Applied Mathematics, 2012, 3, 1910-1920
http://dx.doi.org/10.4236/am.2012.312262 Published Online December 2012 (http://www.SciRP.org/journal/am)

Image Encryption Algorithm Based on a Chaotic Iterative Process

Michael François[1], Thomas Grosges[1], Dominique Barchiesi[1], Robert Erra[2]
[1]Group for Automatic Mesh Generation and Advanced Methods, Gamma3 Project (UTT-INRIA), University of Technology of Troyes, Troyes, France
[2]Network & Information Security, Ecole Supérieure d'Informatique, Electronique, Automatique, Paris, France
Email: thomas.grosges@utt.fr

Received September 17, 2012; revised October 25, 2012; accepted November 5, 2012

ABSTRACT

The paper describes a symmetric encryption algorithm based on bit permutations and using an iterative process combined with a chaotic function. The main advantages of such a cryptosystem is its ability to encrypt securely bit sequences and assuring confusion, diffusion and indistinguishability properties in the cipher. The algorithm is applied on the image encryption where the plain-image is viewed as binary sequence. The results of statistical analysis about randomness, sensitivity and correlation on the cipher-images show the relevance of the proposed cryptosystem.

Keywords: Encryption; Image Encryption; Chaotic Function; Confusion; Diffusion; Indistinguishability

1. Introduction

The use of the internet and network applications becomes indispensable in the modern society and the necessity to protect and secure such applications from prying eyes is essential. On internet, the informations can be your credit card numbers, bank account informations, health/social security informations or even personal communications with everybody. Such a challenge is the concern of cryptography that is developed and used to hide information against unauthorized users. Several encryption algorithms are available and used in information security. They can be classified into two categories: asymmetric (public) and symmetric (private) encryption algorithms. For asymmetric encryption, two keys are used: private and public keys. Public key is used for encryption and private key is used for decryption (e.g. RSA [1]). In symmetric case, only one key which must be secret is used to encrypt and decrypt the data. Some of the best known algorithms are DES, 3DES, BLOWFISH, IDEA or AES [2,3]. DES algorithm uses one 64-bits key where 8 bits are used solely for checking parity. Triple DES (3DES) uses three 64-bits keys. BLOWFISH has a 64-bits block size and a key length from 32 bits to 448 bits. IDEA operates with 64-bits block size and is controlled by a 128-bits key while AES uses various keys of 128, 192 and 256 bits. However, these algorithms do not always keep the same efficiency (quality of the cipher, execution time, etc.) depending on the structure of the input data (texts, images) [4,5].

Here we propose an encryption algorithm which is able to deal efficiently with any type of input data. The input data is considered as a simple binary sequence (or vector) constituted only by the bits "0" and "1". The algorithm directly works on the entire input block of bits and uses an iterative process of substitution-permutation combined with a chaotic function. The algorithmic principle is simple and easy to implement. The encryption key corresponds to the set of seeds values used by the chaotic function. The same secret key is used for encryption and for decryption. The plaintext and ciphertext are bit sequences of various length. Such an algorithm encrypts any kind of binary sequence and produces highly secure ciphers. Here we apply the algorithm on images that are known to have high correlation between adjacent pixel values. The analysis, achieved on the binary corresponding sequences of the cipher-images, is based on the criteria of randomness, sensitivity and correlation. We also show that the size of the key space is large enough to resist to brute-force attacks. The paper is structured as follows. The description of the chaotic function and the encryption/decryption algorithm are given in Section 2. Section 3 presents the results and the security analysis of the proposed cryptosystem before concluding.

2. The Proposed Encryption Method

The present scheme uses an iterative process to encrypt a

Copyright © 2012 SciRes.

sequence of bits. The original input data is treated as a simple vector composed of bits "0" and "1". A chaotic function is used during the iterative process to index the positions to be permuted. Before permutation, the bits of indexed positions will be transformed via a xor operator. This process simultaneously combines the confusion and diffusion properties that are required to obtain a high security level.

2.1. The Used Chaotic Function

The algorithmic process uses a standard chaotic function given by:

$$F(X) = \lambda X(1-X), \tag{1}$$

with λ between 3.57 and 4 [6]. The chaotic behavior of such a function has been widely studied. Here, the value of λ is fixed to 3.9999 which corresponds to a highly chaotic case [7]. In the last decade, several schemes have used this function for image encryption [5,8,9]. The chaotic function is defined by the recurrence relation

$$X_{n+1} = 3.9999 X_n (1-X_n), \forall n \geq 0, \tag{2}$$

where the starting seed X_0 and all others elements X_n are real numbers belonging to the interval]0,1[. Such a chaotic function is used to compute the positions to be permuted into the input binary sequence.

2.2. Description of the Algorithm

The chaotic function given by Equation 2 is used and adapted through the iterative process. The principle of the encryption algorithm is described in four steps:

1) The original input sequence I_0 is transformed into its 1D corresponding binary vector I_0^b. This vector I_0^b is composed only of the bits "0" and "1". For example, if the initial sequence is a RGB-color (resp. gray-level) image of size $N \times M$, the size of I_0^b is $L = 3 \times 8 \times N \times M$ (resp. $L = 8 \times N \times M$).

2) A seed value X_0 is chosen in the interval]0,1[to initiate the iterative relation of Equation 2. The seed X_0 contains d decimal digits and the value of d is computed relatively to the size L of the vector I_0^b.

3) A loop is started on the binary vector I_0^b, with $0 \leq i \leq L-3$. With the current position i in I_0^b, a new position j is computed by using the chaotic function:

$$j = i + 1 + \left[\text{Floor}\left[\alpha X_{i+1}\right] \bmod U \right], \tag{3}$$

with $\alpha = 10^d$ and the value of U is initialized to $L-1$ and decremented by 1 after each iteration. Due to the modular operation, the value $\text{Floor}\left[\alpha X_{i+1}\right]$ is chosen to be larger than U. Thus, the value of d is intrinsically related to the size L of I_0^b:

$$d = \text{Floor}\left[\log_{10} L\right] + 3. \tag{4}$$

During the loop, the computed position j has always a value $i < j < L$ and the elements of I_0^b are transformed as follows:

$$Q_1 = I_0^b[i], \tag{5}$$

$$Q_2 = I_0^b[j] = I_0^b\left[i + 1 + \left[\text{Floor}\left[\alpha X_{i+1}\right] \bmod U\right]\right], \tag{6}$$

$$Q_3 = Q_1 \oplus Q_2, \tag{7}$$

$$I_0^b[i] = Q_3 \text{ and } I_0^b[j] = Q_1, \tag{8}$$

where the symbol $\oplus$ represents the exclusive OR operation bit-by-bit. That permutation process is the same until the end of the loop $(i = L-3)$. One can remark that the new positions j represent the values of the old already shuffled positions i. Therefore, the value into the vector can move several times before fixing.

4) Depending to the encoding of the original sequence, the bits of the vector I_0^b are gathered per small group to form the cipher. For RGB-color (resp. gray-level) images, the bits are gathered per group of 3×8 (resp. 8). This step permits to reconstruct the obtained cipher sequence.

That constitutes the encryption steps for one round (*i.e.* $r = 1$) on the input sequence I_0 using the seed X_0. Therefore, given an input sequence I_0 the system produces a corresponding cipher sequence I_R, where R is the total number of round used by the algorithm. The first step is done once before starting the iterative process and step 4 is used only at the end to construct the obtained cipher. The total number of round R is computed by taking into account security criteria and characteristics of the initial sequence (see Section 2.3). Each cipher I_R is produced by using a key composed of the R values of seeds $\left\{X_0^1, \cdots, X_0^R\right\}$. The **Figure 1** illustrates the main

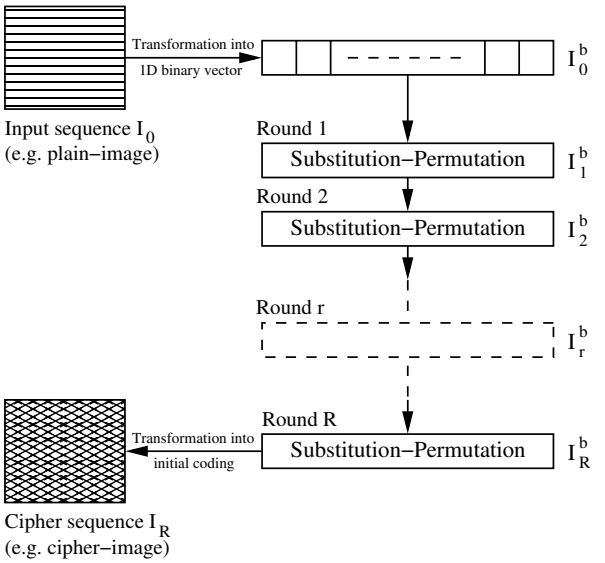


Figure 1. The main steps of the proposed encryption scheme.

Copyright © 2012 SciRes.

steps of the encryption scheme and the corresponding algorithm is given by the **Algorithm 1**. The decryption consists in reversing the process and initiating it from the last seed X_R. All positions $j = i+1+X_{i+1}$ used during the encryption must be computed and stored in a vector. A loop is done through the vector from the end to the start by making all necessary operations. The decryption algorithm is given by **Algorithm 2**.

2.3. Determination of the Round Number *R*

The number of round necessary for encryption/decryption is an important element which is connected to the security of the system. For a higher level of security it must also take into account the characteristics of the binary vector I_0^b (e.g. encoding, size and entropy). The number of round R is determined before starting the encryption process. Given today's computer speed, it is generally accepted that a key space of size smaller than 2^{128} is not secure enough. In the present case, each key is composed of R values of seeds $\left\{X_0^1, \cdots, X_0^R\right\}$. At each round r, the corresponding seed X_0^r belongs to the interval $]0,1[$, with d digits of accuracy (*i.e.* 10^{-d}). Thus, excluding the null seed $0.0\cdots0$, the total number of possibilities for the seed space at each round r is $9\times10^{d-1}$. To satisfy the condition:

$$\left(9\times10^{d-1}\right)^R > 2^{128}, \tag{9}$$

and to avoid any brute-force attack, the minimum number of rounds R_1 is given by:

Algorithm 1. Encryption algorithm.

Require: $I_0; L; R; d = \text{floor}\left[\log_{10} L\right] + 3; X_0^{1,\cdots,R}$;
Ensure: The cipher sequence $I_R\left(I_0 \to I_R\right)$.
Initialisation $\alpha = 10^d; r = 1; F = L-2; I_0^b \leftarrow I_0$;
while $r \le R$ **do**
  $i = 0; U = L-1; X = X_0^r$;
  while $i < F$ **do**
    $X \leftarrow 3.9999 \times X \times (1-X)$
    $j \leftarrow i+1+\left[\, floor\left[\alpha X\right] mod\ U\,\right]$
    $Q_1 \leftarrow I_{r-1}^b[i]$
    $Q_2 \leftarrow I_{r-1}^b[j]$
    $Q_3 \leftarrow (Q_1+Q_2) mod\ 2$
    $I_{r-1}^b[i] \leftarrow Q_3$
    $I_{r-1}^b[j] \leftarrow Q_1$
    $i \leftarrow i+1$
    $U \leftarrow U-1$
  end while
  $r \leftarrow r+1$
end while
  $I_R \leftarrow I_R^b$
return I_R

Algorithm 2. Decryption algorithm.

Require: $I_R; L; R; d = floor\left[\log_{10} L\right] + 3; X_0^{1,\cdots,R}$;
Ensure: The initial sequence $I_0\left(I_R \to I_0\right)$.
Initialisation $\alpha = 10^d; r = R; F = L-2; I_R^b \leftarrow I_R$
while $r > 0$ **do**
  $i = 0; U = L-1; X = X_0^r$;
  while $i < F$ **do**
    $X \leftarrow 3.9999 \times X \times (1-X)$
    $j \leftarrow i+1+\left[\, floor\left[\alpha X\right] mod\ U\,\right]$
    $W[i] \leftarrow j$
    $i \leftarrow i+1$
    $U \leftarrow U-1$
  end while
  $j \leftarrow F-1$
  while $j \ge 0$ **do**
    $i \leftarrow W[j]$
    $Q_1 \leftarrow I_r^b[j]$
    $Q_2 \leftarrow I_r^b[i]$
    $Q_3 \leftarrow (Q_1+Q_2) mod\ 2$
    $I_r^b[j] \leftarrow Q_2$
    $I_r^b[i] \leftarrow Q_3$
    $j \leftarrow j-1$
  end while
  $r \leftarrow r-1$
end while
  $I_0 \leftarrow I_0^b$
return I_0

$$R_1 = \text{Floor}\left[\frac{128}{\log_2\left(9\times10^{d-1}\right)}\right] + 1. \tag{10}$$

The condition $R = R_1$ assures the minimum entropy limit for the seed space but not necessarily all the required qualities for the cipher I_R. Given the algorithmic structure of the system, the number of round R_1 is not sufficient to efficiently encrypt a binary sequence with a large imbalance of bits "0" and "1". Therefore, the total number of rounds R must also be related to the distribution of bits "0" and "1" in the input binary sequence. Let consider that, in binary sequence I_0^b, the occurrence of the bit "0" has a probability $0.5 < p_0 < 1$ and $1-p_0$ for the bit "1". Then, at each new round r, the probability p_r is iteratively modified as follows:

$$p_r = \left(\left(p_{r-1}\right)^2 + \left(1-p_{r-1}\right)^2\right), \forall r \ge 1, \tag{11}$$

and the limit of the suite p_r must converge to 0.50 to ensure a balanced binary proportion in the cipher I_R. The purpose is to find the number of round R_2 satisfying the relation:

$$\lim_{r \to R_2} p_r = 0.50 - \varepsilon_1, \tag{12}$$

Copyright © 2012 SciRes.

with ε_1 a fixed numerical tolerance (here $\varepsilon_1 = 0.001$). The value of R_2 can be computed iteratively and can be large for I_0 with very low Shannon's entropy or small for I_0 with Shannon's entropy closed to its maximum (*i.e.* 1 in base 2). Such a value is computed by the **Algorithm 3**. Another problem occurs when two nearby binary sequences are encrypted using the same key, because the corresponding ciphers can be highly correlated. Thus, the number of round R must also satisfy this case for a better encryption. For that, we consider the most unfavorable case where two sequences I_0^b and $\tilde{I}_0^b$ differ by only one bit. The proportion q_0 of identical elements between these two binary suites is $q_0 = (L-1)/L$. The value of q_0 decreases according to the number of round as follows

$$q_r = q_{r-1}^2, \forall r \geq 1, \tag{13}$$

and must satisfy:

$$\lim_{r \to R_3} q_r \leq \varepsilon_2, \tag{14}$$

where ε_2 is the acceptable criterion of similarity between binary sequences (e.g. $\varepsilon_2 = 0.005$, assuming a rate of identical bits smaller than 0.5%). The value of R_3 satisfying the Equation (14) is given by

$$R_3 = \text{Floor}\left[\log_2\left(\frac{\ln(\varepsilon_2)}{\ln(q_0)}\right)\right] + 1. \tag{15}$$

Such a value of R_3 ensures to obtain two completely different ciphers. It mainly allows to resist to the differential attack or generally the chosen plaintext attack. Therefore, the relevant number of round R of the encryption algorithm is given by:

$$R = \max\{R_1, R_2, R_3\}. \tag{16}$$

The final number of rounds R permits to satisfy simultaneously the criteria of key entropy, maximum Shannon's entropy and sensitivity to initial conditions (sequence and key).

2.4. Bits Propagation Analysis

In this section, we show the evolution of bits propagation into the cipher during the encryption process. To illustrate the bit propagation, we use a particular initial vector I_0 corresponding to a gray-level image composed by only black (0) and white (255) pixels. The image size is 350×350 and the white pixels are gathered in the center of the image as a small white circle. The number of black pixels is 122276 (*i.e.* 99.81%) and white pixels is (*i.e.* 0.19%) (see **Figure 2(a)**). The values of R_1, R_2 and R_3 are 5, 11 and 23 respectively. The encryption round number is $R = 23$. The plain-image is encrypted and the intermediate cipher-images obtained after r = 1, 2, 5, 7, 9, 11, 20 and 23 are shown in **Figures 2(b)-(i)**. We note that, after $r = R_1 = 5$ rounds, the small white circle is vanishing. For $r = 9$, the intermediate cipher-image starts to look like a true random image and from $r = R_2 = 11$ it passes successfully the NIST tests [10]. One can remark that the randomness propagation is gradual before stabilizing and give a true random image. The ciphers obtained by this encryption scheme have all the same appearance and the same randomness quality, whatever the particularity of original binary sequence. The cipher quality does not depend on the structure of the input sequence. Additional analysis is achieved by computing the correlation coefficients [5], NPCR (Number of Changing Pixel Rate) and UACI (Unified Averaged Changed Intensity) [11], of intermediate cipher-

Algorithm 3. Computation of the round number R_2.

Require: p_0 (proportion of bit '0'); $\varepsilon_1 = 0.001$;
Ensure: The round number R_2.
1. $R_2 = 0; p = p_0; dif = |0.50 - p|$;
2. **while** $dif > \varepsilon_1$ **do**
3. $p \leftarrow \left[p^2 + (1-p)^2\right]$
4. $dif \leftarrow |0.50 - p|$
5. $R_2 \leftarrow R_2 + 1$
6. **end while**
7. **return** R_2

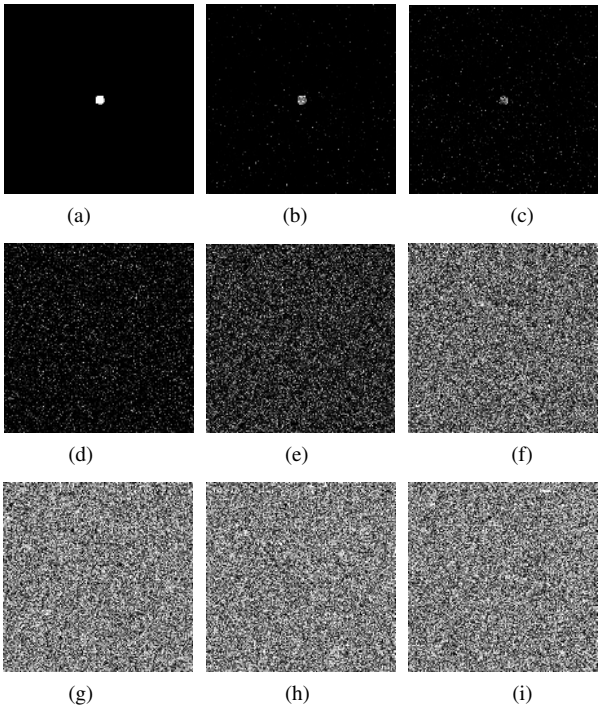


Figure 2. Intermediate cipher-images: (a) The initial unbalanced plain-image; (b, c, d, e, f, g, h and i) The cipher-images obtained after r = 1, 2, 5, 7, 9, 11, 20 and 23 rounds, respectively. For each round, the seed value is chosen arbitrarily in the interval $]0,1[$ with $d = 8$ decimal digits.

Copyright © 2012 SciRes.

images between two consecutive rounds. For two uncorrelated sequences, the correlation coefficient is $\simeq 0$. A strong correlation occurs between two sequences for a coefficient value $\simeq \pm 1$. The evolution of coefficient values is represented at **Figure 3(a)**. One can also remark that, the NPCR and UACI indicator values increase with r (see **Figure 3(b)**). For $r = R_2 = 11$, the values appear to be stabilized for the three indicators. That confirms that from $r = R_2$, the intermediate cipher-images are different and the correlation coefficients, NPCR and UACI values become stable for $r = R_a = \max(R_1, R_2)$. Taking into account R_3 allows to include the plain-image sensitivity and therefore to obtain a better security level of the system. These results show the relevance of the iterative process and the choice of the round number of the encryption algorithm.

2.5. Key Space

An efficient encryption scheme should have a large key space in order to make brute-force attacks infeasible. It is generally accepted that a key space of size smaller than 2^{128} is not secure enough. In the present case, the determination of round number R allows to satisfy the required entropy of the size of key space. For example, for a bits sequence of length $L = 1033848$ $(d = 9, R = 23)$, the algorithm enables to produce exactly $(9 \times 10^8)^{23} \approx 2^{684}$ different cipher sequences. Therefore, the proposed scheme is not vulnerable to brute-force attacks.

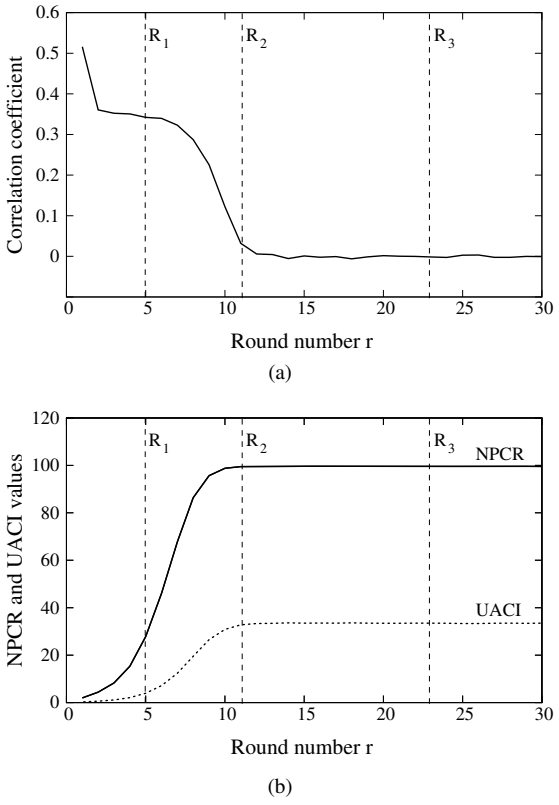


Figure 3. Evolution of coefficients values: (a) The correlation coefficients and (b) The NPCR (in %) and UACI (in %), obtained between intermediate cipher-images of two consecutive rounds.

3. Results and Security Analysis

The cryptosystem is applied on the image encryption where the original image is treated as a sequence of bits. Generally, the adjacent pixels in an image are highly correlated therefore the pixel values are very close or identical throughout the image. The statistical analysis of the encryption scheme is based on the three fundamental properties that are: indistinguishability, confusion and diffusion [12]. Indistinguishability means that the produced ciphers should have a high level of randomness and not to be differentiated from the outputs of a truly random function. The property of confusion means that the plain-image and the cipher-image should be completely decorrelated. By diffusion, the cryptosystem should be very sensitive to the initial condition (*i.e.* a variation of at least one bit in the key or plain-image leads to strong different cipher-images). Therefore, the security of the system is analysed through these three properties.

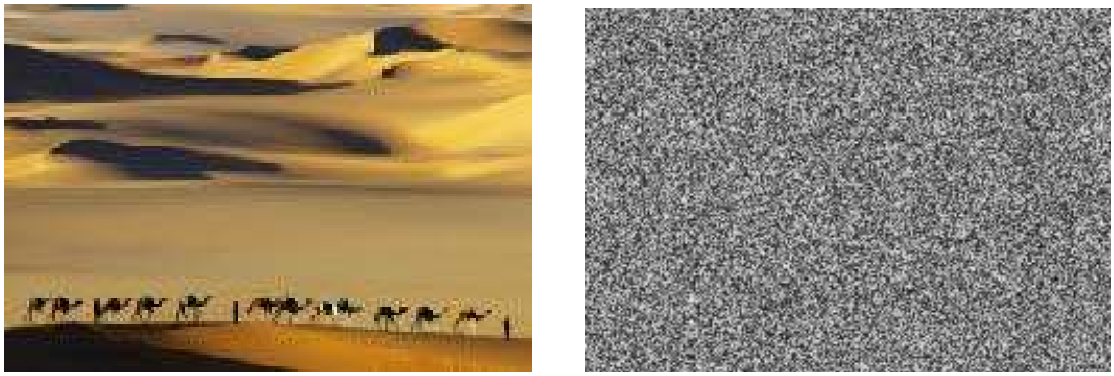
3.1. The Used Images for Analysis

Two image formats are used for the analysis: a RGB-color image and a gray-level image. The RGB-color image, noted I_D^a, has 173 rows and 249 columns (see **Figure 4(a)**). Its binary length is $L = 1033848$, the computed round number R is equal to 23 and the number of decimal digits is $d = 9$. An example of encryption with the key $K_D^a = \{X_0^1, \cdots, X_0^{23}\}$ given in the **Table 1** is presented at the **Figure 4(b)**. The frequency histograms for the red, green and blue channels before and after encryption of the RGB-color image are presented in **Figure 5**. The second image I_C^a of size 361×500 $(L = 1444000)$ encoded in gray-level is presented in **Figure 6(a)**. The round number and the significant digits are $R = 23$ and $d = 9$. An example of cipher-image with the key K_C^a (here $K_C^a = K_D^a$) is given in **Figure 6(b)**. The **Figures 6(c)-(d)** show the frequency histograms of intensity levels before and after encryption of the image I_C^a. The obtained histogram after encryption has a balanced distribution of frequencies.

3.2. Key Sensitivity Analysis

A good encryption system must be highly sensitive to the encryption key. A difference of a single bit into the key must provide a very different encryption/decryption result. In that case, the key is given by R values of arbi

Copyright © 2012 SciRes.



(a) (b)

Figure 4. Example of RGB-color image encryption: (a) The plain-image of a caravan in the desert and (b) Its corresponding cipher-image using the key K_D^a (given in Table 1).

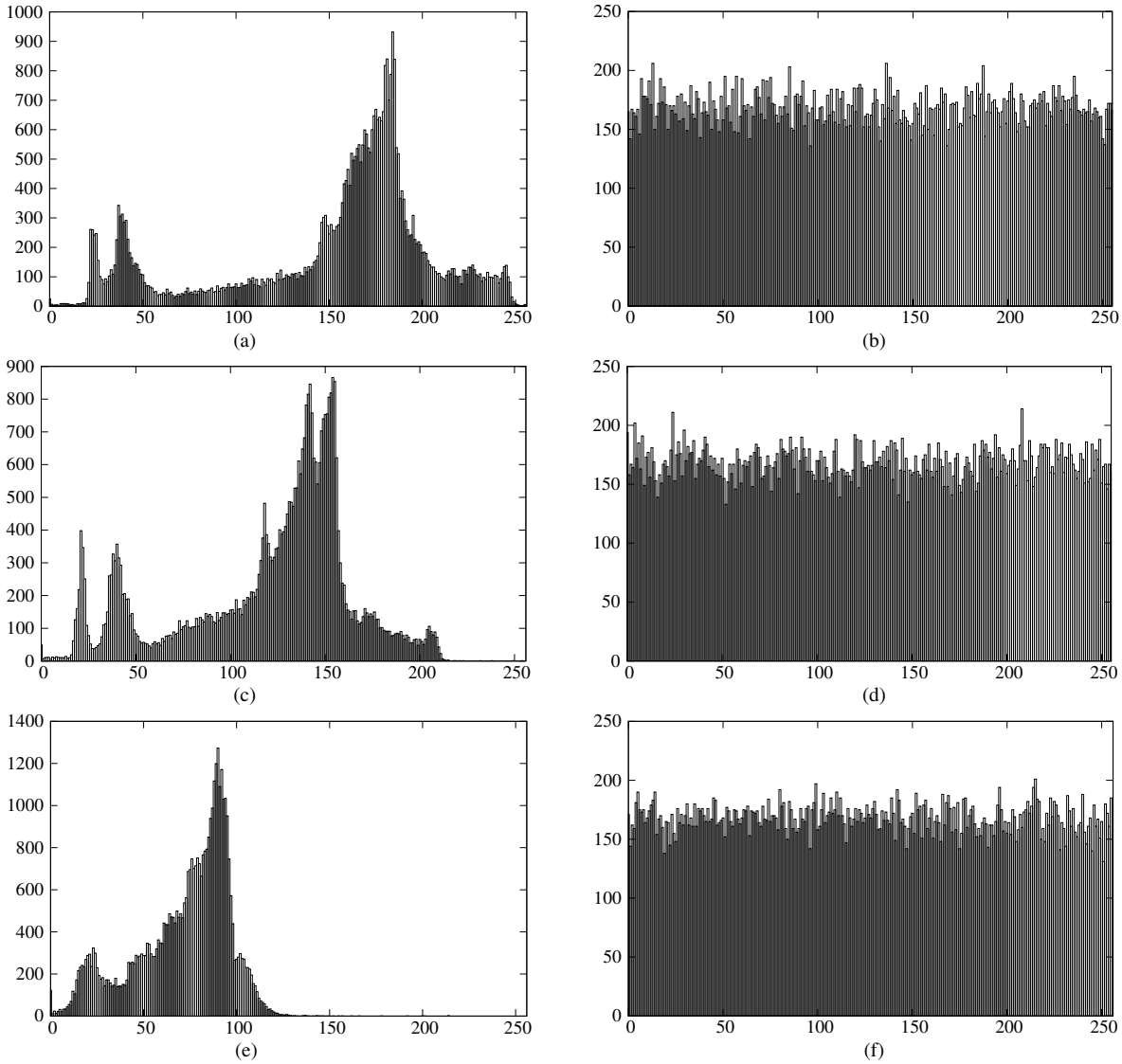


Figure 5. RGB channel distributions: (a), (c) and (e) Show the frequency distributions before encryption for the red, green and blue channels, respectively; (b), (d and (f) Show the associated histograms of the cipher-image, after encryption using the key K_D^a.

Copyright © 2012 SciRes.

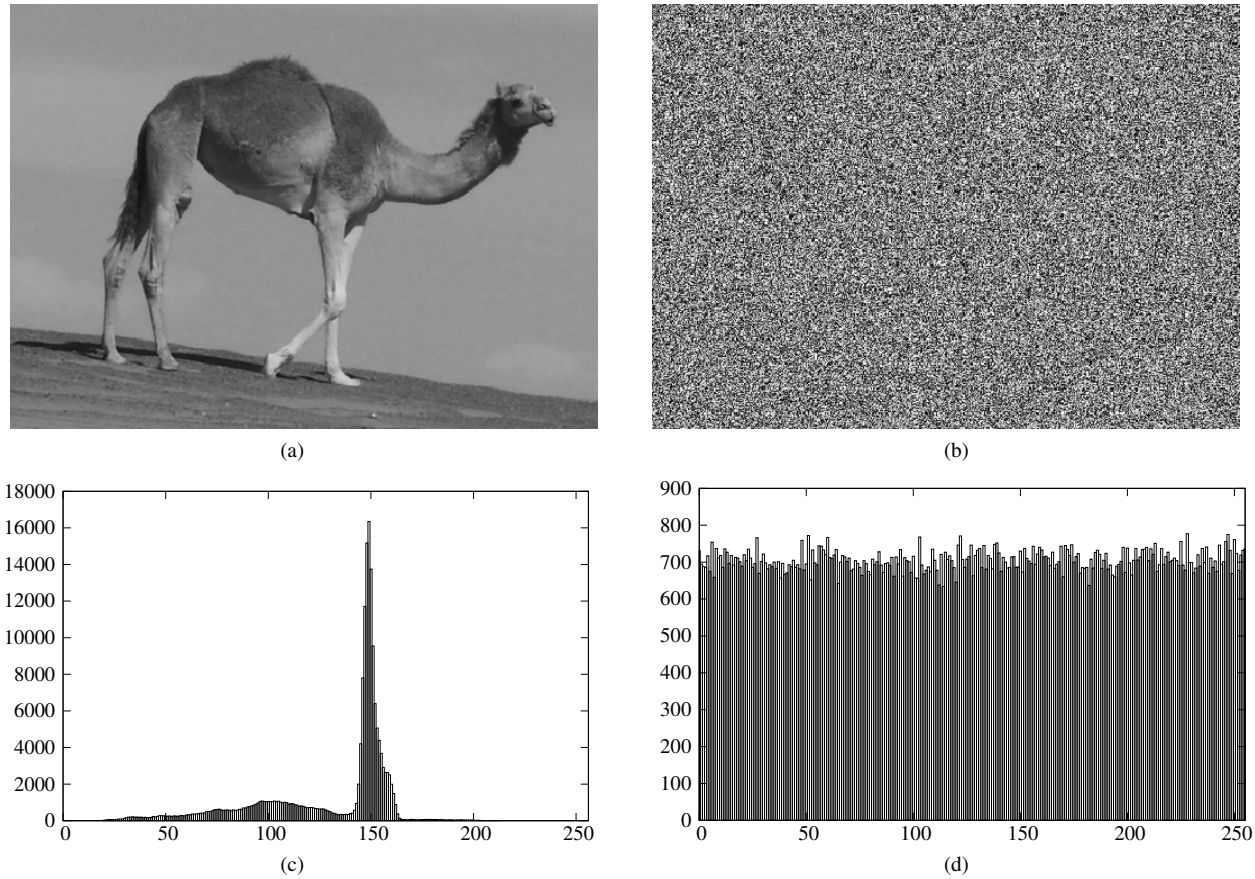


Figure 6. The gray-level dromedary's image and frequency histograms: (a) The original image, (b) The corresponding cipher-image with key K_D^a. Frequency histograms for (c) The original image and (d) The cipher-image.

Table 1. The 23 values of seeds $X_0^1,\cdots,X_0^{23}$ constituting the key K_D^a.

Seeds	Values ($d=9$)		
X_0^1, X_0^2, X_0^3	0.372517362	0.073559321	0.875371003
X_0^4, X_0^5, X_0^6	0.034203719	0.984941322	0.403519327
X_0^7, X_0^8, X_0^9	0.487302137	0.392510603	0.302543081
$X_0^{10}, X_0^{11}, X_0^{12}$	0.812174032	0.665371032	0.004723821
$X_0^{13}, X_0^{14}, X_0^{15}$	0.950317340	0.910451945	0.740255912
$X_0^{16}, X_0^{17}, X_0^{18}$	0.482664019	0.103619439	0.308598253
$X_0^{19}, X_0^{20}, X_0^{21}$	0.018036430	0.804104810	0.282906931
X_0^{22}, X_0^{23}	0.296051483	0.571410332	

trarily chosen seeds in the interval $]0,1[$. The sensitivity study focuses on a variation of the last seed value X_0^R. In the following, we consider the RGB-color image (**Figure 4(a)**) and the gray-level image (**Figure 6(a)**). Each image is encrypted by using the seed values given in **Table 1**. On the last seed, a loop is done from 0.571410332 to 0.571411131 incrementing by 10^{-9}. The variation on the last seed value (*i.e.* X_0^{23}) produces 800 keys. The 800 cipher-images obtained from the 800 generated keys are analysed.

3.2.1. Randomness Analysis

The randomness level of the 800 ciphers is analysed through the NIST tests. Such a suite consists in a statistical package of fifteen tests developed to quantify and to evaluate the randomness of binary sequences produced by cryptographic random or pseudorandom number generators [10]. For each statistical test, a p_{value} probability is computed. A p_{value} of zero indicates that, the sequence appears to be completely non-random. A p_{value} larger than 0.01 means that, the sequence is considered to be random with a confidence level of 99%. For multiple tested sequences at the same time, each test defines a proportion η as the ratio of sequences passing successfully the test relatively to the total number of tested sequences N_k $\left(i.e.\ \eta = n\left[p_{\text{value}} \geq 0.01\right]/N_k\right)$. The proportion η is compared to an acceptable proportion η_{accept} which corresponds to the ratio of sequences that should

Copyright © 2012 SciRes.

pass the test. The range of acceptable proportions, excepted for the tests *Random Excursion-(Variant)* is determined by using the confidence interval defined as $(1-0.01)\pm 3\sqrt{0.01(1-0.01)/N_k}$ [10]. For $N_k = 800$, the acceptable proportion should lie above 97.94%. Each binary sequence is individually tested and the percentage of success η is given for each statistical test. The **Table 2** presents the results of tests for the ciphers obtained from the two plain-images I_D^a and I_C^a. The tested binary sequences pass successfully the NIST tests. Therefore, these sequences can be considered as random sequences. This analysis checks the properties of indistinguishability and confusion.

3.2.2. Correlation Analysis

For each case (RGB and gray-level image), the correlation is analysed globally by computing the correlation coefficients between each pair of produced ciphers. The distribution of the coefficient values are presented by the histograms given by the **Figure 7**. All the coefficient values belong to the interval $[-0.01, 0.01]$. For the RGB-color image, 99.56% of the coefficients have an absolute value smaller than 0.008. In the case of gray-level image, 99.24% of the coefficients have an absolute value smaller than 0.0065. The results show that the cipher-images have a very low correlation. This analysis shows that a small difference between the keys can be propagated and give completely different ciphers (diffusion property).

Additional analysis is achieved by calculating the NPCR and UACI between each pair of the 800 cipher-images. The average and the standard deviation values for the correlation coefficients, NPCR and UACI are computed and presented in the **Table 3**. The obtained values show that the tested ciphers are totally different. Such an analysis confirms the sensitivity of the encryption system in relation with the encryption key.

Table 2. Results of NIST tests on the 800 cipher-images from RGB-color and gray-level images. For each test, the ratio η (in %) of p_{value} is given.

Test	Cipher-Images of Desert		Cipher-Images of Dromedary	
	η in %	Result	η in %	Result
Frequency (Monobit)	98.50	Pass	99.25	Pass
Block-Frequency	99.12	Pass	98.62	Pass
Cumulative Sums (1)	98.50	Pass	99.12	Pass
Cumulative Sums (2)	98.62	Pass	99.12	Pass
Runs	98.62	Pass	99.25	Pass
Longest Run	99.62	Pass	99.25	Pass
Rank	99.12	Pass	99.25	Pass
FFT	99.12	Pass	98.62	Pass
Non-Overlapping	99.00	Pass	98.50	Pass
Overlapping	98.87	Pass	99.25	Pass
Universal	98.62	Pass	99.00	Pass
Approximate Entropy	98.12	Pass	98.75	Pass
Random Excursions	98.45	Pass	98.53	Pass
Random E-Variant	98.45	Pass	98.53	Pass
Serial (1)	99.37	Pass	98.75	Pass
Serial (2)	99.00	Pass	99.37	Pass
Linear Complexity	99.00	Pass	98.75	Pass

3.3. Plain-Image Sensitivity Analysis

A second important analysis concerns the sensitivity related to the original image. Therefore, we produce multiple nearby images by introducing a small variation of one bit in the original image. All these images are encrypted by using the same key and the corresponding ciphers are

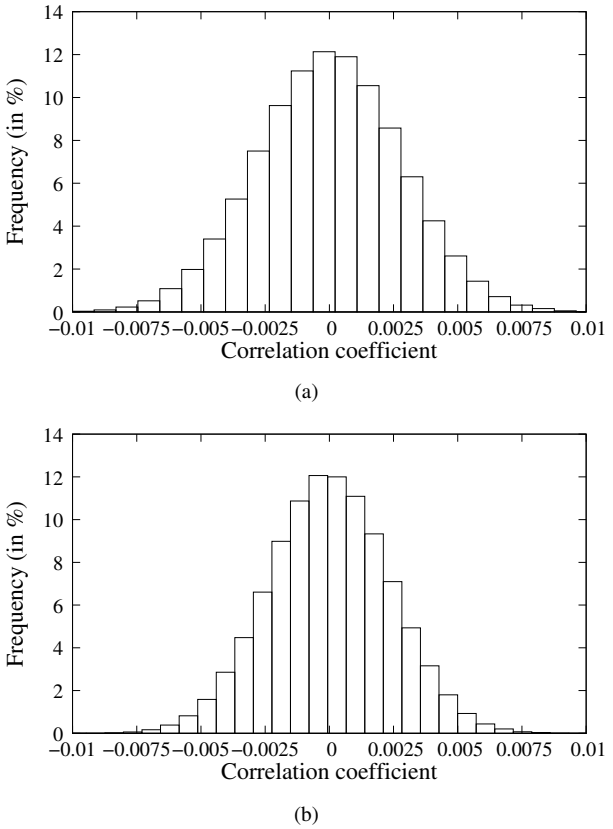


Figure 7. Distribution of the correlation coefficient values between each pair of the 800 cipher-images of (a) The RGB-color image and (b) The gray-level image.

Copyright © 2012 SciRes.

Table 3. Average ϕ and standard deviation σ values of the correlation coefficients, NPCR (in %) and UACI (in %) between the 800 cipher-images of the RGB-color and the gray-level images.

Indicator	RGB		Gray-Level	
	ϕ	σ	ϕ	σ
Coef corr.	0.0022	0.0016	0.0018	0.0013
NPCR	99.6093	0.0172	99.6095	0.0147
UACI	33.4637	0.0660	33.4470	0.0550

analysed. For the RGB-color image, the first pixel (0, 0) of the image at upper left is encoded by the values [94,148,179] (*i.e*. blue, green, red). The value of the blue component (94) is incremented by 1 (*i.e*. from 94 to 233) to form 140 consecutive images. Obviously, the difference between the 140 produced images can not be visually distinguished. These images are encrypted using the same key K_D^a (given in **Table 1**). The same approach is applied to the gray-level image for which the value of the first pixel (147) is decremented by 1 (*i.e*. from 147 to 8) to produce 140 new gray-level images. These images are encrypted with the key K_C^a. For each case, the 140 cipher-images are constructed from the original image by using the same set of seed values.

3.3.1. Randomness Analysis

The NIST tests are used to evaluate the randomness level of the 140 ciphers. With such number of ciphers, the ratio η must overpass $\eta_{\text{accept}} = 96.47\%$. The results of the NIST tests are presented in the **Table 4**. The results show that the tested sequences have a good level of randomness. This analysis checks the properties of indistinguishability and confusion.

3.3.2. Correlation Analysis

The correlation coefficient is calculated between each pair of the 140 cipher-images. The **Figure 8** presents the two histograms of correlation coefficients for values belonging to the interval $[-0.01, 0.01]$.

For the 140 cipher-images of the RGB-color image, 99.02% of the coefficients have absolute values smaller than 0.0072 and for the gray-level images 99.19% of the coefficients have absolute values smaller than 0.0065. These histograms illustrate the small correlation between the tested cipher-images. Such a quality is critical to resist to the chosen plaintext attack. The correlation is also evaluated via the NPCR and UACI indicators. The average and the standard deviation values for the correlation coefficients, NPCR and UACI are computed and presented in the **Table 5**. The results show that the tested ciphers are completely different. A difference of only one bit in the original binary sequence produces different cipher.

Table 4. Results of NIST tests on the 140 cipher-images produced from 140 plain-images of the RGB-color and gray-level images. The images are encrypted with the same key and the ratio η (in %) of p_{value} is given for each statistical test.

Test	Cipher-Images in RGB		Cipher-Images in Gray-Level	
	η in %	Result	η in %	Result
Frequency (Monobit)	100.00	Pass	98.57	Pass
Block-Frequency	99.28	Pass	100.00	Pass
Cumulative Sums (1)	100.00	Pass	98.57	Pass
Cumulative Sums (2)	100.00	Pass	99.28	Pass
Runs	97.85	Pass	100.00	Pass
Longest Run	100.00	Pass	99.28	Pass
Rank	98.57	Pass	97.14	Pass
FFT	99.28	Pass	100.00	Pass
Non-Overlapping	97.85	Pass	97.85	Pass
Overlapping	99.28	Pass	98.57	Pass
Universal	97.85	Pass	97.85	Pass
Approximate Entropy	97.14	Pass	96.62	Pass
Random Excursions	97.67	Pass	97.64	Pass
Random E-Variant	97.67	Pass	97.64	Pass
Serial (1)	98.57	Pass	98.57	Pass
Serial (2)	98.57	Pass	99.28	Pass
Linear Complexity	100.00	Pass	99.28	Pass

3.4. Efficiency Analysis

Efficiency is also an important indicator for a good encryption algorithm. **Table 6** lists the execution time and the key space entropy for the encryption process for different sizes of gray-level images. A comparison with the results of other algorithms is also presented. The used computer is an Intel Core Duo T2300 (1.66 GHz) Processor with 2.5 Go memory under Fedora 14 (Laughlin). One can see that the size of the key space is larger for the proposed algorithm, assuring a secure encryption. Sensitivity to the plain-image should not be neglected because it is part of the assumptions to be verified for an encryption algorithm. If such an hypothesis is not respected, the algorithm can be broken by the chosen plaintext attack. In order to compare the time efficiency with reference algorithms, we also consider the two cases with $R_a = \max(R_1, R_2)$ and $R = \max(R_1, R_2, R_3)$. The speed time of the R_a version of the algorithm (not taking into account the sensitivity to the initial sequence) is achieved

Copyright © 2012 SciRes.

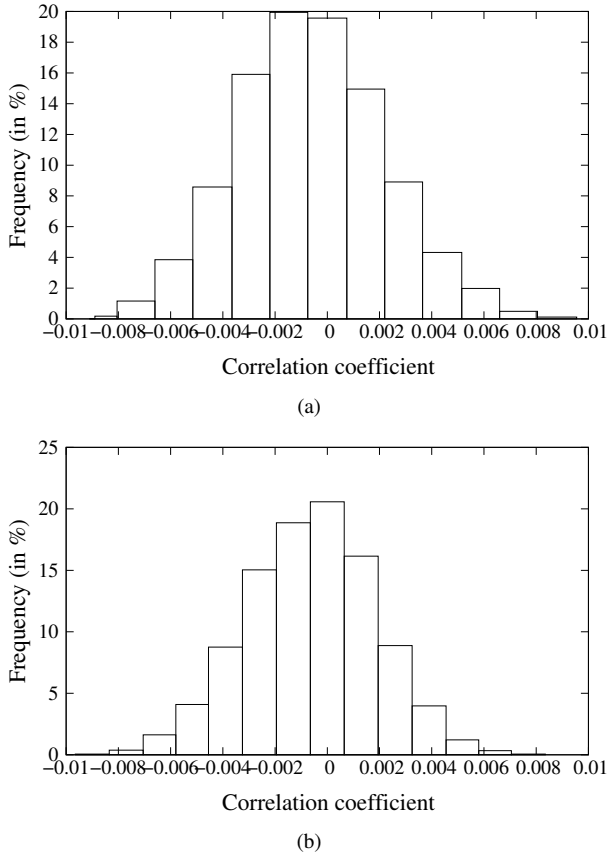


Figure 8. Histogram of correlation coefficient values between the 140 cipher-images obtained from (a) The RGB-color and (b) The gray-level images.

Table 5. Average ϕ and standard deviation σ values of the correlation coefficients, NPCR (in %) and UACI (in %) between the 140 cipher-images of the RGB-color and the gray-level images.

Indicator	RGB		Gray-Level	
	ϕ	σ	ϕ	σ
Coef corr.	0.0021	0.0016	0.0018	0.0014
NPCR	99.6108	0.0164	99.6102	0.0138
UACI	33.4676	0.0675	33.4520	0.0528

in order to compare with reference algorithms which use the correlation, NPCR and UACI indicators. An alternative way to improve the encryption speed is to reduce the value of ε_2 in Equation (15) which determines the number of round R_3.

4. Conclusion

A new symmetric encryption algorithm based on binary permutations was presented. The algorithm uses an iterative process of substitution-permutation combined with a chaotic function. The principle of such a process is to drastically disrupt the internal binary structure of the sequences and progressively induce randomness characteristics. The key space is large enough to resist brute-force attacks. The application to image encryption show that the cryptosystem is very sensitive to key and plain-image. The advantage of such an encryption algorithm is its ability to securely encrypt any kind of binary sequence. Such a cryptosystem can be used for secure storage or transmission of sensible binary data.

Table 6. Encryption time in seconds and key space entropy in bits for the proposed and for reference algorithms. For the proposed algorithm, two cases are considered: $R_a = \max(R_1, R_2)$ and $R = \max(R_1, R_2, R_3)$.

Image Size	Proposed Algorithm (R_a/R)		Ref. Algo. [13,14]	
(8 bits/pixel)	Time	Entropy	Time	Entropy
256×256	0.55/2.25	132/654	6.01 [13]	84
			1.01 [14]	113
512×512	2.45/11.10	148/713	36.53 [13]	84
			4.73 [14]	113
1024×1024	11.66/58.89	148/773	253.87 [13]	84
			19.78 [14]	113

REFERENCES

[1] R. L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communication ACM*, Vol. 21, No. 2, 1978, pp. 120-126. doi:10.1145/359340.359342

[2] B. Schneier and P. Sutherland, "Applied Cryptography: Protocols, Algorithms, and Source Code in C," John Wiley & Sons, Inc., Hoboken, 1995.

[3] NIST, "Announcing the Advanced Encryption Standard (AES)," 2001. http://csrc.nist.gov/CryptoToolkit/aes/

[4] C. C. Chang, M. S. Hwang and T. S. Chen, "A New Encryption Algorithm for Image Cryptosystems," *Journal of Systems and Software*, Vol. 58, No. 2, 2001, pp. 83-91. doi:10.1016/S0164-1212(01)00029-2

[5] G. Chen, Y. Mao and C. K. Chui, "A Symmetric Image Encryption Scheme Based on 3D Chaotic Cat Maps," *Chaos Solitons and Fractals*, Vol. 21, No. 3, 2004, pp. 749-761. doi:10.1016/j.chaos.2003.12.022

[6] Z. Zhu, W. Zhang, K. Wong and H. Yu, "A Chaos-Based Symmetric Image Encryption Scheme Using a Bit-Level Permutation," *Information Sciences*, Vol. 181, No. 6, 2011, pp. 1171-1186. doi:10.1016/j.ins.2010.11.009

[7] N. K. Pareek, V. Patidar and K. K Sud, "Image Encryption Using Chaotic Logistic Map, Image and Vision Computing," Vol. 24, No. 9, 2006, pp. 926-934. doi:10.1016/j.imavis.2006.02.021

Copyright © 2012 SciRes.

[8] Y. Wang, K. W. Wong, X. Liao, T. Xiang and G. Chen, "A Chaos-Based Image Encryption Algorithm with Variable Control Parameters," *Chaos Solitons and Fractals*, Vol. 41, No. 4, 2009, pp. 1773-1783. doi:10.1016/j.chaos.2008.07.031

[9] A. K. Acharya, "Image Encryption Using a New Chaos Based Encryption Algorithm," *Proceedings of the International Conference on Communication*, *Computing and Security*, New York, 12-14 February 2011, pp. 577-581. doi:10.1145/1947940.1948060

[10] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo, "Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," Special Publication 800-22 Revision 1a, National Institute of Standards and Technology, 2010.

[11] Y. Wu, J. Noonan and S. Agaian, "Npcr and Uaci Randomness Tests for Image Encryption," *Cyber Journals*: *Multidisciplinary*, *Journals in Science and Technology*, *Journal of Selected Areas in Telecommunications* (*JSAT*), 2011, pp. 31-38.

[12] G. Álvarez and S. Li, "Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems," *International Journal of Bifurcation and Chaos*, Vol. 16, No. 8, 2006, pp. 2129-2151.

[13] T. Gao and Z. Chen, "Image Encryption Based on a New Total Shuffling Algorithm," *Chaos*, *Solitons and Fractals*, Vol. 38, No. 1, 2008, pp. 213-220. doi:10.1016/j.chaos.2006.11.009

[14] X. Wang and J. Zhang, "An Image Scrambling Encryption Using Chaos-Controlled Poker Shuffle Operation," *Proceedings of International Symposium on Biometrics and Security Technologies*, Islamabad, 23-24 April 2008, pp. 1-6.

Copyright © 2012 SciRes.