IMPLEMENTATION OF IMAGE STEGANOGRAPHY FOR SECURE DATA COMMUNICATION

Afroza Sultana[1], Ide Afroze Nayeema[1] and Ditee Yasmeen[1]
[1] Department of Computer Science and Engineering, Institute of Science and Technology, National University, Dhaka, Bangladesh
Email: [1]afroza640xl@outlook.com, [1]idenayeema@outlook.com, [1]ditee.yasmeen@yahoo.com

Abstract

The urge to communicate without get noticed by any intruder introduced a technique for information security, which is known as steganography. Image steganography is one of the common types of it. Image steganography works with hiding the message into a cover image. This paper concentrates on hiding text-based information into the cover image using the Least Significant Bit (LSB) algorithm.

Keywords: Cryptography, Steganography, Image Steganography, LSB, Embedding, Extracting, MSE, PSNR, Histogram

1. INTRODUCTION

Cryptography is the most widely known and used techniques for information security. Cryptography usually scrambles message to obscure its meaning [1]. Unfortunately, it is sometimes not enough, it may also be necessary to keep the existence of the message secret. And for keeping the existence of the message a secret, steganography is a useful technique. Steganographic techniques have been used for ages and they date back to ancient Greece in 440BC [2]. Mention of steganography also appeared in early publications such as "Stegangraphia," written by Johannes Trithemum before 1606 [3].

2. LITERATURE REVIEW OF IMAGE STEGANOGRAPHY

Today steganography is mostly used on computers with digital data being the carriers and networks being the high-speed delivery channels. Now-a-days, for modern way of communication there are various types of steganography which is as follows:

Text Image Audio/Video TCP/IP packets

Images are the most popular cover objects used for steganography [4]. In the domain of digital images many different image file formats exist, most of them for specific applications. Image steganography is essentially hiding the message within the cover of an image, which is also known as embedding. And getting the message from the image is known as extracting.

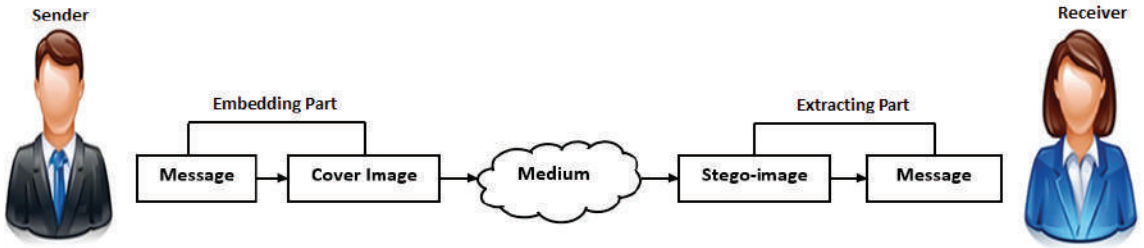


Figure 1: Process of image Steganography

Figure 1 is an example of general process of image steganography. In the sender part, the message is embedded into an image and it is extracted in the receiver part. Image steganography has two part, one is the transform domain and the other is image domain. Using the least significant bit for image steganography is under image domain part.

3. LSB ALGORITHM AND EXISTING SYSTEMS

Least significant bit is the two rightmost bit of an image. In LSB technique, the least significant bit of the image are converted into their binary values. And the secret message is also converted into a binary representation according to the ASCII values of each character of the message. In color image there are three pixel elements, RED, GREEN and BLUE [5]. In a 24-bits color image each of this three color components, a bit can be used, since they are each represented by a byte. So basically 3 bits of each pixel can store a character. An 800 × 600-pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data.

Table 1 is an example of how LSB technique is used to conceal a data into a 24-bits image, the pixel values are chosen randomly. For hiding the number 200, which has the binary representation 11001000, the color components are shown in the table.

Table 1: Binary Representation of RGB Image Components

Pixel	R	G	B
Pixel 1	00101101	00011100	11011100
Pixel 2	10100110	11000100	00001100
Pixel 3	11010010	10101101	01100011

After embedding the number into the pixel values are changed as shown in Table 2.

Table 2: The Resulting Grid of the Pixels after Embedding

Pixel	R	G	B
Pixel 1	00101101	00011101	11011100
Pixel 2	10100110	11000101	00001100
Pixel 3	11010010	10101100	01100011

From the above example, it is clear that the pixel values are changed according to the secret message. However, in this technique when a large amount of data is to be hidden, sometimes two or three of the least significant bit is changed, which creates a great amount of distortion. The larger the amount of distortion the more corrupted the image looks. Figure 2 represents how images are distorted in the existing systems.

A. Real Image B. Stego-image

Figure 2: Distortion in Systems using Original LSB Algorithm

For example, if a pixel of the cover image with the RGB (Red-Green-Blue code) color A8A8A8 # is used, binary 10101000-10101000-10101000, and 1 bit with value 1 is set on each LSB bit of each color component, to hide the message 111, and the result would be 10101001-10101001-10101001:

Table 3 results obtained hiding the message 111 in the pixel 10101000-10101000-10101000 with the LSB method. As shown in the table, when three color pixels are modified it creates a major distortion which can be discovered in statistical analysis.

Table 3: Hiding Message Using LSB Algorithm

	Hexadecimal	Decimal	Red	Green	Blue
Original pixel	A8A8A8	11053224	168	168	168
Modified pixel	A9A9A9	11119017	169	169	169

So using standard LSB algorithm can create huge distortion thus attracting intruders. Thus, using standard LSB algorithm cannot ensure proper security of a system.

4. PROPOSED SYSTEM DESIGN

The proposed system is about overcoming the limitation of original LSB algorithm and after applying the proposed algorithm, producing a stego image with less distortion and high quality. The proposed system is also divided into two parts; embedding and extracting. The working procedure and algorithm for these two parts are given respectively:

4.1 Data Embedding Working Procedure

The embedding of data means, hiding any data into a media. It concerns on hiding a plain text into an image.

1. An image of JPEG, PNG, BMP or Greyscale format is chosen as cover image.
2. The secret message in text format is chosen and it can contain special characters such as * & % # @ ! `, : ? , / etc.
3. Each character of the secret message is converted into ASCII code.
4. Length of the message is calculated and padded with zero to make it 8 characters long if needed and which will be added to the header of the message.
5. Each ASCII code is converted to its 8-bit binary equivalent.
6. To bring the pixel values of the image in 0-255 range it's converted into unsigned 8-bit integer.
7. Cover image is separated in RGB plane and the 1st bit of secret message to last bit of first pixel of red plane, and second bit to the last bit of second pixel of red plane.
8. Now 3rd bit of secret message to last bit of first pixel of green plane, 4th bit to the last bit of second and 5th bit to the last bit of third pixel of green plane.
9. Finally 6th, 7th and 8th bit of secret message to last bit of first pixel, second pixel and third pixel of green plane respectively. So 8 bits of first character of secret text message get embedded in 8 pixels of cover image (2 red, 3 green and 3 blue). Each time when a bit is embedded to a pixel of a plane increase its position by 1 so as to go on next pixel of that plane. This process continues till all bits get embedded in cover image.
10. The insertion occurs in a selected color manner. That means while inserting a bit of message only that color is changed which will not have a huge difference between the previous value of it before insertion and the value after insertion.
11. After all the message is hidden a stego-image is produced.

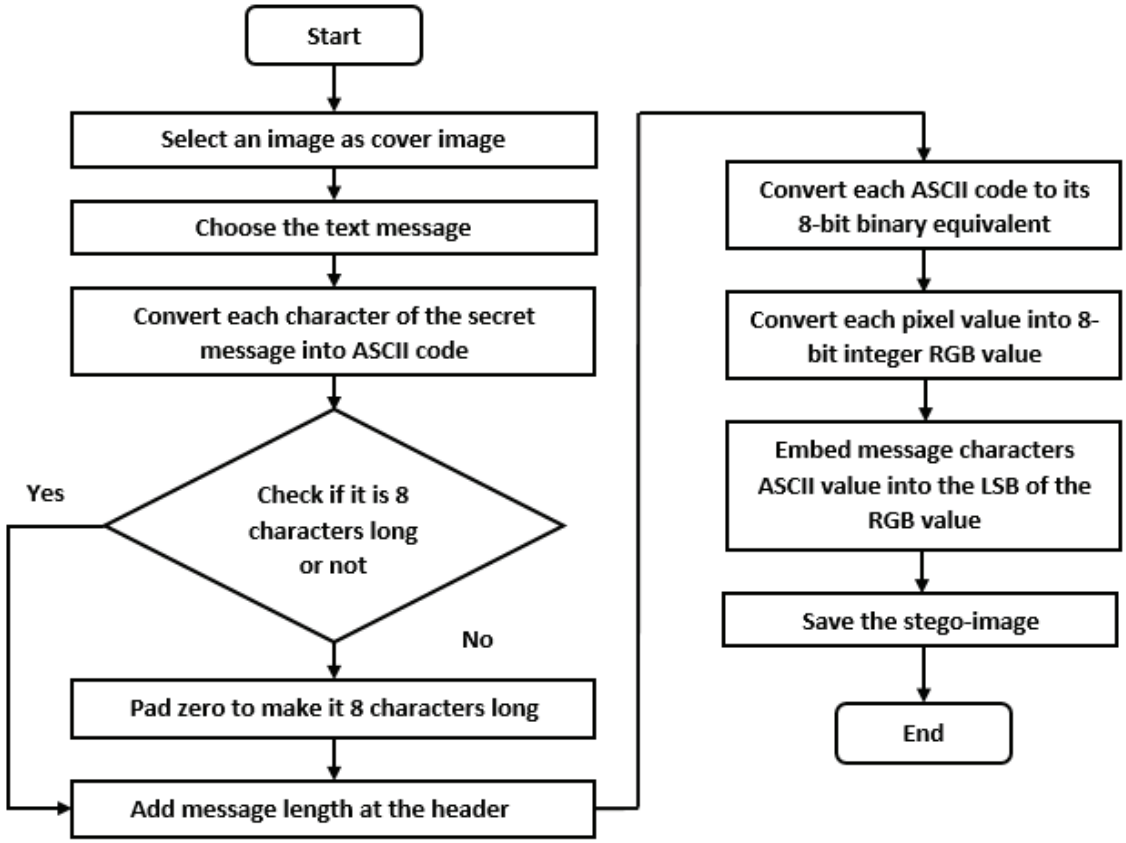


Figure 3: Flow Chart of Embedding Algorithm

4.2 Data Extraction Working Procedure

Data extraction is the act or process of retrieving data out of data sources for further data processing or data storage. The message is extracted in binary form which is converted into corresponding ASCII values to decipher the characters.

1. The stego-image is selected.
2. Stego-image is separated in RGB plane to take mode 2 of first and 2nd pixel of red plane so as to get first 2 bit of first character of secret message.
3. Now mode 2 of 1st, 2nd and 3rd pixel of green plane is performed to get 3rd, 4thand 5th bit of first character respectively.
4. Finally mode 2 of 1st, 2nd and 3rd pixel of red plane is taken to get 6th, 7th and 8th bit of first character of secret message respectively.
5. Each time mode 2 of pixel is taken its position is increased by 1 so as to go to next pixel.
6. These steps are run up to 8 times so as to get header which contain the information of secret message length.
7. Now the same process is repeated again i.e. mode 2 of 1st, 2nd and 3rd pixel of green plane is taken to get 3rd, 4th and 5th bit of first character respectively. Finally, mode 2 of 1st, 2nd and 3rd pixel of red plane to get 6th, 7th and 8th bit of first character of secret message respectively.
8. Each time mode 2 of pixel is taken increase its position value by 1 so as to go to next pixel. This run up to message length.
9. Binary code is converted into decimal which represent ASCII value of secret message.
10. After converting it in character secret message is found that can be saved as text file.

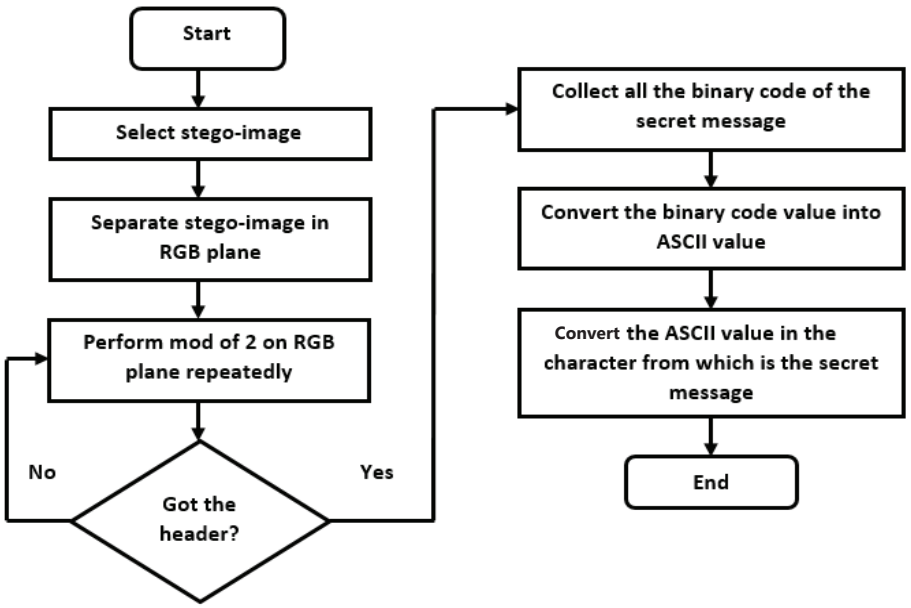


Figure 4: Flow Chart of Extraction Algorithm

After applying the proposed algorithm the changes in the three color pixel are presented in Table 4:

Table 4: Hiding Data in One Selected Color

	Hexadecimal	Decimal	Red	Green	Blue
Original Pixel	9E8C7A	10390650	158	140	122
Modified Pixel	9F8D7A	10456186	159	140	122

Here only the red color is changed and the blue and green plane remains unchanged. The binary representation of the planes before embedding is; 10011110-10001100-01111010 and after embedding is; 10011111-10001100-01111010. Figure 5 shows the assumed stego image after applying the proposed algorithm.

A. Real image B. Stego-image

Figure 5: Assumed Stego-Image after Applying the Proposed Algorithm

5. TESTING

The main concern of steganography is that no one can detect with eye that the image is modified. In addition, to ensure that, the stego image has to be as mostly close looking as the cover image. So to test if the image has high quality there are two types of test; the PSNR value comparison and histogram comparison.

5.1 PSNR Calculation

PSNR is a standard measurement used in steganography technique in order to test the quality of the stego images [6]. The higher the value of PSNR, the more quality the stego image will assure. The PSNR is calculated as follows:

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right)$$

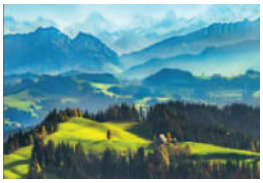
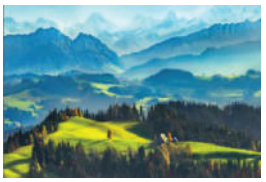
Where, MSE= $\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(C(x,y)-S(x,y)\right)^2$

Here MAX is the maximum possible pixel value of the images. For example, if the pixels are represented using 8 bits per sample, then the MAX value is 255[6]. If the stego image has a higher PSNR value, then the stego image has more quality image.

5.1.1 PSNR Comparison between Test Images

For images of PNG, JPEG, BMP format and GRAYSCALE, PSNR values are compared between cover images and stego images, which is shown in Table 5 while testing the images had the ratio of 900x592.

Table 5: PSNR Values Comparison

Format	Cover Image	Stego Image	PSNR
JPEG			65.11
PNG			89.26
BMP			88.80
GRAYSCALE			64.78

Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, provided the bit depth is 8 bits, where higher is better. Therefore, the images meet the quality range for acceptable PSNR range as the PSNR values of every image are above 50db.

5.2 Histogram Generation

For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges that span the image's color space, the set of all possible colors [7]. Here the X-axis represents the tonal scale (black at the left and white at the right), and Y-axis represents the number of pixels in an image in a certain area of the tonal scale.

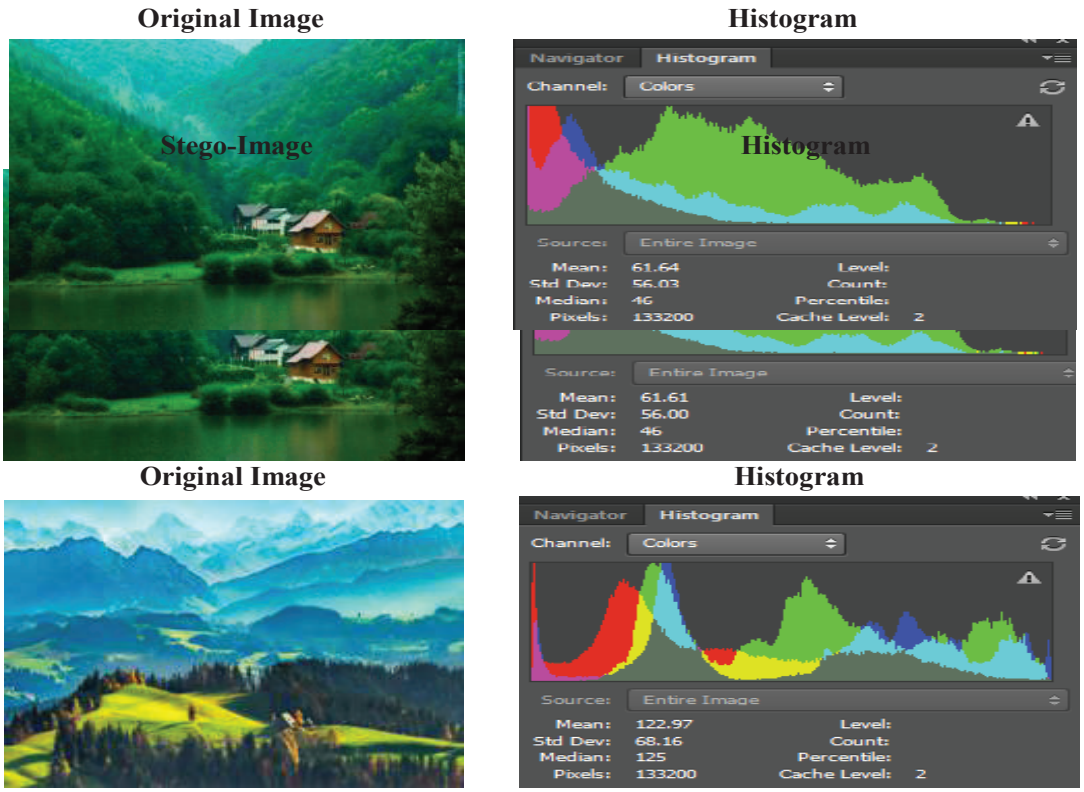
Case One: The first image is taken for comparison in JPEG format, which is shown in Figure 6.

A. Original image

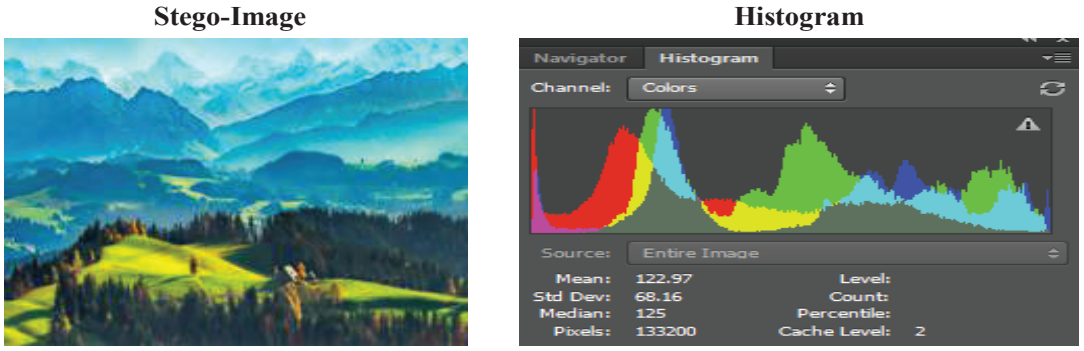
B. Stego-Image

Figure 6: Histogram Comparison for JPEG Format

Case Two: Figure 7 representing a PNG format of cover image and stego image.



A. Original Image

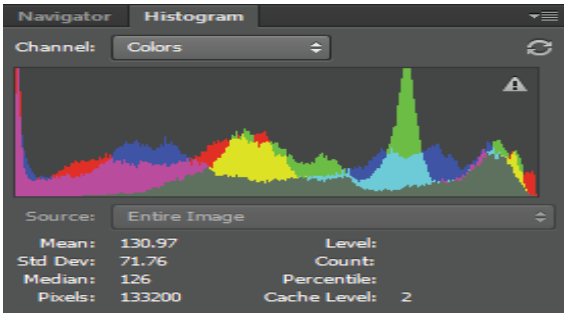


B. Stego-Image

Figure 7: Histogram Comparison for PNG Format

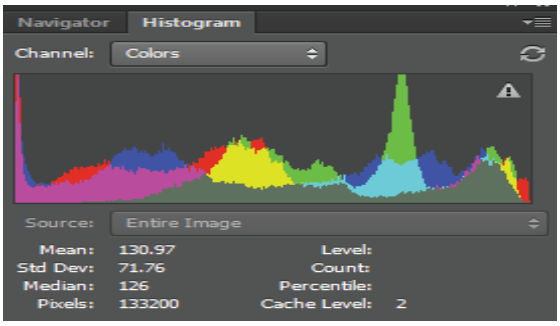
Case Three: BMP format of cover image and stego image are figured out in the following:-

Original Image **Histogram**



A. Original image

Stego-Image **Histogram**



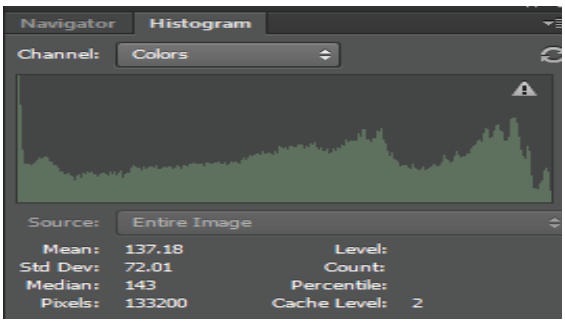
B. Stego-Image

Figure 8: Histogram Comparison for BMP Format

Case Four:

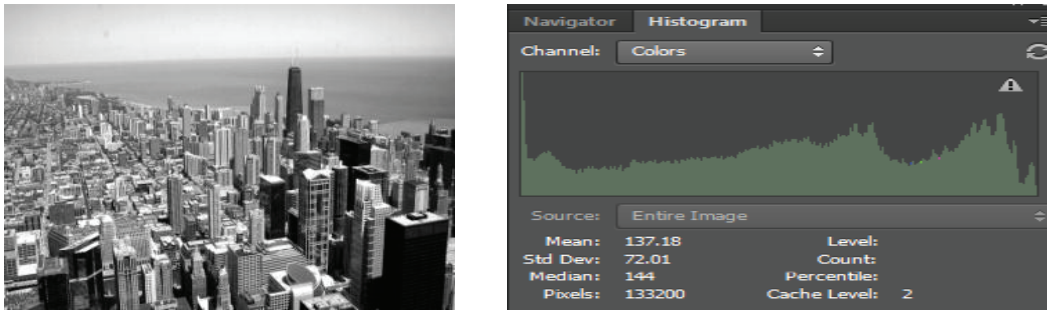
In Figure 9, a cover and stego image of Grayscale format is captured for histogram comparison.

Original Image **Histogram**



A. Original Image

Stego-Image **Histogram**



B. Stego-Image

Figure 9: Histogram Comparison for Grayscale Format

After comparing histogram of cover images with the stego images, it is quite clear, that histogram of stego image is almost similar to cover image, as there is change of only last bit of pixels. Therefore, this method is capable of producing a secret-embedded image that is indistinguishable from the original image by the human eye and cannot be detect by histogram analysis method.

6. CONCLUSION

People have desired to keep certain sensitive communications secret for thousands of years. In our new age of digital media and internet communications, this need often seems even more pressing. This paper presents general information about steganography, the art of data hiding. The paper provides an overview of steganography, general forms of steganography and recent developments in the field. The information presented in this paper is also applied to a program developed by the authors, and some sample runs of the program are presented. In this project focuses on the use of steganography within digital images (JPEG, BMP, PNG and Grayscale) using a modified LSB Substitution. This project concentrates on implementation of pure steganography with an improved LSB algorithm, but every day technology is enhancing and intruders constantly trying to break through the security barrier. To create a more secure method, this project can be improved by using keyed steganography with public key and privet key encryption. Combining steganography and cryptography can create a more secure method for secure data communication.

REFERENCES

[1] T Morkel, J.H.P Eloff, M.S Olivier, "An Overview of Image Steganography". Proceedings of the Fifth Annual Information Security South Africa Conference (Issa2005), 2005.

[2] Cox, I. (2008). Digital watermarking and steganography. Amsterdam: Morgan Kaufmann Publishers.

[3] Dharwadkar, N. And Amberker, B. (2010). Steganographic Scheme for Gray-Level Image Using Pixel Neighborhood and Lsb Substitution. International Journal of Image and Graphics, 10(04), pp.589-607.

[4] Katzenbeisser, S. and Petitcolas, F. (2000). Information hiding techniques for steganography and digital watermarking. Boston: Artech House.

[5] Jamil, T. (1999). Steganography: the art of hiding information in plain sight. IEEE Potentials, 18(1), pp.10-12.

[6] Wang, H. and Wang, S. (2004). Cyber warfare. Communications of the ACM, 47(10), pp.76-82.

[7] Marvel, L., Boncelet, C. and Retter, C. (1999). Spread spectrum image steganography. IEEE Transactions on Image Processing, 8(8), pp.1075-1083.