Increased high capacity Random Bit Plane Algorithm for Image Steganography

W.Regis Anne Ratheesh, MCA, M.Phil, M.E, J.Judith shiny*,S.Nandhini*

Assistant Professor, Department of MCA, Sri Krishna College of Engineering, Tamil Nadu.
E-mail: anneratheesh@gmail.com
*Research Scholars, Sri Krishna College of Engineering, Tamil Nadu.
E-mail: joyjason77@gmail.com

Abstract
Steganography in the digital world is a process by which one medium is embedded in another medium. The medium that is embedded is known as source medium, which is the secret information. The medium in which the source medium is embedded is known as cover medium. After this embedding process the cover medium is send through the network to the receiver. Any hacker who intercepts the message in the network will only see the cover medium and not the source medium. This paper introduces the reader to the principles behind steganography and the Least Significant Bit (LSB) Insertion Algorithm to hide files in BMP or WAV files. The LSB insertion method is probably the most well known steganography technique. It is a common, simple approach for embedding information in a graphical image file. Unfortunately, it is extremely vulnerable to attacks, such as image manipulation. So a modified LSB insertion is proposed. The Increased high capacity Random Bit Plane Algorithm modifies the LSB to increase the capacity of hiding the messages and making it less vulnerable to attacks over the shared channel.

Keywords:
Steganography, Capacity, LSB, Bit Map, Bit Plane, Random.

1. Introduction

Steganography is the art of secret communication. Its purpose is to hide the presence of information, using for example images as covers. Steganography is essential a new method of securing confidential data as well as a technique by which copyrights and ownership can be issued towards digital documents and medium. Steganography is the technique by which data is hidden and transmitted in another medium. The basic principal of steganography lies in the technique by which data is embedded in another medium and transmitting it. If in any case a hacker or any other malicious user intercepts the transmission he/she gets to see only the medium in which the data is hidden and not the data. The hacker doesn't see the embedded data and thus the integrity of the data is intact. Steganography [1] in the digital world is a process by which one medium is embedded in another medium. The medium that is embedded is known as source medium, which is the secret information. The medium in which the source medium is embedded is known as cover medium. In real time application it is always prudent to encrypt the source medium before embedding it into the cover medium. This technique follows the requirement posed by the ``Kerckhoff principle'' in cryptography. This principle states that the security of the system has to be based on the assumption that the enemy has full knowledge of the design and implementation details of the steganographic system. The only missing information for the enemy is a short, easily exchangeable random number sequence, the secret key. Without this secret key, the enemy should not have the chance to even suspect that on an observed communication channel, hidden communication is taking place. The whole process gives a twin layer security to the data. The basic terminologies in steganography are
Steganography: Steganography is the process by which the secret data to be transmitted is embedded in another medium.

Source Medium: Source medium is the data for which security is to be achieved using steganography.
Cover Medium: Cover medium is the medium in which the source medium is embedded. It is also referred to as container.
Stego-Object: Stego-object is the cover medium which has the source medium embedded in it. It can be regarded as the output of the steganography process.
Steganalysis: Steganalysis is analogous to cryptanalysis. It is the technique by which a medium is analyzed for any activity of steganography and if detected steganalysis tries to hamper or retrieve the source medium (data) from the cover medium.

2. Applications of Steganography

There are many applications for digital steganography of images, including copyright protection, feature tagging, and secret communications.

A. Copyright Protection*:* A secret copyright notice or watermark can be embedded inside an image to identify it as intellectual property. This is the watermarking scenario where the message is the watermark. The "watermark" can be a relatively complicated structure. In addition, when an image is sold or distributed an identification of the recipient and time stamp can be embedded to identify potential pirates. A watermark can also serve to detect whether the image has been subsequently modified.

B.Feature Tagging: Captions, annotations, time stamps, and other descriptive elements can be embedded inside an image, such as the names of individuals in a photo or locations in a map. Copying the stego-image also copies all of the embedded features and only parties who possess the decoding stego-key will be able to extract and view the features.

C.Secret Communications*:* In many situations, transmitting a cryptographic message draws unwanted attention. The use of cryptographic technology may be restricted or forbidden by law. However, the use steganography does not advertise covert communication and therefore avoids scrutiny of the sender, message, and recipient.

3. The Steganographic Process

There are two phases in any steganographic system. The first phase is Encoder or Embedding Phase concerns with the embedding of the source medium into the cover medium. The second phase is Decoder or Retrieval Phase concerns with the retrieval of the embedded source medium from the stego-object.

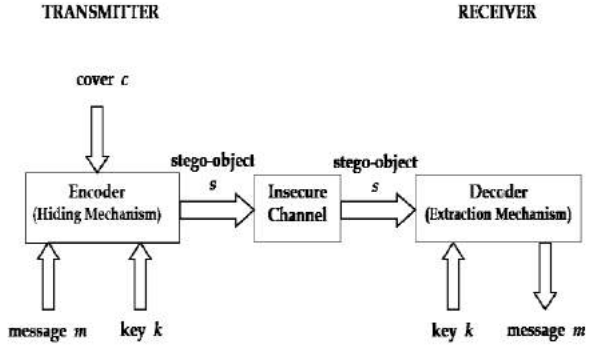

Figure.3.1 The Steganographic process

The embedding process of the steganography application takes data from two data stores i.e. the source medium and the cover medium. It embeds the source medium into the cover medium. The retrieval process of the steganographic application takes the stego-object and retrieves the source medium from the stego-object. To measure the difference between the original cover and stego-image we use the Peak Signal to Noise Ration (PSNR), which expressed as the following equation,

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{MSE}$$

$$MSE = \left(\frac{1}{H \times W}\right)\sum_{i}^{H}\sum_{j}^{W}(x_{ij} - x'_{ij})^2.$$

Where H, W are the size of the cover image xij is the original cover image and x'ij , is the stego-object.

4. Steganography In Image Medium

Steganography [1] in still digital images are used widely for security to data than for authentication purposes. Generally image steganography is about exploiting the limited powers of the human visual system (HVS). Any plain text, cipher-text, other images, or any other file format that can be embedded in a bit stream can be hidden in an image. An *image* is an array of numbers that represent light intensities at various points, or *pixels*. These pixels make up the image's *raster data*. An image size of 640 by 480 pixels, utilizing 256 colors (8 bits per pixel) is fairly common. Such an image would contain around 300 kilobits of data. Digital images are typically stored in either 24-bit or 8-bit per pixel files. 24-bit images are sometimes known as *true color* images. Obviously, a 24-bit image provides more space for hiding information; however, 24-bit images are generally large and not that common. A 24-bit image 1024 pixels wide by 768 pixels high would have a size in excess of 2 megabytes. As such large files would attract attention were they to be transmitted across a network or the Internet, image compression is desirable. Bitmaps are defined as a regular rectangular mesh of cells called pixels, each pixel containing a color value. They are characterized by only two parameters, the number of pixels and the information content or color depth per pixel.

Header-54 bytes	Data –rest of BMP file

The most bitmap is stored, byte after byte, row by row. Files stored by this method are often called raw files. The amount of disk storage required for any bitmap is easy to calculate given the bitmap

	Size in bytes	data type	Name	Comment
Bitmap File Header	2	UINT	BfType	'B' 'M'
	4	DWORD	BfSize	size of file
	2	UINT	bfReserved1	Reserved
	2	UINT	bfReserved2	Reserved
	4	DWORD	BfOffBits	byte location in the file which is first byte of IMAGE
Bitmap Info Header	4	DWORD	BiSize	Size of BITMAPINFOHEADER
	4	LONG	BiWidth	width of bitmap
	4	LONG	BiHeight	height of bitmap
	2	WORD	Biplanes	1
	2	WORD	BiBitCount	1 (mono) or 4 (16 clr) or 8 (256 clr) or 24 (16 Mil)
	4	DWORD	biCompression	RLE Compression
	4	DWORD	BiSizeImage	width x height
	4	LONG	biXPelsPerMeter	

dimensions and color depth. The amount of data which can be hidden within a cover image is directly proportional to both the size of the image, and the specific method used to hide the data. Let us consider a 200x200 pixel image. Using an uncompressed 24bit bitmap image format [9], the data size of the image would be 120000 bytes, or 120 kilobytes. This is calculated by the number of pixels (200 x 200 = 40,000) times the number of bytes per pixel (40,000 x 3 = 20,000). This equates to 960,000 bits which we can possibly modify. The possibilities seem limitless! However, this is not actually the case. Most steganographic techniques use 1 bit per byte to store data, and often times less than that. Also remember that hiding 1 bit of data is not very powerful. We usually want to hide characters, or data, both of which are made of multiple bytes, each byte which is made of 8 bits. This effectively reduces the amount of data we are able to hide to 15,000 bytes or characters (120,000 / 8 = 15,000) [4]. The bitmap brief contents are shown in Table 4.1 and the detail contents of header in Table 4.2.

5. Steganography Techniques

There are few techniques of steganography,

1) Least significant bit (LSB) insertion

2) Masking and filtering techniques

3) Algorithms and transformations

4) Spread spectrum image steganography

Out of these two techniques, lsb insertion [6] is considered to increase the capacity of hiding the messages. In a 24-bit image, each pixel is composed of 24 bits or 3 bytes; one for each of the primary colors red, green and blue. Below is a representation of the pixel. A pixel which is purely red will have its red byte representing 255, green byte representing 0 and blue byte representing 0. So also for purely green and purely blue pixels, where the green byte and the blue byte will represent 255 respectively. Consider a pixel with red, green and blue levels as (113, 64, 64). The binary triplet for these levels will then be (01110001, 01000000, 01000000). The pixel layout will be as shown below. In binary representation, changing the msb of any byte, results in a drastic change in the overall pixel color. For instance, if the msb of the red byte is changed to 1, then the red content of that pixel becomes dominant. Hence, the idea of lsb insertion is to modify the lsb, which causes minimum perceptible color variation. In LSB insertion, one single character is encoded into three pixels. If the pixel values are given as below, when applying LSB techniques to each byte of a 24-bit image, three bits can be encoded into each pixel (as each pixel is represented by three bytes). Any changes in the pixel bits will be indiscernible to the human eye. For example, the letter A can be hidden in three pixels. Assume the original three pixels are represented by the three 24-bit words below:

(00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

The binary value for the letter A is (10000011). Inserting the binary value of A into the three pixels, starting from the top left byte, would result in:
(00100111 11101100 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)
Hence 'A' has been embedded in three contiguous pixels. The process can be repeated for the entire text. Variations of the basic LSB technique have been developed in order to make it more robust. So far, the techniques that have been described are called sequential LSB. That is, the message is laid but across the image data sequentially. One variation [5] would be random LSB, in which the secret data are spread out among the image data in a seemingly random manner. This can be achieved if both the sender and receiver share a secret key. They can use this key to generate pseudorandom numbers, which will identify where, and in what order the hidden message is laid out.The advantage of this method is that it incorporates some cryptography in that diffusion is applied to the secret message. However, it goes beyond just making it difficult for an attacker knows that there is a secret message to figure out the message. It also makes it harder to determine that there was a secret message in the first place. The reason is because the randomness makes the embedded message seem more like noise statistically than in the sequential method.

6. Increased High Capacity Random Bit Plane Algorithm

The increased high capacity random bit plane algorithm considers a 24 bit BMP image where each bit is represented by RGB each of which is 8 bit. The cover-image P is divided into two planes, PM the Most significant bit-plane and P_L the Least significant bit-plane. P_L is chosen to embed the message so that there is no change in the cover image because they are the least significant bits. Then depending on the size of the message(S), the number of pixels is chosen randomly from P_L. P_L is divided into that many bit planes (B).Then the message is embedded by choosing the bit plane randomly.
This increases the capacity of hiding messages [8] nearly by three times that of LSB and there is no change visible to the human eye in the cover medium.

Algorithm:

Input: cover-image P, binary message sequence M, Size of the message S.

Output: stego-object P'.

Step 1:

P' = P.

Step 2:

The image P' is divided into

P'=(PR1,PR2,..,PRn,PG1,PG2,..,PGn,PB1,PB2,..,PBn)

P_M=(PR1,PR2,PR3,PR4,PR5,PG1,PG2PG3,PG4, PG5,PB1,PB2, PB3, PB4,PB5)

P_L=(PR6,PR7,PR8, PG6PG7,PG8, PB6,PB7,PB8)

Step 3:

Use a PseudoRandom Number Generator(PRNG) to randomly select t pixels from P_L.

Let (x_i, y_i) denote the coordinate of the selected pixel. The value of $i = 0,1,\ldots t-1$.

$P_S = t$, t is chosen depending upon size of the message S and rounded into even number and $P_S \leq P_L$.

Step 4:

P_S is divided into Bit Planes Bj of M×N Planes where M=N=3 and $1 \leq j \leq t$.

Step 5:

Let m_i denote the message-bit to be embedded Bj Plane.

Step6:

For all Bj Planes where $1 \leq j \leq t$,

Use a PRNG to generate a random number γ

where the value is $0 \leq \gamma \leq t$.

For that γ, map it to a Bj Plane $(Bj)_{\gamma}$

Embed m_i into $(Bj)_{\gamma}$

End For

Step 7:

Output P'.

This algorithm increases the capacity three times more than LSB. This is also less vulnerable to attack. The characteristics of these methods are the Size of the cover medium doesn't change after embedding the source medium and degradation of the cover medium is not recognizable to the human senses.

7. Experimental Results

The Steganography using random bit plane method implemented and has the main menu in Figure 7.1.Suppose a text file containing a description on steganography is loaded as text file to hide in the BMP file shown in Fig 7.2.
After the text file is loaded there is no effect on the picture before and after the file is hidden. The text file can be retrieved using the option Recover text in the main menu.

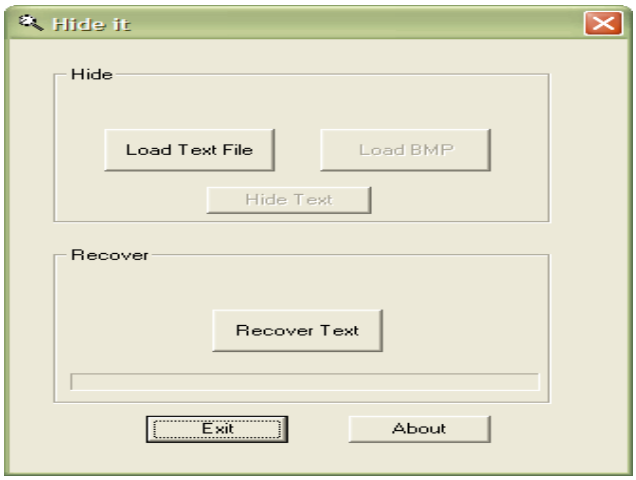

Figure 7.1 Main Screen

The cover medium and the stego-object is in Figure 7.2 ,shows no difference after the text file is loaded.

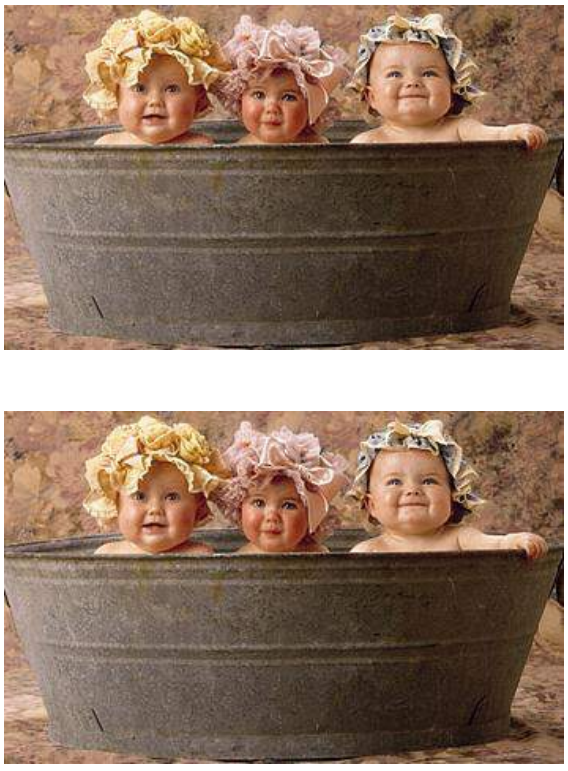

Figure 7.2 Bitmap image before and after the data is hidden

The text file retrieved in Figure 7.3, The difference in images before and after the image is hidden is shown in Figure 7.4,

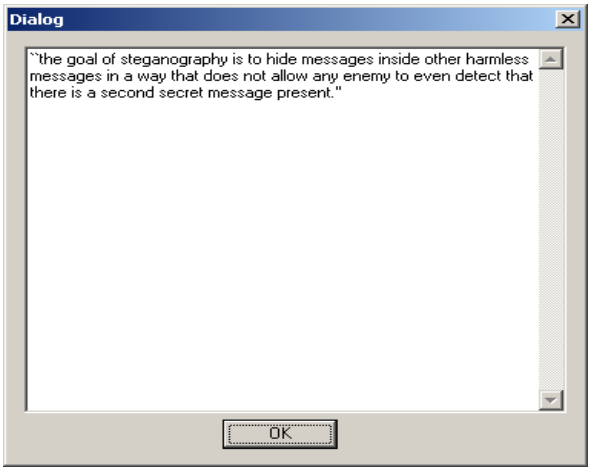

Figure 7.3 The Retrieved text File

Figure 7.4 Image before Data and after the data is hidden.

8. Conclusion

Steganography can be a powerful technique for secretive communication in digital technology and opens a door for new methods of information transmission over shared channels. The ability to use existing file structures specifically bitmap images, make them good candidates to be modified without detection. The increased high capacity bit plane algorithm increases the capacity of the messages to be embedded in the cover. This algorithm can be also enhanced by choosing the best bit plane before embedding a message. The best plane can be calculated as the one with least difference in value between the bits in the character to be embedded and the bits in the plane. Further many other characteristics can also be chosen to choose the best plane like the color depth, resolution and intensity. This algorithm can also be implemented in audio files.

References

1. N.Johnson and S. Jajodia, "Exploring steganography: seeing the unseen," IEEE Computer, pp. 26-34, February 1998.

2.W.Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," IBM Systems Journal, Vol. 35, No. 3 and 4, pp.313-336, 1996.

3.R.Anderson and F. Petitcolas, "On the limits of steganography," IEEE Journal on Special Areas in Communications, Vol. 16, No. 4, pp. 463-473, May 1998.

4. R.B. Wolfgang and E.J. Delp. "A watermark for Digital images," Proceedings of the IEEE International Conference on Image Processing, 111:219-222, September 1996.

5. Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. Kuhn, "Information Hiding – A Survey."Proceedings of the IEEE, vol. 87, no. 7, pp. 10621078,July 1999.

6. Fridrich, J., Goljan, M., Du, R.: Detecting LSB Steganography in Color and Gray Images. Magazine of IEEE Multimedia (Special Issue on Security), October-November, pp. 22-28. (2001).

7. M.M Amin, M. Salleh, S. Ibrahim, M.R. Katmin, and M.Z.I. Shamsudain, " Information Hiding using steganography," *Proc of IEEE*, pp.21-25, 2003.

8. Lee, Yeuan-Kwen and Ling Hwei Chen. "High Capacity Image Steganographic Model." Vision, Image and Signal Processing. IEEE Proceedings 147.3 (2000): 288-294.

9. ttp://www.weprintcolour.com/introtobitmaps.html for bitmap images manipulation.