012d01c887f139b5e00cc5cac. pdf.

4.http://www.infosecwriters.com/text_resources/pdf/steganographyDTEC6823.pdf

5.https://pdfs.semanticscholar.org/55e0/0d76749f731012d01c887f139b5e00cc5cac. pdf.
6. http://paulbourke.net/dataformats/bmp/