IJCSC International Journal of Computer Science & Communication Vol. 1, No. 2, July-December 2010, pp. 113-116

Steganography-Concealing Messages in Images Using LSB Replacement Technique with Pre-Determined Random Pixel and Segmentation of Image

Rita Rana[1] & Er. Dheerendra Singh[2]

[1,2]Bhai Maha Singh College of Engg, Muktsar, Punjab, India
Email: [1]ritadly@yahoo.com, [2]dheerendra_singh76@yahoo.co.in

ABSTRACT

A new steganography approach for hiding encrypted data in an image is proposed. This approach uses the Least Significant Bits (LSB) replacement. First the secret message is encrypted by changing position of bits using transformations. The binary representation of the encrypted data is divided into four parts. The image is also then divided into four block. Each part of data is hidden in each block by selecting a pixel in a random manner as determined by an algorithm designed for it. Experimental results show that the scheme proposed in this paper has a high security level and better stego-image quality.

Keywords: Encryption, Steganography, Least Significant Bits, Random Pixel

1. INTRODUCTION

Steganography is the ***art of concealing the existence of information within seemingly innocuous carriers***. It can be viewed as akin to cryptography. Both have been used throughout recorded history as means to protect information[1]. At times these two technologies seem to converge while the objectives of the two differ. Cryptographic techniques "scramble" messages so if intercepted, the messages cannot be understood. Steganography, in an essence, ***"camouflages"*** a message to hide its existence and make it seem "invisible" thus concealing the fact that a message is being sent altogether. An encrypted message may draw suspicion while an invisible message will not.[3]

Given an original piece of data d, there is a threshold t below which any changes to the data won't be spotted by a person. t depends on the experience of the observer, but there is a minimum that's beyond the capabilities of the human senses. Thus, we can always afford to make a change c on d without being spotted, as long as [4]

$$d + c < t$$

As a rule of thumb, we must give the attacker as little stego-data as we can, so he won't be able to gather any good measurement of entropy.

What distinguishes classical steganography from invisible digital watermarking is that in the former what is important is the hidden message and not what a casual observer can see, while in watermarking[5] we are adding content to some data which are important by themselves, for the purpose of completing them or protecting them.

The most popular and frequently used steganographic method is the Least Significant Bit embedding (LSB)[6]. It works by embedding message bits as the LSBs of randomly selected pixels. The pixel selection is usually determined by a secret stego key shared by the communicating parties. Today, the vast majority of steganographic programs9 available for download on the Internet use this technique (Steganos II, S-Tools 4.0, Steghide 0.3, Wb Stego 3.5, Encrypt Pic 1.3, StegoDos,). [1]

The popularity of the LSB embedding is most likely due to its simplicity as well as the [false] early belief that modifications of pixel values by 1 in randomly selected pixels are undetectable because of the noise commonly present in all digital images of natural scenes. The existing LSB steganography embeds message into cover image by using message bit stream to replace the cover image's LSB directly. For increasing the [2]embedding capacity, two or more bits in each sample value can be used to embed messages without detectably degrading the cover image.

2. PROPOSED STEGANOGRAPHY METHOD

The application Invisible is designed using steganography method that consists of two main processes. The two processes are concealing and extraction of secret message from the image.

2.1. Concealing Message (Sender Side)

The proposed method is designed for BMP images. It first compares the length of the message to be concealed with the size of the image to ensure that the image can hold the secret file. If the size of secret file is more, then a

new image is selected. When using a 24 bit color image, a bit of each of the red, green and blue color components can be used, so a total of 3 bits can be stored in each pixel.

Thus, a 800 × 600 pixel image can contain a total amount of 800x600x3=1.440.000 bits (180.000 bytes) of secret data.

Pixel of cover image(color)
R G B
11001010 11100010 01010101 (24 bit)

1 1 0 0 1 0 1011110101
Text to be inserted (binary form)
Pixel of stego image
R G B
11001011 11100011 01010100 (24 bit)

It follows three levels of security.

Level I- The binary representation of secret message is encrypted using transformations. Rotating by 90 degrees followed by flipping the bits upside down.

Rotation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Thereafter flip-upside-down. The message is then divided into four parts.

Hidden Message	
Text I	Text II
Text III	Text IV

Level II-The image is divided into four blocks as shown below

BLOCK I	BLOCK II
BLOCK III	BLOCK IV

Level III: Within each block a pixel is selected using a predetermined method which then becomes the stego key. The method uses combination of odd and even rows and columns respectively as given below

- odd number of rows and even number of columns

  for i= 1 to height of image step 2

  for j= 2 to width of image step 2

- odd number of rows and odd number of columns

  for i= 1 to height of image step 2

  for j= 1 to width of image step 2

- even number of rows and even number of columns

  for i= 2 to height of image step 2

  for j= 2 to width of image step 2

- even number of rows and odd number of columns

  for i= 2 to height of image step 2

  for j= 1 to width of image step 2

Each text block is inserted in image block, in the sequence block III-II-I-IV.

Algorithm for Concealing messages (Sender Side)

Input: message, cover image

Output: stego image(containing message)

1: store size of message to be hidden in image

2: store method used for insertion

3: Convert message to binary form

4: rotate by 90 degree

5: flip upside down store in an array

6: if message_size<3*rows*cols of image

7: while data left to embed do

8: get block in which to start embedding

9: get predetermined random pixel

10: convert to binary form

11:replace LSB of Red Green and Blue with message bit

12: convert to decimal

13: end while

14: else

15: select an image of appropriate size

16:end if

2.2. Extracting Message (Receiver Side)

The same stego key is used for decoding of secret message from the stego image. The stego key is used to generate the same random number with which selection of the pixels is done and the order of block.

Algorithm Extraction message (Receiver side)

Input: stego image(containing message)

Output: hidden message

1: retrieve size of message hidden in image

2: retrieve method used for insertion

3: while data left to retrieve do

4: get block in which to start retrieving

5: get predetermined random pixel

6: convert to binary form

7: store LSB of red Green and Blue in an array

8: convert to decimal

9: end while

10:flip the lsb bits stored in an array upside down

11: rotate be -90 degree

12: convert to character and read the message.

3. EXPERIMENTAL RESULTS AND DISCUSSION

The algorithm was applied on a bit mapped (bmp) image that has the size of 128 pixels ×128 pixels with 256 colors. To evaluate the impact of the insertion process on the images, hidden message of different number of bytes were taken.

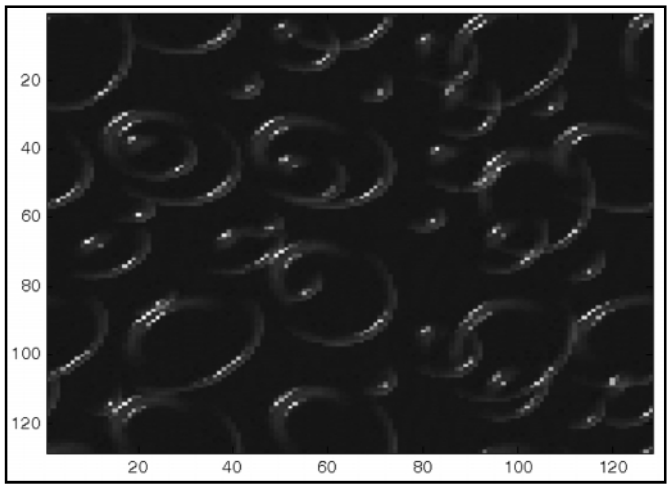

Fig 1: Cover Image

No. of bytes of the image= 128x128x3=49152

Hence it can hold maximum 49152-2=49150 bits as 1 bit is inserted in each byte and 2 bytes are reserved to store size of the text inserted.

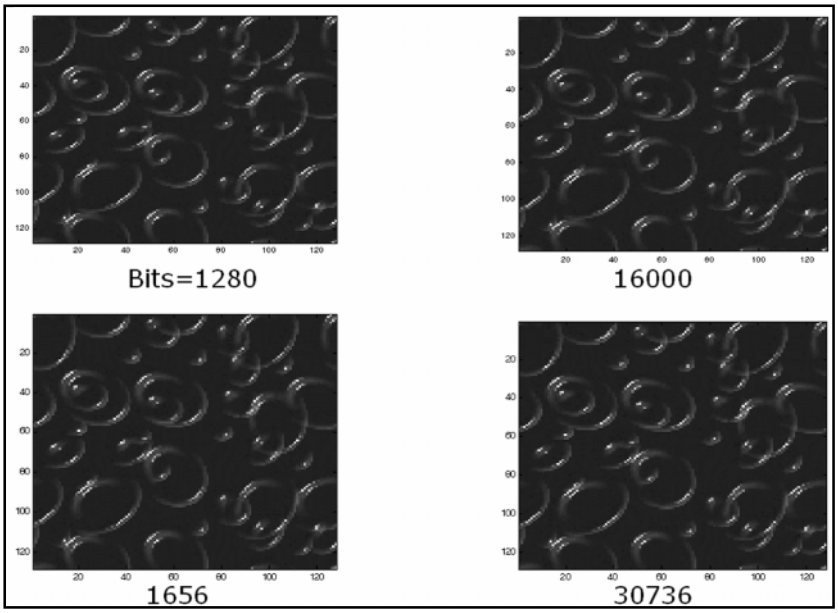

Fig 2: Stego Images with Different Number of Bits Hidden in them

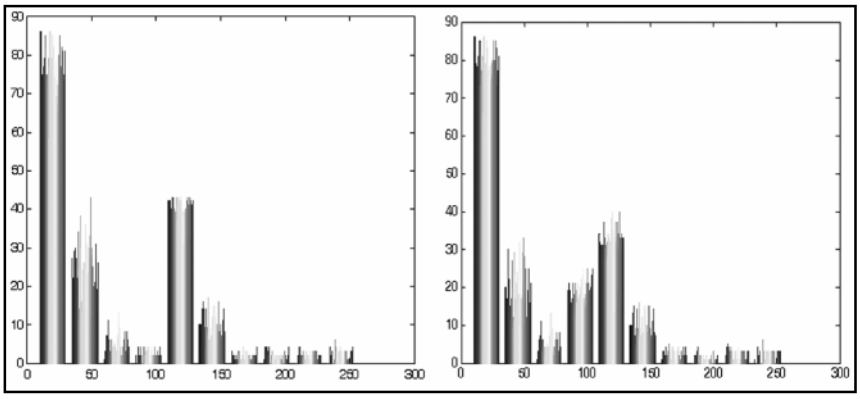

Fig 3: Histogram Analysis of Cover Image and Stego Image for 30736 Bits

It can be observed from the histogram that noise content is not perceived by a human eye hence information upto 80% of the size of the image can easily be hidden in it

Table below shows the part of data for cover and stego image for 11200 bits of data hidden in the cover image.

Table 1
Part of Block I (Rows: 1:5 and Columns 1:5, Red Color)

Cover Image	Stego Image
41 41 33 33 24	40 40 33 33 24
99 24 24 24 16	98 24 24 24 16
24 16 16 16 16	24 16 16 16 16
16 16 16 16 16	16 16 16 16 16
16 16 16 16 16	16 16 16 16 16

Table 2
Part of Block II (Rows: 5:10 and Columns 60:65, Red Color)

Cover Image	Cover Image
24 16 16 16 16 16	24 16 16 16 16 16
24 24 16 16 16 16	24 24 16 17 16 16
24 33 16 16 16 16	24 34 16 16 16 16
24 33 41 16 16 16	24 33 41 16 16 16
24 41 57 24 33 49	24 41 57 24 33 49
41 49 74 41 41 49	41 49 74 41 41 49

Table 3
Part of Block III (Rows:60-65 Column:1-5, Red Color)

Cover Image	Stego Image
16 16 16 16 16	16 16 16 16 16
16 16 16 16 16	16 16 16 16 16
16 16 16 16 16	16 16 16 16 16
16 16 16 16 16	16 16 16 16 16
16 16 16 16 16	17 16 17 16 17
16 16 16 16 16	16 16 16 16 16

Table 4
Part of Block III (Rows:65-70 Column : 65-70, Red Color)

Cover Image	Stego Image
16 16 16 16 16	16 16 16 16 16
16 16 16 16 16	16 16 16 16 16
16 16 16 16 16	16 16 17 16 16
16 16 16 16 16	16 16 16 16 16
16 16 16 16 16	16 15 16 16 16
16 16 16 16 16	16 16 16 16 16

As can be observed the hidden message is embedded in the complete image starting from block 3 then 1, 2 and 4. There is no defined pattern in which the embedding is taking place resulting in high quality stego image.

4. CONCLUSION

In this block and predetermined pixel based stego process, the secret message is first of all encrypted using data encryption standard and secondly, embedding the encrypted data in the binary image by segmenting it into blocks. There are three levels of security hence making it difficult for a steganalyst to decipher the hidden message. The random pixel selection makes it difficult to find the sequence of the message hence resulting in a strong stego image. The breaking of text into blocks and then storing them from block 3 instead of block 1 helps further in making the message secure.

REFERENCES

[1] Neil F. Johnson, Sushil Jajodia, George Mason University(1996), "Exploring Steganography: Seeing the Unseen", *IEEE Computers*, February 1998, pp. 26-34.

[2] W. Bender, D. Gruhl, N. Morimoto, A. Lu (1996), "Techniques for Data Hiding" *IBM Systems Journal*, **35**, Nos 31996.

[3] Neil F. Johnson, Sushil Jajodia, "Center for Secure Information System", George Mason University, "Steganalisys of Images Created Using Current Steganography Software", http://isse.gmu.edu/~csis.

[4] Ross J. Anderson, Fabien A.P. Petitcolas(1998), "On the Limits of Steganography", *IEEE Journal of Selected Areas in Communications*, **16(4)**, 474-481, May 1998.

[5] Lisa M. Marvel, Charles G. Boncelet, and Charles T. Retter (1998), "Reliable Blind Information Hiding for Images", 2nd Information Hiding Workshop, 1998.

[6] Ross Anderson, Roger Needham, Adi Shamir, "The Steganographic File System", 2nd Information Hiding Workshop, 1998.

[7] M. S. Sutaone, M.V. Khandare,PIET's College of Engineering, Pune, "Image Based Steganography Using LSB Insertion Technique".

[8] Sanjeev Manchanda, Mayank Dave and S. B. Singh," Customized and Secure Image Steganography Through Random Numbers Logic", *Signal Processing: An International Journal*, **1,** Issue (1).

[9] Karl Sch R.Chandramouli, Nasir Menon,(2001), "Analsis of LSB Based Image Steganography Techniques".

[10] Kevin Curran, Karen Bailey, University of Ulster, Institute of Technology (2003), Letterkenny, Ireland, "An Evaluation of Image Based Steganography Methods " *International Journal of Digital Evidence Fall 2003*, **2**, Issue 2.

[11] M. M Amin, M. Salleh, S. Ibrahim, M.R.K Atmin, and M.Z.I. Shamsuddin, (2003)," Information Hiding using Steganography", *4* National Conference on Telecommunication Technology Proceedings*, Shah Alam, Malaysia, 0-7803-7773-7/, 2003 IEEE.

[12] Mamta Juneja 1, Parvinder Singh Sandhu2 (2009), "Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption", *2009 International Conference on Advances in Recent Technologies in Communication and Computing*.