Chaotic Encryption Scheme Based on A Fast Permutation and Diffusion Structure

Jean De Dieu Nkapkop[1,2], Joseph Effa[1], Monica Borda[2], Laurent Bitjoka[3], and Mohamadou Alidou[4]
[1]Department of Physics, University of Ngaoundéré, Cameroon
[2]Department of Communications, Technical University of Cluj-Napoca, Romania
[3]Department of Electrical Engineering, Energetics and Automatics, University of Ngaoundéré, Cameroon
[4]Department of Physics, University of Maroua, Cameroon

Abstract: *The image encryption architecture presented in this paper employs a novel permutation and diffusion strategy based on the sorting of chaotic solutions of the Linear Diophantine Equation (LDE) which aims to reduce the computational time observed in Chong's permutation structure. In this scheme, firstly, the sequence generated by the combination of Piecewise Linear Chaotic Map (PWLCM) with solutions of LDE is used as a permutation key to shuffle the sub-image. Secondly, the shuffled sub-image is masked by using diffusion scheme based on Chebyshev map. Finally, in order to improve the influence of the encrypted image to the statistical attack, the recombined image is again shuffled by using the same permutation strategy applied in the first step. The design of the proposed algorithm is simple and efficient, and based on three phases which provide the necessary properties for a secure image encryption algorithm. According to NIST randomness tests the image sequence encrypted by the proposed algorithm passes all the statistical tests with the high P-values. 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.*

Keywords: *Fast and secure encryption, chaotic sequence, Linear Diophantine Equation, NIST test.*

Received March 17, 2015; accepted October 7, 2015

Online Publication IAJIT First

1. Introduction

With the fast development of image transmission through computer networks, especially the Internet, the security of digital images has become a most important concern. Image encryption differs from text encryption due to some intrinsic features of images which include bulk data capacities, high redundancy, strong correlations among pixels, etc. [8]. These features make conventional cipher systems such as DES, AES and RSA unsuitable for practical image encryption [13].

In order to overcome image encryption problems, in recent years, many scientists and engineers have designed image encryption algorithms based on one or more chaotic maps [5, 11]. Due to desirable properties of nonlinear dynamical systems such as high sensitive dependence on initial conditions and control parameter, ergodicity, unpredictability, mixing, etc., which are analogous to the confusion and diffusion properties of Shannon [12], the chaos-based encryption has suggested a new and efficient way to deal with the intractable problem of fast and highly secure image encryption [11].

Fridrich [6] suggested that a suitable chaos-based image encryption algorithm should be composed of two phases: one phase is to permute the order of the image pixels using chaotic map(s) while the other phase is to alter the numerical values representing the color of each pixel, again using chaotic map(s). These two phases are referred to as the confusion phase and the diffusion phase and they form the basis of many existing chaos-based image encryption algorithms [2, 7]. Nevertheless, to assure an efficient encryption scheme, some conditions should be fulfilled such as a large key space, randomness of the cipher-image and a high sensitivity on the initial conditions. A large key space is necessary to resist brute-force attacks [9] and a secure encrypted image corresponds to an image that cannot be statistically distinguished from a truly random sequence. Indeed, the cipher-images should present a good level of randomness and moreover, should be very sensitive to the used of initial key or seeds and to the plain-image [1].

Some existing image encryption algorithms were designed with a fast diffusion strategy, but their permutation is not fast enough because at this stage the discretization of chaotic sequences in finite values is time consuming. To achieve high security level performance, they also need more than one round in their permutation-diffusion structure.

The key challenge now in cryptography being to consider the trade-offs between the security level and efficiency, in this paper, a chaotic cipher for gray images by a fast permutation-diffusion structure is proposed.

In the proposed scheme, image is split in n sub-images. The design of the proposed algorithm is then based on three phases. In phase I, image permutation is based on the sorting of the solutions of the Linear

Diophantine Equation (LDE) [4] whose coefficients are integers and dynamically generated from any type of chaotic systems to enhance the speed at the permutation stage. This method also leads to a stronger permutation effect. In phase II, we generate diffusion template using image diffusion based on Chebyshev map. Then the image is masked by performing XOR operation on the shuffled image and diffusion template. Finally, in phase III, the recombined image is encrypted by using permutation key based on the sorting of the solutions of the LDE. To avoid the cyclic digitization of chaotic numbers in the generation of permutation key and then achieve high speed performance, we generate permutation key at the initialization step by using ascending or descending sorting of the solution of LDE; thereafter, this permutation key is just dynamically updated for each sub-image by including inside $d > 3$ chaotic numbers and then sorting the result for obtaining and updated permutation key. Also, diffusion key is updated for each sub-image.

The rest of the paper is organized in the following manner: comparison between two image permutations is described in section 2. The diffusion phase in the proposed cryptosystem is presented in section 3. The simulation and performance analysis are discussed in section 4 and the conclusions are made in section 5.

2. Comparison Between Two Image Permutations

In order to decorrelate the strong relationship between adjacent pixels, the permutation process is usually used.

2.1. Image Permutation Based On Chirikov Standard Map

To demonstrate the superiority of the proposed image permutation, we recall here image permutation based on Chirikov standard map. In order to incorporate Chirikov standard map into image encryption that operated on a finite set, it has to be discretized. The discretized version Chirikov standard map can be obtained by changing the range of (x, y) from the square $[0, 2\pi)\times[0, 2\pi)$ to the discrete lattice $N\times N$ as follows [3]:

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

where N is the width or length of a square image, and K is a positive integer which can be used as the permutation key.

Chirikov standard map is then employed to shuffle the pixel positions of the plain image. It application to a grayscale test image with 512×512 size is shown in Figure 1. The ciphering key being K=512.

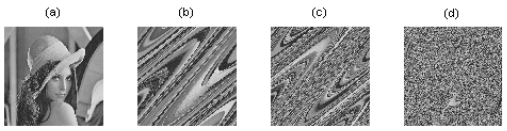

Figure 1. Permutation based on Chirikov Standard Map: (*a*) The plain image of Lena with 512×512 size. (*b*) The test image after applying the Chirikov standard map once. (*c*) The test image after applying the Chirikov standard map three times. (*d*) The test image after applying the Chirikov standard map five times.

However, to get the correlation among the adjacent pixels completely disturbed and the image completely unrecognizable this permutation needs five rounds of iterations. We then propose at the following sub-section to replace this kind of permutation by a one-round permutation based on the sorting of chaotic solutions of the Linear Diophantine Equation (LDE) for permuting the pixel position in the image.

2.2. Image permutation based on the sorting of chaotic solutions of the LDE

The technique used at the permutation stage is based on the ascending or descending sorting of the chaotic solutions of the LDE. The LDE is defined by Equation 2 below [4]:

$$ax + by = e \quad (2)$$

where a, b and e are the constants and belongs to the set of natural integers, x and y the general solutions.

In order to determine the solutions of LDE as a random process, coefficients a and b of the LDE can be evaluated from the chaotic system in the Algorithm 1 below:

Algorithm 1: Fragment to get a and b of the LDE

1. Require: T_1, T_2, W_r, λ_0
2. Initialisation: $W_{\lambda 0} \leftarrow 0$; $Wx_0 \leftarrow 0$;
3. for k=0 to 15 do
$W_\lambda \leftarrow 2^{(k/(k+1))} \times T_1(k+1) + W_{\lambda 0}$
$Wx \leftarrow 2^{(k/(k+1))} \times T_2(k+1) + Wx_0$
end for
4. $\lambda \leftarrow \lambda_0 + W_\lambda / (10Wr)$
5. $x_0 \leftarrow Wx / Wr$
6. Require: x_{01}, x_{02} from chaotic system
7. $a \leftarrow 2 \times fix(2^p x_{01}) + 1$
8. $b \leftarrow 2 \times fix(2^p x_{02}) + 1$

The control parameter λ and the initial condition x_0 are deduced from the keys T_1 and T_2. λ_0 is a constant such that the behavior of Equation (3) remains chaotic for all the range of λ and x. Wr=8160 is the greatest value of $W_{\lambda 0}$. $T_1(k+1)$ and $T_2(k+1)$ respectively correspond to the values assigned to the ASCII symbols of key T_1 and T_2. *fix(.)* is the integer part of function. p is the number of bits used for the quantization. The precision used for the digitization of the chaotic values by using Matlab is about $\varepsilon \approx 10^{-15}$.

In this work, we have considered as chaotic system, the Piece Wise Linear Chaotic Map (PWLCM) described by the following equation:

IAJIT First Online Publication

$$x(n)=F[x(n-1)]=\begin{cases} x(n-1)\times\frac{1}{\lambda}, & if\ \ 0\le x(n-1)\prec\lambda \\ [x(n-1)-\lambda]\times\frac{1}{0.5-\lambda}, & if\ \ \lambda\le x(n-1)\prec 0.5 \\ F[1-x(n-1)], & if\ \ 0.5\le x(n-1)\prec 1 \end{cases} \quad (3)$$

The PWLCM is known to be chaotic when its control parameter λ is within]0, 0.5[and its initial condition is chosen within the interval]0, 1[[4].

Then the coefficients *a* and *b* of the LDE evaluated in the *Algorithm Fragment 1* above are used to calculate the solutions of LDE which is sorted to generate the permutation key for encryption. The procedure is shown in the Algorithm 2 below:

Algorithm 2: Fragment to get permutation key I_Z

1. *Require: N, t*
2. *[G, C, D] ← gcd(a,b)*
3. *x ← mod(C+(b/G)t, N) + 1*
4. *y ← mod(D - (a/G)t, x) + 1*
5. *[I, J] ← sort(x)*
6. *[I_1, J_1] ← sort(y)*
7. *I_z ← J(J_1)*

Where *t*=(0, 1,…,*N*-1), *N* is the length of image. *gcd(x,y)* is the greatest common divisor, *mod(x,y)* is the modulo and *sort(x)* array elements in ascending or descending order. I_Z is the permutation key.

For the security to be strengthened, it is necessary for the permutation key I_Z to be updated for each sub-images. The updating process is carried out by replacing *d*>3 number of values of I_Z with newly generated *d* number of chaotic numbers from the chaotic system and then sorting it for obtaining the updated I_Z.

In the proposed permutation scheme, the total image frame is divided into *n* sub-images. For the first sub-image the encryption is carried out by using *Algorithm Fragment 2* and for the other sub-images, the permutation key is only refreshed without solving LDE equations. This helps to save computational time and at the same time the length of permutation key is large enough to attain high security level.

The application of this method to a grayscale test image with 512×512 size is shown in Figure 2.

Figure 2. Permutation based on the sorting of chaotic solutions of the LDE: (*a*) The plain image of Lena with 512×512 size. (*b*) The test image after applying the proposed permutation once.

As can be seen in this Figure 2, this method leads to a stronger permutation effect as compared to the previous case and need only one round of permutation.

3. Diffusion Phase in the Proposed Cryptosystem

In diffusion stage, the pixel values are modified sequentially to confuse the relationship between cipher image and plain image in order to increase the entropy of the plain image by making its histogram uniform. In this paper, Chebychev map is used to generate keystream *K(n)* in order to mask the pixel in first time. The Chebychev map is described by:

$$x(n+1)=T_k(x_n)=\cos\left(k.\cos^{-1}x_n\right),\quad x_n\in[-1,1] \quad (4)$$

Where $k\in[2, +\infty[$ is control parameter. The initial value *x*(0) and parameter *k* are used as the key.
The diffusion procedure is described as follows:

1. Randomly select a parameter *k* and an initial value *x*(0) for equation 4. Iterate equation 4 *t* times to avoid the harmful effect of the initial values, where *t* is a preset integer and served as secret encryption key, too.
2. Let *I* denotes a gray scale permute image with size *N*×*M*. Reshape *I* and get *P*={*p*(1), *p*(2),…*p*(*n*),… *p*(*N*×*M*)}.
3. Obtain for each iteration one key stream element from the current state of the chaotic according to:

$$K(n)=\mathrm{mod}\left[\ floor\left(((x(n)+1)/2)\times10^{14}\right),L\right] \quad (5)$$

where floor(*x*) returns the value of *x* to the nearest integers less than or equal to *x*, *mod*(*x*, *y*) returns the remainder after division and *L* is the gray levels of plain-image
4. Then apply the bitwise exclusive-OR to the permuted image pixels *P* by the following equation:

$$c(n)=\frac{1}{2}\left(K(n)\oplus\{[p(n)+K(n)]\bmod N\}\oplus c(n-1)\right) \quad (6)$$

Where *p*(*n*), *K*(*n*), *c*(*n*) are the currently operated pixel, key stream element and output cipher-pixel, respectively, and *c*(*n*-1) is the previous cipher-pixel.
5. Alter the control parameter *k* of the Chebychev map in each round iteration as follows:
if *x(n)>0*, then

$$k\leftarrow k+10^{-14}\times c(n\text{-}1) \quad (7)$$

else

$$k\leftarrow k-10^{-14}\times c(n\text{-}1) \quad (8)$$

6. In order to more secure the diffusion process, continue to apply the bitwise exclusive-OR by using the sequence *c*(*n*) of Equation 6 and diffusion key *X* as follows:

$$X\leftarrow mod(bx+ay,256) \quad (9)$$

$$c(n)\leftarrow c(n)\ \oplus X \quad (10)$$

where *x* and *y* are obtained using Algorithm 2, a and b are obtained using Algorithm 1.

IAJIT First Online Publication

Finally, in order to increase the randomness of the entire 1-D ciphered image obtained in the Equation 10 as well as the sensitivity of the cipher to small changes in the plain image, an image dependent initial condition is determined as follows:

Initialisation: $\lambda_0 \leftarrow 0$
for j=1 to $N \times M$ do

$$\lambda_0 \leftarrow \lambda_0 + p(j)/(255 \times L) \quad (11)$$

end for

The control parameter is the same as one used for the generation of I_Z. L is equal to the length of the whole image.

This initial condition is used for the generation of two chaotic integers which are used as coefficients a and b of the LDE to generate the permutation key I_{Zt}. This new permutation key of length L is thus deduced from the LDE for shuffling the 1-D image as a whole.

Notice that, before permuting the image as a whole, the n sub-images used in the permutation and diffusion stages are recombined to obtain a 1-D image.

The decryption procedure is followed in a reversed order of the encryption procedure. The flowchart of the proposed encryption and decryption algorithm is then described in Figure 3.

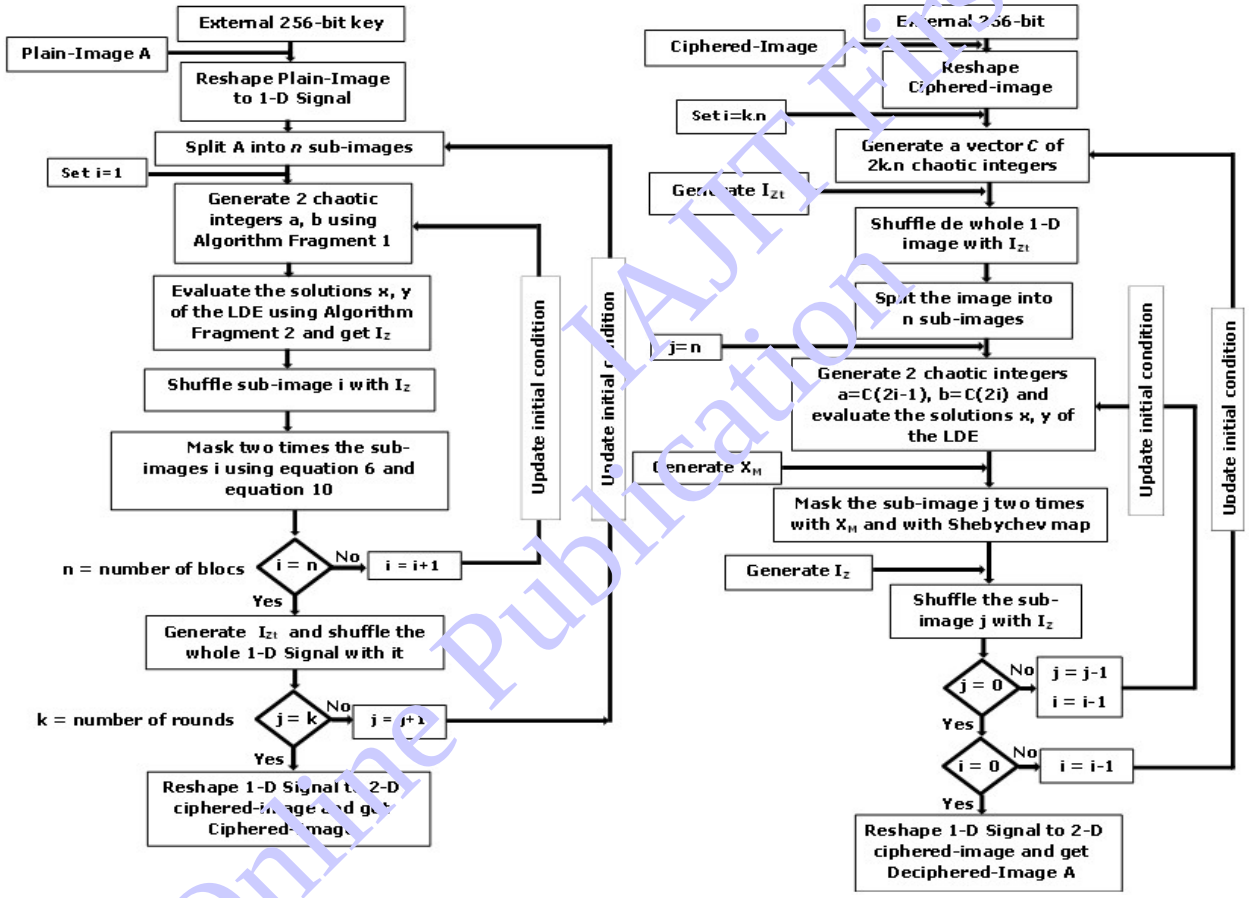

Figure 3. Flowchart of the encryption and decryption algorithm.

4. Simulations and Performance Analysis

4.1. Key Space

The key space is the total number of different keys that can be used in the encryption/decryption procedure. The key of the proposed cryptosystem is composed of two parts: permutation key T_1, T_2 and diffusion key (x_0, k). In our work a 256-bit key corresponding to 32 ASCII symbols is considered.

In hexadecimal representation, the number of different combinations of secret keys is equal to 2^{256}. Therefore, the total number of possible values of x_0 that can be used as a part of the key is approximately 2×10^{32}. The range of k should be restricted to a particular interval of 2π to prevent Chebyshev map from producing periodic orbits, then for k there will be approximately $2\pi\times10^{15}$ different possible values.

By considering only symbols “a-z”, “A-Z” and “0-9”, the complete key space of the proposed image encryption scheme is $62^{32}\times4\pi\times10^{47}\approx2^{347}$ which is large enough to resist brute-force attack.

4.2. Histogram

Image histogram clarifies how the pixel’s values of image are distributed. The histograms of the plain-image and the cipher-image are shown in Figure 4. As

IAJIT First Online Publication

shown in this figure, it is obvious that the histograms of the encrypted image are nearly uniform and significantly different from the histograms of the plain-image. Hence it does not provide any clue to be employed in a statistical analysis attack on the encrypted image. For instance, the histogram in Figure 4(*f*) which corresponds to the ciphered Black image highlights the effectiveness of the algorithm, as all the 256 gray-levels present the same probability.

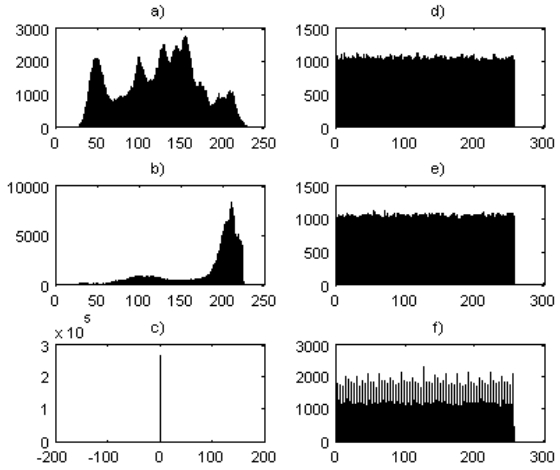

Figure 4. Example of histograms: (*a*)-(*c*) plain-images; (*d*)-(*f*) ciphered images. From top to bottom are presented histograms of Lena 512×512, Airplane 512×512 and Black 512×512.

4.3. Randomness Test

The US NIST designed a set of 15 statistical tests to test randomness of binary sequences produced by pseudorandom number generators.

For each test, a *P*-value is computed from binary sequence. In all tests, if the computed *P*-value is < 0.01, then conclude that the sequence is non-random. Otherwise, conclude that the sequence is random.

In our experiment, m = 2000 different keystreams, each sequence having a length of n = 1000,000 bits which are generated using our scheme. The *P*-values for various tests are listed in Table 1. In this case, we obtained the confidence interval [0.983, 0.996]. We notice that the results of the tests are satisfactory for the whole set of tested outputs. All the sequences pass successfully the NIST tests. These results show the quality of the produced sequences with the pseudo-random number generator.

Table 1. The results of the NIST tests.

Test name	Passing ratio of the test	Uniformity P-value	Result
Frequency	0.9875	0.006355	PASSED
Block frequency	0.9895	0.047478	PASSED
Cumulative sums	0.9870	0.170922	PASSED
Runs	0.9890	0.339271	PASSED
Longest run	0.9895	0.616305	PASSED
Rank	0.9880	0.583145	PASSED
FFT	0.9860	0.096000	PASSED
Non-overlapping template	0.9895	0.999668	PASSED
Overlapping template	0.9870	0.412733	PASSED
Universal	0.9875	0.383827	PASSED
Approximate entropy	0.9905	0.893482	PASSED
Random excursions	0.9922	0.430809	PASSED
Random excursions variant	0.9875	0.892512	PASSED
Serial	0.9895	0.595549	PASSED
Linear complexity	0.9905	0.551365	PASSED

4.4. Correlations Between Adjacent Pixels

First, randomly select 1000 pairs of two adjacent pixels from an image. Then, calculate their correlation coefficient using the following formulas:

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

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

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

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

The results of the correlation coefficients for horizontal, vertical and diagonal adjacent pixels for the plain images and its cipher images are given in Table 2 and 3. Table 2 shows that the correlations coefficients evaluated with the proposed algorithm are better than those presented in the others references.

Table 2. Comparative study of the correlation coefficients of the proposed algorithm with some existing algorithm.

Image		Type	Proposed algorithm	[12]	[2]
Lena	Horizontal	Plain-image	0.9719	0.9404	0.9792
		Cipher-image	-0.0005	0.0088	0.0217
	Vertical	Plain-image	0.9850	0.9299	0.9809
		Cipher-image	-0.0032	-0.0087	0.0086
	Diagonal	Plain-image	0.9593	0.9257	0.9551
		Cipher-image	-0.0002	-0.0060	0.0118

4.5. Information Entropy Analysis

In information theory, entropy is the most significant feature of disorder, or more precisely unpredictability.

IAJIT First Online Publication

Table 3. Correlation coefficients of two adjacent pixels in the others images.

Image	Size	Type	Horizontal	Vertical	Diagonal
Airplane	512×512	Plain-image	0.9663	0.9642	0.9370
		Cipher-image	-0.0018	-0.0035	0.0035
Black	512×512	Plain-image	1	1	1
		Cipher-image	0.0019	0.0012	0.0000

It is well known that the entropy $H(m)$ of a message source m can be measured by:

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

where M is the number of bits to represent a symbol; $p(m_i)$ represents the probability of occurrence of symbol m_i and *log* denotes the base 2 logarithm so that the entropy is expressed in bits.

For a purely random source emitting 2^8 symbols, the entropy is $H(m) = 8$ bits. The test result on different images for one round is defined in Table 4. It appears that the entropy of the ciphered images is almost equal to eight, compared to that of the plain-images. It can also be noticed that the encrypted version of the image "Black" is a truly random image, thus confirming the efficiency of the proposed cipher.

Table 4. Information entropy of plain-images and cipher-images by the proposed algorithm.

Image	Type	Lena	Airplane	Black
Entropy	Plain-image	7.4456	6.7043	0
	Cipher-image	7.9992	7.9993	7.9978

4.6. Differential attack Analysis

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 [14]:

$$NPCR_{AB} = \frac{\sum_{i=1}^{n}\sum_{j}^{n} D(i,j)}{m\times n}\times 100 \tag{16}$$

where

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

The second criterion, UACI is used to measure the average intensity of differences between the two images. It is defined as [13]:

$$UACI_{AB} = \frac{100}{m\times n}\sum_{1}^{m}\sum_{1}^{n}\frac{\left|A(i,j)-B(i,j)\right|}{255} \tag{18}$$

To evaluate the performance promotion of the proposed encryption scheme, the NPCR and UACI are plotted against the cipher cycles and compared with that of the existing scheme, as shown in Figures 5(*a*) and (*b*), respectively. As can be seen from Figure 5, four overall encryption rounds are needed to achieve a satisfactory security level by using Lian *et al.'s* encryption scheme [10], three overall encryption rounds are needed to achieve a satisfactory security level by using Zhu *et al*. [15]; one overall encryption round is needed to achieve a satisfactory security level by using Chong Fu *et al*. [3] and our algorithm. However, compared to Chong Fu *et al's* algorithm, our method improves their NPCR and UACI only with one round of permutation and diffusion process.

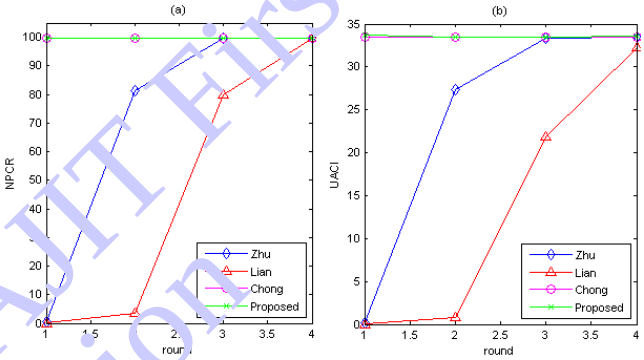

Figure 5. NPCR and UACI performance of the proposed scheme and the others existing scheme. (*a*) NPCR performance. (*b*) UACI performance.

4.7. Key Sensitivity Analysis

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 ciphered image. This property makes the cryptosystem of high security against statistical or differential attacks. To evaluate the key sensitivity, the plain Lena image is encrypted using four slightly different test keys:

Table 5. Slightly different keys for encryption.

(*i*)	x_0 = 0.48729650284971	k = 5.78259581295362	T_1 = azertyuiopqsdfgj	T_2 = azertyuiopqsdfg0
(*ii*)	x_0 = 0.48729650284970	k = 5.78259581295362	T_1 = azertyuiopqsdfgj	T_2 = azertyuiopqsdfg0
(*iii*)	x_0 = 0.48729650284971	k = 5.78259581295361	T_1 = azertyuiopqsdfgj	T_2 = azertyuiopqsdfg0
(*iv*)	x_0 = 0.48729650284971	k = 5.78259581295362	T_1 = azertyuiopqsdfg1	T_2 = azertyuiopqsdfg0
(*v*)	x_0 = 0.48729650284971	k = 5.78259581295362	T_1 = azertyuiopqsdfgj	T_2 = azertyuiopqsdfg2

The corresponding cipher images are shown in Figures 6(*a*), (*b*), (*d*), (*f*) and (*h*), respectively. The differences between any two cipher images are computed and given in Table 6. The differential images between (*a*) and (*b*), (*a*) and (*d*), (*a*) and (*f*) and (*a*) and (*h*) are shown in (*c*), (*e*), (*g*) and (*l*) of Figure 6 respectively.

IAJIT First Online Publication

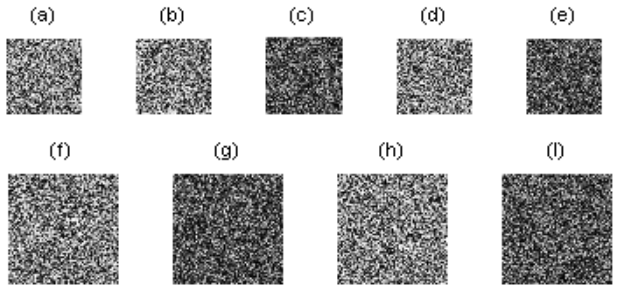

Figure 6. Key sensitivity test: (*a*) Ciphered image using key (*i*). (*b*) ciphered image using key (*ii*). (*c*) Differential image between (*a*) and (*b*). (*d*) Ciphered image using key (*iii*). (*e*) Differential image between (*a*) and (*d*). (*f*) Ciphered image using key (*iv*). (*g*) Differential image between (*a*) and (*f*). (*h*) Ciphered image using key (*v*). (*l*) Differential image between (*a*) and (*h*).

Table 6. Differences between cipher images produced by slightly different keys.

Figure	**Test keys**	**Difference (%)**				
		(*a*)	(*b*)	(*d*)	(*f*)	(*h*)
(*a*)	(*i*)	----	99.6059	99.6040	99.6014	99.6086
(*b*)	(*ii*)	99.6059	----	99.6132	99.6277	99.6346
(*d*)	(*iii*)	99.6040	99.6132	----	99.6239	99.6117
(*f*)	(*iv*)	99.6014	99.6277	99.6239	----	99.6197
(*h*)	(*v*)	99.6086	99.6346	99.6117	99.6197	----

Therefore, the proposed scheme is highly sensitive to the key.

4.8. Efficiency Analysis

Running speed of the algorithm is an important aspect for a good encryption algorithm, particularly for the real-time internet applications. 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 computational time required for 256 gray-scale images of size 512×512 is shorter than 110 *ms*. By comparing this result with those presented in [14], 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.

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*. [3]. 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.

5. Conclusions

In this paper, an encryption algorithm for the fast generation of large permutation and diffusion keys with a good level of randomness and a very high sensitivity on the key has been investigated. The permutation process was generated by sorting the chaotic solutions of the LDE whose coefficients are integers and dynamically generated from PWLCM. The proposed scheme thus requires few chaotic numbers for the generation of complex permutation and diffusion keys. By using this technique, the permutation step and the spreading process are significantly accelerated contrary to that of Chong Fu *et al*. [3]. As a result, one round of encryption with the proposed algorithm is safe enough to resist exhaustive attack, chosen plaintext attack and statistical attack. Simulations have been carried out to compare its performance with that of existing methods. We have also performed an exhaustive testing process of the randomness of the generated binary sequences using the NIST suite in order to prove the viability of the proposed method. This makes it a very good candidate for real-time image encryption applications.

Acknowledgement

J.B.D Tkapkop gratefully acknowledges the AUF for their financial support.

References

[1] Alvarez G., and Li S., "Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129-2151, 2006.

[2] Chen G., Mao Y., and Chui C., "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos Solitons & Fractals*, vol. 21, no. 3, pp. 749-761, 2004.

[3] Chong F., Chen J., Zou H., Meng W., and Zhan Y., "A chaos-based digital image encryption scheme with an improved diffusion strategy," *OPTICS EXPRESS*, vol. 20, no. 3, pp. 2363-2378, 2012.

[4] Eyebe A., Effa J., Samrat L., and Ali M., "A fast chaotic block cipher for image encryption," *Commun. Nonlinear Sci. Numer. Simulat.*, vol. 19, no. 3, pp. 578-588, 2014.

[5] Faraoun K., "A chaos-based key stream generator based on multiple maps combinations and its application to images ," *The International Arab Journal of Information Technology*, vol. 7, no. 3, pp. 231-1215, 2010.

[6] Fridrich J., "Symmetric ciphers based on two dimensional chaotic maps," *International Journal of Bifurcation and Chaos*, vol. 8, no. 6, pp. 1259-1284, 1998.

[7] Guan Z., Huang F., and Guan W., "Chaos-based image encryption algorithm," *Physics Letters A*, vol. 346, no. 1-3, pp. 153-157, 2005.

IAJIT First Online Publication

[8] Jastrzebski K., and Kotulski Z., "On improved image encryption scheme based on chaotic map lattices," *Engineering Transactions*, vol. 57, no. 2, pp. 69-84, 2009.
[9] Li C., Li S., Asim M., Nunez J., Alvarez G., and Chen G., "On the security defects of an image encryption scheme," *Image and Vision Computing*, vol. 27, no. 9, pp. 1371-1381, 2009.
[10] Lian S., Sun J., and Wang Z., "A block cipher based on a suitable use of the chaotic standard map," *Chaos, Solitons & Fractals*, vol. 26, no. 1, pp. 117-129, 2005.
[11] Mohammad S., and Mirzakuchaki S., "A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map," *Signal processing*, vol. 92, no. 5, pp. 1202-1215, 2012.
[12] Shannon C., "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656-715, 1949.
[13] Wang J., and Chen G., "Design of a chaos-based digital image encryption algorithm in time domain," *in IEEE International Conference on Computational Intelligence & Communication Technology*, pp. 26-29, 2015.
[14] Wang Y., Wong K., Liao X., Chen G., "A new chaos-based fast image encryption algorithm," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 514-522, 2011.
[15] Zhu Z., Zhang W., Wong K., and Yu H., "A chaos-based symmetric image encryption scheme using bit-level permutation," *Information Sciences*, vol. 181, no. 6, pp. 1171-1186, 2011.

Jean De Dieu Nkapkop received his Master's degree in Electronics, Electrical Engineering and Automatic, from Department of Physics of the University of Ngaoundere, Cameroon in 2012. Currently, he is preparing his Ph.D. degree in the field of chaos-based cryptography.

Joseph Effa received his Ph.D. in Electronics from the University of Yaounde I, Cameroon in 2009. His research interests include chaos detection, synchronization of chaotic systems, nonlinear transmission line and chaos-based image encryption.

Monica Borda received his Ph.D. in Telecommunication from the University of Bucharest, Romania in 1987. She is Professor in Electronics Engineering and Telecommunication at Communications Department from the Technical University of Cluj-Napoca, Romania. She has more than 40 years' experience of education and research in the topics included Information Theory and Coding, Cryptography and Genomic, Signal Processing and Image Processing.

Laurent Bitjoka received his Ph.D. in Biomedical Engineering from the University of Tours, France in 1994. His research interests include signal processing, image processing, automatic and biophysics.

Alidou Mohamadou received his Ph.D. in mechanics from the University of Yaounde I, Cameroon in 2007. His research interests include nonlinear dynamics and chaos, nonlinear dynamic of DNA, pattern formation, nonlinear transmission line, modulational instability and plasma physics.

IAJIT First Online Publication