A novel symmetric text encryption algorithm based on logistic map

M. A. Murillo-Escobar, F. Abundiz-Pérez, C. Cruz-Hernández, R. M. López-Gutiérrez

***Abstract*—Nowadays, text encryption is recommendable when it is transmitted or stored on insecure channels as Internet. By the other hand, the chaotic systems have excellent characteristics as mixing data, ergodicity, sensitivity to initial conditions, and control parameters, etc., all them useful to design cryptographic algorithms with big size keys and an efficient permutation–diffusion process. In this work, we present a novel symmetric text cipher algorithm based on chaos; we use a 128 bit secret key, two logistic maps with optimized pseudorandom sequences, plain text characteristics, and only one permutation–diffusions round. Several security analysis are presented as secret key size, secret key sensitivity, frequency with histograms, autocorrelation analysis, information entropy analysis, differential analysis, classic attacks analysis, and encryption/decryption time. Based in numerical simulation results, the proposed encryption algorithm presents high security, an excellent encryption time, and it can resist a powerful chosen/known plain text attack; therefore, it can be implemented in real-time applications.**

***Keywords*—**chaos, logistic map, permutation–diffusion round, text encryption.

I. Introduction

In last years, people use the Internet to transmit and store data in text format. Internet is a confortable media to transmit data but at the same time it is dangerous because the data are exposed and can be stolen by hackers to use them in an ilegal way as fraud, theft, warlike purposes, and other. Text encryption is one solution to this security problem; its objetive is generate cipher text (unrecognizable text) from plain text (original text) using a symmetric algorithm (one secret key) or asymmetric algorithm (two secret keys). The telecomunications, industry, military, bank, medicine or personal data are some applications where the text encryption is needed.

Currently, 3DES (Triple Data Encryption Standard) and AES (Advanced Encryption Standard) are used as symmetric data encryption, both are accepted as data encryption standard in EE.UU. [1]; AES has advantage as speed, low memory space, easy to implement, and it is based in permutation-diffusion architecture. The RSA (Rivest, Shamir y Adleman) algorithm is an asymmetric data cipher with security advantage but it is slow than any other symmetric cipher. Recently, the DNA (Deoxyribonucleic acid) characteristics have been proposed for text encryption where the four DNA basis are characterized by binary data, DNA complement operations are used to data encryption, and DNA sequences are used as secret key [2]-[7]. By other hand, chaos has excellent properties as mixing data, ergodicity, initial condition sensitivity, control parameters sensitivity, etc., all them are very useful to design cryptographic algorithms for text or images [8]-[14].

In this work, we present a novel symmetric text encryption algorithm based on chaos. The algorithm uses the plain text characteristics to resist a chosen/known plain text attack with just one permutation–diffusion round (change the position and the symbol value) and two logistic maps. Also, a 32 hexadecimal digits (128 bits) secret key is used. In addition, a complete security analysis is realized and we found the proposed algorithm is highly secure, practical in key handling, and it can be implemented in real-time applications. In Section 2, the encryption algorithm is presented. The analisys results are shows in Section 3. Finally, in Section 4 this work is concluded.

II. Proposed Algorithm

We assume a plain text P with length ℓ based in ASCII printable codes (lowercase letter, uppercase letter, space, numbers, and punctuation signs) characterized by 95 characters: *space*!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_‘abcdefghijklmnopqrstuvwxyz{|}~ and each one has a decimal value between $32-126$ according to ASCII table. In Fig. 1, the block diagram of the encryption scheme is shown.

The one-dimensional logistic map is described as follows

$$x_{i+1} = a x_i \left(1 - x_i\right) \tag{1}$$

where $x_i \in (0,1)$ is the discrete state, with initial condition $x_0 \in (0,1)$ and a is the control parameter with $a \in (3.5699, 4)$ to generate chaotic sequences.

This work was supported by the CONACYT, México under Research Grant No. 166654.

M. A. Murillo-Escobar, F. Abundiz-Pérez, and R. M. López-Gutierrez are with the Engineering, Architecture and Design Faculty, Baja California Autonomous University (UABC), Ensenada, B.C., 22860 México (e-mail: murillo.miguel@uabc.edu.mx, abundizf@uabc.edu.mx, roslopez@uabc.edu.mx).

C. Cruz-Hernández is with the Electronics and Telecommunications Department, Scientific Research and Advanced Studies of Ensenada (CICESE), Ensenada, B.C., 22860 México (e-mail: ccruz@cicese.mx).

ISBN: 978-1-61804-215-6

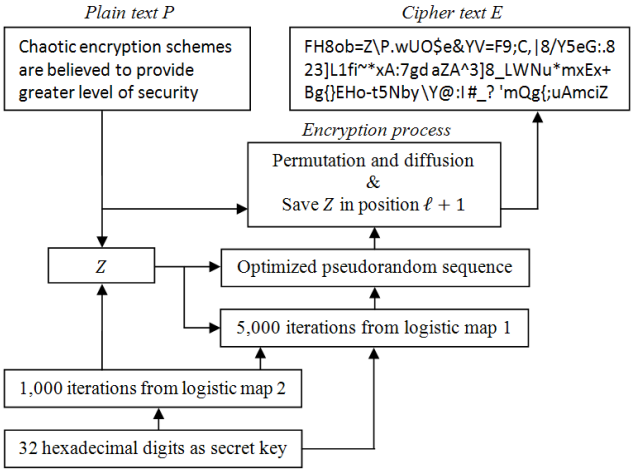

Figure 1: Block diagram of the proposed encryption scheme, where Z is the sum of all plain text characters with chaotic data from logistic map 2.

A. Secret Key Definition

In this section, the secret key is characterized and defined [15]. The initial conditions and control parameters of a chaotic system are not recomendable to be used as secret key. In this work, a secret key of 32 hexadecimal digits is used (128 bits) to calculate the initial condition x_0 and the control parameter a of two logistic maps [11], [16]; with this technique the small key's size from one-dimensional maps is eliminated [17]. TABLE I shows how the initial condition and control parameter are calculated in both logistic maps. In Fig. 2, the largest Lyapunov exponent for all possible secret keys is presented; a positive Lyapunov exponent means a strong key, therefore all secret keys are considered strong.

B. Calculation of Z Value

The Z value is very important to improve the security in the proposed algorithm; in this process, all plain text symbols are summed with chaotic data from logistic map 2 to increase the security against differential attacks. First, the logistic map 2 is iterated $I = \ell + 100$ times by using a_2 and x_{2_0} from TABLE I to generate the chaotic sequence $x^{L2} = \{x_1^{L2}, x_2^{L2}, x_3^{L2}, \ldots, x_I^{L2}\}$ with $x^{L2} \in (0, 1)$ and 10^{-15} decimal precision. After that, the sum of all plain text symbols is calculated as follows

$$S = S + \left[P(i) \times x_{(I+1-i)}^{L2}\right] \bmod 1, \tag{2}$$

where $i = 1, 2, 3, \ldots, \ell$, P represents the plain text in decimal value, S is a variable initialized in zero, and *mod* is the module operation. Then, the Z value needs to be any proportional value between $32 - 126$; this is resolved with the following expresion

$$V = \text{round}(S \times 94) + 32, \tag{3}$$

where *round* is the round operation. Finally, the Z value used on logistic map 1 and encryption process is

$$Z = V/127. \tag{4}$$

TABLE I. CALCULATION OF CONTROL PARAMETER a AND INITIAL CONDITION x_0 IN BOTH LOGISTIC MAPS.

Secret key:	**Control parameter:** a		**Initial condition:** x_0	
32 hexadecimal digits	$\times_1,\times_2,\times_3,\ldots,\times_{32}$ where $\times \in (0-9, A-F)$			
calculations	$A = \frac{(\times_1,\times_2,\ldots,\times_8)_{10}}{2^{32}+1}$	$B = \frac{(\times_9,\times_{10},\ldots,\times_{16})_{10}}{2^{32}+1}$	$C = \frac{(\times_{17},\times_{18},\ldots,\times_{24})_{10}}{2^{32}+1}$	$D = \frac{(\times_{25},\times_{26},\ldots,\times_{32})_{10}}{2^{32}+1}$
logistic map 1	$a_1 = 3.999 + [((A+B+Z) \bmod 1) * 0.001]$		$x_{1_0} = (C+D+Z) \bmod 1$	
logistic map 2	$a_2 = 3.999 + [((A+B) \bmod 1) * 0.001]$		$x_{2_0} = (C+D) \bmod 1$	
ranges	$3.999 < a_{1,2} < 4$		$0 < x_{1,2_0} < 1$	
precision	10^{-15}		10^{-15}	
	where $(a \bmod b) = (a-b) \times (a/b)$ with $b \neq 0$			

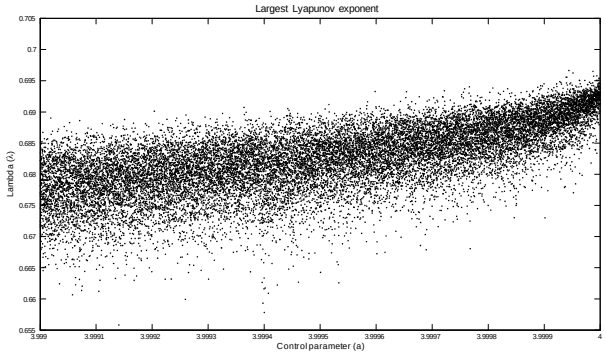

Figure 2: Maximum Lyapunov exponent graphic in $3.999 < a < 4$ range.

C. Encryption Process

The logistic map 1 is iterated $T = 5,000$ times with a_1 and x_{1_0} from TABLE I to generate other chaotic sequence $x^{L1} = \{x_1^{L1}, x_2^{L1}, x_3^{L1}, \ldots, x_T^{L1}\}$ with $x^{L1} \in (0, 1)$ and 10^{-15} decimal precision. After that, a sub-sequence from x^{L1} is calculated according to ℓ plain text symbols as follows

$$PERM_i = \text{round}\left\{\left[\left(x_{(T-\ell+i)}^{L1}\right) \cdot (\ell - 1)\right] + 1\right\}, \tag{5}$$

where $i = 1, 2, 3, \ldots, \ell$, $PERM \in [1, \ell]$ is a vector with length ℓ, *round* is the round operation, and "$\cdot$" represents the multiplication of each value x^{L1} per $(\ell - 1)$. In an efficient permutation process, all plain text symbols must be repositioned. In Equation (5) some values are repeated and it is important to know which ones are repeated to change them by other that are not on it; we look at the repeated values in PERM as follows

$$G_h = \{k_h\}, \; h \ll \ell \tag{6}$$

where k is the missing value from smallest to largest; the vector of missing values G is divided in two parts, after that, each missing value is remplaced by each repeated value in $PERM$ directly in an alternated mode. When this process is completed, we have a pseudorandom sequence $PERM$ with all positions of plain text.

ISBN: 978-1-61804-215-6

A second sub-sequence of length ℓ from x^{L1} is generated to the diffusion process. The logistic map generates many values near to 0 and 1 (Fig. 6(a)) and this would result in an inefficient diffusion process; to avoid this drawback, the logistic map sequence is modified as

$$M_i = \left\{ \left[\left(x^{L1}_{(T-\ell+i)} \right) \cdot (100) \right] + Z \right\} \bmod 1, \qquad (7)$$

where $i = 1, 2, 3, \ldots, \ell$, $M_i \in (0, 1)$ is a vector with length ℓ, *mod* is the module operation, and "$\cdot$" represents the multiplication of each value x^{L1} per 100. The Equation (7) has better distribution (Fig. 6(b)) than data directly from logistic map. Then, M_i data are transformed as follows

$$Y_i = \text{round}\{[M_i \cdot 94] + 1\}, \qquad (8)$$

where $i = 1, 2, 3, \ldots, \ell$, $Y_i \in [1, 95]$ is a vector with length ℓ, *round* is the round operation, and "$\cdot$" represents the multiplication of each value M_i per 94. Finally, the permutation–diffusion process and encryption are calculated as

$$E_i = \{[(P(PERM_i) - 32) + Y_i] \bmod 95\} + 32, \qquad (9)$$

where $i = 1, 2, 3, \ldots, \ell$, $E_i \in [32, 126]$ is the cipher text, $P \in [32, 126]$ represents the plain text in decimal value, and *mod* is the module operation.

D. Z Encryption

The Z value must be used in decryption process but it cannot be calculated from cipher text E directly, so it needs to be added in the cryptogram as follows

$$E_{\ell+1} = V, \qquad (10)$$

where $V \in [32, 126]$.

E. Decryption Process

We assume that the cryptogram is not altered during its transmition. The decrypt process is invert the encrypt process (Fig. 3). First, Z is obtained from the cryptogram with

$$Z = E_{(\ell+1)}/127. \qquad (11)$$

The Z value can be known by any person, but he can not decrypt correctly if it does not has the correct secret key. After that, with the 128 bits correct key and the Z value, $5,000$ chaotic data are calculated in logistic map 1 to determine $PERM$, M y Y as well as in encryption process. Afterwards, the permutation–diffusion process and decryption are calculated with the next expresion

$$D_i(PERM_i) = \{[(E(i) - 32) - Y_i] \bmod 95\} + 32 \qquad (12)$$

where $i = 1, 2, 3, \ldots, \ell$, $D_i \in [32, 126]$ is the plain text recovered, and *mod* is the module operation.

III. Experimental Result

All the analysis are implemented in a Laptop with AMD Turion 64 2.0 GHz CPU, 3.18 GB RAM, and Windows XP 32 bit operative system. MatLab software is used with floating point representation type double (64 bits); therefore we use a 10^{-15} decimal precision.

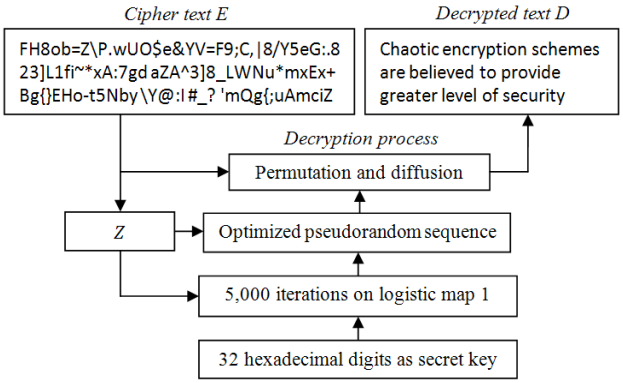

Figure 3: Block diagram of decryption process.

A. Encryption

The proposed algorithm has the capacity to encrypt any printable characters from ASCII table with T maximum length. The first plain text P used is *"Chaotic encryption schemes are believed to provide greater level of security than conventional ciphers"* and the *secret key* used is *"1234567890ABCDEF1234567890ABCDEF"* to generate the cipher text: FH8ob=Z\P.wUO$e&YV =F9;C,|8/Y5eG:.823]L1fi~*xA:7gdaZA^3]8_LWNu*mxEx+B g{}EHo;t5Nby\Y@:I #_? 'mQg{;uAmciZ

B. Secret Key Size Analysis

Nowadays, a cryptosystem must have more than 2^{100} possible secret key (all them strong) to be infallible versus a brute-force attack [15]. The proposed scheme uses 128 bits secret key, therefore the key size is 2^{128} all them strong based in Fig. 2. The key size can be increase to 256 bits with little modifications on the cipher algorithm.

C. Secret Key Sensitivity Analysis

A good cryptosystem must be sensitive at secret keys; this means that two cipher text generated with similar secret key should be very different. In TABLE II(a), the decrypted text is presented with correct secret key; in TABLE II(b)-(c), the decrypted text is presented with two incorrect secret keys but very similar between them. The decrypted text with incorrect key does not show any information related with plain text, therefore the proposed scheme is sensitive to secret key.

D. Frequency Analysis

The frequency analysis is based in histograms. The histogram shows how many times a symbol appears in some text; this kind of attack can give information to find the plain text, the secret key or both. If the histogram of the cipher text has all symbols in a uniform way, the algorithm could resist a frequency attack. In other side, if the histogram is not uniform, it means a weak encryption and it could be subject to a successful frequency attack in the language it was originally encrypted.

ISBN: 978-1-61804-215-6

TABLE II. SECRET KEY SENSITIVITY ANALYSIS: a) DECRYPTED TEXT WITH CORRECT SECRET KEY "1234567890ABCDEF1234567890ABCDEF", b) DECRYPTED TEXT WITH INCORRECT SECRET KEY "12345679**0**ABCDEF1234567890ABCDEF" AND c) DECRYPTED TEXT WITH INCORRECT SECRET KEY "1234567890ABCDEF12345679**0**ABCDEF".

a) Chaotic encryption schemes are believed to provide greater level of security than conventional ciphers
b) 2|b|kE".Qs'SZVh,;n.guh00TNz^&hvqSepPi8}+ Ixk%L8E6#dxwJDsB/=R;*v[1pTwH,dM2f<H^= ,7H!5!57A{czz.>P}kn3*2EDJ
c) p>n]X+1'd>OE1u'rTY[Hl%S6)8EHh4(ABweX evK#pJh)j9s(,2M1/5=Gf\];![HQ(sBJW&+'[oi[t?+) w8^u$<Z=Q;&mNr8vLMet-'^

The plain text (abstract of this work with 1126 characters) histogram is shown in Fig. 4(a). In Fig. 4(b), the cipher text histogram is shown; it is uniform, so the proposed scheme is robust against histogram attacks and letter frequency attacks.

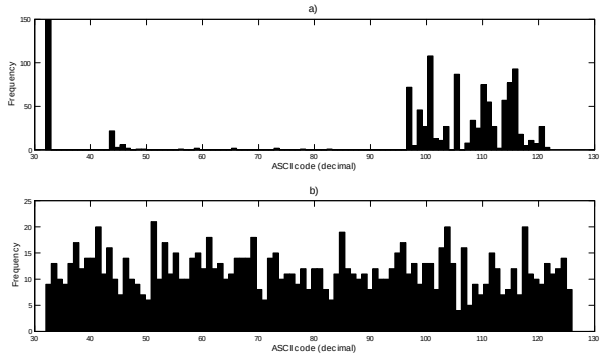

Figure 4: Histograms: a) plain text and b) cipher text.

E. Autocorrelation Analysis

The autocorrelation calculates the similitude between two sequences: the plain text and the plain text displaced t positions. The autocorrelation analysis can be used to determine the secret key length in classic cipher methods. In Fig. 5, the plain text (upper line) and cipher text (lower line) autocorrelation are showed from Fig. 4(a)-(b), respectively. The graphic shows that cipher text autocorrelation is reduced and it is uniform compared with the plain text autocorrelation. Therefore, the proposed scheme can resist an autocorrelation attack.

F. Information Entropy Analysis

When the plain text is encrypted in diffusion stage, the value of all the symbols must be modified. If this process is inefficient, the cipher text would have many identical symbols and it can be subject to an entropy attack. The entropy $H(m)$ from a message m can be calculated as follows

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i)\ \log_2\left(\frac{1}{p(m_i)}\right), \tag{13}$$

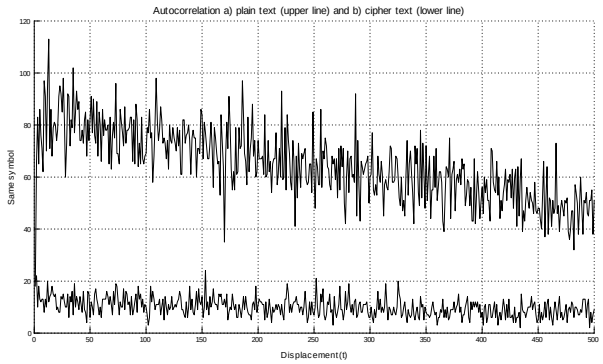

Figure 5: Autocorrelation of a) plain text and b) cipher text.

where N is the number of bits of the message m, 2^N means all possible symbols, $p(m_i)$ represents the probability of m_i, $\log_2$ represent the base 2 logarithm, and the entropy is expressed in bits. If a message m is encrypted with 2^N possible symbols, the entropy should be $H(m) = N$ ideally. In the proposed scheme are 95 different symbols, so the maximum entropy is $H \approx 6.56$.

The plain text entropy (from Fig. 4(a)) is $H(plain\ text) \approx 4.21$ and the corresponding cipher text entropy (from Fig. 4(b)) is $H(cipher\ text) \approx 6.48$, this means all cipher symbols appear with almost the same probability; therefore, the diffusion process is strong.

G. Differential Analysis

In this section, the plain text sensitivity is presented to determine the robustness against differential attacks. The attack is based in the encryption of two similar plain text with same secret key to determine some relation between plain text and cipher text. If a little change in plain image generates a big change in cipher text, then the differential attack results inefficient [12]. There are two metrics to determine this robustness: NPCR (Net Pixel Change Rate) and UACI (Unified Average Changing Intensity); NPCR measures the number of characters that are different between two cipher texts E_1 and E_2 from two similar plain text, the value of NPCR is represented in percentage, where 100% means both cipher texts are totally different. The NPCR is calculated with

$$NPCR = \frac{\sum_{i=1}^{\ell} W(i)}{\ell} \times 100, \tag{14}$$

where ℓ is the text length and

$$W(i) = \begin{cases} 0, & \text{if } E_1(i) = E_2(i) \\ 1, & \text{if } E_1(i) \neq E_2(i) \end{cases} \tag{15}$$

By the other hand, UACI is the intensity difference average between two cipher texts E_1 and E_2, where 100% indicates both texts are totally different in amplitude. The UACI is calculated as follows

$$UACI = \frac{100}{\ell \times 95} \sum_{i=1}^{\ell} |E_1(i) - E_2(i)| \tag{16}$$

ISBN: 978-1-61804-215-6

where ℓ is the text length, $E_1(i)$ and $E_2(i)$ are the symbol value of the cipher text E_1 and E_2. In the scheme proposed, the NPCR and UACI are obtained with the following steps: first, the plain text from Fig. 4(a) is encrypted with one secret key to generate the cipher text E_1; after that, the first symbol of plain text is change from *"I"* to *"J"* and the encryption process is repeated with the same secret key to generate the cipher text E_2. In TABLE III shows the result of NPCR and UACI. Therefore, the scheme proposed is robust against differential attacks.

TABLE III. DIFFERENTIAL ANALYSIS RESULTS OF NPCR AND UACI.

NPCR (%)	98.85
UACI (%)	33.31

H. Robustness Against Classic Attacks

The cryptanalyst knows the encryption algorithm but he does not know the secret key, this is known as Kerckhoffs principle. If a cryptanalyst has access to the encryption machinery, it can implement some differential attacks as cipher text only attack and known/chosen plain text attack [18].

In cipher text only attack, the cryptanalyst has a cipher text and he can decrypt it until find a plain text with sense, therefore is basically a brute force attack. Known/chosen plain text attack are more powerful; its objective is find the secret key to decrypt others cipher texts encrypted with the same secret key. In the proposed scheme, the permutation–diffusion process is realized in one stage; in addition, the chaotic data are calculated from both the secret key and the plain text characteristics. If the same secret key is used to encrypt many plain texts, the cipher texts will result from different chaotic data.

It is known that the logistic map generates many values close to 0 and 1 (Fig. 6(a)) and they are not appropriate to be used in diffusion process; we modified chaotic data in Equation (7) for a better pseudorandom data distribution (Fig. 6(b)) to prevent a known/chosen plain text attack.

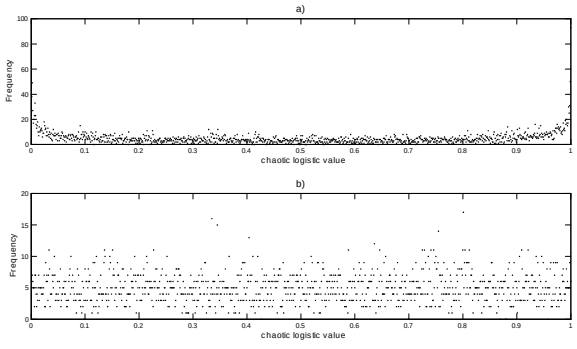

Figure 6: Distribution of 5000 chaotic logistic values with 0.001 of separation, a) data directly from logistic map and b) data used in diffusion process.

I. Encryption Time Analysis

A good encryption algorithm must be robust but it must be fast for real-time applications. The encryption time of 1126 symbols (abstract of this paper) is just 0.140 seconds; the decryption time is 0.109 seconds. Therefore, the algorithm proposed is fast to encrypt text and it can be implemented in real-time applications.

IV. CONCLUSION

A novel symmetric text encryption algorithm based in chaos was presented. A 128 bits secret key was used in a 32 digit hexadecimal format. A complete security analysis was realized over the algorithm with excellent results based in MatLab simulations: it has a big secret key size to resist a force-brute attack, all secret keys are considered strong based in Lyapunov exponent analysis, it is sensitive at secret key and plain text, it can resist an entropy attack, it generates a uniform histogram with low autocorrelation, it is robust against classic attacks, and the encryption time is fast. Therefore, the algorithm proposed can encrypt text with high performance, high security, and it can be implemented in real-time applications.

REFERENCES

[1] Federal Information Processing Standards Publication 197, "Advanced Encryption Standard (AES)", 2001.
[2] L. M. Adleman, "Molecular computation of solution to combinatorial problems", Science, Vol. 266, pp. 1021-2024, 1994.
[3] C. Taylor, V. Risca and C. Bancroft, "Hiding messages in DNA microdots", Nature, Vol. 399, pp. 533-534, 1999.
[4] X. Guozhen, L. Mingxin, Q. Lei and L. Xuejia, "New field of cryptography: DNA cryptography", Chinese Science Bulletin, Vol. 51 No. 12, 1413-1420, 2006.
[5] M. Bordoa and O. Tornea, "DNA secret writing techniques", 8th International Conference on Communications (COMM), 2010.
[6] M. R. Abbasy, A. A. Manaf and M. A. Shahidan, "Data hiding method based on DNA basic characteristics", DEIS 2011, CCIS 194, pp. 53–62, 2011.
[7] L. XueJia, L. MingXin, Q. Lei, H. JunSong and F. XiWen, "Asymmetric encryption and signature method with DNA technology", Sci China Inf Sci, 53: 506–514, doi: 10.1007/s11432-010-0063-3, 2010.
[8] M. Mishra and V.H. Mankar, "Message embedded cipher using 2-D chaotic map", International Journal of Chaos, Control, Modelling and Simulation (IJCCMS) Vol.1, No.1, 2012.
[9] L. Hongjun, W. Xingyuan, "Color image encryption based on one-time keys and robust chaotic maps", Computers and Mathematics with Applications 59, 3320-3327, 2010.
[10] Z. Peng and W. Liu, "Color image authentication based on spatiotemporal chaos and SVD", Chaos, Solitons and Fractals 36, 946–952, 2008.
[11] N.K. Pareek , V. Patidar, K.K. Sud, "Image encryption using chaotic logistic map", Image and Vision Computing 24, 926–934, 2006.
[12] V. Patidar, N.K. Pareek, K.K. Sud, "A new substitution–diffusion based image cipher using chaotic standard and logistic maps", Commun Nonlinear Sci Numer Simulat 14, 3056–3075, 2009.
[13] D. Chen, Y. Chang, "A Novel Image encryption algorithm based on logistic maps", Advances in Information Sciences and Service Sciences, Vol. 3, No. 7, 2011.
[14] X. Wang, L. Teng, X. Qin, "A novel colour image encryption algorithm based on chaos", Signal Processing 92, 1101–1108, 2012.
[15] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems", International Journal of Bifurcation and Chaos, Vol. 16, No. 8, pp. 2129-2151, 2006.
[16] G. Chen, Y. Mao, C.K. Chui, "A symmetric image encryption based on 3D chaotic maps", Chaos Solitons Fractals 21, 749–761, 2004.
[17] H. Gao, Y. Zhang, S. Liang, D. Li, "A new chaotic algorithm for image encryption", Chaos, Solitons and Fractals 29, 393–399, 2006.
[18] T-H Chen, C-S Wu, "Compression-unimpaired batch-image encryption combining vector quantization and index compression", Information Sciences 180, 1690–1701, 2010.