A Secure and Fast Chaotic Encryption Algorithm Using the True Accuracy of the Computer

Jean De Dieu Nkapkop and Joseph Yves Effa
University of Ngaoundéré, Department of Physics, P.O. Box 454, Ngaoundéré, Cameroon
E-mail: golby01@yahoo.fr, effa_jo@yahoo.fr

Monica Borda
Technical University of Cluj-Napoca, Department of Communications,
26-28 Baritiu Street, 400027, Cluj-Napoca, Romania
E-mail: Monica.Borda@com.utcluj.ro

Laurent Bitjoka
University of Ngaoundéré, Department of Electrical Engineering, Energetics and Automatics,
P.O. Box 454, Ngaoundéré, Cameroon
E-mail: lbitjoka@univ-ndere.cm

Alidou Mohamadou
University of Maroua, Department of Physics, P.O. Box 814, Maroua, Cameroon
E-mail: mohdoufr@yahoo.fr

Keywords: secure encryption, fast cryptosystem, chaotic sequences, permutation-diffusion scheme

Received: January 8, 2016

A secure and fast cryptosystem for image encryption based on chaotic generators is proposed. The principle of the method is to use the permutation-diffusion scheme to create computationally secure encryption primitives using the true accuracy of the computer. In the permutation step, integer sequences obtained by the sorting of the solutions of chaotic Logistic map by descending order is used as the permutation key to shuffle the whole image. This stage substantially reduces the correlation between neighbouring pixels. After, in order to increase the entropy of encrypted image, the iteration of the chaotic Skew Tent map is applied, with an exclusive-or scheme, to change the value of the entire pixel. Moreover, to further enhance the security of the cryptosystem, the keystream used in diffusion process is updated for each pixel and the computed encrypted pixel values depends on both the previously encrypted pixels and the random keystream. We proved that the cipher sequence of the algorithm is random and truly random by applying the NIST tests batteries and $\chi 2$-test respectively. Hence, the proposed algorithm can resist the statistical attacks. The extensive cryptanalysis has also been performed and results of our analysis indicate that the scheme is satisfactory in term of the superior security and high speed as compared to the existing algorithms, which makes it a very good candidate for real-time of multimedia data encryption applications.

Povzetek: Prispevek opiše nov način kriptiranja slik s pomočjo kaotičnih generatorjev.

1 Introduction

In today's world, the extension of multimedia technology in which image covers the highest percentage, has promoted digital images to play a more significant role than the traditional texts. The Internet banking, e-business, e-commerce, etc., are the major fields where security is most important. So it is necessary to encrypt image data before transmission over the network to preserve its security and prevent unauthorized access. For this end, most of the conventional encryption algorithms such as Advanced Encryption Standard (AES) [1] are designed with good properties [2, 3]. However, due to bulk volume of data, high correlation among adjacent pixels, high redundancy and real time requirement [4], these ciphers may not be the most desired candidates for image encryption, particularly for fast and real-time communication applications [5]. To meet this challenge, the chaos-based encryption has suggested a new and efficient way to deal with the intractable problem of fast and highly secure image encryption [6]. The properties of chaos such as high sensitive dependence on initial conditions and control parameter, quasi-randomness, ergodicity, unpredictability, mixing, etc. [7], which are analogous to the confusion and diffusion properties of Shannon [8], have granted chaotic dynamics as a promising alternative for the traditional cryptographic algorithms, and also for generating keystream.

Depending on the type of key used in the encryption algorithms, chaos-based cryptosystems are either symmetric or asymmetric. Symmetric encryption, in which the decryption key is identical to the encryption

key, is the oldest method in cryptology and is still used today. By contrast, asymmetric cryptosystems use different keys for decryption and encryption. We consider here typical (symmetric encryption) chaos-based image encryption techniques which rely on two processes: pixel permutation and pixel substitution [9, 10]. The first one, also call pixel confusion is needed to scramble the pixels. But, due to the strong correlation between adjacent pixels of the images, this stage does not guarantee a good level of security [11]. The diffusion stage is thus used to modify the pixel values in order to increase the entropy of the entire image.

Several image encryption algorithms based on this structure are already available in the literature [10, 12, 13, 14]. Each of them has its own strength and limitations more or less in terms of security level and computational speed. Accordingly, some of them have been cryptanalyzed successfully [15, 16, 17, 18, 19]. The common characteristic of these algorithms are: their chaotic generators need to be discretized to the finite sets of integers and that is time consuming and destroyed also their chaotic behaviors. Also, the keystream in the diffusion stage of these algorithms depends on the key only and that is less secured because an attacker can obtain that keystream by known/chosen plaintext attack [16]. So, to enhance the security, in [20], the keystream in the diffusion step depend on both the key and original image. Another method to obtain a high immunity to resist the differential cryptanalysis is to design strong substitution Boxes (S-Boxes) based on chaotic map or strong diffusion properties based on the combination of chaotic function and other techniques [21, 22, 23].

To improve the computational performance and to resist statistical, differential, brute-force attacks, this paper continues the same pursuit with further improvement, in which a one round chaos-based image encryption scheme based on the fast generation of large permutation key with a good level of randomness and a very high sensitivity on the keys is proposed. We use the integer sequences obtained by the descending sorting of the Logistic map as a secret key in the permutation stage. This technique avoids the excess digitization of chaotic values. As consequence, the sensitivity to small changes of the initial condition or control parameters is increased, as the true accuracy of the computer is exploited by using integer sequences. In diffusion process, at first, a random code is generated to get integer numbers from real numbers generated by Skew Tent map. Then, with that numbers, the exclusive-or is performed on the permuted image to computed the cipher image. The proposed approach can be easily implemented and is computationally simple.

The remaining of the paper is organized as follows. The chaotic maps are described in Section 2. In Section 3, the proposed encryption scheme is discussed in detail. Simulation results and security analysis are presented in Section 4 to show the efficacy and validity of the algorithm. Finally, conclusions are drawn in the last Section.

2 Chaotic maps

Chaotic maps are nonlinear maps that exhibit chaotic behavior. The chaotic maps generate pseudo-random sequences, which are used during encryption process [24]. Many fundamental concepts in chaos theory, such as mixing and sensitivity to initial conditions and parameters, actually coincide with those in cryptography [25]. The only difference in this concern is that encryption operations are defined on finite sets of integers while chaos is defined on real numbers. The main advantage using chaos lies in the observation that a chaotic signal looks like noise for the unauthorized users. Moreover, generating chaotic values is often of low cost with simple iterations, which makes it suitable for the construction of stream ciphers. Therefore, cryptosystem can provide a secure and fast means for data encryption, which is crucial for data transmission in many applications. The proposed scheme uses Logistic and Skew Tent maps and they are both discussed hereafter.

2.1 Logistic map

The Logistic map is a very simple non-linear dynamical and polynomial equation of degree two with x output and input variable, one initial condition x_0 and one control parameter λ and can be described as follows:

$$x_{n+1} = \lambda x_n (1 - x_n) \tag{1}$$

Where $x_n \in (0,\ 1)$ is the state of the system, for $n = 0,1,2,...$, and $\lambda \in (0,\ 4)$ is the control parameter. For different values of parameter λ, the Logistic sequence shows different characteristics [26]. For $3,58 \le \lambda \le 4$, the Logistic map Eqn. (1) has a positive Lyapunov exponent and thus is always chaotic. So all the (x_0, λ) where $x_0 \in (0,\ 1)$ and $3,58 \le \lambda \le 4$ can be used as secret keys.

2.2 Skew tent map

The Skew Tent chaotic map [27] can be described as follows:

$$y_{n+1} = \begin{cases} y_n/\alpha, & \text{if } y_n \in [0,\alpha] \\ (1\text{-}y_n)/(1\text{-}\alpha), & \text{if } y_n \in [\alpha,1] \end{cases} \tag{2}$$

Where α is controllable parameter for chaotic maps, y_i and y_{i+1} are the i-th and the $i+1$-th state of chaotic maps. For $\alpha \prec 1$ the system converges to 0 for all initial conditions. If $\alpha = 1$, then all initial conditions less than or equal to 0.5 are fixed points of the system, otherwise for initial conditions $y_0 \succ 0.5$ they converge to the fixed point $1 - y_0$. So all the y_0, $\alpha \in (0,\ 1)$, can be used as secret keys.

3 Proposed encryption scheme

The encryption algorithm consists of two stages: permutation and diffusion of pixels of the entire image as shown in Figure 1.

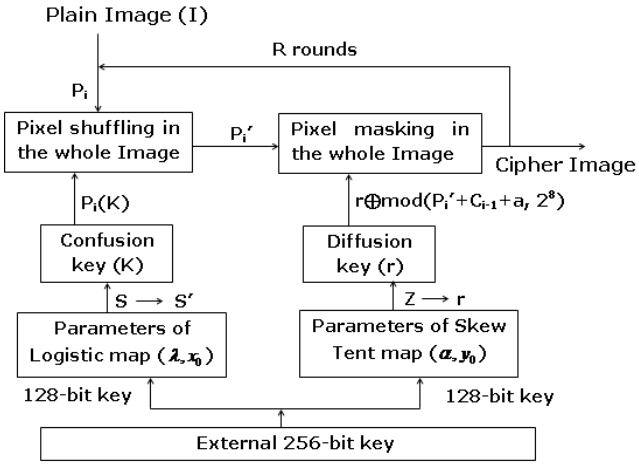

Figure 1: Synoptic of the proposed scheme.

In the proposed algorithm, we use one round (R=1) of confusion and diffusion for encryption.

3.1 Confusion

In this stage, the position of the pixels is scrambled over the entire image without change their values and the image becomes unrecognizable. The purpose of confusion is to reduce the high correlation between adjacent pixels in the plain image. To enhance the degree of randomness and the level of security, the Logistic map described in subsection 2.1 is used in order to generate pseudorandom key stream $S=\{x_1, x_2, ..., x_{M\times N}\}$ as the same size of the plain-image. Let I be a gray original image of size $M\times N$, containing M rows and N columns, and the gray values ranges from 0 to 255. Transform I to a one-dimensional vector $P=\{P_1, P_2, ..., P_{M\times N}\}$, where P_i is the i-th pixel value. Then sort S by descending order, and note $S'=\{x_j, ..., x_8, ..., x_1\}$ with $x_1 \prec ... \prec x_8 \prec ... \prec x_j$, the sorted chaotic values. The positions of sorted chaotic values in the original chaotic sequence are found and stored in $K=\{j, ..., 8, ..., 1\}$. Now, the next step is to scramble the total one-dimensional vector with K by using the following formula:

$$P' = P(K) \tag{3}$$

Where P' is the permuted image and K the permutation key. The reconstruction of P cannot be made unless the distribution of K is determined. The inverse transform for deciphering is given by:

$$P'(K) = P \tag{4}$$

This technique avoids the excess digitization of chaotic values. As consequence, the sensitivity to small changes of the initial condition or control parameters is increased, as the true accuracy of the computer is exploited and the computational time necessary for the generation of large permutation is reduced.

After obtaining the shuffled image, the correlation among the adjacent pixels is completely disturbed and the image is completely unrecognizable. Unfortunately, the histogram of the shuffled image is the same as that of the plain-image. Therefore, the shuffled image is weak against statistical attack and known plain-text attack. As a remedy, we design diffusion next to improve the security.

3.2 Diffusion

The total image is again encrypted with different chaotic numbers. Skew Tent map system shown in section 2.2 is applied here to produce that numbers: $Z=\{y_1, y_2, ..., y_{M\times N}\}$. The masking process is employed to modify the gray values of the image pixels, to confuse the relationship between the plain image and the encrypted image in order to increase the entropy of the plain image by making its histogram uniform. The diffusion function is also used to ensure the plain image sensitivity so that, a very little change in any one pixel of plain image should spread out to almost all pixels in the whole image. Diffusion is performed by using following equation:

$$C_i = r \oplus \mathrm{mod}\left(P_i' + C_{i-1} + a, 2^8\right) \tag{5}$$

Where C_i and C_{i-1} are the value of the currently and previously masking pixel respectively; C_0 can be set as a constant; P_i' is the permuted pixels; $\oplus$ is bitwise XOR operation; a is a positive integer and r is a random code obtained according to the following formula:

$$r = \mathrm{mod}\left(floor\left(y_n \times 2^{20}\right), 256\right) \tag{6}$$

Where, *mod* (x, y) returns the remainder after division and y_n is the state value corresponding to the n-th iteration of the skew tent map from initial state value y_0 and α.

A random code r is computed to get integer numbers from real numbers generated by Skew Tent map.

The key formula in decryption procedure is as follows:

$$P_i' = \mathrm{mod}\left(r \oplus C_i - C_{i-1} - a, 2^8\right) \tag{7}$$

To compute the first encrypted pixel, equation 8 is used.

$$C_1 = r \oplus \mathrm{mod}\left(P_1' + C_0 + a, 2^8\right) \tag{8}$$

Where r is evaluated by using Skew Tent map parameters below for i=1 to generate y.

$$\begin{cases} y_0 = \left(C_{i-1} + a\right) / \left(255 + a + b\right) \\ \alpha = \left(P_i' + a\right) / \left(M \times N + a + b\right) \end{cases} \tag{9}$$

With $a, b, C_0 \succ 0$.

For the security to be strengthened, the keystream r is updated for each pixel and the computed encrypted pixel values C_i depends on the previously encrypted pixels and the keystream, hence algorithm shows resistance to the differential attacks such as known plaintext attack, chosen-plaintext attack, known cipher-text attack and so on.

3.3 Encryption scheme

3.3.1 Encryption algorithm

The encryption algorithm is composed of thirteen steps.
Step 1: Reshape the plaintext image I into 1-D signal P and choose x_0 and λ in (0, 1) and (3.58, 4) respectively;
Step 2: Iterate the Logistic map given in equation 1 for T times to get rid of transient effect, where T is a constant;
Step 3: Continue to iterate the Logistic map for $M\times N$ times, and take out the state $S=\{x_{1+T}, x_{2+T}..., x_{M\times N+T}\}$;
Step 4: Sort S and get S' then, generate the permutation keys K as explained in subsection 3.1;
Step 5: Shuffle the pixels of the whole 1-D signal P with K using equation 3 and get P';
Step 6: Give C_0, choose a, b and evaluate y_0 and α in (0, 1) respectively as shown in equation 9;
Step 7: Iterate the Skew Tent map T times by using equation 2 and get the random code r;
Step 8: Compute the first cipher-pixel C_1 using equation 8 for i=1;
Step 9: Set i=i+1 and update y_0 and α in (0, 1) and get a new chaotic sequence y_T;
Step 10: Evaluate the new random code r by using equation 6;
Step 11: Compute cipher-pixel C_i according to the formula 5;
Step 12: Repeat step 9 to 11 until i reaches $M\times N$, the length of the whole 1-D signal;
Step 13: Reshape the 1-D signal into the 2-D image and get ciphered image.

The decryption involves reconstructing gray levels of the original image from the encrypted image. It is a simple inverse process of the proposed encryption algorithm.

3.3.2 Key schedule

A key of 128 bits or 256 bits is required for symmetric-key cryptosystems for more security [28]. We used an external 256-bit key ($E_1E_2 \cdot \cdot \cdot E_i \cdot \cdot \cdot E_{32}$, where E_i are ASCII symbols) to derive initial conditions and control parameters of the chaotic systems. The key is divided into two blocks of 16 ASCII symbols for the determination of the system control parameter and the initial condition respectively. For each block of 128 bits (corresponding to 16 ASCII symbols), we defined:

$$W=\sum_{i=0}^{15} 2^{\frac{i}{i+1}} P_i \tag{10}$$

where P_i are values (0-255) of ASCII symbols E_i and W is the value from which the control parameters and initial conditions are deduced, depending on the chaotic system. By considering the possible maximum value of ASCII symbols equal to 255 and the upper limit of the weight coefficient $2^{\frac{i}{i+1}}$ equal to 2, the value of W presents an upper limit Wr = 8160, which is used for its normalization.

The flowchart of the proposed encryption algorithm is then described in Figure 2.

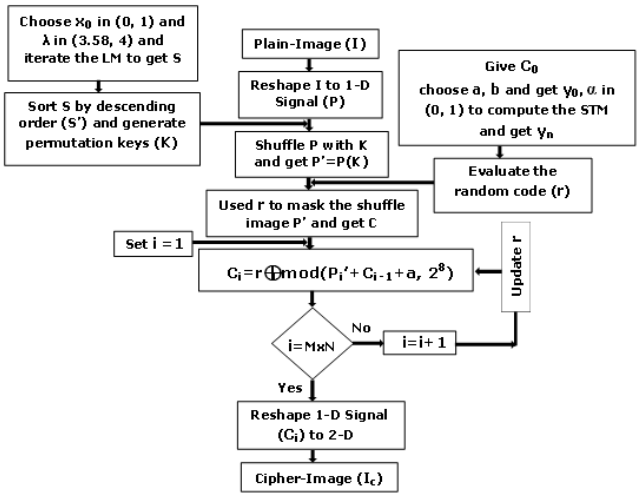

Figure 2: Flowchart of the encryption algorithm.

4 Experiments and security analysis

In this section, the proposed image cryptosystem is analyzed using different security measures. These measures consist of statistical analysis, sensibility analysis, differential attack analysis and speed analysis. Each of these measures which are widely used in the literature in the field of chaos-based cryptography [4, 10, 22, 23, 25, 29, 30, 31] is described in detail in the following subsections.

4.1 Statistical analysis

4.1.1 Randomness test

The National Institute of Standards and Technology (NIST) of the U.S. Government designed a set of fifteen tests to evaluate and quantify the randomness of binary sequences produced by either software or hardware based random or pseudo-random number generators for cryptographic applications [32]. The NIST has adopted two approaches: the examination of the proportion of sequences that pass a statistical test and the distribution of P-values to check for uniformity.

In our experiment, we used m = 2000 different keystreams, each sequence having a length of n = 1000,000 bits which are generated using our scheme. The acceptance region of the passing ratio is given by equation (11), where m represents the number of samples tested (m=2000) and P the probability corresponding to the significance level 0.01 (P = 0.99). In this case, we obtained the confidence interval [0.983, 0.996]. We have summarized the results obtained after applying the NIST test suite on the binary sequences produced by the proposed pseudo-random bit generator in the second and fourth column of Table 1. The proportion for each test computed on the Lena and Baboon encrypted images lies inside the confidence interval. So, the tested binary sequences generated by the proposed chaotic generator are random with respect to all tests of NIST suite with a confidence of 99%. In order to show that binary sequences tested are truly random, the χ2-test is used

[32]. According to that test, the P-values must be greater than 0.0001 to ensure that they could be considered uniformly distributed. The results from the third and fifth column of Table 1 lead us to the conclusion that P-values, for each statistical test, are well uniformly distributed.

These results show the quality of the produced sequences with the pseudo-random number generator. So the proposed map has perfect cryptographic properties.

$$\left[p-3\sqrt{(p(1-p))/m} \ , \ p+3\sqrt{(p(1-p))/m,} \right] \quad (11)$$

4.1.2 Visual test

In this subsection, we perform visual test using Lena and

Test name	Lena		Baboon		Result
	Passing ratio of the test	Uniformity P-value	Passing ratio of the test	Uniformity P-value	
Frequency	0.9907	0.053181	0.9902	0.216399	Good
Block frequency	0.9921	0.214936	0.9862	0.197340	Good
Cumulative sums	0.9886	0.183142	0.9819	0.179402	Good
Runs	0.9895	0.492088	0.9900	0.618276	Good
Longest run	0.9898	0.625312	0.9788	0.498233	Good
Rank	0.9876	0.429910	0.9876	0.429910	Good
FFT	0.9882	0.103644	0.9950	0.112419	Good
Non-overlapping template	0.9891	0.999925	0.9682	0.968566	Good
Overlapping template	0.9880	0.580201	0.9909	0.682005	Good
Universal	0.9869	0.273197	0.9898	0.293941	Good
Approximate entropy	0.9911	0.991018	0.9899	0.969684	Good
Random excursions	0.9934	0.544220	0.9897	0.510334	Good
Random excursions variant	0.9887	0.898591	0.9606	0.699372	Good
Serial	0.9901	0.682476	0.9953	0.708396	Good
Linear complexity	0.9918	0.733601	0.9868	0.599271	Good

Table 1: The statistic results of cipher images by 2010 revised version of NIST statistic test.

Black images of size 512 × 512 encrypted using parameters x_0=0.75, λ=3.393695629843 for the permutation and a=7, b=4 and C_0=23 for the diffusion. As shown in Figure 3(*b*) and (*b*'), the encrypted image is non recognizable in appearance, unintelligible, incomprehensible, random and noise-like image without any leakage of the original information. This demonstrates that the proposed algorithm can be used to protect various images for diverse protections. The decrypted images are exactly same as the original images (Figure 3(*c*) and (*c*')).

4.1.3 Histogram analysis

The histogram of the plain-image and cipher image is in Figure 3(*e*) and (*f*) respectively. We found that the histogram of the ciphered image has approximately a uniform distribution. For instance, the histogram in Figure 3(*f*') which corresponds to the ciphered Black image highlights the effectiveness of the algorithm, as all the 256 gray-levels present the same probability. To confirm this result, we measured the entropy for the ciphered image and we found that it has the value 7.9996 which is close to the ideal value 8.

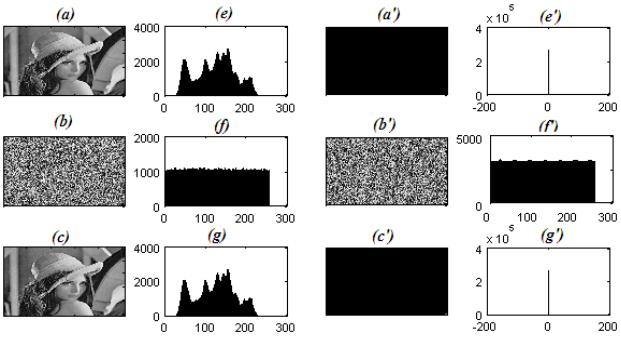

Figure 3: Histogram. (a), (a') original image; (b), (b') ciphered image of (a), (a'); (c), (c') decrypted image of (b), (b'); (e), (e'), (f), (f'), (g), (g') histogram of (a), (a'), (b), (b'), (c), (c') respectively.

4.1.4 Key space analysis

The key space is the total number of different keys that can be used in the encryption/decryption procedure. For an effective cryptosystem, the key space should be sufficiently large enough to resist brute-force attacks. In the proposed algorithm a 256-bit key corresponding to 32 ASCII symbols is considered and the key consists of the initial value x_0, a, b and the parameter λ, where $x_0 \in (0, 1)$, $\lambda \in (3.58, 4)$ et $a, b>0$. In hexadecimal representation, the number of different combinations of secret keys is equal to 2^{256}. Accordingly, the theoretical key space is not less than 2^{256}, which is large enough to resist brute-force attack [10].

4.1.5 Correlation analysis

The proposed chaotic encryption system should be resistant to statistical attacks. Correlation coefficients of adjacent pixels in the encrypted image should be as low as possible [28]. A thousand pairs of two adjacent pixels are selected randomly in vertical, horizontal, and diagonal direction from the original and encrypted images. And then, the correlation coefficient was computed using the formulas below and the results are shown in Table 2 and the visual testing of the correlation distribution of two horizontally adjacent pixels of the plain image and the cipher image produced by the proposed scheme is shown in Fig. 4. It is clear from the results of the sixth row of Table 2 and Fig. 4 that the proposed approach is resistant to statistical attacks. We can also find in Table 2 that the proposed encryption algorithm has much better statistic properties than those in [4, 22, 25, 29, 30, 31], using respectively the same standard gray scale image Lena with size 512x512.

$$r_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2\right)\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{y})^2\right)}} \quad (12)$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (13)$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \quad (14)$$

Where x_i and y_i are greyscale values of i-th pair of adjacent pixels, and N denotes the total numbers of samples.

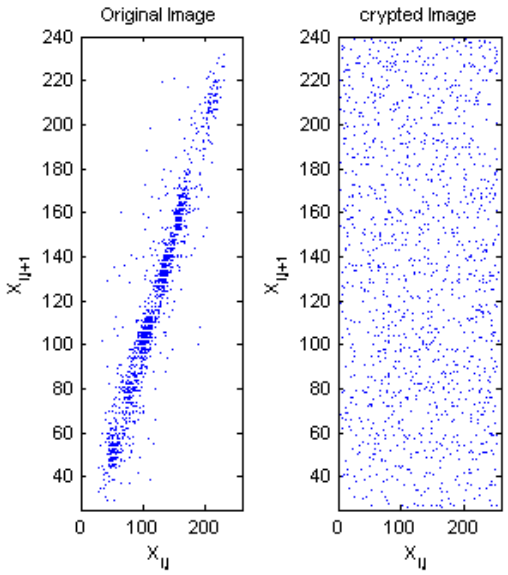

Figure 4: Correlation of horizontally adjacent pixels of the image Lena.

4.1.6 Information entropy

The information entropy can be calculated by:

$$H = -\sum_{i=1}^{2^M} p(m_i)\log_2\left(p(m_i)\right) \quad (15)$$

where M is the number of bits to represent a symbol; $p(m_i)$ represents the probability of occurrence of symbol mi and log denotes the base 2 logarithm so that the entropy is expressed in bits. It is known that if the information entropy is close to 8, the encryption algorithm is secure upon the entropy attack. The results of the third row of Table 2 show that, our scheme is better in the aspect of the information entropy than the other encryption schemes ([4, 22, 25, 29, 30, 31]).

4.2 Key sensitivity analysis

An ideal image encryption procedure should have not only a large key space but also a high key sensitivity. Key sensitivity implies that the small change in the secret key should produce entirely different encrypted image. It means that a slight change in the key should cause some large changes in the ciphered image [28]. This property makes the cryptosystem of high security against statistical or differential attacks. Fig. 5 shows key sensitivity test result. Where the plain Lena image is firstly encrypted using the test key (x_0=0.75, λ=3.393695629843, a=9, b=2). Then the ciphered image is tried to be decrypted using five decryption keys:

(i) x_0=0.75, λ=3.393695629843, a=9, b=2;
(ii) x_0=0.74, λ=3.393695629843, a=9, b=2;
(iii) x_0=0.75, λ=3.393695629842, a=9, b=2;
(iv) x_0=0.75, λ=3.393695629843, a=8, b=2;
(v) x_0=0.75, λ=3.393695629843, a=9, b=3.

It can be observed that the decryption with a slightly different key fails completely. Therefore, the proposed image encryption scheme is highly key sensitive.

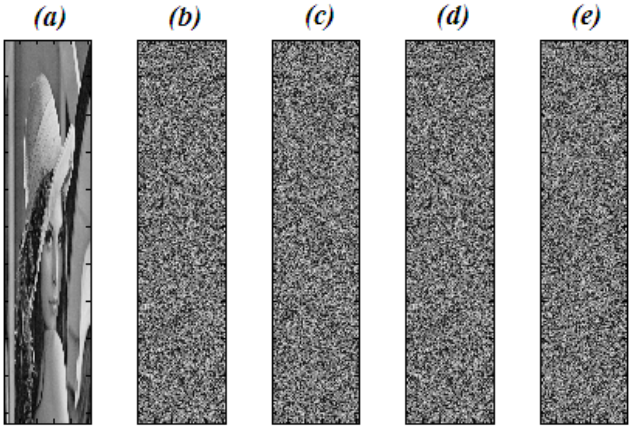

Figure 5: Key sensitivity test: (a) Deciphered image using key (i); (b) Deciphered image using key (ii); (c) Deciphered image using key (iii); (d) Deciphered image using key (iv); (e) Deciphered image using key (v).

4.3 Differential attack analysis

In general, a desirable property for an encrypted image must be sensitive to the small changes in plain-image. An opponent may make a slight change, usually one pixel, in the plain image and compare the cipher images (corresponding to very similar plain images and obtained by the same key) to find out some meaningful relationship between plain image and cipher image, which further facilitates in determining the secret key. If one minor change in the plain image can be effectively diffused to the whole ciphered image, then such differential analysis as known plaintext attack, chosen-plaintext attack, known cipher-text attack and so on, may become inefficient and practically useless.

The diffusion performance is commonly measured by means of two criteria, namely, the Number of Pixel Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). *NPCR* is used to measure the percentage of different pixel numbers between two images. The *NPCR* between two ciphered images A and B of size $M \times N$ is [28]:

$$NPCR_{AB} = \frac{\sum_{i=1}^{M}\sum_{j}^{N} D(i,j)}{M \times N} \times 100 \quad (16)$$

Where,

		Chaos-based encryption algorithm						
Reference		**Proposed**	**[22]**	**[29]**	**[4]**	**[25]**	**[30]**	**[31]**
Entropy		7.9996	7.9971	7,9993	7.9992	7.9994	7.9992	7.9880
NPCR		99.693	99.621	99,603	99.6201	99.639	-	-
UACI		33.621	33.434	33,456	33.4006	33.554	-	-
Correlation coefficients of permuted image	H	0.0002	0.0097	-0,0010	0.0026	0.000707	-0.0155	0.0368
	V	-0.0030	0.0136	-0,0016	0.0034	0.002165	0.0199	-0.0392
	D	-0.0008	0.0178	0,0010	-0.0019	0.014886	0.0244	0.0068
Number of rounds		1	2	10	1	2	-	2
Key space		2^{256}	2^{167}	2.27×10^{57}	2.27×10^{57}	-	2^{451}	2^{153}
Key sensitivity		99.69 percent difference	99.62 percent difference	99,628 percent difference	99.61 percent difference	99.622 percent difference	high	99.61 percent difference

Table 2: Comparison of different chaos-based encryption algorithm.

$$D(i,j)=\begin{cases}1 & A(i,j)\neq B(i,j)\\ 0 & otherwise\end{cases} \quad (17)$$

The expected *NPCR* for two random images with 256 gray levels is 99.609 %.

The second criterion, *UACI* is used to measure the average intensity of differences between the two images. It is defined as [27]:

$$UACI_{AB}=\frac{100}{M\times N}\sum_{1}^{M}\sum_{1}^{N}\frac{|A(i,j)-B(i,j)|}{255} \quad (18)$$

For a 256 gray levels image, the expected *UACI* value is 33.464 %.

The *NPCR* and *UACI* test results are shown in Table 2. The proposed cryptosystem achieves high performance by having $NPCR \succ 0.99609$ and $UACI \succ 0.33464$ and can well resist the known-plaintext, the chosen-plaintext and the known cipher-text attacks. Also, the results of the seventh row of table 2 show that the proposed scheme requires fewer permutation and diffusion rounds than the other algorithms. Indeed, the proposed scheme requires few chaotic numbers for the generation of complex permutation and diffusion keys, which contributes to the raise of the speed performance as compared to the other algorithms ([4, 22, 25, 29, 30, 31]).

4.4 Efficiency analysis

Running speed of the algorithm is an important aspect for a good encryption algorithm, particularly for the real-time internet applications. In general, encryption speed is highly dependent on the CPU/MPU structure, RAM size, Operating System platform, the programming language and also on the compiler options. So, it is senseless to compare the encryption speeds of two ciphers image. We evaluated the performance of encryption scheme by using Matlab 7.10.0. Although the algorithm was not optimized, performances measured on a 2.0 GHz Pentium Dual-Core with 3GB RAM running Windows XP are satisfactory.

The average running speed depends on the precision used for the quantization of chaotic values. For *P*=8, the average computational time required for 256 gray-scale images of size 512x512 is shorter than 100 *ms*. By comparing this result with those presented in Ref. [28], the scheme can be said high-speed as we only used a 2.0 GHz processor and the Matlab 7.10.0 software. Indeed, the modulus and the XOR functions are the most used basic operations in our algorithm. Also, the comparison between the simulations times required at the permutation stage shows that the computational time required in our experiment is three times less than that of Chong Fu et al's. [34].

This means that the actual computational times of our scheme could be at least smaller if implemented in the same conditions than the Chong Fu et al's. algorithm. So, referring to actual fast ciphers [22, 23, 34], our proposed algorithm has a fast running speed. Such a speed is promising for real time applications of multimedia data encryption.

5 Conclusion

In this paper, we proposed a new secure and fast chaos-based algorithm for image encryption using the true accuracy of the computer. In the proposed scheme, the permutation-diffusion design based on the fast generation of large permutation and diffusion key with a good level of randomness and a very high sensitivity has been investigated. This procedure allows to use the true accuracy of the computer by using integer sequences obtained by the descending sorting of the Logistic map as a secret key in the permutation stage. This technique avoids the excess digitization of chaotic values. As consequence, the sensitivity to small changes of the initial condition or control parameters is increased. In the diffusion stage, in order to avoid the known/chosen plaintext attack, we have proposed to link keystream with both the key and original image to mask the whole image. By using these techniques, the spreading process is significantly accelerated contrary to that of Chong Fu et *al*. [34]. According to NIST randomness tests, the image sequences encrypted by the proposed algorithm have no defect and pass all the statistical tests with high P-values. Also, we proved the very good cryptographic performances of the proposed image scheme through an

extensive analysis, performed with respect to the latest methodology from this field. As a result, one round of encryption with the proposed algorithm is safe enough to resist exhaustive attack, differential attack and statistical attack. The new scheme has higher security and faster enciphering/deciphering speeds. This makes it a very good candidate for real-time image encryption applications.

Acknowledgement

J.D.D Nkapkop gratefully acknowledges the Erasmus Mundus – Action 2 for their financial support.

References

[1] H. Dobbertin, V. Rijmen and A.Sowa (2005). Advanced Encryption Standard-AES, *4th International Conference*, AES 2004, Bonn, Germany, May 10-12, Lecture Notes in Computer Science 3373.

[2] F. Riaz, et al. (2012). Enhanced image encryption techniques using modified advanced encryption standard. *Communications in Computer and Information Science*. 281, 385-396.

[3] S. Dey (2012). SD-AEI: An advanced encryption technique for images. *In Proceedings of Digital Information Processing and Communications*, Second International Conference, Lithuania, 68-73.

[4] J. S. A. Eyebe Fouda, et al. (2014). A fast chaotic block cipher for image encryption. *Communications in nonlinear science and numerical simulations*. 19 (3), 578-588.

[5] S. Li, et al. (2007). On the design of perceptual MPEG-video encryption algorithms. *In: IEEE Transactions on Circuits and Systems for Video Technology*, 214-223.

[6] S. Mohammad, S. Mirzakuchaki (2012). A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal processing*. 92(5), 1202-1215.

[7] A. A. Abd El-Latif, et al. (2012). Digital image encryption scheme based on multiple chaotic systems. Sensing and Imaging, *An international journal on continuing subsurface sensing technologies and applications*. 56(2), 67-88.

[8] C. Shannon (1949). Communication theory of secrecy systems. *Bell System Technical Journal*. 28, 656-715.

[9] M. Demba and O. M. Abu Zaid (2013). A Proposed Confusion Algorithm Based on Chen's Chaotic System for Securing Colored Images. *International Journal of Signal Processing Systems*. 1(2), 296-301.

[10] J-Y. Wang, G. Chen (2015). Design of a Chaos-Based Digital Image Encryption Algorithm in Time Domain. *In: IEEE International Conference on Computational Intelligence and Communication Technology* (CICT), 26-28.

[11] S. Li, et al. (2008). A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Processing: Image Communication*. 23, 212-223.

[12] S. Sam, P. Devaraj and R. S. Bhuvaneswaran (2012). A novel image cipher based on a mixed transformed logistic maps. *Multimedia tools and applications*, 56 (2), 315-330.

[13] X. Di, L. Xiaofeng and W. Pengcheng (2009). Analysis and improvement of a chaos-based image encryption algorithm. *Chaos Solitons and Fractals*. 40 (5), 2191-2199.

[14] A. A. Abd El-Latif, L. Li and X. Niu (2014). A new image encryption scheme based on cyclic elliptic curve and chaotic system. *Multimedia tools and applications*. 70 (3), 1559-1584.

[15] P. Jagadeesh, P. Nagabhushan and R. Pradeep Kumar (2012). A New Image Scrambling Scheme through Chaotic Permutation and Geometric Grid based Noise Induction. *International Journal of Computer Application*. 78 (4), DOI: 10.5120/13481-1181.

[16] G. Alvarez, S. Li (2009). Cryptanalyzing a nonlinear chaotic algorithm (NCA) for image encryption. *Commun Nonlinear Sci Numer Simul*. 4 (11), 3743-3749.

[17] R. Rhouma, E. Solak and S. Belghith (2010). Cryptanalysis of a new substitution-diffusion based image cipher. *Commun Nonlinear Sci Numer Simul*. 15 (7), 1887-1892.

[18] C. Li, D. Arroyo and K. Lo (2010). Breaking a chaotic cryptographic scheme based on composition maps. *International Journal of Bifurcation and Chaos*. 20, 2561-2568.

[19] R. Rhouma and S. Belghith (2008). Cryptanalysis of a new image encryption algorithm based on hyper-chaos. *Physics Letters A*. 372 (38), 5973-5978.

[20] E. Solak, et al. (2010). Cryptanalysis of Fridrich's chaotic image encryption. *International Journal of Bifurcation and Chaos*. 20 (5), 1405-14013.

[21] C. Zhu, C. Liao and X. Deng (2013). Breaking and improving an image encryption scheme based on total shuffling scheme. *Nonlinear Dynanics*. 71(1), 25-34.

[22] R. Guesmi, et al. (2014). A novel design of Chaos based S-Boxes using genetic algorithm techniques. *In: IEEE 11th International Conference on Computer Systems and Applications*. 678-684.

[23] X. Wang and Q. Wang (2014). A Novel image encryption algorithm based on dynamic S-boxes constructed by chaos. *Nonlinear Dynanics*. 75 (3), 567-576.

[24] Y. Wang, et al. (2015). A colour image encryption algorithm using 4-pixel Feistel structure and multiple chaotic systems. *Nonlinear Dynamics*. 81 (1), 151-168.

[25] Y. Wang, et al. (2011). A new chaos-based fast image encryption algorithm. *Applied Soft Computing*. 1 (1), 514-522.

[26] A. Gonzalo, L. Shujun (2006). Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurcation Chaos*. 16, 21-29.

[27] Q. Zhang, L. Guo and X. Wei (2010). Image encryption using DNA addition combining with chaotic maps. *Mathematical and Computer Modelling*. 52 (11-12), 2028-2035.
[28] A. Rukhin, et al. (2010). A statistical test suite for the validation of random number generators and pseudo-random number generators for cryptographic applications," *NIST Special Revised Publication 800-22*, April 2010.
[29] J.S.A. Eyebe Fouda, et al. (2012). Efficient Cryptosystem Based on Chaotic Sequences Sorting. *American Journal of Signal Processing*. 2, 15-22.
[30] Z. Tang, J. Song, X. Zhang and R. Sun. Multiple-image encryption with bit-plane decomposition and chaotic maps. *Optics and Lasers in Engineering*.80, 1-11.
[31] C. Fu, et al. (2011). A novel chaos-based bit-level permutation scheme for digital image encryption. *Optics Communication*. 284, 5415-5423.
[32] L. Billings and E. Bollt (2001). Probability density functions of some skew tent maps. *Chaos, Solitons and Fractals*. 12 (2), 365-376.
[33] Y. Wang, et al. (2010). A new chaos-based fast image encryption algorithm. *Applied Soft Computing*. 11 (1), 514-522.
[34] C. Fu, et al. (2012). A chaos-based digital image encryption with an improved permutation strategy. *Optic Express*. 20 (3), 2363-2378.