Implementation of LSB Steganography and Its Evaluation for Various Bits

Deshpande Neeta, Kamalapur Snehal
Computer Science Dept
K.K.Wagh Institute of Engineering
Education & Research, Nashik
India
deshpande_neeta@yahoo.com, kamalapur_snehal@yahoo.com

Daisy Jacobs
School of Information Technology
University of Pretoria, Pretoria 002
South Africa
daisy.jacobs@up.ac.za

Abstract

Steganography is the art of hiding information in information is gaining momentum as it scores over cryptography because it enables to embedd the secret message to cover images. Steganographic techniques offer more promise in digital image processing. The Least Significant Bit embedding technique suggests that data can be hidden in the least significant bits of the cover image and the human eye would be unable to notice the hidden image in the cover file. This technique can be used for hiding images in 24-bit, 8-bit or gray scale format. We emphasize strongly on image Steganography providing a strong focus on the LSB techniques in image Steganography. This paper explains the LSB embedding technique and presents the evaluation results for 2,4,6 Least significant bits for a .png file and a .bmp file.

1. Introduction

Digital content is now posing formidable challenges to content developers, aggregators, distributors and users. The destruction, extraction or modification of the embedded message is required to develop more robust systems so that the digital content processing and organization become ease.

The shift from cryptography to stegnography is due to that concealing the image existence as stegno-images enable to embedd the secret message to cover images. Steganography conceptually implies that the message to be transmitted is not visible to the informal eye. Steganography has been used for thousands of years to transmit data without being intercepted by unwanted viewers. It is an art of hiding information inside information. The main objective of Steganography is mainly concerned with the protection of contents of the hidden information.

Images are ideal for information hiding[1,2] because of the large amount of redundant space is created in the storing of images. Steganography consists of methods of transmitting secret messages. These secret messages are transferred through unknown cover carriers in such a manner that the very existence of the embedded messages is undetectable. Carriers include images; audio, video, text, or any other digitally represented code or transmission. The hidden message may be plaintext, cipher text, or anything that can be represented as a bit stream.

2. Image Steganography

Image compression techniques are extensively used in steganography. Among the two types of image compressions, lossy compression and loss less compression, loss less compression formats offer more promises. Typical examples of loss less compression formats are CompuServe's GIF (Graphics Interchange Format) and Microsoft's BMP (Bitmap) [3].

We have used an 8-bit image size for implementation of our steganography. In using an 8-bit image as the cover-image, many steganography experts recommend using images featuring 256 shades of gray as the palette. Grey-scale images are preferred because the shades change very gradually between palette entries. This increases the image's ability to hide information. Once a suitable cover image has been selected, an image encoding technique needs to be chosen.

Improvement in stegnographic techniques make it possible to apply the Detecting LSB Steganography in Color and Gray- Scale Images which were confined to gray scale images in the initial stages The difficulty in colour images control is solved later on in many techniques such as the analysis of the variation of the gradient energy, the secret message embedded in the target image is detected in both gray and colour images, and the length of the embedded message is estimated. [5,6]

3. Hiding Methods in Image Steganography

Image Steganography has been widely studied by researchers. There are a variety of methods using which information can be hidden in images.

Least Significant Bit Replacement Technique*:* In image steganography almost all data hiding techniques try to alter insignificant information in the cover image. Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image. For instance, a simple scheme proposed, is to place the embedding data at the least significant bit (LSB) of each pixel in the cover image[7,8,9] . The altered image is called stego-image. Altering LSB doesn't change the quality of image to human perception but this scheme is sensitive a variety of image processing attacks like compression, cropping etc. We will be emphasizing more on this technique for the various image formats.

Moderate Significant Bit Replacement Technique: The moderate significant bits of each pixel in the cover image can be used to embed the secret message. This method improves sensitivity to modification, but it degrades the quality of stego-image.

Experiments have shown that the length of hidden messages embedded in the least significant bits of signal samples can be estimated with relatively high precision. A comprehensive survey of steganographic methods was presented in

4. The LSB Technique

The least significant bit i.e. the eighth bit inside an image is changed to a bit of the secret message. When using a 24-bit image, one can store 3 bits in each pixel by changing a bit of each of the red, green and blue color components, since they are each represented by a byte. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data For example a grid for 3 pixels of a 24-bit image can be as follows:

(01010101 01011100 11011000)

(10110110 11111100 00110100)

(11011110 101100101 01101011)

When the number 300, which binary representation is 101101100 is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

(00101101 00011100 11011101)

(10100111 11000100 00001101)

(11010011**1**0101100 01100010)

Here the number 300 was embedded into the first 8 bytes of the grid, only the 5 bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel results in small changes in the intensity of the colors. The human eye cannot perceive these changes - thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the LSB without noticing the difference.

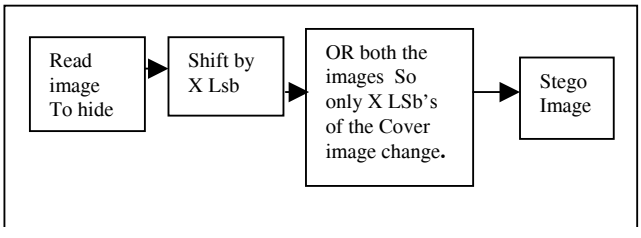
5. Design Details

This section focuses on algorithms of LSB Steganography and Steganalysis

5.1 Algorithm for Hiding (Steganography)

1. Read the original image and the image which is to be hidden in the original image
2. Shift the image to hide in the cover image by X bits
3. And the original image or cover image with 240 which is 11110000 So four MSb's set to 0.Because of this only four LSB's considered further.
4. The shifted hidden image and the result of step 3 are bitored. This makes changes only in the X Lsb bits so that the image is hidden in the original image

In MATLAB we convert it to uint8 format. This image can be called as the stego image



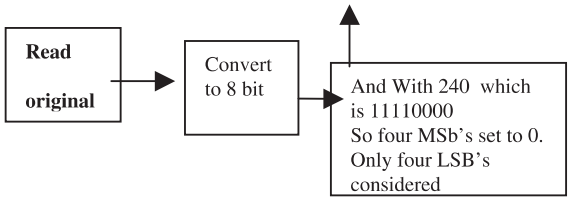


Figure 1 Block Diagram for implemented Logic of LSB embedding

5.2 Algorithm for Steganalysis

1. The stego image is bit shifted by 4 bits since it was shifted by 4 bits to insert it into the original image.
2. This image is then ANDED with 255 i.e. 11111111, which gives the original image. It is ANDED with 255 because initially all the LSB's were made 0.Now it is recovered back.
3. To get it to Uint8 format we, convert it back to uint8 which is the extracted image.

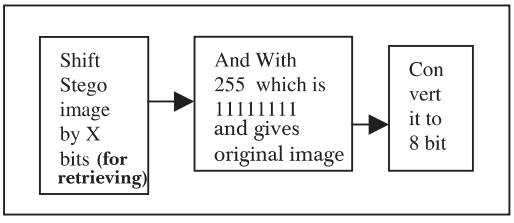


Figure 2 Block Diagram for Steganalysis

6. Image Analysis

6.1 LSB in BMP

Since BMP is not widely used the suspicion might arise if it is transmitted with an LSB stego. When images are used as the carrier in Steganography they are generally manipulated by changing one or more of the bits of the byte or bytes that make up the pixels of an image. The message can be stored in the LSB of one color of the RGB value or in the parity bit of the entire RGB value .A BMP is capable of hiding quite a large message. LSB in BMP is most suitable for applications where the focus is on the amount of information to be transmitted and not on the secrecy of that information. If more number of bits is altered it may result in a larger possibility that the altered bits can be seen with the human eye. But with the LSB the main objective of Steganography to pass a message to a receiver without an intruder even knowing that a message is being passed is being achieved.

Stego –1 bit LSB : This method changes only single LSB in the image. Changing the LSB will only change the integer value of the byte by one. This small change is not noticeable. The visual appearance of a color and hence the image itself is not changed. A proportionately greater change in the visual appearance of a color could be achieved by changing a more significant bit.. The LSB method for 1 bit will involve changing the LSB of one of the colors making up the RGB value of the pixel. This should have very little effect on the appearance of the image. The only restriction here is the size of the image. The above process will produce new colors for the palette. The creation of a new color for every existing color in the palette is possible if the image has a palette size of 128 pixels. If the palette is ordered by luminance that there will be pairs of very similar colors. If a 128 palette image is used it may not result in too much distortion to the original image.

Stego –2 bit LSB: The advantage of this method is that twice as much information can be stored here than in the previous method. By this method two LSBs of one of the colors in the RGB value of the pixels will be used to store message bits in the image. The starting image would still have to have a palette containing 68 colors. Since two bits are involved for storing the stego data, the palette must have a maximum of 64 colors. This will result in a combination of 192 new colors, i.e. three new colors for each existing color. Fewer colors will be available to represent the starting image and hence it will be more degraded than the image used in the method Stego One Bit. This method could instead have used the LSB of two colors in the RGB value, which would have resulted in the same amount of storage space.

Stego –3 bitsLSB: The data hiding capacity is 3 times the storage capacity of Stego One Bit .If the 128 color palette is used the image will be more distorted. In this method three LSBs of one of the colours in the RGB value of the pixels will be used to store message bits. This will result in a combination of 224 new colors, i.e. three new colors for each existing color. This will involve using a palette with a maximum of only 32 colors..

Stego –4 bitsLSB: The data hiding capacity is 4 times the storage capacity of Stego One Bit .If the 128 color palette is used the image will be more distorted. In this method four LSBs of one of the colours in the RGB value of the pixels will be used to store message bits. It should involve a palette of 16 colors. This will result in a combination of 240 new colors, i.e. three new colors for each existing color. This will involve using a palette with a maximum of only 32 colors. Changing 4 LSB bits of each of red, green and blue pixels my result in some amount of texture changes.

6.2 LSB in PNG

Since PNG is widely used the suspicion might not arise if it is transmitted with an LSB stego. When images are used as the carrier in Steganography they are generally manipulated by changing one or more of the bits of the

byte or bytes that make up the pixels of an image. The message can be stored in the LSB of one color of the RGB value or in the parity bit of the entire RGB value .A PNG is capable of hiding quite a large message. LSB in PNG is most suitable for applications where the focus is on the amount of information to be transmitted and not on the secrecy of that information. If more number of bits is altered it may result in a larger possibility that the altered bits can be seen with the human eye. But with the LSB the main objective of steganography to pass a message to a receiver without an intruder even knowing that a message is being passed is being achieved

6.3 LSB in GIF

Since GIF images only have a bit depth of 8, the amount of information that can be hidden is less than with BMP. Embedding information in GIF images using LSB results in almost the same results as those of using LSB with BMP. LSB in GIF is a very efficient algorithm to use when embedding a reasonable amount of data in a grayscale image. GIF images are indexed images where the colors used in the image are stored in a palette. It is sometimes referred to as a color lookup table. Each pixel is represented as a single byte and the pixel data is an index to the color palette. The colors of the palette are typically ordered from the most used color to the least used colors to reduce lookup time. Some extra care is to be taken if the GIF images are to be used for Steganography. This is because of the problem with the palette approach. If the LSB of a GIF image is changed using the palette approach, it may result in a completely different color. This is because the index to the color palette is changed. The change in the resulting image is noticeable if the adjacent palette entries are not similar. but the change is not noticeable if the adjacent palette entries are similar. Most applications that use LSB methods on GIF images have low security because it is possible to detect even moderate change in the image
Solutions to these problems could be
1.Sort the palette so that the color difference between consecutive colors is minimized
2.Add new colors, which are visually similar to the existing colors in the palette.
3.Use Gray scale images. In a 8 bit Gray scale GIF image, there are 256 shades of gray. This results in gradual changes in the colors and it is hard to detect.

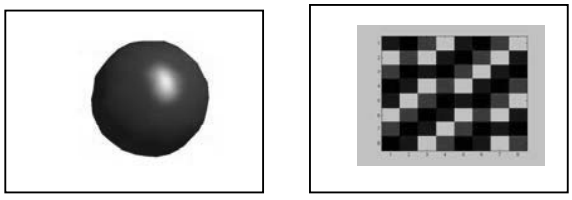
7. Experimented Results

Following experimental results highlights on 2bit, 4 bit, 6 bit LSB Steganography

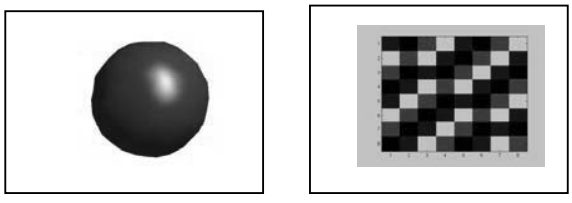
7.1 Results for .png image

2 bit Stego:

Cover Image **Image to hide**

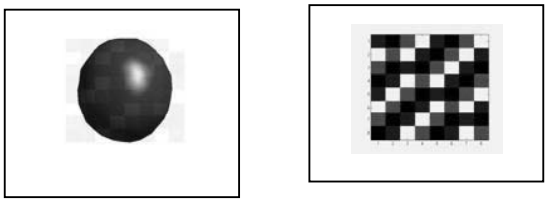


Stego image **Recovered image**



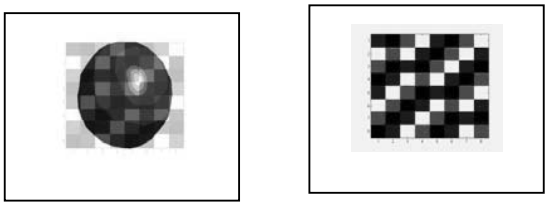
4 bit stego:

Stego image **Recovered image**



6 bit stego:

Stego image **Recovered image**

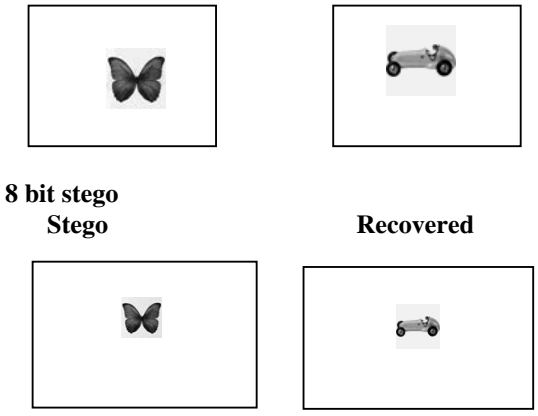


7.2 Results for .bmp file

4 bit Stego

Message **Cover image**

Stego **Recovered**



8 bit stego

Stego **Recovered**

8. Evaluation of Image Quality

For comparing stego image with cover results requires a measure of image quality, commonly used measures are Mean-Squared Error, Peak Signal-to-Noise Ratio [3] and histogram.

8.1 Mean-Squared Error

The mean-squared error (MSE) between two images $I_1(m,n)$ and $I_2(m,n)$ is:

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N}$$

M and *N* are the number of rows and columns in the input images, respectively. Mean-squared error depends strongly on the image intensity scaling. A mean-squared error of 100.0 for an 8-bit image (with pixel values in the range 0-255) looks dreadful; but a MSE of 100.0 for a 10-bit image (pixel values in [0,1023]) is barely noticeable.

8.2 Peak Signal-to-Noise Ratio

Peak Signal-to-Noise Ratio (PSNR) avoids this problem by scaling the MSE according to the image range:

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

PSNR is measured in decibels (dB). PSNR is a good measure for comparing restoration results for the same image, but between-image comparisons of PSNR are meaningless. MSE and PSNR values for each file format is shown in table 1

Table 1: Image quality metrics for bmp file

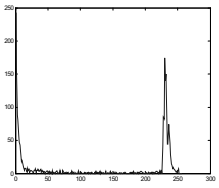
	Cover	Stego image	Cover-Stego image
MSE	224.948	244.162	69.826
PSNR	24.6100	24.2540	29.690

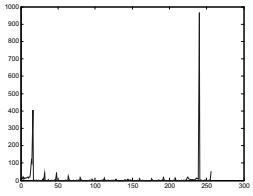
8.3 Histogram

The histogram functions count the number of elements within a range and display each range as a rectangular bin. The height (or length when using rose) of the bins represents the number of values that fall within each range. An image histogram is a chart that shows the distribution of intensities in an indexed or intensity image.Since color images are considered for experimentation, histograms for all the three color components are calculated.

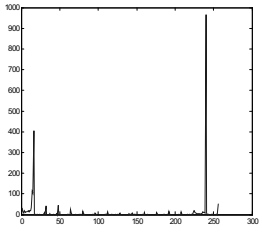
Evaluated Histograms for RGB components of the cover and Stego images

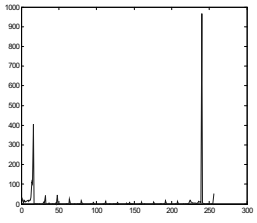
Rhist For Cover image **Rhist For stego image**



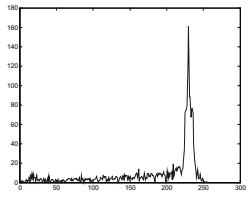


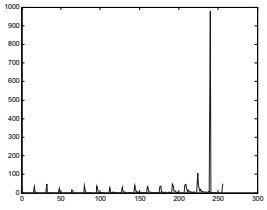
GHist for cover **Ghist for Stego**





Bhist for cover **BHist for cover**





9. Evaluation of different techniques

There are many steganographic algorithms available. One should select the best available algorithm for the given application. Following characteristics are to be evaluated while selecting a particular file format for Steganography. Steganography says that the secret message is to be hidden and it should result in an distortion less image. The distortion must not be visible to the human eye. The amount of data embedded in the image also plays an important role. The algorithm decides how much amount of data could be embedded in the image resulting in a distortion less image. Steganalysis is the technique of detecting the hidden information in the image. The algorithm for Steganography must be such that the steganalysis algorithms should fail. i.e the Steganography algorithms must not be prune to attacks on steganalysis. During communication the intruder could check the original image to remove the hidden information.. He/she may manipulate the image. This manipulation may include cropping or rotation etc of the images. The manipulations done may cause the image distortion. Steganographic algorithms chosen must be such that it overcomes such manipulation and the steganographic data reaches the destination in the required format.

Table 2: Comparison of LSB technique for various file formats

	Lsb In BMP	LSB in GIF	LSB in PNG
Percentage Distortion less resultant image	High	Medium	High
Amount of embedded data	High	Medium	Medium
Steganalysis detection	Low	Low	Low
Image manipulation	Low	Low	low

10. Conclusion

Since BMP uses lossless compression, LSB makes use of BMP image. To be able to hide a secret message inside a BMP file, one would require a very large cover image. BMP images of 800 × 600 pixels found to have less web applications. Moreover such uses are not accepted as valid. For this reason, LSB Steganography has also been developed for use with other image file formats. Although only some of the main image steganographic techniques were discussed in this paper, one can see that there exists a large selection of approaches to hiding information in images. All the major image file formats have different methods of hiding messages, with different strong and weak points respectively. LSB in GIF images has the potential of hiding a large message, but only when the most suitable cover image has been chosen.

11. References

[1] Pfitzmann Birgit. Information Hiding Terminology. First International Workshop, Cambridge, UK, Proceedings, Computer Science 1174. pp. 347-350, May -June

[2] Westfield Andreas and Andreas Pfitzmann, Attacks on Steganographic Systems. Third International Workshop, IH'99 Dresden Germany, October Proceedings, Computer Science 1768. pp. 61- 76, 1999

[3] Moerland, T., "Steganography and Steganalysis", *Leiden Institute of Advanced Computing Science*, Silman, J., "Steganography and Steganalysis: An Overview", *SANS Institute*, 2001 Jamil, T., "Steganography: The art of hiding information is plain sight", *IEEE Potentials*, 18:01, 1999

[4] Johnson, N.F. & Jajodia, S., "Exploring Steganography: Seeing the Unseen", *Computer Journal*, February 1998

[5] Li Zhi,Sui Ai Fen., "Detection of Random LSB Image Steganography" The IEEE 2003 International Symposium on Persona1,Indoor and Mobile Radio Communication Proceedings, 2004.

[6] Jessica Fridrich, Miroslav Goljan, and Rui Du., "Detecting LSB Steganography in Color and Gray- Scale Images", - IEEE Multimedia.

[7]F.Collin,\Encryptpic," http://www.winsite.com/bin/Info?500000033023.

[8] G. Pulcini, \Stegotif," http://www.geocities.com/SiliconValley/9210/gfree.html.

[9] T. Sharp, \Hide 2.1, 2001," http://www.sharpthoughts.org