Image Encryption Using Keystreams Dependent on Plain-images

Ruisong Ye, Rong Fu, Shaojun Zeng, Chuting Lai, Junming Ma

Abstract. A novel chaos-based image encryption scheme is proposed in this paper. The encryption scheme consists of one permutation process and one diffusion process, in which the key streams are both one-time keys in the sense that they are dependent on plain-images. Several merits of the proposed image encryption scheme are achieved, including a huge key space, good statistical properties resisting statistical attack and differential attack, desirable resistance against known-plaintext and chosen-plaintext attacks. Experimental results have been carried out with detailed analysis to show that the proposed scheme can be a potential candidate for practical image encryption.

Keywords: chaotic system • Arnold map • image encryption

1 Introduction

The communications of digital products over network grow rapidly in the past decades, and consequently it has been an urgent need to prevent them from leakages. Many applications, such as military image databases, confidential video conference, private photograph album, etc. require reliable, fast and robust secure system to store and transmit digital images. The requirements to fulfil the security needs of digital images have led to the development of effective image encryption algorithms. Digital images possess some intrinsic features, such as bulk data capacity, redundancy of data, strong correlation among adjacent pixels, being less sensitive as compared to the text data, etc. As a result, most conventional ciphers, such as Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), Advanced Encryption Standard (AES) [1], which consider plain-image as either block cipher or data stream, are thereby not suitable for practical digital image encryption in real time, because their speed is low due to a bulk data volume and strong correlation among adjacent pixels. Fortunately, many chaos-

Ruisong Ye (✉)
Department of Mathematics, Shantou University, Shantou, Guangdong, 515063, China
E-mail: rsye@stu.edu.cn

based image encryption algorithms have been proposed in recent years and have shown their superior performance [2,3,4,5]. The reason of applying chaos theory in cryptography lies in its intrinsic features, such as ergodicity, pseudo-randomness, sensitivity to initial conditions and control parameters, etc. These chaotic characteristics are in accordance with the requirements of confusion and diffusion in cryptography [6].

Recently, a number of chaos-based image encryption algorithms have been broken [7,8,9], among which most of the key streams are not dependent on plain-images. The opponents can analyze the cryptography schemes via chosen-plaintext attack or known-plaintext attack to obtain the keystreams and so equivalently break the cipher systems. To overcome the drawbacks, a novel chaos-based image encryption scheme is proposed in this paper. The image encryption scheme consists of one permutation process and one diffusion process. In both processes, the keystreams not only depend on the cipher keys, but also closely relate to the original plain-images. Several merits of the proposed image encryption scheme are achieved, including a huge key space, good statistical properties resisting statistical attack and differential attack, desirable resistance against s known-plaintext and chosen-plaintext attacks. Experimental results have been carried out with detailed analysis to show that the proposed encryption scheme is highly secure.

2 The Generalized Arnold Map

Arnold map was proposed by V. I. Arnold in the research of ergodic theory; it is also called cat map. The map is a process of clipping and splicing that realign the pixel matrix of digital image. The classical Arnold map is an invertible map described by

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \mod 1 \tag{1}$$

where the notation "$x \bmod 1$" refers to the fractional part of a real number x by adding or subtracting an appropriate integer. The classical Arnold map (1) can be generalized to the following form by introducing two control parameters $a > 0$ and $b > 0$:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & 1+ab \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \mod 1. \tag{2}$$

It is easy to calculate the largest Lyapunov characteristic exponent of the map (2) is greater than 1, implying that the map is always chaotic for $a > 0$, $b > 0$. Note that the leading Lyapunov characteristic exponent is larger than that of the map (1) as $a > 1$, $b > 1$. It implies that the map (2) is in a stronger sense chaotic, and therefore can perform better data mixing.

3 The Image Encyption Scheme

3.1 Permutation Process

Thanks to chaotic nature of generalized Arnold map on the unit square $[0,1)^2$, one can easily get the chaotic orbit $\{(x_k, y_k), k = 0,1,\cdots\}$ of (x_0, y_0) with given control parameters a, b. As long as the orbit point number tends to infinity, the orbit will go through the unit square $[0,1)^2$ theoretically. As for the discrete case, all the pixels will be traversal if we iterate the chaotic map with sufficient times. For the sake of saving workload, we confine the iteration times $MaxIter = 10^5$, and the pixels, which are not ergodic, are arranged orderly in the end of the shuffled image. We denote the plain-image A and the shuffled image B with height H and width W. We also set an initial vector V with length $H \times W$. The permutation process is stated as follows.

Step 1. Set the values of the control parameters $a_1, b_1, a_2, b_2, a_3, b_3$ and the initial conditions x_0, y_0, say

$$a_1 = 1.11,\ b_1 = 10.22,\ a_2 = 20.33,\ b_2 = 30.44,$$
$$a_3 = 40.55,\ b_3 = 50.66,\ x_0 = 0.05,\ y_0 = 0.5.$$

Step 2. Calculate (x_1, y_1) by

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 & a_1 \\ b_1 & 1 + a_1 b_1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \mod 1.$$

Then we get the first pixel coordinates by $(s_1, t_1) = floor(x_1 \times W, y_1 \times H)$, and set $V(1) = A(s_1, t_1)$, $k = 0, j = 1$.

Step 3. Choose the corresponding generalized Arnold map by the gray value g_k of the plain image at (s_k, t_k). There are three generalized Arnold maps can be chosen. Let $g_k \mod 3 = i$, then

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & a_{i+1} \\ b_i & 1 + a_{i+1} b_{i+1} \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix} \mod 1,$$
$$(s_{k+1}, t_{k+1}) = floor(x_{k+1} \times W, y_{k+1} \times H).$$

Step 4. If (s_{k+1}, t_{k+1}) has not appeared before, then set $j = j+1, k = k+1$ and $V(j) = A(s_{k+1}, t_{k+1})$; otherwise, set $k = k+1$. Return to Step 3 until k arrives at $MaxIter$.

Step 5. Put the pixels not processed after Step 4 orderly to the remainder part of vector V. Reshape V into one 2D matrix B denoting the shuffled image.

3.2 Diffusion Process

It is necessary that a secure encryption algorithm should have a good mechanism of diffusion. On one hand, the diffusion processing can render the permutation process non-invertible, which therefore strengthens the security. On the other hand, the diffusion processing can significantly change the statistical properties of the plain-image by spreading the influence of each bit of the plain-image all over the cipher-image. The diffusion process will generally enhance the resistance to statistical attack and differential attack greatly, in which the histogram of the cipher-image is fairly uniform and is significantly different from that of the plain-image. The opponent can't find any useful clues between the plain-image and the cipher-image and so can't break the cryptosystem even after they spend a lot of time and effort. A good diffusion process should yield keystreams strongly related to plain-images. When encrypting different plain-images (even with the same cipher keys), the encryption scheme should generate different keystreams. The diffusion process is outlined as follows.

Step 1. Applying the permutation process to confuse the plain-image A and get a shuffled image B. Set the values of the initial condition z_0 and the control parameters a_4, b_4, c, say $z_0 = 0.33, c = 0.7, a_4 = 60.77, b_4 = 70.88$, in the diffusion process.

Step 2. Let $i = 0$.

Step 3. Apply the following quantization formula to yield one 8-bit pseudo-random grey values $d_1(i), d_2(i)$: $d_1(i) = \text{floor}(L \times x_i), d_2(i) = \text{floor}(L \times y_i)$,where L is the color level (for a 256 grey-scale image, $L = 256$), the "floor" operation on x returns the largest value not greater than x. $x_0 = 0.05$, $y_0 = 0.5$ are set in the permutation process.

Step 4. Compute the pixel grey value in the cipher-image by a bi-directional diffusion transmission:

$$C(2i+1) = \phi(2i) \oplus [(d_1(i) + C(2i)) \bmod 256];$$

$$C(2i+2) = \phi(2i+1) \oplus [(d_2(i) + C(2i+1)) \bmod 256],$$

where $\phi(2i), \phi(2i+1)$ are the grey values of the current operated pixel in the shuffled image which has been rearranged according to the order of row or column to a vector with length $H \times W$, $C(2i)$ is the previous output cipher-pixel grey value. The diffusion process is well defined as the initial condition $C(0)$ is provided. $C(0)$ can be set to be part of the keys in the diffusion process or can just take the value of $d_1(0)$ for simplicity.

Step 5. Compute s by $s = 1 + [C(2i+1) \bmod 2]$ to get the next (x_{i+1}, y_{i+1}) by iterating the generalized Arnold map with control parameters a_4, b_4 on (x_i, y_i) for s rounds. This is the crucial step to generate a keystream depending on the plain-

image since s is related to $C(2i+1)$, so are x_{i+1}, y_{i+1}. The encrypted image not only relates to the cipher keys, but also relates to the plain-image.

Step 6. Let $i = i+1$ and return to Step 3 until i reaches $H \times W / 2$.

The above diffusion process implies that it can't influence the pixels before the tampered pixel with a grey value change. As a remedy, we here add a reverse diffusion process as a supplement to the above diffusion process. The chaotic map used here is the generalized Bernoulli shift map.

Step 7. Iterate the following generalized Bernoulli shift map to produce another pseudo-random grey value sequence

$$z_{k+1} = (z_k / c) \bmod 1,\ \psi(k+1) = \text{floor}(L \times z_{k+1}), k = 0,1,...,H \times W - 1.$$

Step 8. Execute the reverse diffusion process:

$$D(i) = D(i) + 1 \oplus [(C(i) + \psi(i)) \bmod L], i = H \times W, ..., 2, 1,$$

where $D(i), i = 1, 2, ..., H \times W$ are the final encrypted vector consisting of the encrypted image pixel grey-scale values. The value of $D(H \times W + 1)$ should be provided to cipher out the sequence $D(i), i = 1, 2, ..., H \times W$. $D(H \times W + 1)$ can be handled in the same way as $C(0)$.

The complete diffusion process is composed of Step 1 to Step 8. The permutation process and the diffusion process form the proposed image encryption scheme. The original image Lena is encrypted and the result is shown in Fig.1 (b).

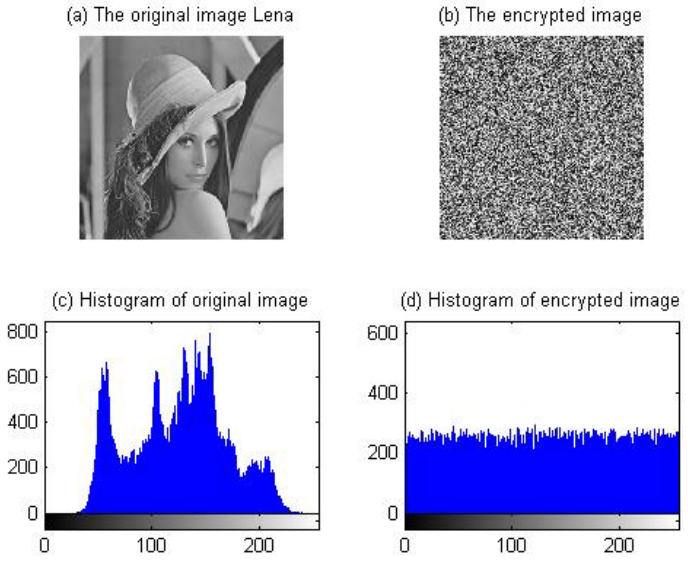

Fig. 1 The encrypted results

4 Security Analysis

4.1 Key Space Analysis

A good image encryption scheme aims to protect secret information so that it must contain sufficiently large key space for compensating the degradation dynamics in PC. Since the permutation process is irrelevant to the diffusion process, the key space consists of the cipher keys in both processes. The control parameters $a_1, b_1, a_2, b_2, a_3, b_3, a_4, b_4$, c and the initial values x_0, y_0, z_0 form the cipher keys. The sensitive tests with respect to all cipher keys have been carried out. To verify the sensitivity of key parameter K, the original plain-image $I = \left(I(i,j)\right)_{M\times N}$ is encrypted with $K = p$, $K = p - \Delta\delta$ and $K = p + \Delta\delta$ respectively while keeping the other key parameters unchanged. The corresponding encrypted images are denoted by I_1, I_2, I_3 respectively. The sensitivity coefficient to the parameter K is denoted by the following formula

$$P_s(K) = \frac{100}{2\times W\times N}\sum_{i,j}\left[N_s\left(I_1(i,j), I_2(i,j)\right) + N_s\left(I_1(i,j), I_3(i,j)\right)\right]$$

where $N_s(x,y) = 1$,if $x \neq y$, otherwise $N_s(x,y) = 0$, and $\Delta\delta$ is the perturbing value. $P_s(K)$ implies the sensitivity to the perturbation of parameter K. Table 1 shows the results of the sensitivity test where the initial key values are set to be

$a_1 = 1.11,\ b_1 = 10.22,\ a_2 = 20.33,\ b_2 = 30.44, a_3 = 40.55,\ b_3 = 50.66,$

$x_0 = 0.05,\ y_0 = 0.5, z_0 = 0.33, c = 0.7, a_4 = 60.77, b_4 = 70.88.$

The variations $\Delta\delta$ are 10^{-16} for x_0, y_0, z_0, c, 10^{-15} for a_1, b_1, and 10^{-14} for the other keys. The results in Table 1 imply that the cipher keys are strongly sensitive. It also implies from the results that the key space is more than 10^{178}, which is large enough to make brute-force attack infeasible.

Table 1 Results regarding the sensitivity to cipher keys

K	x_0	y_0	a_1	b_1
$P_s(K)$	99.60	99.59	99.63	99.63
K	a_2	b_2	a_3	b_3
$P_s(K)$	99.60	99.60	99.58	99.61
K	a_4	b_4	z_0	c_0
$P_s(K)$	99.61	99.57	99.61	99.58

4.2 Statistical Analysis

Shannon pointed out in his masterpiece the possibility to solve many kinds of ciphers by statistical analysis [6]. Therefore, passing the statistical analysis on cipher-image is of crucial importance for a cryptosystem. Indeed, an ideal cryptosystem should be robust against any statistical attack. In order to prove the security of the proposed encryption scheme, the following statistical tests are performed.

(i) Histogram. Encrypt the image Lena with one round, and then plot the histograms of plain-image and cipher-image as shown in Figs.1 (c)–(d), respectively. Fig.1 (d) shows that the histogram of the cipher-image is fairly uniform and significantly different from the histogram of the original image and hence it does not provide any useful information for the opponents to perform any statistical analysis attack on the encrypted image.

(ii) Correlation of adjacent pixels. To test the correlation between two adjacent pixels, the following performances are carried out. First, we select 6000 pairs of two adjacent pixels randomly from coefficient of the selected pairs using the following formulae:

$$Cr = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)D(y)}},\ \operatorname{cov}(x,y) = \frac{1}{T}\sum_{i=1}^{T}\left(x_i - E(x)\right)\left(y_i - E(y)\right),$$

$$E(x) = \frac{1}{T}\sum_{i=1}^{T} x_i, D(x) = \frac{1}{T}\sum_{i=1}^{T}\left(x_i - E(x)\right)^2,$$

where x, y are the gray-scale values of two adjacent pixels in the image and T is the total pairs of pixels randomly selected from the image. The correlations of two adjacent pixels in the plain-image and in the cipher-image are shown in Table 2.

Table 2 Correlation coefficients of two adjacent pixels

	Plain-image	Cipher-image
Horizontal	0.9488	0.0051
Vertical	0.9658	-0.0095
Diagonal	0.9219	-0.0081

4.3 Differential Attack

In general, attackers may make a slight change (e.g., modify only one pixel) of the plain-image to find out some meaningful relationships between the plain-image and the cipher-image. If one minor change in the plain-image will cause a significant change in the cipher-image, then the encryption scheme will resist the differential attack efficiently. To test the influence of only one-pixel change in the plain-image over the whole cipher-image, two common measures are used: number of pixels change rate (*NPCR*) and unified average changing intensity (*UACI*).

$$NPCR = \frac{1}{M \times N} \sum_{i,j} D(i, j) \times 100\%,$$

$$UACI = \frac{1}{M \times N \times 255} \sum_{i,j} \left| C_1(i, j) - C_2(i, j) \right| \times 100\%$$

where C_1, C_2 are the two cipher-images corresponding to two plain images with only one pixel difference, W and H are the width and height of the processed image, D is a bipolar array with the same size as image C_1. $D(i, j)$ is determined as: if $C_1(i, j) = C_2(i, j)$, then $D(i, j) = 0$, otherwise $D(i, j) = 1$.

NPCR measures the percentage of different pixel numbers between the two cipher-images whose plain-images only have one-pixel difference. *UACI* measures the average intensity of differences between the two cipher-images. To resist difference attacks, the values of *NPCR* and *UACI* should be large enough. The test of the plain-image is Lena. We randomly select 10 pixels and change the gray values with a difference of 1, for example, we replace the gray value 73 of the pixel at position (150,11) by 74, and get the *NPCR*=99.70%, *UACI*=39.10%. The mean values of the ten *NPCR* and *UACI* values are 99.78% and 39.13% respectively. The two measure values are exceptionally good undergoing only one round of encryption.

Acknowledgements This research is supported by Innovation and Entrepreneurship Training Program of Guangdong Colleges.

References

1. Schneier B. (1995). Cryptography: Theory and Practice. CRC Press, Boca Raton.
2. Kocarev L. (2001). Chaos-based cryptography: a brief overview. *IEEE Circuits and Systems Magazine*, 1, 6–21.
3. Fridrich J. (1998). Symmetric ciphers based on two-dimensional chaotic maps. *International Journal of Bifurcation and Chaos*, 8, 1259–1284.
4. Zhang G. J. & Liu Q. (2011). A novel image encryption method based on total shuffling scheme. *Optics Communications*, 284, 2775–2780.
5. Ye R. (2011). A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism. *Optics Communications*, 284, 5290–5298.
6. Shannon C. E. (1949). Communication theory of secrecy system. *Bell Syst. Tech. J*, 28, 656–715.
7. Li S. & Zheng X. (2002). Cryptanalysis of a chaotic image encryption method. In: Proc. IEEE Int. Symposium on Circuits and Systems, vol. II, 2002, pp. 708–711.
8. Liu J. M. & Qu Q. (2010). Cryptanalysis of a substitution-diffusion based on cipher using chaotic standard and logistic map. In: Third International Symposium on Information Processing, pp. 67-69.
9. Wang X. & He G. (2011). Cryptanalysis on a novel image encryption method based on total shuffling scheme. *Optics Communications*, 284, 5804–5807.