A Novel Approach to RGB Channel Based Image Steganography Technique

Gandharba Swain[1] and Saroj Kumar Lenka[2]
[1]Research Scholar-CSE, SOA University, Bhubaneswar-751030, Odisha, India.
[2]Professor- CSE, MITS University, Lakshmangarh-332311, Rajasthan, India.

Abstract: *A novel approach to RGB channel based steganography technique is proposed. The RSA algorithm is used for encryption and decryption. In an RGB image, each pixel (24 bits) is having R-channel 8 bits, G-channel 8 bits and B-channel 8 bits. The image is divided into 8 blocks and the cipher text is divided into 8 blocks. One cipher block is allocated to be embedded* **in** *only one image block by a user defined sub key. Out of the three channels in each pixel of the image one is used as the indicator channel. The indicator channel for the different blocks is not the same. The other two channels (called data channels) are used for hiding cipher text bits in 4 least significant bit (LSB) locations. In a data channel 4 bits of cipher text can be embedded, if after embedding the change in pixel value is less than or equal to 7. The two LSBs of indicator will tell whether the cipher text is embedded in only one data channel or in both data channels, so that retrieving can be done accordingly at the receiver. The Technique is implemented and results are analyzed.*

Keywords: *steganography, cryptography, RGB based, indicator channel, data channel, RSA*

Received December 25, 2011; Accepted February 21, 2012

1. Introduction

Information and communication technology has grown rapidly. Internet is the most popular communication medium nowadays. But message transmission over the internet is facing some problems such as data security, copyright control, etc. So we need secure secret communication methods for transmitting data over the internet. Cryptography is a well known method in which the information is encrypted by using a key and then sent on the channel. Steganography is another method in which the communication is not apparent to the intruder. It is an art of covert communication in which the secret information is hidden inside a carrier file such that the change in appearance of the carrier file is not be apparent to normal human eye. Sometimes we use both cryptography and steganography to achieve two levels of security.

Steganography can be categorized into four categories. Those are: Steganography in image, steganography in audio, steganography in video and steganography in text. The image steganography algorithms can be categorized into two categories, namely spatial domain and frequency domain. In each of these categories we can have adaptive and dynamic methods. Adaptive methods are image statistics based, where as dynamic methods are message bit dependent. For hiding information inside images usually Least Significant Bit (LSB) method is used. The LSB method usually does not increase the file size, but depending on the size of the information that is to be hidden inside the file, the file can become noticeably distorted. Instead of hiding a fixed number of bits in the LSBs of each pixel, one can also embed different number of bits in LSBs of different pixels based on pixel value range calculation [6].

Motameni et al. proposed a text in image steganography [11]. This technique presents a way for labeling different colors to identify dark areas of image and then embed the text in those areas. Al-Husainy proposed a text in image steganography by mapping the binary values of characters of the text message to various pixels of the image [1].

In general if the pixels are located in edge areas they can tolerate larger changes than those in smooth areas. The range of changeable pixel value in smooth areas is small, where as in edge areas it is large so that the stego image maintains a good perceptual quality. Wu and Tsai [14] proposed a pixel value differencing method, where a cover image is partitioned into non overlapping blocks of two consecutive pixels. A difference value is calculated from the value of the two pixels in each block. Secret data is embedded into a cover image by replacing the difference values of the two pixel blocks of the cover image with similar ones, in which bits of embedded data are included. Zhang and Wang found that pixel value differencing steganography is vulnerable to histogram based attacks and proposed a modification for enhanced security [15].

Chang and Tseng proposed two sided, three sided and four sided side match schemes [2]. The two sided side match method uses the side information of the upper and left neighboring pixels in order to make estimates. The three sided side match method utilizes upper and left neighboring pixels; and one of the other neighboring pixels. The four sided match method uses the upper, left, right and below neighbors. These two methods proposed by Wu and Tsai; and Chang and Tseng suffer from the fall of boundary problem (FOBP). Kim et al. proposed a scheme to reduce this FOBP problem [9].

Juneja and Sandhu [7] proposed a technique based on LSB array, in which they have taken all the LSB bits of the different pixels as an array called LSB array, mapped the encrypted message block to this LSB array, where maximum matched, there steganographed. In fact a number of images are to be experimented; the image in which the percentage of match is maximum can be used. In RGB images each pixel is represented by 3 bytes to represent the intensities of red, green and blue channels in that pixel. Parvez and Gutub presented a technique based on RGB intensity values of the pixel [12]. They took one of the channels as indicator channel and used one or both of the remaining two channels to conceal data bits. The last two bits of the indicator channel tell whether the data bits are hidden in the other two channels or not. Tiwari and Shandilya proposed two techniques based on RGB [13]. In the first technique they modified the technique of Pervez and Gutub, by changing the indicator for every subsequent pixel. The second technique is a random number generation based, in which a random number generated determines the number of least significant bits that is used to hide the secret data. Kaur et al. [8] proposed a RGB based technique in which embedding was done up to 4 bits in the channels other than the indicator channel based upon the last four bits of the indicator channel. Gutub proposed an improved version of RGB based technique [5]. Ghosal proposed a new pair wise bit based data hiding approach on 24 bit color images [4].

In this proposed technique the image is divided into 8 blocks, the message is divided into 8 blocks; one message block is allocated to one image block through a user defined key. In each image block, one of the channels is made as indicator channel and the remaining two are used for embedding 4 bits of data each, if the change in channel value after embedding is less than or equal to 7. Compared to the other techniques proposed by others this is a unique and stronger one due to the division of the image and message into blocks. Moreover RSA algorithm is used for encryption at the sender and for decryption at the receiver.

The rest of the paper is organized as follows. In section2 the algorithms at sender and receiver are discussed. In section3 the experimental results and in section4 the performance analysis is done. Finally in section5 the paper is concluded.

2. The Algorithms

The algorithms followed at sender and receivers are discussed in subsections 2.1 and 2.2 respectively. The steps in the algorithms are illustrated with appropriate examples too.

2.1. The Algorithm at Sender

1. Divide the image into 8 blocks after leaving some reserved bytes (say, the initial 3099 bytes) at the beginning of the image. The 8 image blocks are B0, B1, B2, B3, B4, B5, B6 and B7.
2. Encrypt the plain text by using RSA algorithm, thus the cipher text is obtained.
3. Divide the cipher text into 8 blocks in the following manner.
   The cipher text comprises of so many bytes. Each byte is 8 bits. Keep together the first bits (most significant bits) of all the bytes, it becomes message block M0. Keep together the second bits of all the bytes, it becomes message block M1. Keep together the third bits of all the bytes, it becomes message block M2. Keep together the fourth bits of all the bytes, it becomes message block M3. Keep together the fifth bits of all the bytes, it becomes message block M4. Keep together the sixth bits of all the bytes, it becomes message block M5. Keep together the seventh bits of all the bytes, it becomes message block M6. Keep together the eighth bits (least significant bits) of all the bytes, it becomes message block M7.
4. Enter the sub key K1, this sub key is for the allocation of message blocks to image blocks for possible embedding. This is a string of digits with a length of 8, formed by the digits 0 through 7, such that each digit occurs only once in the string. For example a sub key K1 = 05432167 means message block M0 is to be allocated to image block B0, message block M5 is to be allocated to image block B1, message block M4 is to be allocated to image block B2, message block M3 is to be allocated to image block B3, message block M2 is to be allocated to image block B4, message block M1 is to be allocated to image block B5, message block M6 is to be allocated to image block B6, and message block M7 is to be allocated to image block B7.
5. The second sub key K2, which tells about the indicator channels of all the 8 different image blocks, is to be calculated from the image blocks. From each block one indicator channel is calculated. One of the red, green and blue channels will be the indicator channel, whose sum over all the pixels in that block is maximum. For example suppose sum1, sum2 and sum3 are the sum of red, green and blue channels of different pixels in a block respectively. If the largest is sum2, then green channel (G) is the indicator channel and red (R) and blue (B) channels

are the data channels. Suppose R is the indicator channel in image block B0, B is the indicator channel in image block B1, B is the indicator channel in image block B2, R is the indicator channel in image block B3, G is the indicator channel in image block B4, B is the indicator channel in image block B5, G is the indicator channel in image block B6 and R is the indicator channel in image block B7, then the sequence is RBBRGBGR. R corresponds to 0, G corresponds to 1 and B corresponds to 2. Thus the sequence is 02201210, which is the sub key K2. Thus our key K= K1 K2 = 05432167 02201210, which is to be hidden in the reserved location of the image (byte number 3001 to 3099) along with message block length using two least significant bits in each byte.

Suppose, I is the indicator channel and the other two channels are channel1 and channel2.

If R is the I channel, then G is channel1 and B is channel2.

If G is the I channel, then R is channel1 and B is channel2.

If B is the I channel, then R is channel1 and G is channel2.

6. Now the different message blocks are allocated to different image blocks and in each image block the indicator is decided. Now the embedding in a block is done as follows.
   a. Take the next pixel of the image block. Take 4 bits of the cipher text.
   b. Compare 4 LSBs of channel1 with 4 bits of cipher text, if the difference is less than or equal to 7 then embed these 4 cipher text bits at those 4 LSBs of channel1. Take next 4 bits of cipher text, go to step(c). Otherwise (if the difference is greater than 7) with the same 4 bits of cipher text go to step(c).
   c. Compare 4 least significant bits of channel2 with 4 bits of cipher text, if the difference is less than or equal to 7 then embed these 4 cipher text bits at those 4 LSBs of channel2. Otherwise do not embed in channel2, and those 4 cipher text bits are to be considered for the next pixel.
   d. Go to step (a).
7. For a pixel if embedding is done as per step (b) only, i.e data is embedded in channel1 only, then the two LSBs of its indicator will be set to 00. If for a pixel embedding is done as per step(c) only i.e., data is embedded in channel2 only, then the two LSBs of its indicator will be set to 01. If for a pixel embedding is done as per both the steps (b) and (c) i.e., 4 bits embedded in channel1 and 4 bits embedded in channel2, then the two LSBs of indicator will be set to 10. If in a pixel data is embedded neither in channel1 nor in channel2, then the last two LSBs of its indicator will be set to 11.
8. Stop.

As an example suppose we have the three pixels as in table1 and the Green channel is the indicator channel. Suppose the cipher text stream is: 0010 1010 0000 1111 0011. In pixel1 channel1 is 10010110 = 150, if the four bits: 0010 are embedded it becomes 10010010 = 146, the difference between 150 and 146 is less than 7, so embedded. Now channel2 of pixel1 is 00011110=30, the next 4 cipher text bits are 1010, if these four bits are embedded it becomes 00011010 = 26, the difference is less than 7, so embedded. The last two bits of indicator of pixel1 are made 10.

In pixel2 channel1 is 00001111=15, the next 4 cipher text bits are 0000, if embedded it becomes 00000000=0, and the difference is more than 7, so not embedded. Then check the channel2 of pixel2. It is 11000100=196, the cipher text bits are 0000, if embedded it becomes 11000000 = 192, and the difference is less than 7, so embedded in channel2 of pixel2. Last two bits of indicator are changed to 01.

In pixel3 channel1 is 10111101 = 189, the next 4 cipher text bits are 1111, if embedded, it becomes 10111111 =191, and the difference is less than 7, so embedded. The channel2 of pixel3 is 11001100 = 204, the next 4 bits of cipher text is 0011, if embedded the channel value becomes 11000011 = 195, the difference is more than 7, so not embedded. Thus the last two bits of indicator become 00. After embedding the pixels are also shown in table1 (lower part).

Table1. Example of embedding.

	Before Embedding		
	Red (Channel1)	Green (**Indicator Channel**)	Blue (Channel2)
Pixel1	1001**0110**	10001111	0001**1110**
Pixel2	00001111	01010100	1100**0100**
Pixel3	1011**1101**	11110000	11001100
	After Embedding		
	Red (Channel1)	Green (**Indicator Channel**)	Blue (Channel2)
Pixel1	1001**0010**	100011**10**	0001**1010**
Pixel2	00001010	010101**01**	1100**0000**
Pixel3	1011**1111**	111100**00**	11001100

2.2.The Algorithm at Receiver

1. Retrieve the sub key K_1, sub key K_2, and the length of the blocks from the reserved location in the image i.e., from byte numbers 3001 to 3099.
2. Suppose K_1= 05432167 and K_2= 02201210. In K_2, 0 means Red, 1 means Green and 2 means Blue. Thus K_2 = 02201210, means Red is the indicator channel in image block B_0, Blue is the indicator channel in image block B_1, Blue is the indicator channel in image block B_2, Red is the indicator channel in image block B_3, Green is the indicator in image

block B_4, Blue is the indicator channel in image block B_5, Green is the indicator channel in image block B_6 and Red is the indicator channel in image block B_7.

3. For each block identify the channel1 and channel2 as the indicator channel is identified. If Red is the indicator channel, then Green is channel1 and Blue is channel2. If Green is the indicator channel, then Red is channel1 and Blue is channel2. If Blue is the indicator channel, then Red is channel1 and Green is channel2.
4. Declare the variables CIPHER.
   Declare the variable CIPHERi for i=0 to 7.
   For i=0 to 7 do the following.
   Take the image Block Bi. Start from the first pixel. Continue till the number of bits extracted is equal to the message block length.
   a. Read the indicator of the pixel.
   b. If the last two bits of indicator are 00, retrieve the 4 LSBs of channel1 and append to the variable CIPHERi.
   c. If the last two bits of indicator are 01, retrieve the 4 LSBs of channel2 and append to the variable CIPHERi.
   d. If the last two bits of indicator are 10, retrieve the 4 LSBs of channel1 and channel2 and append to the variable CIPHERi.
   e. If the last two bits of indicator are 11, neither retrieve from channel1 nor retrieve from channel2 of that pixel.
   f. Take the Next pixel, Go to 4(a).
5. Concatenate all the cipher blocks, CIPHER1 to CIPHER7 to get the CIPHER.
6. Now apply RSA decryption to CIPHER to get the plain text.
7. Stop.

3. The Experimental Results

In figure 1, (a) is the original Leena image and (b) is its stego image with 14869 bytes of cipher text embedded into it, (c) is the Garden-Home image and (d) is its stego image with 20047 bytes of cipher text embedded in it, (e) is the Road image and (f) is its stego image with 15039 bytes of cipher text embedded in it, (g) is the Player image and (h) is its stego image with 30079 bytes of cipher text embedded in it.

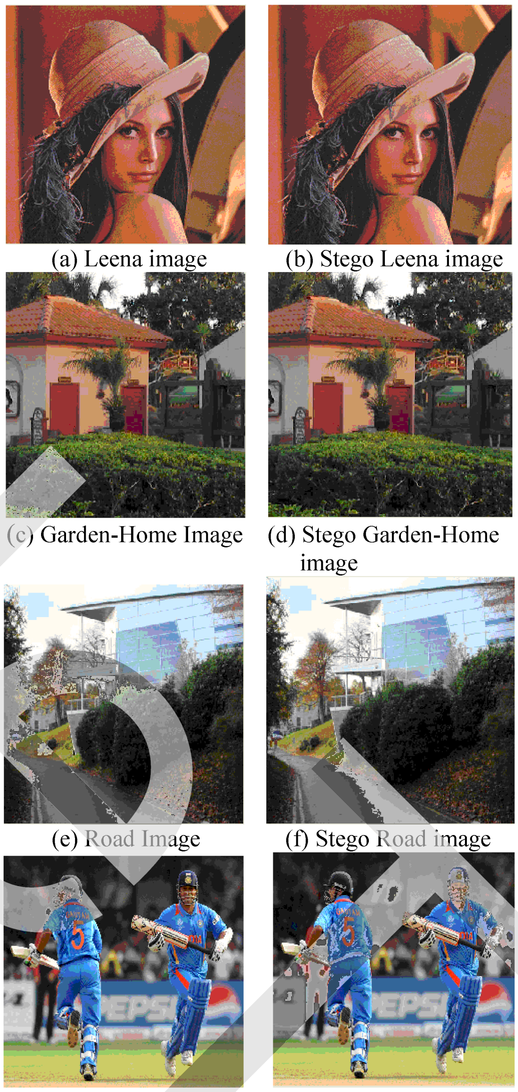
The peak signal to noise ratio (PSNR) at different payloads for different images is as given in table2. PSNR is measured in decibels (dB). PSNR values falling below 30 dB indicate a fairly low quality, i.e., distortion caused by embedding can be obvious; however, a high quality stego image should strive for 40 dB and above [3]. The PSNR is defined below.

$$PSNR = 10 \log_{10}\left(\frac{C_{max}^{2}}{MSE}\right),$$

Where C_{max} = the maximum pixel value, 255 for 8-bit images. MSE called the mean square error is as defined below.

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(S_{xy} - C_{xy}\right),$$

Where x and y are the image coordinates, M and N are the dimensions of the image, C_{xy} is the cover image and S_{xy} is the generated stego image.



(a) Leena image (b) Stego Leena image
(c) Garden-Home Image (d) Stego Garden-Home image
(e) Road Image (f) Stego Road image
(g) Player image (h) Stego Player image

Figure1. (a), (c), (e), (g) are Original images and (b), (d), (f), (h) are their stego images respectively

Table2 Observed PSNR for different images

Image	Image size (in kilo bytes)	Amount of cipher embedded (in bytes)	PSNR (in decibels)
Leena	768	14869	49.67
Garden-Home	1214	20047	50.26
Road	1050	15039	50.79
Player	1837	30079	50.31

4. Performance Analysis

The performance of various steganographic methods can be rated by the three parameters: security, capacity, and imperceptibility [10]. Steganography may suffer from many active or passive attacks, thus a secured steganography technique will survive from passive or active attacks. Moreover if the existence of the secret message can only be estimated with a probability not higher than random guessing in the presence of some steganalytic systems, steganography may be considered secure under such steganalytic systems. Capacity means the amount of information that can be hidden, it should be as high as possible. It can be measured as the size of secret message, or in relative value called bits per pixel, or the ratio of the secret message to the cover image. Stego images should not have severe visual artifacts. Under the same level of security and capacity, the higher the imperceptibility of the stego image, the better is the technique. If the resultant stego image appears innocuous enough, one can believe this requirement to be satisfied well.

Our proposed algorithm is highly secure as it uses allocation of message blocks to image blocks through a sub key and the indicator channel in different blocks is different. The existing steganalytic attacks can not detect the existence of steganography. Moreover we are hiding the encrypted text, not the plain text, which adds another level of security. The encryption algorithm is the RSA algorithm. From fig.1 one can observe that there is no visual artifacts with the stego images, they are looking exactly same as their corresponding original images.

There are some basic notes that should be observed by a steganographer [3]. Those are: (i) In order to eliminate the attack of comparing the original image with the stego image, we can freshly create an image and destroy it after generating the stego image. The images available in the World Wide Web should not be used. (ii) In order to avoid any human visual perceptual attack, the generated stego image must not have visual artifacts. (iii) Smooth homogeneous areas must be avoided, however chaotic areas with naturally noisy back grounds and salient rigid edges can be targeted. (iv) The secret data must be a composite of balanced bit values, since in general; the expected probabilities of bit 0 and 1 for a typical cover image are the same. In such cases encryption provides a balance.

5. Conclusion

In this paper a new RGB channel based steganography is proposed. It provides two levels of security, one at cryptography level and the other at steganography level. The embedding capacity is good. The message blocks are allocated to the image blocks by the sender, using a key; which provides another level of security. The indicator channel is computed freshly for each image block. The embedding into channel1 and/or channel2 is done by a difference calculation of 4 data bits and 4 LSBs of the channel1 and/or channel2. PSNR is better as compared to some of the existing algorithms. No visual artifacts can be observed from the corresponding stego images.

References

[1] Al-Husainy M.A.F., "Image steganography by mapping pixels to letters", *Journal of Computer Science*, Vol. 5, No.1, pp. 33-38, 2009.

[2] Chang C. C., and Tseng H. W., "A steganographic method for digital images using side match", *Pattern Recognition Letters,* Vol. 25, No.12, pp.1431-1437, 2004.

[3] Cheddad A., et al. "Digital image steganography survey and analysis of current methods", *Signal Processing*, Vol. 90, pp.727-752, 2010.

[4] Ghosal S. K., "A new pair wise bit based data hiding approach on 24 bit color image using steganographic technique", *Proceedings of IEMCON-2011*, pp.123-129, 2011.

[5] Gutub A. A., "Pixel indicator technique for RGB image steganography", *Journal of Emerging Technologies in Web Intelligence,* Vol. 2, No.1, pp.56-63, 2010.

[6] Jain Y. K., and Ahirwal R. R., "A novel image steganography method with adaptive number of least significant bits modification based on private stego keys", *International Journal of Computer Science and Security*, Vol. 4, No.1, pp.40-49, 2010.

[7] Juneja M., and Sandhu P. S., "Designing of robust steganography technique based on LSB insertion and encryption", *Proceedings of International Conference on Advances in Recent Technologies in Communication and Computing*, pp.302-305, 2009.

[8] Kaur M., et al. "A dynamic RGB intensity based steganography scheme", *World Academy of Science, Engineering and Technology*, Vol. 67, pp. 833-838, 2010.

[9] Kim K. J., Jung K. H., and Yoo K. Y., "Image steganographic method with variable embedding length", *Proceedings of International Symposium on Ubiquitous Computing*, pp. 210-213, 2008.

[10] Li B., et al. "A survey on image steganography and steganalysis", *Journal of Information Hiding and Multimedia Signal Processing*, Vol.2, No.2, pp.142-172, 2011.

[11] Motameni H., et al. "Labeling method in steganography", *World Academy of Science, Engineering and Technology*, Vol. 24, pp.349-354, 2007.

[12] Parvez M. T., and Gutub A. A., "RGB based variable-bits image steganography", *Proceedings of IEEE Asia Pacific Services Computing Conference*, pp.1322-1327, 2008.

[13] Tiwari N., and Shandilya M., "Secure RGB image steganography from pixel indicator to triple algorithm- an incremental growth",

International Journal of Security and its Applications, Vol. 4, No.4, pp.53-62, 2010.

[14] Wu D. C., and Tsai W. H., "A steganograhic method for images by pixel value differencing", *Pattern Recognition Letters*, Vol.24, No.9-10, pp.1613-1626, 2003.

[15] Zhang X., and Wang S., "Vulnerability of pixel value differencing steganography to histogram analysis and modification for enhanced security", *Pattern Recognition Letters*, Vol. 25, pp. 331-339, 2004.

Gandharba Swain is working as an Associate Professor in the Department of Information Technology, GMR Institute of Technology, Rajam, Andhra Pradesh, India. He received B.Sc(Hons) degree from Berhampur University in 1995, MCA degree from VSS University of Technology (formerly UCE), Burla, in 1999, M.Tech (CSE) degree from NIT, Rourkela, in 2004. He has more than 12 years of teaching experience. He has authored one text book, published several research articles in national and international journals and conferences.

Saroj Kumar Lenka is working as Professor in the Department of Computer Science and Engineering, Mody Institute of Technology and Science-Deemed University, Rajasthan, India. He has received B.Sc. degree from Berhampur University in 1990, B.E in Computer Science and Engineering from Utkal University in 1994, M.Sc. in Information Technology in 2003 from Allahabad Agricultural Institute Deemed University, M.Tech. in Computer Science in 2005 from Kalinga University and Ph.D. in Computer Science in 2008 from Berhampur University, Orissa. He has more than 17 years experience in teaching and published many research articles in national and International journals and conferences.