Steganography Technique for Hiding Text Information in Color Image using Improved LSB Method

Deepesh Rawat
MTech (Digital Communication)
BTKIT,DWARAHAT
UTTARAKHAND, INDIA

Vijaya Bhandari
Asst. Professor (ECE),
BTKIT,DWARAHAT,
UTTARAKHAND, INDIA

ABSTRACT

In this paper author has purposed improved LSB substitution method for hiding text information written in text file into color image. In this method each character of secret message including special character such as space, enter, <,?, $etc. is converted in ASCII code then each value is converted in 8 bit binary number. Each bit of each character is embedded in last LSB of each pixel of cover image. Since only last bit each pixel of cover image get changed, this method is capable of producing a secret-embedded image that is totally indistinguishable from the original image by the human eye. This method is applied to cover image of bitmap and jpeg format to show that this technique is suitable for both format.

Keywords: Cover image, Data hiding, Histogram, LSB method, MSB, PSNR, Steganography,Stego-image

1. INTRODUCTION

The idea of information hiding is nothing new in the history. As early as in ancient Greece there were attempts to hide a message in trusted media to deliver it across the enemy territory. In the modern world of digital communication, there are several techniques used for hiding information in any medium. One of such technique is steganography[1] in which digital media mainly digital images are used as a medium for hiding information and theinformation in the form text, digital image, video or audio file may be used as secret message. The word steganography derived from two Greek words: steganos means covered and graphos means writing and often refers to secret writing or data hiding[2]. The major goal of steganography is to increase communication security by inserting secret message into the digital image, modifying the redundancy or nonessential pixels of the image[3],and is recently become important in a number of application areas especially military and intelligence agencies which require unobtrusive communications. Steganography and cryptography are both ways to protect information from unwanted parties but neither technology alone is perfect and can be compromised. If the presence of hidden information is suspected or evenrevealed, the purpose of steganography is partly defeated [4]. Cryptography and Steganography achieve the same goal via different means. Encryption encodes the datainto an unreadable format called cipher so that an unintended recipient cannot determine its intended meaning. On the other hand steganography attempts to prevent an unintended recipient from suspecting that the data is there [5]. Nowadays, using a combination of steganography and the other methods such as cryptography, information security has improved considerably. In addition to being used in the covert exchange of information, steganography is used in many other fields such as copyright, preventing e-document forging. For the past decade, many steganographic techniques for still images have been presented. A simple and well known method is directly hiding secret data into the least-significant bit (LSB) of each pixel in an image. Then based on the LSB technique, an algorithm hiding text information in color image is developed with improved stego-image quality[12].

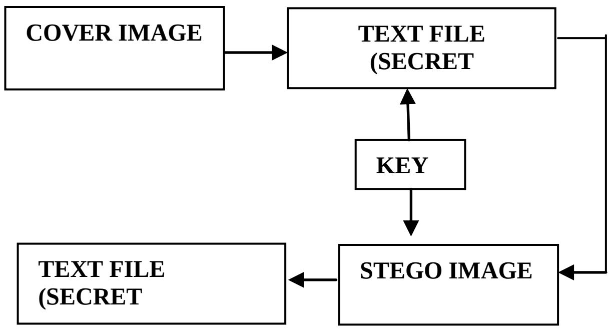

Fig.1. Steganography Process

2. LEAST SIGNIFICANT BIT

LSB based technique is most simple and straightforward approach in which message bits are embed in least significant bits of cover image. In LSB steganography, the least significant bits of the cover media's digital data are used to conceal the secret message [6]. LSB Steganography can be classified by two methods LSB replacement and LSB matching. The terminology LSB replacement/ LSB matching was firstly discussed by T. Sharp [7]. First is LSB replacement which is simplest of the LSB. LSB replacement steganography replace the last bits of cover image with each bits of the message that needs to be hidden. Second method is LSB matching [8] in which each pixel of the cover image is taken mainly in a pseudo-random order which is generated by a secret key, if the LSB of the cover pixel matches the bit of secret data no changes are done otherwise, one is added or subtracted from the cover pixel value, at random. Basic method of data hiding in an image is given as[6]-:

Let C be the original 8-bit grayscale cover-image of $M_c \times N_c$ pixels represented as

$C = \{x_{ij} \mid 0 \le i < M_c , 0 \le j < N_c$
$x_{ij} \in \{0, 1, \ldots\ldots 255\}$ …. (1)

M be the n-bit secret message represented as

$M = \{m_i | 0 \le i < N , m_i \in \{0,1\}\}$ ….. (2)

For embedding the n-bit secret message M into the k-rightmost LSBs of the cover-image C, thesecret message M is rearranged to form a conceptually k-bit virtual image M' represented as

$M' = \{m'_i \mid 0 \leq i < n',\ m'_i \in \{0, 1, \ldots 2^k - 1\}\}$…. (3)
where $n' < Mc \times Nc$.

The mapping between the n-bit secret message $M = \{m_i\}$ and the embedded message $M' = \{m'_i\}$
can be defined as follows:

$$m'_i = \sum_{j=0}^{k-1} m_{i \times k + j} \times 2^{k-1-j}$$

After that, a subset of n' pixels $\{x_{l1}, x_{l2}, \ldots\ldots x_{ln}\}$ is chosen from the cover-image C in a predefined sequence. The embedding process is completed by replacing the k LSBs of x_{li} by m'_i . Mathematically, the pixel value x_{li} of the chosenpixel for storing the k-bit message m'_i is modified to formthe stego-pixel x'_{li} as follows:

$$x'_{li} = x_{li} - x_{li} \bmod 2^k + m'_i$$

Algorithm[9] for LSB Based embedding and extracting process is given as-:

A LSB-based Embedding Algorithm

```
Input -: cover C
for i = 1 to Length(c),  do
$S_j \leftarrow C_j$
for i = 1 to Length(m),  do
Compute index $j_i$ where to store the ith message bit of m
$S_{ji} \leftarrow LSB(C_{ji}) = m_i$
End for
Output -: Stego image S
```

In the extraction process, given the stego-image S, the embedded messages can be directly extracted. Using the same sequence as in the embedding process, the set of pixels $\{x_{l1}, x_{l2}, \ldots\ldots x_{ln}\}$storing the secret message bits are selected from the stego-image. The k LSBs of the selected pixels are extracted and lined up to reconstruct the secret message bits.

A LSB-based Extracting Algorithm

```
Input -: Secret image s
for i = 1 to Length (m),  do
Compute index $j_i$ where to store the ith message bit of m
$m_{ji} \leftarrow LSB(C_{ji})$
End for
```

3. Improved LSB method for hiding text in an image

In this method each character of message including special character such as space, enter etc is converted in ASCII code then each value is converted in 8 bit binary number. Then, a subset of 8pixels $\{x_{l1}, x_{l2}, \ldots\ldots x_{l8}\}$ is chosen from the cover-image C in a predefined sequence. In this method RRGGGBBB sequence is chosen for embedding each bit of a character. The embedding process is completed by replacing the last LSB of cover image by a bit of secret image using key. This process is continued till all secret message bits get inserted in cover image. In the extraction process, given the stego-image S, the embedded messages can be directly extracted from stego image with the help of key, by using the same sequence as in the embedding process, the set of pixels{ RRGGGBBB }, Mathematically, the embedded message bits m' can be recovered by

$m'_i = x'_{li} \bmod 2^k$

Algorithm for embedding secret message-:
Steps to be carried out in this technique and implement in MATLAB are-:

i) Select a color image as cover image
ii) Now choose the text file containing secret information
iii) Enter the key of 8 bit that is key must be lies between 0 – 255
iv) Convert each character of secret message in ASCII code i.e. 97 for 'a', 48 for '0', 32 for 'space bar' etc. In Matlab this can be achieve by "double " command.
v) Determine the length of the message and padded zero to make it 8 characters long and make it header of the message, which means to add message length at the header of message
vi) Convert each ASCII code to its 8 bit binary equivalent
vii) Take cover image and convert in unsigned 8 bit integer so as to bring its pixels value in range [0 255]
viii) Separate cover image in RGB plane and insert 1st bit of secret message to last bit of first pixel of red plane, and second bit to the last bit of second pixel of red plane
ix) Now 3rd bit of secret message to last bit of first pixel of green plane, 4th bit to the last bit of second and 5th bit to the last bit of third pixel of green plane
x) Finally 6th , 7th and 8th bit of secret message to last bit of first pixel , second pixel and third pixel of green plane respectively. So 8 bits of first character of secret text message get embedded in 8 pixels of cover image (2 red, 3 green and 3 blue). Each time when a bit is embedded to a pixel of a plane increase its position by 1 so as to go on next pixel of that plane. This process continues till all bits get embedded in cover image.
xi) Resulting image is stego image containing information

Algorithm for extracting secret message-:
Steps to be carried out in this technique for extraction of the secret message are-:

i) Select stego image, and enter the key
ii) Separate stego image in RGB plane take mode 2 of first and 2nd pixel of red plane so as to get first 2 bit of first character of secret message.
iii) Now take mode 2 of 1st, 2nd and 3rd pixel of green plane to get 3rd, 4thand 5th bit of first character respectively. Finally take mode 2 of 1st, 2nd and 3rd pixel of red plane so as to get 6th , 7th and 8th bit of first character of secret message respectively. Each time mode 2 of pixel is taken increase its position value by 1 so as to go to next pixel. Run this step upto 8 times so as to get header which contain the information of secret message length
iv) Now repeat the same process again i.e. take mode 2 of 1st , 2nd and 3rd pixel of green plane to get 3rd, 4th and 5th bit of first character

respectively. Finally take mode 2 of 1st, 2nd and 3rd pixel of red plane so as to get 6th , 7th and 8th bit of first character of secret message respectively. Each time mode 2 of pixel is taken increase its position value by 1 so as to go to next pixel. Run this upto message length.

v) Convert binary code in decimal which represent ASCII value of secret message.

After converting it in character get secret message which can be saved as text file.

4. SIMULATION RESULT

This section presents experimental results obtained for two cover-image image. The first image is in “bitmap” format (Fig 2) while second image is same but in JPEG format (Fig 3)

Fig.2 Cover image (.bmp)

Fig. 3. Cover image (.jpg)

TEXT FILE
CONTAINING SECRET
INFORMATION

Fig.4 Text File (.txt)

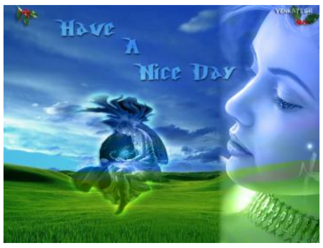

Fig.5.STEGO IMAGE (.bmp)

Secret image is same for both type of cover image i.e. it is the text information written in text file (Fig 4). Since each character of secret image converted into its equivalent ASCII value and then each code is converted in 8 bit binary, and each bit is inserted in last LSB of each pixel cover image, also 8 characters (64 bits for header) are embedded as header which signify secret message length so maximum number of character that can be embedded in an image can be calculated as follow. For example for an image of size 512 × 512 total (512 × 512 × 3) – 64= 786368 characters can be inserted (including special character such as enter, spacebar, comma, etc).

The result of embedding secret message in cover image is shown in Table 1. This result is obtain by taking same cover image of different format (bitmap and jpeg)of size 512 × 512 and a text file containing secret information, approx 5000 characters(including special character).For this we have calculated PSNR, MSE and RMSE.

The PSNR and MSE is then calculated as follows [10]:

$$\text{PSNR} = 10 \log_{10} \frac{\text{MAX}^2}{\text{MSE}} (\text{db})$$

Where

$$\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (\alpha_{ij} - \beta_{ij})$$

Here, $\alpha_{i,j}$ is the pixel of the cover image where the coordinate is (i, j), and $\beta_{i,j}$ is the pixel of the stego-image where the coordinate is (i, j). M and N represent the size of the image. A larger PSNR value indicates that the difference between cover image and the stego-image is more invisible to the human eye[10].

	PSNR	**MSE**	**RMSE**
BITMAP	62.62	**0.0358**	0.189
JPEG	62.54	**0.0362**	0.0190

Table1. Comparison of BITMAP and JPEG images result

5. Histogram Analysis

One of the best way of finding out a good steganography technique is the analyzing the histogram of all stego image and then compare them with original one. Histogram represents the number of pixels that have colors in each of a fixed list of color ranges, that span the image's color space, the set of all possible colors.

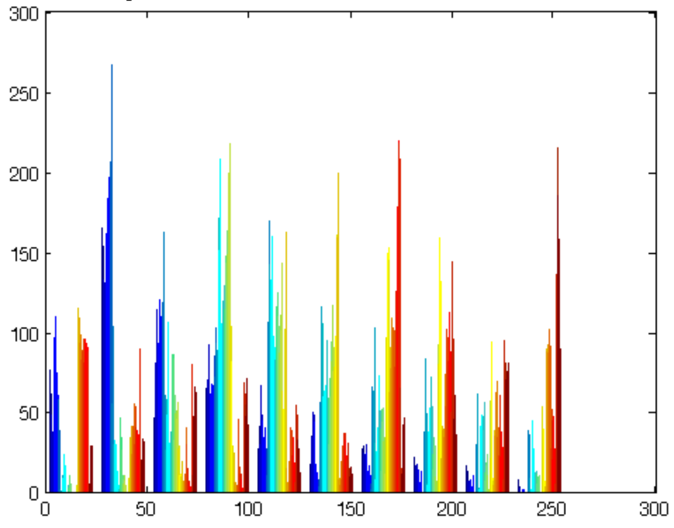

Fig.6. Histogram of cover image (.bmp)

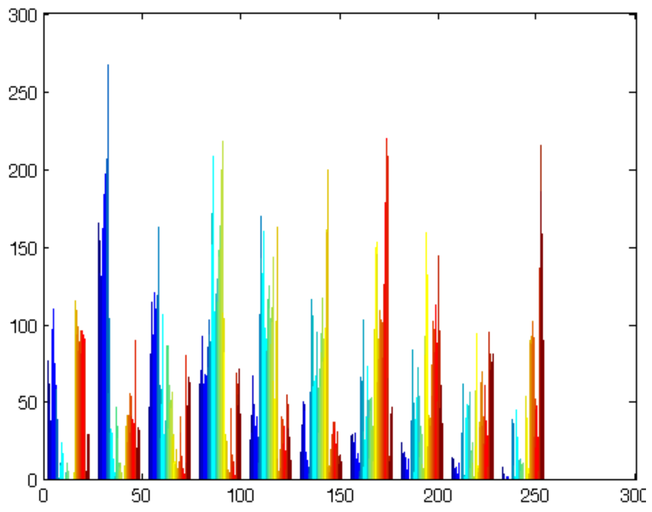

Fig.7. Histogram of stego image containing text information (Improved LSB method)

After comparing histogram of cover image (Fig 6) with the stego image (Fig 7) it is quite clear that histogram of stego image is almost similar to cover image, as there is change of only last bit of pixels. so this method is capable of producing a secret-embedded image that is totally indistinguishable from the original image by the human eye and can't be detect by histogram analysis method

6. CONCLUSION

In this paper we have described improved LSB method for hiding secret information written in text file in color image. We have choose bitmap and jpeg format image to show that this technique can be used to both type image. Also PSNR (peak signal to noise ratio), MSE (Mean Squared Error) and histogram analysis of stego image is calculated. With the increase in cover image size more secret information can be embedded.Since only last bit of each pixel get changed so there is negligible change is histogram which makes stego-image visually indistinguishable from the original cover-image of cover image also PSNR value is very high and MSE is quite low which show stego quality is very good

7. REFERENCES

[1]. M. M Amin, M. Salleh, S. Ibrahim, M.R.K atmin, and M.Z.I. Shamsuddin "Information Hiding using Steganography" 4* National Conference on Telecommunication Technology Proceedings, Shah Alam, Malaysia. 2003 IEEE.

[2] Moerland, T., "Steganography and Steganalysis", Leiden Institute of Advanced Computing Science,www.liacs.nl/home/ tmoerl/privtech.pdf

[3] Feng, J.B., Lin, I.C., Tsai, C.S., Chu, Y.P., 2006. Reversible watermarking: current status and key issues. International Journal of Network Security 2 (May), 161–170.

[4] Wang, H & Wang, S, "Cyber warfare: Steganography vs. Steganalysis", Communications of the ACM, 47:10, October 2004

[5] Westfeld, A., and G. Wolf, Steganography in a Video conferencing system, in proceedings of the second international workshop on information hiding, vol. 1525 of lecture notes in computer science,Springer, 1998. pp. 32-47.

[6] Chan, C.K., Cheng, L.M., 2004. "Hiding data in images by simple LSB substitution". Pattern Recognition 37 (March), 469–474.

[7]. T. Sharp, "An implementation of key-based digital signal steganography," in Proc. Information Hiding Workshop, vol. 2137, Springer LNCS, 2001, pp. 13–26.

[8] R. Chandramouli and NasirMemon, "Analysis of LSB Based Image Steganography Techniques", IEEE 2001.

[9] Neil F. Johnson, S.C. Katzenbeisser,"A survey of steganography technique"

[10] Chung-Ming Wang a, Nan-I Wu a, "A high quality steganographic method with pixel-value differencing and modulus function", accepted 24 January 2007

[11] Mohammad TanvirParvez and Adnan Gutub, "RGB Intensity Based Variable-Bits Image Steganography",APSCC 2008 – Proceedings of 3rd IEEE Asia-Pacific Services Computing Conference, Yilan, Taiwan, 9-12December 2008

[12] DeepeshRawat, VijayaBhandari , " A Steganography Technique for Hiding Image in an Image using LSB Method for 24 Bit Color Image", International Journal of Computer Applications (0975 – 8887) Volume 64– No.20, February 2013