American Journal of Signal Processing 2012, 2(2): 15-22
DOI: 10.5923/j.ajsp.20120202.03

Efficient Cryptosystem Based on Chaotic Sequences Sorting

J. S. Armand Eyebe Fouda[1], J. Yves Effa[2,*], Bertrand Bodo[1], Maaruf Ali[3]

[1]Department of Physics, University of Yaoundé I, Yaoundé, P.O. Box 812, Cameroon
[2]Department of Physics, University of Ngaoundéré, Ngaoundéré, P.O. Box 454, Cameroon
[3]College of Computer Science and Engineering, University of Ha'il, Ha'il, Kingdom of Saudi Arabia

Abstract It is commonly known that mathematical representation of chaotic systems can be used as good candidates for information security. In this paper, we propose an image encryption algorithm based on the use of chaotic sequences sorted by ascending or descending order. The permutation key is defined as a distribution of indices derived from the sorting. This method allows us to easily achieve high performance of pseudorandom permutation through any type of chaotic systems using the true precision of the computer; high-speed encryption algorithm which could meet one of the multimedia encryption specific requirements such as the real-time constraint and high robustness against statistical cryptanalysis. The efficiency of the proposed algorithm is studied in the cases of the piecewise linear chaotic map (PWLCM) and the hyperchaotic Lorenz system. Statistical analyses of the simulation results confirm a high security level of the proposed cipher.

Keywords Chaos Cryptography, Cryptanalysis, Random Permutation, Adaptable Cipher

1. Introduction

Image encryption requires modifying the entropy of the image, thus, its histogram, by creating new words or adjusting the probabilities of the existing words when the whole alphabet is already used. The image encryption methods which are based on chaotic systems attract attention due to their effectiveness for digital multimedia encryption whilst exhibiting the required enhanced sensitivity to initial conditions and system parameters (ergodicity and mixing)[1-4]. Most of the existing chaotic cryptosystems are based on the pixels permutation and the use of the XOR logical function for bit substitution[5-11]. In the Xiang[12] algorithm for example, blocks of bits are circularly shifted and the number of bits to be shifted is randomly determined by the chaotic value obtained from a logistic map after 70 iterations. Even though this procedure is efficient, the total number of permutations is reduced by the precision used for the digitization of the chaotic values- thus leading to a poor exploitation of the alphabet. In order to increase the number of permutations to achieve a stronger encryption, pixels are combined to form complex blocks of 32 or 64 bits (groups of 4 or 8 pixels). In the Socek algorithm[13], the permutational method is a computational approach of degree eight, where indices of bits are permutated using the chaotic values. Although this algorithm is efficient and fast, it is disadvantageous in the increased propagation of errors when a bit error occurs in the encrypted image during transmission. Recently, Abir et al.[14] proposed a method with no propagation error by combining the Xiang and Socek permutation algorithms. The corresponding encryption entropy was greater than the Xiang algorithm result.

Bits permutation is the main part of encryption algorithms; therefore, it is important to develop efficient permutation techniques. The pixel combination provides satisfactory results in terms of permutation possibilities; unfortunately, this procedure is limited by the propagation errors. These errors increase as the number of combined pixels increases. Permutation techniques require that the bits be rearranged in their exact initial order during decryption which justifies the complexity of encryption algorithms. To avoid permutation errors, cryptographers make use of the cyclic bits and pixels permutation methods[14]. Though, these permutation techniques are chaos-controlled, the number of permutations remains less than $8!$ - that is the total number of possibilities for each pixel. For the Xiang permutation algorithm for example, the cyclic shift concerns blocks of bits and the number of bits to be shifted is controlled by the logistic map. For instance, for a block of N-bits divided into two blocks of length N_1 and N_2, the total number of permutation is $N!$ whereas using the cyclic shift the number of permutations is only $N_1!\times N_2!$ ($N!>N_1!\times N_2!$). Whatever the size of block of bits considered, for images encoded on eight bits the significant number of bit permutation per pixel cannot exceed $8!$.

P. Fei et al. also proposed an algorithm for image encryption[15] in which the aim was to randomly encrypt the three

* Corresponding author:
effa_jo@yahoo.fr (J. Yves Effa)
Published online at http://journal.sapub.org/ajsp

colours red (R) green (G) and blue (B) with three chaotic systems. The computational time was relatively large and the algorithmic steps could not be repeated for more security, as it is the case for many other encryption algorithms. Moreover, the correlation of adjacent pixels presented by this method could be improved, according to those presented in[14].

Chen et al.[16] proposed a symmetric cipher in which a two-dimensional chaotic map is generalized to three dimensions for designing a secure real-time image encryption scheme. This approach employs the three-dimensional cat map to shuffle the positions of the image pixels and another chaotic map to confuse the relationship between the original and the ciphered images. Guan et al.[17] presented an image encryption scheme in which shuffling the positions and changing the grey values of image pixels are combined to confuse the relationship between the plain-image and ciphered image.

The aims of the proposed algorithm are to easily achieve high performance of pseudorandom permutation through utilization of any type of chaotic systems using the true precision of the computer (without the need for digitization); high-speed encryption algorithm which could meet one of the multimedia encryption specific requirements such as the demanding real-time constraint and high robustness against statistical cryptanalysis.

The work is organized as follows: the algorithm is proposed in Section Two; the results are discussed in Section Three and the conclusions are made in Section Four.

2. Proposed Encryption Algorithm

The proposed algorithm in this paper is divided into two main steps: the reduction of the correlation between pixels and the increment of the entropy through a random distribution of the indices derived from an ascending or descending sorting of chaotic sequences. The complete algorithm is presented as follows:

1. Define a window of size *m*-by-*n* which could match a part or the whole plaintext image, then reshape it so as to obtain a 1-D signal of length $m \cdot n$;

2. Chose a 256-key and generate a chaotic sequence of length $m \cdot n$, then sort it in the ascending or descending order and save the corresponding distribution of indices Iy as a permutation key;

3. Reshape the whole image into a 1-D signal of length L and split it into windows of length $m \cdot n$;

4. Shuffle pixels in each of the signal window with the permutation key;

5. Mask the pixels in each of the shuffled window with the values in $Iz = Iy \bmod 256$;

6. Generate two chaotic integers k and l less than L and permute blocks of pixels $[k+1, L]$ and $[1, k]$, thereafter $[1, l]$ and $[l+1, L]$ in the whole image;

7. Repeat steps four to six p times;

8. Reshape the shuffled 1-D signal into 2-D image (ciphered image).

Initial conditions and control parameters of the chaotic systems constitute the secret key of the proposed cipher.

Steps one to three of the algorithm concern the initialization and steps four to six perform the pixels and bits shuffling.

The digitization of the chaotic sequence values does not allow the exploitation of the real accuracy of the computer like sorting. Therefore, digitization appears only in step six, which increases the sensitivity of our algorithm to any fluctuation that could occur in the chaotic sequence. Presented in this form, the proposed algorithm can be efficiently combined with any type of chaotic system.

2.1. Permutation of Pixels in a Window

The technique used for the permutation of pixels is based on the ascending or descending sorting of a chaotic sequence. By the chaotic scheme of the values in a generated sequence, we obtained a pseudorandom distribution of the positions (indices) of these values.

We defined X as a chaotic sequence and Y as the corresponding sorted sequence. We also defined $I\mathrm{x}$ as the distribution of values in the sequence X, $I\mathrm{x}$ was sorted by ascending. Sequences X and $I\mathrm{x}$ can then be expressed by:

$$X = \{x_1, x_2, \ldots\ldots x_i \ldots\ldots x_N\} \tag{1}$$

$$I\mathrm{x} = \{1, 2, \ldots\ldots i \ldots\ldots N\} \tag{2}$$

N being the length of X. In the sequence X, values were randomly distributed and these values were all different according to the statistic properties of the chaotic system and the precision of the computer. Sorting the sequence X by ascending, we obtained a new sequence Y in which the indices (positions) of the values were randomly distributed:

$$Y = \{x_8, x_3, \ldots\ldots\ldots, x_N, \ldots\ldots\ldots, x_i\} \tag{3}$$

Let $I\mathrm{y}$ be the sequence containing the position of the values of $I\mathrm{x}$ in the sequence Y, from (3), $I\mathrm{y}$ was expressed as:

$$I\mathrm{y} = \{8, 3, \ldots\ldots\ldots, N, \ldots\ldots\ldots, i\} \tag{4}$$

The ranking of values in $I\mathrm{y}$ differs from a sequence to another when X is a chaotic sequence. In this point of view, $I\mathrm{y}$ can be seen as a random variable and the number of probable events is equal to the number of permutations of the indices, this means N.

To a sequence Z obtained by reshaping an image $I(m,n)$, where $N = m \cdot n$, we applied the distribution $I\mathrm{y}$. Values in Z were initially ranked by ascension of the corresponding indices:

$$Z = Z(I\mathrm{x}) \tag{5}$$

After permuting the pixels in Z with $I\mathrm{y}$, we obtained a new sequence T such that:

$$T = Z(I\mathrm{y}) \tag{6}$$

Sequence T corresponds to the permuted image and the reconstruction of Z cannot be made unless the distribution $I\mathrm{y}$ is determined.

Pixels permutation does not modify the entropy of the

image, but it reduces the correlation between adjacent pixels. Increasing the image entropy requires the construction of new words in the image. This result can be performed by bit operations - hence the step of masking.

2.2. Masking the Pixel in a Window

Contrary to the indices permutation technique, the XOR can only be combined with binary values. For this purpose, we used the distribution of indices Iy for the pixels masking. Since this distribution presents values ranging from 1 to N, it is necessary to bring back values greater than 255 among[0, 255], according to the image format. The set Iz of values used for this operation is given by:

$$Iz = Iy \bmod 256 \tag{7}$$

2.3. Permuting Blocks of Pixels

Steps four and five subdivided the whole image into blocks of small sizes. In order to increase the randomness in the entire ciphered image, pixels should pass from a block to another. For this purpose, two chaotic values were generated and digitized such that the corresponding digital values (natural numbers) k and l correspond to indices of pixels in the whole 1-D image (of length L). These values are used to permute blocks of pixels $[1, k]$ and $[k+1, L]$ first and for the second time blocks $[1, l]$ and $[l+1, L]$ in the whole image. The initial condition of the chaotic system used for the generation of k and l are derived from the following relation:

$$c_{0i} = \frac{\sum_{j=0}^{L-1} T(j)}{255 \times L} \tag{8}$$

where T is the shuffled 1-D image, L its length, i the rank of the round and c_0 is an internal key; the control parameter being the one used for the generation of Iy. Choosing large number of rounds makes it impossible for any statistical attack[18].

2.4. Key Schedule

By today's standards, a key of 128 bits or 256 bits is required for symmetric-key cryptosystems[19]. We used an external 256-bit key ($s_1 s_2 \cdots s_i \cdots s_{32}$, 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:

$$W = \sum_{i=0}^{15} 2^{\frac{i}{i+1}} K_i \tag{9}$$

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 will be 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 the W presents an upper limit $W_r = 8160$, which is used for the normalization of W.

3. Test Results and Security Analysis

A good encryption scheme should resist any kind of known attacks: known-plaintext attack, ciphertext-only attack, statistical attack, differential attack and brute-force attack. In this section, some security analysis results on the scheme are described, including some important ones like key space analysis, statistical analysis, differential analysis, number of pixel change rate (NPCR) and unified average changing intensity (UACI) for one pixel difference in the plain-text image. Two chaotic systems are used for this purpose: the PWLCM as discrete chaotic system and the hyperchaotic Lorenz system as chaotic continuous-time system. Three plain-images were selected for the evaluation of the proposed algorithm: the image of Lena (size $512 \times 512 \times 3$); the image of mandrill (size $512 \times 512 \times 1$) and a uniformly black image (size $512 \times 512 \times 1$) whose statistical properties are summarized in Table 1.

Table 1. Statistical Properties of Plaintext Images

Images	Lena	Mandrill	Black
Horizontal correlation	0.9721	0.9333	1
Vertical correlation	0.9851	0.9121	1
Diagonal correlation	0.9595	0.8666	1
H	7.4565	7.2199	0

The NPCR between two ciphered images A and B is defined by:

$$NPCR_{AB} = \frac{\sum_{i=1}^{m}\sum_{j}^{n} D(i,j)}{m \times n} \times 100 \tag{10}$$

Where:

$$D(i,j) = \begin{cases} 1 & A(i,j) \neq B(i,j) \\ 0 & otherwise \end{cases} \tag{11}$$

$m \times n$ being the size of images A and B. Similarly, the unified average changing intensity (UACI) is defined by:

$$UACI_{AB} = \frac{100}{m \times n} \sum_{1}^{m}\sum_{1}^{n} \frac{|A(i,j) - B(i,j)|}{255} \tag{12}$$

Finally, the entropy H is computed for the measurement of the randomness:

$$H = -\sum_{i=1}^{2^b} p(m_i) \log_2 \left(p(m_i) \right) \tag{13}$$

where $p(m_i)$ is the probability of the event m_i and b the number of bits.

3.1. Case of the PWLCM

The PWLCM is defined by the following equation:

$$x(n) = F[x(n-1)] = \begin{cases} x(n-1) \times \frac{1}{a}, & if \quad 0 \le x(n-1) < a \\ [x(n-1) - a] \times \frac{1}{0.5 - a}, & if \quad a \le x(n-1) < 0.5 \\ F[1 - x(n-1)], & if \quad 0.5 \le x(n-1) < 1 \end{cases} \tag{14}$$

The PWLCM is known to be chaotic when its control

parameter a is within $]0,0.5[$ and its initial condition chosen within the interval $]0,1[$ [20]. For this purpose, the control parameter and the initial condition $x(0)=x_0$ are deduced from the key through the following relation:

$$\begin{cases} a_0 = 0.1934569872365 \\ a = a_0 + W/(10W_r) \\ x_0 = W/W_r \end{cases} \quad (15)$$

a_0 is a constant chosen such that the behavior of the PWLCM remains chaotic for any key derived from (9). In step six was also used the PWLCM with $x_{0i}=c_{0i}$ as initial condition.

Two encryption sub-keys $T_1 = azertyuiopqsdfgi$ and $T_2 = azertyuiopqsdfg0$ were used respectively for the control parameter and the initial condition, thus forming a 32 ASCII symbols key T_1T_2:

($azertyuiopqsdfgiazertyuiopqsdfg0$).

3.1.1. Key Space Analysis

a) The Key Space

The proposed scheme has a 256-bit key corresponding to 32 ASCII symbols. The key space is the number of sets of 32 ASCII symbols that can be built. In hexadecimal representation, the number of different combinations of secret keys is equal to 2^{256}, provided that the length of each block to be shuffled is such that $(m \cdot n)! >> 2^{256}$. By considering only symbols "a-z", "A-Z" and "0-9", the scheme presents $62^{32} = 2.27 \cdot 10^{57}$ secret keys. A cipher with such a large key space can resist all present kinds of brute-force attack.

b) Sensitivity of the Key

High key sensitivity is generally required for preventing adaptive chosen-plaintext attacks and linear cryptanalysis. The key T_2 was partially changed to perform the test on the sensitivity of the key with the proposed approach. The sensitivity of the key is then performed according to the following steps:

First the image of Lena is encrypted by using $T_{2a} = azertyuiopqsdfg0$ as key for initial condition;

Then the least significant bit of T_2 is changed, so that the initial key becomes $T_{2b} = azertyuiopqsdfg1$ in this example, which is used to encrypt the same image of Lena;

Finally, the above two ciphered images, encrypted by the two slightly different keys are compared.

The number of rounds in this experiment is *p=10*. The results thus obtained are illustrated in Figure 1. From left to right and from top to bottom are presented the original image, the first and second encrypted images and the difference of the two encrypted images. The comparison of the ciphered images shows that the image encrypted by $T_{2a} = azertyuiopqsdfg0$ has 99.63% difference from the image encrypted by the key $T_{2b} = azertyuiopqsdfg1$ in terms of pixel grey scale values, although there is only one bit difference in the two secret keys.

In order to quantify the dependency between the two ciphered images, the Pearson's correlation coefficient r_{AB} is used:

$$r_{AB} = \frac{\mathrm{cov}(A,B)}{\sqrt{\mathrm{cov}(A,A)\cdot\mathrm{cov}(B,B)}} \quad (16)$$

where A and B are the images to be compared. Table 2 recapitulates the values obtained for three plain-images.

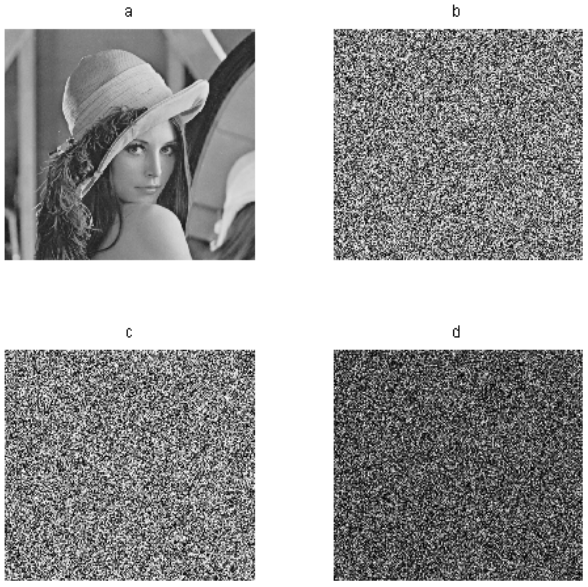

Figure 1. Key sensitive test (result 1). a) Plain-image, b) Image encrypted by T_{2a}, c) Image encrypted by T_{2b}, d) Difference of the two ciphered images

Table 2. Correlation and Pixels Difference Between Images Encrypted by Two Slightly Different Keys

Images	Lena	Mandrill	Black
Correlation between ciphered images	-0,0020	0,0021	0,0016
Pixels difference(%)	99,628	99,596	99,59

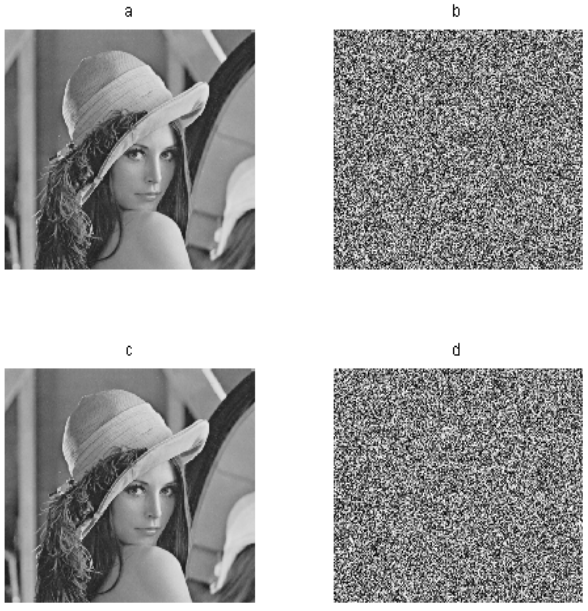

Figure 2. Key sensitive test (result 2). a) Plain-image, b) Image encrypted by T_{2a}, c) Image decrypted by T_{2a}, d) Image decrypted by T_{2b}

Moreover, when a key is used to encrypt an image and another one trivially modified (slightly modified) is used to

decrypt the ciphered image, the decryption process should not succeed. Such a result is confirmed in Figure 2 where the key used for encryption is $T_{2a} = azertyuiopqsdfg0$ and the one used for decryption is $T_{2b} = azertyuiopqsdfg1$. From left to right and from top to bottom are presented the original, the encrypted, the decrypted and the unsuccessfully decrypted images. There is also only one bit difference between the two keys. The results in Table 2 and Figure 2 show that there is no relation existing amongst the encrypted images corresponding to a small change in the key, thus confirming that the proposed algorithm is highly key sensitive.

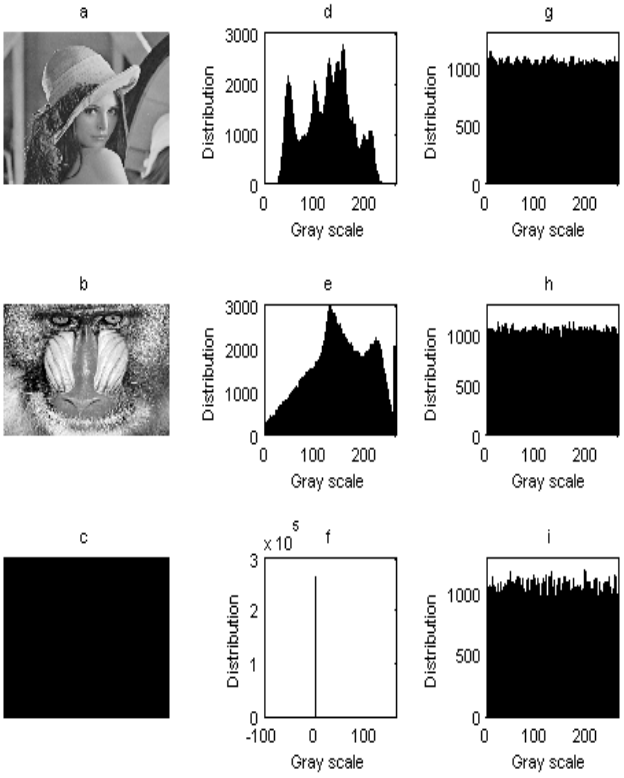

Figure 3. Histograms of plain-images and ciphered images. a)-c) plain-images, d)-f) histograms of plain-images, g)-i) histograms of ciphered images

3.1.2. Statistical Analysis

a) Histogram of the Ciphered Images

The goal in encryption is to obtain some uniform histograms. Histograms of Several 256 gray-scale images of size 512×512 which have different contents were calculated. Typical examples among them are shown in Figure 3. From this figure, the histograms of the ciphered images are fairly uniform and significantly different from those of the plain-images. Although distributive characters of plain-images histograms are all different, the histograms of ciphered images are all fairly uniform, thus making difficult deducing the secret key from the cipher-text during the known/chosen plaintext attacks.

b) Correlation of Two Adjacent Pixels

The correlation distributions of two horizontally adjacent pixels of plaintext and ciphered images are depicted in Figure 4, where the dispersion of the pixels after encryption can be appreciated. The correlation coefficients corresponding to the image of Lena for example are 0.9721 and 0.0010, respectively. Similar results for vertical and diagonal directions were obtained and are presented in Table 3. The analysis of correlation coefficients proves that the proposed encryption technique satisfies zero co-correlation property, thus its robustness against statistical attacks.

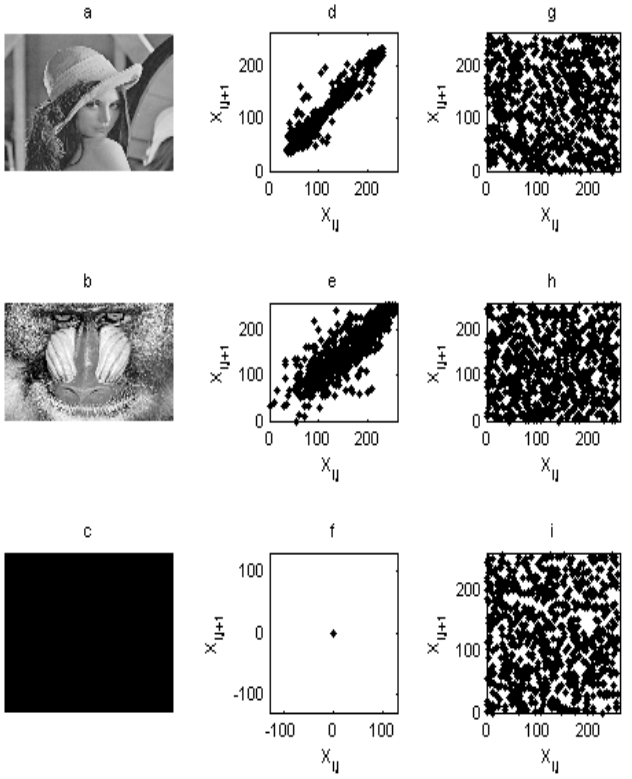

Figure 4. Distribution of horizontally adjacent pixels in plaintext and ciphered images. a)-c) plain-images, d)-f) correlations in plain-images, g)-i) correlation in ciphered images

c) Information Entropy Analysis

Information entropy is defined to express the degree of uncertainties or randomness in a given system. The entropy *H(m)* of a message source is calculated as:

$$H(m) = -\sum_{i=1}^{2^b} p(m_i)\log_2\left(p(m_i)\right) \quad (17)$$

where $p(m_i)$ is the probability of symbol m_i. The entropy is expressed in bits. In the case of 256 gray-scale images, truly random image entropy is equal to eight, which is the ideal value. The entropy of a practical source generating random messages is smaller than the ideal one. However, the entropy of encrypted messages should be eight; otherwise there exists a certain degree of predictability which threatens its security. Table 4 gives the entropy of images encrypted by the proposed scheme. It appears that the entropy of ciphered images is almost equal to eight, compared to that of the plain-images in Table 1.

The above results obtained are comparable to those presented in many other secured ciphers such as the one presented by Chen[16] and that of Guan[17]. In Table 5, a direct comparison can be made between these ciphers and the one

proposed in this paper, in the case of three images.

Table 3. Correlation of Adjacent Pixels in Images Encrypted by the Proposed Algorithm

Images	Lena	Mandrill	Black
Horizontal Correlation	-0,0010	0,0005	-0,0072
Vertical Correlation	-0,0016	-0,0032	0,0065
Diagonal Correlation	0,0010	0,0016	0,0020

Table 4. Information Entropy of Ciphered Images

Images	Lena	Mandrill	Black
H	7,9993	7,9992	7,9970

Table 5. Comparison of Correlation Coefficients

		Correlation coefficient between plaintext and ciphered image		
Image	Type	Guan	Chen	Proposed
Lena	Gray	0.0089	0.0042	-0.0027
Saturn	Gray	0.0136	0.0039	-0.0003
Airplane	Gray	0.0067	0.0025	0.0030

3.1.3. Differential Attack

One of the desirable properties of the cipher is its sensitivity to small changes in the plain-image (single pixel change). In order to test the influence of one-pixel change in the plain-image, encrypted by the proposed scheme, NPCR and UACI of two ciphered images whose corresponding plain-images have only one pixel difference are compared (see Table 6). NPCR for all the images are over 99.5%, showing that the encryption scheme is very sensitive with respect to small pixel changes in the plain-image. UACI values are all close to the ideal value of 33.33%, indicating that the rate of one pixel change is very large. These values of NPCR and UACI, in the case of one pixel change, confirm the high sensitivity of the proposed algorithm with respect to the plain-image, hence its robustness against differential attacks.

Table 6. Sensitivity to Differential Attacks

Images	Lena	Mandrill	Black
Correlation between ciphered images	-0,0007	0,0013	-0,0007
NPCR	99,603	99,609	99,599
UACI	33,456	33,447	33,419

3.1.4. Effect of the Size of the Window and the Number of Rounds

For a high security level, it is necessary to recursively repeat steps 4 to 6 p times. p is considered to be a security parameter. The size of the window can be considered as another security parameter as it allows considerable increase of the number of permutations of the pixels as well as the randomness of the values used for bit substitution. The results presented above were obtained with $p = 10$ rounds and a window of size 256×256 pixels.

Now, let us study the effect of these two parameters on the ciphered image of the mandrill. Figure 5 shows the behaviors of correlation coefficients of adjacent pixels and correlation between plaintext and ciphered images in terms of the number of rounds and the size of the window. From left to right, the columns represent respectively the correlation of horizontally, vertically and diagonally adjacent pixels; the fourth column represents the correlation between plaintext and ciphered images. From top to bottom, are presented in each row, the results of windows of size 32×32, 64×64, 128×128 and 256×256 pixels. It appears on this figure that the increase of the size of the permutation window allows the reduction of the correlation between adjacent pixels. The behavior of these correlation coefficients in terms of the number of rounds is not uniform; large number of rounds does not necessarily guarantee small correlation coefficients. However, the number of rounds should be greater than two for obtaining a high security level. The security level can also be strengthened in step six, by combining more than two permutations.

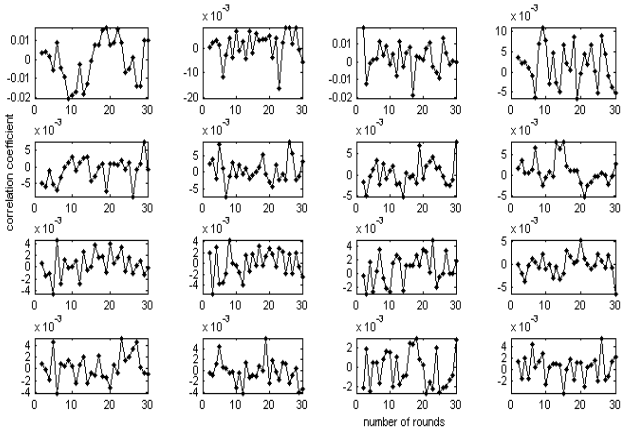

Figure 5. Correlation coefficients in terms of the number of rounds and the size of the permutation window. column 1: correlation of horizontally adjacent pixels, column 2: correlation of vertically adjacent pixels, column 3: correlation of diagonally adjacent pixels, column 4: correlation between plaintext and ciphered image

Similarly, we observe in Figure 6 that the NPCR and UACI do not significantly change in terms of the size of the window as well as the number of rounds.

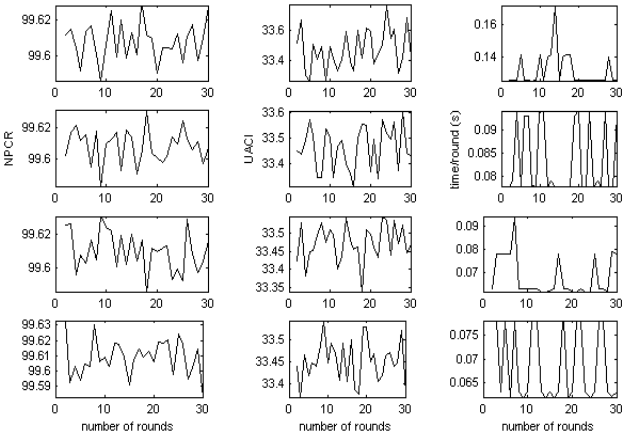

Figure 6. NPCR (column 1), UACI (column 2) and average time/round (column3) in terms of the number of rounds and the size of the permutation window

3.1.5. Speed Performance

Apart from the security consideration, the running speed or execution of the algorithm is also an important aspect for a good encryption scheme. The simulator for the proposed scheme is implemented using Matlab 7.0. 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, according to Figure 6, is 3.83MB/s per round in the case of using windows of size 256×256 pixels.

3.2. Case of the Hyperchaotic Lorenz System

To show the adaptability of the algorithm to any type of chaotic system, a hyperchaotic Lorenz system (18) is used:

$$\begin{cases} \dot{x} = a(y-x) + w \\ \dot{y} = -xz + rx - y \\ \dot{z} = xy - bz \\ \dot{w} = -xz + dw \end{cases} \quad (18)$$

with $a = 10$, $r = 28$, $b = 8/3$ and $d \in [1, 1.3]$. The initial conditions are chosen such that $x_0 \in [-5, -3]$, $y_0 = -3$, $z_0 = 20$ and $w_0 = 10$, the system exhibits hyperchaotic behavior as shown in[21]. As in the case of the PWLCM, x_0 and d are varied as follows:

$$\begin{cases} d = 1 + 3W/(10W_r) \\ x_0 = -3 - 2W/W_r \end{cases} \quad (19)$$

The initial conditions for step six are also chosen such that $y_0 = -3$, $z_0 = 20$, $w_0 = 10$ and:

$$x_{0i} = -3 - c_{0i} \quad (20)$$

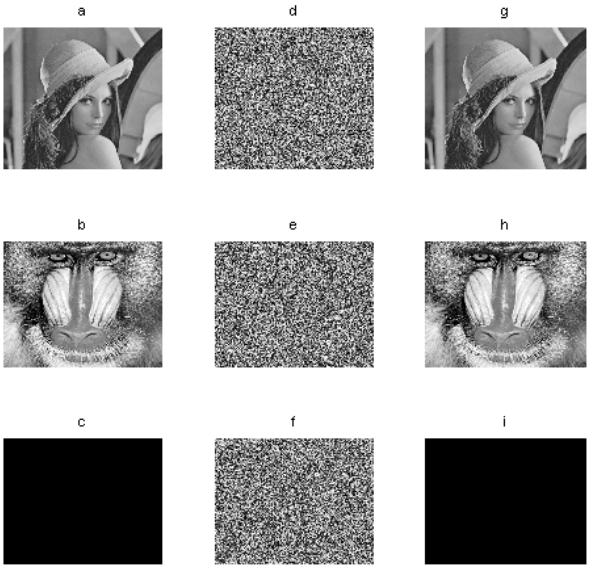

Figure 7. Encrypted and decrypted images by the hyperchaotic Lorenz system: a)-c) plain-images; d)-f) ciphered images; g)-i) deciphered images

The keys used for this experiment are $T_1 = azertyuiopqsdfgh$ and $T_2 = azertyuiopqsdfh0$, and the results produced are given in Table 7. Figure 7 shows the corresponding encrypted and decrypted images presented as follows: plain-images are shown in the first column, the ciphered images in the second and the decrypted images in the third. For this experiment, the number of rounds was also equal to ten and only sequences given by the state x of the hyperchaotic sytem was used. The complexity of the algorithm could be enhanced by combining the four states x, y, z and w in steps four to six. According to Table 7, the sensitivity of the key remains higher, as in the case of the PWLCM.

Table 7. Security Evaluation in the Case of the Hyperchaotic Lorenz System

Images	Lena	Mandrill	Black
Horizontal Correlation	0.0003	0.0035	0.0067
Vertical Correlation	-0.0013	0.0009	-0.0037
Diagonal Correlation	-0.0021	0.0033	0.0020
Correlation between ciphered	2.53×10^{-5}	-0.0020	-
NPCR	99.593	99.613	99.58
UACI	33.478	33.375	33.595
H	7.9992	7.9993	7.9976

4. Conclusions

This paper presented a permutation technique based on chaotic sequence sorting. It takes advantage of the true accuracy of the computer used for chaotic sequence generation, thus increasing the sensitivity of the key. It also presents the advantage that it can be combined with any type of chaotic system, which makes the proposed cipher easily adaptable. According to the execution time, the proposed scheme could guarantee real-time encryption. By comparison with other secured ciphers, the results obtained confirm the high robustness of the proposed scheme against known attacks. In the future, we plan to make the initialization phase dynamic, so as to increase the complexity of the proposed algorithm.

ACKNOWLEDGEMENTS

We thank Dr. Mala William for his assistance in editing of this manuscript.

REFERENCES

[1] L. Tang, “Methods for encrypting and decrypting MPEG video data efficiently”, in Proceedings of the 4th ACM International Multimedia Conference, pp. 219-230, 1996

[2] H. Cheng, and X. Li, “Partial encryption of compressed images and videos”, IEEE Transactions on Signal Processing, 48, pp. 2439-2451, 2000

[3] J.-C. Yen, and J.-I. Guo, “A new chaotic key-based design for image encryption and decryption”, in Proceedings of 2000 IEEE International Conference on Circuits and Systems, ISACS 2000, 4, pp. 49-52, 2000

[4] B. Bhargava, C. Shi, and S.-Y Wang, 2004, MPEG video encryption algorithms: Multimedia Tools and Applications, Kluwer Academic Publishers, 24(1), 57–79.

[5] A. Riaz, and M. Ali, "Chaotic Communications, Their Applications and Advantages over Traditional Methods of Communication", IEEE Commun. Syst., Networks Digital Signal Process, pp. 21-24, 2008

[6] G.J. Millérioux, M. Amigo, and J. Daafouz, 2008, A Connection between Chaotic and Conventional Cryptography, IEEE Trans. Circuits Syst., 55(6), 1695-1703.

[7] L. Kocarev, 2001, Chaos Based Cryptography: A Brief Overview, IEEE Trans. Circuits Syst. Mag., 1(3), 6-21.

[8] G. Alvarez, and S. Li, 2006, Some Basic Cryptographic Requirements for Chaos Based Cryptosystems, Int. J. Bifurcation Chaos, 16(8), 2129-2151.

[9] T. Yang, C.W. Wu, and L.O. Chua, 1997, Cryptography Based on Chaotic Systems, IEEE Trans. Circuits Syst., 44(5), 469-472.

[10] G. Jakimoski, and L. Kocarev, 2001, Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps. IEEE Trans. Circuits Syst., 48(2), 163-169.

[11] S. Li, G. Chen, and X. Zheng, Chaos-based encryption for digital images and videos, in B. Furht and D. Kirovski (Eds.), Multimedia Security Handbook, 4 of Internet and Communications Series, Ch. 3, CRC Press, December 2004.

[12] T. Xiang, X. Liao, K. Wong, G. Tang, and Y. Chen, "A Novel Block Cryptosystem Based on Iterating a Chaotic Map", Phys. Lett. A 349, pp. 109-115, 2006

[13] D. Socek, S. Li, S. S. Magliveras, and B. Furht, "Enhanced 1-D Chaotic Key Based Algorithm for Image Encryption", IEEE Security Privacy for Emerging Areas in Commun Networks, pp. 406-407, 2005

[14] A. Awad, and D. Awad, "Efficient image chaotic encryption algorithm with no propagation error", ETRI journal 32, pp. 774-783, 2010

[15] P. Fei, S-S. Qiu, and L. Min, "An image encryption algorithm based on mixed chaotic dynamic systems and external keys", IEEE int conf commun Circuits & systems, pp. 1135-1139, 2005

[16] G. Chen, Y. Mao, C.K. Chui, "A symmetric image encryption based on 3D chaotic maps", Chaos, Solitons and Fractals, 21, pp. 749-761, 2004

[17] Z.H. Guan, F. Huang, and W. Guan, "Chaos-Based Image Encryption Algorithm", Phys. Lett. A 346, pp. 153-157, 2005

[18] S.G. Lian, J. Sun, Z. Wang, "A block cipher based on a suitable use of chaotic standard map", Chaos, Solitons and Fractals ,26, pp. 117-29, 2005

[19] D.R. Stinson, Cryptography: theory and practice, CRC Press, second edition, 2002

[20] H. Zhou, A design methodology of chaotic stream ciphers and the realization problems in finite precision, Ph.D. thesis, Department of Electrical Engineering, Fudan University, Shanghai, China, 1996.

[21] D. Lu, A. Wang, and X. Tian, "Control and synchronization of a new hyperchaotic system with unknown parameters", International Journal of Nonlinear Science 6, pp. 224-229, 2008