International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.3 (2014), pp.27-44
http://dx.doi.org/10.14257/ijsip.2014.7.3.04

A New Image Steganographic Approach Based on Mod Factor for RGB Images

Gunjan Chugh[1], Rajkumar Yadav[2] and Ravi Saini[3]

[1]PDM College of Engineering, Bahadurgarh (Haryana), India
[2]U.I.E.T, Maharshi Dayanand University, Rohtak (Haryana), India
[3]C.M.R.A. GP Sanghi, Rohtak (Haryana), India
gunjan.chugh17@gmail.com[1], rajyadav76@rediffmail.com[2],
ravisaini1988@rediffmail.com[3]

Abstract

With the increasing rate of unauthorized access and attacks, security of confidential data is of utmost importance. While Cryptography only encrypts the data, but as the communication takes place in presence of third parties, so the encrypted text can be decrypted and can easily be destroyed. Steganography, on the other hand, hides the confidential data in some cover source such that the existence of the data is also hidden which do not arouse suspicion regarding the communication taking place between two parties. In this paper, we have proposed a new steganographic approach for hiding data in digital images based on calculating the modulus of RGB values with the modfactor.

Keywords: *Steganography, Cryptography, Pseudo random number generator*

1. Introduction

With the rapid development of the internet technologies, digital media needs to be transmitted conveniently over the network [1]. While encryption masks the meaning of a communication, instances exist where we would prefer that the entire communication process not be evident to any observer that is, even the fact that communication is taking place is a secret.

Steganography is an art and science of hiding information in some cover media. The word Steganography comes from the Greek origin, means “concealed (covered) writing”. The word ‘steganos’ means “covered or protected” and ‘graphie’ means “writing” [2]. Steganography is thus, not only the art of information hiding, but also the art and science of hiding the fact that communication is even taking place [3, 4]. By embedding one piece of data inside of another, the two become a single entity, thus eliminating the need to preserve a link between the two different pieces of data, or risk the chance of their separation. One application than exhibits the advantage of this facet of steganography is the embedding of patient information within the medical imagery. By doing so a permanent association between these two information objects is created [5-7].

The concept of “What You See Is What You Get” which we encounter sometimes does not always hold true. Images can be more than what we see with our Human Visual System (HVS); hence they can convey more than merely 1000 words. Figure1 shows how a Steganographic system works [8, 9].

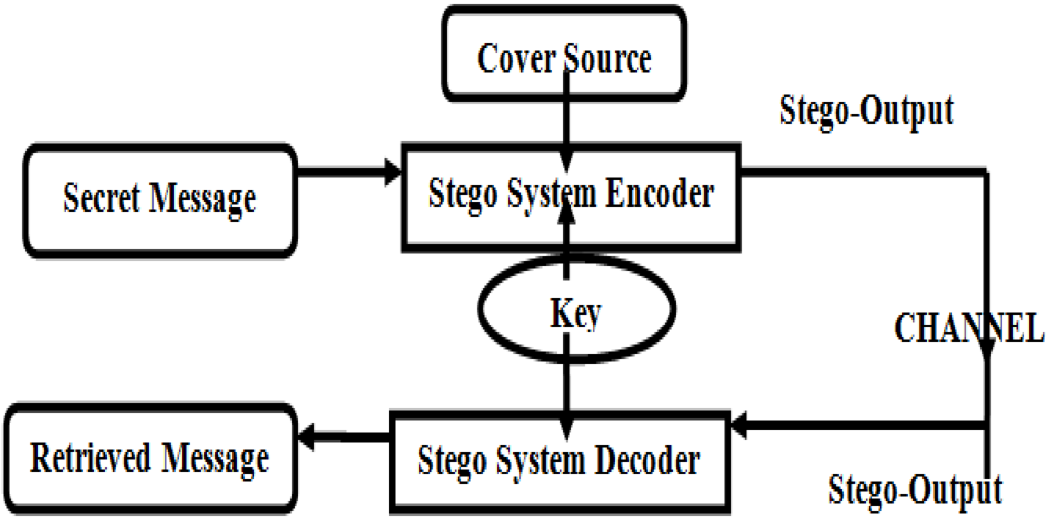


Figure 1. An Overview of Steganographic System

Steganography is not a new science. It dates back to 1499, and it has long history [10].

According to Greek historian Herodotus, the famous Greek tyrant Histiaeus, while in prison, used unusual method to send message to his son-in-law. He shaved the head of a slave to tattoo a message on his scalp. Histiaeus then waited until the hair grew back on slave's head prior to sending him off to his son-in-law. Herodotus provides the first records of steganography in Greece. To communicate Greeks would etch the message they wished to send into the wax coating of a wooden tablet. The tablet would then be transported to the recipient who would read the message, then re-melt the wax to etch their reply [4, 6].

Messages were written on envelopes in the area covered by postage stamps. Messages were hidden using secret inks. Later, chemical effected sympathetic inks were developed. These were chemicals that could be treated with other chemicals causing reactions that would make the result visible. Messages were also hidden in living creatures, for example, by feeding a letter in meat to a dog and then killing him to retrieve it [6, 10, 11].

2. Related Work

(i) LSB (Least Significant Bit) method [12, 13]

It is one of the most common and easiest methods for message hiding. In this method, message is hidden in the least significant bits of image pixels .Changing the LSB of the pixels does not introduce much difference in the image and thus the stego image looks similar to the original image. In case of 24-bit images three bits of pixel can be used for LSB substitution as each pixel has separate components for red, green and blue.

(ii) Pixel Value Differencing (PVD) technique [14]

In this method, Wu and Tsai, selected two consecutive pixels for embedding the message. By checking the difference between two consecutive pixels, payload of this method is determined and it serves as basis to find out whether the two pixels belong to an edge area or smooth area. If the difference is large, it means pixels belong to an edge area and more secret data can be embedded at this location. On the other hand, if difference is small, it means pixels belong to smooth area and less secret data can be embedded at this place. If the original difference value is unequal to the secret message, then the two consecutive pixels are directly adjusted so that the difference value can stand for the secret data.

(iii) 6th, 7th & 8th Bit Method [15]

This method was proposed by Batra, *et al.,* This method increases the chances of message insertion at first instance from 50% to 85.93%. In this method, 6th, 7th and 8th bit of the pixel is used for message insertion. They introduce a time factor, *i.e.,* at some time sender sends thee cover object with message and at some other time sender sends the cover object without message. Thus, the advantage offered by this method is that by even if intruder changes least significant bits of all the pixels even then the message can be retrieved by comparing two cover objects - one containing the message and the other not containing the message.

(iv) Gray Level Modification (GLM) Steganography [16]

In this technique, gray level values of the image pixels are modified. It provides one-to one mapping between the binary data and the selected pixels in an image. A set of pixels are selected from the image. Firstly, all odd selected pixels are made even by changing gray level by one unit. Then, a comparison is made by selecting first bit from the message and first bit of the pixel. If the first bit is even (0) then all pixels have even gray level and are not modified at all. But if the first bit is odd (1) then gray level of the pixel is decremented by one unit to make its value odd. Thus, Gray Level of all the pixels is modified accordingly.

(v) Logical AND Operation and Pixel Position Method [17]

In this technique, Ravi Saini, *et al.,* used the logical AND operation on both selected pixel intensity and selected pixel position. Here, firstly both pixel intensity and pixel position are converted into binary equivalents. Then, four least significant bits of each binary equivalent are removed and logical AND operation is performed on them. To insert 0, result of logical AND must be 0, if not then the pixel intensity are modified such that result of logical AND becomes 0. Similarly, to insert 1 , result of logical AND must be greater than 0 , if not then the pixel intensity are modified such that result of logical AND becomes greater than 0. At the receiver side, after calculating logical AND of pixel intensity and pixel position, if result is 0 then 0 is the message bit else 1 is the message bit. The advantage of this method is that it makes Steganalysis more difficult because it distributes the message uniformly on all the bits of pixel value.

3. Description of the Proposed Method

In a computer, images are represented as arrays of values. These values represent the intensities of the three colours R (Red), G (Green) and B (Blue), where a value for each of three colours describes a pixel *i.e.*, a pixel is a combination of three components(R, G, and B). In the proposed method, the concept of “modfactor” is introduced which is calculated as follows:

$$\boxed{\text{Mod Factor} = 2^n}$$

Here, n is the no. of bits to be hidden in one component of a pixel, so accordingly n can be 2, 3, 4 *etc*. By taking n=2, such that Mod Factor would be 4, two bits are inserted in each component and as each pixel as three components, therefore, in each pixel six bits of the secret message are hidden by the proposed technique. As the proposed method is based on calculating the Mod Factor, therefore, it is also known as Modulus Method for Image Steganography.

Insertion: In case of insertion of message bits, for each pseudorandom pixel location, mod value is calculated by taking modulus of Red, Green and Blue component with the Mod Factor(4). Then comparison is done for the calculated mod value with the message bits to be hidden, as shown below:

Message Bits to be hidden	**Mod Value**
00	0
01	1
10	2
11	3

Retrieval: For retrieval of message, same concept is used as for insertion but in reverse. Receiver will calculate the mod value of RGB components with the Mod Factor (4) for the same pseudorandom pixels as agreed with the sender. Here, comparison is done as follows:

Mod Value	**Retrieved Message Bits**
0	00
1	01
2	10
3	11

As in the proposed method, 6 bits are inserted per pixel (2 bits in each component), therefore the proposed method provides a very large capacity enhancement, *i.e.*, Payload Capacity as compared to the other methods with minimum degradation in image quality.

4. Insertion and Retrieval Process

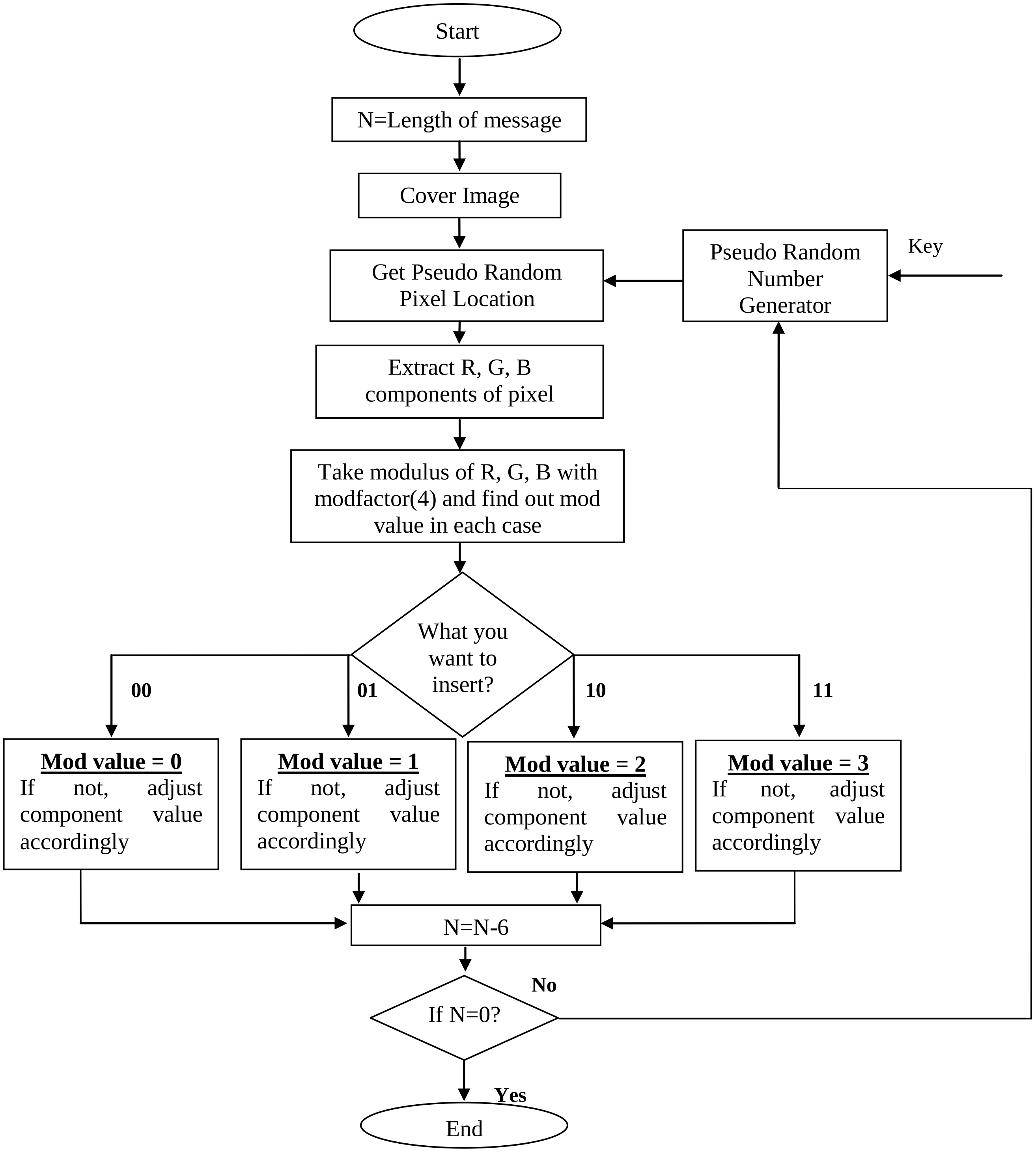

Start
N=Length of message
Cover Image
Get Pseudo Random Pixel Location
Pseudo Random Number Generator
Key
Extract R, G, B components of pixel
Take modulus of R, G, B with modfactor(4) and find out mod value in each case
What you want to insert?
00
01
10
11
Mod value = 0 If not, adjust component value accordingly
Mod value = 1 If not, adjust component value accordingly
Mod value = 2 If not, adjust component value accordingly
Mod value = 3 If not, adjust component value accordingly
N=N-6
If N=0?
No
Yes
End

Figure 2. Proposed Insertion Process

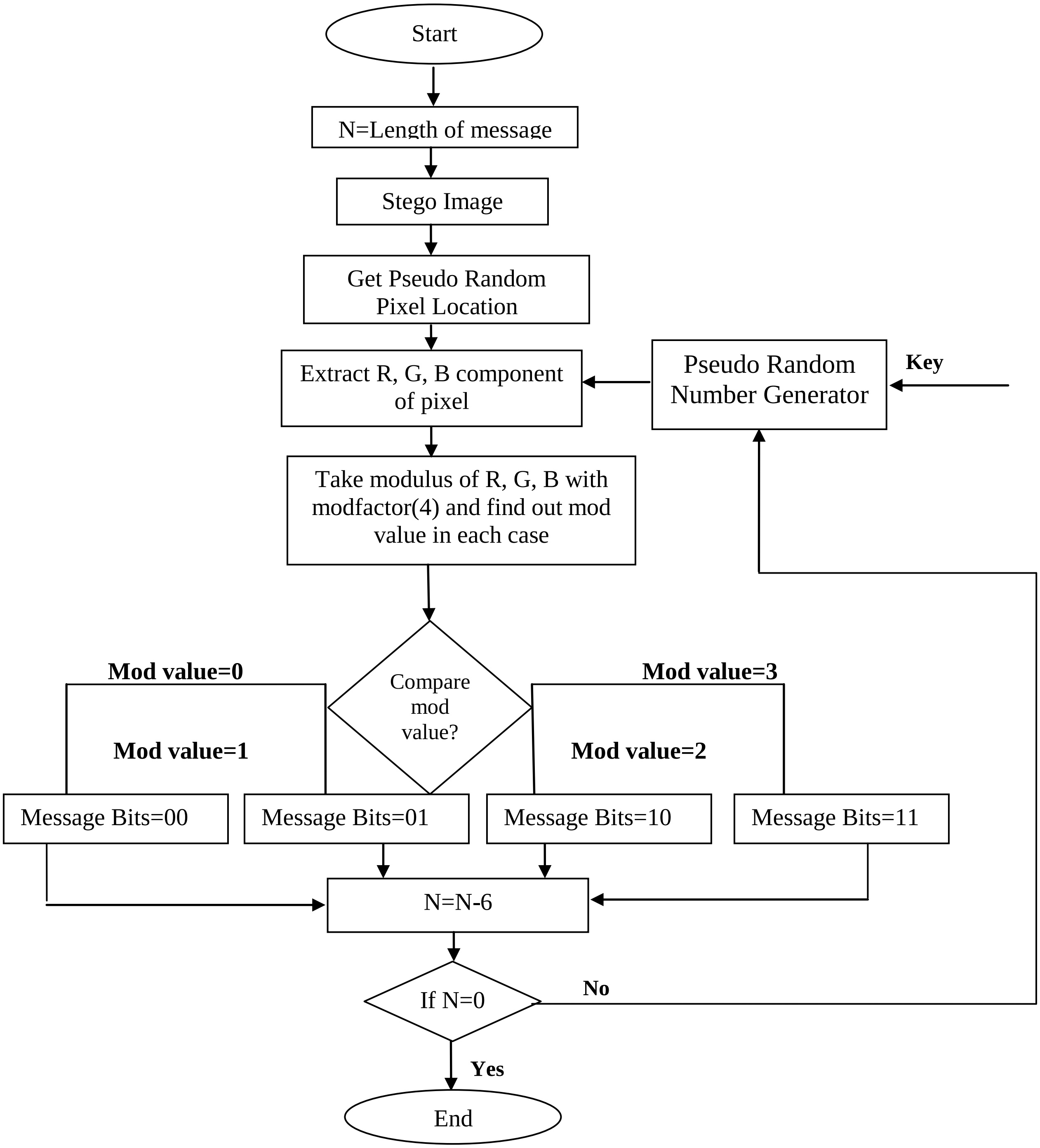


Figure 3. Proposed Retrieval Process

5. Algorithm

5.1. Assumption

(i) Sender and recipient agree on the cover object in which message is supposed to be hidden.
(ii) Both sender and recipient agree on the same pseudo–random key to decide the random locations where message is to be inserted.

5.2 Insertion Algorithm

(i) Find pseudo- random location (L) in the cover image from secret key where message bits are to be inserted.
(ii) Extract Red, Green and Blue components from the selected pseudo random pixel location (L).
(iii) Take modulus of R, G, B components with modfactor (4) and find out the mod value separately in each case.
(iv) Now compare the mod value as calculated in step (iii) with the following:

Message Bits to be hidden	Mod Value
00	0
01	1
10	2
11	3

(v) If the mod value as calculated in Step (iii) do not match with as in Step (iv) then pixel values are adjusted accordingly such that -
1.) If message bits to be inserted are 00, then mod value (after computing the modulus of pixel component with 4) must be 0, if not, and then pixel values are adjusted accordingly so that mod value becomes 0
2.) If message bits to be inserted are 01, then mod value must be 1, if not, then pixel value are adjusted accordingly so that mod value becomes 1
3.) If message bits to be inserted are 10, then mod value must be 2, if not, then pixel value are adjusted accordingly so that mod value becomes 2
4.) If message bits to be inserted are 11, then mod value must be 3, if not, then pixel value are adjusted accordingly so that mod value becomes 3
(vi) END

5.3. Retrieval Algorithm

(i) Trace out the location (L) from the same secret key as used for insertion of message.
(ii) Extract Red, Green and Blue components from the selected pseudo random pixel location (L).
(iii) Take modulus of R, G, B components with modfactor (4) and find out the mod value separately in each case.
(iv) Now compare the mod value as calculated in step (iii) with the following:

Mod Value	Retrieved Message Bits
0	00
1	01
2	10
3	11

(v) It can be explained as follows:

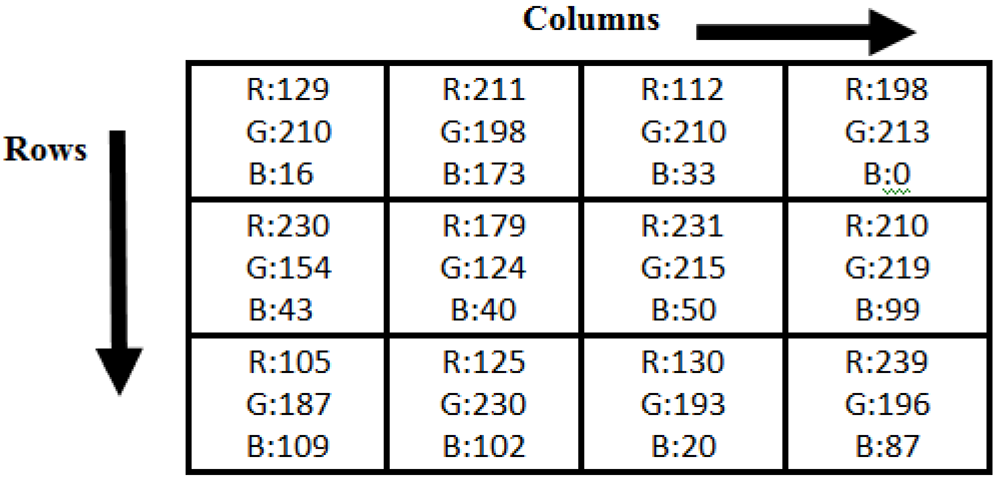
1.) If the mod value(after computing the modulus of pixel component with 4) is 0 , then retrieved message bits are 00
2.) If the mod value(after computing the modulus of pixel component with 4) is 1 , then retrieved message bits are 01
3.) If the mod value(after computing the modulus of pixel component with 4) is 2 , then retrieved message bits are 10
4.) If the mod value(after computing the modulus of pixel component with 4) is 0 , then retrieved message bits are 11

(i) END

6. Example of the Proposed Method

Suppose an RGB image (Figure 4) has the following pixels as shown in Figure 5 and message bits to be inserted in the image are 011010. For insertion of the message bits, firstly selection of pixel locations is to be done by using the pseudo random number generator. Pseudo random number generator uses the same key for insertion as well as retrieval of process

Figure 4

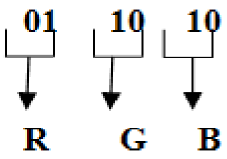

Columns
Rows

R:129 G:210 B:16	R:211 G:198 B:173	R:112 G:210 B:33	R:198 G:213 B:0
R:230 G:154 B:43	R:179 G:124 B:40	R:231 G:215 B:50	R:210 G:219 B:99
R:105 G:187 B:109	R:125 G:230 B:102	R:130 G:193 B:20	R:239 G:196 B:87

Figure 5. Computer Representation of an RGB Image

Pseudo random number generator generates random pixel locations by using key.
Suppose one of the random location generated is: (36, 15)
Original pixel value (R, G, B) at the specified location: **(0, 41, 91)**

Message Bits to be hidden: 011010 (First 2 bits are to be hidden in Red, Next 2 bits in Green and Last two in Blue)



Sender's end:

At this end, Mod Factor is calculated by taking the modulus of the pixel value with 4 and pixel value is modified according to the algorithm 5. 2
0 mod 4 = 0
To hide 01 here, mod value should be 1, therefore, change component value from 0 to 1
41 mod 4 = 1
To hide 10 here, mod value should be 2, therefore, change component value from 41 to 42
91 mod 4 = 3
To hide 10 here, mod value should be 2, therefore, change component value from 91 to 90
Therefore,

New value for pixel = (1, 42, 90)

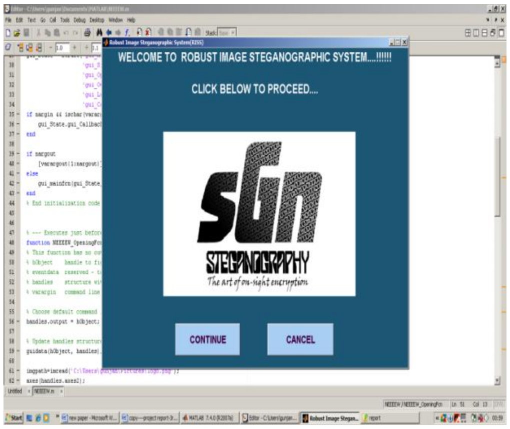
Receiver's end:

At this end, Mod Factor is calculated again with the received pixel values. Based on Mod Factor, message bits are retrieved according to algorithm 5. 3
1 mod 4 = 1
Retrieved message bits = 01
42 mod 4 = 2
Retrieved message bits = 10
90 mod 4 = 2
Retrieved message bits = 10
Therefore,

Retrieved Message Bits= 011010

7. Results & Analysis

This section presents the experimental results obtained after implementing the proposed method in MATLAB. A system is designed and implemented in MATLAB, that shows the working of proposed Image Steganography method. The system is named as Robust Image Steganographic System (RISS) as it provides high value of PSNR and low values of MSE using the proposed method.
Snapshots when the system is executed are shown below:


WELCOME TO ROBUST IMAGE STEGANOGRAPHIC SYSTEM....!!!!!!
CLICK BELOW TO PROCEED....
sGn
STEGANOGRAPHY
The art of on-sight encryption
CONTINUE
CANCEL

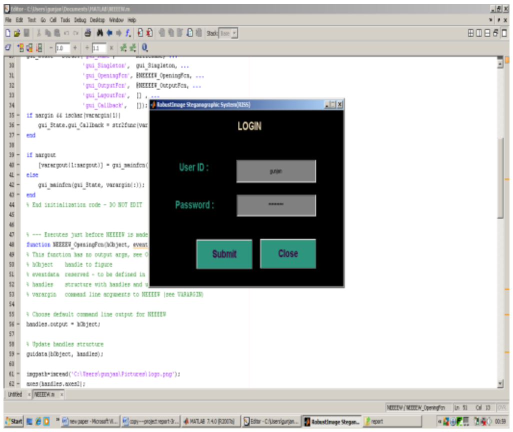
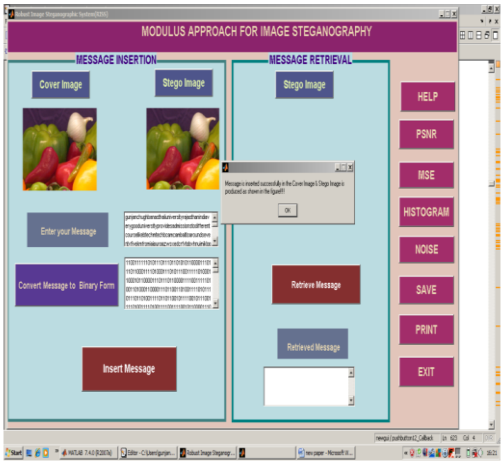


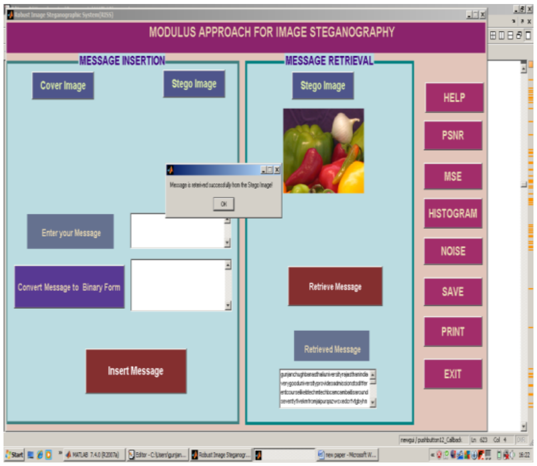
Figure 6 **Figure 7**

Figure 6 shows the first window that pop up when the system is executed. Figure 7 shows login dialog box that provides authentication & will prompt the user for ID and Password. If the user provides correct User ID and Password, the main window will appear where steganography actually takes place.

A Snapshot showing the 'Message Insertion' & 'Message Retrieval' process are shown in Figures 8 and 9.



Message is inserted successfully in the Cover Image & Stego Image is produced as shown in the figure!!!

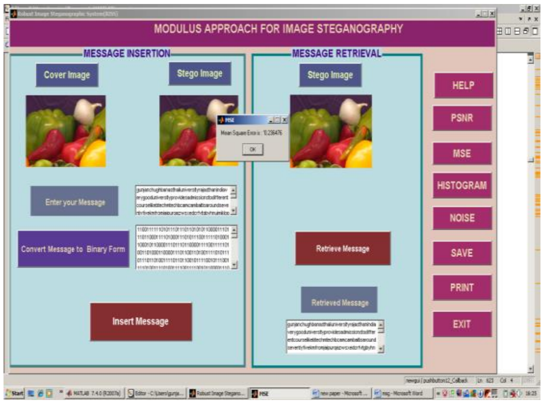


Message is retreived successfully from the Stego Image!

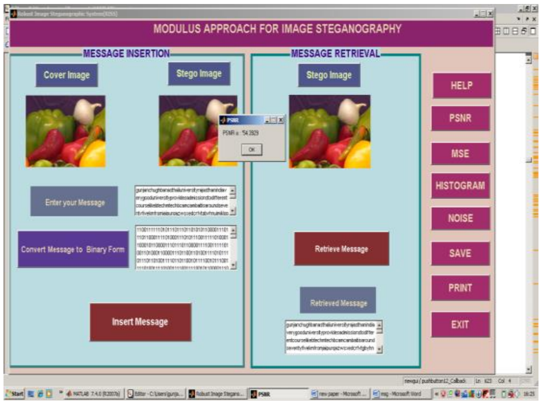
Figure 8 **Figure 9**

In the proposed system, different operations can be performed, for *e.g.*, calculating PSNR and MSE of the cover and stego image, effect of different types of noise on stego image and plotting the graphs of cover image and stego image, (*i.e.*, Histograms).

Calculation of MSE (Figure 10) & PSNR (Figure 11):



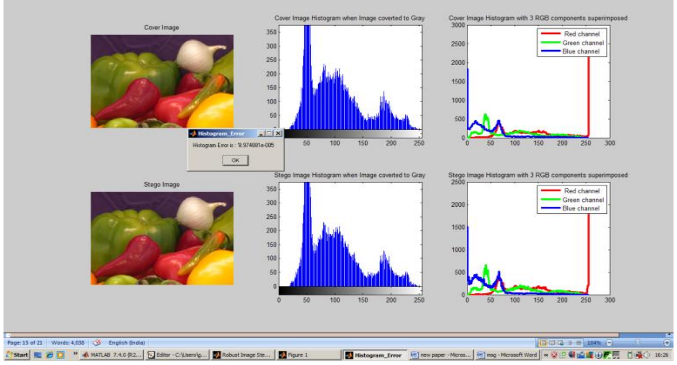
Mean Square Error is : 10.236476



PSNR is : 54.3929

Figure 10 **Figure 11**

Histograms of Cover Image & Stego Image:


Cover Image
Stego Image
Cover Image Histogram when Image coverted to Gray
Cover Image Histogram with 3 RGB components superimposed
Stego Image Histogram when Image coverted to Gray
Stego Image Histogram with 3 RGB components superimposed
Red channel
Green channel
Blue channel
Histogram_Error
OK
0
50
100
150
200
250
300
350
500
1000
1500
2000
2500
3000

Histogram Error is : 8.974881e-005

Figure 12

The database for test images includes different images which are shown below. Robustness of any method depends on different parameters, two of the most important are: PSNR (Peak Signal to Noise Ratio) and MSE (Mean Square Error).

7.1. Comparison based on PSNR

MSE (Mean Square Error) is the average squared difference between a reference image and a distorted image. It is computed pixel-by-pixel by adding up the squared differences of all the pixels and dividing by the total pixel count [12]. It defines the differences between the Cover Image and Stego Image. Lower the MSE, better the quality of Stego Image.

$$\boxed{\begin{array}{c}\textbf{MSE =}\\ \mathbf{1/(M*N)\sum^{n}{}_{i=1}\sum^{m}{}_{j=1}(X_{ij}-Y_{ij})^{2}}\end{array}}$$

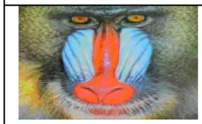
X_{ij} is the Intensity of pixels ij in the Cover Image
Y_{ij} is the Intensity of pixels ij in the Stego Image
N is the Number of pixel rows in the Cover Image
M is the Number of pixel columns in the Cover Image

PSNR (Peak Signal to Noise Ratio) is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation [13, 14]. This ratio is often used as a quality measurement between the original (cover) image and a stego image as it defines the similarities between cover image and stego image. Higher the PSNR, better the quality of the stego image.

$$\boxed{\begin{array}{c}\textbf{PSNR =}\\ \mathbf{10log10\ [\ (255)^2 / MSE]}\end{array}}$$

Experimental results proved that the proposed method provides higher values of PSNR and lower values of MSE for different test images under consideration. The PSNR and MSE results for some standard images of different sizes with variable length of message are shown in Table 1 below:

Table 1. PSNR (in dB) and MSE of the Proposed Method

Cover Image	Message Length (in Bytes)	PSNR	MSE
Lena(204X 204)	**4917**	56.5688	0.1432
	9840	54.6083	0.2250
	17550	53.4223	0.2514
 **Baboon(225 X225)**	**4917**	57.4062	0.1181
	9840	52.5568	0.2272
	17550	54.4474	0.2335
Autumn(345 X 206)	**4917**	58.8244	0.0852
	9840	55.7026	0.1749
	17550	54.2916	0.2421
Peppers(512 X 384)	**9840**	60.2464	0.0614
	17550	57.6658	0.1113
	35114	54.5890	0.2260
Pears(732 X 486)	**17550**	60.3675	0.0597
	35114	57.3072	0.1209
	70233	54.5546	0.2278

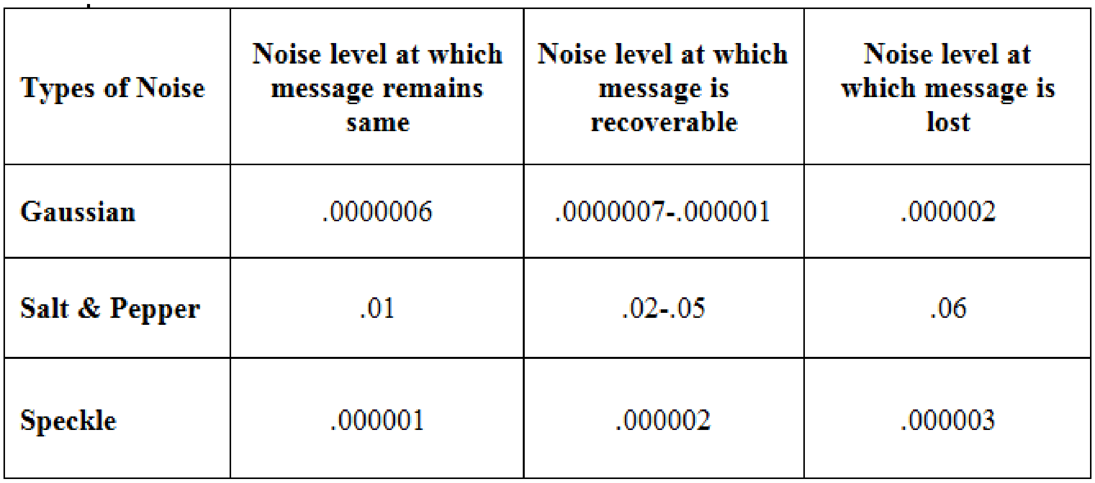
7.2 Comparison Based Upon Different Types of Noises

Different types of noises like ***Gaussian, Salt & Pepper and Speckle are added to*** the stego image and a try is made to recover the message. The results obtained are defined at three levels:

- The Noise Level at which message remains intact.
- The Noise Level at which message is recovered.
- The Noise Level at which message is lost.

Table 2 shows the effect of different types of noise on stego image using proposed method. Experimental results proved that this method provides a higher level of reliability in terms of message recovery than other methods.

Table 2. Effects of Noise on Stego Image using Proposed 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	.0000006	.0000007-.000001	.000002
Salt & Pepper	.01	.02-.05	.06
Speckle	.000001	.000002	.000003

Figure 13 shows the original image. Figure 14 shows the stego image after the insertion of message of length 817 bytes. Figure 15 to Figure 23 shows the stego image with addition of various types of noises at different levels.

Cover Image

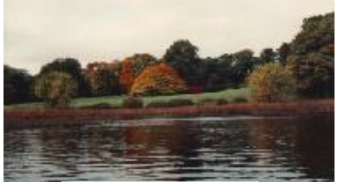


Figure 13

Stego Image

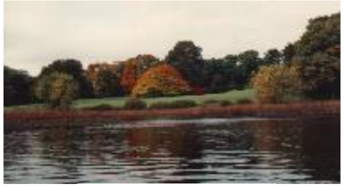


Figure 14

Gaussian Noise with Variance .0000006

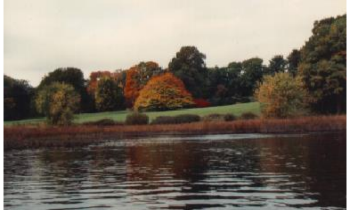


Figure 15

Gaussian Noise with Variance .0000008

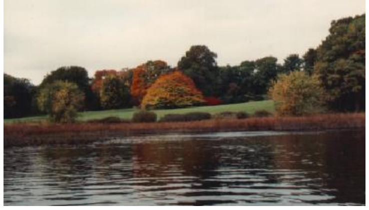


Figure 16

Gaussian Noise with Variance .000002

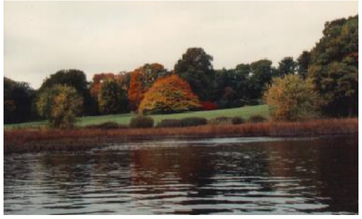


Figure 17

Salt & Pepper Noise with Density. .01

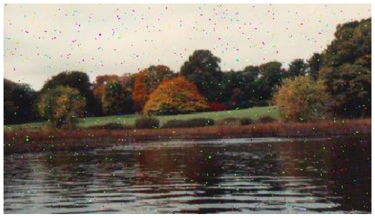


Figure 18

Salt & Pepper Noise with Density .03

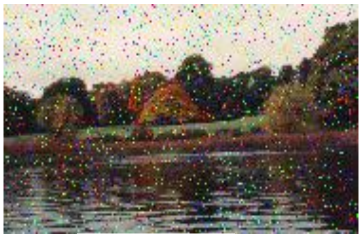


Figure 19

Salt & Pepper Noise with Density .06

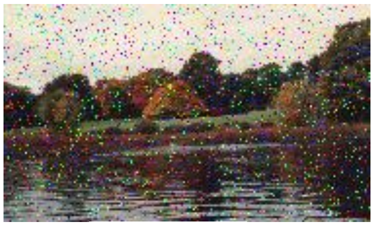


Figure 20

Speckle Noise with Variance=.000001

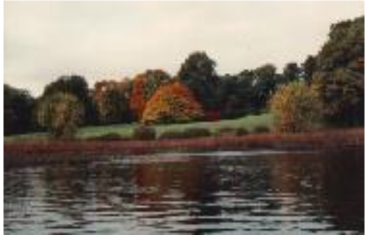


Figure 21

Speckle Noise with Variance=.000002

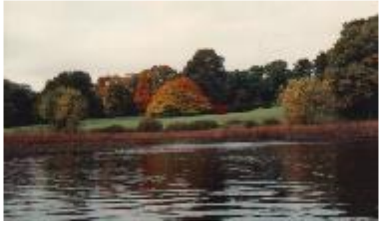


Figure 22

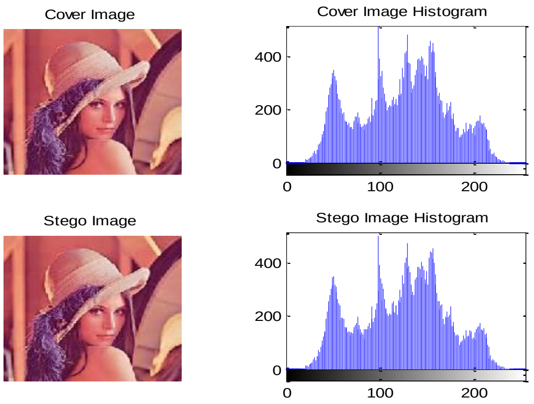
Speckle Noise with Variance=.000003

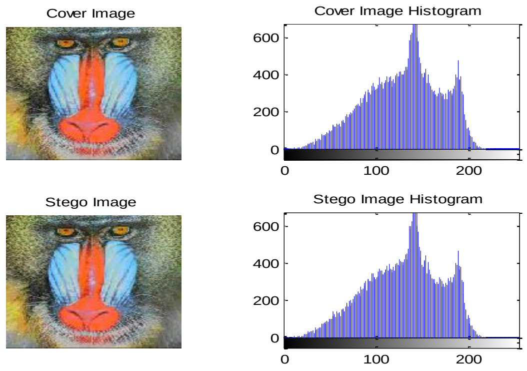
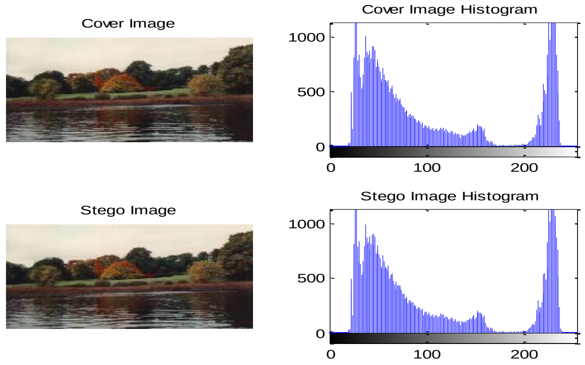
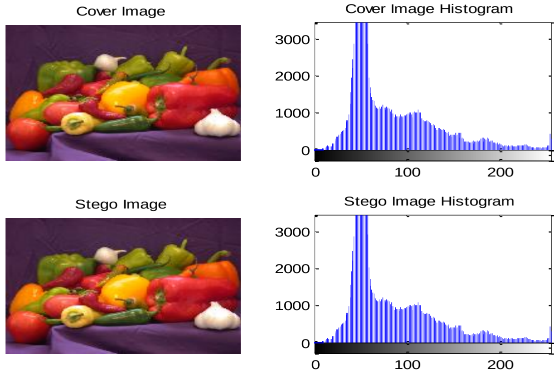
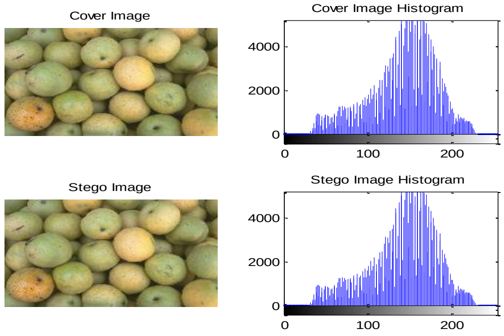
Figure 23

7.3 Security Analysis

The security analysis compares the original image with the stego image based on the Histograms of Images. Comparing the histograms of original image and the stego image gives the clear idea of security. If the change is minimum in the stego image, then stego system is considered to be secure. Table 3 shows histograms of different cover images and their corresponding stego images obtained after inserting message of length 9840 bytes. The histograms showed no change in the lower part of the image but in the upper part it shows a little bit of difference which is negligible when seen by the human eye. Histogram Error, is also calculated which shows the difference in histograms of Cover Image and Stego Image. Histogram error is sum of squared difference between normalized histograms of two images. Lower the error, lower will be the difference in histograms.

Table 3. Comparison of Histograms of Cover Image and Stego Image

Histograms of Cover Image & Stego Image	Histogram Error
	**0.00001419**

	**0.000009795**
	**0.00004313**
	**0.000003920**
	**0.0000004402**

8. Conclusion

In this paper, a new method for image steganography is discussed. The new approach is based on calculating modulus of pixel value (RGB) with the Mod Factor. Techniques used so far, focuses on two or four bits insertion in a pixel which results in lower PSNR and higher MSE but the proposed method embeds six bits per pixel, by taking Mod Factor as 4 (Two in each component) and thus provides capacity enhancement with minimum degradation in stego image quality. Experimental Results shows that the proposed method provides higher PSNR and lower MSE values and thus the proposed method is proved to be robust.

References

[1] A. Kumar and K. Pooja, "Steganography- A Data Hiding Technique", International Journal of ComputerApplications, vol. 9, no. 7 , **(2010).**

[2] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding: A Survey". In Proceedings of the IEEE, Special issue on protection of multimedia content, **(1999).**

[3] J. Kaur and S. Kumar, "Study and Analysis of Various Image Steganography Techniques", International Journal of Computer Science and Technology, vol. 2, no. 3, **(2011).**

[4] R. Yadav, "Study of Information Hiding Techniques and their Counterattacks: A Review Article", International Journal of Computer Science & Communication Networks, vol. 1, **(2011)**, pp. 142-164.

[5] L. M. Marvel, "Image Steganography for Hidden Communication". University of Delaware , Electrical Engineering, PhD Thesis, Springer, **(1999).**

[6] M. Kharrazi, H. T. Sencar and N. Memon, "Image Steganography: Concepts and Practices", Polytechnic University, Brooklyn, USA, **(2004).**

[7] A. Almohammad, "Steganography-Based Secret and Reliable Communications: Improving Steganographic Capacity and Imperceptibility", Department of Information Systems and Computing, Brunel University, PhD Thesis, **(2010).**

[8] T. Morkel, J.H.P. Eloff and M.S.Olivier, "An Overview of Image Steganography", in Proceedings of the Fifth Annual Information Security South Africa Conference Sandton, South Africa, **(2005)**.

[9] A. Cheddad, J. Condell, K. Curran and P. McKevitt, "Digital image steganography: Survey and analysis of current methods", Signal Processing, vol. 90, **(2010)**, pp. 727–752.

[10] J. Ashok, Y. Raju, S. Munishankaraiah and K. Srinivas, "Steganography: An Overview", International Journal of Engineering Science and Technology, vol. 2, no. 10, **(2010).**

[11] R. Yadav, "Analysis of Incremental Growth in Image Steganography Techniques for Various Parameters" International Journal of Computer Technology and Applications, vol. 2, no. 6, **(2011)**, pp 1867-1870.

[12] C.-K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution" The Journal of Pattern Recognition Society, **(2004)**, pp 469–474.

[13] N. F. Johnson and S. Jajodia, "Exploring Stenography: Seeing the Unseen" IEEE Computer, **(1998)**, pp 26-34.

[14] C.-M. Wang, N.-I Wu, C.-S. Tsai and M.-S. Hwang, "A high quality steganographic method with pixel-value differencing and modulus function", J. Syst. Software, **(2007).**

[15] S. Batra, R. Rishi and Rajkumar, "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", International Journal of Security and Its Applications, vol. 4, no. 3, **(2010).**

[16] A. T. Al-Taani and A. M. AL-Issa, "A Novel Steganographic Method for Gray Level Images", International Journal of Computer and Information Engineering, **(2009).**

[17] R. Saini and R. Yadav, "A New Data Hiding method using Pixel Position and Logical AND operation", International Journal of Computer and Electronics Research, vol. 1, no. 1, **(2012)**.

Authors

Gunjan Chugh, she received the M.Tech degree in Computer Science from Banasthali University, Rajasthan (INDIA) in 2013. She is currently working in Computer Science and Engineering Department at PDM College of Engineering, Bahadurgarh, Haryana (INDIA). Her research interest includes Information Hiding, Cryptography, Network Security.

RajKumar Yadav, he received the PhD degree in Computer Science and Engineering from Maharshi Dayanand University, Rohtak, Haryana(INDIA)in 2011.He is currently working in Computer Science and Engineering Department at University Institute of Engineering & Technology (M.D. University Rohtak, Haryana) India. His research interest includes Information Hiding Techniques, Network Security and Biometrics

Ravi Saini, he received the M.Tech (Gold Medallist) degree in Computer Science and Engineering from Maharishi Dayanand University, Rohtak, Haryana (INDIA) in 2011. He is currently working in Computer Science and Engineering Department at Govt Polytechnic Sanghi (Rohtak), under the Department of Technical Education, Haryana. His research interest includes Steganography, Watermarking, Neural Network, *etc.*