International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 13
October 2017

EDGE Based Image Steganography for Data Hiding

M.Krishna[1], V.Devi Satya Sri[2], B S B P Rani[3]
marlapallikrishna@gmail.com[1], vdevisatyasri@gmail.com[2], bsbprani.425@gmail.com[3]

Abstract
Steganography is covert communication, which means to hide the very existence of a message from a third party. Due to growing need for security of data, image steganography is gaining popularity. The traditional image steganography algorithm is Least Significant Bit embedding, but it can be easily detected by the attackers as it embeds data sequentially in all pixels. Instead of sequentially embedding data, data can be embedded in random pixels, but it causes speckles in the image. A better approach is to hide the data in the regions like edges. An attacker has less suspicion of the presence of data bits in edges, because pixels in edges appear to be either much brighter or dimmer than their neighbours. So we present a novel technique to hide data in the edges of the image by extending the Least Significant Bit embedding algorithm. This algorithm hides data in the edge pixels and thus ensures better security against attackers.

Keywords - Steganography, Encoder, Decoder, Edge detection, Stego image.

1. INTRODUCTION

STEGANOGRAPHY comes from the Greek Words: STEGANOS – "Covered", GRAPHIE – "Writing". Generally the sender writes an innocuous message and then conceals a secret message on the same piece of paper. The main goal of steganography is to communicate securely in a completely undetectable manner and to avoid drawing suspicion to the transmission of a hidden data. It is not to keep others from knowing the hidden information, but it is to keep others from thinking that the information even exists.

While legitimate and justifiable applications for Steganography exist, criminally-minded People or Organizations can also use it for underground, illicit and Deliberately secret Communication. However Steganography operates at a more complex level as detection is dependent on recognizing the underlying hidden data. It also includes a vast array of methods of secret communications that conceal the very existence of the message. In Steganography, data is hidden inside a vessel of container that looks like it only, but contains something else. A variety of vessels are possible, such as digital images, sound clips, and even executable files.

Historical tricks include invisible inks, tiny pin punctures on selected characters, minute differences between hands written characters, pencil marks on type written characters, grilles which cover most of the message except for a few characters, and so on.

Steganography is the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message. Due to growing need for security of data image steganography is gaining popularity. The main goal of steganography is to communicate securely in a completely undetectable manner and to avoid drawing suspicion to the transmission of a hidden data Discuss ancient steganography techniques. There are many steganography applications for digital image, including copyright protection, feature tagging, and secret communication**.** Unfortunately the members of terrorist organizations are using steganography as a tool to attack against the western interests.

In general, the information hiding process extracts redundant bits from cover object. The traditional Image steganography algorithm is Least Significant Bit embedding, the advantage of LSB is its simplicity to embed the bits of the message directly into the LSB plane of cover-image and many applications use this method. But it can be easily detected by the attackers as it embeds data sequentially in all pixels. If a steganography method causes someone to suspect that there is secret information in the carrier medium, then this method fails. Instead of sequentially embedding data, data can be embedded in random pixels, but it causes speckles in the image.

The previous algorithms LSB and Random LSB concentrate on hiding data in the least significant bit position of all or some selected pixels. They are not particularly concentrating on special pixels. To overcome these problems we proposed a novel image steganography algorithm based on least significant bit embedding algorithm (LSB) for hiding secret messages in the edges of the image. The organization of the paper is as follows. In section 2 the related work is discussed. In section 3 our proposed method is described. Finally, the simulation results are presented in section 4.

2. LITERATURE SURVEY

The data can be visible in basic formats like: Audio, Video, Text and Images etc. These forms of data are detectable by human hiding, and the ultimate solution was Steganography. The various types of steganography include:

Image Steganography:
The image steganography is the process in which we hide the data within an image so that there will not be any perceived visible change in the original image. The conventional image steganography algorithm is LSB embedding algorithm.

Audio Steganography:
Steganography can be applied to audio files i.e., we can hide information in an audio file, it can be called Audio Steganography. The audio file should be undetectable.

Video Steganography:
Steganography can be applied to video files i.e., if we hide information in a video file, it can be called Video Steganography. The video file should be undetectable by attacker.

Text files Steganography:
Steganography can be applied to text files i.e., if we hide information in a text file, it is called Text Steganography. The general process of steganography i.e., preparing a stego object that will contain no change with that of original object is prepared but using text as a source. The basic image steganography algorithm is Least Significant Bit embedding.

Least Significant Bit embedding (LSB)
In a gray scale image each pixel is represented in 8 bits. The last bit in a pixel is called as Least Significant bit as its value will affect the pixel value only by "1". So, this property is used to hide the data in the image. Here we have considered last two bits as LSB bits as they will affect the pixel value only by "3". This helps in storing extra data.

Least Significant Bit embedding (LSB)
Algorithm 1: LSB algorithm – Encoder
Input: Input image: steg
Input message: m
Output: Stego object

1 Initialize ;
2 **foreach** *character in message m* **do**
3 mask last two bits of image pixel in steg;
4 hide first two bits of the character;
5 **Goto** next pixel in the image;
6 mask last two bits of image pixel in steg;
7 hide next two bits of the character;
8 **Goto** next pixel in the image;
9 mask last two bits of image pixel in steg;
10 hide next two bits of the character;
11 **Goto** next pixel in the image;
12 mask last two bits of image pixel in steg;
13 hide last two bits of the character;
14 **Goto** next pixel in the image;
15 **end**

Algorithm 2: LSB algorithm – Decoder
Input: stegoobject: steg
Output: Message msg

1 Initialize ;
2 **Repeat**
3 3 k1 = Read last two bits of image pixel in steg;
4 **Goto** next pixel in the image;
5 k2 = Read last two bits of image pixel in steg;
6 **Left Shift** k2 two times;
7 **Goto** next pixel in the image;
8 k3 = Read last two bits of image pixel in steg;
9 **Left Shift** k3 four times;
10 **Goto** next pixel in the image;
11 k4 = Read last two bits of image pixel in steg;
12 **Left Shift** k4 six times;
13 **Goto** next pixel in the image;
14 14 m = k1 **XOR** k2 **XOR** k3 **XOR** k4;
15 attach character m to msg;
16 **Until** m = „$"

Random Least Significant Bit Insertion (RLSB)
In this algorithm data is hidden randomly i.e., data is hidden in some randomly selected pixel. Random pixel is generated by using Fibonacci algorithm.
Random Least Significant Bit Insertion (RLSB)
Algorithm 3: RLSB algorithm – Encoder
Input: Input image: steg
Input message: m
Output: Stego object

1 Initialize;
2 **foreach** *character in message m* **do**
3 mask last two bits of image pixel in steg;
4 hide first two bits of the character;
5 **Goto** next *randompixel* in the image;
6 mask last two bits of image pixel in steg;
7 hide next two bits of the character;
8 **Goto** next *randompixel* in the image;
9 mask last two bits of image pixel in steg;
10 hide next two bits of the character;
11 **Goto** next *randompixel* in the image;
12 mask last two bits of image pixel in steg;
13 hide last two bits of the character;
14 **Goto** next *randompixel* in the image;
15 **end**

Algorithm 4: RLSB algorithm – Decoder
Input: Stegoobject: steg
Output: Message msg

1 Initialize ;
2 **Repeat**
3 k1= Read last two bits of image pixel in steg;
4 **Goto** next *random pixel* in the image;
5 k2= Read last two bits of image pixel in steg;
6 **Left Shift** k2 two times;
7 **Goto** next *random pixel* in the image;
8 k3= Read last two bits of image pixel in steg;
9 **Left Shift** k3 four times;
10 **Goto** next *random pixel* in the image;
11 k4= Read last two bits of image pixel in steg;
12 **Left Shift** k4 six times;
13 **Goto** next *random pixel* in the image;
14 m= k1 **XOR** k2 **XOR** k3 **XOR** k4;
15 attach character m to msg;
16 **Until** m = „$"

Algorithm 5: Generation of random pixel
Input: Number x
Size of row n
Output: Location of next random pixel
1 a=0;
2 b=1;
3 **For** *i*=1 *to* x **do**
4 c=a+b;
5 a=b;

6 b=c;
7 **end;**
8 **Return** the location of the b^{th} pixel;

The steganography by using existing algorithms can be easily detected as they hide data in all the pixels sequentially (LSB) or randomly selected pixel (RLSB). If a steganography method causes someone to suspect that there is secret information in the carrier medium, then that method fails.

3. EDGE LEAST SIGNIFICANT BITEMBEDDING (ELSB)

In ELSB, we use all the edge pixels in an image. Here, we first calculate the masked image by masking the two LSB bits in the cover image. Then we identify the edge pixels by using the Canny Edge detection method. After obtaining the edge pixels we hide the data in the LSB bits of the edge pixels only and send the stego object to the receiver. At the receiver, the stego object is again masked at the two LSB bits. Then the canny edge detector is used to identify the edge pixels. We will get same edge pixels at the sender and receiver since we used the same masked image to calculate the edge pixels. Thus we identify the bits where data is hidden. So we extract data from the two LSB bits of the identified edge pixels. Thus message is obtained.

Algorithm 6: ELSB – Encoder
Input: Input image: steg
Input message: m
Output: Stego object

1 Initialize ;
2 **for** i=1 to *m* **do**
3 **for** j=1 to n**do**
4 steg1(i,j) = mask last two bits in each pixel steg(i,j);
5 **end;**
6 **end;**
7 **foreach** *character in message m* **do**
8 hide first two bits of the character;
9 **Goto** next *edgepixel* in the steg1;
10 hide next two bits of the character;
11 **Goto** next *edgepixel* in the steg1;
12 hide next two bits of the character;
13 **Goto** next *edgepixel* in the steg1;
14 hide last two bits of the character;
15 **Goto** next *edgepixel* in the steg1;
16 **End**

Algorithm 7: ELSB – Decoder
Input: stegoobject: steg1
Output: Message msg

1 Initialize ;
2 **for** i=1 to *m* **do**
3 **for** j=1 to n**do**
4 steg2(i,j)=mask last two bits in each pixel of steg1(i,j);
5 **end;**
6 **end;**
7 **Repeat**
8 **Goto** next edge pixel in steg2;
9 k1= Read last two bits of image pixel in steg1**;**
10 **Goto** next edge pixel in steg2;
11 k2 = Read last two bits of image pixel in steg1**;**
12 **Left Shift** k2 two times;
13 **Goto** next *edge pixel* in steg2;
14 k3 = Read last two bits of image pixel in steg1;
15 **Left Shift** k3 four times;
16 **Goto** next *edge pixel* in the steg2;
17 k4 = Read last two bits of image pixel in steg1;
18 **Left Shift** k4 six times;
19 m = k1 **XOR** k2 **XOR** k3 **XOR** k4;
20 attach character m to msg;
21 **Until** m = „$"

4. RESULTS AND DISCUSSIONS

The edge based steganography is to embed secret data in the position of edge pixels, which meets the requirements of both in perception and robustness. The edge based steganography includes Algorithms 6 and 7 for encoding and decoding process respectively. We have used twenty different gray scale images for comparison on three different LSB embedding techniques.

- Sequential LSB (LSB) embedding
- Random LSB (RLSB) embedding
- Edge based LSB (ELSB) embedding

The LSB algorithm is applied on Fig.1 i.e., a jet flying over snow covered hills. The Fig.1 is the cover image and the Fig.2 is the stego image.


Fig.1 Original image


Fig.2 Stego image

The Random LSB algorithm is applied on Fig.3 i.e.,scenary.Fig.3 is the cover image and the Fig.4 is the stego image.

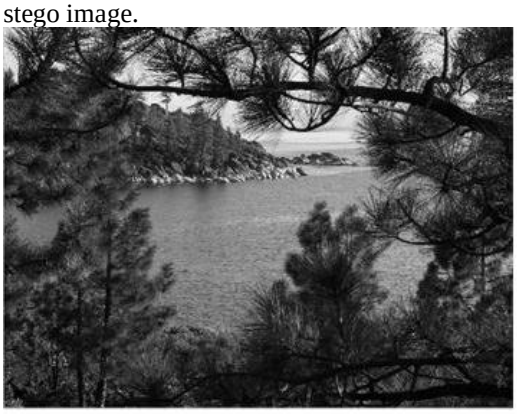
Fig.3 Original image

Fig.4 Stego image

The algorithm (LSB, RLSB) embeds data in all the image pixels sequentially or randomly. Thus attacker can easily detect the presence of hidden image.

The ELSB algorithm uses the canny edge detector for identification of edge pixels. ELSB algorithm is applied on a figure- a jet flying over snow covered hills. The Fig.5 is the cover image and the Fig.6 is the stego image

Fig.5 Original image

Fig.6 Stego image

Fig.7 Edge image

The Fig.7 is edge image obtained after applying the Canny Edge detector

The ELSB algorithm is applied on a figure- Mandril. The Fig.8 is the cover image and the Fig. 9 is the stego image.

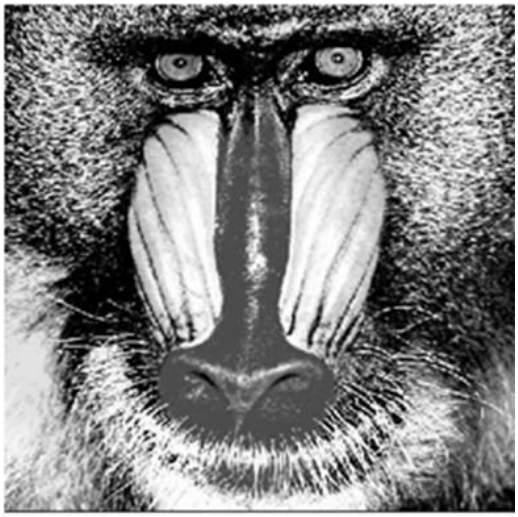

Fig.8 Original image

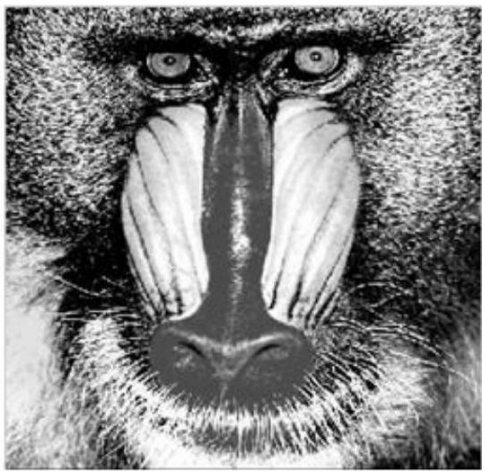

Fig .9 Stego image

Fig.10 Edge image

The Fig.10 is edge image obtained after applying the Canny Edge detector.

The ELSB algorithm is applied on a figure- Lena. The Fig.11 is the cover image and the Fig.12 is the stego image.

Fig.11 Original image

Fig.12 Stego image

Fig.13 Edge image

The Fig.13 is edge image obtained after applying the Canny Edge detector.

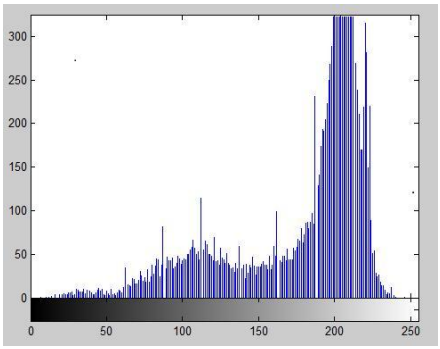

Fig.14 Histogram of Boat

The Fig.14 shows the histogram of the cover image and stego images of boat image.

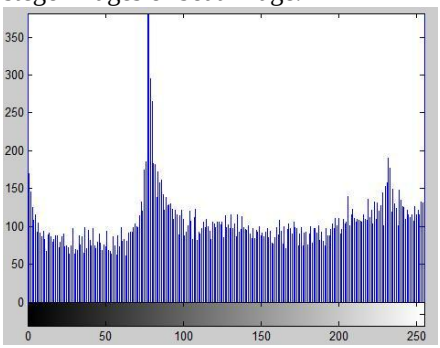

Fig.15 Histogram of Mandril

Fig. 15 shows the histogram of the cover and stego images of the Mandril image.
The histograms of the stego images using RSB and ELSB are compared.

5. CONCLUSIONS

In the Least Significant Bit embedding algorithm (LSB) and Random Least Significant Bit embedding algorithm (RLSB) an attacker can easily detect the presence of hidden image. To overcome these problems a new algorithm is proposed based on least significant bit embedding algorithm (LSB) for hiding secret messages in the edges of the image. The algorithm ELSB hides data in edge pixel. The proposed algorithm is applicable to all kinds of images and can be used in covert communication, hiding secret information like copyrights, trade secrets and chemical formulae.

6. REFERENCES

[1] R. Chandramouli and N. Memon, "Analysis of LSB based Image Steganography", IEEE ICIP, pp. 1022-1022, Oct. 2001.
[2] R.J. Anderson, F.A.P. Petitcolas, “On The Limits of Steganography”, IEEE Journal Of Selected Area in Communicafions, pp. 474-481, May 1998.
[3] N.F. Johnson, S. Jajodia, “Stag analysis: The Investigation of Hiding Information”, IEEE, pp. 113-116, 1998.
[4] K. Raba, "Steganography- the Art of Hiding Data", Information Technology of Journal, 3(3), pp.245-269, 2004.
[5] N.F. Johnson and S. Jajodia, "Exploring Steganography: Seeing Unseen", Computer 31, pp.26-34, 1998.
[6] D. Artz, “Digital Steganography: Hiding Data within Data”, IEEE Internet Computing, pp. 75-80, May-Jun 2001.

[7] D. Verton, "Expert Debate Biggest Network Security Threats", USA Today, 12 April, 2002.
[8] K. Maney, "Bin Laden's Messages could be Hiding in Plain Sight", USA Today 19 December, 2001.
[9] N. Provos, P. Honey man, "Detecting Steganography Content on the Internet".CITI Technical Report 01-11,oct-09,2001.
[10] N. Provos, "Probabilistic Methods for Improving Information Hiding", CITI Technical Report 01-1, January 31,2001.
[11] N.F. Johnson &S. Jajodia, "Steganalysis of Images Created Using Current Steganography Software", in Proceeding for the Second Information Hiding workshop, Portland 0 region, USA, April 1998, pp. 273-289.
June 2001.