A NOVEL CHAOTIC CIPHERING SYSTEM FOR COLOR DIGITAL IMAGES

K. M. FARAOUN
Evolutionary Engineering and Distributed Information Systems Laboratory, EEDIS
Département d'informatique
Djillali Liabès University. Sidi Bel Abbès – Algeria
Kamel_mh@yahoo.fr

ABSTRACT
Chaotic cryptology has been widely investigated recently. A common feature in the most recent developments of chaotic cryptosystems is the use of a single dynamical map in the encoding–decoding process. The main objective of this paper is to provide a set of chaotic systems instead of a single one for cryptography. We propose a symmetric key stream cipher algorithm in which multiple one-dimensional chaotic maps are used instead of a one-dimensional chaotic map. We demonstrate that the produced key streams have good statistical properties, such as uniform distribution and very height sensitivity to initial conditions. The proposed approaches are applied for color images cryptography. Analysis of the results demonstrates the robustness of the algorithm against different cryptanalysis attacks. The proposed cryptosystem is shown to be robust against brute force attack, plain text attack and differential attacks, and can achieve a ciphering rate of 40Mb/S.

Keywords: Chaotic maps, color image ciphering, keystream encryption, Logistic Map.

1.0 INTRODUCTION

Information security is one of the important issues in the present information age, as there is a phenomenal growth in the rate at which the information is being disseminated. Images are the integral part of the information in engineering and industrial applications as well as in medical processes. A direct and obvious way to protect the information from unauthorized eavesdropping is to use an encryption algorithm to mask the information which has led to the development of various number theory based encryption techniques such as DES, AES, IDEA, RSA, etc [1, 2]. Cryptography is always very important in military and business applications to maintain the secrecy of messages and to prevent information from tampering and eavesdropping. However, this conventional number theory based encryption algorithms do not seem to be appropriate for the images due to some intrinsic features of images such as bulk data capacity, high redundancy, strong correlation among adjacent pixels, etc. To provide a better solution to image security problems, a number of image encryption techniques have been suggested during last one and a half decade. Among them are the techniques based on chaotic dynamical systems [3-4] which provide a good combination of speed, high security, complexity, reasonable computational overheads and computation power, etc. The main advantage using chaos lies in the observation that a chaotic signal looks like noise for the unauthorized users. Chaos is a universal, random-like and robust phenomenon in nonlinear systems. Due to the intrinsic nature of chaos such as sensitivity to initial conditions, deterministic oscillations and noise-like behavior, it has acquired much attention for secure communication and cryptology. Due to its interesting properties, chaos can be connected with those of good ciphers, such as confusion and diffusion [5]. Moreover,

generating chaotic signal is often of low cost with simple iterations, which makes it suitable for the construction of stream ciphers.

Chaotic stream ciphers use chaotic systems to generate pseudorandom key stream to encrypt the plaintext element by element. Different chaotic systems have been utilized to generate such key streams: Forré proposed 2-D Hénon attractor [6], Pareek et al. used generalized logistic map [10] while Behnia et al. introduced piecewise linear chaotic map (PWLCM) [8]. The keystreams can then be generated from the outputs of considered chaotic systems by different post-processing methods. This is done by extracting some bits from chaotic orbits determined by the interval reached by chaotic orbits, by cascading multiple chaotic systems [9], or by coupling chaotic systems [7].

In addition, pseudo-random number sequences are useful in many applications including computer simulations, Monte-Carlo techniques in numerical analysis, test problem generation for the performance evaluation of computer algorithms, statistical sampling, stochastic optimization methods (such as simulated annealing), watermarking for image authentication and cryptography. Conventionally, pseudo-random sequence generators based on linear congruential methods and feedback shift registers are popular but not enough secure for cryptographic purposes. For cryptographic applications, several algorithms such as ANSI X9.17 and FIPS 186 are found to be popular [19]. In recent times, several researchers have been exploring the idea of using chaotic dynamical systems for this purpose [23]. The random-like, unpredictable dynamics of chaotic systems, their inherent determinism and simplicity of realization suggests their potential for exploitation as pseudo-random number generators.

In this paper, we propose a novel approach for a fast and secure encryption of digital images, using a combination of chaotic maps to build key stream generator suitable for cryptographic systems with interesting properties: uniform distribution, statistical pseudo- randomness, height sensitivity to initial conditions variation and a very large key space. Using the proposed sequences generator, an elaborated crypto-system is applied to encipher RGB color digital images. The paper is organized as follows: in Section 2, the proposed chaos-based keystream generator is presented; the experimental results are analyzed and discussed in Section 3 and 4; finally, the conclusions and summary are given in Section 5.

2.0 CHAOS BASED ENCRYPTION SCHEMA

2.1. Pseudo-random sequence generation

Our goal in this work is to design a pseudo-random sequence (key stream) generator, using a special combination of chaotic maps.

Consider a one-dimensional non linear chaotic map $\Gamma_X: I \rightarrow I$ such that $I \subset \Re$, and its corresponding differences equation:

$$X_{m+1} = \Gamma_x(X_m) \quad (1)$$

Given an initial value X_0, $\{X_m, m=1,2,\ldots\ldots\}$ is the corresponding chaotic orbit. Γ_X is a continuous mapping that verifies the mixing propriety, the topological transitivity and the density

of periodic points in I. With a proper choice of the initial condition X_0, the generated orbit will be bounded in a limited region that corresponding to the attractor of the system described by (1).

Consider X_{max} and X_{min} the upper and the lower boundaries of the attractor. Then, partition the region $[X_{min}, X_{max}]$ into N disjoint equal sub-regions $\{R_i, 1 \leq i \leq N\}$ such that :

$$[X_{min}, X_{max}] = \bigcup_{i=1}^{N} I_i \qquad I_i \cap I_j = \varnothing \quad \text{for } i \neq j \tag{2}$$

A random n-ary sequence S of length N $\{S_i, 1 \leq i \leq N\}$ is then generated, and a one to one mapping is created between each element S_i and the region R_i. The sequence values belong to the set {1, 2,…, n} taken with a uniform selection probability. So the number of regions N must be a multiplicand of n to ensure that all values are present with the same proportionality in the sequence, hence ensuring a uniform distribution of the final generated stream.

Originally, the association between the sequence elements $\{S_i, 1 \leq i \leq N\}$ and the regions $\{R_i, 1 \leq i \leq N\}$ is at an agreed setting. Hence, we can set the original sequence to an ordered sequence of N value such that:

$$S_i = (i \bmod n) \text{ for } i = 1 \ldots N. \tag{3}$$

The association will then be:

$$S_1 \rightarrow R_1, S_2 \rightarrow R_2, \ldots\ldots\ldots, S_N \rightarrow R_N \tag{4}$$

When the chaotic equation (1) is iterated, X_i values will be distributed chaotically in the system attractor ($[X_{min}, X_{max}]$) in different manners according to the initial value X_0. At each iteration step *t*, one can chooses the S_i value corresponding to the region R_i such that $X_t \in R_i$ as the output of the stream generator at the time *t*. It has been shown that usually such approach leads the key stream to fall rapidly into a short period [12], which will degrade the randomness quality of the stream. To avoid such behavior, the one to one mapping between the sub regions R_i and the random sequence of elements S_i is changed dynamically after each Δ iterations of the map (1). We use the orbit of another chaotic map $\Gamma_Y: I \rightarrow I$ with corresponding differences equation:

$$Y_{m+1} = \Gamma_Y(Y_m) \tag{5}$$

as a pseudo- random sequence to generate the dynamical association.

Let Y_0 be a predetermined value from the interval I used as initial condition for (5). By dropping the first N_0 iterations of (5) we can get its corresponding chaotic orbit:

$$Y_{N0+1}, Y_{N0+2}, \ldots\ldots, Y_{N0+N} \tag{6}$$

with the same length as the sequence S, equal to the number of sub regions R_i.

The sequence (6) is then rearranged in a decreasing order to obtain a new sequence:

$$Y'_1, Y'_2, \ldots\ldots, Y'_N \tag{7}$$

Such that $Y'_j=Y_{N0+i}$ if Y_{N0+i} is located in the j^{th} position after sorting. The sequence S is then rearranged using the sequence (7) and new associations are created like the following:

$$S_{Y'1}\rightarrow R_1,\ S_{Y'2}\rightarrow R_2,\ldots\ldots\ldots,S_{Y'N}\rightarrow R_N \quad (8)$$

This process is repeated after each Δ iteration of the map (1). To ensure randomness, the initial value of (5) is changed each time the sequence is generated. In this work, we choose to set the initial value Y^K_0 after each $K*\Delta$ iteration to:

$$Y^K_0=\text{Fract}(Y_0+X_{K*\Delta}) \quad (9)$$

where,

Y_0 is a predetermined value from I;
$X_{K*\Delta}$ is the last obtained value of the map (1);
Fract(x): give the fractional part of a real number x.

So the initial condition of the map (5) will take the values $Y^1_0,Y^2_0,\ldots,Y^p_0$, respectively when *p* depends on the keystream length *l* and the Δ parameter (*p* is the number of times the association between the sequence elements S_i and the regions R_i is recreated).

The parameter Δ greatly influences the resulting keystream. Accordingly, we propose to change it dynamically during the iterations instead of fixing its value. We actually find that this enhances the key stream randomness and sensibility to initial parameters.

Let use a third chaotic map $\Gamma_Z:I\rightarrow I$ with corresponding differences equation:

$$Z_{m+1}=\Gamma_Z(Z_m) \quad (10)$$

Using predetermined initial value Z_0, the generated orbit $\{Z_m,m=1,2,\ldots\ldots\}$ will serve to produce different values of Δ using the formula :

$$\Delta_i=\text{floor}\ (Z_{N0+i+1} * 10^{\alpha}) \quad (11)$$

When: Δ_i is the number of iterations performed before changing the association;
Z_{N0+i} is the i^{th} value obtained by equation (9) after dropping the first N_0 iterations.
The exponent α is a parameter that depends on the size of the generated stream, and determines the frequency of dynamic association generation. To make good compromise between the execution time and the efficacy of the generated stream randomness, we choose to set this exponent to:

$$\alpha = \text{Floor}(\log_{10}(\text{stream\_size}))-2 \quad (12)$$

Equation (12) has been determined experimentally and proved to give most appropriate result.
Using the maps and the parameters presented above, a keystream of length *l* can be generated as follows:

- Iterate N_0 times the maps (1) and (10) for a given values X_0 and Z_0;

- Set Δ_0 =floor(Z_{N0+1}*10^4) and start iterating (5) from Y^1_0 computed using (9) to generate N-value orbit;
- Rearrange the sequence S using the Γ_Y produced orbit and create the association with regions $\{R_i, 1 \leq i \leq N\}$;
- Iterate (1) for Δ_0 time and produce a key stream element at each iteration using the association and the Γ_X produced orbit :

$$k_i = S_j \text{ such that } X_i \in R_j \tag{13}$$

- Compute the new Δ_1 using (11), Y^1_0 using (9) ;
- Repeat steps 3 and 4 until we get the desired stream length.

The block diagram of the proposed algorithm is illustrated in Fig. 1 below. This algorithm can be used by choosing any combinations of these chaotic maps, Γ_X, Γ_Y and Γ_Z that verify the mixing property and sensibility to initial conditions. Furthermore, different values of n can be used to produce keystreams with different scales (e.g. binary if n=2). In our experiments, to represent Γ_X, Γ_Y and Γ_Z, we choose the logistic map defined by:

$$\Gamma(x) = \mu.x.(1-x) \tag{14}$$

This map has chaotic behavior [13] in the interval I = [0, 1] when $\mu \in [3.57,4]$. We use $\mu=4$ $X_{min}=0$ and $X_{max}=1$ in the implemented approaches.

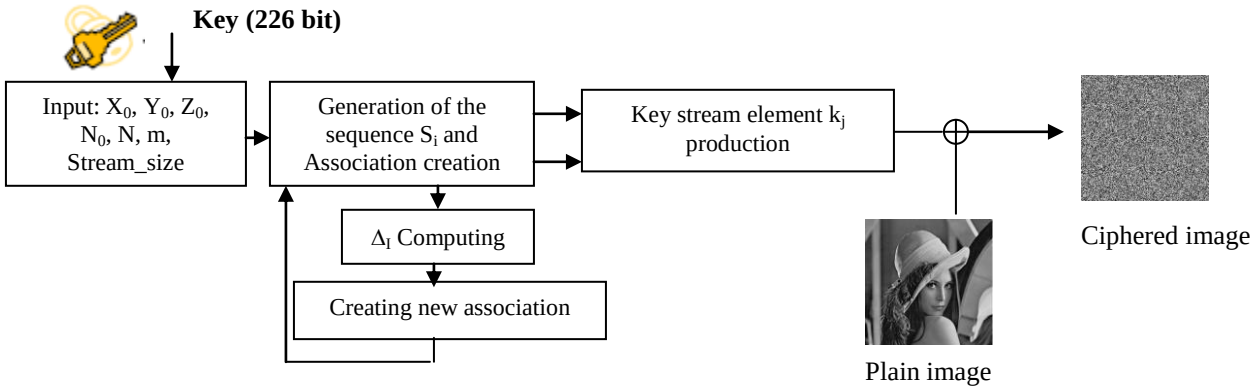


Fig 1. The block diagram of the proposed approach

Using the method explained above, we construct a simple encryption system using the generated key stream. The original color image of size M×M is first transformed into 3 matrixes R, G and B corresponding to the red plain, the green plain and the blue plain. We have three different matrixes which are transformed into a one dimensional single array P of length 3* M×M. A key stream of the same length is then generated using our approach. The array P of the image is combined with the corresponding element of the key stream as follows:

$$\begin{cases} C_{-1} = K_d \\ C_i = (((p_i + k_i) \bmod 256) \oplus C_{i-1}) \oplus k_i \end{cases} \tag{16}$$

where, p_i is the P elements and k_i is the keystream. The c_i is the encrypted image as a one dimensional vector C. This combination is iterated for m times (m ciphering cycle), so the length of the keystream generated initially is 3*m*M*M. The value of m is derived from the encryption key, and the obtained vector is finally rearranged back into three planes to construct the encrypted image. The seed of the diffusion c_{-1} is obtained from the diffusion key K_d.

2.2. Key schema

The key directly used in the proposed encryption scheme is a vector of 7 parameters, the three real values X_0, Y_0, Z_0, the integer values K_d, N_0,N and m. Real values are coded on 64 bit to ensure a precision of 10^{-20}. We use 16 bits to code N_0, 8 bits for K_d, 6 bits for N and 4 bits for m. These lead to a key size of 226 bit, making the key space as large as 2^{226} possible combination. This is larger than the acknowledged most security AES standard.

3.0 KEY STREAM PROPERTIES ANALYSIS

Different experiments are performed to test the statistical properties of the key stream outputted, and its sensibility to initial conditions. We use the precision of 10^{-20} which is easily supported on today's personal computers.

3.1. Statistical distribution of the key stream

From the point of view of strict cryptography, chaotic sequences have to satisfy uniform distribution which is important to prevent any kind of statistical attack. To prove the pseudo-uniformity of the keystream, we use the chi-square test [24] on 1000 generated instances of the key stream with size 10^7, using random combinations of parameters. The chi-square test is applied using:

$$\chi^2_{test} = \sum_{i=1}^{k} \left(\frac{o_i - e_i}{e_i} \right) \tag{17}$$

where n is the number of levels in the output key stream, o_i, and e_i are the occurrence observed and the expected frequencies of each level respectively. When using a significance level of 0.05, we find that $\chi^2_{test} < \chi^2_{255,0.05}$, so the null hypothesis is not rejected and the distribution of the key stream is uniform.

Fig. 2 shows the histogram of a keystream with size 10^7, which is obtained using X_0=0.512, Y_0=0.145825, Z_0=0.36352, N_0=356911, K_d=143 and N=18.

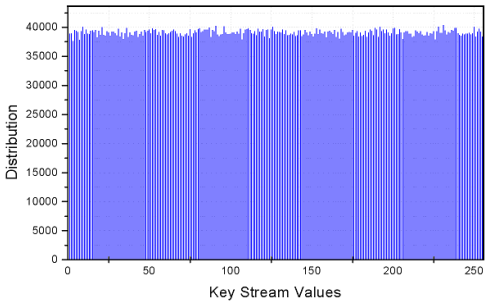


Fig 2. Histogram of the a generated key stream

3.2. The randomness nature of the keystream

To testify the randomness of the keystream, the statistic test suite designed by NIST [21] is applied. It contains an extensive set of statistical tests, evaluating three major categories of random natures: Random walk nature, Pattern checking and Complexity and compression. With the chosen standard parameters, a = 0.01 and $\alpha = 3$, we have P_a (probability of acceptance) = 97.28%. If the successive percentage of any test is smaller than P_a, the sequences are considered to be not good enough and the generated algorithm is not suitable for the usage. Table 12 shows the passing rate of the sequences. It can be observed that all the tests are passed, so the keystream properties are good enough to permit it's use for cryptographic application.

Table 1. Passing rate of the random sequences generated by the proposed approaches

Statistical Test	Rate of the test (%)
FT	98.38%
BFT	98.87%
CST forward sum	98.01%
CST backward sum	98.94%
RET	99.12%
REVT	98.99%
RT	98.52%
LROT	98.51%
NTMT	98.11%
OTMT	98.92%
MUST	99.01%
AET	97.82%
ST	98.01%
MRT	99.10%
SPT	98.45%
LCT	98.73%

3.3. Sensitivity to initial conditions

High key sensitivity is required by secure cryptosystems, which means that the cipher text cannot be decrypted correctly although there is only a slight difference between encryption or decryption keys. This guarantees to some extent the security of a cryptosystem against brute-force attacks.
In our case, the sensitivity is determined with respect to initial values of the different parameters. The cryptosystem will be enough secure to resist brute force attacks when sensitivity to initial parameters is increased. We use a measure of sensitivity analogous to that used in [25]. The change rate in the keystream K is computed by:

$$\begin{aligned} Cdr(p = p_0) &= \frac{Diff(K,K_1) + Diff(K,K_2)}{2 * Size(K)}.100\% \\ Diff(K,K_1) &= \sum_{i=1}^{size(K)} Difp(K[i], K_1[i]) \\ Difp(K[i], K_1[i]) &= \begin{cases} 1 & \text{if } K[i] = K_1[i] \\ 0 & \text{else} \end{cases} \end{aligned} \quad (20)$$

where K is the keystream generated when $p=p_0$, K_1 is the key stream generated when $p=p_0+\Delta p$, and K_2 is the keystream generated when $p=p_0-\Delta p$.

We studied the sensitivity of the keystream according to the parameters X_0, Y_0 and Z_0 with Δp set to 10^{-20}, and according to N_0, N with Δp set to 1. In all cases, the stream size was of 10^6 elements. At each time a value is assigned to a parameter, 300 random values are generated for other ones, and the resulting change rates are averaged.

Fig. 3 and Fig. 4 illustrate the change rate with respect to the variation of real valued and integer parameters respectively. We can see that the value of Cdr is above 97% for all possible values witch ensure high sensitivity to initial parameters.

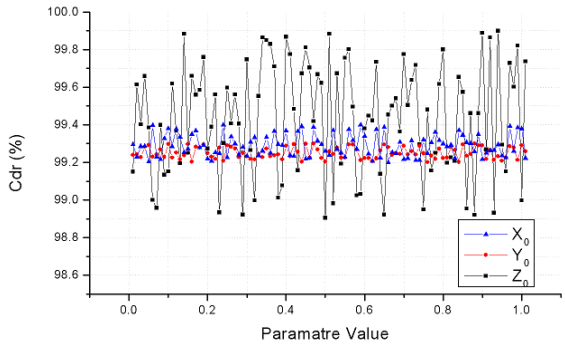
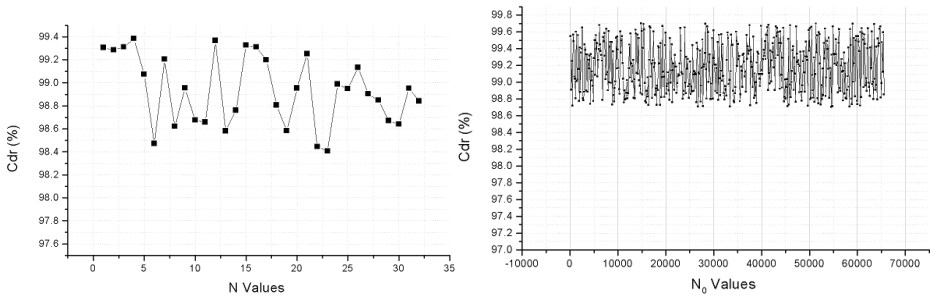


Fig 3. Key sensitivity test of the generated keystream with respect to x_0,y_0 and z_0 parameters

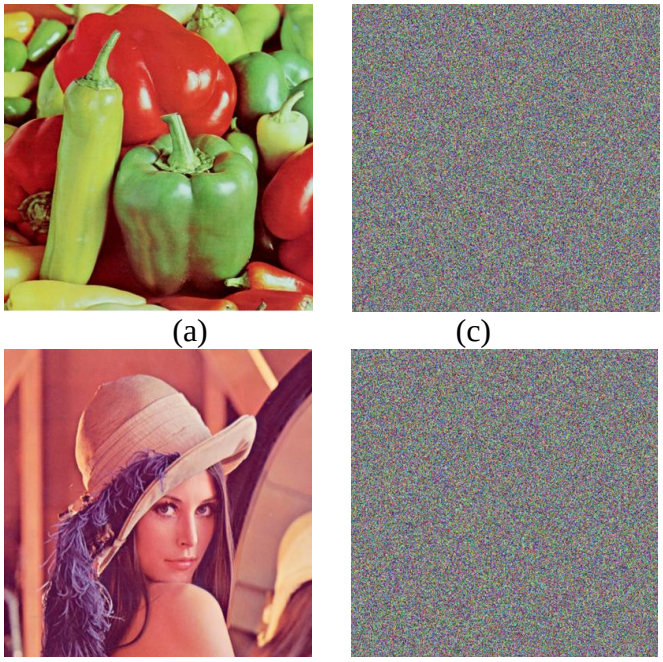


(a) (b)

Fig 4. Key sensitivity test of the generated keystream with respect to: (a) N and (b) N_0

4.0 CRYPTOSYSTEM SECURITY ANALYSIS

Encryption and decryption results are given in this section by demonstrating the efficiency of our proposed algorithm. We take two traditional 512×512 size color images of "Lenna" and "Peppers" as example. The pseudorandom keystream used for substitution is generated by following the scheme proposed. Original images and their corresponding ciphered ones are shown in Fig. 5. Initial parameters were: X_0= 0.567676469135816, Y_0= 0.60354653336181, Z_0= 0.339824617849325, N_0= 6067, N=11 K_d = 136 corresponding to the 226 bit key: "1395F87B4BCB9514041766CE33B5".



(a) (c)

(b) (d)

Fig 5. (a) Plain image "Pepper"; (b) Plain image "Lenna"; (c) Encrypted image "Pepper"; (d) Encrypted image "Lenna"

4.1. Key Sensitivity

Recall that secure cryptosystem requires not only a large key space but also a high key sensitivity. That is, a slight change in the key should cause some large changes in the cipher image. This property makes the cryptosystem of high security against statistical or differential attacks. Since the user key is in 226 bits, the key space is about $1.07 \cdot 10^{68}$, which is sufficient to resist the brute-force attack with the current computer technology.

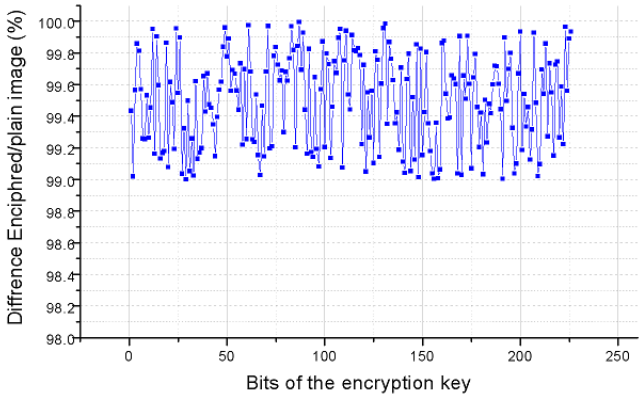


Fig 6. Differences between the corrected decrypted image and those decrypted with the 226 possible keys that differs in one bit from the correct key using Pepper image.

An encryption scheme has also to be key-sensitive, meaning that a small change in the key will cause a significant change in the output. Fig. 6 shows the values of differences between the correct decrypted image and the different decrypted images with the 226 possible keys that differ in one bit from the correct one. It is clear that the difference between the correct image and the one deciphered with the wrong key (one different bit) is above the 99% for each bit of the key. So the key is highly sensible to a very small variation, and that make the cryptosystem secure against plain text attacks and the brute force one.

4.2. Correlation of Adjacent Pixels

To test the correlation properties of the enciphered image, we performed statistical analysis on the encryption algorithm. This is done by testing the correlation between two vertically adjacent pixels, two horizontally adjacent pixels, and two diagonally adjacent pixels, respectively.
We randomly select 1000 pairs of two adjacent pixels from the image and calculate the correlation coefficient of each pair using the following discrete formulas [19]:

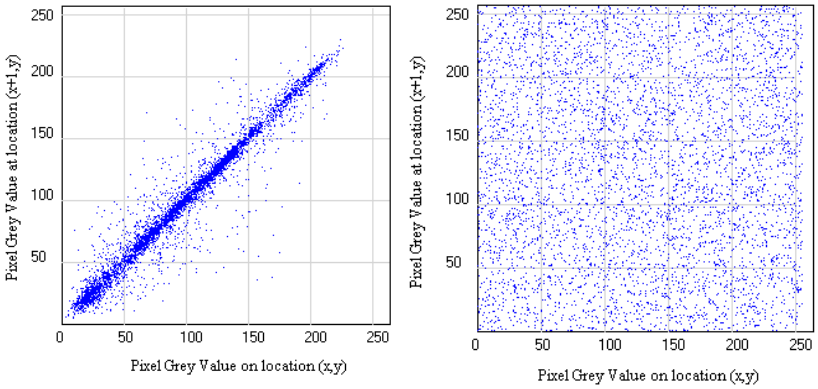
$$r_{xy} = \frac{Cov(x, y)}{\sqrt{D(x)}.\sqrt{D(y)}}$$

$$\text{where} \quad Cov(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{22}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \,, \; D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$$

Here, E(x) is the estimation of mathematical expectations of x, D(x) is the estimation of variance of x, and cov(x, y) is the estimation of covariance between x and y. x and y are grey-scale values of two adjacent pixels in the image.

Figure 7 shows the correlation distribution of two vertically adjacent pixels in the plain-image and those in the ciphered image. The average correlation coefficients are 0.98012569 and 0.00299584 respectively. Similar results for diagonal and vertical directions were obtained. These are shown in Table 2. It is clearly visible that correlation coefficients are very different when comparing the plain and the ciphered images.



(a) (b)

Fig 7. Correlation analysis of two horizontally adjacent pixels in (a) the plain Lena image; (b) the cipher image obtained using the proposed scheme.

Table 2. Correlation coefficients of adjacent pixels of plain image, ciphered image and a random one

	Plain image		**Ciphered image**		**Random image**
	Lena	**Pepper**	**Lena**	**Pepper**	
Horizontal	0.9501247	0.9483285	0.0039015	0.0024879	0.001562
Vertical	0.9801585	0.9758521	0.0024952	0.0021400	0.005922
Diagonal	0.9301023	0.9214752	0.0039785	0.0037585	0.004006

4.3. Resistance against differential attacks

It is clear that if one minor change in the plain image can cause a significant change in the ciphered-image, with respect to both diffusion and confusion, then a differential attack may become inefficient. To test the influence of one-pixel change on our whole encrypted image, two common measures are used: NPCR and UACI. The first one stands for the number of pixels change rate when one-pixel of plain image is changed, while the second one is the unified average changing intensity that measures the average intensity of differences between the plain and the ciphered image.

To avoid the known-plaintext attack and the chosen-plaintext attack, the changes in the cipher image should be significant even with a small change in the original one. According to the proposed encryption process, this small difference should be diffused to the whole ciphered data. They can in fact be reflected by the $NPCR_{R,G,B}$ [22] as shown in Fig. 8, since the images are 24 bit colored. The results are very close to the random case.

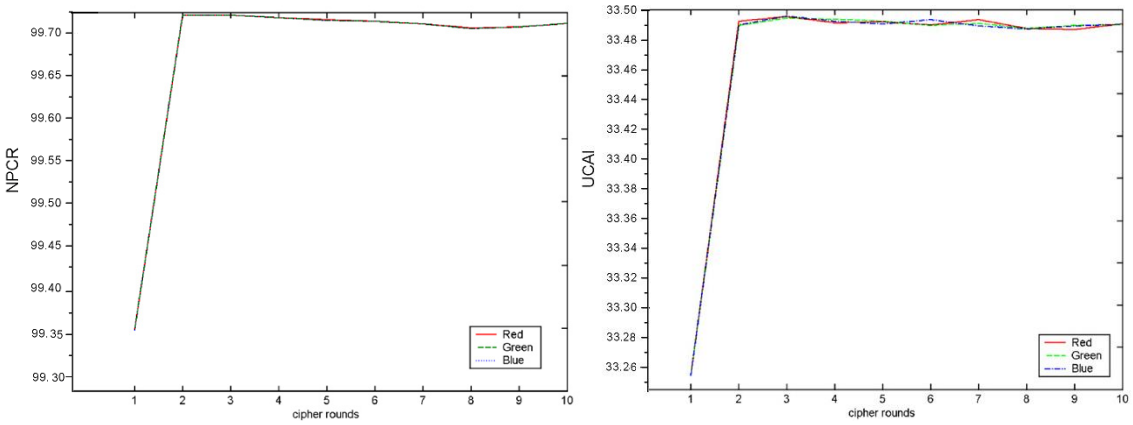


Fig 8. Progress of NPCR and UACI with respect to the iterations m

4.4. Computational complexity analysis

As compared to the traditional block ciphers [19] such as DES, IDEA and NSSU, the proposed chaos-based cryptosystem has some distinct properties. The tests are performed on the encryption speed of the proposed chaotic cryptosystem. To guarantee the maximum of the execution speed, we used Intel Assembly instruction in most of the implementations. The speed of the proposed encryption algorithm is much faster than most of the existing encryptions. Its average speed is about 40 MByte/s with Pentium IV 3GHz personal computer. Table 3 shows the result of the encryption speed and the speed of some well-known encryption algorithms in Crypto++ Library [20] using the same machine. With this speed, this image encryption scheme can be used in the Internet applications over the broadband network, where the encryption and decryption time have to be short compared to the transmission time.

Table 3. Encryption speed of the proposed scheme and some well-known algorithms

Algorithm	**Speed (MB/s)**
Proposed Approach	41.785
DES	6.212
AES (192-bit key)	11.472
AES (256-bit key)	10.972

5.0 CONCLUSION

In this paper, we proposed a cryptographic system for color image ciphering encryption using multiple chaotic maps combination. The system is in a stream-cipher architecture, where the pseudo-random key stream generator is constructed using three chaotic maps, serving the purpose of stream generation and random mixing, respectively. It is found that the proposed method assure more enhancement of the randomness nature of the key stream and its sensitivity to initial conditions variation even under finite precision implementation. Our approach is applicable to almost all chaotic maps with mixing property and that the achievement of the key streams with the desired long cycle length is almost easy. A statistical analysis on both stream generation system and the encryption scheme is given. From the experimental results, we can conclude that this algorithm outperforms some of the existing schemes, both in term of speed and security. Having a high throughput, the proposed system is ready to be applied in any fast real time encryption applications and suitable for practical use in the secure transmission of confidential information over the Web.

REFERENCES

1. Schneier B. Applied Cryptography - Protocols, Algorithms, and Source Code in C. John Wiley & Sons, Inc., New York, Second Ed., 1996.
2. Daemen J, Sand B, Rijmen V. The Design of Rijndael: AES - The Advanced Encryption Standard. Springer-Verlag, Berlin, 2002
3. Kocarev, L., Jakimoski, G. Stojanovski, T., and Parlitz U.: From chaotic maps to encryption schemes. In Proc. IEEE Int. Symposium Circuits and Systems, 4 (1998) pp.514–517.

4. Chong Fu, Zhen-chuan Zhang, Ying Chen, and Xing-wei Wang. "An Improved Chaos-Based Image Encryption Scheme". Y. Shi et al. (Eds.): ICCS 2007, Part I, LNCS 4487. pp. 575 – 582, © Springer-Verlag Berlin Heidelberg (2007).
5. H.S. Kwok, Wallace K.S. Tang. "A fast image encryption system based on chaotic maps with finite precision representation". Chaos, Solitons and Fractals 32 (2007) pp.1518–1529
6. Forré, R."The Hénon attractor as a keystream generator". In Advances in Cryptology-EuroCrypt'91, LNCS, Berlin Springer-Verlag. 0547 (1991) 76–81.
7. Li,S., Mou, X., Cai, Y."Pseudo-Random Bit Generator Based on Couple Chaotic Systems and its Applications in Stream-Cipher Cryptography". INDOCRYPT'2001, LNCS, Springer-Verlag,Berlin, 2247 (2001) 316–329.
8. S. Behnia a, A. Akhshani a, S. Ahadpour b,c, H. Mahmodi a, A. Akhavand. "A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps". Physics Letters A 366 (2007) pp.391–396.
9. Xiping He, Qingsheng Zhu, and Ping Gu. "A New Chaos-Based Encryption Method for Color Image". G. Wang et al. (Eds.): RSKT 2006, LNAI 4062, pp. 671–678. Springer-Verlag Berlin Heidelberg, (2006).
10. N.K. Pareek, Vinod Patidar , K.K. Sud . "Image encryption using chaotic logistic map". Image and Vision Computing 24 (2006), pp.926–934.
11. Kotulski Z, Szczepanski J. "On constructive approach to chaotic pseudorandom number generator". Proc Regional Conference on Military Communication and Information Systems, CIS Solutions for an Enlarged NATO, RCMIS2000, Zegrze (2000) pp.191 – 203.
12. Kelber, K., Götz, M., Schwarz, W.: Generation of chaotic signals with n-dimensional uniform probability distribution by digital filter structure. Proc. Of the 7th IEEE Digital Signal Processing Workshop (DSPWS'96) Norway: Loen, September,(1996)486–489.
13. Marcel Ausloos and Michel Dirickx. "The Logistic Map and the Route to Chaos: From the Beginnings to Modern Applications (Understanding Complex Systems)". ISBN: 3540283668. Publisher: Springer, First edition (February, 2006)
14. Kwok-Wo Wong, Bernie Sin-Hung Kwok, Wing-Shing Law. "A fast image encryption scheme based on chaotic standard map". Physics Letters A 372 (2008) pp.2645–2652.
15. David Williams. "Weighing the Odds: a Course in Probability and Statistics". Cambridge University Press, 2001, p.548. (ISBN 052180356X).
16. Kolmogorov-Smirnov Test Table: http://www.eridlc.com/onlinetextbook/appendix/table7.htm. Accessed 17/11/09.
17. A.D. Santis, A.L. Ferrara, B. Masucci, Discrete Appl. Math. 154 (2006) p.234.
18. C. E. Shannon, "Communication theory of secrecy systems" Bell Syst. Tech. J., vol. 28, pp. 656-715, 1949.
19. Vanstone SA, Menezes AJ, Oorschot PC. Handbook of applied cryptography. London: CRC Press; 1996.
20. Crypto++ Library, http://www.cryptopp.com. Accessed 17/11/09.
21. NIST. NIST Special Publication 800-22, http://csrc.nist.gov/rng/rng2.html. Accessed 17/11/09.
22. Chen G, Mao YB, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons & Fractals 2004;12:749 61.
23. Falcioni M, Palatella L, Pigolotti S, Vulpiani A 2006 Properties Making a Chaotic System a Good Pseudo Random Number Generator, ePrint arXiv:nlin.CD/0503035

24. Papoulis, A.: Probability, Random Variables, and Stochastic Processes. McGraw-Hill, New York (1965)
25. Shiguo Lian, JinshengSun and Zhiquan Wang. "Security analysis of a chaos-based image encryption algorithm». Physica A 351 (2005). pp. 645–661.