International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 5, May - 2014

A Novel Steganographic Technique Using LSB Replacement in Image and Audio

Vivek Akula,
Master of Technology, ECE
Department, GITAM University,
Visakhapatnam.

Bhaskara Rao Jana,
Assistant Professor, ECE
Department, Anil Neerukonda
Institute of Technology & Sciences,
Visakhapatnam.

J. Beatrice Seventline,
Associate Professor, ECE
Department, GITAM University,
Visakhapatnam.

***Abstract*-The most popular technique of data hiding in steganography is by modifying Least Significant Bit (LSB) of the cover image. Here we propose a new approach to the LSB replacement in order to improve the quality of stego data. There can be no doubt that replacement of LSBs in digital images is a common choice for steganography but it remains popular in free steganography software. Moreover, this is the mechanism which inspires the majority of existing hiding methods. Here in this paper, implementation of the proposed method to different kinds of cover data and the use of DCT histograms to detect the presence of the hidden data is discussed. In this paper, we lay down a foundation for a thorough analysis of steganography and steganalysis and use this analysis to create practical solutions to the problems of detecting and encoding.**

Key words: LSB replacement, Steganography, PSNR.

1. INTRODUCTION

With the recent advances in computing technology and its intrusion in our everyday life, the need for private and personal communication has increased. Privacy over digital communication is desired when confidential information is being shared between two entities using computer communication.In this digital world there are techniques like watermarking, cryptography and steganography to handle this duty. Both techniques are main aspects of present world communications security. Digital watermarking is a technique which allows an individual to add hidden copyright notices or other verification messages to digital audio, video, or image signals and documents. Cryptography is a crude technique which hides the data by converting data into an unreadable form. And it arises suspicion of the presence of hidden data. On other hand the steganography deals the hidden data without raising any suspicions.

1.1. Steganography

Steganography is an art and science of encoding hidden messages in such a way that no one other than the sender and intended recipient suspects the existence of message. Basic representation of the steganography is as follow: Cover image + Message to be hidden + Stego key = Stego image.

There are various steganography techniques to hide the data. In steganography the message that is to be hidden is converted into bits and these bits are hidden into the cover. The common approaches for message hiding in images include LSB (Least Significant Bit) insertion methods, the frequency domain techniques and the spread spectrum techniques. Steganography is shown in the fig 1.

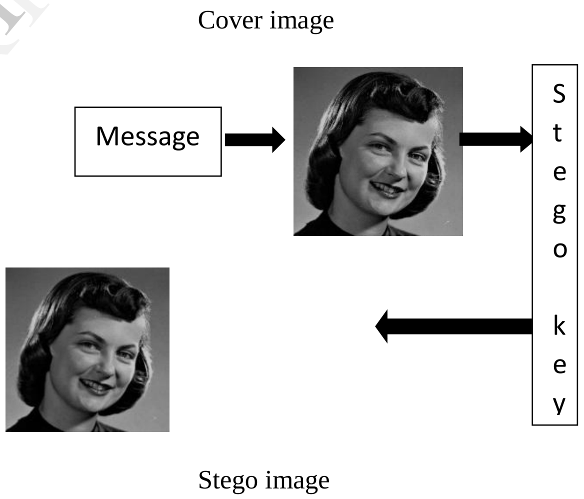

Fig 1: Steganography block diagram

Imperceptibility and the capacity are two important properties of any Steganography schemes, the former ensures that the embedding is imperceptible (cannot be detected by human eyes), and the latter indicates the efficiency of covert communication.

1.2. Steganalysis

Steganalysis is the art and science of detecting messages hidden using steganography. The goal of steganalysis is to identify suspected packages, determine whether or not they have a message encoded into them. There are various methods, mostly classified into signature and statistical steganalysis which are further divided into various techniques. Signature steganalysis works by observing the changes happened to cover image and detect the presence of the hidden data. Statistical steganalysis works by exploiting the cover image statistical characteristics to detect the presence of hidden data [1].

2. LSB REPLACEMENT

2.1 Image LSB replacement

LSB represents the Least Significant Bit. This technique is employed by replacing the least bit of the pixel of the cover image. As the least bit is only the modified bit it does not change the value of pixel it varies little bit that cannot be identified by the naked eye or just looking at. For a gray scale image each pixel is comprised of 8 bits. And for a colour image each pixel is formed by three 8 bits holding the information of RGB colour intensity.

To enhance message security we implement a layer of scrambling, called transposition, during the encoding process. Each bit of every message word is stored in the following pattern: RGBBGRRG; where R, G, and B mean the message value is stored in the Red, Green, and Blue Channels of the next available pixel. This means that every message word uses three pixel's worth of Red and Green Channel values while only require two pixel's worth of Blue Channel values. To implement this pattern the *stegancoder*and *stegandecoder*functions use a set of horizontal and vertical counters to determine what the next available pixel location is for each color channel. After a bit is encoded the counter(s) are increased to indicate the next available pixel until the entire message has been encoded.

LSB Replacement is done in similar fashion as shown below:

Let us see how LSB steganography works

Pixel from the picture

Pixel value: [206, 78, 76]

Binary Representation of the pixel value

206: 1100111**0**

78: 0100111**0**

76: 0100110**0**

The LSB of red component is replaced with the hidden message bit.

If the message bit is 1

Then the modification is 1100111**1** red component of the pixel and other components are left unaltered.

The modified pixel will be [207, 78, 76].

Pixel value: [206, 78, 76]

Binary Representation of the pixel value

206: 1100111**0**

78: 0100111**0**

76: 0100110**0**

The LSB of red component is replaced with the hidden message bit.

If the message bit is 1

Then the modification is 0100111**1** green component of the pixel and other components are left unaltered.

The modified pixel will be [207, 79, 76].

Pixel value: [206, 78, 76]

Binary Representation of the pixel value

206: 1100111**0**

78: 0100111**0**

76: 0100110**0**

The LSB of red component is replaced with the hidden message bit.

If the message bit is 1

Then the modification is 0100110**1** blue component of the pixel and other components are left unaltered.

The modified pixel will be [207, 78, 77].

The human eye cannot process these minute changes in colour difference. As the values are not changed much the cover image remains the same without much alteration. So that there is no chance for suspicion of data presence.

2.2 Audio LSB Replacement

The audio steganography follows the same method as the image steganography. First we choose an audio file of .wav format which acts as the cover medium for hiding the data. In our paper we considered a mono audio signal with a .wav extension as it is easier to work on using the matlab. Here we consider a 16 bit audio signal with an 8 kHz frequency signal for the paper.

The audio signal is break down to its respective values and convert them to the binary format respectively. But, as the audio signal is of 16 bit each value represents 16 bit binaryformat. The LSB is 15th bit of every binary format of the value and this is replaced with the encrypted message which is also converted to binary data. Hence the encoding is performed in such order to hide the data within the cover audio. The headers are formed with an 8bit and stored them accordingly in the cover data. Upon retrieving the first bits from the cover they are decrypted and obtain the number for the message data that is hidden and type of the message.

While decoding the data it use the same mechanism as it used to obtain the header file and reconstruct the data into text or the image based on the header information.

3. STEGANOGRAPHIC ENCODING OF DATA

The message may be in text format or image format is first breakdown to bits notation. And the cover image pixels are considered one by one, and lsb of the each pixel is replaced by the message bits.

Any steganographic algorithm is simply composed of stego function F and inverse of stego function F^{-1}.At transmitter section F converts cover image C and message I to stego image S as output. At receiver section stego image S is decoded using decoding algorithm which is F^{-1} to produce the information I [2].

If we represent the steganographic system as Ψ. Then the representation would be as follow:

$\Psi = \{F, F^{-1}, C, I, S\}$

$S = F(C, I)$

$I = F^{-1}(S)$

The above shown mathematical representation is common for all the steganographic algorithms.

In this paper we used the Simple Exclusive OR (XOR) to encrypt the data with the key and then modified the LSB of pixel with encrypted the data. Here I used a sequential encoding to hide the data.The data message could be an image or the text file. So in order to differentiate this I used header file concatenated with the encrypted data.

An 8 bit header is concatenated with the hidden data at beginning. Now the message is encrypted using a simple exclusive or (XOR) [3].

Message Bit	**KEY Bit**	**Encrypted Bit**
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: XOR operation

ENCODING:
Message : 11101000
Key : 10100011
Encrypted : 01001011

The Encrypted header along with the message is now embedded into the cover image forming the stego image.

3.1. *Steganographic Decoding*

Decoding is recovering the hidden message from the stego image using the key which is known to sender and receiver. The decoding process is the reverse process of the encoding. It obtains the LSB bits of the pixels and process them to find whether the data is image or text. Then it decrypt the bits to find the hidden message [3].

DECODING:
Encrypted: 01001011
Key : 10100011
Recovered Message: 11101000

As observed the recovered message is as same as the original message.

First the LSB bits are obtained and the header bits are calculated to know the type of message. The decoding process is done based on the header value. The decrypted data is used to obtain the hidden data.

The Bits obtained from decoding are used for reconstructing the original image or the message. The reconstruction is done carefully to obtain the hidden message.

4. LSB REPLACEMENT STEGANOGRAPHY

4.1 *Flow graph*

The whole process is depicted as a flow and explained as shown in following figure

Encoding:

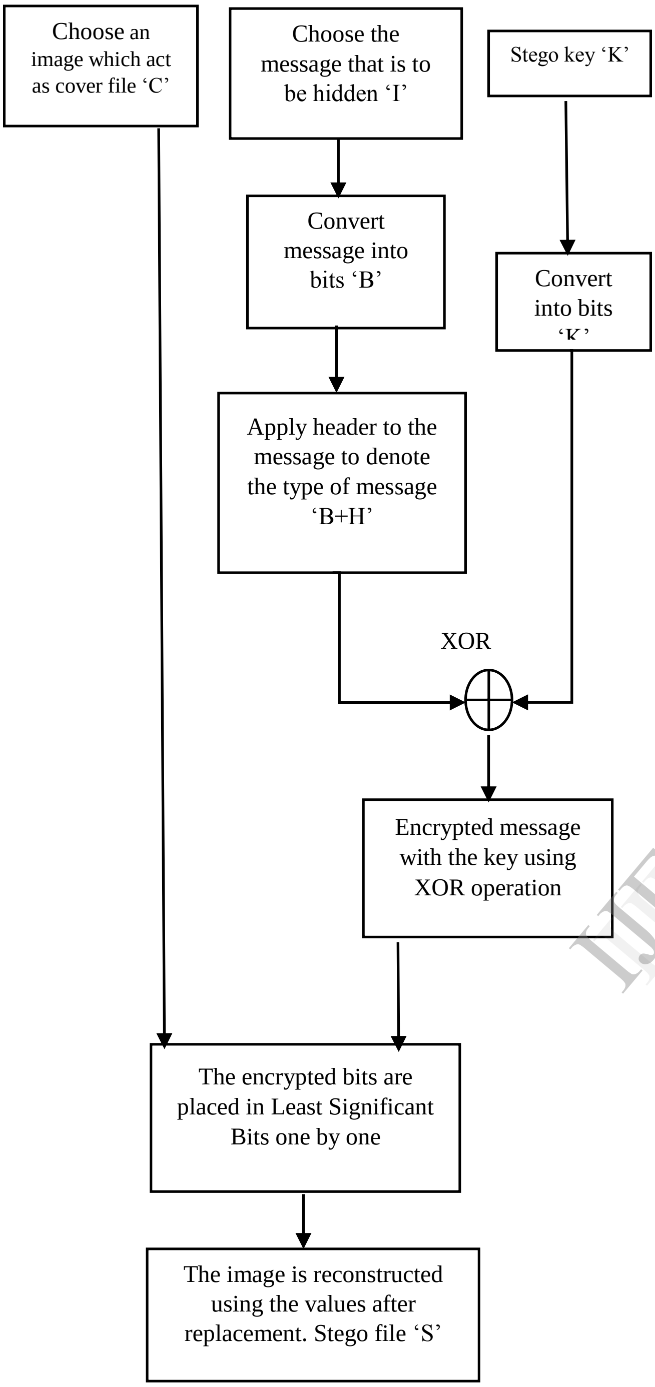
Choose an image which act as cover file 'C'
Choose the message that is to be hidden 'I'
Stego key 'K'
Convert message into bits 'B'
Convert into bits 'K'
Apply header to the message to denote the type of message 'B+H'
XOR
Encrypted message with the key using XOR operation
The encrypted bits are placed in Least Significant Bits one by one
The image is reconstructed using the values after replacement. Stego file 'S'

Fig.2: Encoding flow

Decoding:

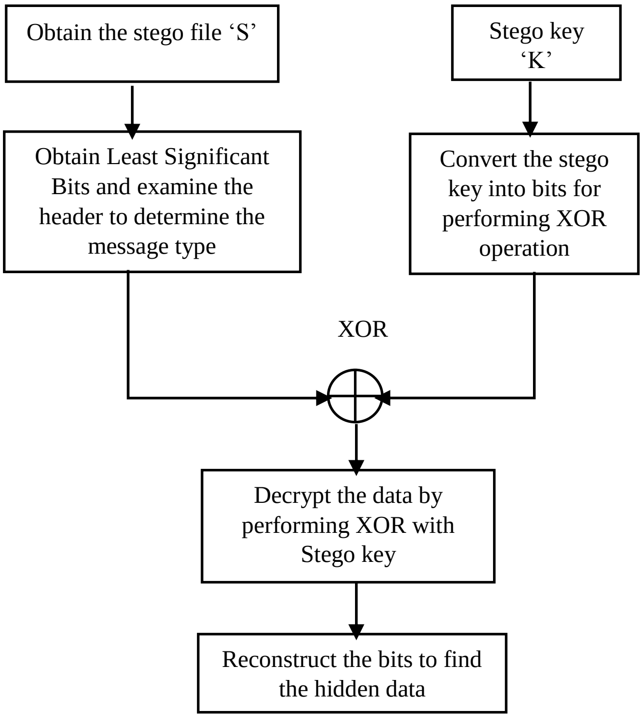
Obtain the stego file 'S'
Stego key 'K'
Obtain Least Significant Bits and examine the header to determine the message type
Convert the stego key into bits for performing XOR operation
XOR
Decrypt the data by performing XOR with Stego key
Reconstruct the bits to find the hidden data

Fig.2: Decoding flow

The above figures 2, 3 explains the steganographic encoding and decoding process.

4.2 *Algorithm*

The steganographic process is done in following steps:

Encoding:

Step 1: First choose the image to be a cover image and obtain the pixel data.

C: Cover file

$P_{1 \text{ to } n}$: Pixel values/ sample values (n= no of pixels / samples)

Step 2: Now prepare the data to be hidden either text or image.

I: Message data to be hidden

Step 3: Now break down the message into bit format.

I= $I_{1 \text{ to } 8}$ (Each element of the message is break down to 8 bits)

Step 4: Prepare a stego key which is key element for the steganography hiding process.
K: Stego Key

Step 5: Now convert the Stego key into bit format
K: $K_{1 \text{ to } 8}$ (similar to message data)

Step 6: Now add the header to the message so as to differentiate the type of message being hidden
H: Header bits
I_m= H+I

Step 7: Now perform the XOR operation between the header concatenated message and the Stego key
H_d= (H+I)$\oplus$ (K)

Step 8: Thus obtained data bits are used in hiding. Each bit is replaced with the least bit of each pixel respective colour component.
P_1 = P_1 {0 to 7}
H_d= H_d {1 to N}; N= no of bits to be hidden
P_1 (7) =H_d (1)

Step 9: After the replacement is done the file is reconstructed back which is known as Stego file.
S: Stego file
S= F {C, I, K}
F is encoding function.

Decoding:
Step 1: Obtain the stego file with the hidden data to process and break down to pixels and obtain pixel values.

S: Stego file
$S_{1 \text{ to } n}$ =pixel values.

Step 2: Now obtain the LSB bits and find out the header bits to determine the type of message.
H (1) =S_1 (7); Least bit of pixel 1
H=H (1 to 8); the header bits

Step 3: Now using the header value we know the type of message and the bits that are to be obtained.
O: Obtained bits

Step 4: Now perform the XOR operation on the bits obtained using the Stego key.
K: Stego key
P= O $\oplus$ K

Step 5: Thus the obtained data is used to reconstruct the data which is hidden message.
I=F^{-1}{S, K}
F^{-1} is inverse function.

5. EXPERIMENTAL SIMULATION

The algorithm is tested using PSNR (Peak Signal to Noise Ratio). PSNR is used to test the quality of the stego images. The higher the value of PSNR the better is the Stego image quality [4].

PSNR is obtained as following:

Cover file : C of size M×M

Stego file : S of size N×N

Cover and stego images with pixel values (p, q) from 0 to M-1 and 0 to N-1.

Calculate the MSE (Mean Square Error) between cover file and the stego file.

$$MSE = 1/MN \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} (C(p.q) - S(p,q))^2$$

$$PSNR = 10.\log_{10}(MAX^2/MSE)$$

MAX is the maximum pixel value of the images [4].

If the stego image obtained from the process has large PSNR value then the image has higher quality. Here we consider the PSNR value of stego image with hidden text and hidden image within the cover image. The values are as shown in table 2.

Cover data	**Size of cover**	**Message Hidden**	**Size of Hidden message**	**SNR (dB)**	**PSNR (dB)**
Lenna.jpg (colour)	768KB	Text	17	81.4122	92.9096
		Image	16.1KB	49.0649	60.4144
Lenna.bmp (colour)	26.7KB	Text	17	82.7249	93.8642
		Image	16.1KB	49.7313	61.0839
Lenna1.jpg (grayscale)	80.7KB	Text	17	82.3464	92.4617
		Image	16.1KB	40.6410	60.7462

Table 2: PSNR values

The figures represent the cover images and the stego images with the hidden data i.e., image and text.

The histograms represents the cover image and stego image separately considering each colour from RGB of the jpeg image.

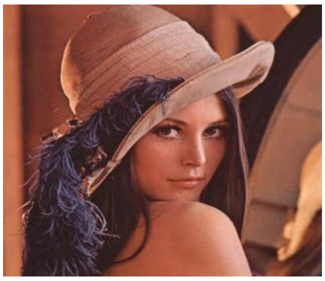

Fig 4: Cover image without hidden message

The figure 4 represents the cover image without any hidden data.

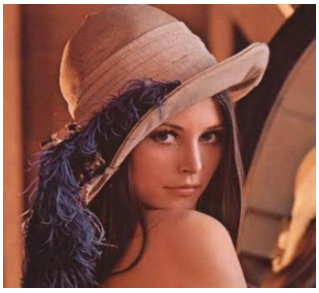

Fig 5: Stego image with hidden data (image)

The figure 5 represents stego data with hidden image in it. There is no difference in between the images.

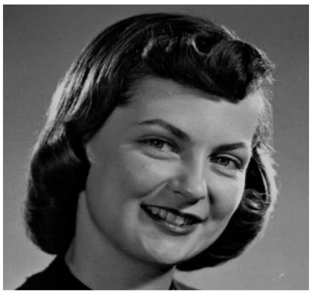

Fig 6: Cover image without data

The figure 6 is another example for steganography.

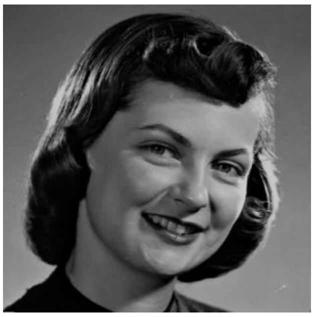

Fig 7: Stego image with hidden data (Text)

The figure 7 represents the stego image with the text data as hidden data.

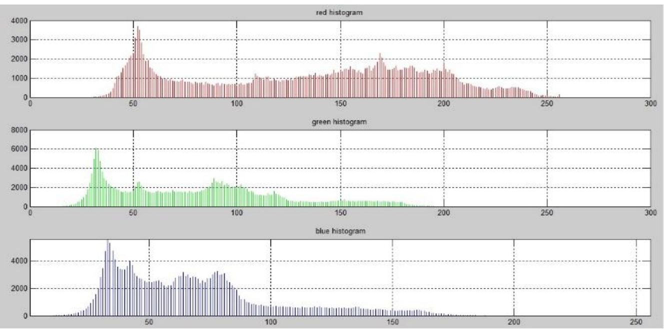

Fig 8: Histogram of cover image

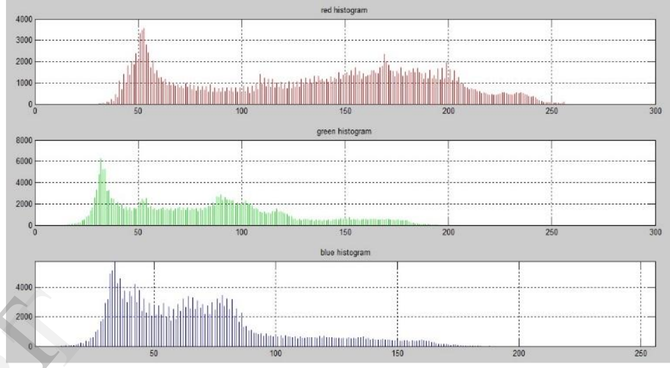

Fig 9: Histogram of stego image with the image hidden in it

The figure 8 and 9 represents the histograms of the cover image and stego image showing the little variations over the image. We can observe the variation over the red histograms. Similarly the blue histograms vary and very minute changes for green histograms. Similarly text encoded stego image produce the histograms with very little variations much smaller than that of the image encoded stego image.

Other than the histograms as seen earlier we could also plot the DCT coefficients and observe the changes in the stego image from the cover image.

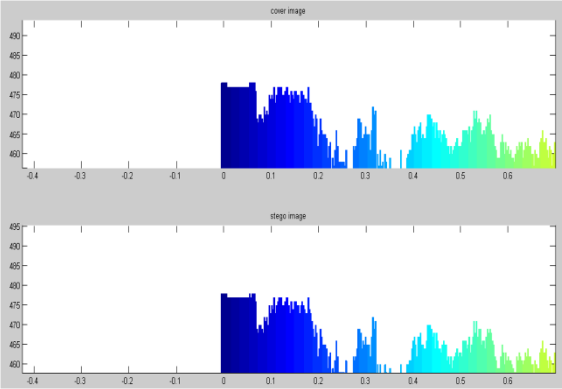

Fig 10: DCT histogram

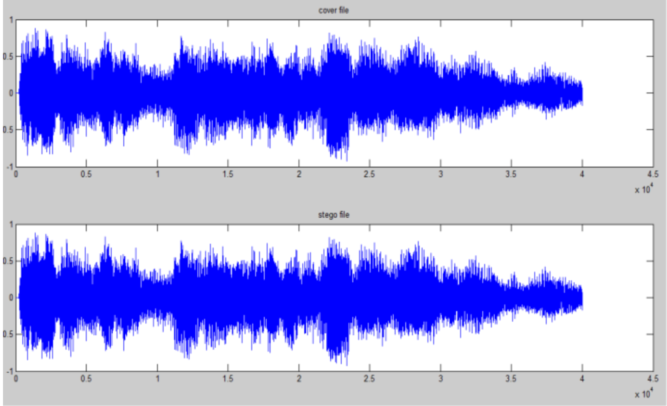

Fig 11: Cover data and stego data (audio format)

The figure 11 represents the results of the LSB replacement technique in audio format. And it is observed that the data is hidden well in the audio file using the LSB replacement.

6. CONCLUSION

The results show that the algorithm proposed here is effective to bring an improvement in PSNR value. Altering the weight of cover data and message data seem to improve the PSNR value. The proposed approach is tested on various data and provided better quality in steganography data.

The proposed algorithm is also tested with audio signals as the hidden data and proved to provide better results as observed from figures. This proves that the algorithm can be used on different types of the cover data.

REFERENCES

[1] AroojNissar, A.H.Mir, Classification of steganalysis techniques: A study, ELSEVIER.

[2] N.Provos, P.Honeyman, *Hide and Seek: An Introduction to Steganography*, IEEE Computer Security 2003.

[3] David Pipkorn, Preston Weisbrot, Steganography- The Hidden Message.

[4] R.Ibrahim, T.S.Kuan, Steganography algorithm to hide secret message inside an image.

[5] V.J.Rehna, M.K.Jeya Kumar, “A Strong Encryption Method of Sound Steganography by Encoding an Image to Audio”, 2012