A Novel Keyless Algorithm for Steganography

Supriya Rai and Ruchi Dubey

Abstract—**Steganography is one of the most powerful tools for information hiding. In this paper, we have modified least significant bit (LSB) substitution method for data hiding. Conventional LSB technique uses the least significant bit of consecutive pixels for embedding the message which draws suspicion to transmission of a hidden message. If the suspicion is raised, then the goal of steganography is defeated. Still LSB technique is the most widely used as it is simple. In our implementation pixels to be substituted with information are selected randomly which makes it superior to the conventional approach. The robustness of the algorithm is further increased by using keyless steganography. This paper proposes a novel technique to hide information in a 24 bpp RGB image using modified LSB substitution method.**

Index Terms--**bits per pixel (bpp), least significant bit (LSB), pixel, RGB, steganography.**

I. Introduction

Steganography means concealment of information in such a way that no one, apart from the sender and intended recipient, suspects the existence of the information.

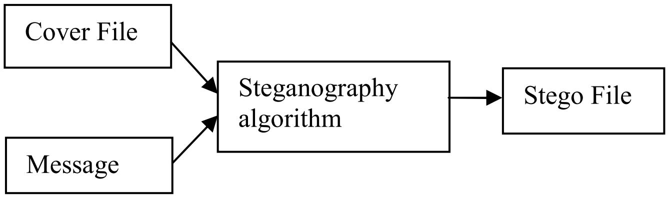

Fig. 1. Steganography process.

Section II discusses the existing work in the field of LSB substitution method for steganography. Section III gives out the details of the proposed technique. It is a keyless random location selection algorithm. In Section IV, the proposed technique is implemented. The algorithm is implemented on a 24 bpp RGB bmp image. Section V compares the two images graphically. Section VI shows the experimental results after implementation. Section VII concludes the paper highlighting the advantages of the proposed technique.

II. Previous Work on LSB Technique

Steganography has been used for information hiding since ancient times. The first recorded use of steganography can be traced back to 440 BC when Demaratus sent a warning about a forthcoming attack to Greece by writing it directly on the wooden backing of a wax tablet before applying its beeswax surface.

In modern times steganographic technologies have been an important part of the future of security and privacy on open systems such as internet. Much research work has been done in this field till date. An introduction to steganographic technologies and their applications is given in [1]. The most common method for steganography is LSB insertion method [2]. LSB method comes under substitution techniques of steganography. For hiding maximum data more than one LSB can be modified. 4 LSB substitution method which modifies last four bits of a pixel is one such work, see [3]. The LSB substitution method is a versatile technique for steganography and can be used for various file formats see [4]. Security of hidden message can be increased manifold by complementing steganography with cryptography [5]-[6]. Random LSB insertion method in which the secret data are spread out among the image in a random manner requires a secret key to retrieve this data. Steganography with secret key along with cryptography [7] has been implemented for random insertion of data bits. In the following section a novel technique has been proposed for random LSB insertion which uses 'no key'.

III. Proposed Technique

Reference [8] shows that we can hide the message by substituting the LSB of each pixel with information bits in 24 bpp RGB image. 24 bpp RGB image is a 24 bit depth colour image using RGB colour model. 24 bit refers to 8 bit for each RGB colour channel, i.e. 8 bits for red, 8 bits for green and 8 bits for blue. This implies that we can store three bits of information per pixel at the LSB of RGB. By changing the LSB of RGB values of each pixel, we may get maximum 2X2X2=8 different shades. This change in the pixel bits will be indiscernible to the human eye.

Divide the image into appropriate number of parts. Say we are using a 1024X768 image. We divide the image in 3 rows and 4 columns as shown in the figure. Each block has a dimension of 256X256 pixels.

I	II	III	IV
V	VI	VII	VIII
IX	X	XI	XII

Fig. 2. Image divided into 256X256 blocks.

Let us say the first pixel to be modified is a shade of pink which lies in block XI and has value 250:218:221 i.e. 11111010:11011010:11011101

Supriya Rai is with the Department of Instrumentation and Control, Netaji Subhas Institute of Technology, New Delhi-110078, India (e-mail: raisupriya90@yahoo.com).

Ruchi Dubey is with the Department of Information Technology, Galgotias College of Engineering and Technology, Greater Noida-211004, India (e-mail: ruchi.dubey29@yahoo.com).

978-1-4673-0455-9/12/$31.00 ©2012 IEEE

The LSB values of RGB are 0:0:1 which are substituted with three information bits. Therefore, each pixel can carry 3 bits of data. The altered pixel is used to select the next pixel to be modified. The next pixel may lie in any of the next two consecutive blocks i.e. XII or I in this case.

Each block is a grid of 256 pixels in horizontal direction and 256 pixels in vertical direction i.e. 256X256. To get an address in this grid we need 16 bits as they are required to represent 256X256 in binary, i.e. 8 bits for horizontal 256 pixels and 8 bits for vertical 256 pixels. One extra bit is needed to select the next block from the two subsequent blocks as it can have two values 0 or 1. Here, when our current location is in block XI, 0 represents block XII and 1 represents block I.

Note that a particular space of the image will have different shades of same colour. Therefore, there would be very little change in most significant bits of RGB values. If these bits form most significant bits of the next location which is chosen from the current pixel location then same location may be pointed repeatedly. This is due to the fact that there is very little change in most significant bits of RGB values within an area as the colour intensity of that area, which may be an entire block, is nearly the same. For instance RGB values of dodger blue 1 is 00011110:10010000:11111111 and dodger blue 2 is 00011100:10000110:11101110. Notice that most significant bits of the two colours are the same.

To avoid this and to increase overall randomness, 8 bits for the horizontal 256 pixels are derived from LSB to LSB-4 of red and LSB-3 to LSB-5 of blue (LSB of red being the most significant bit of next location), 8 bits for the vertical 256 pixels are derived from LSB to LSB-4 of green and LSB to LSB-2 of blue (LSB of green being the most significant bit of next location) and the bit for selecting from subsequent two blocks is derived from LSB of blue of the current pixel.

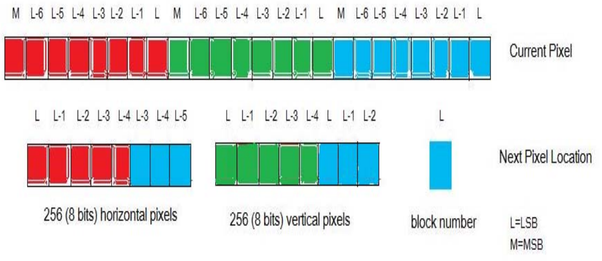
M L-6 L-5 L-4 L-3 L-2 L-1 L M L-6 L-5 L-4 L-3 L-2 L-1 L M L-6 L-5 L-4 L-3 L-2 L-1 L
Current Pixel
L L-1 L-2 L-3 L-4 L-3 L-4 L-5
L L-1 L-2 L-3 L-4 L L-1 L-2
L
Next Pixel Location
256 (8 bits) horizontal pixels
256 (8 bits) vertical pixels
block number
L=LSB
M=MSB

Fig. 3. Next pixel location derived from current pixel.

Even after following the above procedure there is a slight possibility that the new location generated was already modified. To fix this, store all the modified pixel locations in a file which is checked before each modification to avoid location clash.

IV. Implementation

The first pixel to be modified is known to sender as well as intended receiver. From the value of this pixel we determine the location of the next pixel to be modified.

Let us take an example demonstrating the technique: We have used 1024X768 bmp image. ASCII describes a communications system where 7-bit words represent printable symbols and control codes.

The message to be embedded is 'meetatnine'.

ASCII of m= 109= 1101101

ASCII of e= 101= 1100101

Three pixels are required to store 7 bits of one letter or symbol. Only red colour of the third pixel is used. Green and blue colours remain unaltered. Say the first pixel to be modified is (1,1) which is in block I.

Original pixel value: 01110000:10010001:11000110

Modified pixel value: 01110001:10010001:11000110

Horizontal location of next pixel= 10001000=136

(LSB to LSB-4 of red and LSB-3 to LSB-5 of blue)

Vertical location of next pixel= 10001011=139

(LSB to LSB-4 of green and LSB to LSB-2 of blue)

Block number of next pixel=0

(LSB of blue)

As the current block is I and block number for next pixel is 0, therefore, the next block in which pixel is to be altered is II. We have divided the image in 256X256 blocks. End coordinates of the first block are (1,1), (256,1), (256,256) and (1,256). As we are moving from block I to block II, thus, the next pixel location is (136,256+139) = (136,395).

TABLE I
Observation Table

Pixel location	Original pixel value		Modified pixel value	message bit	Messag e
(1,1) Block I	R	01110000	01110001	1	m
	G	10010001	10010001	1	
	B	11000110	11000110	0	
(136, 395) Block II	R	10111000	10111001	1	
	G	11001101	11001101	1	
	B	11011110	11011110	0	
(158, 691) Block III	R	11011101	11011101	1	
	G	11101010	11101010	--	
	B	11110000	11110000	--	
(187, 848) Block IV	R	11101011	11101011	1	e
	G	11110001	11110001	1	
	B	11101101	11101100	0	
(469, 137) Block V	R	00110011	00110010	0	
	G	00011101	00011101	1	
	B	00010000	00010000	0	
(330, 340) Block VI	R	01010001	01010001	1	
	G	01001101	01001101	--	
	B	01000001	01000001	--	

From the above observation we can see that only four pixels are changing, that too by a single shade. In pixel location (1,1) only red value is changed after embedding message bits. The original and modified values of the pixel are 112:145:198 and 113:145:198 which are both very close shades of blue.

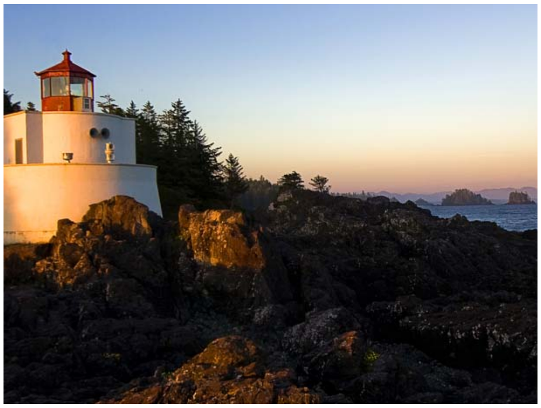

Fig. 4. Original image of a light house

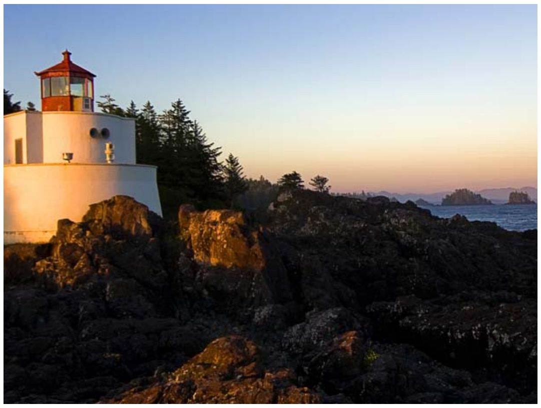

Fig. 5. Modified image containing message 'meetatnine'

There is always a possibility of location clash i.e. the new location generated was previously modified. To make sure that no clash occurs a file is created that stores all modified pixel locations. This file is checked before each modification to avoid overwriting the already modified pixel. If at some stage clash occurs then shift the current location to the next pixel and modify it. At the time of steganalysis this file is again created to keep track of clashes. Number of modified pixels should be known for steganalysis. This number is stored in any pixel location known to the intended receiver. By applying reverse algorithm message is retrieved.

V. Performance Analysis

The accomplishment of the proposed algorithm can be highlighted by studying colour histograms of original and modified images. Red, green and blue planes of the image are plotted separately. X axis of the histogram represents pixel intensity whose range is [0,255]. Y axis of the histogram represents the number of pixels of a particular value or intensity. So, if there is an image that contains 50 pixels with an intensity of 0 then x value will be 0 and y value will be 50.

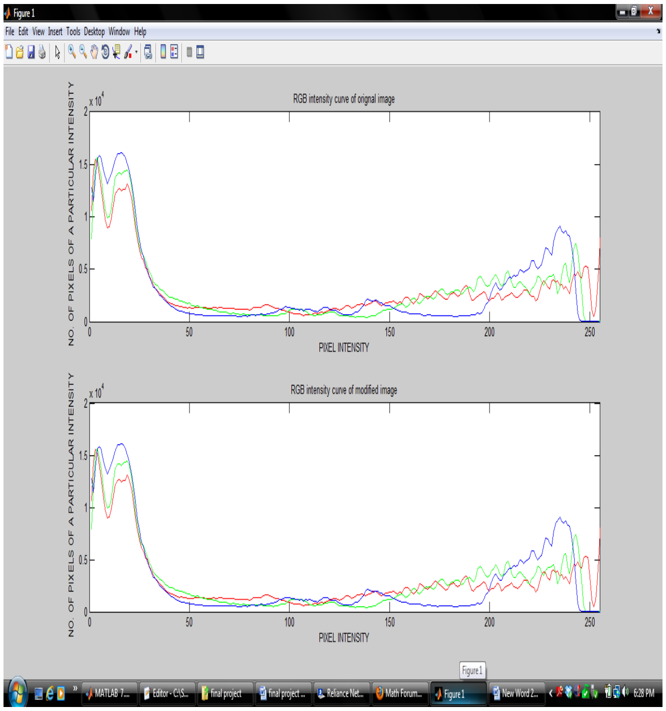

Fig. 6. Intensity plot

On comparing the above histograms it is concluded that modifications in the original image are difficult to detect as the patterns of the two histograms are nearly the same.

By further comparison using modified Euclidean distance method proposed by Jain and Vailaya [9]. This method is used for comparison of coloured histograms. I and Q are the histograms of the original image and the image containing the message respectively. Formula proposed for comparing the histograms is as follows:

$$S = 1.0 - \frac{\sqrt{\sum_{r}(I_R(r) - Q_R(r))^2 + \sum_{g}(I_G(g) - Q_G(g))^2 + \sum_{b}(I_B(b) - Q_B(b))^2}}{6}$$

where S is similarity coefficient.

Using this method the calculated value of similarity coefficient is 1.0. This implies that the two histograms are nearly the same.

VI. Novelty Of Proposed Algorithm

The scheme used for selecting and modifying pixels is the focus of the proposed algorithm. In conventional methods selection of pixels is done in an orderly fashion, usually using a key, whereas in the proposed algorithm selection of pixels to be modified is performed randomly. This makes the algorithm securer than conventional algorithms. Another highlight of this algorithm is that the capacity of storing information per pixel is greatly increased without perceivable changes in the modified image. The above mentioned points highlight the uniqueness of this algorithm and justify its novelty.

VII. Results

By adopting the above stated methodology stego process has been performed. Following are some stego covers and stegged images.

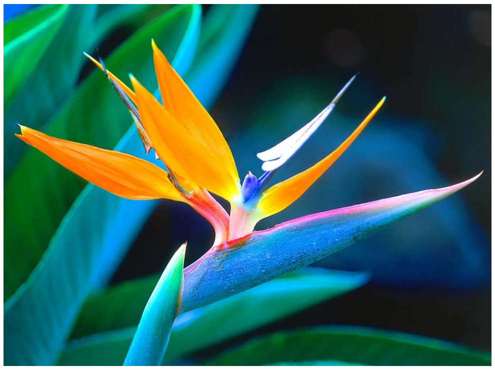

Fig. 7. Original image of a coloured flower

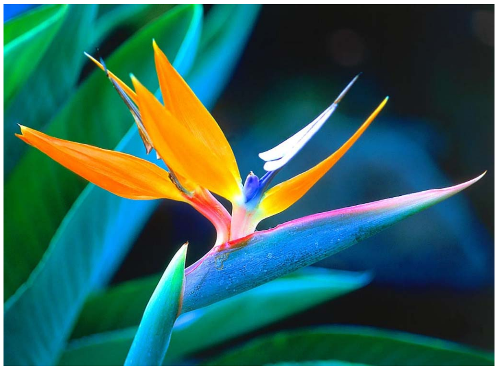

Fig. 8. Modified image containing message 'beautifulflower'

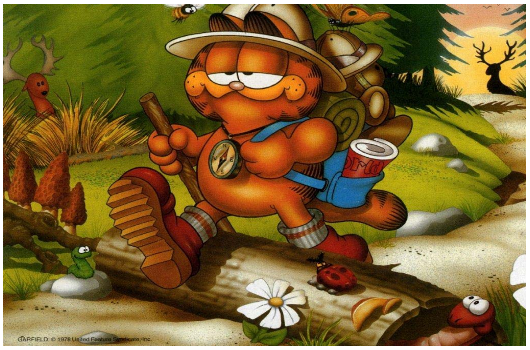

Fig. 9. Original image of a cartoon character

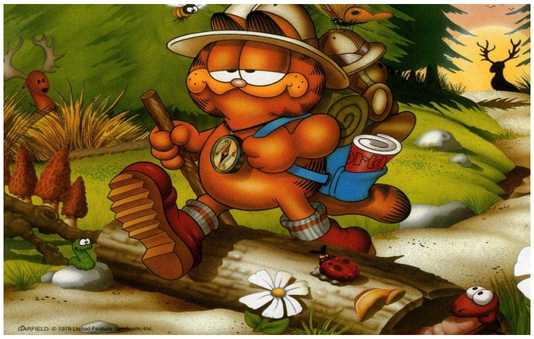

Fig. 10. Modified image containing message 'cutegarfield'

VIII. Conclusion

The block based steganography provides robust and effective technique for information hiding. It was shown by experimental results that the proposed method offers a significant improvement over the conventional techniques. LSB technique used for steganography of 8 bit format is far more vulnerable to attacks as compared to 24 bit format. The advantage of using a bmp file is that it is capable of hiding a large message. The randomness in pixel selection renders detection of hidden information difficult. In this algorithm storage space is significantly increased by increasing the number of modifiable bits per pixel. The main highlight is that the proposed steganography process requires no key. Mainly no attacker can identify the presence of secret data due to the high quality of stego image. Even if the attacker is suspicious the complete retrieval of hidden message is impossible due to randomness of pixels containing the information.

References

[1] M. M Amin, M. Salleh, S . Ibrahim, M.R.K Atmin, and M.Z.I. Shamsuddin, "Information hiding using steganography," IEEE 4th National Conference on Telecommunication Technology Proceedings, Shah Alam, Malaysia, pp. 21-25, January 2003.

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

[3] S .K. Moon and R.S. Kawitkar, "Data security using data hiding," IEEE International Conference on Computational Intelligence and Multimedia Applications, India, pp. 247-251, January 2007.

[4] V. Lokeswara Reddy, Dr. A. Subramanyam and Dr.P. Chenna Reddy, "Implementation of LSB steganography and its evaluation for various file formats," Int. J. Advanced Networking and Applications, vol. 2, pp. 868-872, 2011.

[5] Gandharba Swain and Saroj Kumar Lenka, "A hybrid approach to steganography embedding at darkest and brightest pixels," Proceedings of the International Conference on Communication and Computational Intelligence ,Kongu Engineering College, Perundurai, Erode, T.N., India, pp.529-534, December 2010.

[6] William Stallings, "Cryptography and Network Security, Principles Practice" Edition 3rd. Prentice Hall 2003, ISBN 0-13-091429-0.

[7] M. S. Sutaone and M.V. Khandare, "Image based steganography using LSB insertion technique," IET International Conference on Wireless, Mobile and Multimedia Networks, India, pp. 146-151, January 2008.

[8] Beenish Mehboob and Rashid Aziz Faruqui, "A steganography implementation," IEEE-International symposium on Biometrics & security technologies, ISBAST, Islamabad, April 2008.

[9] A Vadivel, A.K.Majumdar and Shamik Sural, "Performance comparison of distance matrices in context-based image retrieval applications," IIT kharagpur research work.