A Novel Chaos-Based Symmetric Image Encryption Using Bit-Pair Level Process

Rongjun Ge, Guanyu Yang, Jiasong Wu, Yang Chen, Gouenou Coatrieux, Limin Luo

▶ **To cite this version:**

Rongjun Ge, Guanyu Yang, Jiasong Wu, Yang Chen, Gouenou Coatrieux, et al.. A Novel Chaos-Based Symmetric Image Encryption Using Bit-Pair Level Process. IEEE Access, 2019, 7, pp.99470-99480. 10.1109/ACCESS.2019.2927415 . hal-02277196

HAL Id: hal-02277196

https://univ-rennes.hal.science/hal-02277196v1

Submitted on 10 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 20, 2019, accepted July 2, 2019, date of publication July 8, 2019, date of current version August 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927415

A Novel Chaos-Based Symmetric Image Encryption Using Bit-Pair Level Process

RONGJUN GE[1,2,3], GUANYU YANG[1,2,3], JIASONG WU[1,2,3], YANG CHEN[1,2,3,4], (Senior Member, IEEE), GOUENOU COATRIEUX[5], AND LIMIN LUO[1,2,3], (Senior Member, IEEE)

[1]Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Centre de Recherche en Information Biomedicale Sino-Francais (LIA CRIBs), F-35000 Rennes, France
[3]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 210096, China
[4]School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China
[5]Institut Mines-Telecom, Telecom Bretagne, INSERM U1101 LaTIM, 29238 Brest, France

Corresponding author: Yang Chen (chenyang.list@seu.edu.cn)

This research was supported in part by the State's Key Project of Research and Development Plan under Grant 2017YFA0104302, 2017YFC0109202 and 2017YFC0107900, in part by the National Natural Science Foundation under Grants 61871117 and 81530060. This work is also in part supported by the Postgraduate Research and Practice Innovation Program of Jiangsu Province (Grant KYCX17_0104).

ABSTRACT Nowadays, chaos-based image encryption is recognized as an effective choice in secure communications. This paper proposes a novel approach, which uses bit-pair level XOR, add, and rotation from top to bottom and enhances diffusion with pixel level XOR operator from the lower right corner to the upper left corner. Due to the bit-pair level encryption, the proposed scheme has ensured great security and high encryption speed. The chaotic series employed for encryption are obtained from the modified pulse-coupled spiking neurons circuit map. It is suitable for use encryption because of its abundant parameters: wide chaotic range, ergodicity, complexity, and high sensitivity. Furthermore, the results show the superiority of this scheme compared to the other four methods in terms of robustness to differential attack.

INDEX TERMS Chaos, image encryption, bit-pair level process, security.

I. INTRODUCTION

With the rapid development of multimedia technology and the widespread popularity of personal digital device and internet, the protection of images from unauthorized parties is a core issue. Indeed, compared to text data, images have some special properties such as bulk data capacity, high redundancy and strong correlation among adjacent pixels, which make conventional text encryption method unadapted. Many works have focused on appropriate image cryptosystems with a wide variety of strategies, such as Fourier transform [1], [2], wavelet transform [3], compressive sensing [4], Arnold transform [5], [6], SCAN [7] and chaos [8], [9]. Among these strategies, chaos-based image encryption has regarded as one of efficient and excellent encryption method due to the properties of chaotic systems, including high sensitivity to their initial values and control parameters, state ergodicity, pseudo randomness of evolution, structure complexity [11].

In chaos-based image encryptions, chaotic systems are generally used to generate key streams. Once chaotic systems and their initial conditions which are known as secret keys in encryption systems are decided, pseudo-random key streams are given. And encrypted images are produced by combining these key streams with plaintext images. Besides, a slight change of initial values or control parameters leads to significantly different outputs of chaotic systems and cause totally different key streams, which guarantees the security of cryptosystem. By leveraging the properties, like state ergodicity and structure complexity, chaotic systems further bring a long-term unpredictable encryption.

In 1989, chaos theory was first introduced to cryptosystem by British mathematician Matthews using logistic map [12]. This work demonstrates the feasibility of chaos-based encryption. Nowadays, chaos further draws considerable attention of researchers in image encryption [8]–[10], [13]–[30], due to the increasing security requirement of image data and properties of chaotic system. In most of these image encryption algorithms, confusion-diffusion structure is chosen as a general structure in designing image

The associate editor coordinating the review of this manuscript and approving it for publication was Aniello Castiglione.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

encryption scheme. This structure is in good accord with the confusion and diffusion properties of cryptography mentioned by Shannon in [31]. In confusion-diffusion structure, confusion means the ciphertext should be noise-like, while diffusion brings a slight change in plaintext carries a great change in ciphertext. These two properties guarantee the effectiveness and security of chaos-based cryptography. Confusion-diffusion structure was suggested by Fridrich in [14] that encrypted images by permutating pixel position and diffusing pixels value with chaos for the first time. Compared with permutating pixels, the permutation of bits of pixels provides higher security, though the computation cost may be increased. Therefore, the tradeoff between security and speed needs to be taken into account. In [27], Wang et al. chose pixels as permutation units. The pixels are confused via row and column permutations. After such reorganization, permuted pixels are XORed with the secret keys which corresponds to Logistic map, and then the chaotic iterative value is refreshed according to the encrypted pixels. Thus, the confusion and diffusion of the pixels are done simultaneously to get a high security. In [28], Zhou et al. proposed a chaotic system by combining existing chaotic maps and applied it in image encryption. The encryption scheme has a 4-round-encryption structure. Each encryption round is constituted of 5 steps, that are: random pixel insertion, row separation, 1D substitution, row combination and image rotation. Due to the random pixel insertion, the images are encrypted randomly, non-repeatedly and unpredictably. Compared to [27] and [28], Zhu *et al.* [29] further consider the 8 bit encoding of grayscale pixels , and permute the 4 most significant bits individually, whereas the 4 least significant bits are relocated as a whole to save time. The permutation is achieved based on the Arnold cat map the parameters of which are determined by Logistic map. In the diffusion phase, each pixel value is altered sequentially by the chaotic iterative value and the output of the Logistic map. This image encryption scheme reaches a high encryption speed, and needs at least 3 encryption rounds to resist differential attack. In [30], Xu et al. also works on pixel bit planes, considering the four higher bit planes as a group and the four lower bit planes as another group. In the a first time, two binary sequences are transformed from these two groups, and diffuse each other with chaos, cyclic shifts and the XOR operation. By next, elements of both binary sequences are swapped by the order from the piecewise linear chaotic maps. These two encryption steps perform secure via only one round.

Since typical chaotic systems have been studied thoroughly, it is securer to use a new chaotic system in encryption. The chaotic system used in this paper is obtained from a modified pulsed-coupled spiking neurons circuit (MPSNC) [32]. It is controlled by 6 independent sensitive parameters which can provide more rich keys selections and bring a larger key space. Furthermore, it also exhibits a wider range of chaotic iteration and parameter settings, and outputs a more complex chaotic series. All of these advantages support the suitability of MPSNC map in image encryption.

In this paper, a chaos-based symmetric image encryption using bit-pair level process is proposed. Differently to bit level and pixel level encryption, we propose to use bit-pair level process reaches a high security with fast encryption speed to reach a tradeoff. More clearly, our chaos based cryptosystem is designed in two steps. Firstly, each pixel is divided into 4 bit-pairs, and the whole image with size of $m \times n$ is transformed into a bit-pair level matrix with size of $2m \times 2n$. Every 16 adjacent pixels form a bit-pair block with size of 8×8. One block is XORed with the former block, and added with remainder after modular calculation of sum of elements in the former one. Then the block is further divided into 4 concentric regions from inside to outside. These 4 regions are rotated for several bit-pair controlled by the block itself and MPSNC map. Performing a multi-functionality, such a combined operation spreads the influence of former block and confuses the bit-pairs. In the second step, a XOR operator is designed using the MPSNC map and acts on the pixels to diffuse pixels from lower the right corner to the upper left corner. Thus, the weakness of the first step is overcome, and diffusion property is highly enhanced. In summary, these two steps guarantee the cryptosystem to meet the classic confusion and diffusion Shannon requirements, effectively and efficiently.

The rest of this paper is organized as follows. Section 2 introduces the MPSNC map. Section 3 describes our chaos based cryptosystem while Section 4 analyzes its performance based on the experiment results. Section 5 concludes the paper.

II. THE MPSNC MAP

A. INTRODUCTION OF MPSNC MAP

In the proposed algorithm, the MPSNC map is selected to generate the needed key streams and is defined as (1)-(3).

$$x(i+1) = F(x(i)) = \begin{cases} F_1(x(i)) \bmod p & \text{for odd } i \\ F_2(x(i)) \bmod p & \text{for even } i \end{cases} \tag{1}$$

$$F_1(x(i)) = \begin{cases} \dfrac{-k_{12} sin(2\pi rx(i)) + k_{11}(x_m(i) - \frac{1}{4}) + 1}{s} + x(i), \\ \qquad 0 \le x_m(i) < \dfrac{1}{2} \\ \dfrac{-k_{12} sin(2\pi rx(i)) - k_{11}(x_m(i) - \frac{1}{4}) + 1}{s} + x(i), \\ \qquad \dfrac{1}{2} \le x_m(i) < 1 \end{cases} \tag{2}$$

$$F_2(x(i)) = \begin{cases} \dfrac{-k_{22} sin(2\pi rx(i)) + k_{21}(x_m(i) - \frac{1}{4}) + 1}{s} + x(i), \\ \qquad 0 \le x_m(i) < \dfrac{1}{2} \\ \dfrac{-k_{22} sin(2\pi rx(i)) - k_{21}(x_m(i) - \frac{1}{4}) + 1}{s} + x(i), \\ \qquad \dfrac{1}{2} \le x_m(i) < 1 \end{cases} \tag{3}$$

where p is the denominate of fraction form of r in the lowest term, $x \in [0, p)$, $x_m(i) = x(i)\ mod\ 1$, and $|k_{11}|/4 + |k_{12}| < 1$, $|k_{21}|/4 + |k_{22}| < 1$.

B. PERFORMANCE EVALUATION OF MPSNC MAP

According to the relationship between parameter and dynamics stated in [32], the chaotic behavior of MPSNC map can be established with suitable parameter settings. In MPSNC map F, there are 6 independent parameters, known as k_{11}, k_{12}, k_{21}, k_{22}, s and r. In this section, the superiority of using map F in image encryption is analyzed considering bifurcation diagram, Lyapunov exponents, permutation entropy and sensitivity, with fixed parameters $k_{11} = 1.5$, $k_{21} = 2.2$, $k_{22} = 0.2$, $r = 0.2$ and $s = 0.3$. Notice that even though it is possible to access a rich and flexible selections of key space, in this section, only the parameter k_{12} and the initial value $x(1)$ are chosen as keys so as to introduce the proposed algorithm concisely and clearly.

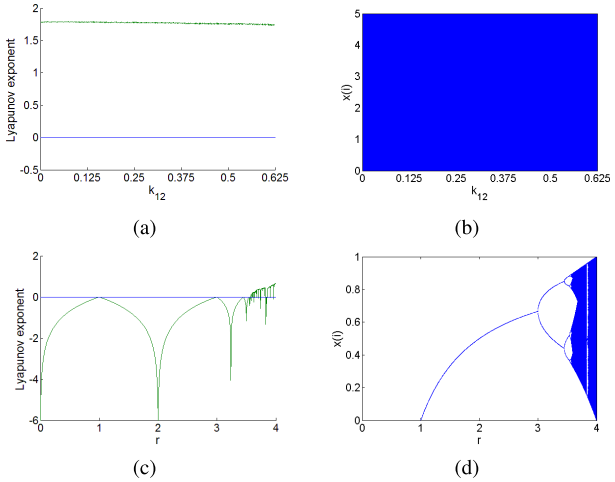


FIGURE 1. Chaotic behavior of MPSNC and logistical maps. (a) Bifurcation diagram of the MPSNC map for $k_{12} \in [0, 0.615)$; (b) Lyapunov exponent of the MPSNC map for $k_{12} \in [0, 0.615)$; (c) Bifurcation diagram of the Logistical map for $r \in [0, 4)$; (d) Lyapunov exponent of the logistical map for $r \in [0, 4)$.

1) BIFURCATION DIAGRAM AND LYAPUNOV EXPONENTS

As shown in Figs.1(a) and (b), MPSNC map F is chaotic and ergodic within the whole value range of $x(n)$, $[0, 5)$, in the entire domain of k_{12}, $[0, 0.625)$, with a uniform distribution and no periodic windows. Meanwhile the bifurcation diagram and Lyapunov exponent spectrum of Logistical map which is the most commonly used chaotic map in image encryption, defined as (4), is exhibited in Figs.1(c) and (d). Although the domain of parameter in the typical Logistical map is wider, its chaotic domain, $r \in [3.57, 4)$, is narrower. Besides, the ergodic range of Logistical map is only covered by $[0, 1)$. To conclude, the MPSNC chaotic map provides a larger key space than Logistical map.

$$x(i+1) = f(x(i)) = rx(i)(1 - x(i)) \tag{4}$$

2) PERMUTATION ENTROPY

The complexity of chaotic series can be measured by permutation entropy (PE) [33]. For normalized PE, the more closed

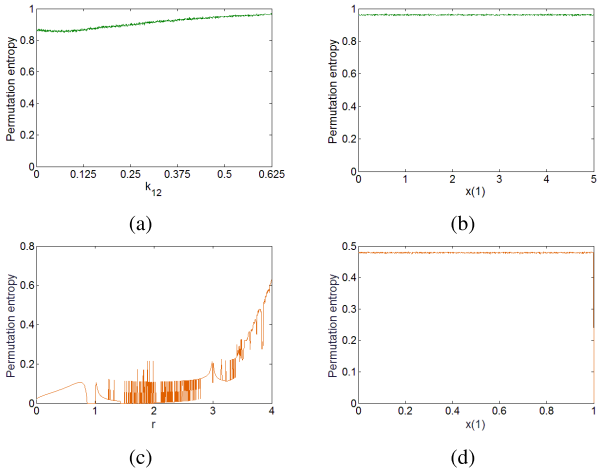


FIGURE 2. The MPSNC map gains higher PEs than the logistical map. (a) PEs of MPSNC map for $x(1) = 3.5$ and $k12 \in [0, 0.615)$; (b) PEs of MPSNC map for $k_{12} = 0.6$ and $x(1) \in [0, 5)$; (c) PEs of logistical map for $x(1) = 0.3$ and $r \in [0, 4)$; (d) PEs of logistical map for $r = 3.8$ and $x(1) \in [0, 1)$.

to 1, the more unpredictable chaotic series are generated. In Fig. 2 provides PE of both MPSNC chaotic and Logistical maps with different parameter and initial values. In each calculation, the embedding dimension $n = 6$, time lag $\tau = 1$, and series with 5000 iterative values were used. As it can be seen, compared with Logistical map, MPSNC chaotic map PE is much closer to 1. This means that more complex and unpredictable series can be generated than with Logistic map.

3) SENSITIVITY

Chaotic map used in image encryption should be sensitive to the initial value and the control parameters. In Figs. 3(a and (b) illustrate the sensitivity of MPSNC chaotic map to $x(1)$ and k_{12} , respectively. For the original orbit, the parameter $k_{12} = 0.6$, and the initial value $x(1) = 4.8$. In Fig. 3(a), the orbit for comparison has tiny change on initial point $x(1)$ with 10^{-15} bigger, while the two orbits in Fig. 3(b) have the same starting point but with different k_{12} values. Visually, both the orbits in Figs. 3(a) and (b) are visually distinguishable after nearly 20 iterations. Regarding the Logistical map, a change of 10^{-15} of the initial value of $x(1)$ and of the parameter r causes a different iterative orbit after nearly 80 iterations, as it can be seen in Figs. 3(c) and (d). It is clear that the MPSNC chaotic map is more sensitive to the variation of $x(1)$ and k_{12}.

III. NOVEL IMAGE ENCRYPTION ALGORITHM

The proposed image encryption algorithm includes two main steps. As we will see in the sequel in more details, each pixel is first divided into 4 bit-pairs. An image of size $m \times n$ is thus transformed into a $2m \times 2n$ matrix. In accordance with the order horizontally from the upper left to the lower right, each 8×8 bit-pair block is encrypted by applying a set of XOR, addition and rotation operations. In the next step, another XOR operation is carried among pixels, encrypted pixels and

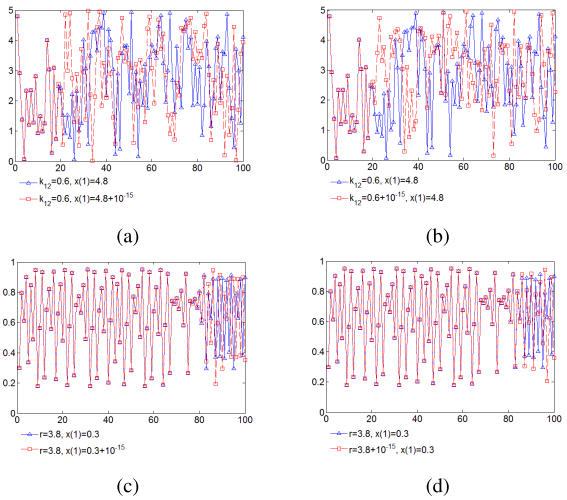


FIGURE 3. The MPSNC is more sensitivity to the initial value and the control parameters than the Logistical maps. (a) Sensitivity of MPSNC to $x(1)$; (b) Sensitivity of MPSNC to k_{12}; (c) Sensitivity of the Logistical map to $x(1)$; (d) Sensitivity of the Logistical map to r.

TABLE 1. Percentage of pixel information carried by different bits.

Bit	Percentage
P^8	50.1961%
P^7	25.0980%
P^6	12.5490%
P^5	6.27451%
P^4	3.13725%
P^3	1.56863%
P^2	0.784314%
P^1	0.392156%

a pseudo random sequence, from the lower right to the upper left vertically, to enhance the diffusion property.

A. BIT-PAIR LEVEL ROTATION

The gray value of one pixel of a grayscale image can be represented by (5).

$$P_{i,j} = P^8_{i,j} \times 2^7 + P^7_{i,j} \times 2^6 + P^6_{i,j} \times 2^5 + P^5_{i,j} \times 2^4 + P^4_{i,j} \times 2^3 + P^3_{i,j} \times 2^2 + P^2_{i,j} \times 2^1 + P^1_{i,j} \times 2^0 \quad (5)$$

where $P^s_{i,j}$ is the value of the s-th bit of the pixel located at the position (i, j).

Each of a pixel has different influence on pixel [29]. In particular, the percentage of pixel information carried by the s-th bit can be calculated with (6), and the results are recorded in Table 1.

$$I(s) = \frac{2^{s-1}}{\sum_{s=1}^{8} 2^{s-1}} \quad (6)$$

As it can be seen from Table 1, the higher 4 bits carry nearly 94.1176% of the image information. This means that the higher the bit, the more it contains information. In order to avoid it in the cipher image, our strategy consists in transforming each pixel into a bit-pair array with size of 2×2 as illustrated in Fig. 4.

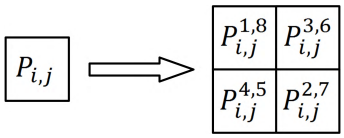


FIGURE 4. Transformation of pixel to bit-pair array.

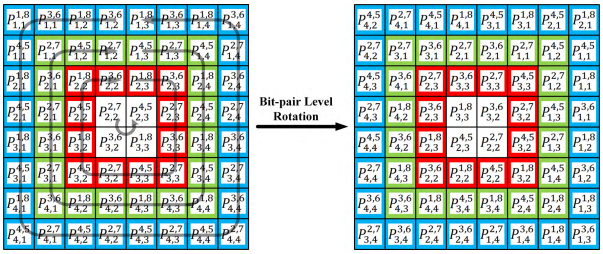


FIGURE 5. Illustration of the bit-pair level rotation parameterized with $(t_{r1}, t_{r2}, t_{r3}, t_{r4}) = (2, 8, 4, 19)$.

In Fig. 4, $P^{s,t}_{i,j} = P^s_{i,j} \times 2^1 + P^t_{i,j} \times 2^0$, and $P^{s,t}_{i,j}$ is a combination of lower bit and higher bit. This strategy can balance the pixel information carried by bits. Furthermore, the bit-pair level rotation is carried on the 8×8 bit-pair block constructed by 16 bit-pair array with size of 2×2 to confuse these 64 bit-pairs, as shown in Fig. 5. So the original image with size of $m \times n$ is changed into a matrix with size of $2m \times 2n$. And the bit-pair block composed of 64 bit-pairs is divided into 4 concentric regions from inside to outside. From the innermost region to the outermost region, every region is rotated in turn, depending on the control instructions t_{r1}, t_{r2}, t_{r3} and t_{r4} generated by (7)-(8). In addition, to enhance the security, the rotation direction is linked with the number of "1" of bit-pair in the corresponding region. If the parity is odd, the corresponding region is rotated counterclockwise, or clockwise rotation on the contrary.

$$x_r(i+1) = F(x_r(i)) \quad (7)$$

$$\begin{cases} t_{r1}(j) = floor[x_r(j + 2000) \times 10^{10}] \bmod 4 \\ t_{r2}(j) = floor[x_r(j + 2000 + L_r) \times 10^{10}] \bmod 12 \\ t_{r3}(j) = floor[x_r(j + 2000 + 2 \times L_r) \times 10^{10}] \bmod 20 \\ t_{r4}(j) = floor[x_r(j + 2000 + 3 \times L_r) \times 10^{10}] \bmod 28 \end{cases} \quad (8)$$

where x_r is a chaotic sequence, L_r is equal to $mn/16$, the "floor" operator rounds the number to the nearest integer towards minus infinity, $i = 1, 2, \ldots, 4 \times mn/16 + 2000$ and $j = 1, 2, \ldots, mn/16$.

Fig. 5 illustrates such a bit-pair level rotation under the parameterization such as t_{r1}, t_{r2}, t_{r3} and t_{r4} equal 2, 8, 4 and 19, as well as the number of "1" of the corresponding regions are odd, even, even and odd. Therefore, the 4 regions labeled with white, red, green and blue are counterclockwise rotated with 2 bit-pairs, 8 bit-pairs, 4 bit-pairs and 19 bit-pairs, respectively. One block rotation consists of 4 region rotations, with 4 control instructions. So that each 16 bit-pairs of 4 pixels only need in average 1 region rotation with 1 control instruction. For a image with size of $m \times n$, just $mn/4$ region

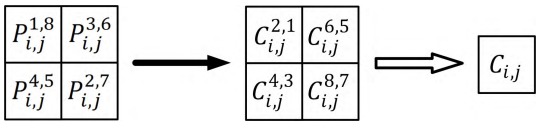


FIGURE 6. Transformation of bit-pair array to pixel.

rotations and control instructions is able to confuse it efficiently and effectively.

B. ENCRYPTION OPERATION

As stated above, the bit-pair level rotation is an efficient method to confuse the bit-pair matrix. However, the modification of one pixel can not spread to the whole image after rotation, due to the limitation of block. To do so, one original block is XORed with the previous rotated one. Then the XORed block is added with the remainder in the division of sum of elements in the previous rotated block by 4. And the initial block used to process the first block is got from the chaotic sequence taken from the former 2000 elements of chaotic sequence in (7) to simplify the encryption algorithm in this paper. The initial pixel block described as (9) is transformed into initial bit-pair block according to Fig. 4.

$$P0_{i,j} = floor[x_r(1984 + (i-1)\times 4 + j) \times 10^{10}] \; mod \; 256, \tag{9}$$

where $i = 1, 2, 3, 4$, and $j = 1, 2, 3, 4$.

The "XOR", "add" and initial block are very useful not only to diffuse the influence of the former pixels, but also to enhance the encryption effect to image with big region of the same pixel value.

After completing the "XOR", "add" and rotation operation on a block, move right with 8 bit-pairs in the row direction and get another block needs to be processed, until being up to the end horizontally. Then, move down with 8 bit-pairs and back to the far left and get another block needs to be processed. Repeatedly, the first encryption step is completed, while it comes to the end of the matrix with size of $2m \times 2n$. Finally, an image is reconstructed by transforming this processed matrix with size of $2m \times 2n$ into the matrix with size of $m \times n$. Finally, it is important to note that the transformation of bit-pair array to pixel is not just simply the reverse process of the transformation in Fig. 4. In Fig. 6, $C_{i,j}$ is used to represent the pixel after the first encryption step, and is transformed from the processed bit-pair array $C^{s,t}_{i,j}$ with (10).

$$C_{i,j} = C^{8,7}_{i,j} \times 64 + C^{6,5}_{i,j} \times 16 + C^{4,3}_{i,j} \times 4 + C^{2,1}_{i,j} \tag{10}$$

After the first encryption step, a change for pixels can spread out to the pixels behind them, as a result of the processing order from the upper left corner to the lower right corner. Here, a XOR operation is carried from the lower right to the upper left vertically to enhance the diffusion property.

First of all, the pixels of the image I after the first encryption step form a one-dimensional array denoted as Q in accordance with the order vertically from the lower right to the upper left. The XOR is operated on Q using (11).

$$\begin{cases} x_d(i+1) = F(x_d(i)) \\ t_d(j) = floor[x_d(j+2000) \times 10^{10}] \; mod \; 256 \\ E(j) = [t_d(j) \times 10^3 + E(j+1)] \; mod \; 256 \\ \quad \oplus [Q(j) + t_d(j) \times 10^2] \; mod \; 256 \; \oplus \; t_d(j) \end{cases} \tag{11}$$

where x_d is a chaotic sequence and "*floor*" rounds the number to the nearest integer towards minus infinity, $i = 1, 2, \ldots, m \times n + 2000$ and $j = m \times n, m \times n - 1, \ldots, 1$. Besides, $Q(m \times n)$ is refreshed by $[\sum_{u=m-7}^{m} \sum_{v=n-7}^{n} I_{u,v} - I_{m,n}] \; mod \; 256$ in case of the situation that change exists in the last block not the last pixel of I.

In (11), $E(j)$ denotes the j-th pixel value after the second encryption step, and $E(m \times n + 1)$ is defined in (12). The encrypted image is built by transforming the one-dimensional cipher array E into two-dimensional $m \times n$ cipher matrix. So, the change of later pixels is also able to influence former pixels, after the second encryption step:

$$E(m \times n + 1) = \begin{cases} F_1(x_E) \; mod \; p & for \; odd \; l \\ F_2(x_E) \; mod \; p & for \; even \; l \end{cases} \tag{12}$$

where $x_E = (\sum_{u=m-31}^{m} \sum_{v=n-31}^{n} I_{u,v} - I_{m,n}) \; mod \; 256 \; / 256 \times p$, p is the denominate of fraction form of control parameter r in the lowest term, $l = (\sum_{u=m-31}^{m} \sum_{v=n-31}^{n} I_{u,v} - I_{m,n}) \; mod \; 256$, and functions F_1 and F_2 are defined in (2)-(3).

C. DECRYPTION OPERATION

The decryption procedure is the reverse process of the encryption described above. That is, firstly, making a reverse diffusion on encrypted image using (13), then transforming Q to $m \times n$ image I, and refreshing $I_{m,n}$ with $(\sum_{p=m-7}^{m} \sum_{q=n-7}^{n} I_{p,q} - I_{m,n}) mod \; 256$. Finally, bit-pair level rotation, "add" and "XOR" are carried out on the resulting image in accordance with the order horizontally from the lower right corner to the upper left corner.

$$\begin{cases} x_d(i+1) = F(x_d(i)) \\ t_d(j) = floor[x_d(j+2000) \times 10^1 0] \; mod \; 256 \\ Q(j) = \{E(j) \oplus t_d(j) \oplus [t_d(j) \times 10^3 + E(j+1)] \; mod \; 256 \\ \quad +256 - t_d(j) \times 10^2 \; mod \; 256\} \; mod \; 256 \end{cases} \tag{13}$$

where x_d is a chaotic sequence, "*floor*" rounds the number to the nearest integer towards minus infinity, $i = 1, 2, \ldots, m \times n + 2000$ and $j = 1, 2, \ldots, m \times n$.

IV. EXPERIMENT AND ANALYSIS

In this section, some experiments and analyses are performed. All the tests in this paper are conducted under MATLAB 8.1.0.604 (R2013a) in a laptop with the Windows 10 operating system of 64-bit, Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz and 8 GB RAM. Besides, k_{12} and $x(1)$ of MPSNC map F used in the first encryption step and the second encryption step are chosen as keys, while the other parameters

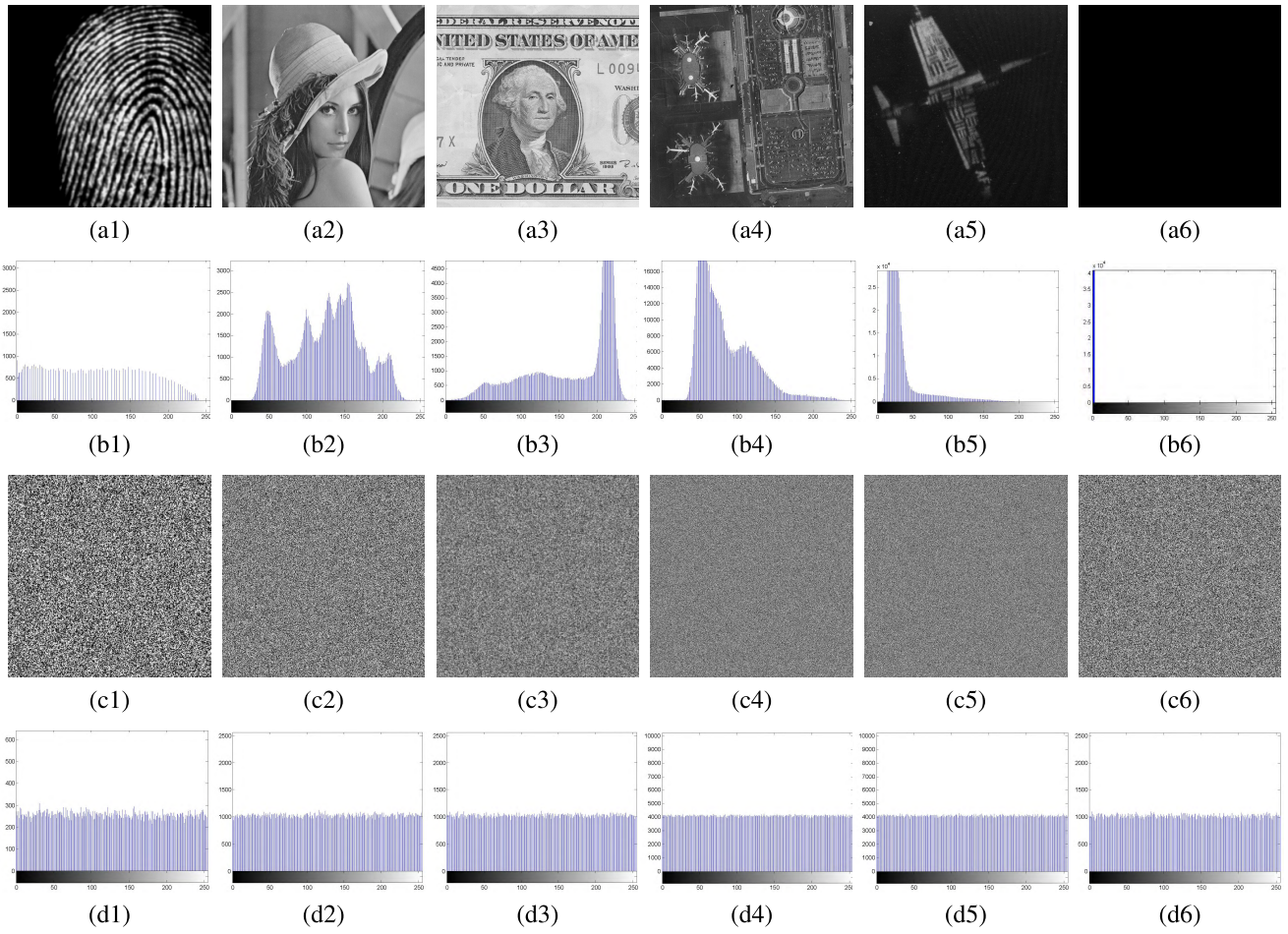


FIGURE 7. Encryption and histograms. Our proposed algorithm is able to successfully encrypt the meaningful image into noise-like and make the histograms uniform to obscure the information, even for the black image. (a1)-(a6) The original images; (b1)-(b6) Histograms of the original images; (c1)-(c6) The cipher images; (d1)-(d6) Histograms of the cipher images.

are fixed as $k_{11} = 1.5$, $k_{21} = 2.2$, $k_{22} = 0.2$, $r = 0.2$ and $s = 0.3$. In order to distinguish these two key groups utilized in different phases, $(k_{12}, x(1))$ used in the first step and the second step are denoted as $key_1 = (k_{12}^r, x^r(1))$ and $key_2 = (k_{12}^d, x^d(1))$, respectively.

A. ENCRYPTED RESULTS AND CORRESPONDING HISTOGRAMS

As examples to reveal the effect and security of the proposed image encryption algorithm visually, the encryption tests of several images, including Fingerprint, Lena, Dollar, Airfield2, AirplaneU2 and Black image, are given in Fig. 7. The keys are set as $key_1 = (0.6, 2.71)$ and $key_2 = (0.35, 4.8)$. It is clear from Figs. 7(c1) - (c6), the encrypted images are all like the noise, and impossible to find the content of the original versions.

Furthermore, a histogram is used to reflect the distribution of pixel values of an image. The histograms of cipher images in Fig. 7 is highly uniform, which are significantly different from the corresponding histograms of the original images. A uniform histogram is good at resisting statistical attacks [16], [34]. Thus it can be said that the proposed algorithm is good at transforming images into noise-like encrypted images with uniform distributed histograms, even the plain image is whole black.

B. KEY SPACE

As discussed in the description of the proposed algorithm, the keys are divided into two steps. Meanwhile, the chaotic map utilized in the scheme has 6 independent control parameters and 1 initial value. If the precision of the control parameters and initial values reaches 10^{-15}, the key space can be up to $10^{15\times7\times2} = 10^{210}$ when all the parameters and initial values are regarded as keys, theoretically. In the experiments of this paper, only $(k_{12}^r, x^r(1))$ and $(k_{12}^d, x^d(1))$ are selected as keys for illustrating the process of encryption concisely. In the condition that several parameters are fixed as $k_{11} = 1.5, k_{21} = 2.2, k_{22} = 0.2, r = 0.2$ and $s = 0.3$, the rest parameters like k_{12}^r and k_{12}^d are in range of $(0, 0.625)$, $x^c(1)$ and $x^d(1)$ belong to $[0, 5)$. Besides, the length of each key is set as 15 decimals. So the key space of the keys recommended reaches $(0.625\times10^{15}\times5\times10^{15})^2 = 9.7656\times10^{60}$. It is larger than 2^{100} which is the demand suggested in [11] for resisting brute-force attack, implying infeasibility of brute-force attack to the proposed algorithm.

TABLE 2. The RMSD between the cipher images with $x^r(1) = 2.71$, $k_{12}^d = 0.35$, $x^d(1) = 4.8$ and tiny fluctuation of 10^{-15} in k_{12}^r. Our algorithm is extremely sensitivity to the encryption key k_{12}^r with the high RMSDs between its two cipher images of tiny different k_{12}^r.

k_{12}^r	0.6 vs $0.6+10^{-15}$	$0.6+10^{-15}$ vs $0.6+2\times10^{-15}$	$0.6+2\times10^{-15}$ vs $0.6+3\times10^{-15}$	$0.6+3\times10^{-15}$ vs $0.6+4\times10^{-15}$	$0.6+4\times10^{-15}$ vs $0.6+5\times10^{-15}$
RMSD	104.3933	104.3357	104.6767	104.4691	104.5163

TABLE 3. The RMSD between the cipher images with $k_{12}^r = 0.6$, $k_{12}^d = 0.35$, $x^d(1) = 4.8$ and tiny fluctuation of 10^{-15} in $x^r(1)$. Our algorithm is extremely sensitivity to the encryption key $x^r(1)$ with the high RMSDs between its two cipher images of tiny different $x^r(1)$.

$x^r(1)$	2.71 vs $2.71+10^{-15}$	$2.71+10^{-15}$ vs $2.71+2\times10^{-15}$	$2.71+2\times10^{-15}$ vs $2.71+3\times10^{-15}$	$2.71+3\times10^{-15}$ vs $2.71+4\times10^{-15}$	$2.71+4\times10^{-15}$ vs $2.71+5\times10^{-15}$
RMSD	104.3418	104.2743	104.3090	104.5978	104.5577

TABLE 4. The RMSD between the cipher images with $k_{12}^r = 0.6$, $x^r(1) = 2.71$, $x^d(1) = 4.8$ and tiny fluctuation of 10^{-15} in k_{12}^d. Our algorithm is extremely sensitivity to the encryption key k_{12}^d with the high RMSDs between its two cipher images of tiny different k_{12}^d.

k_{12}^d	0.35 vs $0.35+10^{-15}$	$0.35+10^{-15}$ vs $0.35+2\times10^{-15}$	$0.35+2\times10^{-15}$ vs $0.35+3\times10^{-15}$	$0.35+3\times10^{-15}$ vs $0.35+4\times10^{-15}$	$0.35+4\times10^{-15}$ vs $0.35+5\times10^{-15}$
RMSD	104.3967	104.2907	104.6183	104.4935	104.5714

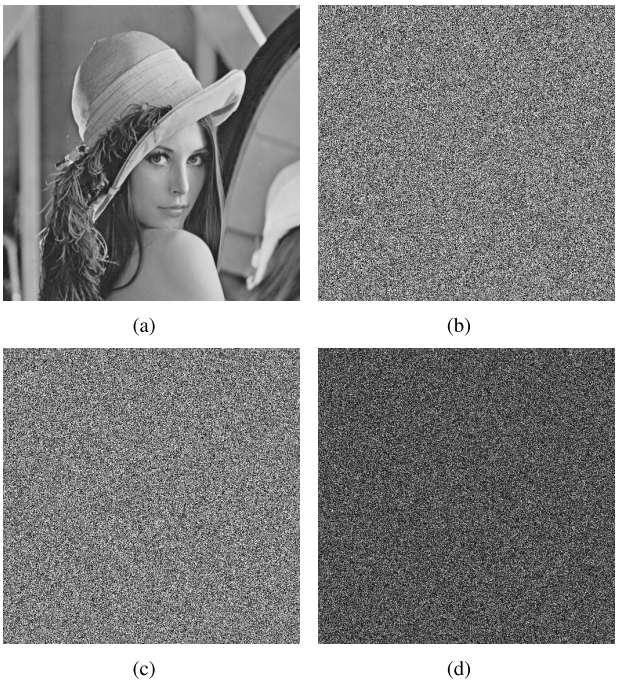


FIGURE 8. Key sensitivity analysis of the encryption. Our algorithm is sensitive to the encryption keys which are the guarantee of security. (a) The original image; (b) The cipher image I_{e1} with $key_1 = (0.6, 2.71)$ and $key_2 = (0.35, 4.8)$; (c) The cipher image I_{e2} with $key_1 = (0.6 + 10^{-15}, 2.71)$ and $key_2 = (0.35, 4.8)$; (d) Difference between two cipher images $I_{dif} = |I_{e1} - I_{e2}|$.

C. KEY SENSITIVITY

The algorithm is designed to be sensitive to the encryption keys which are the guarantee of security. In order to verify this, the images encrypted using different keys are compared. The original keys are set as $key_1 = (0.6, 2.71)$ and $key_2 = (0.35, 4.8)$. Encrypted image in Fig. 8(b) is obtained using the original keys to encrypt Fig. 8(a). For comparison, one of the keys used to encrypt Fig. 8(a) into Fig. 8(c) is modified. The modified key k_{12}^r is changed from 0.6 to $0.6 + 10^{-15}$ while the other keys remain the original value. The proportion of different pixels between Figs. 8(b) and (c) reaches 99.5693%. Fig. 8(d) shows the plot of the difference between these two ciphers, using (14).

$$I_{dif} = |I_{e1} - I_{e2}|, \tag{14}$$

where I_{e1} and I_{e2} are two encrypted images.

Furthermore, root mean square difference (RMSD) defined in (15) is used to reflect the sensitivity to encryption keys, numerically. Fig. 8(a) is encrypted with increasing k_{12}^r, $x^r(1)$, k_{12}^d and $x^d(1)$ by 10^{-15} a time, and the corresponding RMSDs are displayed in Tables 2-5. It can be seen that all the RMSDs are greater than 104, which reveals the huge difference between each pair of ciphers.

$$RMSD = \left\{ \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} [I_{e1}(i,j) - I_{e2}(i,j)]^2 \right\}^{1/2} \tag{15}$$

All of these significant differences mentioned above between the encrypted images are caused by a tiny change 10^{-15} on encryption keys. This means that the proposed algorithm is extremely sensitive to the fluctuation in encryption keys.

In a symmetric encryption algorithm, the encrypted image can only be decrypted correctly using the same keys as the encryption keys. Meanwhile, the algorithm must be sensitive to the decryption keys. Fig. 9(b) are the encrypted results of Fig. 9(a), using $key_1 = (0.6, 2.71)$ and $key_2 = (0.35, 4.8)$. The decrypted results using the correct symmetric decryption keys are displayed in Fig. 9(c). For comparison, decryption key $x^d(1)$ is changed to $4.8 + 10^{-15}$, while the other

TABLE 5. The RMSD between the cipher images with $k_{12}^r = 0.6$, $x^r(1) = 2.71$, k_{12}^d and tiny fluctuation of 10^{-15} in $x^d(1)$. Our algorithm is extremely sensitivity to the encryption key $x^d(1)$ with the high RMSDs between its two cipher images of tiny different $x^d(1)$.

$x^d(1)$	4.8 vs $4.8+10^{-15}$	$4.8+10^{-15}$ vs $4.8+2\times10^{-15}$	$4.8+2\times10^{-15}$ vs $4.8+3\times10^{-15}$	$4.8+3\times10^{-15}$ vs $4.8+4\times10^{-15}$	$4.8+4\times10^{-15}$ vs $4.8+5\times10^{-15}$
RMSD	104.6533	104.4285	104.4373	104.5215	104.5028

TABLE 6. The information entropy of the original image and encrypted image in Fig. 7. Our encrypted images reaches the high information entropy exceeding at least 7.997604, which means effectively embedding the redundant information for obscuring the meaningful information in the original image.

Image	Fingerprint	Lena	Dollar	Airfield2	AirplaneU2	Black
Original image	5.196340	7.445507	6.978502	6.830330	5.641454	0.000000
Cipher image	7.997604	7.999216	7.999377	7.999844	7.999810	7.999245

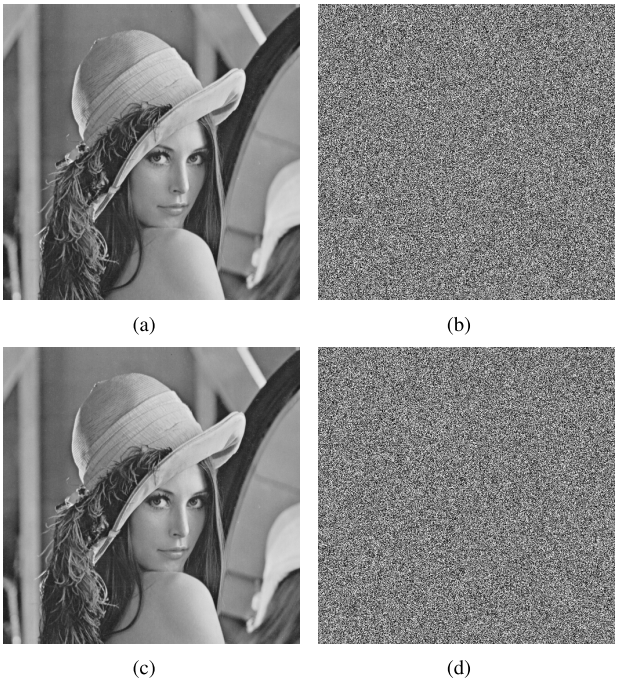


FIGURE 9. Key sensitivity analysis of the decryption. Our algorithm is sensitive to the keys in decryption, which ensures the security. (a) The original image; (b) The cipher image with $key_1 = (0.6, 2.71)$ and $key_2 = (0.35, 4.8)$; (c) The decrypted image with the correct keys; (d) The decrypted image with the wrong keys $key_1 = (0.6, 2.71)$ and $key_2 = (0.35, 4.8 + 10^{-15})$.

keys remain unchanged. Utilizing this reconstructed key set, Fig. 9(b) is decrypted unsuccessfully. This failed result is displayed in Fig. 9(d) with error rate 99.6075%. We can see that the algorithm is hypersensitive to decryption keys, and the decryption keys need to be the same as the encryption ones for the security.

D. INFORMATION ENTROPY

For describing the information redundancy and the feature of randomness, Shannon [31] has introduced entropy into information theory as information entropy (IFE). In an image encryption system, IFE is given by (16).

$$H(l) = \sum_{i=0}^{2^N-1} p(l_i) log_2 \frac{1}{p(l_i)} \tag{16}$$

where $p(l_i)$ represents the probability of gray level l_i in a N-bit image.

The larger the IFE is, the more randomness and the higher information redundancy that the image preforms. For a grayscale image with 2^8 gray levels, the maximum value of IFE can reach 8. In Table 6, the IFEs of the original images and the encrypted images shown in Fig. 7 are listed. It can be found that the IFE of the image after encryption is very close to 8, even the original image is a black one, which means the proposed algorithm performs well on improving the information entropy.

E. CORRELATION

The relationship between the variables can be measured by correlation coefficient which is described by (17).

$$\begin{cases} \rho_{xy} = \dfrac{E\{[x - E(x)][y - E(y)]\}}{\sqrt{D(x)}\sqrt{D(y)}} \\ E(x) = \dfrac{1}{l}\sum_{i=1}^{l} x_i \\ D(x) = \dfrac{1}{l}\sum_{i=1}^{l} [x_i - E(x)]^2 \end{cases} \tag{17}$$

In general, there is a strong correlation between adjacent pixels in a meaningful image. Our encryption operation have an strong ability to break this correlation. Because the selected pixels are random and the number of the selected pixels is finite, using correlation coefficient calculated from a group of pixels to demonstrate the correlation of an image is insufficient. Thus, we decide to randomly select 5 groups of 2000 pixel-pairs from the original image and encrypted image in each direction. Then the corresponding correlation coefficients and the averages of the absolute values of the correlation coefficients along the same direction are calculated. In Table 7, the calculated results of the original image and encrypted image of image Lena are listed. Besides, Fig. 10 shows the correlation distribution of the original and encrypted versions of image Lena in all directions.

From Table 7 and Fig. 10, it is easy to see that the proposed algorithm can greatly break the correlation between adjacent pixels.

TABLE 7. The correlation coefficients of the original image and encrypted image of image Lena. Our encryption excellently breaks this correlation in the meaningful image.

	Original image			Cipher image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
1	0.9761	0.9862	0.9561	-0.0058	-0.0068	-0.0103
2	0.9710	0.9834	0.9621	-0.0048	0.0090	-0.0011
3	0.9708	0.9845	0.9614	-0.0016	0.0032	0.0009
4	0.9735	0.9843	0.9593	-0.0052	0.0057	0.0065
5	0.9744	0.9822	0.9513	-0.0095	0.0073	0.0040
Average of absolute	0.9732	0.9841	0.9580	0.0054	0.0064	0.0046

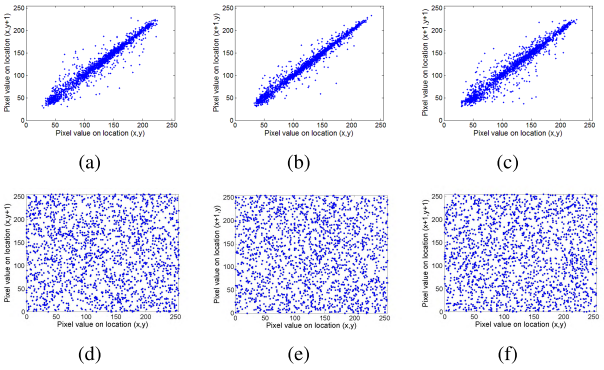


FIGURE 10. Correlation distribution. Our encryption successfully makes correlation distribution uniform, indicating breaking the correlation in the meaningful image. (a), (b) and (c) Correlation of two horizontally, vertically, and diagonally adjacent pixels of original Lena image, respectively; (d), (e) and (f) Correlation of two horizontally, vertically, and diagonally adjacent pixels of cipher Lena image, respectively.

F. RESISTANCE TO DIFFERENTIAL ATTACK

In order to study the resistance of encryption algorithm to differential attack, the number of pixels change rate (NPCR) and the unified average changing intensity (UACI) were calculated between the cipher images encrypted from two original images that have tiny difference. If a tiny modification in original image cause a significant change in encrypted image, the differential attack is considered invalid. Let $I_{e1}(i,j)$ and $I_{e2}(i,j)$ denote the pixels located at (i,j) in encrypted images I_{e1} and I_{e2} with size of $m \times n$. And the NPCR and UACI are defined by (18)-(20).

$$D(i,j) = \begin{cases} 1 & I_{e1}(i,j) \neq I_{e2}(i,j) \\ 0 & I_{e1}(i,j) = I_{e2}(i,j) \end{cases} \tag{18}$$

$$NPCR = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)}{M \times N} \times 100\% \tag{19}$$

$$UACI = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{|I_{e1}(i,j) - I_{e2}(i,j)|}{255}}{M \times N} \times 100\% \tag{20}$$

Figs. 7(a1) - (a6) and their modified versions obtained by changing the last bit of the first, middle and last pixels are chosen as the plain images. The corresponding NPCRs and UACIs at one round of encryption are listed in Tables 8 and 9, compared with four existing encryption algorithms. As can be seen, NPCR and UACI of the proposed algorithm respectively exceed 99.5968% and 33.4038%, respectively, which performs much better than the others. Furthermore, the proposed algorithm also gets higher average NPCR 99.7641% and average UACI 33.6158%, compared with 99.5415% and 33.3274% of Ref. [27], 99.6229% and 33.5813% of Ref. [28], 0.2312% and 0.0781% of Ref. [29], and 99.6085% and 33.3947% of Ref. [30]. Thus, these comparison results support the fact that the proposed algorithm has a superior property in resisting differential attack.

TABLE 8. Comparative study on NPCR. Our proposed algorithm gains highest NPCR for most robustly resisting the differential attack.

Image	Position of modified pixel	**Proposed algorithm**	Ref. [27]	Ref. [28]	Ref. [29]	Ref. [30]
Fingerprint	(1,1)	0.996033	0.993622	0.994202	0.003998	0.996338
	(128,128)	0.996124	0.993515	0.996826	0.012466	0.995941
	(256,256)	1.00	0.995880	0.998291	0.002060	0.995956
Lena	(1,1)	0.996269	0.993645	0.996929	0.006367	0.995960
	(256,256)	0.996040	0.994873	0.995598	0.000881	0.995846
	(512,512)	1.00	0.995853	0.998062	0.001396	0.996319
Dollar	(1,1)	0.996033	0.995060	0.996101	0.002857	0.995975
	(256,256)	0.998077	0.996063	0.994473	0.001781	0.996082
	(512,512)	1.00	0.995945	0.997200	0.000935	0.995975
Airfield2	(1,1)	0.996143	0.995821	0.996091	0.000492	0.996101
	(512,512)	0.996602	0.995820	0.996333	0.000437	0.996101
	(1024,1024)	1.00	0.996092	0.997079	0.000838	0.996107
AirplaneU2	(1,1)	0.996100	0.995690	0.995616	0.000620	0.996140
	(512,512)	0.996109	0.995496	0.996861	0.000008	0.995876
	(1024,1024)	1.00	0.996194	0.997087	0.000454	0.996039
Black	(1,1)	0.995968	0.996025	0.996075	0.003250	0.996090
	(256,256)	0.998043	0.995766	0.995239	0.001637	0.996471
	(512,512)	0.999996	0.996117	0.994057	0.001133	0.996208
Average		0.997641	0.995415	0.996229	0.002312	0.996085

TABLE 9. Comparative study on UACI. Our proposed algorithm gains highest UACI for most robustly resisting the differential attack.

Image	Position of modified pixel	**Proposed algorithm**	Ref. [27]	Ref. [28]	Ref. [29]	Ref. [30]
Fingerprint	(1,1)	0.335371	0.330781	0.337452	0.001342	0.333363
	(128,128)	0.334084	0.331958	0.337913	0.004134	0.333657
	(256,256)	0.334263	0.332643	0.339175	0.000671	0.333657
Lena	(1,1)	0.335119	0.334079	0.340017	0.002162	0.334811
	(256,256)	0.337580	0.332442	0.340809	0.000302	0.333596
	(512,512)	0.337922	0.334242	0.333755	0.000496	0.335377
Dollar	(1,1)	0.334931	0.335613	0.337421	0.000989	0.333899
	(256,256)	0.334192	0.334573	0.336034	0.000618	0.334752
	(512,512)	0.338521	0.333907	0.336414	0.000316	0.333900
Airfield2	(1,1)	0.334551	0.333467	0.336024	0.000179	0.334842
	(512,512)	0.334912	0.336444	0.327836	0.000153	0.334842
	(1024,1024)	0.337206	0.333871	0.334910	0.000276	0.334598
AirplaneU2	(1,1)	0.334993	0.332269	0.332589	0.000208	0.334855
	(512,512)	0.334997	0.331782	0.336921	0.000001	0.328820
	(1024,1024)	0.345540	0.334567	0.336922	0.000155	0.334571
Black	(1,1)	0.334237	0.328457	0.330628	0.001161	0.334176
	(256,256)	0.334038	0.332896	0.335337	0.000516	0.333550
	(512,512)	0.338381	0.334947	0.334479	0.000388	0.333787
Average		0.336158	0.333274	0.335813	0.000781	0.333947

TABLE 10. Comparative study on speed of one round encryption. The proposed algorithm gets a higher one round encryption speed than Ref. [27], Ref. [28], Ref. [30]. It is also more quicker than Ref. [29], because it only needs one round process for effective encryption while Ref. [29] need at least three rounds.

Image	Size	**Proposed algorithm**	Ref. [27]	Ref. [28]	Ref. [29]	Ref. [30]
Fingerprint	256×256	0.2219s	0.2619s	0.5167s	0.1148s	0.7775s
Lena	512×512	0.8772s	1.0357s	2.0353s	0.5089s	3.0935s
Airfield2	1024×1024	3.7637s	4.2695s	8.4618s	2.3434s	12.9843s

G. RESISTANCE TO BLIND DECRYPTION

Our excellent cryptosystem is robust enough to withstand blind decryption. In order to address it, the decryption of the cipher image with correct and wrong keys is carried out.

The mean square error (MSE) between the original image and the decrypted image is used to judge the decryption. MSE is defined as (21).

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[I_o(i,j) - I_d(i,j)]^2}{M \times N} \times 100\% \qquad (21)$$

where $I_o(i,j)$ and $I_d(i,j)$ are the pixels at grid (i,j) of original image I_o and decrypted image I_d with size of $m \times n$.

Two hundred and fifty different key combinations of $(k_{12}^r, x^r(1)) and (k_{12}^d, x^d(1))$ are used, with the correct keys $key_1 = (0.6, 2)$ and $key_2 = (0.35, 4)$. The plots of the corresponding MSE are displayed in Fig. 11, which indicates that only the MSE of the correct keys is equal to 0, and all the others nearly reach 8000. Therefore, the proposed algorithm proves to be extremely robust to withstand blind decryption.

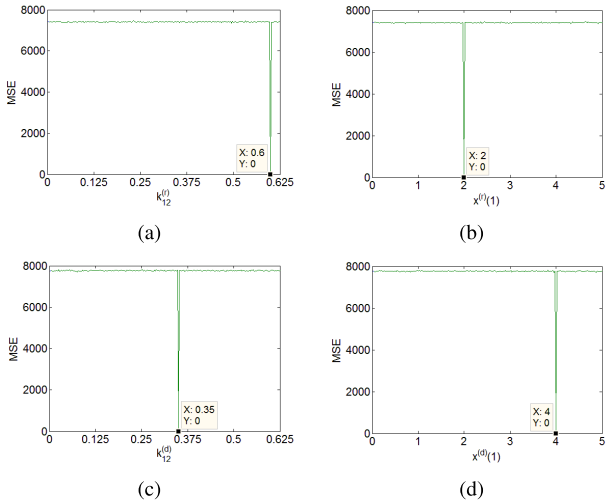


FIGURE 11. Analysis of MSE. Our excellent cryptosystem is robust to withstand the blind decryption. (a) The MSE curve with various values of k_{12}^r; (b) The MSE curve with various values of $x^r(1)$; (c) The MSE curve with various values of k_{12}^d; (d) The MSE curve with various values of $x^d(1)$.

H. SPEED PERFORMANCE

Besides security, speed is another important requirement of image encryption. Here, the images with different sizes (256×256, 512×512 and 1024×1024) are encrypted by the proposed encryption algorithm and the methods in Ref. [27], Ref. [28], Ref. [29] and Ref. [30] with 100 times to calculate the average encryption time. It is found from Table 9 that the time of one encryption round the proposed algorithm consumes is shorter than that for the methods in Ref. [27], Ref. [28] and the method in Ref. [30], but longer than that of Ref. [29]. However, Ref. [29] requires at least three encryption rounds to guarantee high NPCR and UACI as demonstrated in its literature. In summary, the proposed one gets the highest speed for image encryption.

V. CONCLUSION

In this paper, a novel chaos-based symmetric image encryption scheme has been proposed. It is realized by two steps: bit-pair level process and pixel level diffusion. First, each pixel is divided into 4 bit-pairs, and the image is transformed into a bit-pair matrix. For each 8×8 bit-pair block, it is XORed with the previous processed one and added with the remainder in the division of sum of elements in the previously processed block by 4. Then, four regions from inside to outside of the block are rotated in accordance with the orders obtained from the block itself and chaotic map. This operation not only relocates the bits of pixels, but also spreads the influence of former pixels to latter ones. In the second step, an XOR operation is applied on the pixels from the lower right corner to the upper left corner in order to enhance the diffusion property. Experiment results and analysis show that the proposed algorithm reaches a high security, even the object is a black image. Especially compared with Ref. [27], Ref. [28], Ref. [29] and Ref. [30], it gets higher NPCR and UACI after one round of encryption. Furthermore, it also encrypts image faster than the methods in Ref. [27], Ref. [28], Ref. [29] and Ref. [30] due to bit-pair level process, especially bit-pair level rotation. Moreover, the chaotic series utilized in encryption is obtained from the MPCSNC map. The advantages of this chaotic map lie in its abundant parameters, wide chaotic range, ergodicity, structure complexity and high sensitivity, which all prove the superiority of this map in encryption.

ACKNOWLEDGMENT

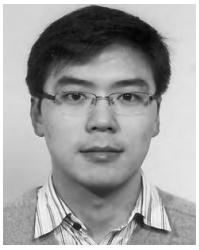
Rongjun Ge and Guanyu Yang contributed equally to this work.

REFERENCES

[1] B. M. Hennelly and J. T. Sheridan, "Image encryption and the fractional Fourier transform," *Opt.-Int. J. Light Electron Opt.*, vol. 114, no. 6, pp. 251–265, 2003.
[2] J. B. Lima and L. F. G. Novaes, "Image encryption based on the fractional Fourier transform over finite fields," *Signal Process.*, vol. 94, pp. 521–530, Jan. 2014.
[3] G. Bhatnagar, Q. M. J. Wu, and B. Raman, "Discrete fractional wavelet transform and its application to multiple encryption," *Inf. Sci.*, vol. 223, pp. 297–316, Feb. 2013.
[4] R. Huang, K. H. Rhee, and S. Uchida, "A parallel image encryption method based on compressive sensing," *Multimedia Tools Appl.*, vol. 223, no. 1, pp. 71–93, Sep. 2014.
[5] W. Chen, C. Quan, and C. J. Tay, "Optical color image encryption based on arnold transform and interference method," *Opt. Commun.*, vol. 282, no. 18, pp. 3680–3685, Sep. 2009.
[6] Z. Liu, M. Gong, Y. Dou, F. Liu, S. Lin, M. A. Ahmad, J. Dai, and S. Liu, "Double image encryption by using Arnold transform and discrete fractional angular transform," *Opt. Lasers Eng.*, vol. 50, no. 2, pp. 248–255, Feb. 2012.
[7] S. S. Maniccam and N. G. Bourbakis, "Image and video encryption using SCAN patterns," *Pattern Recognit.*, vol. 37, no. 4, pp. 725–737, Apr. 2004.
[8] X. Wang, L. Liu, and Y. Zhang, "A novel chaotic block image encryption algorithm based on dynamic random growth technique," *Opt. Lasers Eng.*, vol. 66, pp. 10–18, Mar. 2015.
[9] A. Souyah and K. M. Faraoun, "An image encryption scheme combining chaos-memory cellular automata and weighted histogram," *Nonlinear Dyn.*, vol. 86, no. 1, pp. 639–653, Oct. 2016.
[10] G. Zhou, D. Zhang, Y. Liu, Y. Yuan, and Q. Liu, "A novel image encryption algorithm based on chaos and Line map," *Neurocomputing*, vol. 169, pp. 150–157, Dec. 2015.
[11] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcation Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.
[12] R. Matthews, "On the derivation of a 'chaotic' encryption algorithm," *Cryptologia*, vol. 13, no. 1, pp. 29–42, 1989.

[13] B. Furht and D. Kirovski, Eds., *Multimedia Security Handbook*. Boca Raton, FL, USA: CRC Press, 2004.
[14] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998.
[15] J. Lang, "Image encryption based on the reality-preserving multiple-parameter fractional Fourier transform and chaos permutation," *Opt. Laser. Eng.*, vol. 50, no. 7, pp. 929–937, Jul. 2012.
[16] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals*, vol. 21, pp. 749–761, Jul. 2004.
[17] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Inf. Sci.*, vol. 297, pp. 80–94, Mar. 2015.
[18] A. Kanso and M. Ghebleh, "A novel image encryption algorithm based on a 3D chaotic map," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, no. 7, pp. 2943–2959, 2012.
[19] Y. Zhang, D. Xiao, Y. Shu, and J. Li, "A novel image encryption scheme based on a linear hyperbolic chaotic system of partial differential equations," *Signal Process., Image Commun.*, vol. 28, no. 3, pp. 292–300, 2013.
[20] S. Lian, J. Sun, and Z. Wang, "A block cipher based on a suitable use of the chaotic standard map," *Chaos Solitons Fract.*, vol. 26, no. 1, pp. 117–129, Oct. 2005.
[21] O. Mirzaei, M. Yaghoobi, and H. Irani, "A new image encryption method: Parallel sub-image encryption with hyper chaos," *Nonlinear Dyn.*, vol. 67, no. 1, pp. 557–566, 2012.
[22] J.-X. Chen, Z.-L. Zhu, C. Fu, H. Yu, and L.-B. Zhang, "A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism," *Commun. Nonlinear Sci.*, vol. 20, no. 3, pp. 846–860, Mar. 2015.
[23] W. Zhang, H. Yu, and Z.-L. Zhu, "Color image encryption based on paired interpermuting planes," *Opt. Commun.*, vol. 338, pp. 199–208, Mar. 2015.
[24] J. S. Fouda, A. Eyebe, J. Y. Effa, S. L. Sabat, and M. Ali, "A fast chaotic block cipher for image encryption," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 3, pp. 578–588, 2014.
[25] X. Wang, L. Teng, and X. Qin, "A novel colour image encryption algorithm based on chaos," *Signal Process.*, vol. 92, no. 4, pp. 1101–1108, Apr. 2012.
[26] A. Belazi, A. A. A. El-Latif, and S. Belghith, "A novel image encryption scheme based on substitution-permutation network and chaos," *Signal Process.*, vol. 128, pp. 155–170, Nov. 2016.
[27] X. Wang, Q. Wang, and Y. Zhang, "A fast image algorithm based on rows and columns switch," *Nonlinear Dyn.*, vol. 79, no. 2, pp. 1141–1149, 2015.
[28] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, no. 11, pp. 172–182, 2014.
[29] Z.-L. Zhu, W. Zhang, K.-W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Inf. Sci.*, vol. 181, no. 6, pp. 1171–1186, 2011.
[30] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," *Opt. Lasers Eng.*, vol. 78, pp. 17–25, Mar. 2016.
[31] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.
[32] R. Ge, L. Zhang, T. Zhang, S. Li, G. Gui, and Y. Ma, "A modified pulse-coupled spiking neuron circuit with memory threshold and its application," *IEICE Electron. Express*, vol. 13, no. 8, 2016, Art. no. 20151121.
[33] C. Bandt and B. Pompe, "Permutation entropy: A natural complexity measure for time series," *Phys. Rev. Lett.*, vol. 88, no. 17, 2002, Art. no. 174102.
[34] E. B. Corrochano, *Handbook of Geometric Computing*. Berlin, Germany: Springer, 2005.

RONGJUN GE received the B.E. and M.E. degrees from the School of Information Science and Engineering, Lanzhou University, Lanzhou, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University, Nanjing, China. His current research interests include image encryption, chaos, medical image processing, and machine learning.



GUANYU YANG received the B.E. and M.E. degrees from the School of Biomedical Engineering, Southeast University, Nanjing, China, in 2002 and 2005, respectively, and the Ph.D. degree from the University of Rennes1, Rennes, France. He is currently an Associate Professor with the Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University. His current research interests include image analysis, pattern recognition, and computerized-aid diagnosis.

JIASONG WU received the B.E. degree in biomedical engineering from the University of South China, Hengyang, China, in 2005, and the joint Ph.D. degree from the Laboratory of Image Science and Technology (LIST), Southeast University, Nanjing, China, and the Laboratoire Traitement du signal et de l'Image, University of Rennes 1, Rennes, France, in 2012.

He is currently a Lecturer with the Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University. His research interests mainly include fast algorithms of digital signal processing and their applications. He received the Eiffel Doctorate Scholarship of Excellence from the French Ministry of Foreign Affairs, in 2009, and the Chinese Government Award for outstanding self-financed students abroad from the China Scholarship Council, in 2010.

YANG CHEN received the M.E. and Ph.D. degrees in biomedical engineering from First Military Medical University, Guangzhou, China, in 2004 and 2007, respectively. Since 2008, he has been a Professor with the Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University, Nanjing, China. His research interests include medical image reconstruction, image analysis, pattern recognition, and computerized-aid diagnosis.

GOUENOU COATRIEUX received the Ph.D. degree in signal processing and telecommunication from the University of Rennes1, Rennes, France, in collaboration with Ecole Nationale Supérieure des Télécommunications, Paris, France, in 2002. He is currently a Professor with the Information and Image Processing Department, Institut Mines-Télécom, Telecom Bretagne, Brest, France. His research is conducted in the LaTIM Laboratory, INSERM U1101, Brest. His research interests include data security, encryption, watermarking, secure processing of outsourced data, digital forensics in medical imaging, and electronic patient records.

LIMIN LUO received the Ph.D. degree from the University of Rennes, Rennes, France, in 1986. He is currently a Professor with the Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University, Nanjing, China. His current research interests include medical imaging, image analysis, computer-assisted systems for diagnosis and therapy in medicine, and computer vision.