IJCSC International Journal of Computer Science and Communication Vol. 3, No. 1, January-June 2012, pp. 49-53

APPLICATION OF A LARGE KEY CIPHER IN IMAGE STEGANOGRAPHY BY EXPLORING THE DARKEST AND BRIGHTEST PIXELS

Gandharba Swain[1] and Saroj Kumar Lenka[2]

[1]Research Scholar-CSE, SOA University, Bhubaneswar-751030, Odisha, India, *E-mail: gswain1234@gmail.com.*

[2]Professor, Department of CSE, MITS University, Lakshmangarh-332311, Rajasthan, India, *E-mail: lenka.sarojkumar@gmail.com.*

ABSTRACT

In this paper a technique for secure communication in internet is proposed. It includes a new cryptographic algorithm and a new steganographic approach. The cryptographic algorithm is a new block cipher with a large key. The block size is 128 bits and the key size is 512 bits. The secret message is encrypted by this block cipher, and then the resultant cipher text is embedded in four least significant bits (LSBs) of darkest and brightest pixels. Each pixel of the image is one byte. The brightest pixels are those pixels whose gray values falls in the range 224 to 255 in 8 bit gray scale and darkest pixels are those pixels whose gray values falls in the range 0 to 31 in 8 bit gray scale. This steganographic approach supports high capacity and the stego images are highly imperceptible. The results has been compared with existing steganographic schemes to evaluate its overall performance.

Keywords: Steganography, cryptography, darkest pixel, brightest pixel, large key cipher.

1. INTRODUCTION

Information and communication technology has grown rapidly. Internet is the most popular communication medium nowadays. But message transmissions over the internet is facing some problems such as data security, copyright control, etc. So we need secure secret communication methods such as steganography. 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 two categories we can have adaptive and dynamic methods. Adaptive methods are image statistics based, where as dynamic methods are message bit dependent. When hiding information inside images usually least significant bit (LSB) method is used. In the LSB method the 8^{th} bit of every byte of the carrier file is substituted by one bit of the secret information. 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 [1]. In image steganography, a pixel can carry secret bits by adding/subtracting one to/from the gray value. This kind of ±1 steganography can hide a longer message than simple LSB embedding. Zhang et al. proposed a double layered embedding method to further improve the embedding efficiency of ±1 steganography [2]. In [3] a text in image steganography is proposed. This technique presents a way for labeling different colors to identify dark areas of image and then embed the text in those areas. In [4] a text in image steganography is proposed by mapping the binary values of characters of the text message to various pixels of the image. Wu and Tsai proposed a pixel value differencing method, where a cover image is partitioned into non overlapping blocks of two consecutive pixels [5]. 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. Chang and Tseng [6] employed two sided, three sided and four sided side match schemes in which correlation of a pixel with its neighboring pixels is taken into account to make embedding decisions. Parvez and Gutub presented a technique based on RGB intensity values of the pixel [7]. They took one of the channels as indicator channel and used one or both of the remaining two channels to conceal data bits. Juneja and Sandhu [8] proposed a technique based upon 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 this paper a new cryptography algorithm (called large key cipher) for encryption and decryption and a new embedding technique i.e., the 4 LSBs of the darkest

and brightest pixels are proposed. In section 2, the working of the large key cipher is discussed, in section 3 the embedding approach, in section 4 the proposed algorithm; in section 5 the experimental results are discussed and in section 6 the paper is concluded.

2. THE LARGE KEY CIPHER WITH 512 BIT KEY

The Large key cipher is a block cipher with 128 bit (16 characters) block length and 512 bit (64 characters) key length. The key is splitted into 8 sub keys of equal length. The sub key generation, encryption and decryption processes are as discussed below.

2.1. Sub Key Generation

The large key cipher receives 64 characters (= 512 bits) as a key. This key is divided into 4 sub keys each 16 characters. Then K_1 becomes a sub key with first 16 characters, a matrix with 4 rows and 4 columns. K_2 becomes a sub key with second 16 characters, a matrix with 4 rows and 4 columns. K_3 becomes a sub key with next 16 characters, a matrix with 4 rows and 4 columns. K_4 becomes a sub key with last 16 characters, a matrix with 4 rows and 4 columns. Find the determinants of these matrices and sort them in ascending order of their determinants. Now they are KS_0, KS_2, KS_4, and KS_6. Now compute the transpose of K_1, K_2, K_3, K_4 and sort them in ascending order of their determinants and rename the sorted list as KS_1, KS_3, KS_5, and KS_7. Thus we have now 8 sub keys KS_0, KS_1, KS_2, KS_3, KS_4, KS_5, KS_6 and KS_7.

The plain text which includes characters, special characters and digits is divided into different blocks, each 16 characters. If the last block is less than 16 characters, then at the end blank spaces can be appended to make it 16 characters long. Suppose P is a block of plain text (16 characters), a matrix with 4 rows and 4 columns, then the encryption and decryption processes are as discussed below.

2.2. The Encryption Process

For i = 0 to 7

{

P_1= CircularShiftLeft (P)

P_2= XOR (P_1, KS_i)

P_3= Stir (P_2)

P=P_3

}

C=P

Where P is the plain text. KS_i, for i = 0 to 7 are the sub keys. P_1, P_2, P_3 are the intermediate results and C is the cipher text. The CircularShiftLeft, XOR and Stir operations are as discussed below.

2.3. The Decryption Process

For i=7 to 0

{

C_1=Stir(C)

C_2=XOR(C_1, KS_i)

C_3=CirShiftRight(C_2)

C=C_3

}

P=C

Where C_1, C_2, C_3 are the intermediate results. CircularShiftRight operation is as discussed below.

2.4. Calculation of the CircularShiftLeft and CircularShiftRight

The CircularShiftLeft operation is defined as follows. The first row is kept unchanged. Second row elements shifts one position left in a circular manner. Third row elements moves two positions left and fourth row elements moves three positions left in a circular manner.

The CircularShiftRight operation is defined as follows. The first row is kept unchanged. Second row elements shifts one position right in a circular manner. Third row elements shifts two positions right and fourth row elements shifts three positions right in a circular manner. The CircularShiftLeft is inverse of CircularShiftRight operation.

2.5. The Stir Operation

B = Stir (A), where A and B are matrices with 4 rows and 4 columns, each element is 8 bits. This stir operation is defined as follows. The first two bits (i.e., 1st and 2nd bits) of each element of A in a row are combined to form the first element of B in that row. Next two bits (i.e., 3rd and 4th bits) of each element of A in a row are combined to form 2nd element of B in that row. Next two bits (i.e., 5th and 6th bits) of each element of A in a row are combined to form next element of B in that row. The last two bits (i.e., 7th and 8th bits) of each element of A in a row are combined to form the last element of B in that row. This stir operation is reversible, i.e., stir (Stir (A)) = A.

2.6. The XOR Operation

If A and B are two matrices of same order, then XOR (A, B) is the bit by bit exclusive-or operation of the respective elements of the two matrices. For example if A = 10101011 and B = 10011001, then C = XOR(A, B) = 00110010. The XOR operation is reversible, i.e., If C = XOR (A, B) then A = XOR(C, B) and B = XOR(C, A).

3. THE EMBEDDING TECHNIQUE

The embedding is done at 4 least significant bit (i.e., 5th, 6th, 7th and 8th) locations of the darkest and brightest pixels of the image. The darkest pixels are those whose gray

values are in the range 0 to 31 in an 8 bit gray scale. Similarly brightest pixels are those whose gray values are in the range 224 to 255. Four bits of the cipher text will be taken and be embedded at 5^{th}, 6^{th}, 7^{th} and 8^{th} bit locations of a darkest or a brightest pixel. Then next 4 bits of cipher text will be taken and will be embedded at 5^{th}, 6^{th}, 7^{th} and 8^{th} bit locations of the next darkest or brightest pixel. This is continued till the end of cipher text.

Pixels having gray values in the range 0 to 15, after changing the 5^{th}, 6^{th}, 7^{th} and 8^{th} bits, their values falls in the range 0 to 15. Similarly pixels having gray values in range 16 to 31, after changing the 5^{th}, 6^{th}, 7^{th} and 8^{th} bits, their values will fall in the range 16 to 31 also. In general pixels having gray values in range 0 to 31, if their 5^{th}, 6^{th}, 7^{th} and 8^{th} bits are changed the changed value also falls in the range 0 to 31. Pixels having gray values in range 240 to 255, after changing the 5^{th}, 6^{th}, 7^{th} and 8^{th} bits, their values falls in the range 240 to 255 also. Similarly the pixels having gray values in range 224 to 239, after changing the 5^{th}, 6^{th}, 7^{th} and 8^{th} bit locations the changed values will fall in range 224 to 239 too. In general the pixels with values in range 224 to 255, after change will fall in same range. It is clear from the above discussions that a pixel value after changing the 5^{th}, 6^{th}, 7^{th} and 8^{th} bit locations will make a maximum displacement of 15. In 8 bit gray scale we have 256 colors; our eyes can hardly identify 10 to 15 colors. Thus the change in quality of the entire image can not be noticeable to our eyes.

4. THE PROPOSED ALGORITHM

4.1. Algorithm at Sender

Step 1: Convert the carrier image to binary. *Step2:* Divide the secret message into blocks, each comprising of 16 characters. If last block is less than 16 characters, append blank spaces to make it a length of 16. *Step 3:* Accept the 64 character (512 bit) key and formulate the 8 sub keys as discussed in key generation section. *Step 4:* Apply encryption process of larger key cipher as discussed above to convert each plain text block into cipher text. Step5: Keep all the cipher text blocks together to form the complete cipher text. *Step 6:* Transform these cipher text to binary. *Step 7:* Embed the cipher text into binary image at 4 LSBs of darkest and brightest pixels, and then we get the stego binary image. Now convert this stego binary image to stego image and send to receiver.

4.2. Algorithm at Receiver

Step 1: Transform the received image to binary. *Step 2:* Retrieve the embedded cipher text bits from the darkest and brightest pixels of the binary stego image. *Step 3:* Keep them together, convert to text and divide into blocks, each 16 characters. *Step 4:* Accept the 64 character (512 bit) key and formulate the 8 sub keys as discussed in key generation section. *Step 5:* Apply decryption process of the large key cipher to each cipher text blockto get the plain text block. *Step 6:* Keep together all the plain text blocks, thus we get the secret message.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

The technique is implemented using java programming language. The results are observed for more than 50 images. Two sample observations are as discussed below.

In figure 1, (a) is the original Temple image and (b) is it's histogram, (c) is the stego Temple image with 10 kilo

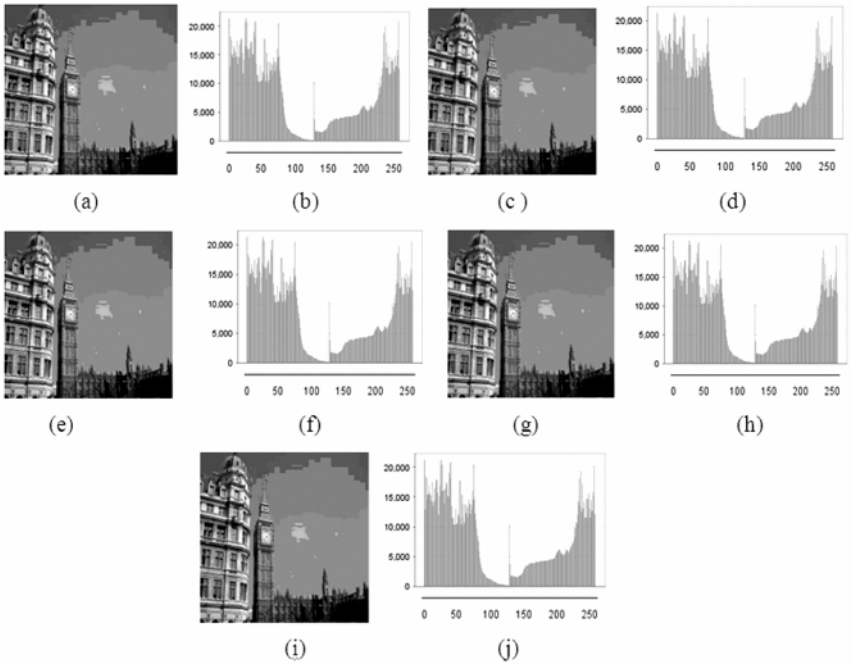

Figure 1: (a) Original Temple Image, (c), (e), (g), (i) are Stego Temple Images and (b), (d), (f), (h), (j) are their Histograms Respectively.

bytes (KBs) of cipher text embedded and (d) is it's histogram, (e) is the stego Temple image with 20 KBs of cipher text embedded, and (f) is it's histogram, (g) is the stego Temple image with 30 KBs of cipher text embedded and (h) is it's histogram, (i) is the stego Temple image with 40 KBs of cipher text embedded and (j) is it's histogram.

In figure 2, (a) is the original Lady-Man image and (b) is it's histogram, (c) is the stego Lady-Man image with 10 KBs of cipher text embedded and (d) is it's histogram, (e) is the stego Lady-Man image with 20 KBs of cipher text embedded, and (f) is it's histogram, (g) is the stego Lady-Man image with 30 KBs of cipher text embedded and (h) is it's histogram, (i) is the stego Lady-Man image with 40 KBs of cipher text embedded, and (j) is it's histogram.

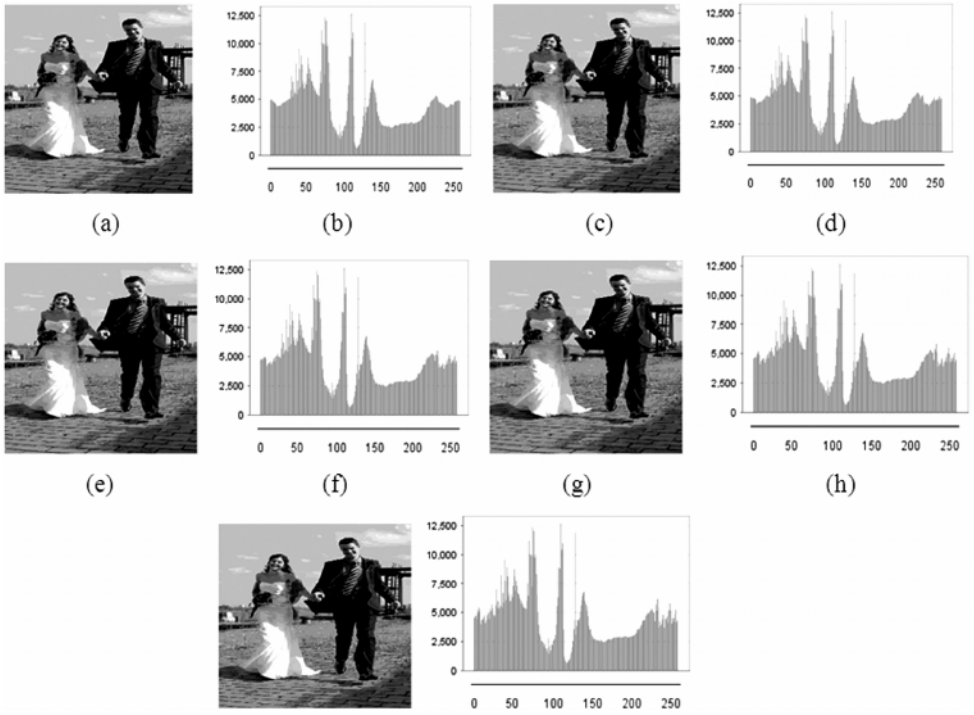

Figure 2: (a) Original Lady-Man Image, (c), (e), (g), (i) are Stego Lady-Man Images and (b), (d), (f), (h), (j) are their Histograms Respectively.

The peak signaz to noise ratio (PSNR) at different payloads for Temple and Lady-Man image is as given in Table 1. 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.

Maximum Embedding Capacity (in bits) = (Total no of pixels – No of Reserved pixels)$_*$ (No of bits per pixel). The image size and maximum embedding capacity of the images is as recorded in Table 2.

Table1
Recorded PSNR for the Two Images

Message Length (in KBs)	*PSNR in Decibels*	
	Temple image	*Lady-Man image*
10	52.12	49.64
20	49.16	46.64
30	47.30	44.89
40	46.17	43.65

Table 2
Embedding Capacity

Image	*Size (in KBs)*	*Capacity (in Bits)*	*Capacity (in KBs)*
Temple	2036	3790516	462.71
Lady-Man	1149	1222348	149.21

The performance of various steganographic methods can be rated by the three parameters: security, capacity, and imperceptibility. This proposed algorithm is highly secure as it uses an encryption algorithm with a large key and embedding at darkest and brightest pixels, not in all pixels. If you see the histograms in figure 1, and figure 2 there is no difference between the histogram of original image and the histograms of its stego images. The encryption algorithm is a new block cipher which uses a 512 bit key, so a stronger algorithm compared to algorithms like IDEA (International Data Encryption Algorithm) and DES (Data Encryption Standard). The capacity i.e., the amount of message that can be embedded is also quite good as shown in Table 2. It can

be observed from the stego images in figure 1 and figure 2 that there is no visual artifacts, showing the presence of steganography.

6. CONCLUSION

In this paper a new steganography technique with a large key cipher is proposed. It provides two levels of security, one at cryptography level and the other at steganography level. If at all the intruder suspects it is quite difficult for him to steal the data because of the two levels of security. After the cipher text is embedded, the degradation in image quality is not apparent to normal human eye. Capacity and PSNR values are also good. No visual artifacts can be observed from the stego images.

REFERENCES

[1] Y. K. Jain, R. R. Ahirwal, "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*, **4**, No.1, pp. 40-49, 2010.

[2] W. Zhang, X. Zhang and S. Wang, "A Double Layered Plus-Minus One Data Embedding Scheme", *IEEE Signal Processing Letters*, **14**, No. 11, 2007, pp. 848-851.

[3] H. Motameni, M. Norouzi, M. Jahandar and A. Hatami, "Labeling Method in Steganography", *Proceedings of World Academy of Science, Engineering and Technology*, **24**, pp. 349-354, October 2007.

[4] M. A. F Al-Husainy, "Image Steganography by Mapping Pixels to Letters", *Journal of Computer Science*, **5**, No. 1, pp. 33-38, 2009.

[5] D. C. Wu and W. H. Tsai, "A Steganograhic Method for Images by Pixel Value Differencing", *Pattern Recognition Letters*, **24**, No. 9-10, 2003, pp. 1613-1626.

[6] C. C. Chang and H. W. Tseng, "A Steganographic Method for Digital Images Using Side Match", *Pattern Recognition Letters*, **25**, No. 12, 2004, pp. 1431-1437.

[7] M. T. Parvez and A. A. Gutub, "RGB Based Variable-Bits Image Steganography", *Proceedings of IEEE Asia Pacific Services Computing Conference*, 2008, pp. 1322-1327.

[8] M. Juneja and P. S. Sandhu, "Designing of Robust Steganography Technique Based on LSB Insertion and Encryption", *Proceedings of International Conference on Advances in Recent Technologies in Communication and Computing*, 2009, pp. 302-305.