ISSN:0975-9646

LSB based Steganography using Bit masking method on RGB planes

Shahin Shabnam
Dept. of Computer Science
Assam University, Silchar
Assam, India

Prof **K Hemachandran**
Dept. of Computer Science
Assam University, Silchar
Assam, India

Abstract— **Steganography hides the existence of information that needs to be exchanged or transferred through some public media like internet. This is done by embedding the secret message in some innocent cover-medium, like image. In addition to the transfer of information, there exists the concern for capacity, security and robustness of the method. In this paper we present a novel, effective data hiding scheme that embeds a message into a cover image, taking the advantage of the 24-bits, in each pixel of the RGB image. The method does not use any secret key and depending on the message bit status (prime or unprime) the position of bit to be embedded in the image is determined. The technique is analyzed in terms of Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) and compared with similar results of the earlier work and that the proposed technique gives better results.**

Keywords— ***Steganography, RGB, Bitmaps, AND, OR, Bit masking, MSE, PSNR***

I. INTRODUCTION

With the advent of Internet, the communication among people has been facilitated to a great extent. But it also enabled illegal users to access the data being transmitted. Hence, to protect important data from illegal access, systems have been developed to encrypt or hide the data in some form, prior to their transmission [1].

In encryption the encrypted data exists in meaningless form and may attract the attention of interceptors, who may eventually succeed in decrypting. Another drawback may be the existence of the important data to be transmitted in noticeable form.

The steganography uses the fact that the data to be communicated secretly is hidden in the media. It is the art of camouflaging the secret data in an innocent container. There may be different file formats that can be used as the container like audio, video and image. But the selection of a particular steganographic medium is important as it significantly affects the steganographic system design and security [2].

Digital images are the most popular cover medium being used due to its frequency of use over the Internet. Further, images potentially contain much visual redundancy so that they can provide large capacity to hide secret data by exploiting the insensitivity of the human visual system. Redundancy can be described as the bits of a file that offer accuracy more than needed for the object [3]. Hence they can be altered without this change being noticed easily.

The quality of a good steganographic algorithm is described by its ability to fulfill some specific and essential requirements like :- Imperceptibility, Robustness and Payload/Capacity [4]

- The imperceptibility is the most important requirement of a steganographic system, as the strength of steganography system depends on its ability to be unnoticed by the human senses (visually or acoustically).
- Robustness defines how strong the used steganographic technique exhibits against the changes. It measures the capability of the embedded secret data to endure different types of intentional and unintentional modifications. It is necessary that a steganography algorithm does not leave any mark in the covert medium in order to be able to avoid and pass by statistical analysis without being detected.
- Payload/Capacity is the size of embedded data that can be hidden into a particular innocent cover medium relative to the size of this medium.

Steganography is often confused with cryptography which merely hides the integrity of the information so that it makes no sense to any third party other than the creator and recipient. Thus, if given enough time, someone can eventually decrypt the data using cryptanalysis.

However, the technique of steganography is similar to digital watermarking with a big distinction that, watermarking focuses on ensuring nobody remove or alter the content of watermarked data, which is obvious in existence. Whereas, steganography hides the mere existence of data. [5]

II. RELATED WORKS

LSB steganography is the most widely used steganographic techniques due to its simplicity and straightforward approach. The secret message is stored in the least significant bit plane of the cover file. This technique becomes difficult to detect in the sense that small amount of changes are being made in the cover image. Different authors have used the simple LSB techniques where LSB of pixels are replaced by the secret bits [10] [11]. This concept may sometimes pose some serious security issues. Venkatraman et al. [12] had used the concept of attaching stego-key with the embedding process to make the retrieval of the secret information somewhat difficult. Variations to the basic LSB based

steganography can be seen in [13] [14]. Perceptual transparency is achieved by embedding the secret data beside the edges of object [15]. Adnan et al. have used pixel indicator technique [16] to increase the capacity aspect of LSB technique. In this, one of the RGB channels of cover image is selected and 2 LSBs of secret data are embedded in it. In [17] the difference of the two consecutive pixels taken as a block is determined and replaced with the secret bits similar to the one of cover image. Wu et al. proposed a palette modification scheme by embedding one message bit in each pixel of the image [18]. Mehdi and Mureed improved the Kekre's algorithm on LSB method and increased the embedding capacity while retaining the quality of stego image [19]. Here 5 LSBs of each pixel with lower intensity are used for embedding and maintaining a matrix to determine the positions where all of the 5LSBs utilized and where less than that for embedding. Further the message bits are XORed before embedding for extra security. However this method has increased the computational complexity many folds to achieve security with a somewhat degraded picture quality.

III. PROPOSED SYSTEM

The following figure describes the basic process of steganography as shown in fig1.

Stego-image is referred as an image that is obtained by embedding secret data into cover image. Thus the authenticated data is provided with confidentiality and integrity. The embedding process hides the secret message by using the stego-key. After embedding is finished, the cover media can be transmitted to the receiver. At the receiving end, the receiver having the proper stego-key and decryption key, can extract the secret message from the cover medium. The success of steganography is dependent on the secrecy of the cover medium. Once the cover medium is public then the success depends on the robustness of the algorithm used.

Image steganography can be categorized into two broad categories, namely-

- Spatial domain
- Transform domain

Spatial domain techniques embed message or the secret data in the intensity of the pixels of cover image directly [6]. Whereas, in the transform domain, images are first transformed and then the message is embedded in the image [7]. An image is an array of numbers representing light intensities at various points. They are either stored in 8-bit or 24-bit files which are relatively large and uncommon on the internet and may attract attention while transmission. File compression is thus required before transmitting such file. Now compression is of two types [8]-

- Lossy compression
- Lossless compression

Lossless compression does not remove any information from the original image but represent the data in some mathematical formulae and remove the redundant data. The image integrity is hence maintained.

Lossy compression on the other hand saves space by discarding excess image data from the original image. Details that are removed are too small for the human eye to differentiate [9] resulting in close approximation of the original image. As images are considered as good cover medium due to the presence of visually redundant pixel, they are used often. Images can be classified into three types: Binary (Black and White), Gray scale and Red Green Blue (RGB). Binary image having one bit value per pixel (0 or 1), Gray scale having 8 bits per pixel (00000000-11111111) and RGB with 24 bits per pixel. Embedding in binary is tougher in terms of security as modification in single bit gets easily noticeable whereas in RGB it is more suitable having more numbers of information present that can be modified without making it easily noticeable.

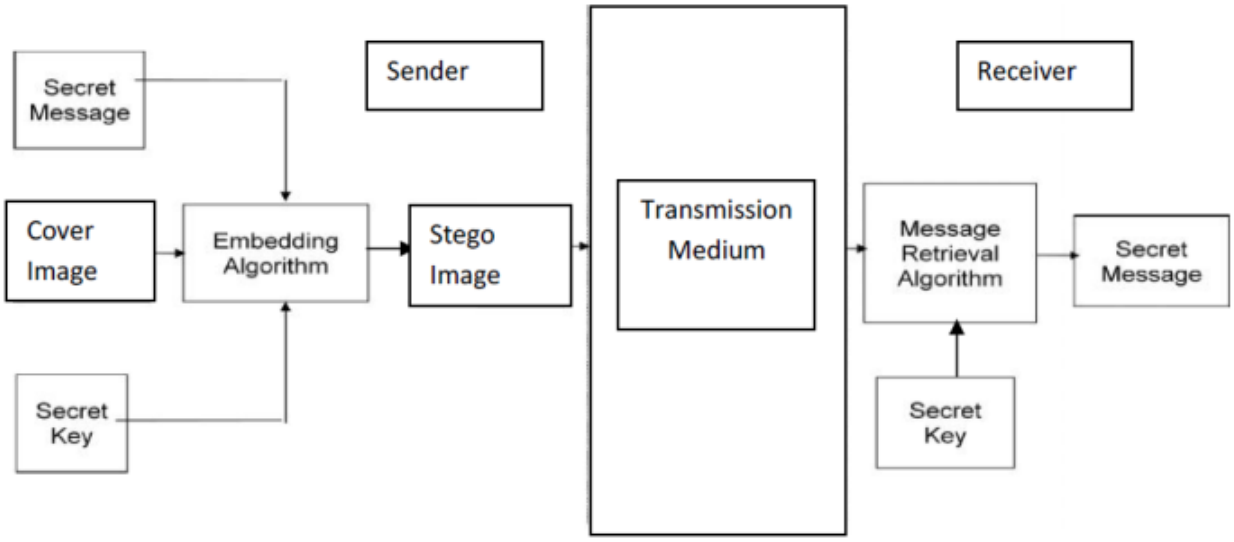

Fig. 1. Block diagram of Steganographic process

In the proposed work using the LSB embedding, the least significant bit of the cover image is not simply replaced by the information but depending on the information bits, rules are set for changes.

We start by taking the two types of files, namely the cover image file and the secret message file. The image size and the message length are being determined. The cover image used is colored and hence the image is being split into 3 matrices (i.e :- Red, Green, Blue) and the pixels are extracted. The secret message to be hidden is converted into binary. Now taking the secret data bits, we first determine if the bit is 0 or 1 and accordingly we take the first matrix of Red, if the message bit is 0, we bit AND the first bit of Red matrix with 254th color intensity of the same cover image else if the secret bit is 1, then the cover bit of the Red matrix is bit OR with the 1st color intensity of the cover image. Now taking the 2nd message bit, the same procedure is applied but for the Green matrix and the 3rd message bit for the Blue matrix. Completing the process for the first three secret bits, the fourth one is tested again for the next pixel bit position of the Blue matrix in the cover image, then the Green and Red plane and the loop continues iteratively for the whole text message bits in all the three matrices as (RGB->BGR->RGB->BGR….). On embedding the whole text message, the stego image is obtained by modifying the bit values of the cover image.

In this procedure we have exploited the property of bit mask of OR and AND binary operation in the color planes to determine if the secret message bit is set(1) or clear(0). So our process is secure in the sense that one cannot determine whether we are using AND or OR operation to the pixel bits as this depends on our secret message bit values which cannot be known at the initial stage of detection. The pixel intensities changes depending on the operation performed on them. The algorithm utilizes 3 bits per pixel for embedding which are extracted bit by bit taking each of the RGB plane of a pixel and performing the modulo division by 2 sequentially (RGB->BGR->RGB->BGR…..). The stego image was stored in .bmp format to avoid any loss of data during the decoding. A BMP file format also called bitmap or DIB file format (for device-independent bitmap), is an image file format used to store bitmap digital images. JPEG file formats gets changed if any change in pixel occurs and easily identified by the third party that some information has been hidden in the image. Even if large amount of data is to be hidden in the image JPEG is not the correct choice.BMP file formats are best suited to hide quite large message. BMP is most suitable for applications, where the first focus is on the amount of information to be transmitted and then on the secrecy of that information [20].

Message – Thanks

Symbol	Binary
T	01010100
h	01101000
a	01100001
n	01101110
k	01101011
s	01110011

We are to send this message over a network by hiding it in a cover RGB image. Let the values of R, G and B for the first pixel of image be given in Fig2

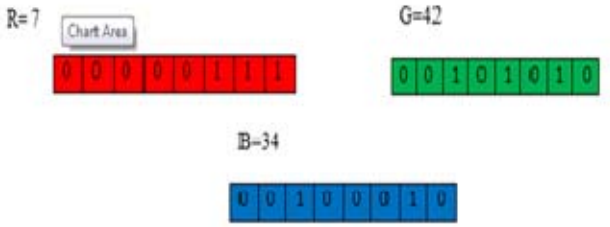

Fig2

Now to embed the message bit by bit we take the first letter ‘T’ i.e. its binary value and the first bit

0	1	0	1	0	1	0	0

According to rule we are to AND the Red component bits with the 254th color intensity of the image that is 11111110 and the resultant will be the corresponding Red component of the stego image as shown in Fig3.

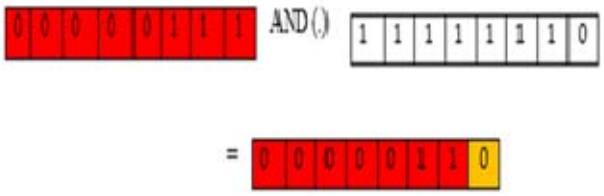

Fig3

After the performance of the AND operation, it is noticed that manipulation in the LSB of the Red- plane is being done.

Coming to the second message bit we find-

0	1	0	1	0	1	0	0

According to rule we are to OR the Green component bits with the 1st color intensity of the image that is 00000001 and the resultant will be the corresponding Green component of the stego image as shown in Fig4.

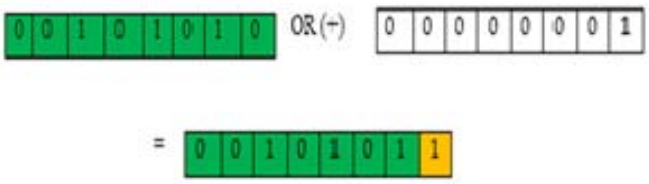

Fig4

After the performance of the OR- operation, it is noticed that manipulation in the LSB of the Green- plane is being done.

Similarly, for the third message bit we get it to be-

0	1	0	1	0	1	0	0

This is now used in the Blue component bits, following the rule as mentioned and shown in Fig5.

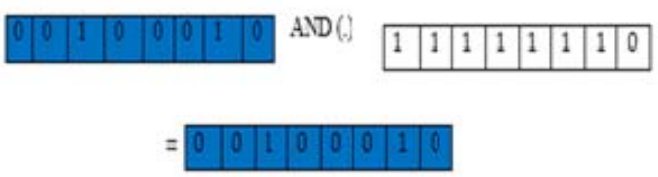

Fig5

No changes is the Blue plane is noticed, hence no change in the color intensity is done.

This is continued for the entire text message bit by bit in all the R, G & B channels of the cover image pixels sequentially hence obtaining the stego image.

IV. EXPERIMENTAL RESULTS

The proposed system has been implemented using MATLAB environment and tested on different images which are available in the public domain. The performance of the steganographic method is evaluated on the basis of Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR)

The MSE is given as:

$$MSE = 1/H * \sum_{i,j}(C(i,j) - S(i,j))^2 \quad (1)$$

where H is the dimension of the image
C (i, j) is the intensity of cover image and
S (i, j) is the intensity of stego image
The PSNR is given as:

$$PSNR = 10 \log10 L/MSE \quad (2)$$

where L is the peak signal level of image (=255 in case of color image)

Different color images of 512*512 sizes and secret message "Abraham Linclons letter to his son's teacher" are taken and the above process is performed to get the stego image.

MSE and PSNR are determined for the stego images and compared the result with an existing LSB embedding technique [19].

Cover Image	Embedded data (bytes)	Existing Algorithm (19)		Proposed Algorithm	
		MSE	PSNR	MSE	PSNR
Lena	1785	0.0174	65.7180	0.0026	73.9249
Baboon	1785	0.0070	69.6609	0.0026	73.9298
Pepper	1785	0.0723	59.5402	0.0026	73.9734

TABLE1: VALUES OF MSE & PSNR FOR COMPARISONS OF THE ALGORITHMS

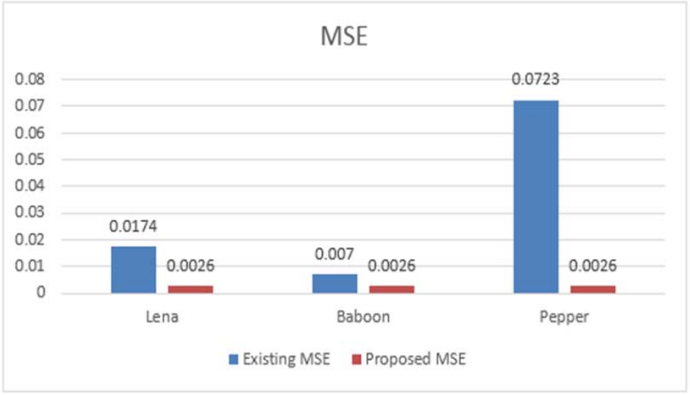

TABLE2: HISTOGRAM SHOWING THE DIFFERENCE IN MSE

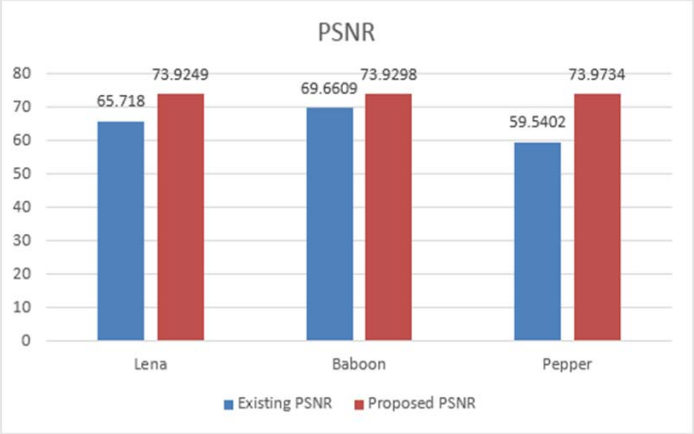

TABLE3: HISTOGRAM SHOWING THE DIFFERENCE PSNR

This shows the improvement in the embedding of the text message in various cover image while maintaining the quality of the image.

CONCLUSION

The process of steganography has been performed on MATLAB platform successfully with larger cover images selected for larger text message to be embedded. The stego images have been further tested successfully for steganalysis with the two most common quantities of LSB detectors, namely- Sample Pairs and Triples Analysis. So our process is secure in the sense that one cannot determine whether AND or OR operation is used on the pixel bits as this depends on our message bit values and the sequence of plane used for embedding. The pixel intensities change accordingly hence forming stego image. Given to choose a steganographic system, an individual will definitely opt for a secured system rather than a less secure large carrier of crucial message. And a secured system automatically tends to be a robust one. Results shows better performance in terms of quality of the stego image obtained.

REFERENCE

1. W. Stallings, Cryptography and Network Security: Principles and Practice, third ed., Pearson Education, New Jersey, 2003.
2. Katzenbeisser, S., & Petitcolas, F. (2000). Information hiding techniques for steganography and digital watermarking. Artech house.
3. Currie, D.L. & Irvine, C.E., "Surmounting the effects of lossy compression Steganography", *19th National Information Systems Security Conference*, 1996.
4. Morkel, T., Eloff, J. H., & Olivier, M. S. (2005, June). An overview of image steganography. In ISSA (pp. 1-11).
5. Cole, E., & Krutz, R. D. (2003). Hiding in plain sight: Steganography and the art of covert communication. John Wiley & Sons, Inc..
6. Johnson, N. F., & Jajodia, S. (1998, January). Steganalysis of images created using current steganography software. In Information Hiding (pp. 273-289). Springer Berlin Heidelberg
7. Lee, Y.K. & Chen, L.H., "High capacity image steganographic model", Visual Image Signal Processing, 147:03, June 2000
8. Moerland, T., "Steganography and Steganalysis", Leiden Institute of Advanced Computing Science
9. "Reference guide: Graphics Technical Options and Decisions", http://www.devx.com/projectcool/Article/19997
10. Kessler, G. C. (2004). An overview of steganography for the computer forensics examiner. Forensic Science Communications, 6(3), 1-27.
11. Artz, D. (2001). Digital steganography: hiding data within data. internet computing, IEEE, 5(3), 75-80.
12. S. Venkatraman, A. Abraham, M. Paprzycki, "Significance of Steganography on Data Security", International Conference on Information Technology: Coding and Computing (ITCC'04), 5-7 April 2004.
13. K. Bailey, K. Curran, "An Evaluation of Image Based Steganography Methods", Multimedia Tools & Applications, Vol. 30, No. 1, pages 55-88, July 2006.
14. Karim, S. M., Rahman, M. S., & Hossain, M. I. (2011). A new approach for LSB based image steganography using secret key. In Computer and Information Technology (ICCIT), 2011 14th International Conference on (pp. 286-291). IEEE.
15. Hempstalk, K. (2006). Hiding behind corners: Using edges in images for better steganography.
16. Gutub, A. A. A. (2010). "Pixel indicator technique for RGB image steganography." Journal of Emerging Technologies in Web Intelligence, 2(1), 56-64.
17. Wu, D. C., & Tsai, W. H. (2003). A steganographic method for images by pixel-value differencing. Pattern Recognition Letters, 24(9), 1613-1626.
18. Wu, M. Y., Ho, Y. K., & Lee, J. H. (2004). An iterative method of palette-based image steganography. Pattern Recognition Letters, 25(3), 301-309.
19. Hussain, M., & Hussain, M. (2010, June). Pixel intensity based high capacity data embedding method. In Information and Emerging Technologies (ICIET), 2010 International Conference on (pp. 1-5). IEEE.
20. Kumar, B. R., Suresh, K., Basheer, S. K., & Kumar, M. R. K. (2012). Enhanced Approach to Steganography Using Bit planes". International Journal of Computer Science and Information Technologies, 3(6), 5472-5475.