Novel Steganography System using Lucas Sequence

Fahd Alharbi
Faculty of Engineering
King Abdulaziz University
Rabigh, KSA

Abstract*—Steganography is the process of embedding data into a media form such as image, voice, and video. The major methods used for data hiding are the frequency domain and the spatial domain. In the frequency domain, the secret data bits are inserted into the coefficients of the image pixel's frequency representation such as Discrete Cosine Transform (DCT) , Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT) . On the other hand, in the spatial domain method, the secret data bits are inserted directly into the images' pixels value decomposition. The Lest Significant Bit (LSB) is consider as the most widely spatial domain method used for data hiding. LSB embeds the secret message's bits into the least significant bit plane (Binary decomposition) of the image in a sequentially manner . The LSB is simple, but it poses some critical issues. The secret message is easily detected and attacked duo to the sequential embedding process. Moreover, embedding using a higher bit plane would degrade the image quality. In this paper, we are proposing a novel data hiding method based on Lucas number system. We use Lucas number system to decompose the images' pixels values to allow using higher bit plane for embedding without degrading the image's quality. The experimental results show that the proposed method achieves better Peak Signal to Noise Ratio (PSNR) than the LSB method for both gray scale and color images. Moreover, the security of the hidden data is enhanced by using Pseudo Random Number Generators (PRNG) for selecting the secret data bits to be embedded and the image's pixels used for embedding.

Keywords—Steganography; LSB; Lucas; PSNR; PRNG

I. INTRODUCTION

The goal of the data hiding system is to communicate a secret message in a way that would not be noticeable by an intruder [1-3]. There are two techniques are regularly used for data hiding, the frequency domain [4-5] and the spatial domain [6-10]. In the frequency domain data hiding techniques, the secret data bits are inserted into the coefficients of the image pixel's frequency representation.

Among the frequency image pixel's frequency representation are Discrete Cosine Transform (DCT) [11], Discrete Fourier Transform (DFT) [12] and Discrete Wavelet Transform (DWT) [13]. On the other hand, in the spatial domain techniques, the secret data bits are inserted directly into the images' pixels value decomposition. In this section, we are discussing the Lest Significant Bit (LSB) which is consider as the most widely spatial domain method used for data hiding.

The least significant bit (LSB) data hiding technique [6-10] is the process of inserting the secret message's data bits into the least significant bits of the image (Binary Form) in a sequential manner [14-17].

For illustration, let I be the original gray scale image where each pixel is represented using 8-bit format, thus each pixel's value is varies from 0 up to 255 as illustrated at Table 1.

TABLE I. ORIGINAL IMAGE PIXELS (BINARY)

Pixels before embedding	
1st	01001100
2nd	01001101
3rd	01001110
4th	01001111
5th	01010000
6th	01001011
7th	01010001
8th	01010001

The secret message is simply the letter Z with its binary representation as 01011010. The LSB embedding process is shown at Table 2, where 8 pixels are used to hide the letter Z. The secret message's bits are inserted into the least significant bit of the image's pixels in a sequential manner.

The LSB data hiding technique is simple and the effect on the image quality is limited and hardly noticed by the human eye duo to the small value of the bit (least significant bit) used for embedding. On the other hand, the LSB is easy to be detected and attacked by simply extracting or changing the least significant bits of each pixel.

TABLE II. LSB EMBEDDING

Secret bits (Z)	Pixels after embedding
0	0100110**0**
1	0100110**1**
0	0100111**0**
1	0100111**1**
1	0101000**1**
0	0100101**0**
1	0101000**1**
0	0101000**0**

On the other hand, using higher bits for embedding the secret message would enhance the security and at the same time degrade the image quality. For example, as illustrated at Table 3, using the fifth bit for hiding the secret message would degrade the image quality duo to the fact that the value of the fifth bit is 16 and the impact would be clear.

TABLE III. LSB EMBEDDING (USING FIFTH BIT)

Secret bits (Z)	Pixels after embedding	difference
0	010**0**1100	0
1	010**1**1101	16
0	010**0**1110	0
1	010**1**1111	16
1	010**1**0000	0
0	010**0**1011	0
1	010**1**0001	0
0	010**0**0001	-16

In this paper, we propose using Lucas number system for image pixel's value decomposition to allow using higher bit plane without degrading the image quality. Also, we are enhancing the security of the data hiding system by using Pseudo Random Number Generator to select the next pixel used for embedding. The rest of the paper is organized as follows: Section II discusses the Lucas based hiding system; Section III describes enhancing the data hiding system's security by using Pseudo Random Number Generators to select the next pixel for embedding; Section IV presents experimental results; we finally conclude in Section V.

II. LUCAS BASED HIDING SYSTEM

Now, we are proposing using the Lucas numbers [18-19] for pixels values decomposition. The Lucas sequence generated using the following formula

$$L_n = L_{n-1} + L_{n-2} \quad , n > 2 \tag{1}$$

Where, $L_1 = 2$ and $L_2 = 1$

The image's pixels values would be represented as the sum of the non-consecutive Lucas numbers [20-21]. To represent the range of 0 to 255, we need 12-bit of Lucas digits.

L_{12}	L_{11}	L_{10}	L_9	L_8	L_7	L_6	L_5	L_4	L_3	L_2	L_1
199	123	76	47	29	18	11	7	4	3	1	2

Now, we represent the pixel value of 26 using Binary and Lucas representation as follows

The binary representation is 00011010
The Lucas representation is 000001010010

Let consider hiding the bit value of 0 using the fifth bit in each decomposition of the pixel value of 26. The result is as following

The binary representation is 000**0**1010
The Lucas representation is 0000010**0**0010

The pixel value after LSB embedding using the fifth bit is 10, while the pixel value after embedding using Lucas decomposition is 19. It is clear that using the Lucas decomposition would result in higher quality data embedding duo to the fact that the Lucas bits are less significant than those of the Binary decomposition.

Here, we reconsider the example of hiding the letter Z in eight pixels (Table 1). The eight pixels values in Lucas system are shown at Table 4.

TABLE IV. ORIGINAL IMAGE PIXELS (LUCAS)

Pixels before embedding	
1st	001000000000
2nd	001000000010
3rd	001000000001
4th	001000000100
5th	001000001000
6th	000101010100
7th	001000001010
8th	001000001010

Table 5 shows the result of embedding the letter Z into eight pixels using Lucas decomposition, where the image quality after embedding is better than that achieved using LSB technique duo to the fact that the digits in Lucas system are less significant (7) than those in binary system (16).

TABLE V. LUCAS EMBEDDING (USING FIFTH BIT)

Secret bits (Z)	Pixels after embedding	difference
0	0010000**0**0000	0
1	0010000**1**0010	7
0	0010000**0**0001	0
1	0010000**1**0100	7
1	0010000**1**0000	3
0	0001010**0**0100	-7
1	0010000**1**0010	3
0	0010000**0**1010	0

III. PSEUDO RANDOM NUMBER GENERATOR

The LSB technique is simple but not secure. The intruder can easily recover the hidden message by extracting the least significant bits. On the other hand, using higher bits for embedding would degrade the image quality. In this section, we enhance the data hiding system's security by using Pseudo Random Number Generators to select the next pixel for embedding. The Pseudo Random Sequence is generated using Non-Linear forward feedback shift Register (NLFFSR) [22-23]. To start generating the Pseudo Random Sequence, the registers simply loaded with any initial value except zero and with each clock (step) a new Random Number is generated. The feedback function of the Pseudo Random Sequence Generator is designed based on the characteristic polynomial of the Generator. The characteristic polynomial is in the form of

$$P(x) = x^n + a_{n-1}x^{n-1} ..... + a_1 x + a_0 \tag{2}$$

Where, n represents the number of registers and the length of the generated sequence is

$$N = 2^n - 1 \tag{3}$$

Let consider a gray scale image $I_{R,C}$, where C is the number of coulombs and R is the number of pixels (rows) in each coulomb.

To increase the security of the data hiding system, we use three Random Sequence Generators to select the next pixel and bit plane used for embedding $I^{b}_{i,j}$. The first Generator selects a coulomb j in the image, the second Generator selects a pixel (row) i at the selected coulomb and the third Generator selects the bit plane ($b = 1, 2.....8$) used for embedding. On the other hand, the color image pixel is represented by three colors Red, Green, and Blue.

Each color is represented by 8-bit, thus each color's value varies from 0 up to 255. In this case, we use four Random Sequence Generators to select the next pixel, color and bit plane used for embedding $I^{b}_{i,j,k}$, where k ($k = 1, 2, 3$) is the selected color. Moreover, we may use two Random Sequence Generators to select the byte y of the secret data and the bit x to be embedded, S^{x}_{y}. The secure data hiding system may uses up to six Random Sequence Generators to perform the data embedding process as the following

$$I^{b}_{i,j,k} = S^{x}_{y} \quad (4)$$

Now, we enhance the embedding of the letter Z into the eight pixels by using Random Sequence Generators. Let assume that the coulomb j has been selected and it is contain the pixels values in the sequence shown at Table 1 and the sequence of the rows i (pixel) are in the following random sequence (6,8,3,5,2,4,7,1). The embedding process is shown at Table 6, where both the image quality and the data security are enhanced.

TABLE VI. LUCAS EMBEDDING + PRNG (USING FIFTH BIT)

Secret bits (Z)	Pixels after embedding	difference
0	0001010**0**0100	-7
1	0010000**1**0010	3
0	0010000**0**0001	0
1	0010000**1**0000	3
1	0010000**1**0010	7
0	0010000**0**0100	0
1	0010000**1**0010	3
0	0010000**0**0000	0

IV. EXPERIMENTAL RESULTS

In this section, we are evaluating the performance of the LSB data hiding technique and the proposed lucas based data hiding technique. The evaluation is performed by using a gray scale 512 × 512 pixels image and a color 512 × 512 × 3 image for data hiding. The quality of the embedding techniques are evaluated by the Peak Signal to Noise Ratio (PSNR), where it is defined as

$$PSNR = 10\log_{10}\left(\frac{I^{2}_{MAX}}{MSE}\right) \quad (5)$$

Where, I^{2}_{MAX} is equal to 255 as the maximum possible value of a pixel in the gray scale images or represents the maximum possible value of a color in the color images. The Mean Square Error (MSE) for the gray scale images is computed as follows

$$MSE = \frac{1}{R \times C}\sum_{i=1}^{R}\sum_{j=1}^{C}\left(\left|I_{i,j} - I'_{i,j}\right|\right)^{2} \quad (6)$$

Where, $I_{i,j}$ is the original image's pixel value and $I'_{i,j}$ is the image's pixel value after embedding.

Also, The Mean Square Error (MSE) for the color images is computed as follows

$$MSE = \frac{1}{R \times C \times L}\sum_{i=1}^{R}\sum_{j=1}^{C}\sum_{k=1}^{L}\left(\left|I_{i,j,k} - I'_{i,j,k}\right|\right)^{2} \quad (7)$$

Where, L is the number of colors in the color image pixel.

A. Performance Evaluation using Gray Scale Image

In this experiment, the performance of the data hiding techniques is evaluated using different bit planes for embedding the secret message's bits. The hiding capacity of the 512 × 512 pixels original image is 32768 data bytes, where each pixel is used for hiding a single data bit. We vary the bit plane used for embedding from the first bit up to the eighth bit for each data hiding technique. In each case we hide 32768 data bytes and evaluate the performance by computing the PSNR.

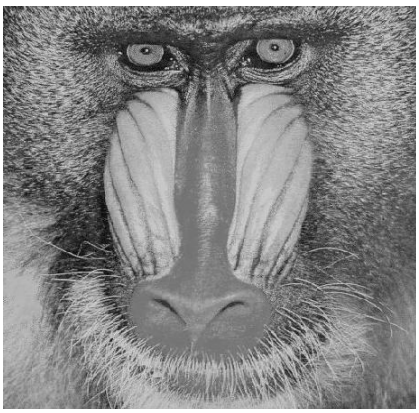
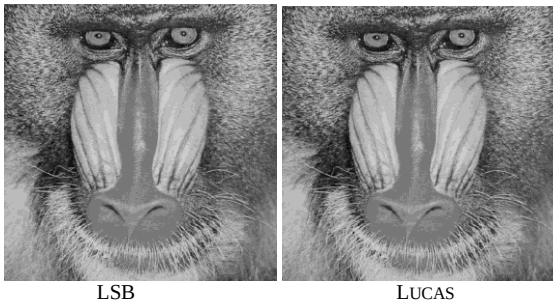


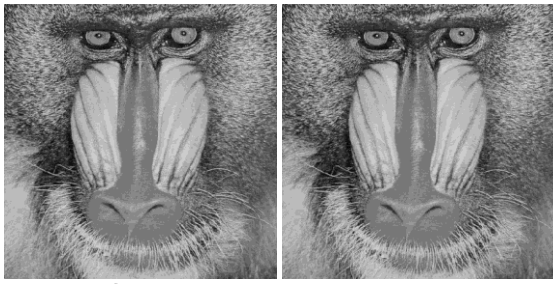
Fig.1. ORIGINAL GRAY SCALE IMAGE

The performance of the Embedding Techniques is shown at Figures (2-9). The quality of the covered image using the LSB hiding technique is degrade using higher bit plane for embedding. On the other hand, Lucas hiding technique maintains a better image quality for all cases. Moreover, Table 7 shows that Lucas hiding technique outperforms the LSB in achieving better Signal to Noise Ratio (PSNR) duo to the fact that digits in Lucas number system are less significant than those in binary system. Conversely, the first bit in the binary system is less significant than that of the Lucas number system , thus the LSB achieves better PSNR than the Lucas hiding technique with using the first bit for imbedding which is consider the least security bit plane for data hiding.



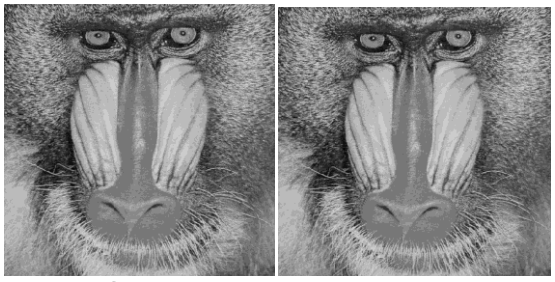
LSB LUCAS

Fig.2. EMBEDDING TECHNIQUES PERFORMANCE (1ST BIT)



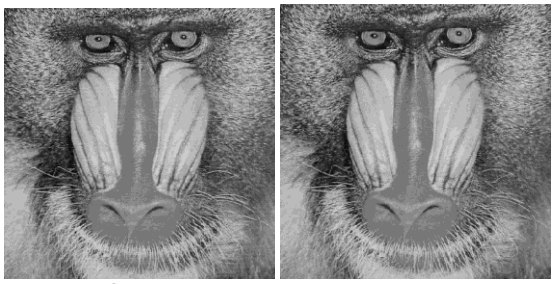
LSB LUCAS

Fig.3. EMBEDDING TECHNIQUES PERFORMANCE (2ND BIT)



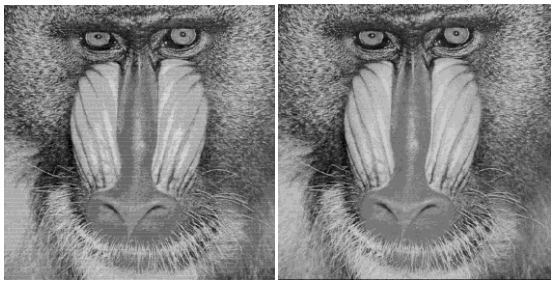
LSB LUCAS

Fig.4. EMBEDDING TECHNIQUES PERFORMANCE (3RD BIT)



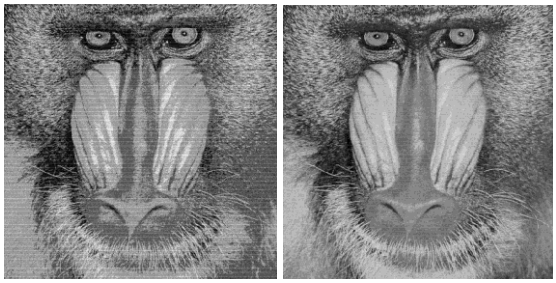
LSB LUCAS

Fig.5. EMBEDDING TECHNIQUES PERFORMANCE (4TH BIT)



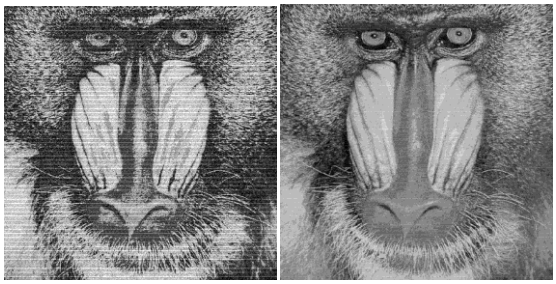
LSB LUCAS

Fig.6. EMBEDDING TECHNIQUES PERFORMANCE (5TH BIT)



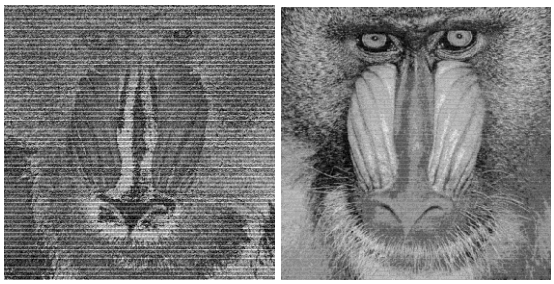
LSB LUCAS

Fig.7. EMBEDDING TECHNIQUES PERFORMANCE (6TH BIT)



LSB LUCAS

Fig.8. EMBEDDING TECHNIQUES PERFORMANCE (7TH BIT)



LSB LUCAS

Fig.9. EMBEDDING TECHNIQUES PERFORMANCE (8TH BIT)

TABLE VII. EMBEDDING TECHNIQUES PERFORMANCE (PSNR)

Bit plane	Peak Signal to Noise Ratio (PSNR)	
	LSB	Lucas
1st	77.0399	54.6944
2nd	50.7404	64.1260
3rd	41.7281	52.1163
4th	35.4584	43.6703
5th	27.3609	38.8316
6th	20.522	33.4006
7th	14.9971	27.7555
8th	8.9157	22.72

B. Performance Evaluation using Color Image and PRNG

In this experiment, the performance of the data hiding techniques are evaluated using a color 512 × 512 × 3 image (Figure 10) for data hiding. The hiding capacity of the original image is 98304 data bytes, where each pixel is used for hiding three data bits, one bit in each color. The embedding process

(Eq. 4) is performed using five Pseudo Random Sequence Generators. The first Generator selects a secret data byte y from the 98304 data bytes and the second Generator picks the bit x to be embedded in the following sequence ($x = 5,7,8,1,2,4,3,6$). The third and the fourth Generators select the position (coulomb j & raw i) of the pixel to be used for embedding as shown at Figures (11-12). The fifth Generator selects the color k in the following sequence ($k = 2,1,3$). Finally, we vary the bit plane b used for embedding from the first bit up to the eighth bit and in each case we hide 98304 data bytes. For performance evaluation, we compute the PSNR for each case. The performance of the LSB Embedding Technique is shown at Figures (13-20). The higher the bit plane used for embedding the higher the impact on the covered image quality. On the other hand, the data hiding system using Lucas numbers and PRNGs maintained a better image quality and enhanced the data security as illustrated at Table 8.

Fig.10. ORIGINAL COLOR IMAGE

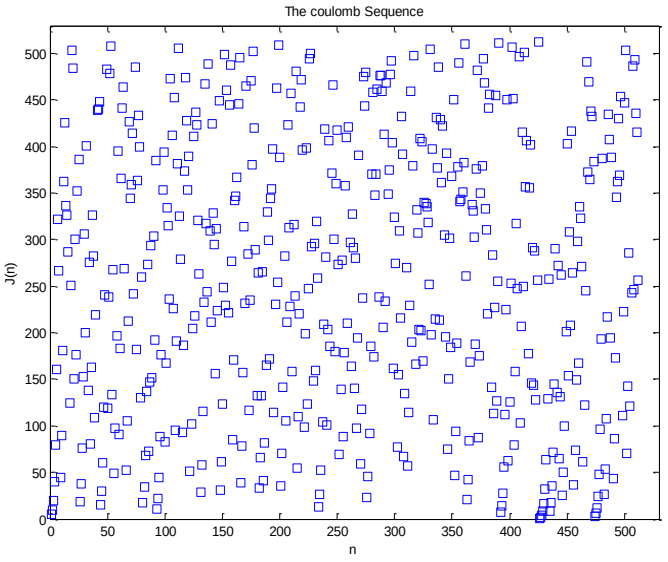


Fig.11. THE THIRD GENERATOR

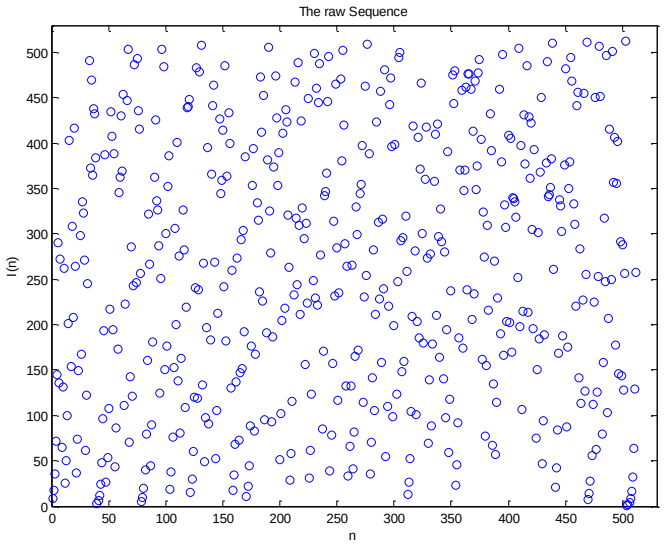


Fig.12. THE FOURTH GENERATOR

LSB LUCAS + PRNG

Fig.13. EMBEDDING TECHNIQUES PERFORMANCE (1ST BIT)

LSB LUCAS + PRNG

Fig.14. EMBEDDING TECHNIQUES PERFORMANCE (2ND BIT)

LSB LUCAS + PRNG

Fig.15. EMBEDDING TECHNIQUES PERFORMANCE (3RD BIT)

LSB LUCAS+ PRNG

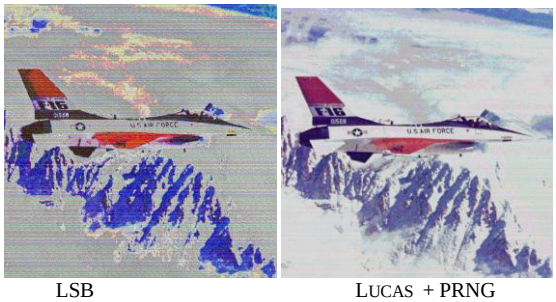
Fig.16. EMBEDDING TECHNIQUES PERFORMANCE (4TH BIT)

LSB LUCAS + PRNG

Fig.17. EMBEDDING TECHNIQUES PERFORMANCE (5TH BIT)

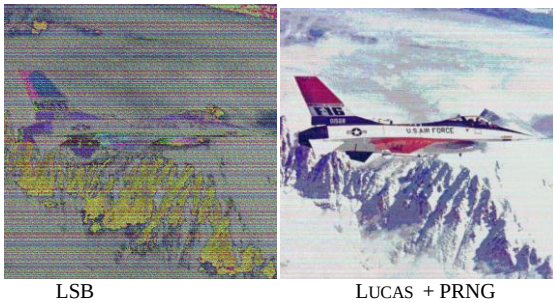
LSB LUCAS + PRNG

Fig.18. EMBEDDING TECHNIQUES PERFORMANCE (6TH BIT)



LSB LUCAS + PRNG

Fig.19. EMBEDDING TECHNIQUES PERFORMANCE (7TH BIT)



LSB LUCAS + PRNG

Fig.20. EMBEDDING TECHNIQUES PERFORMANCE (8TH BIT)

TABLE VIII. EMBEDDING TECHNIQUES PERFORMANCE (PSNR)

Bit plane	Peak Signal to Noise Ratio (PSNR)	
	LSB	Lucas + PRNG
1st	55.4721	53.5292
2nd	51.8705	54.0239
3rd	43.9574	51.1686
4th	36.5404	44.9436
5th	29.5467	40.1102
6th	23.4324	34.9022
7th	15.5457	30.1889
8th	9.2364	25.6393

V. CONCLUSIONS

Steganography is used to communicate an important data in a way that would not be noticeable by others. The least significant bit (LSB) is the most widely used technique for data hiding. The LSB process is simple but not secure. Also, using higher bit plane for hiding data would degrade the covered image's quality. In this paper, we are proposing a novel data hiding method based on Lucas number system. We use Lucas number system to decompose the images' pixels values to allow using higher bit plane for embedding without degrading the image's quality. Also, we enhanced the data security by using Pseudo Random Number Generators for selecting the image's pixels, colors and bits used for embedding secret data. Moreover, PRNGs are used to select the secret data bytes and bits to be embedded. The performance of the LSB and the proposed Lucas based method are evaluated by computing the Peak Signal to Noise Ratio (PSNR), where the proposed method achieved better performance than the LSB regarding the image quality and data security.

REFERENCES

[1] S. Katzenbeisser, F.A.P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Norwood, MA, 2000.

[2] W. Bender, D. Gruhl, N. Morimoto, A. Lu, ―Techniques for data hiding‖ IBM Syst. J. 35 (3&4) (1996) 313–336.

[3] Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography, 2Nd Ed. ISBN: 978-0123725851

[4] Guorong Xuan, Yun Q. Shi, Zhicheng Ni, "Reversible data hiding using integer wavelet transform and companding technique," IWDW04, Korea, October 2004.

[5] Mauro Barni, Franco Bartolini, Vito Cappellini,Alessandro Piva(1998), "A DCT-domain system for robust image watermarking", Signal Processing, Vol. 66 (1998), pp.357–372.

[6] Jessica Fridrich and Miroslav Goljan, "On estimation of secret message length in LSB steganography in spatial domain," in Security, Steganography, and Watermarking of Multimedia Contents VI, Proceedings of SPIE 5306, pp. 23-34, 2004.

[7] Y. Qiudong, X. Liu, "A new LSB matching steganographic method based on steganographic information table", IEEE International Conference on Intelligent Networks and Intelligent Systems, pp. 362-365, 2009.

[8] S.M.M. Karim, M.S. Rahman, M.I. Hossain, "A new approach for LSB based image steganography using secret key", IEEE International

Conference on Computer and Information Technology, pp. 286-191, 2011.

[9] C.H. Yang, C.Y. Weng, S.J. Wang, H.M. Sun, "Adaptive data hiding in edge areas of images with spatial LSB domain systems", IEEE Transactions on Information Forensics and Security, Vol. 3, pp. 488-497, 2008.

[10] K. Ghazanfari, S. Ghaemmaghami, S.R. Khosravi, "LSB++: An improvement to LSB+ steganography", IEEE Region 10 Conference: Tencon 2011, pp. 364-368,2011.

[11] J. R. Hernandez, J. M. Rodr´ıguez, and F. P´erez-Gonz´alez, "Improving the performance of spatial watermarking of images using channel coding," Signal Process. 80(7), pp. 1261–1279, 2000.

[12] Nabin Ghoshal , Jyostna Kumar Mandal, "A Novel Technique for Image Authentication in Frequency Domain using Discrete Fourier Transformation Technique", Malaysian Journal of Computer Science, ISSN 0127-9094, Vol. 21, No. 1, pp. 24-32, 2008.

[13] F. Battisti, K. Egiazarian, M. Carli, and A.Neri, "Data hiding based on Fibonacci-Haar transform," in Mobile Multimedia/Image Processing for Military and Security Applications, SPIE Defense and Security, Vol. 6579, May 2007.

[14] Chi-Kwong Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," Pattern Recognition, pp. 469–474, Mar. 2004.

[15] Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography, 2Nd Ed. ISBN: 978-0123725851

[16] Swanson, M. D., Kobayashi, M., Tewfik, A. H.: Multimedia Data-Embedding and watermarking Technologies, Proc. IEEE, vol. 86, 1064 – 1087, 1998

[17] N. Johnson, Digital Watermarking and Steganography: Fundamentals and Techniques , The Computer Journal. (2009)

[18] G P. Ribenboim, My Numbers, My Friends, Springer, 2000, ISBN 0-38798911-0

[19] L. E. Dickson, "Recurring Series; Lucas' Un, Vn," History of the Theory of Numbers: Divisibility and Primality, Dover Publications, New York, Vol. 1, 2005, pp. 393-411.

[20] Brown, J. L. Jr. "Zeckendorfs Theorem and Some Applications", Fib. Quart. 2, 16 3-168, 1964.

[21] Phillips G.M., "Zeckendorf representation", in Hazewinkel, Michiel, Encyclopaedia of Mathematics, Springer, ISBN 978-1556080104, Picione, 2001.

[22] L. T. Wang and E. J. McCluskey, "Linear feedback shift register design using cyclic codes," IEEE Trans. Computer., vol. 37, pp. 1302-1306, Oct. 1988.

[23] A. Fuster and L. J. Garcia, "An efficient algorithm to generate binary sequences for cryptographic purposes," Theoretical Computer Science, vol. 259, pp. 679-688, May 2001.