An Ameliorate Image Steganography Method using LSB Technique & Pseudo Random Numbers

Kalyan Das
Department of Information Technology
St,Thomas' College of Engineering & Technology, Kolkata, India

Debanjan Choudhury
Department of Computer Science and Engineering
St,Thomas' College of Engineering & Technology, Kolkata, India

Prof. Samir Kumar Bandyopadhyay
Department of Computer Science & Engineering
University of Calcutta, Kolkata, India

Abstract

Steganography allows the hiding of secret data in non-secret medium. One of the major concerns in image steganography is the selection of pixels on the cover image where the secret image data is to be embedded. The objective of the paper is to propose an image steganography algorithm involving pseudorandom number generation and least significant bit alteration. The secret data is a black and white image. The cover image used is a color image having RGB channels. 100% recovery of the secret data has been shown, with the PSNR between the original cover and the modified cover image considerably high.

Keywords: Steganography, Cover Image, Data Hiding, Least Significant Bit, Pseudorandom Number Generation

I. INTRODUCTION

In today's world there is an increased need for security on the internet. Sensitive data needs to be sent from one person to another without being intercepted by a third person. The two major ways this is achieved is through cryptography and steganography [1]. While cryptographic algorithms deal with transforming the secret data into an unreadable format, steganography hides the data altogether [2]. Using cryptography, the attacker will know that there is some secret data that is being sent. But because the data is in encrypted format, it will be difficult for the attacker to retrieve the data. However, in case of steganography the secret data is hidden in some other carrier. Thus, the attacker just by seeing the message being sent will not know whether it contains any secret or not.

There are several techniques by which steganography can be achieved in images [3]. These can be broadly divide into substitution techniques, transform domain techniques, statistical techniques and spread spectrum techniques. Based on domain, image steganography can be classified into Spatial Domain and Transform or Frequency Domain. In substitution technique, certain bits in the cover image are replaced with bits from the secret image. When done properly the resulting cover has very little perceivable change. Least Significant Bit method is one of the most popular methods used. The pixels which will be altered in the cover image needs to be first selected.

In this proposed method, the selection of the pixels on the cover image which will be altered is done using pseudorandom number generation. Then bits from the secret image is inserted into the Least Significant Bit of the selected pixels. Only the Red channel is modified. The purpose is to make the method more secure. As only one bit is changed on a single channel for a particular pixel, the change cannot be identified visually [4].

II. LITERATURE SURVEY

A. *Least Significant Bit*

Least Significant Bit or LSB is one of the simplest steganography method in which the secret message is directly inserted into the pixels of the cover image [5]. This method is known to have a good imperceptible value, which means that the human vision cannot detect the image changes and is also effective [6].

The following example shows the way LSB steganography works:

– Secret message: 11010010

8 pixels of the cover image:

11010101 11011100 11011111 11010010
11010101 11010011 11101101 11100111

The underlined bits are the respective LSBs of the pixels on the cover image. These bits will be replaced by the bits from the secret message.
Same 8 pixels after insertion of message:

11010101 11011101 11011110 11010011
11010100 11010010 11101101 11100110

Since only a single LSB bit has been altered, the maximum change in the decimal values of the pixels will only be by 1. This change cannot be detected just by looking at the original cover and the modified cover side by side.

B. Pseudorandom Number Generator

Pseudo Random Number Generator (PRNG) refers to any algorithm that uses mathematical formulas to produce sequences of numbers. PRNGs generate a sequence of numbers approximating the properties of random numbers. The generated sequences are not truly random. It is determined by an initial value known as the seed.
Two important characteristics of PRNG are as follows:
1) Efficient
PRNG has to produce many numbers within a short amount of time.
2) Deterministic
Using the same initial seed value will generate the same sequence of numbers for a particular PRNG.
Mersenne Twister is the most widely used general-purpose pseudorandom number generator [7]. The generated number sequence is used to select the pixels where the secret information will be hidden.

III. PROPOSED METHOD

In the proposed steganography method, the secret is a rectangular image which is made up of only black and white pixels. The pixel values can be either 0 denoting black pixel or 255 denoting a white pixel. Alternately, the secret image can be a binary image having pixels with only 0 or 1 value. The cover image which is used is RGB color image having 3 different color channels. The size of the secret image has to be less than the size of the cover image for the proposed method to work.

The cover image is of dimension M x N where M is the number of rows and N is the number of columns. The secret image has a dimension of m x m, where m is the number of rows as well as the number of columns. The pixel information from the secret image is hidden in the cover image using LSB steganography. The secret image is read in the row major order. The pixels that are chosen in the cover image to hide the information are chosen based on pseudorandom number generation. This pixel selection is randomized so that it becomes difficult for an unauthorized person to reconstruct the secret image from the cover image.

The cover image is divided into N columns. Every column has M number of pixels corresponding to the M rows in the cover image. Out of these M pixels present in every column, only a certain number of pixels are chosen where the secret image bits are inserted. Before moving on to actually selecting pixels in a particular column and hiding information in them, the order in which the columns are selected are randomized. Initially, a single seed value is known called SeedR. This seed value is used to calculate SeedC.

$$\text{SeedC} = \lfloor (\text{SeedR} \times \text{m}/\text{M}) \bmod 1000 \rfloor \quad (1)$$

SeedC is used to generate the randomized column sequence ColSeq which gives the order in which the N columns are to be selected when hiding secret information.

Once ColSeq has been generated, a column is picked from the sequence and then another pseudorandom number generator selects the pixels in that column for hiding the secret information. Only a certain number of pixels in a single column are allowed to be changed. This number, called MaxP, is calculated by the following formula:
MaxP=size of the secret image/number of columns in the cover image

$$\text{MaxP} = \left\lceil \frac{\text{m}^2}{\text{N}} \right\rceil \quad (2)$$

Out of the M pixels present in a column, MaxP number of pixels are selected based on a seed. However, when generating the sequence of numbers MaxP+1 numbers are generated. This number sequence generated is also done using a PRNG. For the first column selected from ColSeq, the initial known seed SeedR is used. For every corresponding column this seed value is changed using the following equation (3)

$$\text{TempSeed} = \text{SeedR} \times \text{randNo} \times \text{i} \times \text{j} \quad (3)$$

Where randNo is the last additional number that was generated when generating the sequence for selecting MaxP number of pixels. (i,j) corresponds to the pixel in the secret image whose information is currently supposed to be hidden. The first four digits of Temp Seed forms the new seed.

Since every pseudorandom number generator is deterministic in nature, using the same seed for every run of the PRNG will lead to the same number sequence being generated. Selecting the same order of pixels for every column in the cover image is undesirable. Hence changing the seed value introduces randomness.

After MaxP pixels have been altered in a single column, the process is repeated for the next column as per ColSeq until all the secret image bits have been hidden in the cover image.

Algorithm for hiding the secret

1) Step 1: Use the known seed value SeedR to calculate SeedC using eq (1)
2) Step 2: Use PRNG with seed as SeedC to generate the sequence ColSeq.
3) Step 3: Calculate the value of MaxP using eq (2)
4) Step 4: Select the first unused column from ColSeq
5) Step 4.1: Use PRNG with the seed as SeedR to select MaxP pixels from the M pixels available in the column. These pixels are used one by one in the order they are generated to hide the secret image pixels.
6) Step 4.2: Select the pixel on the secret image to be hidden. The secret image is read in row major order.
7) Step 4.2.1: If the secret image pixel is white, set the LSB of the selected pixel (Red channel) on the cover image to 1.
8) Step 4.2.2: If the secret image pixel is black, set the LSB of the selected pixel (Red channel) on the cover image to 0.
9) Step 4.3: Once MaxP number of secret image information have been hidden, the value of seed is changed
10) Step 4.3.1: Calculate TempSeed using eq (3)
11) Step 4.3.2: Select the first four digits of TempSeed. This is the new value of SeedR.
12) Step 4.3.3: Set the seed of PRNG to SeedR
13) Step 5: Repeat Step 4 until all the secret image information has been hidden.

The hiding process has been shown in the following figure.

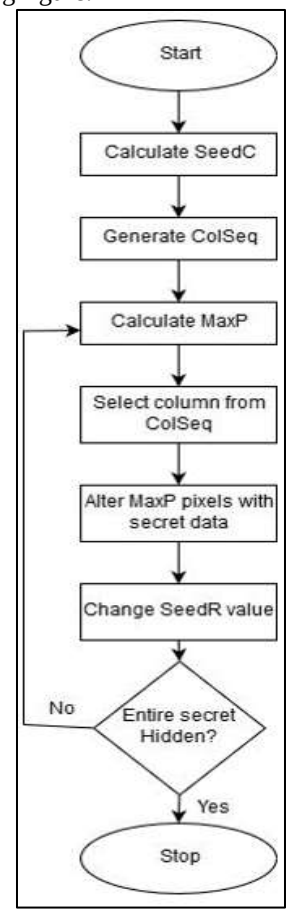

Fig. 1: Flowchart for the Hiding Process

At the receiver's side, the initial seed SeedR and the size of the secret image is known. For decryption, the process of pseudorandom number generation is repeated in the same way it was done for encryption using the exact same seed values. Because of deterministic nature of every PRNG, the same number sequence will be generated every time for the same seed value. Thus, the same pixels on the modified cover will be selected in the same order in which they were altered during the hiding process. Only the last bits of the selected pixels are checked. If the LSB is 1, then the corresponding secret pixel is white. If the LSB is 0, the corresponding secret pixel is black. The secret is reconstructed in row major method.

Algorithm for retrieving the secret

1) Step 1: Use the known seed value SeedR to calculate SeedC using eq (1)
2) Step 2: Use PRNG with seed as SeedC to generate the sequence ColSeq.
3) Step 3: Calculate the value of MaxP using eq (2)
4) Step 4: Select the first unused column from ColSeq.

5) Step 4.1: Use PRNG with the seed as SeedR to select MaxP pixels from the M pixels available in the column. These pixels are used one by one in the order they are generated to retrieve the hidden data
6) Step 4.2: Check the LSB of the selected pixels (Red Channel) as generated.
7) Step 4.2.1: If the LSB is 1, the secret pixel is white. The pixel is reconstructed.
8) Step 4.2.2: If the LSB is 0, the secret pixel is black. The pixel is reconstructed.
9) Step 4.3: Once MaxP number of secret image information have been retrieved, the value of seed is changed.
10) Step 4.3.1: Calculate TempSeed using eq (3)
11) Step 4.3.2: Select the first four digits of TempSeed. This is the new value of SeedR.
12) Step 4.3.3: Set the seed of the PRNG to SeedR
13) Step 5: Repeat Step 4 until the entire secret image has been reconstructed.

IV. EXPERIMENTAL RESULT

The proposed method has been run on MATLAB. The secret image used is a black and white image with dimension 128 x 128. Different RGB cover images have been used to show the result. The following figures show the implementation of the proposed algorithm using 'Baboon' as the cover image. The dimension for the cover image is 512 x 512. The fig 2 shows the secret image and fig 3 shows the cover image.

Fig. 2: Secret Image 'One'

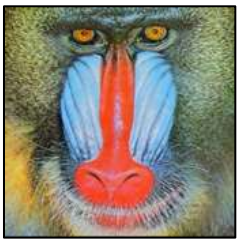

Fig. 3: Cover Image 'Baboon'

The secret image has been hidden in the cover image to produce the modified cover image as shown in fig 4.

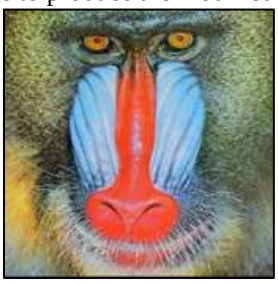

Fig. 4: Modified Cover Image

The receiver uses the retrieval process to recover the secret image as shown in fig 5.

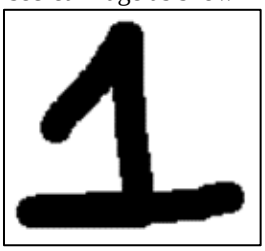

Fig. 5: Recovered Secret Image

Image quality metrics have been used to analyze the image quality after the applying the proposed algorithm [8]. The metrics have been applied on the cover images.

A. Mean Squared Error

Mean Squared Error (MSE) is calculated using the following formula as given by eq (4).

$$\mathrm{MSE} = \frac{1}{\mathrm{M}\times\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{M}}\sum_{\mathrm{j}=1}^{\mathrm{N}}[\mathrm{x}(\mathrm{i},\mathrm{j}) - \mathrm{y}(\mathrm{i},\mathrm{j})]^2 \quad (4)$$

Lower value of MSE is preferable.

B. Peak Signal-to-Noise Ratio

Peak Signal-to-Noise Ratio (PSNR) is used to determine the quality between two images. It is calculated using the formula given in eq (5).

$$\text{PSNR} = 20 \log_i \frac{\text{MAX}_i}{\sqrt{\text{MSE}}} \tag{5}$$

Where MAX_i is the maximum intensity value a pixel can have. MSE is calculated using eq (4). The higher the PSNR value the better is the image quality. This is because the denominator of eq (5) has MSE which is the variable for error. This means that when the error is low, the PSNR will be high.

C. Structural Similarity Index

Structural Similarity Index (SSIM) deals with the perceived value of the changes between two images. SSIM is calculated using eq (6).

$$\text{SSIM} = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c2)}{(\mu_x^2+\mu_y^2+c1)(\sigma_x^2+\sigma_y^2+c2)} \tag{6}$$

Where, x and y are two windows of size N x N on the two images being compared. The value of N is usually 8.
μ_x is the average of x;
μ_y is the average of y;
σ_x^2 is the variance of x;
σ_y^2 is the variance of y;
σ_{xy} is the covariance of x and y.
Higher values of SSIM means that the two images will look more similar and it will be harder to detect any changes between the images.

Table – 1
Quality Analysis using Various Metrics

Name	*Dimension*	*MSE*	*PSNR*	*SSIM*
Female	*256x256*	*0.0421*	*61.8912*	*0.9999*
Flowers	*362x500*	*0.0151*	*66.3280*	*1.0000*
Cablecar	*480x512*	*0.0111*	*67.6945*	*0.9999*
Baboon	*512x512*	*0.0104*	*67.9796*	*1.0000*
Lena	*512x512*	*0.0103*	*68.0112*	*1.0000*
Peppers	*512x512*	*0.0106*	*67.8966*	*1.0000*
Tulips	*512x768*	*0.0071*	*69.6190*	*1.0000*
Barbara	*576x720*	*0.0066*	*69.9616*	*1.0000*
Boat	*576x768*	*0.0061*	*70.2773*	*1.0000*
Goldhill	*576x720*	*0.0064*	*70.0367*	*1.0000*

Table I shows the different quality metric values for various cover images used. The MSE value is low. A low value in MSE indicates that the corresponding PSNR value will be high. This is evident from the PSNR values which has been experimentally found. High SSIM values indicate that the original cover image and the modified cover image are visually almost identical.

In [9], a PRNG has been used to generate a sequence that tells which bit on a pixel is to be altered. Only the Blue channel is affected. The second, third, or fourth least significant bit may be selected for any particular pixel for hiding of secret data bits. The first least significant bit is avoided. The least significant bit position to be used is given by the sequence which is generated using the PRNG.

In [10], a (2-1-2) layer selection scheme has been used. For the first selected pixel the Blue and Green channels are altered. For the next selected pixel, the Blue channel is altered. In the next iteration the Blue and Green channels are altered. The secret bits are embedded in the form of (3-2-3). A PRNG has been used to select the pixels.

Table – 2
Comparison Table

Cover Image		*PSNR*		
Name	*Dimension*	*Proposed Method*	*Method in [8]*	*Method in [9]*
Female	*256x256*	*61.8912*	*47.2030*	*57.3572*
Flowers	*362x500*	*66.3280*	*52.9184*	*61.5905*
Cablecar	*480x512*	*67.6945*	*54.4509*	*63.0454*
Baboon	*512x512*	*67.9796*	*53.5623*	*63.4699*
Lena	*512x512*	*68.0112*	*53.5519*	*63.4355*
Peppers	*512x512*	*67.8966*	*53.5011*	*62.9974*
Tulips	*512x768*	*69.6190*	*55.0983*	*65.1203*
Barbara	*576x720*	*69.9616*	*55.4157*	*65.3310*
Boat	*576x768*	*70.2773*	*55.6902*	*65.4071*
Goldhill	*576x720*	*70.0367*	*55.6704*	*65.2956*

The data Table II shows that the proposed method results in higher PSNR values than the other methods. This indicates that the proposed method performs better in terms of embedding secret bits in the cover image.

V. CONCLUSION

In the proposed method only the Least Significant Bit of certain selected pixels are being modified. Thus, difference between the original cover image and the modified cover image cannot be perceived by the human eye. The randomness induced by using the Pseudorandom Number Generator ensures that even if the attacker identifies that the cover image being transmitted has some secret data hidden in it the retrieval of the same would be very difficult. Without prior knowledge of the initial seed, the size of the secret image, or the equations by which the seed value is changed, the decryption process cannot be completed. The proposed method ensures that the secret information is dispersed uniformly amidst the various pixels of the cover image and there is no region with considerably higher or lower concentration of modified pixels compared to any other region. The cover image can also be grayscale where there is only one channel. In the proposed method the secret image is completely retrieved with no loss in image quality. The PSNR between the original cover image and the modified cover image is also high indicating that the hiding is successful and unnoticeable. Other forms of secret data like text can also be hidden. For that the secret data stream needs to be converted into the 8-bit binary representation of the corresponding ASCII values. Grayscale secret images can also be used with pixel values represented in the 8-bit binary format.

REFERENCES

[1] Prof. Mukund R. Joshi, Renuka Avinash Karkade, "Network Security with Cryptography", IJCSMC, Vol. 4, Issue. 1, January 2015, pg. 201—204
[2] Jasleen Kour, Deepankar Verma, "Steganography Techniques –A Review Paper", International Journal of Emerging Research in Management &Technology, Vol. 3, Issue. 5, pg. 132—135
[3] Sumeet Kaur, Savina Bansal, R. K. Bansal, "Steganography and Classification of Image Steganography Techniques", 2014 International Conference on Computing for Sustainable Global Development (INDIACom), pg. 870—875
[4] R. Chandramouli and N. Memon, "Analysis of LSB based Image SteganographyTechniques," Proceedings of International Conference on Image Processing, vol. 3, October 2001, pg. 1019—1022
[5] Champakamala B.S, Padmini.K, Radhika D. K, "Least Significant Bit algorithm for image steganography", International Journal of Advance Computer Technology, Volume 3, Number 4, pg. 34—38
[6] E. Walia, P. Jain, Navdeep, "An Analysis of LSB & DCT based Steganography", Global Journal of Computer Science and Technology, Vol. 10 Issue 1, April 2010, pg 4-8.
[7] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator", Jan. 199 ACM Trans. on Modeling and Computer Simulation, vol. 8, no. 1, Jan 1998, pg. 3—30
[8] Kuryati Kipli, Shankar Krishnan, Nurdiani Zamhari et al, "Full Reference Image Quality Metrics and their Performance", 2011 IEEE 7th International Colloquium on Signal Processing and its Applications, pg. 33—38
[9] Unik Lokhande, A. K. Gulve, "Steganography using Cryptography and Pseudo Randomm Numbers", International Journal of Computer Applications, Vol 96, No 19, June 2014, pg. 40—45
[10] Marwa M. Emam, Abdelmgeid A. Aly, Fatma A. Omara, "An improved image steganography method based on LSB technique with Random Pixel Selection", International Journal of Advanced Computer Science and Applications (IJACSA), Vol 7, No 3, 2016, pg. 361-366