© 2016 IJSRSET | Volume 2 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section : Science and Technology

New Text Encryption Algorithm Based on Block Cipher and Chaotic Maps

Ekhlas Abass Albhrany[1], Dr. Luma Fayeq Jalil[2], Prof. Dr. Hilal Hadi Saleh[3]

[1]Department of Computer Science, Mustansiriyah University, Baghdad, Iraq
[2,3]Department of Computer Science, University of Technology, Baghdad

ABSTRACT

In this paper, new algorithm for text encryption based on block cipher and chaotic maps is proposed. The proposed algorithm is encrypted and decrypted a block size of (8×8) byte. The nonlinear substitution S-box component that previously designed based on the method in [16]. Each block is first permuted by using 2D Standard map and then substituted by the bytes in S-box. The resulted block is then Xored with the key. A random key generator based on Tent map is proposed to generate the key sequences that used in the encryption and decryption process. The outcomes from key space test, differential assault examination, information entropy test, correlation test of the plaintext and ciphertext characters have demonstrated that the proposed algorithm can oppose cryptanalytic, statistical and brute force assaults, and accomplish more elevated amount of security.

Keywords: text encryption, block cipher, chaotic map, 2D Standard map, Tent map.

I. INTRODUCTION

The principle motivation behind this paper to plan a novel block text encryption algorithm by using chaos theory. Chaos theory reliably assumes a dynamic part in current cryptography. The primary point of interest of the chaos-based method depends on the arbitrary behaviour and the beginning conditions sensitivity. Chaos and cryptography are differenced from each other in light of the way that the chaos systems are characterized just on real numbers [1], while cryptography manages with systems characterized on limited number of integers [2]. In [3,4,5], the close relationship between chaotic maps and cryptosystems has been observed. This relationship can be built up: first chaos ergodicity versus cryptography confusion. Second: sensitivity to the beginning conditions and control parameters of chaotic maps versus cryptosystem dissemination attribute for a little plaintext and mystery key changing. Third: chaotic randomness behaviour can be utilized for producing pseudorandom sequences as a cryptography key. In years ago, individuals utilize the Internet to send and save information in content setup. Web is a confortable media to send data regardless, meanwhile it is hazardous in light of the way that the data are revealed likewise, can be stolen by software engineers to utilize them in an unlawful route as blackmail, theft, warlike purposes, and other. One response for this security issue; its goal is making ciphertext from plaintext utilizing a symmetric calculation (one mystery key).

Numerous researchers have attempted endeavors to explore piece encryption calculation so as to advance short preparing time in encryption and decryption. The DNA traits have been produced for text encryption where the four DNA reason are depicted by binary data, DNA supplement operations are utilized for data encryption; besides, groupings are utilized as secret key. The reference [7] has proposed procedure on matrix scrambling which depends on arbitrary capacity, moving and switching methods of round line. This strategy empowers the dispersion handle and is having a one of a kind method of unscrambling it back to the plaintext and is anything but difficult to actualize utilizing network scrambling system. Ultimately, in [8] present symmetric cipher for text algorithm taking into account disorder; they used a mystery key of 128 piece, two logistic maps with advanced pseudorandom sequences, characteristics of plaintext, and a single

stage of permutation. In this paper, a new block and chaotic encryption / decryption system for text is suggested. The proposed algorithm consists of three transformations which implemented based on the chaotic system.

The paper parts are sorted out as takes after: section 1 introduced basic theory of the chaotic functions, the section 2 discuss the propped algorithm. The statistical and security analysis of the proposed algorithm is achieved in Section 5 and Section 6, before conclusions.

1. Basic theory.

In this paper two chaotic maps are used: 2d standard map and 2d tent map.

1.1 2D Standard map.

The chaotic Standard map was presented in [10], [3], and is defined as: -

$$\begin{cases} a_{i+1} = (a_i + b_i) \bmod 2\pi, \\ b_{i+1} = (b_i + K\sin(a_i + b_i)) \bmod 2\pi, \end{cases} \quad (1)$$

Where a_i and b_i are real values belong to $[0,2\pi)$ and the control parameter K is a positive integer and K> 0. The discretised Standard map was found in a directed method by replacing

$$x = aN/2\pi,\ y = bN/2\pi,\ K = kN/2\pi$$

Into Equation (4), which maps from

$$[0, 2\pi) * [0,2\pi) \text{ to } N * N.$$

After discretization, the map will be

$$\begin{cases} x_{i+1} = (x_i + y_i) \bmod N, \\ y_{i+1} = (y_i + K\sin\frac{x_{i+1}N}{2\pi}) \bmod N, \end{cases} \quad (2)$$

This discretised Standard map properties may not be tantamount to the first one, but it can be carried out in the domain of integer numbers, which decreases the complexity of computational. Also it is more proper for encryption of real-time data. The standard map utilized to execute the scrambling of data [3].

The corners pixels in standard map of a square image have some specific characteristic. For example, after the iterations of the map for a number of times, the pixel at (0, 0) location remains unaltered. Keeping in mind the end goal to maintain a strategic distance from it, [11] a strategy to stay away from this shortcoming, the pixels position at the corners (0, 0), (N- 1, 0) , (N- 1,N - 1) and (0,N -1) is adjusted. That is, the natural scan order is changed into arbitrary one. After the chaotic map iteration, an arbitrary-pair (r_x, r_y) is created, which demonstrates the location of an arbitrary chosen pixel in the square image.

Both r_x and r_y parameters belong to range [0.. N-1] and the modified chaotic map becomes

$$\begin{cases} x_{i+1} = (x_i + r_x + y_i + r_y) \bmod N, \\ y_{i+1} = (y_i + r_y + K\sin\frac{x_{i+1}N}{2\pi}) \bmod N. \end{cases} \quad (3)$$

Through the study and the experience of Standard map equation, its inverse is calculated using the proposed equation

$$\begin{cases} y_i = \left(y_{i+1} - r_y - K\sin\frac{x_{i+1}N}{2\pi}\right) \bmod N, \\ x_i = (x_{i+1} - r_x - y_i - r_y) \bmod N. \end{cases} \quad (4)$$

1.2 Tent map.

Tent map is a discrete time chaotic system described by relation [12]:-

$$x_{n+1} = \begin{cases} \frac{x_n}{a}, & \text{if } x \in [0, a], \\ \frac{(1 - x_n)}{(1 - a)}, & \text{if } x \in (a, 0] \end{cases} \quad (5)$$

Where, x_n is represent the system current state and $a \in [0,1]$ is the control parameter. Tent map has uniform invariant probability density in [0, 1] interval.

2. The proposed algorithm.

The proposed algorithm for text encryption consists of two major algorithms: encryption algorithm and `decryption algorithm. Each algorithm has three main steps which are:-

- Create the Substitution S-Boxes.

- Generation of key using the proposed Pseudo Random Number Generator.
- Encryption and Decryption algorithms.

We will describe each step in details in the next section.

2.1 Create the Substitution S-Boxes.

In the proposed algorithm, the S-box that is created based on the method in [13] is used. The proposed S-box is a table of 16 × 16 integer values (256 bytes). The S-box is created by utilizing 2d Logistic map and 2d Cross map.

2.2 The Proposed Pseudo Random Number Generator.

The core of Pseudo Random Number Generator (PRNG) is the Tent chaotic map. Four integer numbers are generated in each round of the generator. The main steps of the proposed PRNG are:

Step 1 : Input the keys of the generator which are the initial condition (x_0) and control parameters (a). These numbers are entered to the Tent map. Each one is floating point number with precision of 10^{-16}.

Step 2 : Iterate the Tent map 100 times and the outcomes are overlooked to remove the chaotic map transient effect.

Step 3 : Iterate Tent map two times. The two outputs are Xored to produce one output.

Step 4 : The resulted floating number output is translated to binary sequence of random length.

Step 5 : The binary sequence is translated to four integer numbers. Each number is in the rang [0..255]. The first number (8-bit number) is started from the bit at the location (1) of the sequence. The second number is started from the bit at the location (10) of the sequence. The third number is started from the bit at the location (20) of the sequence. The last number is started from bit at the location (30) of the sequence.

Step 6 : The steps from step 3 are repeated until the coveted number of keys is reached. When the number of generated keys of one block (64 byte) plus to the control values and parameters of chaotic maps (Standard map (r_1, r_2, k) and Cat map (a, b)) is reached to 69 byte, the parameters of Tent map x_0 and a are modified using simple addition operation between the initial and last value of these parameters in order to increase the complexity of detect the keys.

2.3 Encryption Algorithm

The design tools of the proposed text encryption are based on chaotic map with non-linear transformation functions. The main steps of the proposed encryption algorithm are

Step 1 : Input the plaintext file into T array which is a one dimensional array, the initial parameters (x_0,y_0) to create the S-box and lastly the initial parameter (x_0) and the control value(a) for the PRNG. These parameters numbers are floating point numbers where the precision is 10^{-16} and considered as the keys of the algorithm.

Step 2 : Create the S-box in the method discussed in [13].

Step 3 : Generate the key by using the proposed PRNG algorithm. The generated key is transformed into blocks $K_1K_2K_3K_4…..K_t$, where B_i ($1 \le i \le t$) denotes the i-th key block with size 8×8 byte. In addition, for each block five parameters that are necessary for permutation using Standard map and form byte substituted in S-box are generated also by the proposed PRNG.

Step 4 : T array is divided into blocks $B_1B_2B_3B_4…..B_t$, where B_i ($1 \le i \le t$) denotes the i-th plaintext block with size 8×8 byte. When the last block of the plaintext is less than 8×8 pixels, it treats as special array B (1×L) where L is the number of byte in this block.

Step 5 : For each block do three transformation :-
Permutation transformation: each block is diffused using Standard map.
Mixing transformation: each byte in resulted block is Xored with the byte in the key block.
Substitution transformation: each byte in the resulted block is substituted using S-box.

Step 6 : The output ciphertext is saved in file.

2.4 Decryption Algorithm

The main steps of the decryption algorithm are:-

Step 1 : Input the ciphertext file into C array which is a one dimensional array, the initial parameters (x_0,y_0) to create the S-box and lastly the initial parameter (x_0) and the control value(a) for the PRNG.

Step 2 : Create the S-box.

Step 3 : Generate the key by using the proposed PRNG algorithm. The generated key is transformed into blocks $K_1K_2K_3K_4\ldots K_t$, where B_i ($1\leq i \leq t$) denotes the i-th key block with size 8×8 byte. In addition, for each block five parameters that are necessary for inverse Standard map permutation and for inverse byte substituted in S-box, are generated also by the proposed PRNG.

Step 4 : C array is divided into blocks $B_1B_2B_3B_4\ldots B_t$, where B_i ($1\leq i \leq t$) denotes the i-th ciphertext block with size 8×8 byte.

Step 5 : For each block do three inverse transformations :-

Invers Substitution transformation: each byte in the ciphertext block is substituted using inverse substitute S-box.

Mixing transformation: each byte in the resulted block is Xored with the byte in the key block.

Invers Permutation transformation: each block is return to its original positions using inverse Standard map.

Step 6 : The output plaintext is saved in file.

3. Experiment result.

The proposed algorithm is implemented using Delphi 7 programming language and the tests are performed on a Laptop with an Intel (R) Pentium(R) CPU B960 @2.20 GH and 2 GB RAM running on Windows 8.1.

3.1 Encryption.

The proposed algorithm has the ability to encrypt and decrypt any character from ASCII table. Figure (1) demonstrates a case of plaintext of size 154 characters and its relating ciphertext utilizing the proposed algorithm and the key are: -

For S-box the parameters are:- x_0=0.9542316752453422, y_0=0.2879675436523319.

For the proposed PRNG the parameters are: - kx_0=0.5467654430043221, a=0.3555400254899316.

3.2 The Security Analysis

The security analysis on the proposed encryption algorithm can be done by analyses

Key space analysis: -In order to make brute-force attacks infeasible, the proposed algorithm ought to have an extensive key space. The size a key space that is smaller than 2^{128} is not secures enough [14]. Here, the key space is constructed form the parameters that needs for creating the S-box (initial values x_0 and y_0) and parameters values (kx_0 and a) that are needed for generating the key. These parameters are floating point numbers, where each one belongs to [0, 1]. If the precision of each parameter is 10^{-16}, the total space of keys is $2^{213}((10^{16})^4)$. The key space is adequately far reaching to contradict an extensive variety of brute-force attacks.

Key Sensitivity Analysis: - the secret key sensitivity is very important attribute for any decent cryptosystem. This demonstrates that two ciphertext created utilizing somewhat diverse keys ought to be altogether different. In Table 1 the plaintext of size 305 characters is encoded utilizing three slightly different keys. The ciphertext with inaccurate key does not demonstrate any data related with plaintext, hence the proposed algorithm is sensitive to secret key.

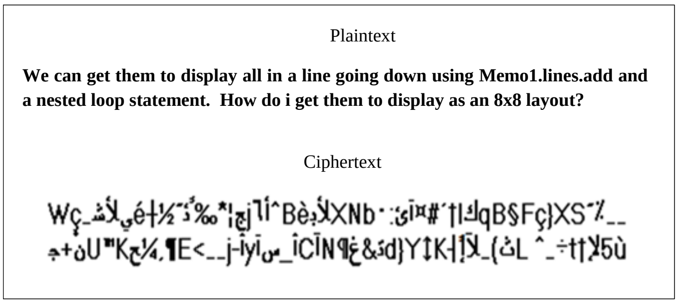

Figure 1: Example of proposed encryption algorithm.

Table 1: Key Sensitivity Example

Plaintext	Encrypted key	Ciphertext
Information security becomes a hot problem which is concerned by our society. Security of the information over the insecure mode of communication, Internet, has been an area of research for years. There have been several techniques developed for encryption/decryption of the information over the years	x_0=0.7698045333215892 y_0=0.1187435556350017 kx_0=0.8869443655400213 a=0.3007654432198765	[illegible]
	x_0=0.7698045333215893 y_0=0.1187435556350017 kx_0=0.8869443655400214 a=0.3007654432198765	[illegible]
	x_0=0.7698045333215892 y_0=0.1187435556350018 kx_0=0.8869443655400213 a=0.3007654432198766	[illegible]

3.3 Statistical Attack Analysis

The statistical analysis of the plaintext and the encrypted im can be considering by:

- Histogram analysis: - indicates how often an image shows up in the content. The histogram can offer information to find the plaintext, the secret key or both. On the off chance that the histogram of the all images in figure content is reasonably equally circulated over the scale, no data about the plaintext can be can be accumulated through histogram examination. The histogram of the plaintext of size 2000 characters is appeared in Figure 2 (a). In Figure 2(b), the ciphertext histogram is appeared; it is uniform, thus the proposed method is powerful against histogram attacks in addition to frequency attacks.
- Correlation coefficient analysis: Correlation evaluation is to check the relationship between plaintext and ciphertext. The distribution of correlation for two horizontally adjacent bytes in the plaintext of size 2000 characters its ciphertext are shown in Figure3 (a)-(b). This Figure shows that the ciphertext is uniform compared with the plain text correlation Figure. In addition the result of correlation coefficient of this plaintext and its cipher text is **-0.0193** which means the correlation between the plaintext and its cipher text is very small.

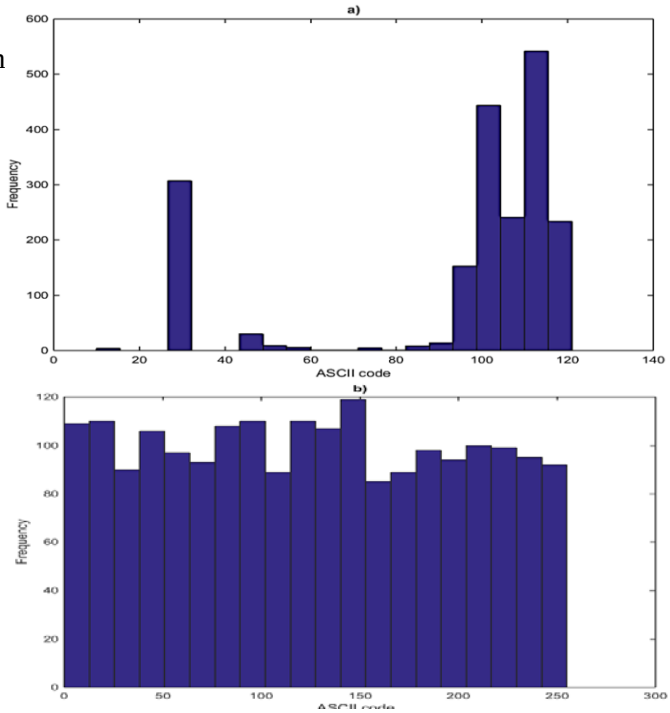

Figure (2): Histograms of : a-plaintext and b-ciphertext

3.4 Differential attack analysis.

A decent encryption algorithm that stays away the known-plaintext and chosen-plaintext attacks ought to have the alluring property where little distinction of the plaintext ought to be diffused to the entire cipher text. In differential attack, attackers frequently roll out a little improvement for the plaintext, and use the proposed algorithm to encrypt for the plaintext before and after changing, through standing out two ciphertexts from make sense of the relationship between the plaintext and the ciphertext. Two measurements are utilized to decide this powerful [6,15]:

- NPCR (Net Pixel Change Rate): - measures the quantity of characters that are diverse between two ciphertexts C1 and C2 from two similar plain texts. The NPCR value is expressed in percentage. When both cipher texts are totally different, the NPCR value is 100%. The NPCR is computed using the following equation:

$$\mathrm{NPCR} = \frac{\sum_{i=1}^{N} W(i)}{N} \times 100\% \qquad (6)$$

Where N is the text length and

$$W(i) = \begin{cases} 0, & \text{if } C_1(i) = C_2(i) \\ 1, & \text{if } C_1(i) \neq C_2(i) \end{cases} \qquad (7)$$

Where $C_1(i)$ and $C_{2(i)}$ are the symbol value of the cipher text C1 and C2.

- UACI (Unified Average Changing Intensity): - is the intensity difference average between two ciphertexts C1 and C2. When UACI is 100% means that both ciphertexts are very different in amplitude. The UACI is calculated as follows:

$$\mathrm{UACI} = \frac{100}{N \times 95} \sum_{i=1}^{N} |C_1 - C_2| \qquad (8)$$

In the proposed algorithm, the NPCR and UACI are acquired with the accompanying steps: to begin with, the plaintexts from Figures (1),(2) and Table 2 are encrypted with the required keys to produce the cipher text C1,C2 and C3; after that, the first symbol of each plaintext is changed to next character and the encryption process is repeated with the same keys to create the new ciphetexts NC1, NC2 and NC3. In Table 2 demonstrates the result of NPCR and UACI.

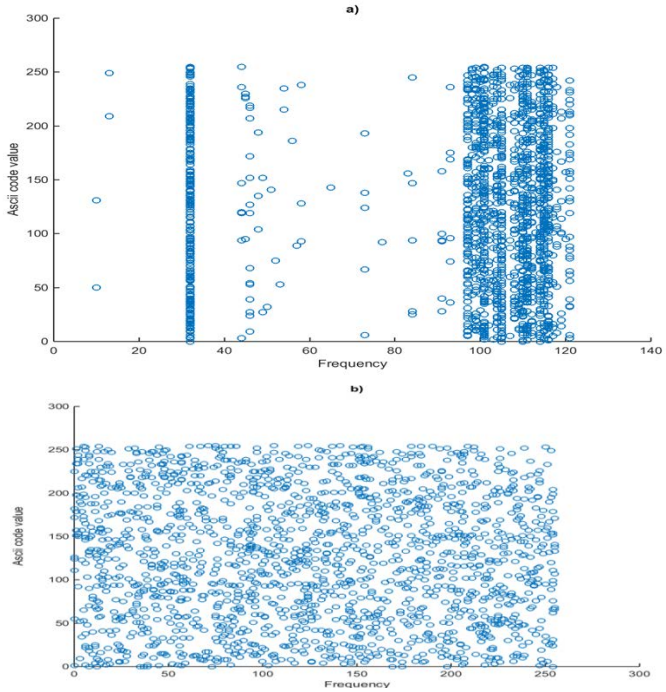

Figure (3) : correlation analyses: a) plaintext correlation and b)ciphertext

Table 2: The results of UACI and NPCR.

plaintext	NPCR	UACI
Plaintext in Figure (1)	99.5500	32.7982
Plaintext in Table (1)	98.6494	33.0866
Plaintext in Figure (2)	99.5000	33.1067

So that, the proposed algorithm is powerful against differential assaults.

3.5 Information Entropy Analysis.

Information Entropy is a scientific hypothesis of information correspondence and capacity. The plaintext information entropy can be computed as [16]: -

$$H(m) = \sum_{i=1}^{2^n-1} p(m_i) \log_2 \frac{1}{p(m_i)} \qquad (9)$$

Where N is the number of bits of the message m, 2^N implies every single possible characters, $p(m_i)$ demonstrates the m_i likelihood and the entropy is represented in bits. If a message is encrypted with 2^N with 2N conceivable character, the entropy ought to be H(m) = N in a perfect world. In the proposed algorithm, there are 255 distinct characters, so the greatest entropy is equivalent to 8.

Table 3: Results of entropy analysis of the proposed algorithm

The ciphertext	Entropy
Ciphertext in Figure (1)	7.99648965524622
Ciphertext in Table (1)	7.19648965524622
Ciphertext in Figure (2)	7.29907499110329

II. CONCLUSION

In this paper, new text encryption method based on combination of a chaotic map and block cipher is presented. The main idea is to encrypt and decrypt a block size of 8X8 byte based on permutation and substitution the byte in S-box. A random key generator based on Tent map generates key sequences that used in the encryption and decryption process. Security analyses demonstrate that the proposed algorithm has attractive attributes such as the key space analysis; statistical attack analysis and differential attack analysis are implemented outwardly and numerically. Test comes about demonstrate that the proposed encryption method is secure because of its in view of its expansive key space; it's highly sensitivity to the cipher keys and plaintext. As a result of all these pleasant attributes the proposed algorithm becomes a good choice for other multimedia encryption, for example sounds, images and even videos.

III. REFERENCES

[1] J. Gickenheimer and P. Holmes, "Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields". Berlin, Germany: Springer, 1983.

[2] B. Schneier, Applied Cryptography: "Protocols, Algorithms, and Source Code in C". New York: Wiley, 1996.

[3] J. Fridrich, "Symmetric Ciphers Based on Two-Dimensional Chaotic Maps", International Journal Bifurcation Chaos, vol. 8, no. 6, June 1998, pp.1259-1284.

[4] L. Kocarev, "Chaos-based cryptography: A brief overview", IEEE Circuits and Systems Magazine, vol. 1, no. 3, pp. 6-21.

[5] X.Y. Wang and Yu Q., "A Block Encryption Algorithm Based on Dynamic Sequences of Multiple Chaotic Systems". Communications in Nonlinear Science and Numerical Simulation, 2009, vol. 14, no. , pp. 574-581.

[6] L. XueJia, L. MingXin, Q. Lei, H. JunSong and F. XiWen, “Asymmetric Encryption and Signature Method with DNA Technology", Science China Information Sciences, 2010, vol. 53, no. 3, pp. 506-514.
[7] M. Kiran Kumar, S. Mukthyar Azam and Shaik Rasool, “Efficient Digital Encryption Algorithm Based on Matrix Scrambling Technique”, International Journal of Network Security & Its Applications (IJNSA), October 2010, vol.2, no.4. pp. 31-41.
[8] M. A. Murillo-Escobar, F. Abundiz-Pérez, C. Cruz-Hernández and R. M. López-Gutiérrez, “novel symmetric text encryption algorithm based on logistic map”, Proceedings of the 2014 International Conference on Communications, Signal Processing and Computers.
[9] E. A. Jackson, Perspectives in Nonlinear Dynamics, Cambridge University Press, vol. 1, Reprint Edition, 1991.
[10] F. Rannou, “Numerical Study of Discrete Plane Area-Preserving Map”, Astron & Astrophys: vol. 31, 1974, pp. 289–301.
[11] S. Lian, J. Sun and Z. Wang, “A Block Cipher Based on a Suitable Use of the Chaotic Standard Map”, Chaos, Solitons and Fractals, 2005, vol. 26, pp. 117–129.
[12] A. Luca, A. Ilyas and A. Vlad, “Generating Random Binary Sequences Using Tent Map”. Proc. IEEE Int. Symposium on Signals, Circuits and Systems (ISSCS), Iasi, Romania, June 30-July 1, 2011, pp. 81-84.
[13] F. J. Luma , H. S. Hilal and A. Ekhlas, “ New Dynamical Key Dependent S-Box based on Chaotic Maps”. IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: vol. 17, N4, 2015, pp. 91-101.
[14] M. François, T. Grosges, D. Barchiesi and R. Erra, “Pseudo- random number generator based on mixing of three chaotic maps”, Communications in Nonlinear Science and Numerical Simulation, vol. 19, no. 4,2014, pp. 887–895.
[15] G. Chen, Y. Mao and C. K. Chui, “A Symmetric Encryption Scheme Based on 3D Chaotic Cat Map”, Chaos, Solitons & Fractals, vol. 21, July 2004, pp. 749-761.
[16] A. Jolfaei and A. Mirghadri, “Image Encryption Using Chaos and Block Cipher”, Computer and Information Science, vol. 4, no. 1, January 2011, pp. 172 – 185.