CYCLIC COMBINATION METHOD FOR DIGITAL IMAGE STEGANOGRAPHY WITH UNIFORM DISTRIBUTION OF MESSAGE

Rajkumar Yadav[1], Ravi Saini[2] and Kamaldeep[3]

[1]U.I.E.T, Maharshi Dayanand University, Rohtak-124001, Haryana, India
rajyadav76@rediffmail.com
[2]U.I.E.T, Maharshi Dayanand University, Rohtak-124001, Haryana, India
ravisaini1988@rediffmail.com
[3]U.I.E.T, Maharshi Dayanand University, Rohtak-124001, Haryana, India
kamalmintwal@gmail.com

ABSTRACT

In this paper, a new image steganography technique for embedding messages into Gray Level Images is proposed. This new technique distributes the message uniformly throughout the image. The image is divided into blocks of equal sizes and the message is then embedded into the central pixel of the block using cyclic combination of 6th, 7th & 8th bit. The blocks of the image are chosen randomly using the Pseudo Random Generator seeded with a secret key. In proposed method, cyclic combination of last three bits of pixel value provide 100% chances of message insertion at the pixel value and division of image into blocks distribute the message uniformly into the image. This method also provides minimum degradation in image quality that cannot be perceived by human eye.

KEYWORDS

LSB Method, Cryptography, Steganography, Pseudo Random Number Generator

1. INTRODUCTION

In recent years, everyone is moving towards digital world. With the rapid development of the internet technologies, digital media needs to be transmitted conveniently over the network. Attacks, unauthorized access of the information over the network become greater issues now days. Cryptography and Steganography are the solutions to these security related issues. Steganography is an art and science of hiding the data in some cover media. In Greek, steganography means “covered writing” [1]. Steganography is different from Cryptography which is about concealing the content of message whereas Steganography is about concealing the existence of message itself [2].

Steganography techniques uses different media like image files, audio files, video files and text files for secret communications. Depending upon the cover media we can classify the steganography into many parts:

- **Text Steganography**
- **Image Steganography**
- **Audio Steganography**
- **Video Steganography**

DOI : 10.5121/acij.2011.2604

There are many parameters that affect steganography techniques. These parameters include hiding capacity, perceptual transparency (or security), robustness, complexity, survivability, capability and detectability [3, 4, 5, 6].

- **Hiding Capacity**

Hiding capacity is the size of information that can be hidden relative to the size of the cover. A larger hiding capacity allows the use of a smaller cover for a message of fixed size, and thus decreases the bandwidth required to transmit the stego-image.

- **Perceptual Transparency**

The act of hiding the message in the cover necessitates some noise modulation or distortion of the cover image. It is important that the embedding occur without significant degradation or loss of perceptual quality of the cover. In a secret communications application, if an attacker notices some distortion that arouses suspicion of the presence of hidden data in a stego-image, the steganographic encoding has failed even if the attacker is unable to extract the message. Preserving perceptual transparency in an embedded watermark for copyright protection is also of paramount importance because the integrity of the original work must be maintained.

- **Robustness**

Robustness refers to the ability of embedded data to remain intact if the stego-image undergoes transformations, such as linear and non-linear filtering, addition of random noise, sharpening or blurring, scaling and rotations, cropping or decimation, lossy compression, and conversion back to digital form (such as in the case when a hard copy of a stego-image is printed and then a digital image is formed by subsequently scanning the hardcopy.)

- **Tamper Resistance**

Beyond robustness to destruction, tamper-resistance refers to the difficulty for an attacker to alter or forge a message once it has been embedded in a stego-image, such as a pirate replacing a copyright mark with one claiming legal ownership. In a copyright protection application, achieving good tamper resistance can be difficult because a copyright is effective for many years and a watermark must remain resistant to tampering even when a pirate attempts to modify it using computing technology decades in the future.

- **Other Characteristics**

Computational complexity of encoding and decoding is another consideration and individual applications may have additional requirements. For example, for a copyright protection application, a watermark should be resistant to collusion attacks where many pirates work together to identify and destroy the mark.

In this present study, first the image is divided into blocks of equal length. After that the message is hidden in the central pixel of the selected block by using cyclic combinations of last three bits. Our technique distribute the message uniformly throughput the image and is more immune to noise imperfections and steganalysis attacks.

The rest of the paper is organized as follows:

Section 2 reviews various methods of image steganography. Section 3 consists our proposed method i.e. CCM. Section 4 shows how pixel values changes during insertion of message. In Section 5, some experimental results and analysis is shown. Section 6 provides conclusion of our work and also gives some attention towards future work.

2. Methods of Image Steganography

2.1 LSB Method [7]

In this method, least significant bit of pixel value is used for insertion of message. This method is easy to implement but it has many disadvantages associated with it.

- Message can be easily recovered by the unauthorized person as message is in LSB.

- As message is hidden in LSB, so intruder can modify the LSB of all the image pixels in the way the hidden message can be destroyed.

- LSB is most vulnerable to hardware imperfections or quantization of noise.

2.2 6th & 7th Bit Method [8]

In this method, Parvinder et al used the 6th & 7th bit for the insertion of message. They didn't use any LSB. They overcome the disadvantages associated with LSB method. But this method also has its own disadvantage. The main disadvantage associated with it is that this method provides only the 50% chances of message insertion at a pixel value.

2.3 PVD (Pixel Value Differencing Method) [9]

The pixel value differencing (PVD) method proposed by Wu and Tsai can successfully provide both high embedding capacity and outstanding imperceptibility for the stego-image. The pixel value differencing (PVD) method segments the cover image into non overlapping blocks containing two connecting pixels and modifies the pixel difference in each block (pair) for data embedding. A larger difference in the original pixel values allows a greater modification.

2.4 Cover Region and Parity Bits Method [10]

In this technique, the image is divided in a minimum of L(m) contiguous and disjoint regions and their use are defined by a pseudo-random number generator (PRNG).

$$P(I) = \sum_{j \in i} LSB(C_j) \bmod_{2.} ---------- (1)$$

It is necessary only one LSB flipping of any pixel of the region to change the parity region value.

3. Description of Proposed Method

In this method, the message is uniformly distributed throughout the image. For this purpose, first the image is divided into blocks of equal size. Size of each block depends upon the size of image and length of the message. After that, the central pixel of selected block is calculated. The block is selected using Pseudo Random Number Generator which is seeded with a secret key. Now, the message bit is inserted at the central pixel band based upon cyclic combination of last three bits. Cyclic combinations of last three bits are used separately for insertion of 0 & 1 in the following manner (given in Figure 1).

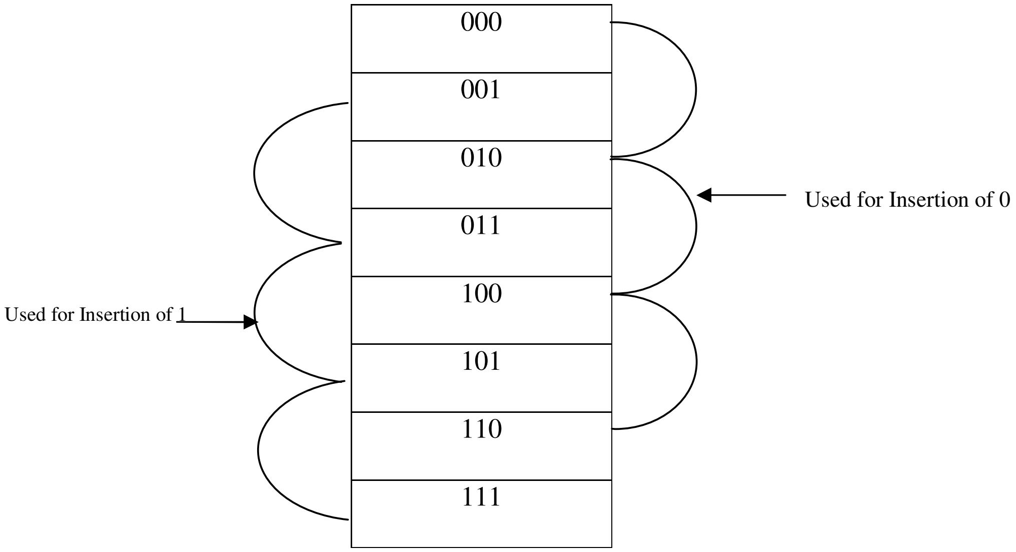

Figure 1. Cyclic combinations of last three bits

The combinations 000, 010, 100, 110 are used for insertion of 0 and 001, 011, 101, 111 are used for insertion of 1. If corresponding combination does not exist for insertion of a particular bit then we make corresponding combination by adding or subtracting 1 to the pixel value.

3.1 Hypothesis and Assertions

Hypothesis-1

In digital image, small variations in pixel value are imperceptible to human eye. Our hypothesis is that changing +1 or -1 unit in the pixel value is imperceptible to human visual system (HVS).

Hyptothesis-2

The length of each block depends on size of image and length of the message and in each block one message it is inserted.

Assertion-1

The cyclic combinations of last three bits are chosen for insertion of message because it satisfies hypothesis-1 and provides minimum change in pixel value i.e. +1 or -1.

Assertion-2

According to hypothesis-2, uniform distribution of the message bits in image is guaranteed.

Assertion-3

Length of message is known to both sender and receiver.

3.2 Insertion Algorithm

i) Compute the blocking factor (BF) using the cover image size in pixels i.e. I(p) and the message length L(m) in bits:

$$BF = Abs\left[\frac{I(P)}{L(M)}\right].-------------(2)$$

ii) The image is divided in at least L(m) blocks of size BF. They are disjoint and continuous, each one of them is used to store only one bit of message.

iii) The block for insertion of message bit is chosen by using Pseudo-Random Number Generator which uses a secret key that is shared between sender & receiver.

iv) With the block i indicated by PRNG, we calculate its central pixel C(i):

$$C(i) = Abs\left[\frac{B(F)\times(2i-1)+1}{2}\right].------(3)$$

v) If want to insert 0 then go to step (vi) else go to step (vii).

vi) a) If the combination of last three bits of C(i) have value 000, 010, 100 or 110, then insert 0 at C(i) and go to END. (In this case no change in pixel value is required)

b) If the combination of last three bits of C(i) have value 001, 011, 101 or 111, then make these combinations equal to 000, 010, 100 or 110 by adding or subtracting 1 to pixel value C(i), insert 0 at C(i) and go to END. (In this case +1 or -1 change in pixel value is required)

vii) a) If the combination of last three bits of C(i) have value 001, 011, 101 or 111, then insert 1 at C(i) and go to END. (In this case no change in pixel value is required)

b) If the combination of last three bits of C(i) have value 000, 010, 100 or 110, then make these combinations equal to 001, 011, 101 or 111 by adding or subtracting 1 to the pixel value C(i). Insert 1 at C(i) and go to END. (In this case +1 or -1 change in pixel value is required)

viii) END.

3.3 Retrieval Algorithm

i) Compute the blocking factor (BF) using the cover image size in pixels i.e. I(p) and the message length L(m) in bits as given by equation (2).

ii) The image is also divided in at least L(m) blocks of size BF at the retrieval end.

iii) The block where message bit is present is chosen by using Pseudo-Random Number Generator by using a secret key.

iv) With the block i indicated by PRNG, we calculate its central pixel C(i) as given by equation (3).

v) Check whether at C(i), the combinations of last three bits are 000, 010, 100 or 110. If yes, then 0 is the message bit else 1 is the message bit.

vi) END.

4. CHANGE IN PIXEL VALUES AFTER INSERTION OF MESSAGE

In simple Gray Level Image, each pixel is represented by 8 bit. So, there are 256 possible values of a pixel. Now, we see how these 256 values can change during insertion of message. Table 1 shows how these pixel values changes during insertion of 0 and Table 2 shows how pixel values changes during insertion of 1.

Table 1. Change in pixel values during insertion of ‘0’

Decimal Value	Pixel value before insertion of ‘0’	Last three Bits before Insertion of ‘0’	Pixel value after insertion of ‘0’	Last three Bits After Insertion of ‘0’	Change in Pixel value & comment for insertion of ‘0’
0	00000000	000	00000000	000	NC, Insert
1	00000001	001	00000010	010	+1, Insert
2	00000010	010	00000010	010	NC, Insert
3	00000011	011	00000100	100	+1, Insert
4	00000100	100	00000100	100	NC, Insert
5	00000101	101	00000110	110	+1, Insert
6	00000110	110	00000110	110	NC, Insert
7	00000111	111	00001000	000	+1, Insert
8	00001000	000	00001000	000	NC, Insert
9	00001001	001	00001010	010	+1, Insert
10	00001010	010	00001010	010	NC, Insert
11	00001011	011	00001100	100	+1, Insert
12	00001100	100	00001100	100	NC, Insert
13	00001101	101	00001110	110	+1, Insert
14	00001110	110	00001110	110	NC, Insert
15	00001111	111	00010000	000	+1, Insert
.	.		.		.
.	.		.		.
.	.		.		.
127	01111111	111	10000000	000	+1, Insert
128	10000000	000	10000000	000	NC, Insert
.	.		.		.
.	.		.		.
.	.		.		.
254	11111110	110	11111110	110	NC, Insert
255	11111111	111	11111110	110	-1, Insert

* NC = No Change

Table 2. Change in pixel values during insertion of ‘1’

Decimal Value	Pixel value before insertion of ‘1’	Last three Bits before Insertion of ‘1’	Pixel value after insertion of ‘1’	Last three Bits After Insertion of ‘1’	Change in Pixel value & comment for insertion of ‘1’
0	00000000	000	00000001	001	+1, Insert
1	00000001	001	00000001	001	NC, Insert
2	00000010	010	00000001	001	-1, Insert
3	00000011	011	00000011	011	NC, Insert
4	00000100	100	00000011	011	-1, Insert
5	00000101	101	00000101	101	NC, Insert
6	00000110	110	00000101	101	-1, Insert
7	00000111	110	00000111	111	NC, Insert
8	00001000	000	00000111	111	-1, Insert
9	00001001	001	00001001	001	NC, Insert
10	00001010	010	00001001	001	-1, Insert
11	00001011	011	00001011	011	NC, Insert
12	00001100	100	00001011	011	-1, Insert
13	00001101	101	00001101	101	NC, Insert
14	00001110	110	00001101	101	-1, Insert
15	00001111	111	00001111	111	NC, Insert
.	.		.		.
.	.		.		.
.	.		.		.
127	01111111	111	01111111	111	NC, Insert
128	10000000	000	01111111	111	-1, Insert
.	.		.		.
.	.		.		.
.	.		.		.
254	11111110	110	11111111	111	+1, Insert
255	11111111	111	11111111	111	NC, Insert

5. Results & Analysis

5.1 From Table 1 & Table 2, we can calculate the following:

i) Chances of Message Insertion at a pixel value

= (Pixel Values where we can Insert Message/Total Possible Values of a Pixel)*100

= (256/256)*100

= 100%

ii) Chances when no change in pixel value is required after insertion of message

= (Pixel Values where no change is required after insertion of message/Total pixel values where we can insert the message)*100

= (128/256)*100

= 50%

5.2 Comparison Based Upon Different Types of Noises

We added different types of noises to the stego image and try to recover the message. The results that we got are defined at three levels:

- The Noise Level at which message Remain Intact.
- The Noise Level at which message is recovered.
- The Noise Level at which message is lost.

The results that we got are compared with LSB Method and 6th & 7th Bit Method. Figure 2 shows the original image. Figure 3 shows the stego image after the insertion of message of length 2048 bits by CCM. Figure 4 to Figure 12 shows the stego image (Figure 3) with addition of various types of noises at different levels.

Figure 2. Original Image

Figure 3. Stego Image

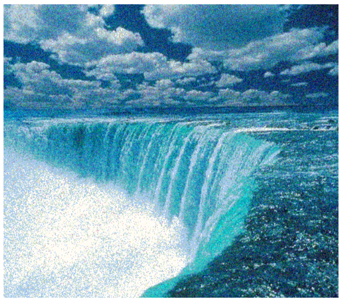

Figure 4. Stego Image with Gaussian Noise (Variance 0.0000004)

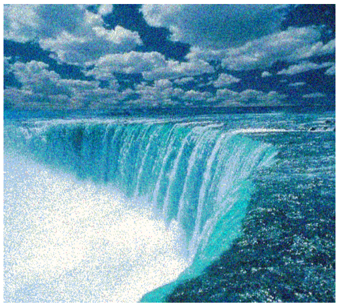

Figure 5. Stego Image with Gaussian Noise (Variance 0.0000006)

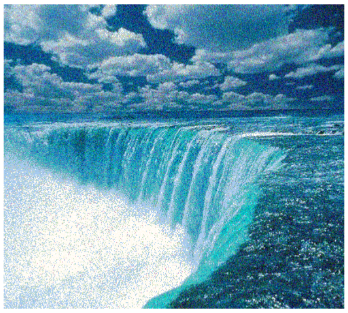

Figure 6. Stego Image with Gaussian Noise (Variance 0.0000009)

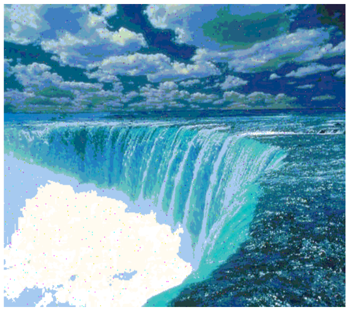

Figure 7. Stego Image with Salt & Pepper Noise (Density 0.004)

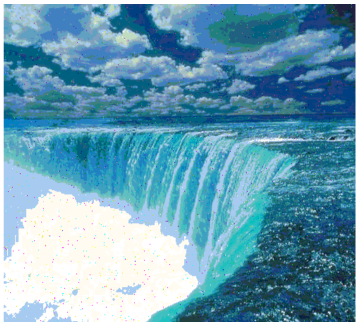

Figure 8. Stego Image with Salt & Pepper Noise (Density 0.006)

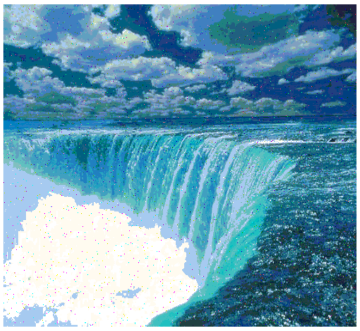

Figure 9. Stego Image with Salt & Pepper Noise (Density 0.009)

Figure 10. Stego Image with Speckle Noise (Variance 0.000005)

Figure 11. Stego Image with Speckle

Figure 12. Stego Image with Speckle Noise (Variance 0.00001)

Table 3 shows the result of LSB method after addition of different noises. Table 4 shows the results of 6th & 7th bit method after addition of different noises. Table 5 shows the result of CCM after the addition of different noises. By comparing the results of Table 3, 4 & 5, we found that our method provides more immunity against various types of noises.

Table 3. Effects of noise on stego image using LSB Method

Types of Noise	Noise level at which message remains same	Noise level at which message is recoverable	Noise level at which message is lost
Gaussian	0.0000002	0.0000003-0.0000006	0.0000007
Salt and Pepper	0.003	0.004-0.008	0.009
Speckle	0.000003	0.000004-0.0001	0.0002

Table 4. Effects of noise on stego image using 6th, 7th Bit Method

Types of Noise	Noise level at which message remains same	Noise level at which message is recoverable	Noise level at which message is lost
Gaussian	0.0000003	0.0000004-0.0000007	0.0000008
Salt and Pepper	0.003	0.004-0.009	0.01
Speckle	0.000004	0.000005-0.00009	0.0001

Table 5. Effects of noise on stego image using CMM Method

Types of Noise	Noise level at which message remains same	Noise level at which message is recoverable	Noise level at which message is lost
Gaussian	0.0000004	0.0000005-0.0000008	0.0000009
Salt and Pepper	0.004	0.005-0.008	0.009
Speckle	0.000005	0.000006-0.000009	0.00001

5.3 Security Analysis

The security analysis compare the original image (Figure 2) with the stego image (Figure 3) based on the histogram of images. Comparing the histograms of original image and the stego image gives us the clear idea of security. If the change is minimum in the stego image, then stego system is considered to be secure. The stego image after applying did not show any visual difference. The histograms of original image and stego image are given Figure 13 & Figure 14 respectively. The histograms showed no change in the lower part of the image but in the upper part it shows a little bit of difference.

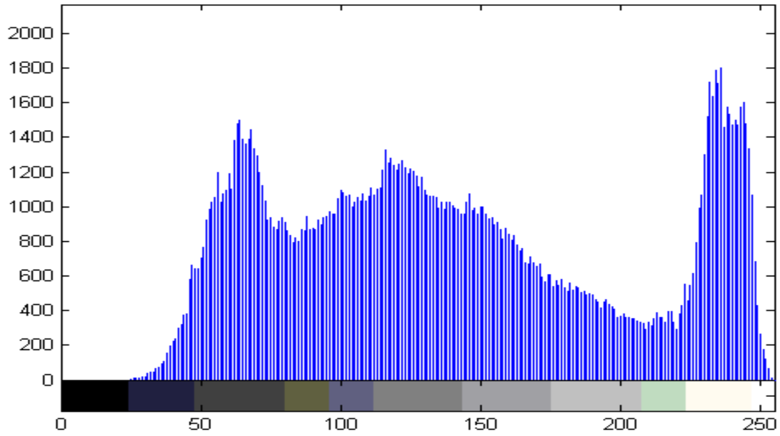

Figure 13. Histogram of Original Image (Given in Figure 2.)

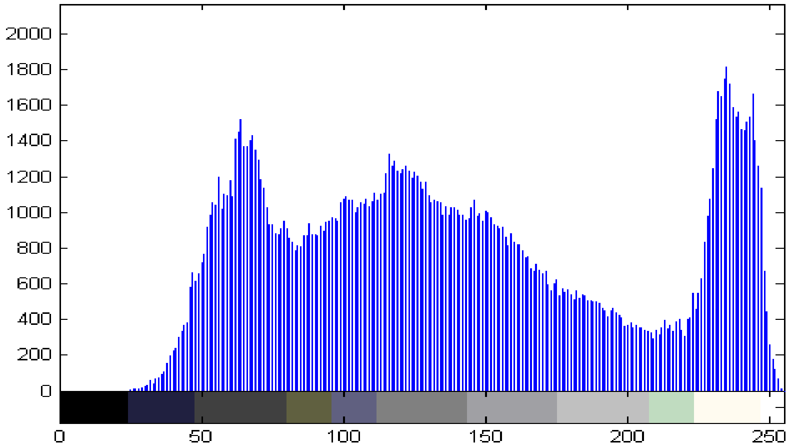

Figure 14. Histogram of Stego Image (Given in Figure 3.)

5.4 Strong Degree of Tamper Resistance

CCM provides strong degree of Tamper Resistance. As in case with LSB, intruder can change LSB's of all pixel values. In this way hidden message will be destroyed and change fall in the range of +1 or -1 only. This was the major security threat with LSB method. CCM removes this security threat. If intruder changes LSB's of all pixel values with our method then at the receiver end there are two clues which reveal that intruder has tampered the image:

- At some pixel locations, the change becomes +2 or -2 which is visible to human eye.
- The message is only inserted at the central pixel of block and changes are made at the other pixels also by the intruder.

So, if intruder tampers the image then at the receiver end, it becomes visible that intruder has changed the image. In that case, receiver may ask the sender to retransmit the message.

6. CONCLUSION AND FUTURE WORK

We have proposed Cyclic Combination Method (CCM) for digital image steganography. This method uses the cyclic combination of last three bits for insertion and retrieval of message at the central pixel of the selected block. The block for insertion and retrieval of message bit are selected by using pseudo random number generator that is seeded with a secret key which is shared between sender and receiver. This method also distributes the message uniformly in the image. This method also provides greater immunity to various types of noises. This method provides minimal change at a pixel value i.e. of +1 or -1 and does not provide any clue to the intruder to identify difference between original image and stego image. This method also provides strong degree of temper resistance. If the intruder tries to tamper with the stego image then it becomes visible at the receiver end that intruder had tempered with the stego image. Future work will concentrate on improving the robustness of this technique by using it in the frequency domain.

7. REFERENCES

[1] A. Gutub & M. Faltani (2007), "A Novel Arabic Text Steganography Method Using Letter Points and Extension", WASET International Conference on Computer Information and System Science and Engineering (ICCISSE), Vienna, Austria, May 25-27.

[2] RJ Anderson & FAP Petitcolas (1998), "On the Limits of Stegnography", IEEE Journal on selected Areas in Communications, Vol. 16 No 4, pp 474-481.

[3] R. Chandramouli & N.D. Memon (2003), "Steganography capacity: A steganalysis perspective", Proc. SPIE Security and Watermarking of Multimedia Contents, Special Session on Steganalysis.

[4] S.K. Pal, P.K. Saxena & S.K. Muttoo (2004), "Image steganography for wireless networks using the handmaid transform", International Conference on Signal Processing & Communications (SPCOM).

[5] M. T. Parvez & A. Gutub (2008), "RGB Intensity Based Variable-Bits Image Steganography", APSCC 2008-Proceedings of 3rd IEEE Asia-Pacific Services Computing Conference, Yilan, Taiwan, 9-12 December.

[6] Eugene T. Lin & Edward J. Delp, “A Review of Data Hiding in Digital Images”, Video and Image Processing Laboratory (VIPER), Indiana.

[7] Neil F Johnson & Sushil Jajodia (1998), “Exploring Stenography: Seeing the Unseen”, IEEE Computer, pp 26-34.

[8] Parvinder Singh, Sudhir Batra & HR Sharma (2005), “Evaluating the Performance of Message Hidden in 1st and 2nd Bit Plane”, WSEAS Transactions on Information Science and Applications, issue 8, vol 2, pp 1220-1227.

[9] D.C. Wu & W.H. Tsai (2003), “A steganographic method for images by pixel-value differencing”. Pattern Recognition Letters, 24: 1613-1626, 2003.

[10] J.M. Rodrigues, J.R. Rios & W. Puech, “SSB-4 System of Steganography using bit 4”.

[11] Rajkumar, Ravi, Gaurav & Suraj Parkash, (2010)“Effects of Noise on Various Image Steganography Techniques”, In the proceedings of National Conference on Emerging Trends in Mobile Technologies & Security, Department of Computer Science & Applications, M.D. University, Rohtak.

[12] Jessica Fridrich, Miroslav Goljan & Rui Du (2001), “Detecting LSB Steganography in Color and Gray-Scale Images”, IEEE Multimedia, issue 4, vol 8.

[13] W Stallings (2003). Cryptography and network security: Principles and practice. In Prentice Hall.

[14] R. Chandramouli & N.D. Memon (2003), “Steganography capacity: A steganalysis perspective”, Proc. SPIE Security and Watermarking of Multimedia Contents, Special Session on Steganalysis.

[15] S.Craver & N. Memon (1998), “Resolving Rightful Ownership with Invisible Watermarking Techniques: Limitations, Attacks and Implications”, IEEE Trans.,Vol 16,No. 4,pp. 573-586.

[16] W. Bender, D. Gruhl, N. Morimoto & A. Lu (1996), “Techniques for data hiding,” IBM Systems Journal, vol. 35, no. 3-4, pp. 313–335.

[17] N. Nikolaidis and I. Pitas (1998), “Robust image watermarking in the spatial domain,” Signal Processing, vol. 66, no. 3, pp. 385–403.

[18] Jing Dong & Tieniu Tan, “Security Enhancement of Biometrics, Cryptography and Data Hiding by Their Combinations”, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, P.O. Box 2728, 10190, Beijing, China.