A Fast Image Encryption Algorithm Based on Chaotic Maps and the Linear Diophantine Equation

Jean De Dieu Nkapkop¹, Joseph Yves Effa¹, Jean Sire Armand Eyeb E Fouda², Mohamadou Alidou³,⁴, Laurent Bitjoka⁵ and Monica Borda⁶

¹. Department of Physics, Faculty of Science, University of Ngaoundéré, Ngaoundéré 237, Cameroon
². Department of Physics, Faculty of Science, University of Yaoundé 1, Yaoundé 237, Cameroon
³. Department of Physics, Faculty of Science, University of Maroua, Maroua 237, Cameroon
⁴. Department of Physics, Faculty of Science, University of Douala, Douala 237, Cameroon
⁵. Department of Electrical, Energetics and Automatics, ENSAI, University of Ngaoundéré, Ngaoundéré 237, Cameroon
⁶. Department of Communications, Faculty of Electronics, Telecommunications and Information Technologies, Technical University of Cluj-Napoca, Cluj-Napoca 400027, Romania

Corresponding author: Joseph Yves Effa (effa_jo@yahoo.fr)

Abstract: In recent years, chaos-based encryption has suggested a new and efficient way to deal with the intractable problem of fast and highly secure image encryption. In this paper, we propose a one round chaos-based image encryption scheme based on the fast generation of large permutation and diffusion keys. In this scheme, at the permutation step, chaotic numbers are generated using a logistic map to shuffle the pixel positions without changing its value and at diffusion step, shuffled image is split in $n$ sub-images and the combination of PWLCM (piecewise linear chaotic map) with solutions of LDE (linear Diophantine equation) are generated to mask the pixels in each sub-image. To achieve high speed as well as high security level performances, the encryption process is carried out by updating above diffusion key by including inside $d > 3$ chaotic numbers. The main advantages of such a method are the abilities to produce a large key space to resist brute-force attacks. Extensive cryptanalysis has been performed on the proposed scheme using various types of analysis such as entropy analysis, differential analysis, various statistical analyses, key sensitivity analysis, key space analysis and speed analysis. The experimental results indicate that the proposed algorithm has a satisfactory security level with a low computational complexity, compared to the two round encryption schemes, which renders it a good candidate for real-time secure image transmission applications.

Key words: Image encryption, linear Diophantine equation, chaotic maps, the sorting of chaotic sequences.

1. Introduction

During the past two decades, the most extensive studies have been done in the theory of chaos in different fields of physics, engineering, biology, and economics as well [1]. Since 1990s, many researchers have noticed that there exist a close relationship between chaos and cryptography: many properties of chaotic systems have their corresponding counterparts in traditional cryptosystems [2]. Due to bulky data capacity and high correlation among pixels in image files, traditional techniques are not suitable for image encryption in real time [3]. Compared with traditional methods (such as AES and DES), chaos-based image encryption schemes have shown superior performance because they have many important properties, such as the sensitive dependence on initial conditions and control parameters, pseudorandom property, non-periodicity and topological transitivity, etc. [4]. Those properties are of great importance in diffusion and confusion processes [5].
In Ref. [6], Fridrich suggested that a chaos-based image encryption scheme should compose the iterations of two processes: permutation and diffusion. This architecture forms the basis structure of a number of chaos-based image ciphers recently proposed in Refs. [7-16]. In order to decorrelate the strong relationship between adjacent pixels, the permutation process is usually used. The permutation operation only shuffles the pixels positions without changing its value. But, the shuffled and original images have the same entropy and therefore, the shuffled image is weak against statistical attack and known plain-text attack. Diffusion step is adopted to increase the level of disorder in an image. In the diffusion process, the pixel values are altered sequentially and the change made to a particular pixel depends on the accumulated effect of all the previous pixel values. However, in Refs. [7-16], most of achievements are focus on security improvements, while only a few are dealing with efficiency issues. In Refs. [7-12], for example, authors deal with security improvements. Many of them need at least tree-rounds of the substitution-diffusion process to obtain satisfactory performance. While in Refs. [13-16], authors focus on efficiency improvements and only need one round of the substitution-diffusion process to perform high security level and speed performance. However, some of proposed algorithms lead to a longer processing time in a single round.

For the most widely investigated permutation-diffusion type image cipher, the diffusion is the highest cost of the whole cryptosystem. This is because in the diffusion stage, the subsequent quantization is required by the key stream generation and a considerable amount of computation load is devoted to the real number arithmetic operation. Therefore, the key problem of design an efficient image cryptosystem is how to either reduce the computational complexity or improve the diffusion effect of the diffusion module. In this paper, to avoid the cyclic digitization of chaotic numbers in the generation of diffusion and permutation keys and then achieve high speed performance, we propose to sort chaotic sequences of the logistic map in order to shuffle entire image at the permutation step. At the diffusion step, this shuffle image is then split in $n$ sub-image and we combine the piecewise linear chaotic map with solutions of the linear Diophantine equation to mask the pixels in each sub-image. The sorting of chaotic sequences by ascending order is considered as the permutation or diffusion key. That diffusion key is dynamically updated by including inside chaotic numbers. The main advantages of all the above methods are the abilities to produce a large key space and to enhance the speed of the proposed algorithm. Experimental results indicate that a satisfactory security level can be obtained with only one cipher cycle by using the new scheme. The remainder of this paper is organized as follows: Logistic map and the permutation operator based on it are described in Section 2; in Section 3, the proposed image cryptosystem is presented; the key schedule is described in Section 4; the security and efficiency analysis are discussed in Section 5 and the conclusions are made in Section 6.

2. Image Permutation Based on Chaotic System

2.1 The Logistic Map

The chaotic system that we consider here is the logistic map. The logistic map is one dimensional chaotic system with $x$ output and input variable, one initial condition $x_0$ and one control parameter $\lambda$ and can be described as follows:

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

(1)

where $x_n \in (0, 1)$ is the state of the system (for $n = 0, 1, 2, \ldots$) and $\lambda \in [0, 4]$ is the control parameter. For different values of parameter $\lambda$, the logistic sequence shows different characteristics. For $3.58 \leq \lambda \leq 4$, the logistic map has a positive Lyapunov exponent and thus is always chaotic as shown in Fig. 1 [17, 18]. So all the $(x_0, \lambda)$ where $x_0 \in [0, 1]$ and $3.58 \leq \lambda \leq 4$ can be
A Fast Image Encryption Algorithm Based on Chaotic Maps and the Linear Diophantine Equation

2.2 Construction of the Permutation Key Based on Chaotic System

A permutation of plain-image pixels is usually performed to reduce the high correlation among neighboring pixels. In this section, we use logistic map described in Subsection 2.1 to generate chaotic sequences \( x \) and then sort that chaotic numbers by ascending order for a generation of permutation key. However, the method can be used with any type of chaotic system. A new method for permuting the position of pixel in the image is then designed.

Without loss of generality, for a 256 gray-scale image, we assume that the original plain-image \( A \) has a size of \( m \times n \). We sort the chaotic sequences to shuffle \( A_{m \times n} \) using algorithm fragment 1 and we then obtain the permuted image \( A'_{m \times n} \).

- Algorithm Fragment 1 (to get shuffle image)
  Get rid of transient effect;
  Require: \( x_0; \lambda \);
  Initialisation: \( x(1, 1) = \lambda \cdot x_0 \cdot (1 - x_0) \);
  for \( i = 2 : m \times n \)
      \( x(i, 1) = \lambda \cdot x(i-1, 1) \cdot (1 - x(i-1, 1)) \);
  end
  \([y, \text{Index}] = \text{sort}(x,'\text{ascend}')\);
  \( A'_{m \times n} = A_{m \times n}(\text{Index}) \);
  \textbf{Index} is the permutation key for encryption. It corresponds to the position index sequences (an array of indices) while \( y \) corresponds to ascending order of chaotic values.

Sequence \( A'_{m \times n} \) corresponds to the permuted image and the reconstruction of \( A_{m \times n} \) cannot be made unless \( \text{Index} \) is determined.

3. The Proposed Image Cryptosystem

Such a permutation function is now integrated in the encryption algorithm. The encryption algorithm proposed in this paper is based on permutation-diffusion architecture. The overall architecture of our cryptosystem shown in Fig. 2 is divided into three stages: permutation of pixels in the image, masking the pixel in a sub-image and permuting the image as a whole.

3.1 Permutation of Pixels: Confusion Stage

The technique used for the permutation of pixels is based on the ascending or descending sorting of the solutions of the chaotic system as presented in Subsection 2.2. 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 employ a diffusion procedure next to improve the security.

3.2 Masking the Pixel in a Sub-Image

A diffusion of plain-image pixels is usually performed to increase the entropy of the plain image by making its histogram uniform. Pixel masking is performed by using the logical XOR operation. The masking of the image is performed using a diffusion key \( X_M \) such that

\[
X_M = X \mod 256
\]  

(2)

The procedure is shown in the Algorithm Fragment 3 below.

For the fast generation of diffusion key, the LDE (linear Diophantine equation) is used. The LDE is
A Fast Image Encryption Algorithm Based on Chaotic Maps and the Linear Diophantine Equation

Fig. 2 Synoptic of the proposed scheme.

defined by Eq. (3) below:

$$ax + by = c$$  \hspace{1cm} (3)

The LDE yields the fast generation of large diffusion key, which is based on the sorting of its solutions. The general solution $(x, y)$ of Eq. (3) is such that

$$\begin{cases} x(t) = x_0 + \frac{b}{\Lambda} t \\ y(t) = y_0 - \frac{a}{\Lambda} t \end{cases}$$  \hspace{1cm} (4)

where $(x_0, y_0)$ is a particular solution; $\Lambda$ is the greatest common divisor (GCD) of $a$ and $b$; and $t \in \mathbb{Z}$.

For the set of solutions to be bounded, Eq. (4) can be rewritten as

$$\begin{cases} x(t) = x_0 + \frac{b}{\Lambda} t \mod (v_c(t)) \\ y(t) = y_0 - \frac{a}{\Lambda} t \mod (x(t)) \end{cases}$$  \hspace{1cm} (5)

where $t = (0, 1, \ldots, L - 1)$, $L$ is the diffusion length same as length of image, $v_c(t)$ is a constant as length of sub-image $\forall t$.

The fragment part of this algorithm is given below:

- Algorithm Fragment 2

Require: $H_1; H_2; W_{r}; \lambda_0$;

Initialisation: $W_0 = 0; W_{IC0} = 0$;

for $k = 0:15$

$$W_{\lambda\text{mbda}} = 2^k((k + 1))*key1(k + 1) + W_{\lambda\text{mbda}};$$

end

$$\lambda = \lambda_0 + W_{\lambda\text{mbda}}/(10\times W_{r});$$

$x_0 = W_{IC}/W_{r};$

Require: $x_{01}; x_{02}$ from chaotic system (PWLCM)

$$a = 2*\text{fix}(x_{01}.*2^p) + 1;$$

$$b = 2*\text{fix}(x_{02}.*2^p) + 1;$$

In this fragment algorithm, the control parameter $\lambda$ and the initial condition $x_0$ are deduced from the keys $H_1$ and $H_2$, as follow:

$$\lambda = \lambda_0 + W_{\lambda\text{mbda}}/(10\times W_{r})$$  \hspace{1cm} (6)

where, $\lambda_0$ is a constant such that the behavior of the system (9) remains chaotic for all the range of $\lambda$ and $x$;

$$W = \sum_{k=0}^{15} 2^{i/k}\text{key}1_k,$$ where $\text{key}1_k$ corresponds to the values assigned to the ASCII symbols of key $H_1$, $W_{r} = 8160$ is the greatest value of $W_{\lambda\text{mbda}}$;

$$x(0) = W_{IC}/W_{r};$$  \hspace{1cm} (7)

where $W_{IC} = \sum_{i=0}^{15} 2^{i/k}\text{key}2_k$ with $\text{key}2_k$ corresponds to the values assigned to the ASCII symbols of key $H_2$.

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 this algorithm fragment 2 as
A Fast Image Encryption Algorithm Based on Chaotic Maps and the Linear Diophantine Equation

\[
\begin{align*}
  a &= 2 \cdot E\left(2^p \cdot x_{01}\right) + 1 \\
  b &= 2 \cdot E\left(2^p \cdot x_{02}\right) + 1
\end{align*}
\]  

(8)

where \(2 \leq p \leq 50\) is the number of bits used for the quantization, \(x_{01}\) and \(x_{02}\) are the chaotic numbers and \(E(\cdot)\) is the integer part of function. Depending on the chaotic system, \(p\) is selected for increasing the sensitivity of the key, as the sensitivity of the chaotic system on the initial conditions should be judiciously exploited by the quantization process. The precision used for the digitization of the chaotic values by using Matlab is about \(\epsilon = 2^{-52} \approx 10^{-15}\).

The chaotic numbers are generated using a chaotic system. In this work, we have considered PWLCM (piece wise linear chaotic map) described by the following equation:

\[
x(n) = F[x(n-1)] = \begin{cases} 
  x(n-1) \times \frac{1}{r}, & \text{if } 0 \leq x(n-1) < r \\
  \frac{1}{0.5-r} \times [x(n-1) - r], & \text{if } r \leq x(n-1) < 0.5 \\
  F[1 - x(n-1)], & \text{if } 0.5 \leq x(n-1) < 1
\end{cases}
\]  

(9)

The PWLCM is known to be chaotic when its control parameter \(r\) is within \([0, 0.5]\) and its initial condition is chosen within the interval \([0, 1]\) [19].

Then these chaotic numbers are used to generate the coefficients of the LDE (linear Diophantine equation). The solutions of LDE are used to generate the diffusion key for encryption. In the proposed scheme, the total image frame is divided into sub-images. For the first sub-image the encryption is carried out by using algorithm fragment 3 and for the other sub-images, the diffusion key is only refreshed without solving LDE equations. This helps to save computational time and at the same time the length of diffusion key is large enough to attain high security level. Once all the sub-images are encrypted, the total image is again encrypted with a different chaotic number.

The coefficients \(a\) and \(b\) are then used to generating diffusion key as explained in Subsection 2.2 using Algorithm Fragment 3 below.

- **Algorithm Fragment 3**
  
  \( N = \text{length}(t) \);

\[ [G, C, D] = \gcd(a, b) \]

\[ x = \text{mod}(C + (b/G)*t, N) + 1 \]

\[ y = \text{mod}(D - (a/G)*t, x) + 1 \]

\[ [I, J] = \text{sort}(x, \text{‘ascend’}) \]

\[ [I_1, J_1] = \text{sort}(y, \text{‘ascend’}) \]

\[ I_Z = J(I_1) \]

\[ X = \text{mod}(b*x + a*y, 256) \]

In the Algorithm Fragment 3, \(I_Z\) is the permutation key for permuting the image as a whole. For the security to be strengthened, it is necessary for the permutation key \(I_Z\) to be updated. The updating process is carried out by replacing \(d\) 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 \(I_Z\).

**3.3 Permuting the Image as a Whole**

In this sub-section, we use the permutation key \(I_Z\) as shown in Algorithm Fragment 3. In order to increase the randomness in the entire ciphered image as well as the sensitivity of the cipher to small changes in the plain image, an image dependent initial condition is determined. 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 \(I_{Zi}\). \(I_{Zi}\) is a permutation of length \(L\) (equal to the length of the whole image) used for shuffling the 1-D image as a whole. The refreshing of \(I_{Zi}\) is obtained by inserting \(c_r\) in \(I_{Zi}\) at the location corresponding to its value; thereafter \(I_{Zi}\) is left shifted by \(c_s\) samples and then sorted to obtain the refreshed permutation key. The initial condition of the chaotic system \(c_0\) that depends on the pixel values of the image can be obtained using

\[
c_0 = \frac{\sum_{j=0}^{L-1} A'(j)}{255 \times L}
\]

(10)

where \(A’\) is the shuffled 1-D image, the control parameter is same as one used for the generation of \(I_Z\). Choosing large number of permutation length make it impossible for any statistical attack, as the number of possible combinations of the permutations
$n_p$ exponentially increases:

\[ n_p = (N!)^{\frac{L}{N}} \]  \hspace{1cm} (11)

where \( L/N \) is the number of sub-images, also equal to the number of permutations combined per round. \( N \) is the total number of sub-images in a plain image. This number of possibilities suggests a high robustness of the proposed scheme against statistical attacks.

The high security and low computational complexity are achieved by generating only one permutation from the sorting of the solutions of the LDE, then by dynamically updating \( d \) number of integers (\( d > 3 \)) in the permutation. This allows reducing the computational time required by the duplication of the image-scanning stage during permuting the image as a whole.

In Fig. 3, we present a flowchart of the suggested encryption and decryption algorithm.

4. Key Schedule

A key of 128 bits or 256 bits is required for symmetric-key cryptosystems for more security [20]. We used an external 256-bit key \((S_1, S_2, \ldots, S_n, \ldots, S_{32})\) where \( S_i \) are ASCII symbols to derive initial conditions and control parameters of the chaotic system. 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

\[ W = \sum_{i=0}^{15} 2^{i+1} K_i \]  \hspace{1cm} (12)

where \( K_i \) are values (0-255) of ASCII symbols \( S_i \) and \( W \) is the value from which the control parameters and initial conditions are deduced, depending on the chaotic system.

5. Performance Test and Analysis

5.1 Security Analysis

The crucial measure for the quality of a cryptosystem is its capability to withstand the attempts of an unauthorized participant, or an opponent, to gain knowledge about the unencrypted information. A good cryptosystem should resist all kinds of known attacks, such as known/chosen plain-text attack, cipher-text only attack, statistical attack, differential attack, and various brute-force attack. In this subsection, we perform statistical analysis using image Lena of size 512 \times 512, as shown in Fig. 4a. Some security analysis has been performed on the proposed scheme, including the most important ones like key space analysis, statistical analysis and key sensitivity analysis, which has demonstrated the satisfactory security of the proposed scheme, as discussed in the following.

5.1.1 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 large enough to make brute-force attack infeasible. 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 \(2^{256} \). By considering only symbols ‘a-z’, ‘A-Z’ and ‘0-9’, the complete key space of the proposed image encryption scheme is \(62^{32} = 2.27 \times 10^{57} \) which is large enough to resist brute-force attack.

5.1.2 Statistical Analysis

It is well known that passing the statistical analysis on cipher-image is of crucial importance for a cryptosystem actually, an ideal cipher should be strong against any statistical attack. In order to prove the security of the proposed image encryption scheme, the following statistical tests are performed.

(a) Histogram

An image histogram illustrates that how pixels in an image are distributed by plotting the number of pixels at each grayscale level. The distribution of cipher-text is of much importance [19]. More specifically, it should hide the redundancy of plain-text and should not leak any information about the plain-text or the relationship between plain-text and cipher-text. The
A Fast Image Encryption Algorithm Based on Chaotic Maps and the Linear Diophantine Equation

Fig. 3 Flowchart of the encryption and decryption algorithm.

Fig. 4 Histogram: (a) Lena of size 512 × 512; (b) shuffle image; (c) masking image of (b); (d)-(f) histogram of (a)-(c) respectively.

histograms of plain-image (Fig. 4a) and its ciphered image (Figs. 4b and 4c) produced by the proposed scheme are shown in Figs. 4d-4f, respectively. Where Fig. 4b is the shuffle image obtained with secret key \( x_0 = 0.75 \) and \( \lambda = 3.93695629843 \), and Fig. 4c the masking image of Fig. 4b obtained with \( H_1 = 'azertyuiopqsdfgj' \), \( H_2 = 'azertyuiopqsdfgj' \), the size \( N = 64 \times 64 \) of the sliding window and the number of rounds \( k = 1 \). It is clear from Fig. 4e that the histograms of the shuffle image is the same of the plain-image and it is clear from Fig. 4f that the histogram of the cipher-image are fairly uniform and significantly different from that of the plain image and hence does not provide any clue to employ statistical attack.

(b) Correlation of adjacent pixels

There is a very good correlation among adjacent
pixels in the digital image. Eq. (13) is used to study the correlation between two adjacent pixels in horizontal (HC), vertical (VC) and diagonal orientations (DC).

\[
\rho_{xy} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y}) \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2 \frac{1}{N} \sum_{i=1}^{N} (y_i - \bar{y})^2}
\]  

(13)

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 image and its cipher image are given in Table 1. 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. 5.

It is clear from Fig. 5 and Table 1 that the strong correlation between adjacent pixels in plain image is greatly reduced in the cipher image produced by the proposed scheme.

(c) Information entropy analysis

In information theory, entropy is the most significant feature of disorder, or more precisely unpredictability. It is well known that the entropy \(H(m)\) of a message source \(m\) can be measured by

\[
H = -\sum_{i=1}^{M} p(m_i) \log_2(p(m_i))
\]

(16)

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. This means that the cipher-images are close to a random source and the proposed algorithm is secure against the entropy attack.

If the output of a cipher emits symbols with the entropy value of less than \(M\), there is a certain degree of predictability which threatens its security. The test result on different image for one round is defined in Table 2. It appears that the entropy of ciphered images is almost equal to eight.

5.1.3 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 [21]

\[
NPCR_{AB} = \frac{\sum_{i,j} D(i,j)}{m \times n} \times 100
\]

(17)

where

\[
D(i,j) = \begin{cases} 
1 & A(i,j) \neq B(i,j) \\
0 & \text{otherwise}
\end{cases}
\]

(18)

The NPCR value for two random images, which is an expected estimate for a good image cryptosystem, is given by

\[
NPCR_{\text{expected}} = \frac{1 - \frac{1}{2^{\log_2(N)}}}{N} \times 100
\]

(19)

where \(N\) is the gray levels of the image. For instance, the expected NPCR for two random images with 256 gray levels is 99.609%.

Fig. 5  Correlation of horizontally adjacent pixels of the image Lena.
A Fast Image Encryption Algorithm Based on Chaotic Maps and the Linear Diophantine Equation

Table 1  Correlation of adjacent pixels in plain-images and cipher-images by the proposed algorithm.

<table>
<thead>
<tr>
<th>Image</th>
<th>Size</th>
<th>Type</th>
<th>HC</th>
<th>VC</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>512 × 512</td>
<td>Plain-image</td>
<td>0.9719</td>
<td>0.9850</td>
<td>0.9593</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cipher-image</td>
<td>0.0006</td>
<td>-0.0018</td>
<td>0.0003</td>
</tr>
<tr>
<td>Mandrill</td>
<td>512 × 512</td>
<td>Plain-image</td>
<td>0.9333</td>
<td>0.9121</td>
<td>0.8666</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cipher-image</td>
<td>-0.0001</td>
<td>-0.0005</td>
<td>-0.0017</td>
</tr>
<tr>
<td>Airplane</td>
<td>512 × 512</td>
<td>Plain-image</td>
<td>0.9663</td>
<td>0.9642</td>
<td>0.9370</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cipher-image</td>
<td>0.0031</td>
<td>0.0006</td>
<td>-0.0016</td>
</tr>
<tr>
<td>Cameraman</td>
<td>256 × 256</td>
<td>Plain-image</td>
<td>0.9335</td>
<td>0.9592</td>
<td>0.9087</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cipher-image</td>
<td>-0.0024</td>
<td>0.0034</td>
<td>0.0030</td>
</tr>
</tbody>
</table>

Table 2  Information entropy of plain-images and cipher-images by the proposed algorithm.

<table>
<thead>
<tr>
<th>Images</th>
<th>Type</th>
<th>Lena</th>
<th>Mandrill</th>
<th>Airplane</th>
<th>Cameraman</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entropy</td>
<td>Plain-image</td>
<td>7.4456</td>
<td>7.2199</td>
<td>6.7043</td>
<td>7.0097</td>
</tr>
<tr>
<td></td>
<td>Cipher-image</td>
<td>7.9994</td>
<td>7.9993</td>
<td>7.9994</td>
<td>7.9972</td>
</tr>
</tbody>
</table>

The second criterion, UACI is used to measure the average intensity of differences between the two images. It is defined as [21]

\[
UACI_{ab} = \frac{100}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left| A(i, j) - B(i, j) \right| \frac{1}{255} \tag{20}
\]

The UACI value for two random images is given by

\[
UACI_{\text{expected}} = \frac{1}{N^2} \sum_{i}^{N-1} \left( \frac{i(i+1)}{N-1} \right) \times 100 \tag{21}
\]

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

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 Figs. 6a and 6b, respectively. As can be seen from Fig. 6, four overall encryption rounds are needed to achieve a satisfactory security level by using Lian [9] encryption scheme, three overall encryption rounds are needed to achieve a satisfactory security level by using Zhu [8] encryption scheme; one overall encryption rounds are needed to achieve a satisfactory security level by using Fu [16] encryption scheme; one overall encryption rounds are needed to achieve a satisfactory security level by using the proposed scheme, a fairly good result can be obtained with only one cipher cycle, thus the efficiency of the image cryptosystem is significantly improved.

5.1.4 Key Sensitivity Analysis

This test is intended to emphasize the diffusion property of the proposed cryptosystem under consideration with respect to small changes in keys. This is important because otherwise an intruder might reconstruct parts of the plain-image from the observed cipher-image by a partly correct guess of the key used for encryption. The key sensitivity of a cryptosystem can be observed in two ways: (1) the cipher image cannot be correctly decrypted even though there is only a slight difference between the encryption and decryption keys; (2) completely different cipher images should be produced when slightly different keys are used to encrypt the same plain image.

To evaluate the key sensitivity of the first case, the plain Lena image is firstly encrypted using the test key \((x_0 = 0.75, \lambda = 3.93695629843, H_1 = \text{'azertyuiopqsdgfij'}, H_2 = \text{'azertyuiopqsdgf0'})\) and the resultant cipher image is shown in Fig. 7a. Then the ciphered image is tried to be decrypted using five decryption keys:

(i) \((x_0 = 0.75, \lambda = 3.93695629843, H_1 = \text{'azertyuiopqsdgfij'}, H_2 = \text{'azertyuiopqsdgf0'})\);

(ii) \((x_0 = 0.74, \lambda = 3.93695629843, H_1 = \text{'azertyuiopqsdgfij'}, H_2 = \text{'azertyuiopqsdgf0'})\);

(iii) \((x_0 = 0.75, \lambda = 3.93695629842, H_1 = \text{'azertyuiopqsdgfij'}, H_2 = \text{'azertyuiopqsdgf0'})\);
A Fast Image Encryption Algorithm Based on Chaotic Maps and the Linear Diophantine Equation

(iv) $x_0 = 0.75$, $\lambda = 3.93695629843$, $H_1 = \text{‘azertyuiopqsdfg1’}$, $H_2 = \text{‘azertyuiopqsdfg0’}$;

(v) $x_0 = 0.75$, $\lambda = 3.93695629843$, $H_1 = \text{‘azertyuiopqsdfgj’}$, $H_2 = \text{‘azertyuiopqsdfg2’}$.

The resultant decrypted images are shown in Figs. 7b-7f respectively. The differences between wrong deciphered images Figs. 7c-7f to plain image are 99.5911%, 99.6136%, 99.6166% and 99.6086%, respectively.

To evaluate the key sensitivity of the second case, the plain Lena image is encrypted using four slightly different test keys: (ii), (iii), (iv) and (v) above described. The corresponding cipher images are shown in Figs. 8a, 8b, 8d, 8f and 8h, respectively. The differences between any two cipher images are computed and given in Table 3. The differential images between Figs. 8a and 8b, Figs. 8a and 8d, Figs. 8a and 8f and Figs. 8a and 8h are shown in Figs. 8c, 8e, 8g and 8i, respectively.

5.2 Efficiency Analysis

Apart from the security consideration, efficiency is also an important aspect for a good image cryptosystem, particular for 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

![Fig. 6 NPCR and UACI performance of the proposed scheme and the others existing scheme. (a) NPCR performance; (b) UACI performance.](image)

![Fig. 7 Key sensitivity test: result 1. (a) Ciphered image using key (i); (b) deciphered image using key (i); (c) deciphered image using key (ii); (d) deciphered image using key (iii); (e) deciphered image using key (iv); (b) deciphered image using key (v).](image)
A Fast Image Encryption Algorithm Based on Chaotic Maps and the Linear Diophantine Equation

Fig. 8  Key sensitivity test: result 2. (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); (i) differential image between (a) and (h).

Table 3  Differences between cipher images produced by slightly different keys.

<table>
<thead>
<tr>
<th>Figures</th>
<th>Test keys</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(a)</td>
</tr>
<tr>
<td>(a)</td>
<td>(i)</td>
<td>---</td>
</tr>
<tr>
<td>(b)</td>
<td>(ii)</td>
<td>99.6227</td>
</tr>
<tr>
<td>(d)</td>
<td>(iii)</td>
<td>99.6101</td>
</tr>
<tr>
<td>(f)</td>
<td>(iv)</td>
<td>99.6113</td>
</tr>
<tr>
<td>(h)</td>
<td>(v)</td>
<td>99.6059</td>
</tr>
</tbody>
</table>

256 gray-scale images of size 512 × 512 is shorter than 130 ms. By comparing this result with those presented in Ref. [21], 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. On the other hand, the superior security of our algorithm compared to existing methods, compensates for speed shortcomings, if any.

6. Conclusions

Chaotic cryptography has received considerable attention when many researchers pointed out the existence of a strong relation between chaos and cryptography. The main advantage using chaos lies in the observation that a chaotic signal looks like noise for the unauthorized users and 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 keystream to encrypt the plaintext element by element. Different chaotic systems can be utilized to generate such keystreams. The key challenge being to consider the trade-offs between the security level and speed performance, in this paper, an encryption algorithm for the fast generation of large permutation and diffusion keys by combining a chaotic system with LDE has been investigated. At the permutation step, we have proposed to sort chaotic sequences of the logistic map in order to shuffle entire image. This procedure allows avoiding the cyclic digitization of chaotic numbers in the generation of permutation key. The shuffle image is then split in $n$ sub-image and we combine the Piecewise Linear Chaotic Map with solutions of the Linear Diophantine Equation to mask the pixels in each sub-image at the diffusion stage. Instead of generating large permutation key for each sub-image, the proposed algorithm used only one permutation from the sorting of the solutions of the LDE, then dynamically updated four integers in this permutation by using the values of chaotic system.
A Fast Image Encryption Algorithm Based on Chaotic Maps and the Linear Diophantine Equation

These values are sorted to obtain the new permutations. We stress out that this approach is applicable to all chaotic maps with mixing property and that the achievement of the keystreams with the desired long cycle length is almost easy. By using this technique, the spreading process is then significantly accelerated. As a result, the high level of security can be achieved with one encryption round. Simulations have been carried out to compare its performance with that of existing methods. The new scheme exhibits both higher security level and fast enciphering/deciphering speeds. Theoretical analysis and experimental results indicate that the proposed scheme possesses the advantages of acceptable encryption speed, large key space and high level of security, and can be implemented efficiently. Thus, the proposed encryption scheme can be suitable for applications like the real-time image encryption and secure transmission of confidential information in the Internet.

References