International Journal of Information Technology and Knowledge Management
July-December 2008, Volume 1, No. 2, pp. 497-510

ENHANCED CHAOS BASED SPREAD SPECTRUM INFORMATION SECURITY

A. Kumar & M. K. Ghose

Abstract

An Enhanced version of chaos based spread spectrum image steganography [1] has been proposed by incorporating chaos-based encryption, error correction code and chaotic modulation in spread-spectrum image Steganography. In proposed method, an enhanced chaos-based encryption, using an external secret key of 256-bits, and piecewise linear chaotic map (PWLCM-s) are employed. The initial condition values of PWLCM are derived using external key. To make encryption more robust, the secret key is modified after encrypting each block of 256 bits. The pseudo random sequence derived from PWLCM used for steganography and chaotic modulation. Findings confirm that the proposed method is secure from various types of attacks, exhibits good encryption speed and better security performance.

Keywords: Information Security, Chaos, Steganography, Turbo codes.

1. Introduction

While transmitting the information over the internet one needs to protect the important information from unauthorized users by using various encryption techniques [2, 3, and 4].

Most of the conventional encryption algorithms such as AES, DES, and RSA etc. are not suitable for image and video encryption [2, 3, and 4] as these techniques require a large computational time and high computing power. Hence these techniques are not preferred for image or video encryption.

For this reason various fast image encryption algorithms proposed, chaos based encryption is one of them. There exists relationship between the chaos and cryptography such as 1) Ergodicity and confusion. 2) Sensitivity to initial condition and diffusion with a small change in the secret key or plain text. 3) Mixing property and diffusion. 4) Deterministic dynamics and deterministic pseudo-randomness. 5) Structure complexity and Algorithm complexity. Hence, a large number of chaos-based cryptosystems has been proposed [5, 6, and 7], but many of the proposed technique lack robustness and security [8, 9, and 10].

Yen-Guo's *et al*. proposed many chaotic image encryption methods [11, 12], by using the basic idea of the chaotic map which serves as a chaotic pseudo random sequence generator (PRNG) and PRNG is used to control secure permutations or substitutions of

pixels. The various tecnqniue like BRIE and TDCEA proposed by Yen-Guo's has been successfully crypt analyzed [13, 14]. The Enhanced 1-D Chaotic Key-based Algorithm (EKCBA) for image encryption proposed by Li et al. [15]. Pareek *et al*. proposed various cryptosystem [16, 17, and 18]. In this technique the initial conditions and for the control parameter derived from the external key the chaotic ciphers proposed by Pareek et al. have been crypt analyzed by Alvarez *et al*. [8 and by Wei *et al.* [19] respectively. A. Kumar *et al.* [1] proposed the image encryption using 128-bit external key and PWLCM (Piece Wise Linear Chaotic Map).The initial condition value derived from the 128-bit external key and to make the cryptosystem more robust by modifying the secret key encrypting each block.

Data hiding has evolved as potential applications for information protection such as access control of digital multimedia (e.g. Watermarking) [20, 21], secret communications (e.g. Steganography) [22, 23], temper detection, and others. It is the art of hiding a message signal in a host signal without any perceptual distortion of the host signal.

Spread spectrum image steganography (SSIS) is a method involving the use of spread spectrum communication techniques like error-control coding and modulation in Steganography [24].

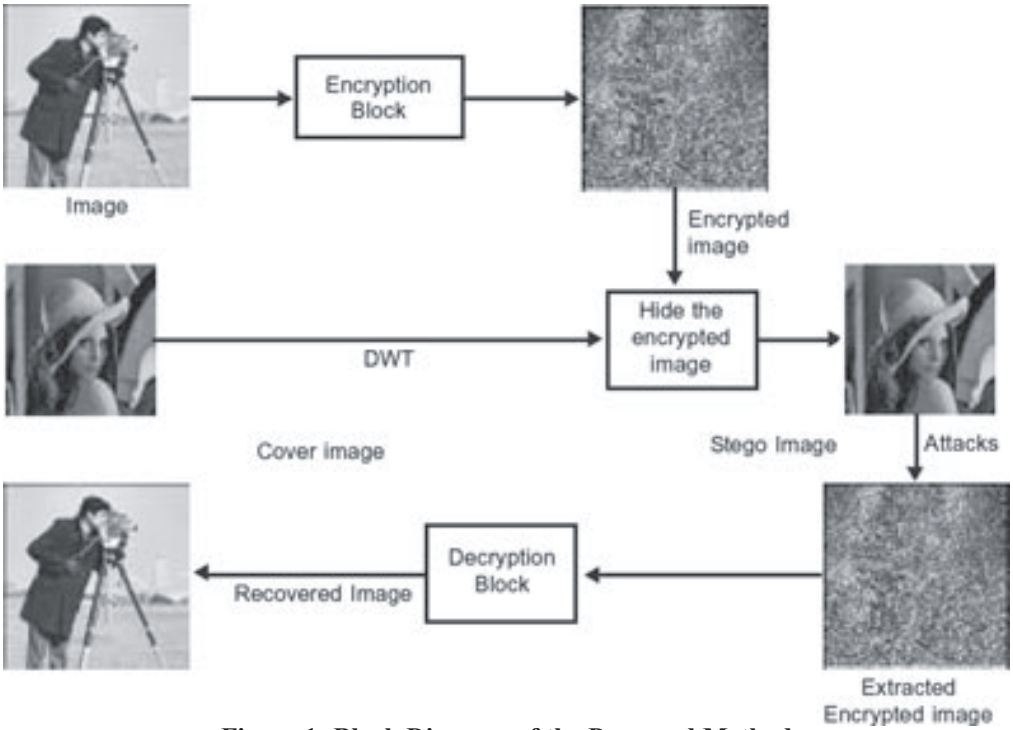


Figure 1: Block Diagram of the Proposed Method

Satish *et al* [25], they encrypted the message using the chaotic sequence and encoded with error correction code, further they modulated the output using chaotic sequence and embedded signal interleaved using the key. A. Kumar *et al*. [1] modified this technique by using PWLCM (Piece Wise Linear Chaotic Map) and by modifying the cox's method [26].

2. PROPOSED METHOD

Here, the main components and their functionality, operation and data flow of each component are discussed. The propose scheme is shown in figure 1 which is composed of the encryption, embedding, and decryption system.

2.1. Encryption Process

In encryption the image data is divided into blocks of 256-bit each. The secret key is used to generate the initial condition and threshold values. The two pseudorandom sequences are generated using PWLCM based on generated initial condition and threshold values. Further, data is divide into two blocks of 128-bits each (First block consists of all odd number bytes and second block all the even number bytes) as well as secret key in same manner. Each block is encrypted using pseudorandom sequences. The number of rounds depends upon the secret key. The secret key is modified after completion of each round.

The procedure of the encryption is discussed step by step as follows:

(1) The proposed process uses the 256-bit external secret key. The secret key is divided into a block size of 8-bit each

$$K = K_1 K2 \ldots K_{32} \tag{1}$$

where K_i represents one block of 8-bit each of the secret key.

(2) Here, PWLCM (Piecewise Linear Chaotic Maps) are employed as follows

$$X_{n+1} = C_{\mu_x}(X_n) \tag{2a}$$

$$Y_{n+1} = C_{\mu_y}(Y_n) \tag{2b}$$

The initial condition values (X_0, Y_0), threshold values (μ_x, μ_y) are calculated using mathematical manipulation based on secret key.

(3) Divide the secret key into two keys of 128-bits each in such a way that Key_1 as all odd numbers and Key_2 as all even numbers and further manipulate the keys.

$$Key_1 = K_1 K_3 \ldots K_{31} \tag{3a}$$

$$Key_2 = K_2 K_4 \ldots K_{32} \tag{3b}$$

$$K_x = Key_1 \oplus Key_2 \tag{3c}$$

$$K_y = Key_1 \otimes Key_2 \tag{3d}$$

$$K_x = K_{x1}\, K_{x2} \ldots K_{x16} \tag{3e}$$

$$K_y = K_{y1} K_{y2} \ldots K_{y16} \tag{3f}$$

(4) (a) The initial condition value of calculated as:

$$G_1 = \begin{pmatrix} (K_{x1})_{10} + (K_{x5})_{10} + \\ (K_{x9})_{10} + (K_{x13})_{10} \end{pmatrix} \text{mod}256 \tag{4a}$$

$$G_2 = \begin{pmatrix} (K_{x2})_{10} + (K_{x6})_{10} + \\ (K_{x10})_{10} + (K_{x14})_{10} \end{pmatrix} \text{mod}256 \tag{4b}$$

$$G_3 = \begin{pmatrix} (K_{x3})_{10} + (K_{x7})_{10} + \\ (K_{x11})_{10} + (K_{x15})_{10} \end{pmatrix} \text{mod}256 \tag{4c}$$

$$G_4 = \begin{pmatrix} (K_{x4})_{10} + (K_{x8})_{10} + \\ (K_{x12})_{10} + (K_{x16})_{10} \end{pmatrix} \text{mod}256 \tag{4d}$$

Convert the block into a binary string as:

$$B_x = G_{11} G_{12} \ldots G_{21} \ldots G_{48} \tag{5}$$

$$X_0 = \begin{bmatrix} G_{11} * 2^0 + G_{12} * 2^1 + G_{18} * 2^7 + \\ G_{21} * 2^8 + G_{22} * 2^9 + G_{28} * 2^{15} + \\ G_{31} * 2^{16} + G_{32} * 2^{17} + G_{38} * 2^{23} + \\ G_{41} * 2^{24} + G_{42} * 2^{25} + G_{48} * 2^{31} \end{bmatrix} \tag{6}$$

$$X_0 = \left(\left(X_{0 \diagup 2^{32}} \right) + 0.2 \right) \text{ mod}0.9 \tag{7}$$

(b) The initial condition value of Y_0 calculated as:

$$H_1 = \begin{pmatrix} (K_{y1})_{10} + (K_{y5})_{10} + \\ (K_{y9})_{10} + (K_{y13})_{10} \end{pmatrix} \text{mod}256 \tag{8a}$$

$$H_2 = \begin{pmatrix} (K_{y2})_{10} + (K_{y6})_{10} + \\ (K_{y10})_{10} + (K_{y14})_{10} \end{pmatrix} \text{mod}256 \tag{8b}$$

$$H_3 = \begin{pmatrix} (K_{y3})_{10} + (K_{y7})_{10} + \\ (K_{y11})_{10} + (K_{y15})_{10} \end{pmatrix} \text{mod}256 \tag{8c}$$

$$H_4 = \begin{pmatrix} (K_{y4})_{10} + (K_{y8})_{10} + \\ (K_{y12})_{10} + (K_{y16})_{10} \end{pmatrix} \text{mod}256 \tag{8d}$$

Convert the block into a binary string as:

$$B_y = H_{11}H_{12} \ldots H_{21} \ldots H_{48} \tag{9}$$

$$Y_0 = \begin{bmatrix} H_{11} * 2^0 + H_{12} * 2^1 + H_{18} * 2^7 + \\ H_{21} * 2^8 + H_{22} * 2^9 + H_{28} * 2^{15} + \\ H_{31} * 2^{16} + H_{32} * 2^{17} + H_{38} * 2^{23} + \\ H_{41} * 2^{24} + H_{42} * 2^{25} + H_{48} * 2^{31} \end{bmatrix} \tag{10}$$

$$Y_0 = \left(\left(Y_{0/2^{32}} \right) + 0.2 \right) \text{ mod}0.9 \tag{11}$$

(c) The threshold values calculated as:

$$\mu_x = (\sqrt{X_0 * Y_0}) \text{mod}(0.5) \tag{12a}$$

$$\mu_y = \left(\sqrt{\frac{X_0}{Y_0}} \right) \text{mod}(0.5) \tag{12b}$$

(5) Generate two pseudorandom sequence of 128-bit each after leaving the first 1000 iterations and consider the value between 0.2 and 0.8 only as i_k and j_k on the bases of the initial condition and threshold values

The real numbers is converted into binary sequence using the following equations

$$L_k = (\text{int } (23 * (i_k - 0.2)/0.9) + 1) \text{ mod}1 \tag{13a}$$

$$M_k = (\text{int } (23 * (j_k - 0.2)/0.9) + 1) \text{ mod}1 \tag{13b}$$

Where $k = 1, 2, \ldots, 128$.

(6) Read 32 bytes from the image file, and divide these 32 bytes into two blocks of 16 bytes each by taking all odd number bytes in block one and all even number bytes in block two.

(7) Modify each block using byte substitution for creating confusion.

(8) First block XOR with L_k [1 – 128)] and second block ex-or with M_k (1 – 128].

(9) XOR First block with K_x and second block with K_y.

(10) Again modify each block using byte substitution, XOR second block with L_k [1–12], XOR first block with M_k[1 –128], and XOR first block K_y and XOR second block with K_x.

(11) Repeat the steps 7 to 10

$$\text{int} = \begin{pmatrix} (K_{y1})_{10} + (K_{y16})_{10} \\ (K_{x1})_{10} + (K_{x16})_{10} \end{pmatrix} \tag{14}$$

(12) Apply turbo code [27] to both blocks.

(13) After encrypting one block of 256-bit modify the secret key and divide secret key into two parts as mentioned in step 3.

(14) Take $X_0 = {}_{i128}$, $Y_0 = j_{128}$, $\mu_x = \mu_x * \mu_y \text{ (mod(0.5))}$, (15)

$$\mu y = \left(\mu_x / \mu_y\right)(\text{mod}(0.5)) . \tag{16}$$

(15) Repeat the steps 5 to 14 until whole file is encrypted.

Decryption is exact opposite of the encryption process.

2.2. Hiding Information

2.2.1. Spread Spectrum using Chaotic Shift Keying (CSK)

The encrypted message is further modulated using chaotic shift keying concept, so that there will be minimum effect of error or noise added during transmission of the data.

In CSK

If (data bit ==1) then

Symbol generated by oscillator is transmitted.

Else

Inverted Symbol is transmitted.

PWLCM used in this case

$$Z_{n+1} = C_{\mu_z}(Z_n) \tag{17}$$

Where the threshold and initial condition values are derived as follows:

$$Z_0 = \left(\left(\left(\sum_{q=1}^{16} \begin{pmatrix} K_{xq} \\ +K_{yq} \end{pmatrix}_{10} \right) \bmod(256) \right) 0.25 \right) \bmod 1 \tag{18}$$

$$\mu_z = \left(\left(\left(\sum_{q=1}^{16} \begin{pmatrix} K_{xq} \\ *K_{yq} \end{pmatrix}_{10} \right) \right) 0.25 \right) \bmod(0.5) \tag{19}$$

2.2.2. Hiding Data

Cover image converted to *YIQ* format and take the 4-level wavelet transform and obtain the highest '*m*' component in the LH-band. The data is hiding in the selected wavelet components' with the modified Cox's method [1].

β is the pseudorandom sequence (act as key during data hiding) which is calculated using PWLCM the initial condition value are

$$\beta_0 = \left(\left(\left(\prod_{q=1}^{16} \begin{pmatrix} K_{xq} \\ +K_{yq} \end{pmatrix}_{10} \right) \bmod(256) \right) 0.25 \right) \bmod 1 \tag{20}$$

$$\mu_\beta = \left(\left(\left(\prod_{q=1}^{16} \begin{pmatrix} K_{xq} \\ *K_{yq} \end{pmatrix}_{10} \right) \right) 0.25 \right) \bmod(0.5) \tag{21}$$

$$\beta_{n+1} = C_{\mu\beta}(\beta_n) \tag{22}$$

3. ANALYSIS OF THE PROPOSED METHOD

An encryption technique and robust steganography should be resistant to various types of attacks. Various security analysis of the proposed method performed in this section.

3.1. Statistical Analysis of the Encryption

For a perfect cipher system, cipher should be robust against any type of statistical attack. Hence, various statistical analyses performed like histograms, the correlation coefficient for several images and its corresponding encrypted images, and sensitivity analysis for checking the proposed cryptosystem robustness.

(a) Histogram analysis: Analysis of the histograms of several encrypted as well as original images performed. Figure 2(a) original Lena image and its histogram figure

2(c). Figure 2(b) encrypted image and its histogram figure 2(b). Histogram of the encrypted image is nearly uniform and quite different from the original image. Thus, it does not give any clue of statistical attack on the proposed scheme.

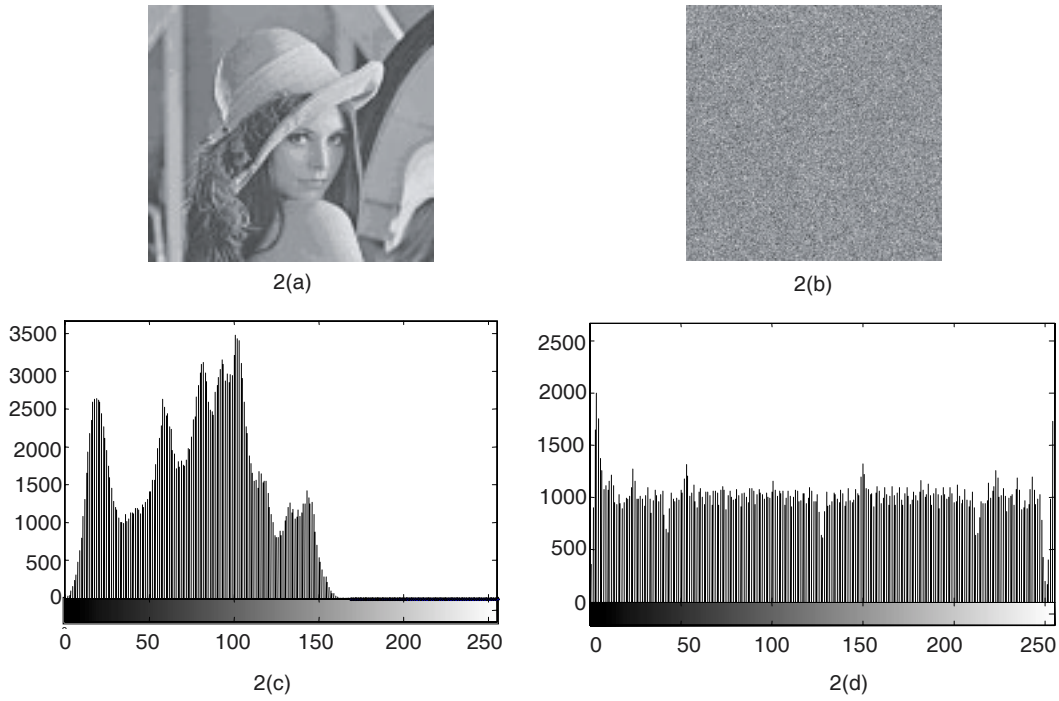


Figure 2: Histogram Analysis: (a) Original Lena Image (b) Encrypted Image (c) Histogram of Original Image (d) Histogram of Encrypted Image

(b) Correlation coefficient analysis: For the calculation of correlation coefficient of adjacent pixels, following formula is used

$$C_r = \frac{N\sum_{j=1}^{N}(x_j \times y_j) - \sum_{j=1}^{N} x_j \times \sum_{j=1}^{N} y_j}{\sqrt{\left(N\sum_{j=1}^{N} x_j^2 - (\sum_{j=1}^{N} x_j)^2\right) \times \left(N\sum_{j=1}^{N} y_j^2 - (\sum_{j=1}^{N} y_j)^2\right)}} \tag{23}$$

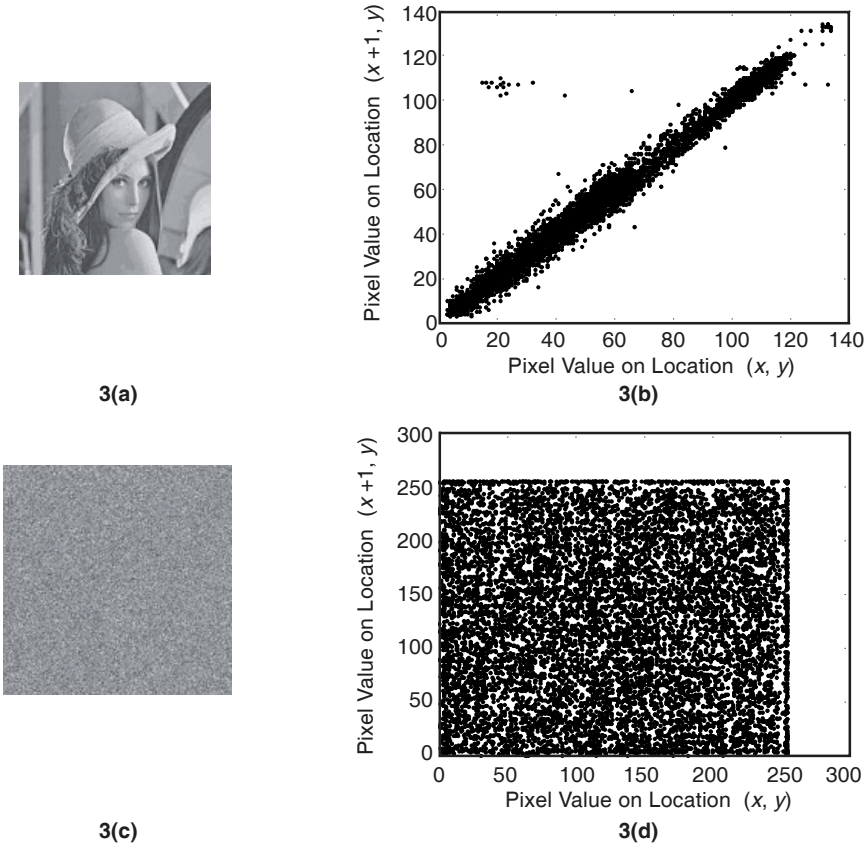
where x and y are the value of the two adjacent pixels of the image and N is the total number of pixels selected from the image.

The correlation coefficients calculated, of the original image its highly correlated and there is negligible correlation in the encrypted image.

Table 1
Correlation Coefficients for the Two Adjacent Pixels in the Original and Encrypted Shown in Fig. 2

	Original Image(Figure 2a)	*Encrypted Image (Figure 2b)*
Horizontal	0.8823	0.0035
Vertical	0.5623	0.0129

The average correlation coefficient between the original image and the encrypted image is very small which implies that there is no correlation between the original image and its corresponding encrypted image. In addition to the histogram analysis, the correlation is analyzed between horizontally adjacent pixels in the several images and their encrypted images. In Fig. 3, shows distribution of two adjacent pixels in the original and encrypted images shown in Fig. 2.



3(a) 3(b)

3(c) 3(d)

Figure 3 Correlation of Two Adjacent Pixels: Frames (b) and (d) Respectively, Show the Distribution of Two Horizontally Adjacent Pixels in the Plain and Encrypted. Where 3(a) & 3(c) are Original and Encrypted Images

3.2. Sensitivity Analysis of the Encryption

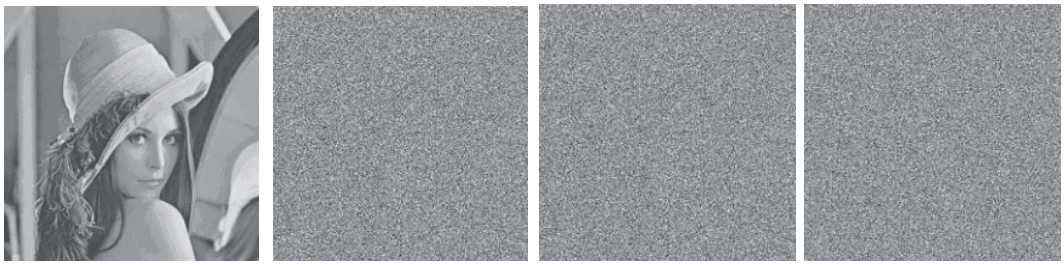
The perfect image encryption method should be sensitive to the secret key i.e. any change in single bit of the secret key should produce a completely different encrypted image. For this purpose any bit of the key is changed and tested for three times. Due to strong avalanche effect of the proposed encryption technique the encrypted images generated are quite different from each other.

(1) The original image figure 4(a) is encrypted by using the secret key 'ABCDEFFEDCBA0123456789FEDCBAFF05ABCDEFFEDCBA 0123456789FEDCBAFF05' as 'K_1' and the resultant image is referred as encrypted image A as figure 4(b).

(2) The same original image is encrypted by using secret key 'BBCDEFFEDCBA0123456789FEDCBAFF05ABCDEFFEDCBA 0123456789FEDCBAFF05' (The first byte of the key is changed) as 'K_2' and the resultant image is referred as encrypted *B* as figure 4(c).

(3) The same original image is encrypted by using secret key 'ABCDEFFEDCBA0123456789FEDCBAFF05ABCDEFFEDCBA 0123456789FEDCBAFF06' (the least significant bit is changed in the secret key) as 'K_3' and the resultant image is referred as encrypted *C* as figure 4(d).

(4) The three encrypted images *A*, *B* and *C* are compared.



(a) (b) (c) (d)

Figure 4: Key Sensitivity Test (a) Original Lena Image (b) Encrypted Image with Key K_1 (c) Encrypted Image using Key K_2 (d) Encrypted Image using Key K_3

Correlation between the corresponding pixels of the three encrypted images is calculated. For this calculation, the MATLAB function corr2 (Compute 2-D correlation coefficient). In the Table 2, the results of the correlation coefficients between the corresponding pixels of the three encrypted images *A*, *B* and *C*. There is no correlation between the three encrypted images.

Table 2
Correlation Coefficients between the Corresponding Pixels of the Three Different Encrypted Images Obtained by using Slightly Different Secret Key of an Image Shown in Figure 3

Image1	*Image 2*	*Correlation coefficient*
Image A	Image B	0.00713
Image B	Image C	0.00281
Image A	Image C	0.00415

3.3. Time Analysis of the Encryption

The speed of the encryption is one of the most important factors. The time analysis has been done on P-4 with 245MB RAM computer. The average time taken by the algorithm for different size of images is shown in the table 3.

Table 3
Average Ciphering Speed of a Few Different Sized Grayscale Images

Image size (in pixels)	*Bits/pixels*	*Average encryption/decryption time(s)*
128 × 128	8	0.05–0.06
256 × 256	8	0.22–0.24
512 × 512	8	0.36–0.47

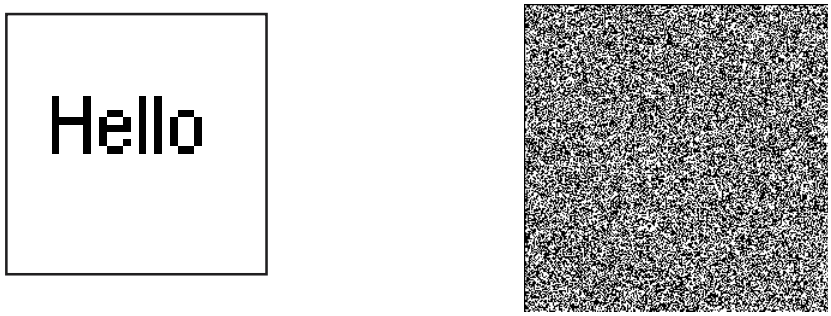
3.4. Key Space Analysis

For any secure image encryption method, the key space should be large enough to make the brute force attack infeasible. The proposed image encryption method has 2^{256} different combinations of the secret key. The image cipher with such a long key is sufficient for reliable practical use. A key longer than this would require more computational time for encryption/decryption.

3.5. Attacks on the Stego-image

The encrypted image Hello.bmp (of size 32*32 in 8-bit colors) as shown in figure 5. The cover image Lena of size 1024 *1024 in 24-bit color. A four level wavelet transform applied to the cover image and hide the encrypted image inside all the four level. Various types of attacks on the stego-image perform like compression, rotation, resize, blurring, etc.

Table 4 summarizes a set of results which are obtained by applying different types of attacks to the stego-image when data hide in LH band of 1st, 2nd, 3rd and fourth level. The PSNR value of the original image and the recovered image is calculated.



5(a) 5(b)

Figure 5: (a) Original Image (b) Encrypted Image

Table 4
Performance of the Proposed Method against Various Attacks (LH band)

Attack name	*Level 1 (PSNR)*	*Level 2 (PSNR)*	*Level 3 (PSNR)*	*Level 4 (PSNR)*
Blurring	0.448493	1.29935	6.11233	15.5298
Contrast	100% recovery	52.9562	24.7823	22.1623
Cropping	2.99272	4.1983	8.92372	74.8407
Dark	47.4239	11.1299	7.6829	7.09425
Deblur	1.78072	9.85182	22.0235	35.9738
Histogram Equalization	36.9426	19.2159	11.5493	9.12835
Median Filter	1.99648	8.12930	14.7742	15.9720
Resize	1.82392	9.92659	15.9264	38.9173
Rotation	1.9476	6.31878	10.9374	12.1983
Weiner Filter	5.9267	9.0126	14.7354	18.6381
Salt	18.97	12.4415	8.9226	12.8739
Jpeg (90%)	33.0275	38.0903	46.8162	49.4029
Jpeg (80%)	16.4886	23.3392	29.2443	37.4999
Jpeg (75%)	14.0918	18.8026	22.1926	32.4368
Jpeg (50%)	11.1666	11.4786	15.6872	29.1932
With different	10.8109	12.4513	10.9872	11.4415

The proposed technique resist the various types of the attacks like JPEG2000 compression, Weiner Filter, Resize, Rotation, various types of noise, and other types of the attacks. The proposed technique is highly robust against the various types of attacks like known-plain text, cipher-text only, chosen text etc.

4. CONCLUSION

The proposed scheme in this paper yields highly robust results. Here, the 256-bit key used as external key which is used to generate the pseudorandom sequence and also used for the data hiding. The external key is modified after encrypting each block.

REFERENCES

[1] A. Kumar, A. Albagul, M. K. Ghose, K. Negrat, "Improved Chaos Based Spread Spectrum Image Steganography", Proceedings of the 10th IASTED International Conference, *Signal and Image Processing* (SIP 2008), August 18-20, 2008 Kailua-Kona, HI, USA, 350–355.

[2] J. Daemen and V. Rijmen. The Design of Rijndael, AES - Advanced Encryption Standard}, ISBN 3-540-42580-2 (Springer-Verlag Berlin Heidelberg, New York).

[3] "Data Encryption Standard (DES)," National Bureau Standards FIPS Publication **46**, (1977). R. L. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems, *Commun. Assoc. Comput.* (1978), 120–126.

[4] C. W. Wu, L. O. Chua, A Simple Way to Synchronize Chaotic Systems with Applications to Secure Communication Systems, *Int. J. Bifurcat. Chaos* **3** (6), (1993), 1619–1627.

[5] R. Lozi, L. O. Chua, Secure Communications via Chaotic Synchronization. II. Noise Reduction by Cascading Two Identical Receivers, *Int. J. Bifurcat*. *Chaos* **3** (5), (1993), 1319–1325.

[6] S. Li, X. Mou, Y. Cai, Improving Security of a Chaotic Encryption Approach, *Phys. Lett. A* **290**, (3–4), (2001), 127–133.

[7] G. ´Alvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of a Discrete Chaotic Cryptosystem using External Key, *Phys. Lett.* A **319**, (2003), 334–339.

[8] S. Li, X. Mou, Y. Cai, Z. Ji, J. Zhang, On the Security of a Chaotic Encryption Scheme: Problems with Computerized Chaos in Finite Computing Precision, *Comput. Phys. Commun.* **153**, (2003), 52–58.

[9] S. Li, X. Mou, B. L. Yang, Z. Ji, J. Zhang, Problems with a Probabilistic Encryption Scheme Based on Chaotic Systems, *Int. J. Bifurcat. Chaos* **13**, (10), (2003), 3063–3077.

[10] H. C. Chen, J. I. Guo, L. C. Huang, and J. C. Yen, "Design and Realization of a New Signal Security System for Multimedia Data Transmission," *EURASIP J. Appl. Signal Process*, 2003, (13), (2003), 1291–1305.

[11] J.-C. Yen and J.-I. Guo, "Design of a New Signal Security System," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS '02)*, **4**, 121–124, Scottsdale, Ariz, USA, May 2002.

[12] S. Li and X. Zheng, "On the Security of An Image Encryption Method," in *Proc. IEEE International Conference on Image Processing (ICIP '00)*, Rochester, NY, USA, *http://www.hooklee.com/pub.html*, **2**, (September 2002), (925–928).

[13] Chengqing Li, Shujun Li (Corresponding author), Guanrong Chen, Gang Chen and Lei Hu, "Cryptanalysis of a New Signal Security System for Multimedia Data Transmission," *EURASIP Journal on Applied Signal Processing*, vol. 2005, (8), 1277–1288, 2005.

[14] Socek ,D; Shujun Li; Magliveras, S.S.; Furht, B.; Enhanced 1-D Chaotic Key-Based Algorithm for Image Encryption, International conference on Security and privacy for Emerging Areas in Communications Network, 2005, SecureComm 2005, 5-9 Sept 2005, 406-407.

[15] N.K. Pareek, Vinod Patidar, K.K. Sud, Discrete Chaotic Cryptography using External Key, *Phys. Lett*. A 309, (2003), 75–82.

[16] N.K. Pareek, Vinod Patidar, K.K. Sud, Cryptography using Multiple Onedimensional Chaotic Maps, Commun. *Nonlinear Sci. Numer. Simul*. **10** (7), (2005), 715–723.

[17] N.K. Pareek, Vinod Patidar, K. K. Sud, Image Encryption using Chaotic Logistic Map, *Image and Vision Computing* **24**, (2006), 926–934.

[18] J. Wei, X. Liao, K.-W. Wong, T. Zhou, Cryptanalysis of a Cryptosystem using Multiple One-Dimensional Chaotic Maps, *Communications in Nonlinear Science and Numerical Simulation*, in press, doi: 10.1016/j.cnsns.2005.06.001 (2006).

[19] Jiwu Huang, Yun Q. Shi, Yi Shi, "Embedding Image Watermarks in DC Components," *IEEE Trans. CSVT* **10** (6), (2000), 974–979.

[20] Shinfeng D. Lin, Chin-Feng Chen," A Robust DCT-based Watermarking for Copyright Protection," *IEEE Trans. Consumer Electron*. **46** (3), (2000), 415–421.

[21] W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Techniques for Data Hiding. *IBM Systems Journal,* **35**, (3-4), (1996). 313–336

[22] N.F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," *IEEE Computer*, **31**, (2), February 1998, 26–34.

[23] L. M. Marvel, C. G. Boncelet, and C. T. Retter, "Spread Spectrum Image Steganography," *IEEE Trans. Image Processing*, **8**, (Aug. 1999), 1075–1083.

[24] L. M. Marvel, C. G. Boncelet, and C. T. Retter, "Spread Spectrum Image Steganography," *IEEE Trans. Image Processing*, **8**, (Aug. 1999), 1075–1083.

[25] Satish, K.; Jayakar, T.; Tobin, C.; Madhavi, K.; Murali, K.; Chaos based Spread Spectrum Image Steganography Consumer Electronics, *IEEE Transactions* **50** (2), (May 2004), 587–590.

[26] I. J. Cox, J. Kilian, F. Thomson, T. Shamoon , Secure Spread Spectrum Watermarking for Multimedia, *IEEE Trans of Image Processing*, **6**(12), (1997) 1673–1687.

[27] A. Kumar and N. Rajpal, "Turbo Codes for Error Control Code", *J. Inform. Technol*., **2**, (2004), 29–33.

A. Kumar & M. K. Ghose
Department of Computer Sc. & Engg.
SMIT, Majitar
Sikkim, INDIA
E-mail: *kumarsmu@yahoo.com*
mkghose2000@yahoo.com