INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY (IJCET)

ISSN 0976 – 6367(Print)
ISSN 0976 – 6375(Online)
Volume 4, Issue 2, March – April (2013), pp. 31-44
© IAEME: www.iaeme.com/ijcet.asp
Journal Impact Factor (2013): 6.1302 (Calculated by GISI)
www.jifactor.com

IJCET
© I A E M E

STEGANOGRAPHY BASED ON RANDOM PIXEL SELECTION FOR EFFICIENT DATA HIDING

Shamim Ahmed Laskar [1] and Kattamanchi Hemachandran [2]

[1] Research Scholar, Department of Computer Science, Assam University, Silchar, Assam, India
[2] Professor and Head, Department of Computer Science, Assam University, Silchar, Assam, India

ABSTRACT

In this paper we present a novel steganographic approach to increase the security of the data hidden in a cover RGB image. Here we have used LSB insertion method that hides the bits of a secret message into the least significant bit in the red plane of the pixels within a cover image. The pixels are selected by using a random number generator. It is commonly seen that the changes in the LSB of the colour cannot be detected due to noise that is presents in the digital images by the human visual system. The central idea of the proposed method is to increase security, so the data is embedded only into the red plane of the image. We have also explained the method that extracts the hidden message at the receiving end using a key. The main objective of the paper is to combine both the preferences and the resistance to the visual and statistical attacks for a large amount of the data to be hidden in a cover image.

Keywords: Steganography, Human Visual System, Least Significant Bit (LSB), RGB image, cover-image, stego-image, pseudo random number generator, key, PSNR, MSE.

1. INTRODUCTION

Data is the heart of computer communication and over the years, different methods have been proposed and created to accomplish the goal of using steganography to hide data. The idea is to embed the data into a significantly larger object so that the changes are undetectable by the human eye [1, 5]. The word steganography comes from the Greek language, "Stegos" meaning hidden or covered, and "Graphia" simply meaning writing [4]. Within the field of Computer Forensics, investigators are aware of the fact that steganography can be an effective means that enables concealed data to be transferred inside of seemingly

innocuous carrier files [2]. Every digital file is composed of a sequence of binary digits (0 or 1). It is also a relatively simple task to modify the content of a file by changing a single bit in the sequence. Steganography is a data hiding technique similar to cryptography and watermarking. While the Watermarking ensures the message integrity and Cryptography scrambles a message, the Steganography hides message. It has been observed that all the digital file formats can be used for steganography, but the formats those are with a high degree of redundancy are more suitable. The redundant bits of an object are those bits that can be altered without the alteration being detected easily. Image, audio and video files especially comply with this requirement [4]. Given the proliferation of digital images, and given the high degree of redundancy present in a digital image (despite compression), there has been an increased interest in using digital images as cover-objects for the purpose of steganography. There are many methods that enables in embedding information inside an image. The information can be embedded inside an image file in any order or in specific areas that makes the information invisible and undetectable from third party. Successful embedding depends on the selection of the cover image and the method involved. It is also important to note that steganographic technique not only involves in embedding information inside digital media but also able to successfully retrieve the information from the media. Data hiding should be capable of embedding data with the following conditions: An observer does not notice the presence of the data, the embedded data should be directly encoded into the media and the data remain intact across varying data file formats. Data hiding in images take advantage of the limited power of the human visual system (HVS).

2. IMAGE STEGANOGRAPHY TECHNIQUE

To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image. This numeric representation forms a grid and the individual points are referred to as pixels. These pixels make up the image's raster data. Image steganography is about the exploitation of the limited powers of the human visual system (HVS) [5]. From a mathematical point of view, an image may be defined as a two-dimensional function *F*, where *F*(x, y) gives the value of the pixel at the position (x, y). For black and white images, *F*(x, y) will be a single value representing the gray level of the point; for colour images, *F*(x, y) will be a tuple representing a colour in a colour system. [28]. The RGB (Red-Green-Blue) colour model has three basic primary (basic) colours: red, green, and blue. All other colours are obtained by combining them [5, 12], as shown in the Fig. 1.

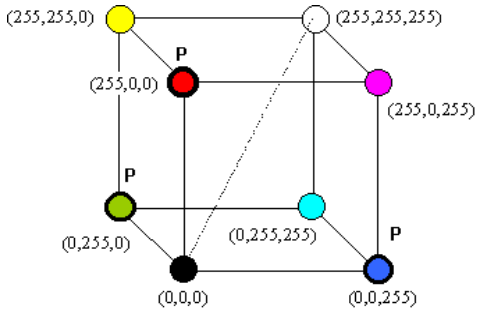

Fig 1: RGB Colour Space. The colours with a P are the primary colours. The dashed line indicates where to find the grays, going from (0,0,0) to (255,255,255).

An image is the most common type of Digital carrier used for steganography [1]. Images consist of pixels with contributions from primary colours (red, green, and blue) adding to the total colour composition of the pixel [8]. Each pixel typically has three numbers associated with it, one each for red, green, and blue intensities, and these value often range from 0-255 [17]. Depending on the depth of colour desired in the final image, each component is represented by a separate number of bits. In the case of a 24 bit bitmap, each colour component has eight bits [12, 16]. Represented as decimal contributions for ease of reading, a value of (255, 0, 0) would describe a 100% red pixel. By mixing the contribution of each component a large palette of colours can be represented. Value mixtures such as (31, 187, 57) can result in a dark green while (255, 255, 0) represents pure yellow. When any specific colour is viewed closely, single digit modifications to the contribution level are imperceptible to the human eye. (i.e. a pixel with a value of (255, 255, 0) is indistinguishable from (254, 255, 0). Figure 2 illustrates the impact of modifying one bit in the red contribution for two yellow boxes.

(255, 255, 0) **(254, 255, 0)**

Fig 2: Red Channel LSB Modification Example Colour Change

Digital images often have a large amount of redundant data and the steganographic methods use this redundant data to hide the desired information. This is relatively easy because an image, being an array of pixels, typically contains an enormous amount of redundant information [6, 11]. The security of stego-images depends entirely on their ability to go unnoticed with extra loaded data [5]. It is important to remember that if you hide information inside of an image file and that file is converted to another image format, it is most likely that the hidden information may be lost.

3. LSB BASED DATA EMBEDDING

Least significant bit (LSB) is the most popular method of embedding scheme where information is hidden in the spatial domain of an image. This method is probably the easiest way of hiding information in an image and yet it is surprisingly effective [7, 8]. It works by using the least significant bits of an image to hide the most significant bits of another [1, 11]. LSB steganography is described as follows: if the LSB of the pixel value $I(i, j)$ is equal to the message bit m to be embedded, $I(i, j)$ remain unchanged; if not, set the LSB of $I(i, j)$ to m [40]. The message embedding procedure can be described using Equation (1) as follows,

$$I_S(i,j) = \begin{cases} I(i,j) - 1 & LSB\big(I(i,j)\big) = 1 \; and \; m = 0 \\ I(i,j) & LSB\big(I(i,j)\big) = m \\ I(i,j) + 1 & LSB\big(I(i,j)\big) \neq 0 \; and \; m = 1 \end{cases} \quad (1)$$

where LSB($I(i, j)$) stands for the LSB of $I(i, j)$ and m is the next bit to be embedded.
As we already know each pixel is made up of three bytes consisting of either a 1 or a 0. If we had the following pixel:

R G B
1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 1 0 0 1 1 0 0
The LSB would be as follows:
R G B
1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 1 1 0 0 1 1 0 0

The most significant bits (MSB) are the ones to the left, and the least significant bits are the one to the right. Changing the MSBs will have a noticeable impact on the colour, however, changing the LSBs will not be noticeable to the human eye [22]. In general though 01101010 could be changed to 01101011, 01101000, or 01101001 and would go unnoticed by the casual observer. The last two bits could be used to embed data, allowing a message of **10110001** [19]. The image formats typically used in the LSB substitution are lossless and the data can be directly manipulated and recovered [25].

Information bits can be embedded in image's LSB sequentially or randomly distributed in image pixels. In sequential embedding approach, the LSB's of the image is replaced by the message bit sequentially or successively. But there is disadvantages of sequential embedding as the message is encoded in the image file sequentially (ordered), so we can find clusters of bits embedded, resulting in abrupt changes in the bits statistics, and this makes the detection easier.

Whereas in random embedding of data, there are no such clusters because the embedded bits are scattered randomly in the image, so we cannot expect the detection process difficult as compared to sequential embedding. In the random embedding, the message bits are randomly scattered throughout the whole image using a random sequence to control the embedding sequence. The key used to generate pseudorandom numbers which is shared by both the sender and receiver, which will identify where, and in what order the hidden message is laid out.

4. RELATED WORKS

The advantages of LSB based data hiding method is that it is simple to embed the bits of the message directly into the LSB plane of image and many techniques use these methods [11]. The LSB modification does not result in image distortion and thus the resulting stego-image will look identical to the cover-image [29]. Several variations of the basic LSB based steganographic techniques were described by Johnson, and Katzenbeisser [3]. They also describe a substitution technique for embedding message into the LSB bits of the palette of GIF or BMP image format. Bailey and Curran provide an evaluation of various techniques concerning spatial steganographic that principally applies to GIF images [29]. They discussed different strengths in terms of resistance to different types of steganalysis or their ability to maximize the size of the message that could be stored.

In [14] the authors emphasize strongly on image Steganography providing a strong focus on the LSB techniques in image Steganography. This paper explained the LSB embedding technique and presents the evaluation results for 2, 4, 6 least significant bits for a PNG and BMP file. By applying an optimal pixel adjustment process to the stego-image obtained by the simple LSB substitution method, the image quality of the stego-image can be greatly improved with low extra computational complexity [30].

In [31], the authors proposed a method of data hiding based on LSB Matching and the usage of Boolean functions in stream cipher where the cover media for embedding are

grayscale images, and the Boolean function is used for encryption and controlling the pseudo-random increment or decrement of LSB. An LSB steganalysis technique that can detect the existence of hidden messages that are randomly embedded in the least significant bits of natural continuous-tone images is proposed in [32]. They also discussed that the formation of some subsets of pixels whose cardinalities change with LSB embedding, and such changes can be precisely quantified under the assumption that the embedded bits are randomly scattered. An adaptive Steganography scheme is proposed by Santoshi and Rao [33]. The adaptive quantization embedded is introduced and employed by block-wised fashion. They also constructed contrast-correlation distortion metric to optimally choose quantization steps for image blocks to guarantee more data being embedded in busy areas.

An image steganographic approach has been proposed comprising an edge-based scheme [34]. They discussed that there exist some smooth regions in natural images, which would cause the LSB of cover images not to be completely random or even to contain some texture information just like those in higher bit planes. If embedding a message in these areas, the LSB of stego images becomes more random, and according to their analysis and experiments it is easier to detect. This paper proposes a new approach which hides an executable file in images.

A complex method of image based Steganography is known as the patchwork algorithm [35]. This algorithm selects n pairs of pixels randomly using pseudorandom generator from a given image and the brightness of the brighter pixel is increased and decreases the brightness of the other. The contrast change between these two pixels forms part of the bit pattern for the hidden file. Thus the contrast of this set is increased without any change in average luminosity of the image. With suitable parameters, Patchwork even survives compression using JPEG but may cause visible changes if the data size is large in case of insertion at transform domain. However, it embeds only one bit of information [36]. Lee and Chen have introduced an image steganographic model and have proposed a new high- capacity embedding and extracting method that is based on the variable-sized LSB insertion [37]. In the embedding part, based on the contrast and luminance property, they used three components to maximize the capacity, minimize the embedding error and eliminate the false contours. Using the proposed method, they embedded at least four message bits in each pixel .The stego-key is used to locate the embedding positions on the cover-image, which is used as the seed of the random number generator.

In this steganographic method [38], LSB technique is used for embedding the characters of the secret text message into image file using LSB but before embedding the entire text is chopped up and each segment are randomized at bit level. All the segments of text are randomly inserted into different regions of the cover image. A pseudo-random sequence generator function is used to generate a pseudo-random sequence to embed each of the unit of secret message into the logical square regions or blocks of the cover image in a random fashion.

In[39] the authors proposed an optimal key permutation method using genetic algorithms to solve the problem of hiding important data in the rightmost k LSBs of the cover-image when k is large, which may involve huge computation time to find the optimal result. In paper [18], the authors applied random numbers logic based steganographic methods on least significant bit transformation and layout management schemes for hiding data/image into image(s). They also formulated variants based on users' choices and calculated the results of such combinations.

5. PROPOSED DATA HIDING TECHNIQUE

In the Proposed Method, we suggest an algorithm that would affect the visuality of the image so little that it is almost impossible to notice the changes in the image by human being's visual interpretation. In the present study an LSB embedding robust steganographic method has been proposed, where we use two data files. The first is the innocent-looking *cover* image that will hold the information to be hidden. The second is the information file, to be embedded in the image. When combined, the cover image and the embedded message make a *stego-image.* A key is used as seed for the Pseudo-Random Number Generator is needed in the embedding process [21]. By using the key, the chance of getting attacked by the third party is reduced [36]. In general the embedding process inserts a mark X, in an object Y. A key K [4, 10], usually produced by a random number generator is used in the embedding process and the resulting marked object Y is generated by mapping

$$X * Y * K = Y \qquad (2)$$

A pseudorandom number generator (PRNG) can be used to choose the pixels randomly and embed the message [23, 24]. This will make the message bits more difficult to find and hopefully reduce the realization of patterns in the image [9, 18]. Data can be hidden in the LSB of a particular colour plane (Red plane) of the randomly selected pixel in the RGB colour space. If the message is much smaller than the capacity of the image, a problem may occur whereby the information will be packed into one part of the image for example the top half. This is solved by using a PRNG which will spread the message all over the image [10]. Hence the noise also will be randomly distributed. A pseudo random number generator calculates and selects the order of pixels to be chosen for data embedding based on the key.

5.1 Data embedding procedure

The message to be hidden is converted into the bytes that are each character in message is converted into its ASCII equivalent. For an example if we take the character "A" in the message then "A"= 1000001 is stored in byte array. Because ASCII value for "A" is 65 and binary equivalent is 1000001. As image comprises of pixel contribution from red, green and blue components and each pixel has numbers from the colour components (for 24-bit bitmap image each of red, green and blue pixel has 8 bit). At 8 bit of the colour number, if we change the least significant bit, our sighted system can detect changes in pixel and thus it is possible to replace message bits with image pixel bit. For example if a pixel's red plane value is a 10111011, and we want to store the information in least significant bit, at the worst situation the pixel value changes to 10111010, examinations shows that HVS cannot distinguish this alteration [15]. So we save our information into least significant bits of red plane of the pixel. If we change the LSB in a byte of an image, we either add or subtract one from the value it represents [27]. This means we can overwrite the last bit in a byte without affecting the colour.

In order to hide the message, data is first converted into byte format and stored in a byte array. The message is embedded in each bit into the LSB position of each pixel's red plane. It uses the first pixel (at spot 0) to hide the length of message (number of character). Suppose our original pixel as bits: (r7 r6 r5 r4 r3 r2 r1 r0, g7 g6 g5 g4 g3 g2 g1 g0, b7 b6 b5 b4 b3 b2 b1 b0) In addition, our character (bytes) has some bits: (c7 c6 c5 c4 c3 c2 c1 c0).

In our present work we place the character bits in LSB of red pixel only instead of placing it in red, green and blue LSB's. The purpose is to make the message much more secure. The message is stored in byte of red. So selected pixels are scattered and security of message is higher.

Then we can place the character bits in the lowest red pixel, next character bits in the next lowest red pixel, and so on. (r7 r6 r5 r4 r3 r2 r1 c0, g7 g6 g5 g4 g3 g2 g1 g0, b7 b6 b5 b4 b3 b2 b1 b0).

If we take an example of pixel (225,100,100) represented in binary form (11100001, 01100100, 01100100) into which to embed message character "d" having binary value 1100100 (ASCII value 100) then after embedding the first bit of "d" in pixel's red plane we can obtain New pixel as (224, 100,100) represented in binary (11100000, 01100100, 01100100).

Here we can notice that the pixel value of (225, 100,100) is changed to (224,100,100).From experiments it is observed that such changes will not have noticeable colour difference in the image. At worst case the decimal value of pixel may increase or decrease by one. Such change in the pixel value does not affect the image and is not detectable.

5.2 Embedding Algorithm

In the process of LSB encoding method, a random number generator is used to select the hiding points in the cover image. A pseudorandom number generator (PRNG) can be used to choose random pixels in which to embed the message [18]. A random number generator is used to randomly distribute and hide the bits of a secret message into the least significant bit of the pixels within a cover image. The transmitting and receiving end share the stego key [21, 36]. The stego key is usually a password used to seed a pseudo-random number generator to select pixel locations in an image for embedding the secret message and the output is a random sequence K_1……..K_n where n is the length of message bits [3, 23]. The sequence is then used by the sender to generate the sequence of pixel indices yi where,

$$y_1 = k_1 \qquad (3)$$

$$y_i = y_{i-1} + k_i, \quad i \geq 2 \qquad (4)$$

Message bit i would then be embedded into the LSB of the pixel y_i and thus the order in which the secret message bits are embedded would be determined pseudo randomly.

1) Read character from text file that is to be hidden and convert the ASCII value of the character into equivalent binary value into an 8 bit integer array.
2) Read the RGB colour image into which the message is to be embedded and Extract the red component of the host image.
3) Read the last bit of red pixel i.e. from RGB (8+8+8) bits.
4) Initialize the key which gives the random position of the red pixel to be processed for embedding.
5) Check the last bit is 0 or 1 and present in each pixel and store the cumulative sum into two integer variables suppose x1 and x2 holding total number of 0s & 1s respectively.
6) Pick up the message bit. If the message bit is zero (or one), check if x1 > x2 otherwise swap x1 and x2. Do the reverse operations for the message bit one (zero).
7) If value of flag is 0 then embed the data bit (either 0 or 1) to the last bit of colour red of the pixel otherwise if flag is 1 then inverse the data bit embed it) to the last bit of colour red of the pixel.
8) Write the above pixel to Stego Image File.

6. TECHNIQUE TO RETRIEVE HIDDEN MESSAGE

In the process of extraction against LSB encoding, the process first takes the key and this key takes out the points of the LSB where the secret message is randomly distributed [18]. Steganalysis of LSB searches the hidden bits of a secret message into the least significant bit of the pixels within a cover image using the random interval method [3, 9]. In decoding algorithm the stego-key must match i.e. the stego-key which was used in encoding should match because the stego-key sets the hiding points of the message in case of encoding [1, 36]. As in encoding process random number generator is used to select the hiding points in the cover image. The receiver can extract the embedded messages exactly using only the stego-key, which is used as the seed of the random number generator.

6.1 Message extraction algorithm

Since the receiver knows the seed k, he can reconstruct ki and therefore the entire sequence of pixel indices yi. In the random insertion method the location of the pixels depends on a stego key [10], whose size k should be in the range $n < k < I$, where n is the size of message and I is the size of cover image [3, 4]. The method commences by searching for the first prime number p that exceeds the key k, primitive root a, is then obtained, which is a number whose powers generate all the distinct integers from 1 to (p-1) in some permuted order [23]. Each power of this primitive root to generate these integers is called the discrete algorithm. This primitive root a, is then used to generate a set of random and distinct numbers, $yi = a^i \bmod p$, where i is the bit index of the secret message [3]. Bit I of the message then goes into LSB of pixel yi. In this way it is ensured that the bits of the secret message are inserted into distinct LSBs.

1) Open the Stego image file in read mode and from the Image file, read the RGB colour of each pixel.
2) Extract the red component of the host image.
3) Read the last bit of each pixel i.e. from RGB (8+8+8) bits.
4) Initialize the key that gives the position of the message bits in the red pixel that are embedded randomnly.
5) Read the last bit of pixel in colour red. Based on its value set integer variable checkflag 0 or 1.
6) For decoding, select the pixels using the same pseudo-random sequence. Check if the 2 pixels are within the pre-specified range α. If x1>x2, the message bit is zero (one) otherwise the message bit is one (zero).
7) If check flag is 0 then read the last bit of each pixel that is the LSB of colour red and put it directly in an array otherwise take the invert value of the last bit & put it on array.
8) Read each of pixels in this way & then content of the array converts into decimal value that is actually ASCII value of hidden character.
9) If terminating character's ASCII found print nothing otherwise print the corresponding character of the calculated ASCII value.

7. EXPERIMENTAL ANALYSIS

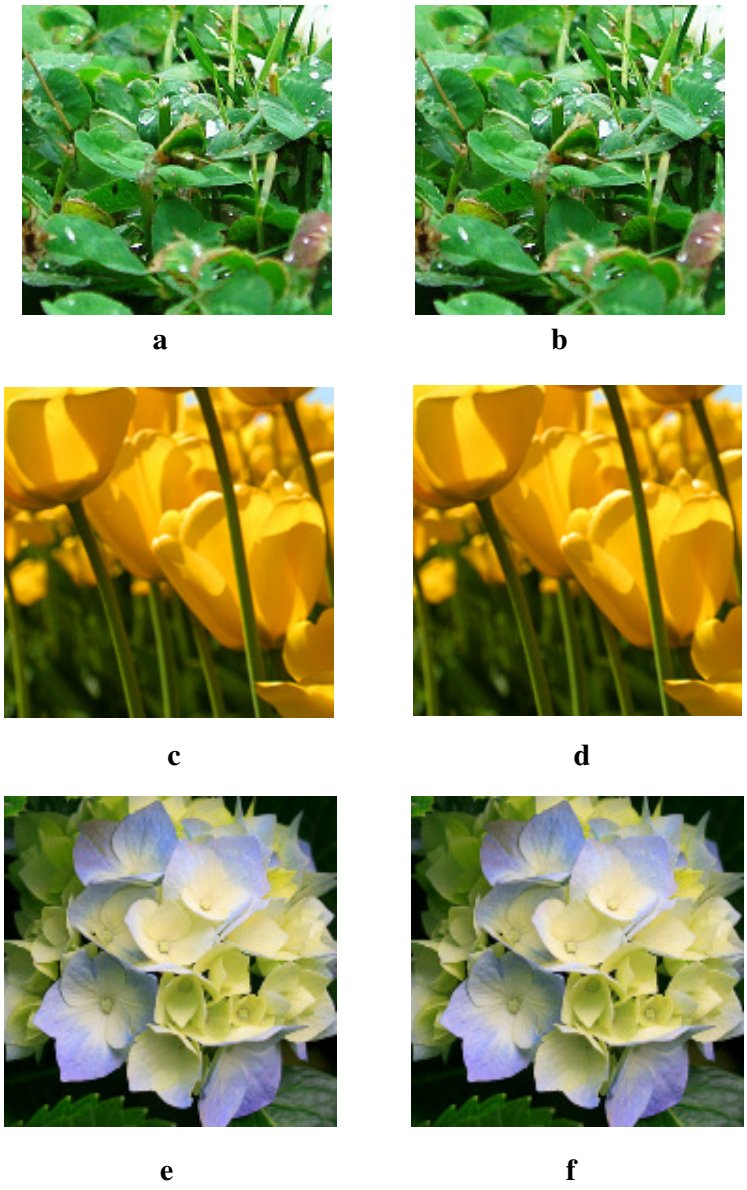

Fig. 3. clovers: (**a**) Original image (**b**) stego image, tulips: (**c**) Original image (**d**) stego image, flower: (**e**) Original image (**f**) stego image.

In our experimental procedure we selected test images ranging from 69 KB to 91 KB for testing the proposed method. The experimental work of the proposed method was conducted using Matlab 7.9.0. We created stego images by embedding messages of 1300 bytes to 2000 bytes into the images using random LSB replacement method. The difference of the stego-image can hardly be distinguished after embedding. It is observed that human visual system(HVS) can hardly differentiate the original image and the stego-image and also

the stego-images does not generate any suspicion as shown in figure 3 (a) and (b), (c) and (d), (e) and (f). As in present method a pseudorandom number generator (PRNG) is used to choose random pixels in an image by permuting the pixel indices with a secret key. Based on the system evaluation, the encoding and decoding processes can work fantastically well to hide and reveal information. The messages were successfully embedded into the cover images. The generated stego-image is sent over to the intended recipient. The whole idea of the proposed method is to model a technique that enables secure data communication between sender and receiver. Furthermore, the secret messages were also retrieved successfully without encountering any loss of data. Most importantly, the modification of the cover image is not perceptible on the stego image at all and thus arouses no suspicion to third parties.

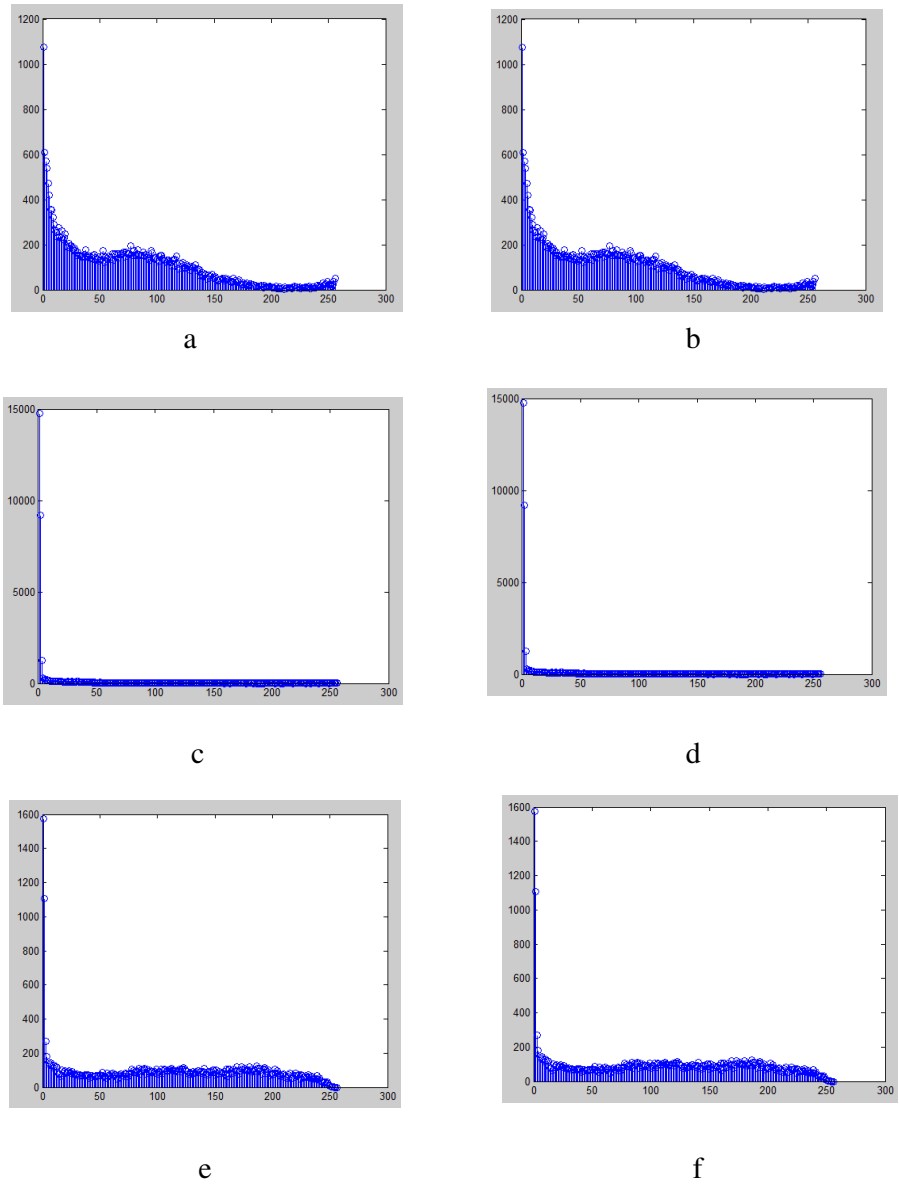

Fig. 4. clovers: (**a**)Histogram of figure 3a. (**b**)Histogram of figure 3b., tulips: (**c**) Histogram of figure 3c. (**d**) Histogram of figure 3d., flower: (**e**) Histogram of figure 3e. (**f**) Histogram of figure 3f.

The work not only aims to preserve the visual integrity of the image used for embedding but also the method should be free from statistical attacks because with the advances in steganalysis technique various statistical methods can detect modification in image bits. Statistical undetectability is one of the characters of a steganographic algorithm. Distortion analysis of stego images is carried out by studying distortion / similarity statistically. Distortion between two different images is measured by considering Mean Square Error (MSE) and Histogram Similarity (HS) [5, 20]. If the distortion occurs after the detection program is implemented then it is detected that the images may containing the hidden data otherwise not.

The statistical analysis compares the original image with the stego image based on histogram (first order statistics) of images. Comparing the histogram of the original channels, before and after embedding can give a clear idea of the security. There seems no difference in the value of the pixel intensity in the range 0 to 255 for the cover image and its stego image. The statistical change (histogram) between the original image and stego-image cannot be predicted as in figure 4 (a) and (b), (c) and (d), (e) and (f). The differences between the images before and after hiding the data cannot be sensed through histograms of the RGB channels [26].

Usually, the invisibility of the hidden message is measured in terms of the Peak Signal-to-Noise Ratio [7, 9]. To analyze the quality of the embedded texture image, with respect to the original, the measure of PSNR (peak signal to noise ratio) has been employed:

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \qquad (5)$$

where mean square error (MSE) is a measure used to quantify the difference between the cover image I and the stego (distorted) image I' [5,13]. If the image has a size of M * N then

$$MSE = \frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - I'(i,j)]^2 \qquad (6)$$

Generally speaking, when the payload increases, the MSE will increase, and this will affect the PSNR inversely. So, from trade-off it was found that MSE decrease causes PSNR increase and vice-versa [20]. PSNR is often expressed on a logarithmic scale in decibels (dB). PSNR values falling below 30 dB indicate a fairly low quality, i.e., distortion caused by embedding can be obvious; however, a high quality stego-image should strive for 45 dB and above [13].

TABLE I. MSE and PSNR values for the Original and Stego images

Original Image	Stego Image	MSE (%)	PSNR (dB)	Data Embedded (in bytes)	Data Extracted (in bytes)
clovers.bmp 80 KB	steg_ clovers.bmp 80 KB	0.0854	58.81	1548	1548
tulips.bmp 91 KB	steg_ tulips.bmp 91 KB	0.0747	59.39	2002	2002
flowers.bmp 69 KB	steg_ flowers.bmp 69 KB	0.0989	58.17	1300	1300

Our results as shown in Table I. indicate that embedding process introduces less perceptual distortion and higher PSNR. High PSNR value indicates distortion caused by embedding cannot be obvious [5]. Note that in case of *clovers* PSNR=58.17 dB means that the quality degradations could hardly be perceived by a human eye. The PSNR for images *tulips* and *flower* are 59.39 dB and 58.17 dB respectively. It was found that PSNR is constantly above 58 dB as seen in table I and also the size of the image have not changed after embedding. As in our method we tested the images ranging from 69 KB to 80 KB to embed 1300 to 2000 bytes. Furthermore, the secret information is also retrieved without encountering any loss of data.

8. CONCLUSION

This work deals with secure transmission of data. This system deals with implementation of security using steganography i.e., hiding large amount of information in an image without disturbing the image clarity and quality. By randomizing the embedding approach through the algorithm, the estimate of the cover statistics can be effectively disabled. So we have processed the LSB embedding with lossless compression. Besides that, our results indicate that the LSB insertion using pseudo random number generators are best in case of lossless compression. The outcome of the paper is to create a method that can effectively hide a message inside a digital image file. In this study, we have investigated on steganalysis process. The paper focuses on the approach like increasing the security of the message and reducing the distortion rate. With this approach, it is possible to embed message into LSB of red pixel of a colour image. We showed that our method provide effective data hiding by presenting the HVS perceptual test results, and statistical image properties.

9. ACKNOWLEDGEMENTS

One of the authors (Shamim Ahmed Laskar) gratefully acknowledges UGC for granting Research fellowship (Maulana Azad National Fellowship).

REFERENCES

[1] M. Conway, " Code Wars: Steganography, Signals Intelligence, and Terrorism", Knowledge Technology & Policy, Volume 16, Number 2, pp. 45-62, Springer, 2003.
[2] R. Radhakrishnan, M. Kharrazi and N. Menon, "Data Masking: A New Approach for Steganography?" Journal of VLSI Signal Processing 41, pp. 293–303, 2005.
[3] N. F. Johnson and S. C. Katzenbeisser, A survey of steganographic techniques, In Information Hiding, Artech House, Norwood, MA, 2000, pp. 43-78.
[4] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding-A Survey", Proceedings of the IEEE, 87(7), pp.1062-1078, July 1999.
[5] B E Carvajal-Gamez , F J Gallegos-Funes and J L López-Bonilla, Scaling Factor for RGB Images to Steganography Applications, Journal of Vectorial Relativity, vol.4, no.3, 2009, pp.55-65.
[6] A. Yadollahpour and H. M. Naimi, Attack on LSB Steganography in Colour and Grayscale Images Using Autocorrelation Coefficients, European Journal of Scientific Research, ISSN 1450-216X Vol.31 No.2, 2009, pp.172-183.
[7] E. Walia, P. Jain and Navdeep, An Analysis of LSB & DCT based Steganography, Global Journal of Computer Science and Technology, Vol. 10 Issue 1, pp 4-8. April 2010.
[8] N.F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen", IEEE, Computer, vol. 31, no. 2, pp. 26-34, Feb. 1998.
[9] W. Luo and F. Huang and J. Huang, "A more secure steganography based on adaptive pixel-value differencing scheme", Multimed Tools Appl (2011) 52: pp 407–430, Springer , 2011.
[10] N. Dedic, G. Itkis, L. Reyzin, and S. Russell, " Upper and Lower Bounds on Black-Box Steganography", J. Cryptol. (2009) 22: pp 365–394, Springer , 2009.

[11] R. Chandramouli and N. Memon, "Analysis of LSB based Image Steganography Techniques," Proceedings of International Conference on Image Processing, vol. 3, pp. 1019-1022, October 2001.

[12] M. Kaur, S. Gupta, P. S. Sandhu and J. Kaur, "A Dynamic RGB Intensity Based Steganography Scheme", World Academy of Science, Engineering and Technology 67, pp 833-836, 2010.

[13] G. Ulutas , M. Ulutas and V. Nabiyev, "Distortion free geometry based secret image sharing", Procedia Computer Science 3 (2011) 721–726 , Elsevier Inc, 2011.

[14] Neeta, Deshpande, Kamalapur Snehal, and Daisy Jacobs. "Implementation of LSB steganography and its evaluation for various bits." In Digital Information Management, 2006 1st International Conference on, IEEE, 2006, pp. 173-178.

[15] E. T. Lin and E. J. Delp, "A Review of Data Hiding in Digital Images", Proceedings of the Image Processing, Image Quality, Image Capture Systems Conference (PICS '99), pp. 274-278, April 1999.

[16] E. J. Baker, "Steganography in Images by Using Intersecting Planes", Eng. & Tech. Journal, Vol. 29, No. 7 , pp. 1265-1275, April, 2011.

[17] C. Hosmer "Discovering Hidden Evidence", Journal of Digital Forensic Practice, Vol. 1, No. 1, pp. 47–56, Taylor & Francis Group, ISSN: 1556-7281, 2006.

[18] S. Manchanda, M. Dave and S. B. Singh, Pseudo random numbers Based Methods for Customized and Secure Image Steganography, IRMA International Conference, 2007, Idea Group Inc., 1608-1614.

[19] E. Walia, P. Jain and Navdeep, "An Analysis of LSB & DCT based Steganography", Global Journal of Computer Science and Technology, Vol. 10 Issue 1 (Ver 1.0), pp 4-8, April 2010.

[20] Z. Ni, Y-Q. Shi, N. Ansari, and W. Su, "Reversible Data Hiding", IEEE Transactions on Circuits and Systems for Video Technology, VOL. 16, No. 3, pp. 354-362, March 2006.

[21] S. Venkatraman, A. Abraham and M. Paprzycki, "Significance of Steganography on Data Security", Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), 2004, 0-7695-2108-8/04.

[22] J. Fridrich and M. Goljan, "Digital image steganography using stochastic modulation", SPIE Symposium on Electronic Imaging, San Jose, CA, 2003.

[23] Tuomas Aura, "Practical Invisibility in Digital Communication", in Proceedings of the First International Workshop, Cambridge, UK, May-June, 1996.

[24] Elke Franz, Anja Jerichow, Steffen Moller, Andreas Pfitzmann, Ingo Stierand, "Computer Based Steganography: How It Works and Why Therefore Any Restrictions on Cryptography Are Nonsense, at Best", in Proceedings of the First International Workshop, Cambridge, UK, May-June 1996.

[25] J. Fridrich, M. Goljan and R. Du, "Lossless Data Embedding-New Paradigm in Digital Watermarking", Special Issue on Emerging Applications of Multimedia Data Hiding, Vol. 2002, No.2, pp. 185-196, February 2002.

[26] T. Zang and X. Ping, Reliable Detection of LSB Steganography based on the Difference image Histogram, Acoustics, speech and signal Processing , ICASSP, International Conference on Vol.3,pp-III-545-8, 2003.

[27] W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Techniques for data hiding", IBM System J. 35 (3–4), pp. 313– 336. 1996.

[28] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", 2nd edition, Prentice Hall, Inc, 2002.

[29] K. Bailey and K. Curran. "An evaluation of image based steganography methods." Multimedia Tools and Applications, Vol. 30, no. 1, 2006, Springer, 55-88.

[30] Chan, Chi-Kwong, and L. M. Cheng. "Hiding data in images by simple LSB substitution." Pattern Recognition 37, no. 3, 2004, 469-474.

[31] S. Song, J. Zhang, X. Liao, J. Du, and Q. Wen. "A Novel Secure Communication Protocol Combining Steganography and Cryptography."Procedia Engineering 15 (2011): 2767-2772.

[32] S. Dumitrescu, X. Wu, and N. Memon. "On steganalysis of random LSB embedding in continuous-tone images." In Image Processing. 2002. Proceedings. 2002 International Conference on, vol. 3, pp. 641-644. IEEE, 2002.

[33] N.Santoshi and B.Lokeswara Rao, A Secure and Lossless Adaptive Image Steganography with Mod-4 LSB Replacement Methods Using Image Contrast, International Journal of Scientific & Engineering Research, Volume 3, Issue 8, August-2012, 1-9.

[34] P. Mohan Kumar and K. L. Shunmuganathan, Developing a Secure Image Steganographic System Using TPVD Adaptive LSB Matching Revisited Algorithm for Maximizing the Embedding Rate, Information Security Journal: A Global Perspective, vol. 21, Taylor & Francis, 2012, 65–70.

[35] F. A.P. Petitcolas, R. J. Anderson, and M. G. Kuhn, Attacks on copyright marking systems, In Information Hiding, Springer Berlin/Heidelberg, 1998, pp. 218-238.

[36] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, Techniques for data hiding, IBM Systems Journal , vol. 35, no. 3 & 4, 1996, pp. 313-336.

[37] Y. Kuen Lee and L. Hwei Chen, High capacity image steganographic model, IEE Proc.-Vis. Image Signal Process., Vol. 147, No. 3, June 2000, 1-15.

[38] S. Pujari and S. Mukhopadhyay, An Image based Steganography Scheme Implying Pseudo-Random Mapping of Text Segments to Logical Region of Cover Image using a New Block Mapping Function and Randomization Technique, International Journal of Computer Applications (0975 – 8887) Volume 50 , No.2, July 2012, 40-46.

[39] M. Mohamed, F. Al-Afari and M. Bamatraf, Data Hiding by LSB Substitution Using Genetic Optimal Key-Permutation, International Arab Journal of e-Technology, Vol. 2, No. 1, January 2011, 11-17.

[40] N. Jain, S. Meshram and S. Dubey, Image Steganography Using LSB and Edge – Detection Technique, International Journal of Soft Computing and Engineering, Volume-2, Issue-3, July 2012, 217-222.

[41] Vismita Nagrale, Ganesh Zambre and Aamir Agwani, "Image Stegano-Cryptography Based on LSB Insertion & Symmetric Key Encryption", International journal of Electronics and Communication Engineering & Technology (IJECET), Volume 2, Issue 1, 2011, pp. 35 - 42, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.

[42] Nagham Hamid, Abid Yahya, R. Badlishah Ahmad and Osamah M. Al-Qershi, "An Improved Robust and Secured Image Steganographic Scheme", International journal of Electronics and Communication Engineering & Technology (IJECET), Volume 3, Issue 3, 2012, pp. 22 - 33, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.

[43] Reena M Patel and D J Shah, "Concealogram: Digital Image in Image using LSB Insertion Method", International journal of Electronics and Communication Engineering & Technology (IJECET), Volume 4, Issue 1, 2013, pp. 230 - 235, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.