Indian Journal of Science and Technology, *Vol 10(38), DOI: 10.17485/ijst/2017/v10i38/87014,* October *2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A New Method of Image Steganography using Last Three Bit Plane of Gray Scale Images

Kamaldeep Joshi* and Rajkumar Yadav

Maharshi Dayanand University, Rohtak – 124001, Haryana, India; kamalmintwal@gmail.com, rajyadav76@rediffmail.com

Abstract

Objectives: The objective of this technique is to maximize the capacity and imperceptibility of the secret data with minimal modification to the cover image. **Methods:** In this paper, a new image steganography method is proposed that hides secret data in a selected carrier image. The proposed steganography method primarily extracts the last three bit of the selected pixel and then performs the XOR operation with 3rd, 1st and 3rd, 2nd bit. On the basis of result of XOR operation every bit of secret data is embedded one by one on LSB of the selected pixel value. **Findings:** This method comes under +1 or -1 change in pixel value of a Stego image. Due to +1 or -1 modification, the method shows minimum degradation in the Stego image. As each pixel can carry data bit, the chance of message hidden is 100%. Experimental results indicate that our method achieves good image quality and fairly large data hiding capacity. For experimental results, four standard images were taken and analyzed on the basis of different parameters such as PSNR, MSE, Mean and Standard Deviation, Number of pixels used and Number of pixels changed etc. The experimental results show that our method gives good imperceptibility as degradation is only +1 and -1 in a pixel. The proposed method was better when compared to LSB and chaotic method. **Novelty/Improvement:** The chance of message hidden is 100%. The proposed method improves the results to achieve good capacity and imperceptibility.

Keywords: Mean, Standard Deviation, LSB, MSE, PSNR, XOR

1. Introduction

Steganography is an art and science of hiding the existence of the data so that it becomes invisible to someone else[1]. The aim of steganography is to protect our data from unauthorized access. In this technique, the data may be hidden in a carrier file. The file may be an image, audio and video file[2]. In case of an image steganography, the data is hidden in a Carrier image[3].

Let I be the image in which data is to be hidden and S is the data to be hidden using an insertion algorithm. S and I become I', which is equal to I+S. I' is transferred over an open channel. At the receiver end, the extraction algorithm separates data as well as image. The data is used for some purpose and the image may be discarded.

Steganography can be applied on number of application such as copyright protection, enhancing robustness of an image, in smart cards to hide users' data in an image, in medical imaging system to keep the record of the patients, in audio-video synchronization, in watermarking, to hiding the logos or information. In military security, in secure communication. In TV broadcasting, in temper proofing, in feature tagging to keep features such as name, age, color etc[4,5].

There are many image steganography approach exists with their own advantages and limitations. LSB is considered as most studied and simple method in the literature. In this method, the least significant bit of a pixel is replaced with data bit[6,7]. For example if the letter *k* is to be hidden using LSB method, then convert its decimal or

**Author for correspondence*

ASCII value into binary form i.e. 01101011. These eight bits are hidden on eight LSBs of the selected or consequent pixels. The limitation of this method is that by only picking the consecutive eight LSBs the intruder can access the data[8]. After that the modification was made where the data was put at random locations. These locations were generated by PRNG. The PRNG generates eight random pixels and the data was put at these pixels on their LSBs. But the limitation with this method was that the sender and receiver must agree on a same key or the key needs to be sent from sender to receiver[9,8] proposed an algorithm which hides the data at 1st and 2nd bit plane. It overcomes the disadvantage of LSB methods but the message carrying capacity of this message was quite low. The chance of message insertion at PRNG at first chance was 50%. 50% chance, when message bit is inserted when no change in pixel value required and 12.5% chances, when the change was required. Hence, it provides low message capacity. It also comes under the category of +1 and -1 in Stego image[8]. Proposed an algorithm which involves 6th, 7th and 8th bit of pixel value. In this method, 6th, 7th, and 8th bit of the pixel was used with a time factor t_1. It showed good results than previous methods. It enhances the chance of insertion at first chance to 85.93%[10]. PVD hides the number of bit on a pixel based upon pixel value difference from the selected and previous pixel[11]. The change, in this method exceeds from +1 And +1. In[12] the GLM method was proposed which works on the gray values of an image. This method hides the data by modification in gray values of an image. Gutub presented a key less image steganography in RGB images using LSB method. This method was tested for last two LSBs[13]. Hence the change in pixel value is +3 or -3. Rajkumar et al. proposed another method based on parity i.e. if the parity of a pixel is odd then hide 0 otherwise hide 1. The data hiding capacity of this method is 98.82%[14]. The change in Stego image is +1 and -1 but the message capacity is low. Also this method was easy to break by hit and trail method only one gauss is required i.e. whether the value of a pixel has even or odd.

The rest of this paper is planned as follows. Section 2 describes the projected method. Section 3 shows the experimental results. Finally, Section 4 presents the conclusion of the paper.

2. Proposed Method

In this paper, a new image steganography technique is proposed by performing XOR operation on the bits of pixel values. In our technique the data is hidden on LSB of the pixel values. In this technique three rightmost LSBs of the pixel value are extracted and XOR operation on 1st, 3th and 2nd, 3th bits is performed. If the result of both XOR operations is 00 or 11 then message bit 0 is hidden in the LSB of pixel value and if the result of XOR is 01 or 10 then message bit 1 is hidden at the LSB of the selected pixel. Assume the length of the message is known to sender as well as receiver. The flowcharts of the inclusion and retrieval of message are given by the Figures 1 and 2 respectively.

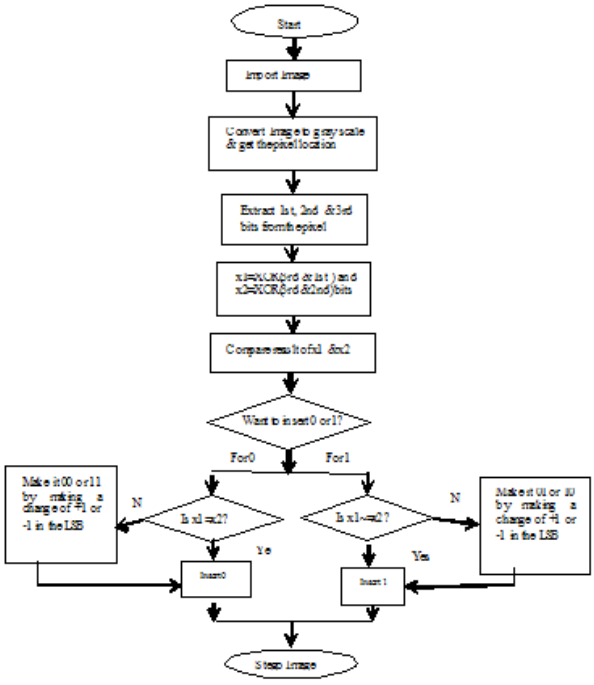

Figure 1. Flow chart for Insertion of message.

Let I be the cover image of R*C pixels and S be the N-bit secret message and x be the pixel value of I and s be the bit of secret message, then

$$I = \{x_{ij} \mid 1<=i<=R,\ 1<=j<=C,\ x_{ij} \in \{0,1,\ldots\ldots 255\}\} \quad (1)$$

$$S = \{s_N \mid 1<=N<=n,\ s_N \in \{0,1\}\} \quad (2)$$

The Stego image can be represented by I' which is given by eq.3 by applying the algorithm given in section 2.1.

$$I' = I + S \quad (3)$$

At the other end the reverse process is carried out and the message is extracted using the algorithm in section

2.2. The message is separated from the cover image by the eq. 4.

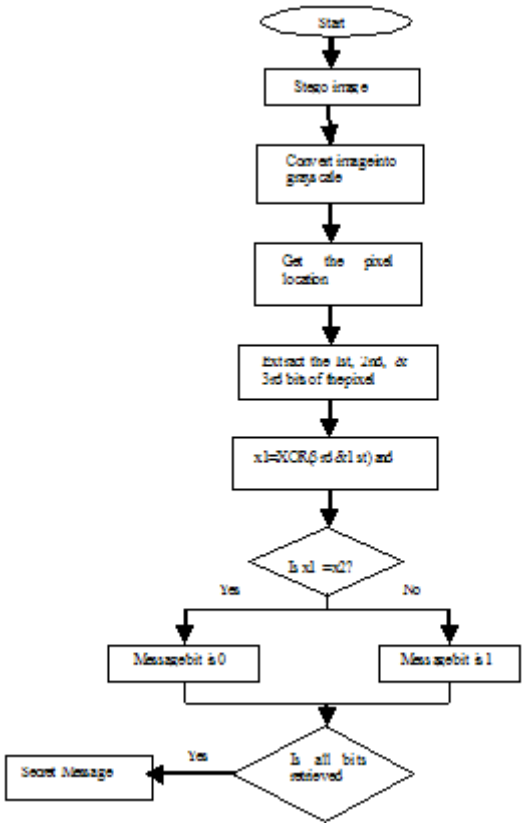

Figure 2. Flow chart for Retrieval of message.

$$S=I'- I \quad (4)$$

The data can be inserted by the method given as follows:

2.1 Pseudo-code for Insertion of the Message at Sender's End

1. clc;
2. clear all; % Clears all existing variables.
3. clear; % Removes all variables from the workspace, releasing them from system memory.
4. close all; % Deletes all figures whose handles are not hidden.
5. a1=input('enter the message'); %Enter Message
6. N=length(a)*8; %Message Length in Bits
7.save N;
8. b=decimal_to Bin(a)); % Concerts decimal to binary
9. len=length(a); % Message Length in bits
10. save len;
11. S=b(i,j); % Message in bits in a 1*N size matrix.
12. I=imread('baboon.tif'); % Read a gray image in a variable I;
13. [r,c]=size(I); % finds the size of I
14. x=uint8(zeros(r,c)); %Initialize a temporary matrix x to zero which is equal to image size
15. a=1;
16. t=0;
17. count=0; % A counter counts the number of pixel changed
18. tic; % tic tac records the time used in execution
19. for l=1:r
20. for m=1:c
21. a1=bitget(I(l,m),1); %Finds 1st LSB
22. a2=bitget(I(l,m),2); %Finds 2nd LSB
23. a3=bitget(I(l,m),3); %Finds 3rd LSB
24. x1=XOR(a3,a1);
%Performs XOR operation between a1 and a2;
25. x2=XOR(a3,a2);
26. if(a<=N) %Compare index 'a' and Message length;
27. m1=S(a); %assign the value of S in m1
28. if (x1==x2 && m1==0)
%Checks if xi1 and x2 are equal and message bit is 0 then no change in I and copied into x i.e. temporary variable.
    a. x(l,m)=I(l,m);
29. end
30. if (x1~=x2) && (m1==0 && a1==0)
    a. x(l,m)=I(l,m)+1;
    b. count=count+1;
31. end
32. if (x1~=x2) && (m1==0 && a1==1)
    a. x(l,m)=I(l,m)-1;
    b. count=count+1;
33. end
34. if (x1==x2) && (m1==1 && a1==0)

```
    a. x(l,m)=I(l,m)+1;
    b. count=count+1;
35. end
36. if (x1==x2) && (m1==1 && a1==1)
    a. x(l,m)=I(l,m)-1;
    b. count=count+1;
37. end
38. if (x1~=x2) && (m1==1)
    a. x(l,m)=I(l,m);
39. end
40. else
    a. x(l,m)=I(l,m);
41. end
42. a=a+1;
43. end
44. end
45. toc;
46. imwrite(uint8(x),'stego.bmp'); %Writing x into an image called as Stego image
```

The extraction process of the proposed method is given as follows:

2.2 Pseudo-code for Extraction of the Message at Receiver's End

```
1. clc;
2. clear all;
3. close all;
4. clear;
5. c1 = imread('stego.bmp');
6. [m,n] = size(c1);
7. load N;
8. K = 1;
9. v=double(zeros(1,8));
10. tic;
11. for i = 1:m
12. for j = 1:n
13. b1 = bitget(c1(i,j),1);
14. b2 = bitget(c1(i,j),2);
15. b3 = bitget(c1(i,j),3);
16. x1 =xor(b3,b1);
17. x2 =xor(b3,b2);
18. if K<=N
19. if (x1==x2)
20. v(K) = 0;                    %V is a temporary matrix stores message bit extracted from Stego image
21. end
22.
23. if (x1~=x2)
24. v(K) = 1;
25. end
26.
27. K = K+1;
28. end
29. end
30. end
31. toc;
32. v;                           %Final message matrix.
33. v=reshape(v,8,[]);           %Reshape final matrix into N*8.
34. load len;
35. message=convert v in decimal value
36.
37. MESSAGE=char(Message);       %Convert Message into readable form i.e. character
38. disp(MESSAGE);               %Displays message
```

3. Experimental Results

The proposed method had been tested on four Images of size 512x512 including "Lena", "Baboon", "Peppers" and "Tulips". Various parameters were used for measuring the performance of the proposed method for example PSNR, MSE, Insertion and retrieval time, percentage of pixels used and percentage of pixels. The method is evaluated on three different message sizes say 1024 bits, 2048 bits and 4096 bits. We have used MATLAB 7.11.0 to run programs with a core i7 CPU 3.40 GHz, 4 GB Memory for the following experiments.

3.1 PSNR

PSNR stands for Peak Signal to Noise Ratio. It measures the similarity between two images. So the PSNR must be high for a good method or we can say that the better the PSNR value the better is the method. The PSNR of any image can be calculated by the following eq.[15]:

$$MSE = \frac{1}{R \times C} \sum_{i=1}^{R} \sum_{j=1}^{C} (x_{ij} - x'_{ij})^2 \quad (5)$$

where,

D is the maximum value that a pixel can have, for 8-bit images: D=255.

MSE is the mean square error.

3.2 MSE

MSE stands for Mean Square Error. It shows the difference between two images. The MSE of the images should be low for a good technique. The MSE can be calculated using the eq. given[16]:

$$PSNR = 10\log_{10}[\frac{D^2}{MSE}] \quad (6)$$

where, R and C are the number of rows, columns in the cover image. x_{ij} is the intensity of Pixel ij in cover image. x'_{ij} is the intensity of Pixel ij in the Stego image.

Figure 3 shows the Stego and original image of Lena. Figures 4 – 6 show the histograms of Stego image having 1024, 2048 and 4096 bit of data respectively. Figure 7 indicates Difference between Original and Stego image for message size 1024, 2048 and 4096 bits respectively. Table 1 gives the Results of the proposed method using various parameters for Lena image.

Figure 3. Original and Stego image.

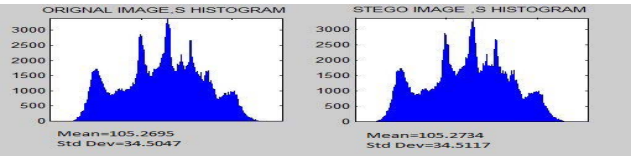

Figure 4. Histogram for message size 1024 bits.

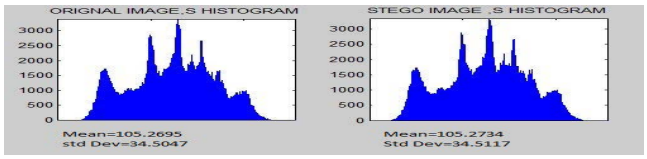

Figure 5. Histogram for message size 2048 bits.

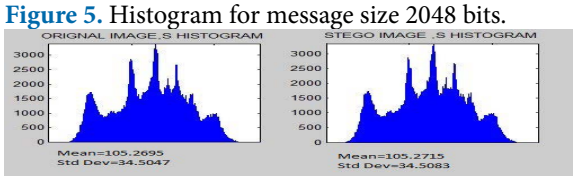

Figure 6. Histogram for message size 4096 bits.

Figure 7. Difference between Original and Stego image for message size 1024, 2048 and 4096 bits respectively.

Figure 8 shows the Stego and original image of Baboon. Figures 9 – 11 show the histograms of Stego image having 1024, 2048 and 4096 bit of data respectively. Figure 12 indicate Difference between Original and Stego image for message size 1024, 2048 and 4096 bits respectively. Table 2 gives the Results of the proposed method using various parameters for Baboon image.

Figure 13 shows the Stego and original image of Peppers. Figures 14–16 show the histograms of Stego

Table 1. Results of the proposed method using various parameters

Image Size	Message Size	% of Pixels Used	% of Pixels Changed	PSNR	MSE	Insertion Time (in Seconds)	Retrieval Time (in Seconds)
512*512	1024bits	0.39	0.1892	75.3614	0.0019	0.838382	0.346117
512*512	2048bits	0.78	0.3796	72.3380	0.0038	0.905275	0.447872
512*512	4096bits	1.62	0.7736	69.2455	0.0077	0.913534	0.467697

Table 2. Results of the proposed method using various parameters

Image Size	Message Size	% of Pixels Used	% of Pixels Changed	PSNR	MSE	Insertion Time (in Seconds)	Retrieval Time(in Seconds)
512*512	1024	0.39	0.1877	75.5493	0.0018	0.854624	0.419129
512*512	2048	0.78	0.3899	72.2217	0.0039	0.887008	0.487535
512*512	4096	1.62	0.7090	69.1544	0.0079	0.924411	0.548283

image having 1024, 2048 and 4096 bit of data respectively. Figure 17 indicate Difference between Original and Stego image for message size 1024, 2048 and 4096 bits respectively. Table 3 gives the Results of the proposed method using various parameters for Peppers image.

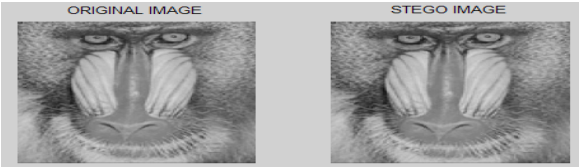

Figure 8. Original and Stego Image

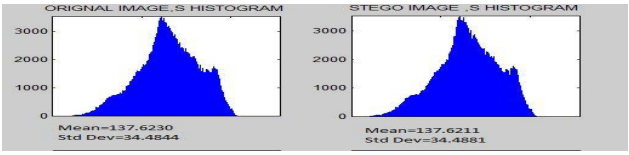

Figure 9. Histogram for message size 1024 bits.

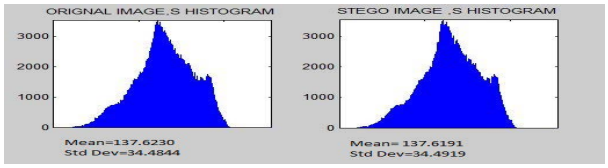

Figure 10. Histogram for message size 2048 bits.

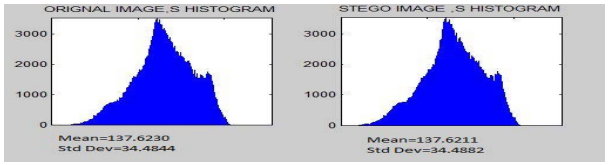

Figure 11. Histogram for message size 4096 bits.

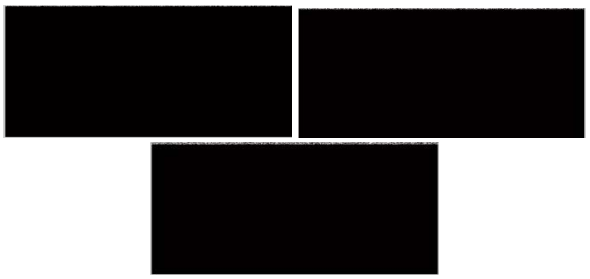

Figure 12. Difference between Original and Stego image for message size 1024, 2048 and 4096 bits respectively.

Figure 13. Original and Stego image.

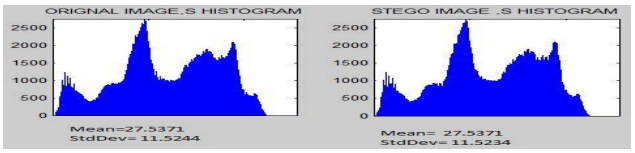

Figure 14. Histogram for message size 1024 bits.

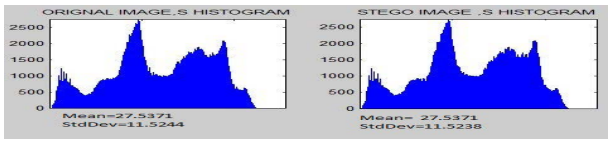

Figure 15. Histogram for message size 2048 bits.

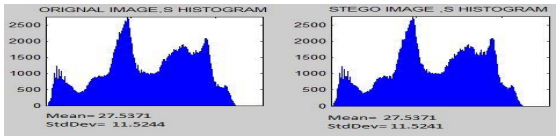

Figure 16. Histogram for message size 4096 bits.

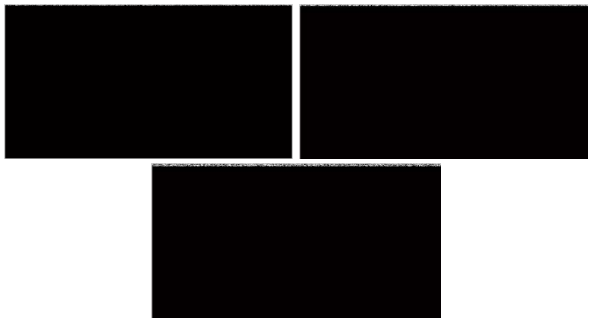

Figure 17. Difference between Original and stego image for message size 1024, 2048 and 4096 bits respectively.

Figure 18 shows the Stego and original image of Triffy. Figures 19 – 21 show the histograms of Stego image having 1024, 2048 and 4096 bit of data respectively. Figure 22 indicate difference between Original and Stego image for message size 1024, 2048 and 4096 bits respectively. Table

Table 3. Results of the proposed method using various parameters

Image Size	Message Size	% of Pixels Used	% of Pixels Changed	PSNR	MSE	Insertion Time (in Seconds)	Retrieval Time (in Seconds)
512*512	1024	0.3906	0.1987	75.1478	0.0020	0.754332	0.378467
512*512	2048	0.7813	0.4036	72.0713	0.0040	0.786992	0.405213
512*512	4096	0.7987	0.7986	69.1064	0.0080	0.799362	0.421158

4 gives the Results of the proposed method using various parameters for Triffy image.

Figure 18. Original and Stego image.

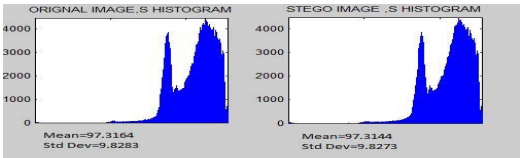

Figure 19. Histogram for message size 1024 bits.

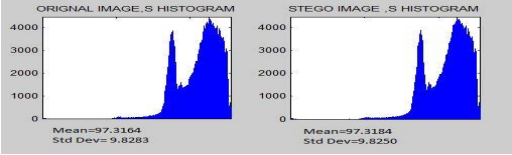

Figure 20. Histogram for message size 2048 bits.

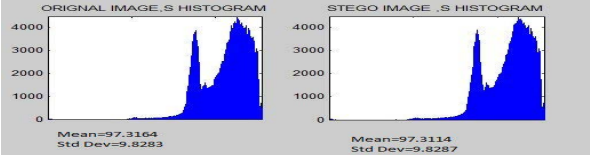

Figure 21. Histogram for message size 4096 bits.

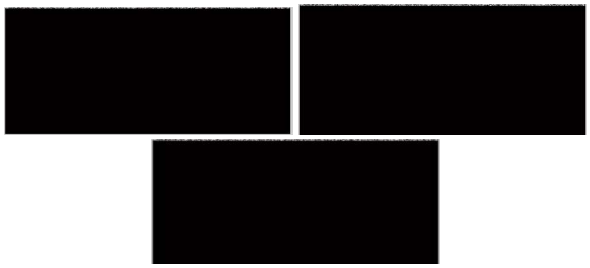

Figure 22. Difference between Original and Stego image for message size 1024, 2048 and 4096 bits respectively.

Table 5 provides the comparative numerical values of the dimension of the images, and Peak Signal to Noise Ratio (PSNR) analysis for the methods in [17] and our proposed method for two images, Lena & Baboon. As we have obtained more PSNR value, so it is obvious to get higher MSE value.

4. Conclusion

We have proposed a new data hiding method based on XOR. For image of Lena of our method gives the PSNR 75.3614db and the percentage of pixels used is 0.39%, Insertion and Retrieval time is 0.838382, 0.346117 respectively. For image of baboon gives the PSNR 75.5493db, Insertion and Retrieval time is 0.854624, 0.419129 respectively. For image of Peppers gives the PSNR 75.1478db,

Table 4. Results of the proposed method using various parameters

Image Size	Message Size	% of Pixels Used	% of Pixels Changed	PSNR	MSE	Insertion Time (in Seconds)	Retrieval Time (in Seconds)
512*512	1024	0.3906	0.1938	75.2576	0.0019	0.758008	0.366969
512*512	2048	0.7813	0.3914	72.2047	0.0039	0.778735	0.370899
512*512	4096	1.5625	0.7874	69.1691	0.0079	0.792470	0.383019

Table 5. Comparison of Existing schemes with proposed Scheme

Technique	Image	Image Size	PSNR on Message= 1024 bits	PSNR on Message= 2048 bits	PSNR on Message= 4096 bits
LSB	Lena	512*512	53.6442	50.1237	42.5700
LSB	Baboon	512*512	53.3499	50.2320	47.2027
Chaotic Approach	Lena	512*512	38.6432	35.5419	32.5876
Chaotic Approach	Baboon	512*512	34.1444	31.3183	28.2366
Proposed Method	Lena	512*512	75.3614	72.3380	69.2455
Proposed Method	Baboon	512*512	75.5493	72.2217	69.1544

Insertion and Retrieval time is 0.754332, 0.378467 respectively. For image of Triffy gives the PSNR 75.2576db, Insertion and Retrieval time is 0.758008, 0.366969 respectively. In the proposed method, number of bit be hidden is equal to the no of pixel in an image. The experimental results show that our method gives good imperceptibility as degradation is only +1 and -1 in a pixel. The proposed method was compared with LSB and chaotic method, which shows good results.

5. References

1. Vidya G, Hema Preetha R, Shilpa GS, Kalpana V. Image steganography using Ken Ken Puzzle for Secure Data Hiding. Indian Journal of Science and Technology. 2014 Jan; 7(9):1410–13.
2. Ramalingam M, Isa NAM. A steganography approach over video images to improve security. Indian Journal of Science and Technology. 2015 Jan; 8(1):79–86. Crossref
3. Jambhekar ND, Dhawale CA. Bit level key agreement & exchange protocol for digital image steganography. Indian Journal of Science and Technology. 2015 Jul; 8(15):1–7. Crossref
4. Taylor P. Multimedia digital rights protection using watermarking techniques; 2015 Jun. p. 37–41.
5. Cheddad A, Condell J, Curran K, Mc Kevitt MC. Digital image steganography: Survey and analysis of current methods. Signal Processing. 2010; 190(3):727–52. Crossref
6. Johnson NF, Jajodia S. Exploring steganography: Seeing the unseen. Computer. 1998; 31(2):26–34. Crossref
7. Chan CK, Cheng LM. Hiding data in images by simple LSB substitution. Pattern Recognition. 2004; 37(3):469–74. Crossref
8. Parvinder S, Sudhir B, Sharma HR. Evaluating the performance of message hidden in first and second bit plane. WSEAS Transaction on Information Science and Technology. 2005; 2(8):1220–22.
9. Franz E, Jerichow A, Moller S, Pfitzmsnn A, Stierand I. Computer based steganography: How it works and why therefore restrictions on cryptography are nonsense, at best, information hiding, Springer-Verlag: Berlin Heidelberg; 1996. p. 7–21.
10. Batra S., Rishi R. Insertion of message in 6th, 7th and 8th bit of pixel values and its retrieval in case intruder changes the least significant bit of image pixels. 2010; 4(3):1–10.
11. Wang CM, Wu NI, Tsai CS, Hwang MS. A high quality steganographic method with pixel-value differencing and modulus function. Journal of Systems Software. 2008; 81(1):150–8. Crossref
12. Potdar V, Chang E. Gray level modification steganography for secret communication. IEEE International Conference on Industrial Informatics, Berlin, Germany, INDIN'04, Berlin; 2004. p. 223–8.
13. Gutub AAA. Pixel indicator technique for RGB image steganography. Journal of Emerging Technology Web Intelligence. 2010; 2(1):56–64. Crossref
14. Rajkumar Y, Rishi R, Batra S. A new steganography method for gray level images using parity checker. International Journal of Computer Applications. 2010; 11(11):1–7. Crossref
15. Hajizadeh H, Ayatollahi A, Mirzakuchaki S. A new high capacity and EMD-based image steganography scheme in spatial domain. 2013 21st Iranian Conference on Electrical Engineering, Mashhad, ICEE'13; 2013. p. 1–6.
16. Gangwar A, Shrivastava V. Improved RGB-LSB steganography using secret key. International Journal of Computer Trends and Technology. 2013; 4(2):85–9.
17. Aziz M, Tayarani NMH, Afsar M. A cycling chaos-based cryptic-free algorithm for image steganography. Nonlinear Dynamics. 2015; 80(3):1271–90. Crossref