Signal Processing: *Image Communication* 27 (2012) 249–259

Contents lists available at SciVerse ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

A new image encryption scheme based on a chaotic function

M. François [a], T. Grosges [a,*], D. Barchiesi [a], R. Erra [b]

[a] *Group for Automatic Mesh Generation and Advanced Methods (Gamma3 UTT-INRIA), University of Technology of Troyes, 12 rue Marie Curie - BP 2060 - Troyes Cedex 10010, France*

[b] *Department of Network & Information Security, Ecole Supérieure d'Informatique, Electronique, Automatique (ESIEA), 9 rue Vésale, Paris 75005, France*

ARTICLE INFO

Article history:
Received 29 July 2011
Accepted 24 November 2011
Available online 2 December 2011

Keywords:
Image encryption
Chaotic function
Statistical analysis
Confusion
Diffusion
Indistinguishability

ABSTRACT

In recent years, several methods of secure image encryption were studied and developed through chaotic processes or functions. In this paper, a new image encryption scheme based on a coupling of chaotic function and xor operator is presented. The main advantages of such a method are the abilities to produce a large key space to resist brute-force attacks, and to encrypt securely images with any entropy structure assuring indistinguishability, confusion and diffusion properties in the corresponding cipher-images. The results of several statistical analysis about randomness, sensitivity and correlation of the cipher-images show that the proposed cryptosystem is efficient and secure enough to be used for the image encryption and transmission. Moreover, the implementation of the corresponding algorithm is easy and only integers are used.

1. Introduction

The rapid growth of technology concerns all scientific research fields including the processing and transmission of digital images. In many fields like military, medical, industry, multimedia, communication or even personal, million of images are stored or transmitted through internet every day. Depending on the application domain, the need to protect these images against unauthorized users has become a challenge. Consequently, during the last years, several image encryption schemes have been proposed. Such encryption schemes are based on scan patterns methodology [1], double random phase encoding [2], iterative random encoding and gyrator transformation [3], vector quantization [4], quadtree compression [5,6] and chaos maps with total shuffling [7] or Kolmogorov flow [8]. Due to the intrinsic characteristics of chaotic systems, the use of chaos-based cryptographic schemes seems to be an appropriate response for secure image encryption. Indeed, the algorithms based on chaos offer the advantages to be very sensitive to the initial conditions and to satisfy a good combination of speed, confusion, diffusion and complexity [9,10]. During the encryption process, some of such algorithms use only one-dimensional chaotic map [11,12]. To increase the complexity in the cipher-algorithm, two-dimensional or high-dimensional chaotic maps are also used [13–17]. In the context of image encryption, due to the strong correlation between adjacent pixels of the images [7,16], the application of only permutations in the encryption process does not guarantee a good level of security [18]. For a best encryption, the majority of methods propose to mix and to change the values of the pixels simultaneously. Nevertheless, to assure an efficient encryption scheme, some conditions should be fulfilled such as a large key space, randomness of the cipher-image and a high sensitivity on the initial conditions (seed and plain-image). A large key space is necessary to resist brute-force attacks [19,20] and a secure encrypted image corresponds to an image that cannot be statistically distinguished from a truly random sequence. Indeed, the cipher-images should present a good level of randomness [9,10]. Moreover, the cipher-image should be very sensitive to the used initial key or seeds and to the plain-image [10,21].

In this paper we propose a new encryption/decryption algorithm based on a chaotic function using linear congruences. Such a function is coupled with a xor operation during the encryption process to increase the

* Corresponding author. Tel.: +33 3 25 71 84 30; fax: +33 3 25 71 56 49.
E-mail address: thomas.grosges@utt.fr (T. Grosges).

0923-5965/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2011.11.003

unpredictability in the cipher-image as well as a large key space to resist to attacks. Statistical analysis are realized on the binary sequences of the ciphers to evaluate their cryptographic qualities. This paper is structured as follows. The description of the chaotic function and the encryption/decryption method as well as the technical details are given in Section 2. Section 3 presents the security analysis of the proposed method, before concluding.

2. The proposed encryption method

Generally, the adjacent pixels in an image are strongly correlated. Therefore, to increase the quality of the cipher-image, the encryption process is directly achieved on the bits of the plain-image. Let us consider a plain-image I_0 of dimension $N \times M$ (i.e. N rows and M columns). The image I_0 is shuffled by using a chaotic function based on linear congruences. This function is inspired by recurrences used for pseudo-random numbers generation [22]. Such a function is used to compute the positions that will be shuffled in the encryption algorithm. The function uses a degressive modulo, related to the size of the input vector, and is defined by the recurrence relation

$$X_{n+1} = [[X_n^2 \bmod S] \times X_n + X_g] \bmod S \quad (1)$$

with the initial position $X_0 = g$ and $X_g = g^2$, the seed g in $\{1, \ldots, L\}$ and L being the binary size of the image I_0 (e.g. for a 256 gray-level image, its binary size is $L = 8 \times N \times M$ and $L = 3 \times 8 \times N \times M$ for a RGB-color image). To avoid any problem of periodicity, the space of seed values g is limited to L. The value S is initialized to $L-1$ and decremented after each iteration. The positions are computed with a memory effect (i.e. an index i peruses the input size L and a permutation is done between each new computed position X_n and i). Therefore, the value of a position can be permuted several times before fixing. The principle of the function is to shuffle the starting positions $1, \ldots, L$ by considering an initial seed X_0. In the following, the chaotic behaviour of the function is characterized before describing the algorithm of encryption.

2.1. Chaotic behaviour of the function of recurrence

We analyse the chaotic behaviour of the function given by Eq. (1) for various fixed parameters. For dynamical systems, the Lyapunov exponent characterizes the velocity of evolution between two near trajectories and is given, for dynamical system, by [23,24]

$$\lambda(x_0) = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \ln |f'(x_i)| \quad (2)$$

with the iterative function $x_{n+1} = f(x_n)$ and two near initial conditions x_0 and $x_0 + \epsilon$. A positive value of Lyapunov exponent λ represents a quantitative measurement of the

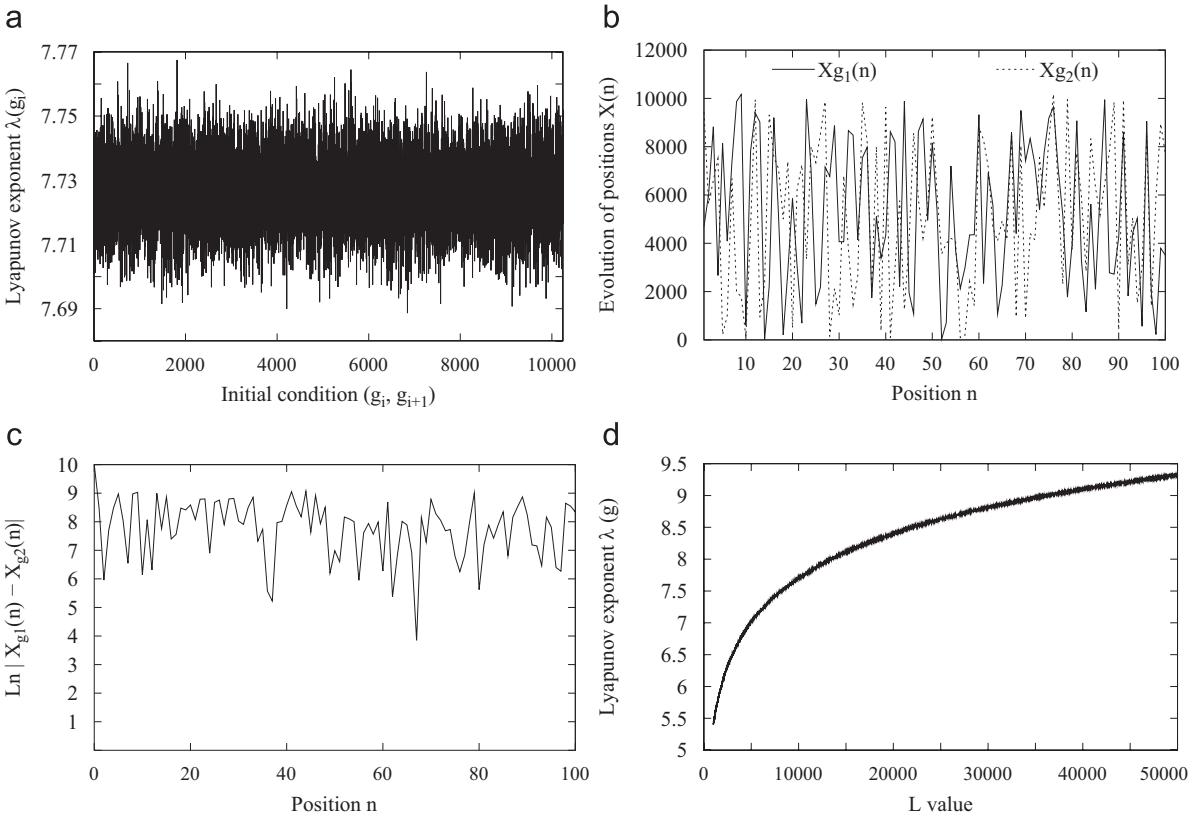

Fig. 1. Chaotic behaviour of the function of Eq. (1): (a) Lyapunov exponent between generated suites for two consecutive seeds from 1 to $L = 10\,240$, (b) sensitivity on the initial conditions for 100 iterations, (c) log difference between two sequences generated by the seeds $g_1 = 4713$, $g_2 = 4714$ for 100 iterations, and (d) Lyapunov exponent as function of L values for a fixed seed value $g = 954$.

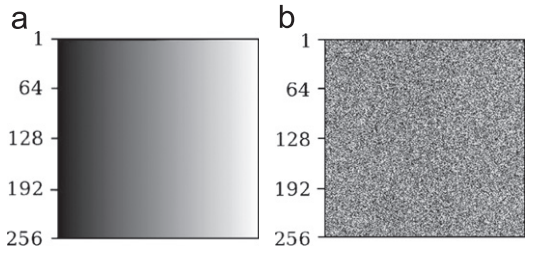

Fig. 2. Transformations and permutations of the matrix in gray-level image: (a) each line is an ordinate sequence of increasing values from 0 to 255 and (b) permutation of each line using Eq. (1) with the initial seed $g_i = i$ (i.e. i being the line number).

chaotic behaviour of the function. In the present case, the initial conditions are corresponding to the seed values and the Lyapunov exponents between the suites generated for two consecutive seeds are computed. As the function of Eq. (1) is not differentiable, the value of λ is given by

$$\lambda(g) = \frac{1}{L}\sum_{i=1}^{L} \ln\left|\frac{V_g[i] - V_{g'}[i]}{g - g'}\right|, \tag{3}$$

where V_g and $V_{g'}$ correspond to position vectors generated by the seeds g and g', respectively. With the seed values $g_i = i$, where $1 \le i \le L$ and $L = 10\,240$, the Lyapunov exponents $\lambda(g_i)$ are computed for two near seed values (g_{i-1} and g_i). The sensitivity to the initial conditions is depicted in Fig. 1(a). All the corresponding Lyapunov exponents are positive and belong to the interval [7.69, 7.77]. Fig. 1(b) shows an example of the evolution of the positions, for two closed seeds $g_1 = 4713$ and $g_2 = 4714$, as function of the position n. It clearly appears that the two trajectories (position evolutions) are very sensitive to the initial conditions (i.e. seed values g_i). For these two trajectories, we present the $\log|X_n - X'_n|$ as function of the position n (see Fig. 1(c)). Finally, for a seed value $g = 954$, the evolution of the Lyapunov exponent values as function of L is computed and presented in Fig. 1(d). The exponents are positive and show the chaotic behaviour of the function in Eq. (1). To illustrate the behaviour of the recurrence function of Eq. (1) as function of the seed values g, we construct a 256×256 matrix. Each line of the matrix is an ordinate sequence of values 0–255 (as pixel intensity value in gray-level, see Fig. 2(a)). Fig. 2(b) presents all the new permuted positions with the seeds $g_1 = 1$ to $g_{256} = 256$. This shows, as an image in gray-level, the direct influence of the function of Eq. (1) on the position values after application of permutations with the seeds $g_i = i$. This illustrates the sensitivity to the initial condition (seed value). Let us turn to the whole algorithm of encryption and decryption.

2.2. Description of the algorithm

Such a chaotic function is now integrated in the encryption algorithm. The detailed description of the encryption algorithm is composed of four steps:

1. The plain-image denoted I_0 is transformed into its 1D corresponding vector I_0^b made up of the binary sequences at each pixel of I_0 taken in sequential order. Therefore, the binary initial vector I_0^b contains only the values 0 and 1 and its size is $L = 8 \times N \times M$ for a gray-level image (or $L = 3 \times 8 \times N \times M$ for RGB-color image).
2. A pseudo-randomized seed g in $\{1, \ldots, L\}$ initiates the relation of recurrence given by Eq. (1).
3. Do loop for the initial binary vector $I_0^b[i]$, where i is the current position in the vector I_0^b and construction of a second vector component $I_0^b[j]$ in a new chaotic position $j = i + 1 + X_{i+1}$ with Eq. (1). The elements of the vector I_0^b are transformed to $I_0^b[i] = Z_3$ and $I_0^b[j] = Z_1$ with
$$Z_1 = I_0^b[i],$$
$$Z_2 = I_0^b[j] = I_0^b(i + 1 + X_{i+1}),$$
$$Z_3 = Z_1 \oplus Z_2, \tag{4}$$
where the symbol $\oplus$ represents the exclusive OR operation bit-by-bit (XOR). This process is achieved until the end of the loop.
4. The bits of the vector I_0^b are gathered per package of $(3) \times 8$ to form the cipher-image I_1. That constitutes the steps for one round for the cipher algorithm using the seed g. A complete encryption scheme produces a cipher-image I_R, where R is the total number of rounds used to encrypt the plain-image I_0.

As mentioned at the third step (Eq. (4)), the xor operation is combined with the permutation of elements on positions i and j to increase the impredictability in the cipher-image I_R. The main loop of the encryption scheme is given by Algorithm 1. The encryption of the plain-image necessitates the storage of all the pixels in an 1D vector and the memory space complexity is $\mathcal{O}(L)$. Moreover, the number of rounds in the algorithm is automatically adapted to the plain-image to satisfy secure encryption (see the following subsection).

Algorithm 1 (*Main loop of the encryption algorithm* ($I_0 \longrightarrow I_R$)).

Require: $I_0; L; R; g_{1,\ldots,R}$;
  Initialisation $r = 1;\ F = L-2;\ I_0^b \leftarrow I_0$
  while $r \le R$ **do**
    $i = 0;\ S = L-1;\ X = g_r;\ X_g = X \times X;$
    while $i < F$ **do**
      $X \leftarrow [((X \times X \bmod S) \times X) + X_g] \bmod S$
      $j \leftarrow i + 1 + X$
      $Z_1 \leftarrow I_{r-1}^b[i]$
      $Z_2 \leftarrow I_{r-1}^b[j]$
      $Z_3 \leftarrow (Z_1 + Z_2) \bmod 2$
      $I_{r-1}^b[i] \leftarrow Z_3$
      $I_{r-1}^b[j] \leftarrow Z_1$
      $i \leftarrow i + 1$
      $S \leftarrow S - 1$
    end while
    $r \leftarrow r + 1$
  end while
  $I_R \leftarrow I_R^b$
  return I_R

The process of decryption is similar to the encryption one, achieved in the reverse order. By starting from the last seed g_R to the first one g_1, for each seed, that needs to compute the coefficients X_i by using Eq. (1), and to store

the value $i+1+X_i$ in a vector V. The values of positions are recovered by iteration on the vector V from its end while xoring and shuffling bits in the cipher. The main loop of the decryption scheme is given by Algorithm 2.

Algorithm 2 (*Main loop of the decryption algorithm* ($I_R \longrightarrow I_0$)).

Require: I_R; L; R; $g_{1,\ldots,R}$;
  Initialisation $r=R$; $F=L-2$; $I_R^b \leftarrow I_R$
  while $r>0$ **do**
    $i=0$; $S=L-1$; $X=g_r$; $X_g=X\times X$;
    while $i<F$ **do**
      $X \leftarrow (((X\times X \bmod S)\times X)+X_g) \bmod S$
      $j \leftarrow i+1+X$
      $V[i] \leftarrow j$
      $i \leftarrow i+1$
      $S \leftarrow S-1$
    end while
    $j \leftarrow F-1$
    while $j \geq 0$ **do**
      $i \leftarrow V[j]$
      $z1 \leftarrow I_r^b[j]$
      $z2 \leftarrow I_r^b[i]$
      $z3 \leftarrow (z1+z2) \bmod 2$
      $I_r^b[j] \leftarrow z2$
      $I_r^b[i] \leftarrow z3$
      $j \leftarrow j-1$
    end while
    $r \leftarrow r-1$
  end while
  $I_0 \leftarrow I_0^b$
  return I_0

2.3. Key space and determination of R

Given today's computer speed, it is commonly accepted that a key space of size smaller than 2^{128} is not secure enough [25]. In the present case, for one round (i.e. $R=1$), a number of L different ciphers, corresponding to seeds g in $\{1,\ldots,L\}$, can be produced. Therefore, with increasing the round number R, the total number of ciphers that can be generated is L^R. To satisfy the relation $L^R \geq 2^{128}$ and to avoid success of brute-force attacks, the minimum number of rounds R_1 to be used for the encryption is

$$R_1 = \mathrm{Floor}\left[\frac{128}{\log_2 L}\right]+1. \tag{5}$$

The value of the round number R_1 cannot be applied for all kind of images. Indeed, for a plain-image with very low entropy, more than R_1 rounds is necessary to assure the randomness of the cipher. The choice of the final number of rounds R is then related to the distribution of bits '0' and '1' in the plain-image. Assuming that in the plain-image, the occurrence of the bit '0' (resp. the bit '1') has a probability denoted $P_0(0)$ (resp. $P_0(1)=1-P_0(0)$), then at each new round r, the probability $P_r(0)$ is iteratively modified as $P_r(0)=P_{r-1}^2(0)+(1-P_{r-1}(0))^2$. To satisfy an uniform occurrence of bits 0 and 1, the limit of the suite $P_r(0)$ must be 0.50. The goal is to find a number of rounds R_2 satisfying the relation

$$\lim_{r\to R_2} P_r(0) = 0.50-\epsilon_1, \tag{6}$$

where ϵ_1 is a fixed numerical tolerance (here $\epsilon_1=0.001$). With such a given tolerance value, the value of R_2 is computed with Algorithm 3. The round number R_2 will be large for plain-images with very low Shannon-entropy and can be small for plain-images with Shannon's entropy closed to its maximum (i.e. 1 in base 2 or 8 in base 256). Nevertheless, for encryption of similar plain-images with the same seeds (i.e. sensitivity to plain-image), the security level is not maximum due to the high correlation between the cipher-images. To avoid high correlation between the cipher-images, an additional hypothesis must be taken into account. By considering two plain-images I_0 and I_0' of size L (in bits) and differing by only n_b bits (e.g. $n_b=1$ for only one bit in the worst case), the frequency t_0 of identical elements between these two images is equal to $t_0=(L-n_b)/L$. The frequency decreases according to the number of rounds and is given by

$$t_r = t_{r-1}^2 \quad \text{with } r\geq 1, \tag{7}$$

and must satisfy the relation

$$\lim_{r\to R_3} t_r \leq \epsilon_2, \tag{8}$$

where ϵ_2 is the acceptable criterion of similitude between binary sequences (e.g. ϵ_2 is fixed to 0.005, corresponding to a rate of identical bits smaller than 0.5%). The minimum number of round R_3 satisfying Eq. (8) is given by

$$R_3 = \mathrm{Floor}\left[\log_2\left(\frac{\ln(\epsilon_2)}{\ln(t_0)}\right)\right]+1. \tag{9}$$

With these three indicators (R_1, R_2 and R_3), the number of rounds R for encryption, assuring a maximum security level is

$$R = \max\{R_1,R_2,R_3\}. \tag{10}$$

Such a number of rounds R permits to satisfy simultaneously the criteria of key entropy, maximum Shannon's entropy and sensitivity to initial conditions (plain-image and key). As an example, with a 181×259 RGB-color image of size $L=1\,125\,096$, the number of round $R=23$ and the process enables to produce exactly $1\,125\,096^{23}$(i.e. $\simeq 2^{462}$) different ciphers from different keys. The values of the parameters concerning seeds g_i, with i in $\{1,\ldots,R\}$, can be pseudo-randomized in the set $\{1,\ldots,L\}$ for each round. The secret key corresponds to the sequence formed by the chosen (randomly or arbitrarily) seed values $\mathcal{K}=\{g_1,\ldots,g_R\}$, with each seed value g_i in $\{1,\ldots,L\}$.

Algorithm 3 (*Computation of* R_2 ($P_0(0)\longrightarrow R_2$)).

Require $P_0(0)$; $\epsilon_1=0.001$;
  $R_2=0$; $P_{R_2}=P_0(0)$; $dif=|0.50-P_{R_2}|$;
  while $dif>\epsilon_1$ **do**
    $P_{R_2} \leftarrow [P_{R_2}^2+(1-P_{R_2})^2]$
    $dif \leftarrow |0.50-P_{R_2}|$
    $R_2 \leftarrow R_2+1$
  end while
  return R_2

3. Security analysis

A good encryption scheme should be efficient and be able to resist to all kinds of cryptanalytic, statistical or brute-force attacks [19,20]. To meet this challenge, any

cryptosystem should have at least the three basic cryptographic characteristics: indistinguishability, confusion and diffusion [10,21].

1. Indistinguishability: for all used key, the ciphers should not be differentiated from the outputs of a truly random function (i.e. the ciphers should have a high level of randomness). This property is often omitted during the security analysis (strong condition).
2. Confusion: indicates that it should have no pattern or any relationship between the plain-image and the cipher-image. When the cipher-image respects the previous property, the confusion property is generally verified.
3. Diffusion: a difference of at least one bit in the keys (resp. in the plain-images) leads to completely different cipher-images (i.e. high sensitivity to key and plain-image).

The security analysis methods satisfying these three properties are presented and are used in the following to show the efficiency of the scheme and its security level in image encryption and transmission.

3.1. Indistinguishability, confusion and diffusion analysis

The purpose of the analysis is to check the three characteristics (indistinguishability, confusion and diffusion) on the cipher-images produced by the cryptosystem. The three following approaches are used to evaluate the randomness of the produced ciphers and the correlation that can exist between these ciphers.

1. Method 1: analysis of the randomness (i.e. indistinguishability and confusion) quality of each cipher belonging to a group of ciphers. These ciphers are considered as simple binary sequences and are individually analysed through the statistical tests suite NIST (National Institute of Standards and Technology of the U.S. Government). These NIST tests consist in a statistical package of fifteen statistical tests developed to quantify and to evaluate the randomness of binary sequences produced by cryptographic random or pseudorandom number generators [26]. For each statistical test, a set of p_{value} is computed. The fixed significance level chosen is $\alpha = 0.01$ what means that only 1% of the tested sequences are expected to fail. A sequence passes a statistical test whenever the $p_{value} \geq \alpha$ and fails otherwise. In case of testing multiple sequences at the same time, each test defines a proportion τ as the ratio of ciphers passing successfully the test relatively to the total number of ciphers $N_{ciphers}$ (i.e. $\tau = n[p_{value} \geq \alpha]/N_{ciphers}$). This proportion τ is compared to an acceptable proportion τ_{accept} that corresponds to the ratio of sequences that should pass the test. The range of acceptable proportions τ_{accept} is determined by using the confidence interval defined as [26]: $(1-\alpha) \pm 3\sqrt{\alpha(1-\alpha)/N_{ciphers}}$, (e.g. with 800 ciphers, the τ_{accept} is equal to 97.94%).
2. Method 2: analysis of the correlation (i.e. diffusion) between the cipher-images by computing the correlation coefficients of each pair of cipher-image. This method is used because the auto-correlation of each cipher is indirectly analysed through Method 1 (NIST tests). The two cipher-images are given as 1D vectors $I_x = [x_1, \ldots, x_l]$ and $I_y = [y_1, \ldots, y_l]$ where the size $l = (3) \times N \times M$ is depending on the codage of I_x and I_y (i.e. coded in gray-level or RGB-color). Two not correlated sequences are corresponding to correlation coefficient $C_{I_x I_y} = 0$ and a strong correlation occurs for $C_{I_x I_y} \simeq \pm 1$ [16,19]. We classify all the coefficients $C_{I_x I_y}$ calculated for each pair of sequence and the distribution of these coefficients is given by a histogram.
3. Method 3: to bring an additional response to the correlation between the cipher-images (i.e. diffusion), the two following indicators are used: *NPCR* and *UACI*. The *NPCR* gives an evaluation of the percentage of the difference between pixel values of two images and the *UACI* measures the average intensity of differences between these two images [7].

In the following, these methods are applied to analyse the sensitivity of the cryptosystem on the keys and on the plain-images. Such a process is illustrated through two images: one RGB-color and the second in gray-level.

The first image I_F^a is Fairy's image (i.e. a 181×259 RGB-color image with $L = 1\,125\,096$) illustrated in Fig. 3(a). An example of encryption of such an image is given in Fig. 3(b) and is achieved with the key $\mathcal{K}_F^a = \{g_1, \ldots, g_{23}\}$ (i.e. $R = 23$). The sequence of seed values for $\{g_1, \ldots, g_{23}\}$ is {17 654, 84 287, 7487, 1984, 12 314, 10, 74 120, 130 014, 95 210, 1914, 70 553, 2835, 19 800, 299 314, 83 721,

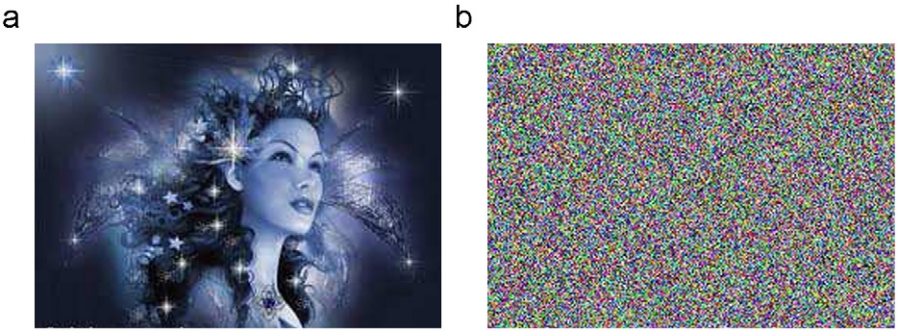

Fig. 3. The RGB-color Fairy's image: (a) the plain-image and (b) the corresponding cipher-image obtained with the key $\mathcal{K}_F^a$.

610 990, 210, 65 521, 396, 1 109 094, 230 014, 63 010, 10 246}. With Fairy's image, the histograms corresponding to the associated blue, green, and red channels before and after encryption with the key $\mathcal{K}_{\mathrm{F}}^{\mathrm{a}}$ are shown in Fig. 4(a–f).

The second image I_L^a, illustrated in Fig. 5(a), is the Lena's image (i.e. a 512×512 gray-level image with $L=2\,097\,152$). Fig. 5(b) shows the cipher-image obtained with the key $\mathcal{K}_{\mathrm{L}}^{\mathrm{a}} = \{g'_1, \ldots, g'_{24}\}$ (i.e. $R=24$). The seed sequence is $\{g'_1, \ldots, g'_{24}\} = \{75, g_1, \ldots, g_{23}\}$. Fig. 5(c,d) show the gray-level histograms of Lena's image before and after encryption with key $\mathcal{K}_{\mathrm{L}}^{\mathrm{a}}$. One can remark that, for these two images, the occurrence distributions of pixel values after encryption are quasi-uniform.

3.2. Key sensitivity analysis

The sensitivity on the secret key is an essential factor in any image encryption scheme. Indeed, a small deviation in the input should cause a large change in the output. For Fairy's image, the encryption process assures a maximum security with $R=23$ (i.e. a key-space of 462 bits of entropy). We analyse sensitivity of the produced cipher-image as function of the seed values. The considered encryption keys $\{g_1, \ldots, g_{23}\}$ are the seed sequences with $\{g_1, \ldots, g_{22}\} = \{17\,654,\ 84\,287,\ 7487,\ 1984,\ 12\,314,\ 10,\ 74\,120,\ 130\,014,\ 95\,210,\ 1914,\ 70\,553,\ 2835,\ 19\,800,\ 299\,314,\ 83\,721,\ 610\,990,\ 210,\ 65\,521,\ 396,\ 1\,109\,094,\ 230\,014,\ 63\,010\}$ and g_{23} in $\{10\,001, \ldots, 10\,800\}$. The seeds

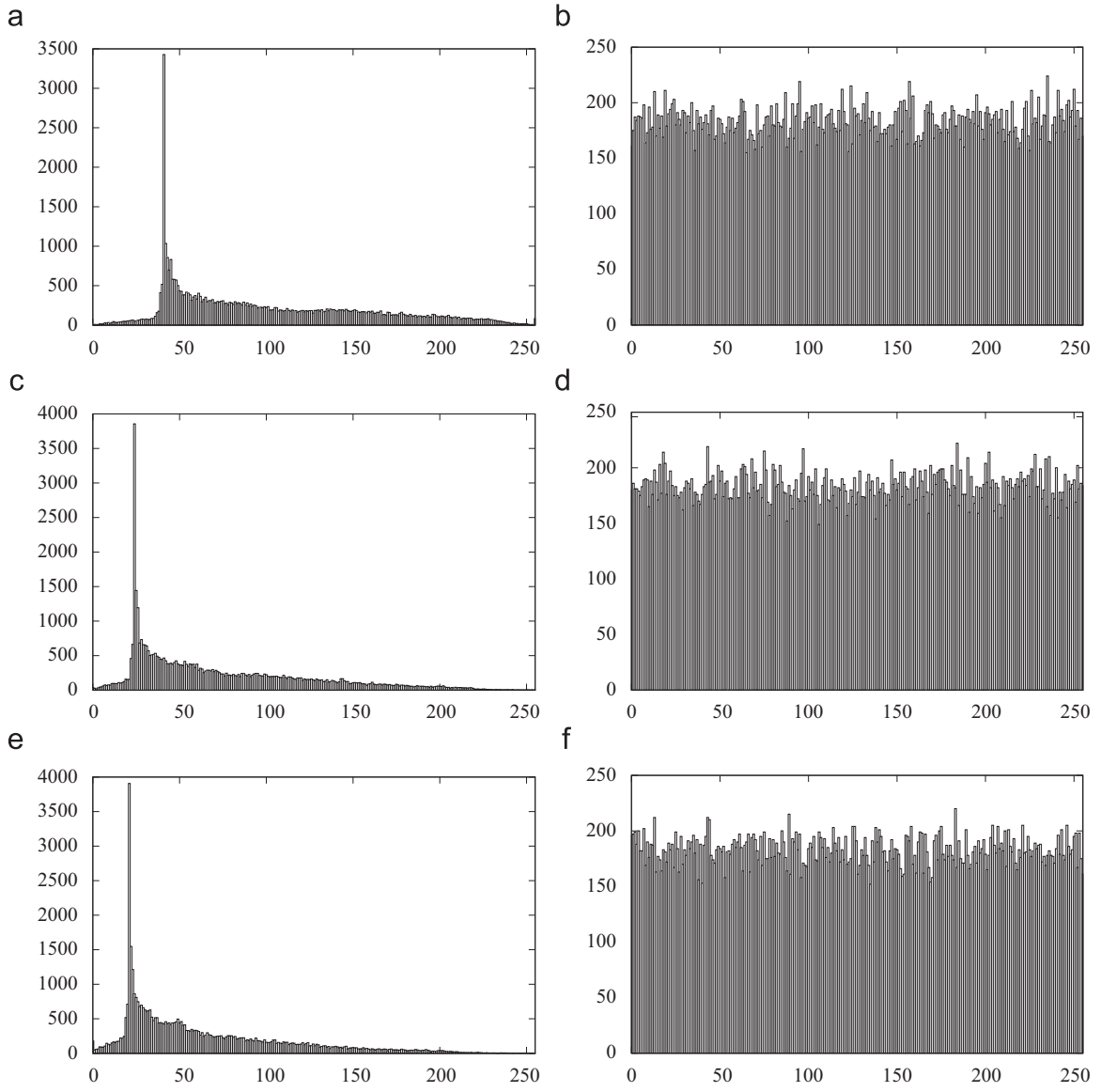

Fig. 4. RGB channel distributions: (a, c and e) show the frequency distributions before encryption of the Fairy's image for the blue, green and red channels, respectively. (b, d and f) show the associated histograms of the cipher-image, after encryption with the key $\mathcal{K}_{\mathrm{F}}^{\mathrm{a}}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

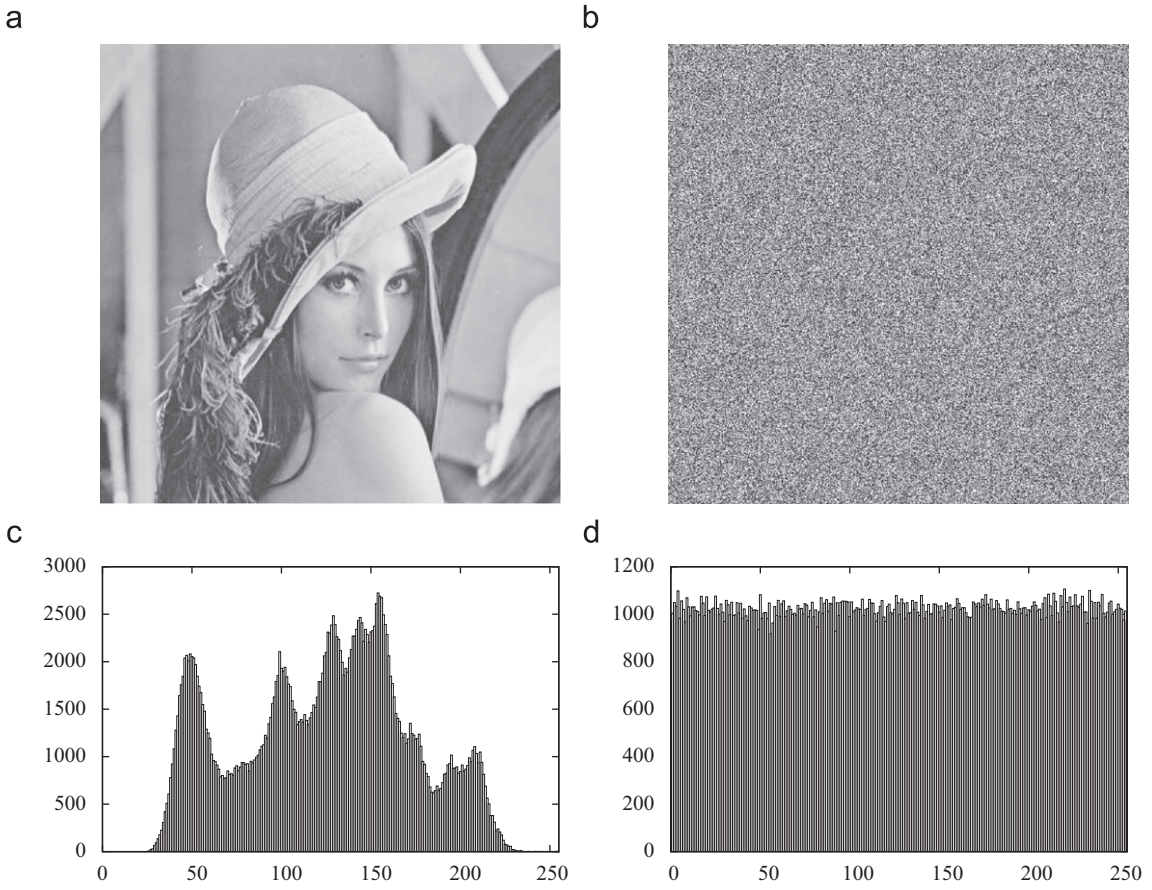

Fig. 5. The gray-level Lena's image and frequency histograms: (a) the plain-image, (b) the corresponding cipher-image with key $\mathcal{K}_{\mathrm{L}}^{\mathrm{a}}$. Frequency histograms for (c) the plain-image and (d) the cipher-image.

g_i, with $1 \leq i \leq 22$, are chosen in the set $\{1,\ldots,1\,125\,096\}$. For Lena's image, the analysis is achieved for $R=24$ (i.e. assuring a key-space of 504 bits of entropy) with $\{g'_1,\ldots,g'_{24}\}=\{75,g_1,\ldots,g_{23}\}$. In each case, the number of generated ciphers is equal to 800. We note that, for the last seed value (i.e. g_{23} for Fairy's image and g'_{24} for Lena's image), the selection of 800 successive seed values has been done to analyse the sensitivity due to the seed values on the produced cipher-images. Moreover, the analysis is also achieved on a set of five 256×256 RGB-color images and a set of five 512×512 gray-level images. These images come from the USC-SIPI image data base and are corresponding to the RGB-color images 4.1.01.tiff to 4.1.05.tiff (resp. the gray-level images 7.1.01.tiff to 7.1.05.tiff) in the miscellaneous volume. For each downloaded image, 800 cipher-images are produced by using the keys $\{g_1,\ldots,g_{23}\}$ and $\{g'_1,\ldots,g'_{24}\}$ for RGB-color and gray-level images, respectively.

3.2.1. Analysis with Method 1

The goal of the analysis is to test the randomness quality of the 800 produced ciphers as function of the seed values for the two plain-images (Lena and Fairy). The results obtained with the NIST tests on these ciphers, are given in Table 1. For the tests achieved on the set of the five RGB-color (resp. the five gray-level images), the indicators are the average values $\tau_{average}$ of ratio τ and are given in Table 2. For the tests "Non Overlapping", "Random Excursions" and "Random Excursions Variant" the smallest percentage of all under tests are presented. We notice that, in Table 1, the ciphers pass successfully all the NIST tests ($\tau \geq \tau_{accept}=97.94\%$) and can be considered as good candidates of random binary sequences. Moreover, the results of Table 2 show that all the average values $\tau_{average}$ are larger than $\tau_{accept}=97.94\%$. That shows the quality of tested ciphers as function of the key and for different plain-images. We note that the analysis refers to the first property about "indistinguishability" (strong condition).

3.2.2. Analysis with Method 2

For each plain-image (Fairy and Lena), the correlation coefficients between the 800 produced cipher-images are computed. The histograms of the coefficients $C_{I_xI_y}$ between the cipher-images are presented in Fig. 6(a). We show that these coefficients are close to 0 and belong to the interval $[-0.0085, 0.0085]$. Moreover, 99.18% of the values of the correlation coefficients between the ciphers produced from Fairy's image do not exceed absolute values larger than 0.0065 and 99.07% of the values of

Table 1
Results of the NIST tests on the 800 produced ciphers for Fairy's and Lena's images. The ratio τ of p_{value} passing the tests and the result for each test are presented.

Test name	Fairy's image		Lena's image	
	τ	Result	τ	Result
Frequency	98.00	Success	99.12	Success
Block-frequency	99.00	Success	98.75	Success
Cumulative sums (1)	98.12	Success	99.25	Success
Cumulative sums (2)	98.00	Success	99.37	Success
Runs	98.75	Success	98.87	Success
Longest run	99.37	Success	99.62	Success
Rank	99.00	Success	99.25	Success
FFT	98.37	Success	98.87	Success
Non-overlapping	98.00	Success	98.50	Success
Overlapping	99.12	Success	98.75	Success
Universal	98.12	Success	98.50	Success
Approximate entropy	98.50	Success	98.62	Success
Random excursions	98.01	Success	98.04	Success
Random e-variant	98.41	Success	98.22	Success
Serial (1)	98.87	Success	99.00	Success
Serial (2)	98.87	Success	99.00	Success
Linear complexity	98.75	Success	98.87	Success

Table 2
Results of the NIST tests on the 800 produced cipher-images for the set of five RGB-color and gray-level images. The average value $\tau_{average}$ of the ratio τ of p_{value} passing the tests and the result for each test are given.

Test name	Set of RGB-color images		Set of gray-level images	
	$\tau_{average}$	Result	$\tau_{average}$	Result
Frequency	99.17	Success	99.04	Success
Block-frequency	98.74	Success	99.02	Success
Cumulative sums (1)	99.12	Success	98.95	Success
Cumulative sums (2)	99.17	Success	99.14	Success
Runs	99.10	Success	99.02	Success
Longest run	98.92	Success	98.84	Success
Rank	99.12	Success	98.87	Success
FFT	98.92	Success	98.67	Success
Non-overlapping	98.47	Success	98.40	Success
Overlapping	98.42	Success	98.17	Success
Universal	98.82	Success	98.84	Success
Approximate entropy	98.62	Success	98.79	Success
Random excursions	98.47	Success	98.37	Success
Random e-variant	98.48	Success	98.40	Success
Serial (1)	99.07	Success	98.94	Success
Serial (2)	99.04	Success	99.00	Success
Linear complexity	98.89	Success	98.97	Success

the correlation coefficients between the ciphers produced from Lena's image have absolute values smaller than 0.0048. For the set of five RGB-color images (resp. five gray-level images), the histogram of correlation takes into account all the correlation coefficients between the produced ciphers (see Fig. 6(b)). The correlation coefficients are close to 0 and belong to the interval $[-0.0085, 0.0085]$. More than 99.08% of the values of the correlation coefficients between the ciphers produced from the five RGB-color images do not exceed absolute values larger than 0.0061 and 99.42% of the values of the correlation coefficients between the ciphers produced from the five gray-level images have absolute values smaller than 0.0053. These results show that only a very small (or negligible) correlation can be detected between the cipher-images. Such an analysis refers to the second property: the "diffusion". The quality of randomness and decorrelation between cipher-images are essential for any image encryption scheme to resist chosen-ciphertext attacks.

3.2.3. Analysis with Method 3

Actually in the previous analysis of correlation, the key sensitivity has already been tested on a bloc of 800 successive keys for the two plain-images. To illustrate the information given by the *NPCR* and *UACI* coefficients, the plain-images for Fairy and Lena are encrypted by using the keys $\mathcal{K}_F^a = \{g_1,\ldots,g_{22},10\,246\}$ and $\mathcal{K}_L^a = \{75,g_1,\ldots,g_{22},10\,246\}$, respectively. We analyse the cipher-images produced by encryption with two slightly different keys $\mathcal{K}_F^b = \{g_1,\ldots,g_{22},10\,245\}$, $\mathcal{K}_F^c = \{g_1,\ldots,g_{22},10\,247\}$ for Fairy's image and $\mathcal{K}_L^b = \{g'_1,\ldots,g'_{23},10\,245\}$, $\mathcal{K}_L^c = \{g'_1,\ldots,g'_{23},\ 10\,247\}$ for Lena's image. The correlation, *NPCR* and *UACI* coefficients for the Fairy and Lena are given in Tables 3 and 4, respectively. These small values of the correlation coefficients show that the cipher-images are weakly correlated and that the cryptosystem is very sensitive to the seed values. In addition, the same study is applied on each of the five RGB-color images (resp. five gray-level images), the average and the standard deviation values for the correlation coefficients, the *NPCR* and the *UACI* are presented in Tables 5 and 6.

3.3. Plain-image sensitivity analysis

To analyse the sensitivity to the plain-image (i.e. analysis of the diffusion property), a modification of only one bit is applied on the binary components of the plain-image. Therefore, for each case (Fairy's and Lena's images), we consider three initial images almost identical (but differing by only one bit). The encryption of these three plain-images with different seeds produces completely different cipher-images. The encryption of these plain-images with the same seeds can produce very close cipher-images. The goal is to analyse the propagation of the initial difference between three near plain-images through encryption process. First, we consider as original image in RGB-color, Fairy's image I_F^a of Fig. 3(a). Its pixel values, at the upper left position (0,0) (resp. lower right position (180,258)), are [183, 145, 133] (resp. [41, 24, 21]) for the blue, green and red channels. A second image I_F^b consists in duplicating Fairy's image and in changing the first pixel value of the blue channel from 183 to 182. The third image I_F^c consists in duplicating again Fairy's image and in changing the last pixel value of the red channel from 21 to 20. Between these three images, no visual difference can be observed. We compute and analyse the correlation coefficients, the *NPCR* (number of pixel value change rate) and the *UACI* (unified average changing intensity) between the corresponding three cipher-images using the same key. The same approach is applied on Lena's image I_L^a to produce a second image I_L^b obtained by changing the first pixel gray-level value (i.e. 162–163) and for the third image I_L^c by modifying the last pixel value (i.e. 108–109). The results for Fairy's and Lena's images

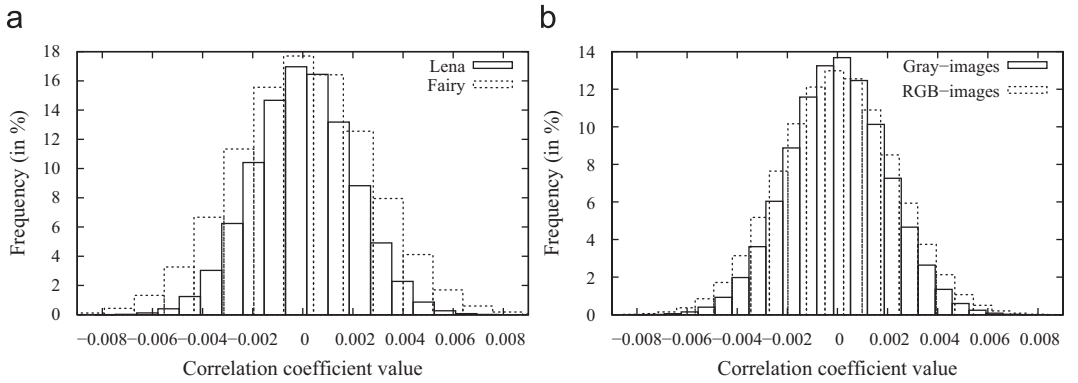

Fig. 6. Distribution of correlation coefficients $C_{I_k I_y}$: (a) the coefficients of Fairy's and Lena's tested cipher-images and (b) the coefficients for the sets of five RGB-color and gray-level images.

Table 3
Correlation coefficients, *NPCR* (in %) and *UACI* (in %) between the three cipher-images for Fairy using three slightly different keys $\mathcal{K}_F^a$, $\mathcal{K}_F^b$ and $\mathcal{K}_F^c$.

Keys 1/2	Corr.	*NPCR*	*UACI*
$\mathcal{K}_F^a/\mathcal{K}_F^b$	−0.0007	99.61	33.42
$\mathcal{K}_F^a/\mathcal{K}_F^c$	−0.0030	99.61	33.55
$\mathcal{K}_F^b/\mathcal{K}_F^c$	0.0044	99.61	33.39

Table 4
Correlation coefficients, *NPCR* (in %) and *UACI* (in %) between the three cipher-images of Lena using three slightly different keys $\mathcal{K}_L^a$, $\mathcal{K}_L^b$ and $\mathcal{K}_L^c$.

Keys 1/2	Corr.	*NPCR*	*UACI*
$\mathcal{K}_L^a/\mathcal{K}_L^b$	0.0008	99.62	33.45
$\mathcal{K}_L^a/\mathcal{K}_L^c$	0.0014	99.64	33.47
$\mathcal{K}_L^b/\mathcal{K}_L^c$	0.0004	99.63	33.47

Table 5
Average and standard deviation values of the correlation coefficients, *NPCR* (in %) and *UACI* (in %) between the cipher-images for the set of the five RGB-color images using the three slightly different keys $\mathcal{K}_F^a$, $\mathcal{K}_F^b$ and $\mathcal{K}_F^c$.

Keys 1/2	Corr.		*NPCR*		*UACI*	
	m_{corr}	σ_{corr}	m_{NPCR}	σ_{NPCR}	m_{UACI}	σ_{UACI}
$\mathcal{K}_F^a/\mathcal{K}_F^b$	0.0021	0.0011	99.60	0.02	33.43	0.07
$\mathcal{K}_F^a/\mathcal{K}_F^c$	0.0014	0.0007	99.60	0.01	33.43	0.05
$\mathcal{K}_F^b/\mathcal{K}_F^c$	0.0014	0.0010	99.61	0.01	33.40	0.07

are shown in Tables 7 and 8, respectively. The values of the coefficients in Tables 7 and 8 show clearly the difference of the tested cipher-images produced by slightly different plain-images.

A statistical analysis on plain-image sensitivity is also achieved by producing 100 near images for each initial Fairy's and Lena's images. For Fairy's image, the blue channel value of the first pixel (i.e. 183) is decremented by 1 (i.e. $pixel_F(0,0) = [183, \ldots, 84]$), forming 100 similar images excepting at the blue channel value. The same process is applied on the gray-level Lena's image, where

Table 6
Average and standard deviation values of the correlation coefficients, *NPCR* (in %) and *UACI* (in %) between the cipher-images of the five gray-level images using the three slightly different keys $\mathcal{K}_L^a$, $\mathcal{K}_L^b$ and $\mathcal{K}_L^c$.

Keys 1/2	Corr.		*NPCR*		*UACI*	
	m_{corr}	σ_{corr}	m_{NPCR}	σ_{NPCR}	m_{UACI}	σ_{UACI}
$\mathcal{K}_L^a/\mathcal{K}_L^b$	0.0011	0.0004	99.60	0.02	33.47	0.04
$\mathcal{K}_L^a/\mathcal{K}_L^c$	0.0009	0.0005	99.61	0.01	33.49	0.02
$\mathcal{K}_L^b/\mathcal{K}_L^c$	0.0014	0.0008	99.60	0.01	33.42	0.02

Table 7
Correlation coefficients, *NPCR* (in %) and *UACI* (in %) between the three cipher-images produced from the three slightly different plain-images of Fairy I_F^a, I_F^b and I_F^c. Two examples of encryption are achieved with the keys $\mathcal{K}_F^a$ and $\mathcal{K}_F^c$.

Image 1/2	$\mathcal{K}_F^a$			$\mathcal{K}_F^c$		
	Corr.	NPCR	UACI	Corr.	NPCR	UACI
I_F^a/I_F^b	−0.0017	99.60	33.43	−0.0013	99.64	33.43
I_F^a/I_F^c	0.0059	99.62	33.35	0.0016	99.60	33.43
I_F^b/I_F^c	−0.0019	99.61	33.54	0.0017	99.60	33.38

the value 162 of the first pixel is decremented by 1 to obtain 100 near images (i.e. $pixel_L(0,0) = [162, \ldots, 63]$). The used keys for the encryption are $\mathcal{K}_F^a$ (resp. $\mathcal{K}_L^a$) for Fairy's (resp. Lena's) images. For each set of 100 near images, the average and the standard deviation values of the correlation coefficients, the NPCR and the UACI are computed. Each pair of modified images is analysed and the results are presented in Table 9, for both sets of modified Fairy's and Lena's images. The average values of correlation coefficients, *NPCR* and *UACI*, are stable with only small standard deviations. The obtained results, clearly show the high sensitivity related to the plain-image. The encryption process assures a maximum level of security by taking into account simultaneously the key space, the maximum Shannon's entropy, the randomness of the cipher-images and the plain-image sensitivity against differential attacks.

Table 8
Correlation coefficients, *NPCR* (in %) and *UACI* (in %) between the three cipher-images produced from the three slightly different plain-images of Lena I_L^a, I_L^b and I_L^c. Two examples of encryption are achieved with the keys $\mathcal{K}_L^a$ and $\mathcal{K}_L^c$.

Image 1/2	$\mathcal{K}_L^a$			$\mathcal{K}_L^c$		
	Corr.	NPCR	UACI	Corr.	NPCR	UACI
I_L^a/I_L^b	−0.0001	99.60	33.46	0.0015	99.60	33.46
I_L^a/I_L^c	0.0008	99.60	33.47	−0.0010	99.60	33.52
I_L^b/I_L^c	0.0048	99.62	33.37	0.0002	99.63	33.51

Table 9
Statistical analysis on correlation coefficients, *NPCR* and *UACI* for the 100 near plain-images produced from Fairy's and Lena's images, respectively.

Indicator	Fairy's images		Lena's images	
	m_F	σ_F	m_L	σ_L
Corr. coef.	0.0022	0.0017	0.0015	0.0011
NPCR	99.61	0.01	99.60	0.01
UACI	33.45	0.06	33.46	0.04

Table 10
Comparison of time complexity for encryption/decryption of gray-level images for several image sizes. Each image pixel is encoded on 8 bits. The entropy of the produced key space is presented.

Image size (in pixels)	*R*-value	Entropy of the key-space (bits)	Encryption/ decryption (s)	Ref. [12]
64×64	18	270	0.03/0.04	0.19
128×128	20	340	0.14/0.15	0.95
256×256	22	418	0.90/0.96	6.01
512×512	24	504	7.86/8.02	35.59
1024×1024	26	598	44.50/45.72	253.88

3.4. Time complexity

Finally, the time complexity of the algorithm for encryption/decryption is evaluated. Several images for different sizes have been considered and the time complexity is given. The time complexity analysis is achieved on Intel(R) Pentium(R) M processor 1700 MHz with 2048 MB RAM personal computer. The algorithm is coded in *C* and compiled by *gcc*-4.6.0 on Fedora release 11 (Leonidas). The results are shown in Table 10. The encryption scheme permits a gain factor of about five relatively to the method used in Ref. [12] (see Table 10). Instead, the range of the key space is increased (e.g. the key-space entropy in Ref. [12] is only 84, 75 and 113 bits). That necessarily leads to increase the complexity time for the algorithm but assures a secure transmission especially if the computation time is not a constraint.

4. Conclusion

A new image encryption scheme using a chaotic function based linear congruences was presented. The process is the coupling of a chaotic function with the xor operation during the binary treatment of the cipher algorithm. This method has drastically disrupted the internal binary structure of the images and progressively induced randomness characteristics. We have shown that such a scheme is able to produce a large number of cipher-images, whose cryptographic qualities have been evaluated through different statistical analyses. The key space is large enough to resist brute-force attacks and statistical analysis show the high key and plain-image sensitivity. The cipher-images pass successfully the NIST tests and a negligible correlation between cipher-images can be guaranteed. The advantage of the encryption scheme is its automatic adaption to the entropy of the plain-image assuring secure cipher-image. Moreover, only integers are used during the encryption/decryption processes that is important for the portability architecture. Finally, we conclude that the proposed scheme is expected to be useful for applications with a secret key constituted by the used seed values.

Acknowledgments

Authors thank the Centre de Calcul Intensif ROMEO II-III for computational facilities, the Région Champagne-Ardennes and the Conseil Régional de l'Aube for financial supports.

References

[1] N. Bourbakis, C. Alexopoulos, Picture data encryption using SCAN pattern, Pattern Recognition 25 (6) (1992) 567–581.
[2] R. Refregier, B. Javidi, Optical image encryption based on input plane and Fourier plane random encoding, Optics Letters 20 (7) (1995) 767–769.
[3] Z. Liu, Q. Guo, L. Xu, M.A. Ahmad, S. Liu, Double image encryption by using iterative random binary encoding in gyrator domains, Optics Express 18 (11) (2010) 12033–12043.
[4] C.C. Chang, M.S. Hwang, T.S. Chen, A new encryption algorithm for image cryptosystems, Journal of Systems and Software 58 (7) (2001) 83–91.
[5] H.K.L. Chang, J.L. Liu, A linear quad tree compression scheme for image encryption, Signal Processing 10 (4) (1997) 279–290.
[6] H. Cheng, X.B. Li, Partial encryption of compressed image and videos, IEEE Transactions on Signal Processing 48 (8) (2000) 2439–2451.
[7] G. Zhang, Q. Liu, A novel image encryption method based on total shuffling scheme, Optics Communications 284 (12) (2011) 2775–2780.
[8] J. Scharinger, Fast encryption of image data using chaotic Kolmogorov flow, Journal of Electronic Engineering 7 (2) (1998) 318–325.
[9] A.K. Acharya, Image encryption using a new chaos based encryption algorithm, in: Proceedings of the International Conference on Communication, Computing and Security, NY, USA, 2011, pp. 577–581.
[10] G. Álvarez, S. Li, Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems, International Journal of Bifurcation and Chaos 16 (8) (2006) 2129–2151.
[11] N.K. Pareek, V. Patidar, K.K. Sud, Discrete chaotic cryptography using external key, Physics Letters A 1–2 (309) (2003) 75–82.
[12] X. Wang, J. Zhang, An image scrambling encryption using chaos-controlled Poker shuffle operation, in: Proceedings of International Symposium on Biometrics and Security Technologies, 23–24 April 2008, pp. 1–6.

[13] J. Fridrich, Symmetric ciphers based on two dimensional chaotic maps, International Journal of Bifurcation and Chaos 8 (6) (1998) 1259–1284.
[14] E. Solak, C. Çokal, O.L. Yildiz, T. Biyikoglu, Cryptanalysis of Fridrich's chaotic image encryption, International Journal of Bifurcation and Chaos 20 (5) (2010) 1405–1413.
[15] G. Chen, Y. Mao, C.K. Chui, A symmetric image encryption based on 3D chaotic maps, Chaos, Solitons & Fractals 21 (3) (2004) 749–761.
[16] T. Gao, Z. Chen, A new image encryption algorithm based on hyper-chaos, Physics Letters A 372 (4) (2008) 394–400.
[17] K.W. Wong, B.S.H. Kwok, W.S. Law, A fast image encryption scheme based on chaotic standard map, Physics Letters A 372 (15) (2008) 2645–2652.
[18] S. Li, C. Li, G. Chen, N.G. Bourbakis, K.T. Lo, A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, Signal Processing: Image Communication 23 (3) (2008) 212–223.
[19] N.K. Pareek, V. Patidar, K.K. Sud, Image encryption using chaotic logistic map, Image and Vision Computing 24 (9) (2006) 926–934.
[20] C. Li, S. Li, M. Asim, J. Nunez, G. Alvarez, G. Chen, On the security defects of an image encryption scheme, Image and Vision Computing 27 (9) (2009) 1371–1381.
[21] S. Li, X. Zheng, Cryptanalysis of a chaotic image encryption method, in: Proceedings of the IEEE International Symposium on Circuits and Systems, vol. 2, Scottsdale, AZ, USA, 2002, pp. 708–711.
[22] L. Blum, M. Blum, M. Shub, A simple unpredictable pseudo-random number generator, The SIAM Journal on Computing 15 (2) (1986) 364–383.
[23] A. Wolf, J.B. Swift, H.L. Swinney, J.A. Vastano, Determining Lyapunov exponents from a tile series, Physica 16 (3) (1985) 285–317.
[24] E. Aurell, G. Boffetta, A. Crisanti, G. Paladin, A. Vulpiani, Predictability in the large: an extension of the concept of Lyapunov exponent, Journal of Physics A: Mathematical and General 30 (1) (1997) 1–26.
[25] B. Schneier, Self-study course in block cipher cryptanalysis, Cryptologia 24 (1) (2000) 18–34.
[26] A. Rukhin, et al., A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Special Publication, Revision 1a, 2010.