Nonlinear Dyn (2016) 83:2067–2077
DOI 10.1007/s11071-015-2465-7

ORIGINAL PAPER

Chaotic image encryption algorithm using wave-line permutation and block diffusion

Guodong Ye · Haiqing Zhao · Huajin Chai

Received: 5 June 2015 / Accepted: 21 October 2015 / Published online: 29 October 2015

Abstract An efficient and secure image encryption algorithm is proposed in this manuscript using SHA-3 hash function together with double two-dimensional Arnold chaotic maps. Classical encryption architecture, i.e., permutation plus diffusion, is employed in our scheme. To avoid time consumption of sorting operation for pixel position index in permutation stage, a novel wave-line-based confusion is suggested with four random directions of shuffling. The keystream generated by Arnold map is used for vertical and horizontal circular confusions, respectively, in which the initial conditions are updated by the SHA-3 hash values of chaotic matrix and a new vector produced from the plain-image. As a result, the proposed scheme can resist the known-plaintext attack compared with some existing encryption methods. Furthermore, in diffusion stage, a blocking method is designed with the outputs of hash values in the former block permuted image which are used to update again the initial conditions for Arnold map. The current block will influence the next block during the iterations, of which can resist well the chosen-plaintext attack. Numerical results show that the proposed encryption algorithm can have higher security and faster implementation for digital image communication.

Keywords Image encryption · SHA-3 · Arnold map · Wave-line · Block

G. Ye (✉) · H. Chai
College of Science, Guangdong Ocean University, Zhanjiang 524088, Guangdong, China
e-mail: guodongye@hotmail.com

H. Zhao
School of Mathematics and Computation Science, Lingnan Normal University, Zhanjiang 524048, Guangdong, China

1 Introduction

Cryptographic approaches for digital images have being studied extensively in secure communications [1–4]. Due to the special properties of images such as bulk data capacity, high redundancy, large size, and strong correlation among adjacent pixels, traditional text encryption methods cannot be applied directly. To meet the challenge of digital image security problem in open network, chaos-based methods [5,6] have been proposed in recent years. The reasons may ascribe to the characteristics of the chaotic system, for example, sensitivity to system parameters, ergodicity, and unpredictability. These can be considered analogous to ideal cryptographic properties in cipher such as confusion, diffusion, and avalanche. As a result, chaotic encryption algorithm has become popular for image encryption.

In [7], Amin et al. suggested a new chaotic block cipher scheme for image encryption, of which encrypts bits in block instead of pixels in block. So, it can avoid the weakness upon differential attacks and slow speed performance. Simple one-dimensional chaotic maps such as sine map, logistic map, and tent map were

used in [8]. It presented a new parametric switching chaotic system-based image encryption and showed high security. Classical structure for image encryption is permutation plus diffusion. However, the authors in [9] proposed another encryption scheme, i.e., only two rounds of diffusion can also achieve high sensitivity, high complexity, and high security. An improved chaos-based image encryption algorithm was designed in [10]. Chebyshev maps were adopted to generate the permutation array, and then a XORed operation was implemented in diffusion stage. To satisfy the real-time communication, we like to take fast algorithm and avoid some operations with high computation complex (for example sorting operation [11]). Moreover, the authors in [12] proposed a new chaos-based hash function that run in parallel to improve hashing speed. However, some existing algorithms were found in low security. Özkaynaka et al. [13] pointed out that with some pairs of chosen plain-images and the corresponding cipher-images, the encryption algorithm in [14] can be broken successfully. Reference [15] revealed that the keystream remains unchanged when encrypting different images using [16]. Therefore, it is not secure enough when the chosen-plaintext attack is applied. Recently, many other cryptoanalysis methods and image encryption schemes were also proposed, for example, an attack on Fridrich's chaotic image encryption [17], a combined encryption with authentication scheme [18], and a wavelet- and time-domain-based encryption algorithm [19].

Based on the analysis above, we propose an efficient and secure chaotic image encryption algorithm, in which a wave-line permutation is adopted firstly in confusion stage. Here, the keystream is not only dependent on the keys but also on the SHA-3 hash values of the plain-image (seeing the sum vector v in next section). So, it can resist the known-plaintext attack. Furthermore, blocking method is introduced in diffusion stage by which the algorithm can decrease time consumption. Meanwhile, SHA-3 is used again to make an influence of tiny change in the plain-image on the cipher-image. As a result, the diffusion operation can resist the chosen-plaintext attack. The rest of this paper is organized as: Sect. 2 introduces the chaotic Arnold map and SHA-3 hash function. Then, the processes of wave-line-based permutation and block diffusion are described in Sect. 3 with detail encryption steps. Experimental results are given in Sect. 4 to display the simulations. Section 5 presents security analyses to explain the performance of the proposed encryption algorithm followed by conclusions in Sect. 6.

2 Chaotic Arnold map and SHA-3 function

Two-dimensional Arnold map [20] is defined in following Eq. (1) with two integer control parameters a and b.

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & 1+ab \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \bmod 1 \tag{1}$$

where x_i and y_i are dropped in the field of [0, 1), and mod denotes the modulus after division. We can calculate easily the largest Lyapunov characteristic exponent of (1), i.e., $\lambda = 1 + \frac{ab+\sqrt{a^2b^2+4ab}}{2}$. So, the map is continuously chaotic if we let control parameter a and b be positive numbers. Particularly, $a \geq 1$ and $b \geq 1$ can be set to get stronger sense of chaos [20]. Figure 1 shows the chaotic behavior of Arnold map (1) by x coordinate and y coordinate (here, we set $a = 1, b = 1$).

As we know that SHA-3 is the newest cryptographic hash function [21] compared with SHA-1 and SHA-2. There are two properties for SHA-3. The first one is for any length message as input, it can generate a fixed length hash values as output, for example, 224-bit, 256-bit, 384-bit, or 512-bit. The second one is the sensitivity. That is to say, any tiny change in the message input will lead to greatly different hash values as output. SHA-3 function is employed to generate a 256-bit length of hash values in this algorithm. However, before being applied to our encryption scheme, they are divided into 32 groups, each group has 8-bit binary numbers, and are transformed into 32 decimal numbers. For example, the hash values of Lena image in Fig. 2a are shown in Fig. 2b using SHA-3. More about the SHA-3 function (known as Keccak) can be studied from Keccak sponge function family in [21].

3 The proposed image encryption algorithm

3.1 Permutation based on wave-line and SHA-3

Wave transmission is a common physical phenomenon, it can transmit over the air starting from the signal base station. Usually, we study the regular wave such as sine wave in Fig. 3a. It has a fixed amplitude and a constant wavelength with a source point. However, irregular wave is also existed in our life and is interesting

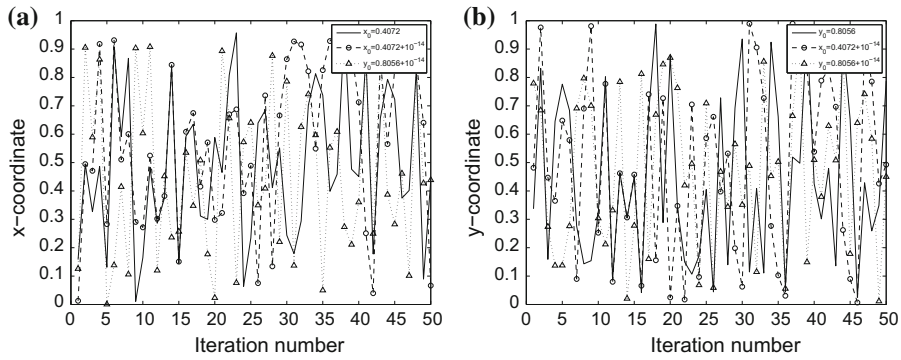

Fig. 1 Chaotic behavior of Arnold map: **a** x-coordinate, **b** y-coordinate

(b)

4	44	45	115	83	255	108	216
206	244	177	192	181	29	28	77
185	51	191	113	212	104	77	85
155	174	174	210	112	245	227	65

Fig. 2 Lena image: **a** plain-image, **b** 32 hash values

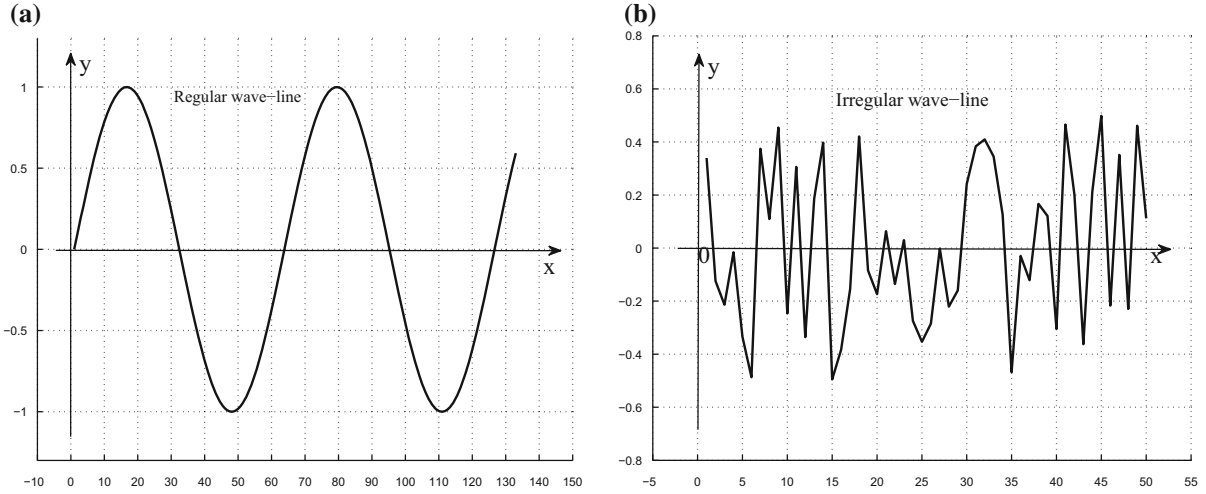

Fig. 3 The shape of wave-line: **a** regular, **b** irregular

seeing Fig. 3b. Here, we use two-dimensional Arnold map (Eq. 1) to simulate the irregular wave-line in this paper. The iterated values x_i and y_i are seen as two wave-lines in cartesian plane.

Assume that the plain-image P is in size of $M \times N$, two wave-lines are transmitted crossing the middle of image vertically and horizontally. We take the case of horizontal wave-line (Fig. 3b) as an instance for per-

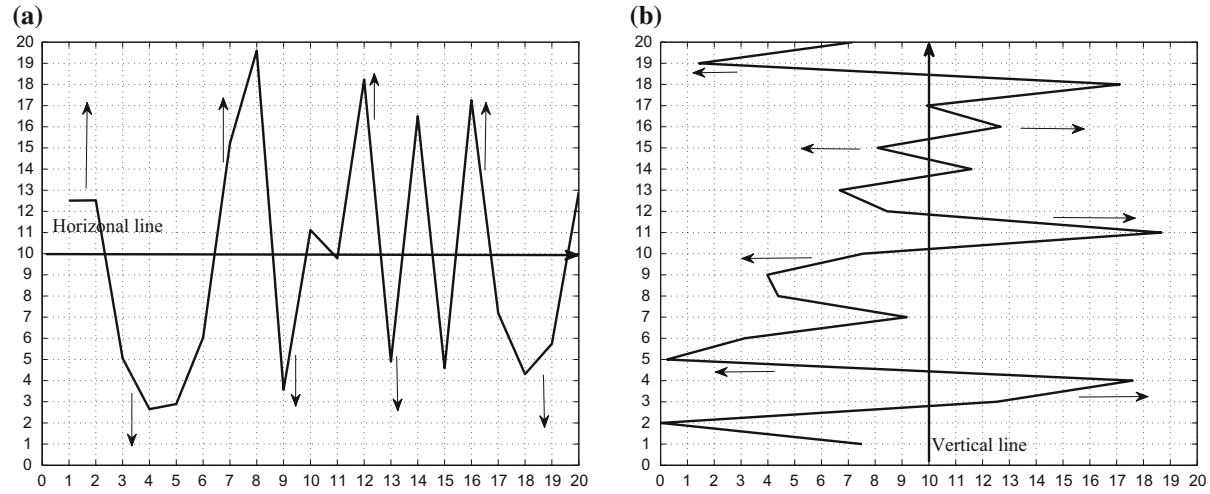

Fig. 4 Permutation in: **a** *horizonal line*, **b** *vertical line*

mutation operation. As we can see that some parts of the wave-line are above the x coordinate while some parts are below it. Of course, there are also some points left on the middle line. Activated by this phenomenon, we suggest a new pixel position permutation for the plain-image.

Permutation operation changes only the pixel positions but not the pixel values for the plain-image, i.e., just pixel positions confusion in the stage of permutation. In the first place, we compute the sum of the plain-image $s = \Sigma P_{M\times N}$. Then a vector v with size of $1 \times N$ is generated by Eq. (2). Using secret keys x_0 and y_0 set to Arnold map, we can get a chaotic matrix $A_{M\times N}$ with control parameters $a = (s \bmod M) + 1$, $b = (s \bmod (M + N)) + 1$. After transforming A into integer numbers of 0–255 by Eq. (3) and adding vector v to the last row of A, a new matrix B of size $(M + 1) \times N$ can be obtained. SHA-3 hash function is applied to this matrix B to produce hash values. Obviously, different images will have different hash values which are used to update the initial conditions for Arnold map. Take out the former 16 numbers as a group G_1 and the left 16 numbers as another group G_2, and suppose x_0 and y_0 be the initial keys for Arnold map, then we establish a mathematical model in Eq. (4) to update them. After iterating the Arnold map for some rounds we get sets $\{x_0, x_1, x_2, \ldots\}$ and $\{y_0, y_1, y_2, \ldots\}$. To achieve higher randomness, some former values are thrown away, then, two new sets, S_1 with N values and S_2 with M values, are obtained as wave-lines for vertical and horizontal permutation.

$$v_i = ((s \bmod 256) + i) \bmod 256,\ i = 1, 2, \ldots, N \tag{2}$$

$$A = \text{floor}(A \times 10^{14}) \bmod 256 \tag{3}$$

where floor(x) rounds the element of x to the nearest integer toward minus infinity.

$$\begin{cases} x_0 = x_0 + \text{sum}(G_1) \times 10^{-5} \bmod 1 \\ y_0 = y_0 + \text{sum}(G_2) \times 10^{-5} \bmod 1 \end{cases} \tag{4}$$

There are two directions in horizonal line, i.e., upwards and downwards as shown by arrow directions in Fig. 4a. Here, we consider $y = \text{floor}(M/2)$ as the middle line of the plain-image horizontally. The upwards circular permutation [22] is chosen if the elements of S_1 is bigger than floor($M/2$), i.e, vertical movement to the up. Otherwise, the downwards circular permutation is taken. However, all elements of the S_1 should be discretized as Eq. (5) before being used. Similarly, The same process can be applied to the vertical line direction seeing Fig. 4b using middle line $x = \text{floor}(N/2)$ with discretized Eq. (6). As a result, we can finish the permutation encryption if both directional confusions have been completed with all four directions of circular operation.

$$S_1 = \text{floor}(S_1 \times 10^{14}) \bmod M \tag{5}$$

$$S_2 = \text{floor}(S_2 \times 10^{14}) \bmod N \tag{6}$$

3.2 Diffusion based on blocking and SHA-3

Keystream generated dependent on both key and the plain-image can resist well the known-plaintext attack.

To avoid further the chosen-plaintext attack, the connection of any two pixels should be established [23]. Considering time consumption, blocking method is adopted in diffusion operation. Suppose that the permuted image is $\overline{P}$. Then, it is divided into two parts vertically $\overline{P}_1$ (the left block) and $\overline{P}_2$ (the right block), each one has size of $M \times N/2$. Arnold map (1) with new given initial keys $\overline{x}_0$ and $\overline{y}_0$ is employed again together with SHA-3 hash function. The second block $\overline{P}_2$ is firstly selected out, then, we calculate the hash values of it by SHA-3 algorithm and get 32 integers. Let fv and lv be the first and last values of them respectively. Then, control parameters are set to be $a = fv + 1$ and $b = lv + 1$ which are dependent on the permuted image.

Similarly to the permutation stage, two groups G_3 and G_4 are formed by dividing these 32 hash values averagely, each one has 16 numbers. Then, the keys $\overline{x}_0$ and $\overline{y}_0$ for Arnold map are updated again using following Eq. (7). As a result, $MN/2$ chaotic values can be got with updated keys iterated into Arnold map. Then, we reshape them into a matrix $K_{M\times N/2}$. However, before being applied to the diffusion operation, we should do a preprocess as following Eq. (8) for K to let all elements fall into 0–255. After that, modular diffusion is employed to the sub-blocks $\overline{P}_1$ and $\overline{P}_2$ by Eq. (9). To make any one-bit change in the plain-image lead to a significantly different cipher-image $C = [C_1, C_2]$, at least two rounds of iteration using Eq. (9) should be considered.

$$\begin{cases} \overline{x}_0 = \overline{x}_0 + \text{sum}(G_3) \times 10^{-5} \bmod 1 \\ \overline{y}_0 = \overline{y}_0 + \text{sum}(G_4) \times 10^{-5} \bmod 1 \end{cases} \tag{7}$$

$$K = \text{floor}(K \times 10^{14}) \bmod 256 \tag{8}$$

$$\begin{cases} C_1 = \overline{P}_1 + K \bmod 256 \\ C_2 = \overline{P}_2 + C_1 \bmod 256 \end{cases} \tag{9}$$

3.3 Encryption steps

Detail steps of the proposed image encryption algorithm are listed as follows. Here, at least two rounds of iteration for diffusion operation should be taken to enhance the security [24]. Moreover, the flowcharts of encryption and decryption processes can also be seen in Figs. 5 and 6, respectively.

Step 1 Suppose the plain-image is $P_{M\times N}$ with secret keys x_0, y_0, $\overline{x}_0$, and $\overline{y}_0$ for Arnold map. Generate a chaotic matrix A with the same size to P.

Step 2 Calculate the sum s of P and the vector v by Eq. (2). Iterate the Arnold map using control parameters a and b produced from s, and obtain matrices A and B. Then compute the hash values for B using SHA-3 algorithm, and divide them into two groups G_1 and G_2.

Step 3 Update the initial keys for x_0 and y_0 using Eq. (4), and iterate Arnold map again to get S_1 and S_2.

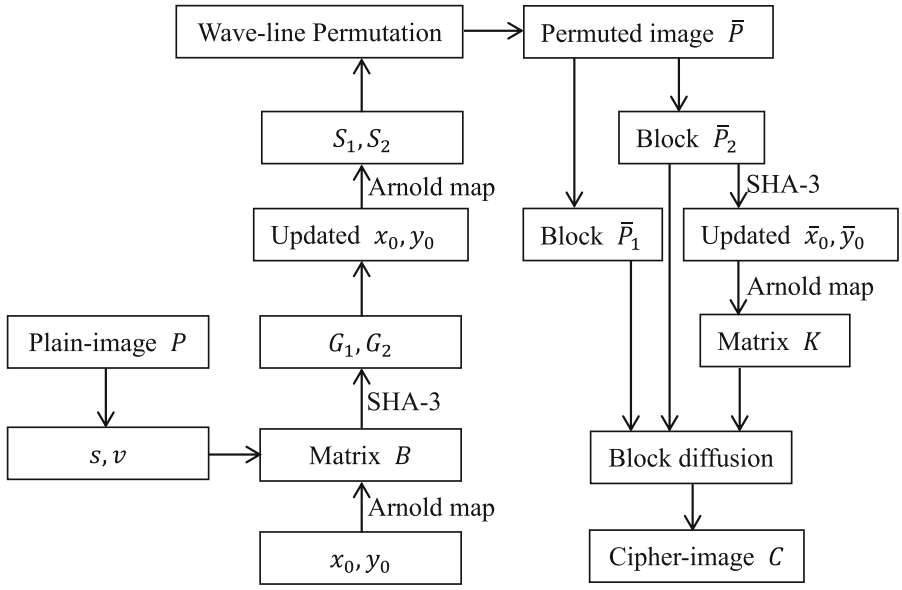

Fig. 5 The flowchart of encryption process

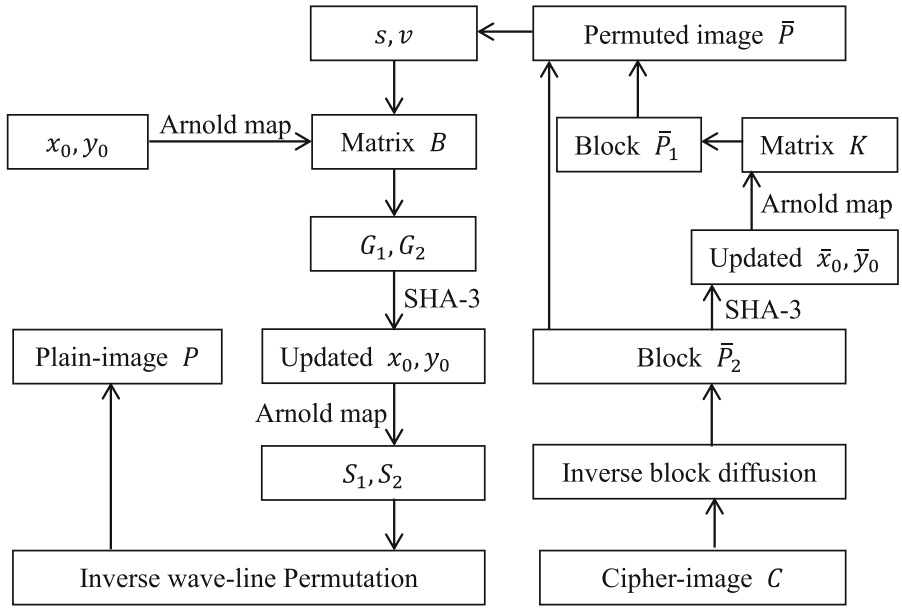

Fig. 6 The flowchart of decryption process

Step 4 Do the circular permutation for P by wave-line based method. Then, a permuted image $\overline{P}$ is formed.

Step 5 Divide $\overline{P}$ into two blocks $\overline{P}_1$ and $\overline{P}_2$ vertically. Compute the hash values of $\overline{P}_2$ by SHA-3 algorithm. Set control parameters a and b be the first value and last value of the hash values.

Step 6 Update keys $\overline{x}_0$, $\overline{y}_0$ by Eq. (7), and get chaotic matrix K after iterating Arnold map.

Step 7 Do the diffusion operation for $\overline{P}_1$ and $\overline{P}_2$ with several rounds, cipher-image C is obtained finally.

4 Numerical experiments

In this section, we do the test for the proposed algorithm. Initial conditions for Arnold map are set randomly to be $x_0 = 0.4114$, $y_0 = 0.5872$, $\overline{x}_0 = 0.3315$, $\overline{y}_0 = 0.6009$. Control parameters are self-adaptively produced according to the plain-image and the permuted image. All experimental tests are implemented by MATLAB R2011b on a platform of Windows 7 with an Intel(R) Core(TM) i7-3770, 3.40 GHz CPU. Lena image of size 256×256 as the plain-image is shown in Fig. 7a. With one round of permutation and three rounds of diffusion, Fig. 7c shows the cipher-image after using our encryption method. Permuted image in Fig. 7b can also be got if permutation operation is taken only.

5 Security analyses

5.1 Key space analysis

The key space consists of four initial conditions for double Arnold maps, i.e., x_0, y_0, $\overline{x}_0$, $\overline{y}_0$. If we set the floating precise to be 10^{-14}, then, the probabilities of key combination can reach as large as 10^{56} (secure requirement is at least 10^{30} [25]). Thus, It is big enough for our image encryption algorithm to resist the brute-force attack. Of course, in order to enlarge the key space, other higher dimensional chaotic systems instead of Arnold map can be employed, for example, hyperchaotic system [14].

5.2 Sensitivity analysis for keys and the plain-image

An ideal encryption scheme should be very sensitive to every secret key and every pixel in the plain-image [11]. By using our method, Fig. 7d, e show us that a wrong decrypted image will be raised if there is any small change in keys x_0 and $\overline{x}_0$, respectively. The same is to the keys y_0 and $\overline{y}_0$. Of course, a correct decrypted image can be recovered if all keys used are correct seeing Fig. 7f.

To test the influence of a tiny change in one pixel even just one-bit in the plain-image on the whole cipher-image, NPCR (number of pixel change rate) and

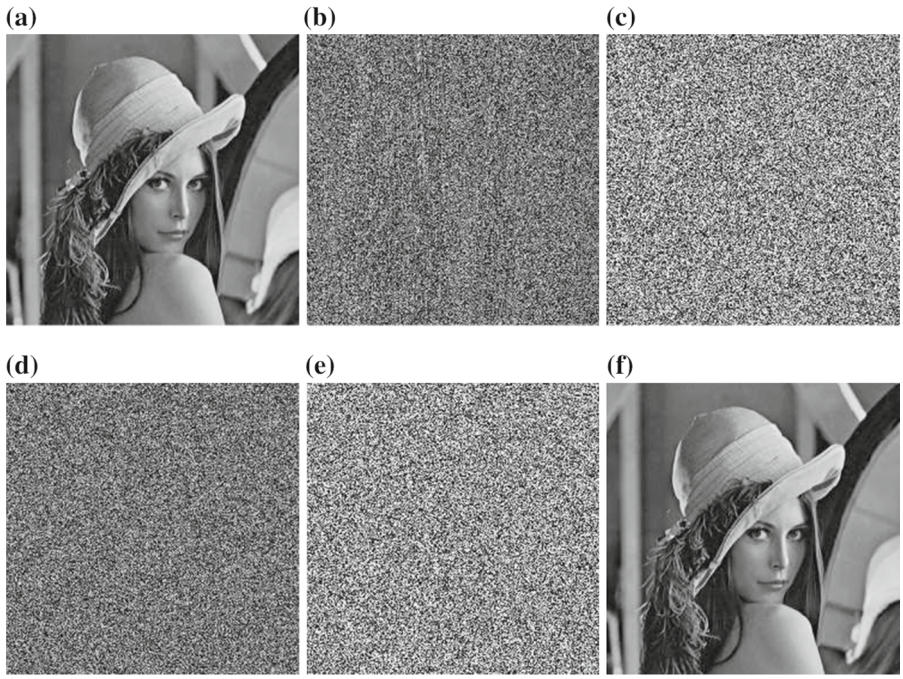

Fig. 7 Tests: **a** plain-image, **b** permuted image, **c** cipher-image, **d** decryption with 10^{-14} changed in x_0, **e** decryption with 10^{-14} changed in $\overline{x}_0$, **f** correct decryption

Table 1 UACI and NPCR for different images

Images	Boat	Lena	Cameraman	Barb	Baboon
Size	512 × 512	256 × 256	256 × 256	512 × 512	512 × 512
UACI	33.52958	33.39281	33.53431	33.48699	33.40222
NPCR	99.62959	99.55444	99.59870	99.62349	99.61586

UACI (unified averaged changing intensity), defined in following Eqs. (10) and (11), are commonly used to evaluate it. The test results are listed in Table 1 for different images by the proposed algorithm. From these results, we see that the encryption scheme is very sensitive with respect to any small change (i.e., one-bit) in pixel of the plain-image. All values of UACI are bigger than 33.3 % while all NPCR values surpass 99.5 %.

$$\text{NPCR} = \frac{\sum_{ij} D(i,j)}{M \times N} \times 100\,\% \tag{10}$$

$$\text{UACI} = \frac{1}{M \times N}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\,\% \tag{11}$$

where C_1 and C_2 are two cipher-images whose corresponding plain-images have just only one-pixel difference. $D(i,j) = 0$ if $C_1(i,j) = C_2(i,j)$; otherwise, $D(i,j) = 1$.

5.3 Statistical analysis

For a natural meaningful image, there exists strong correlation among two adjacent pixels [26]. We can compute the correlation coefficient for the plain-image and the cipher-image by following formula (12). By choosing some pairs of adjacent pixels from the plain-image and its corresponding cipher-image in vertical line randomly (Here, Cameraman image is used for test). Figure 8 shows the correlation distributions. So, we can reduce the high correlation coefficients by the proposed encryption algorithm.

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)D(y)}} \tag{12}$$

where $cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$, $D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$, $E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$. Here, x_i and y_i represent the gray values of two adjacent pixels in the image.

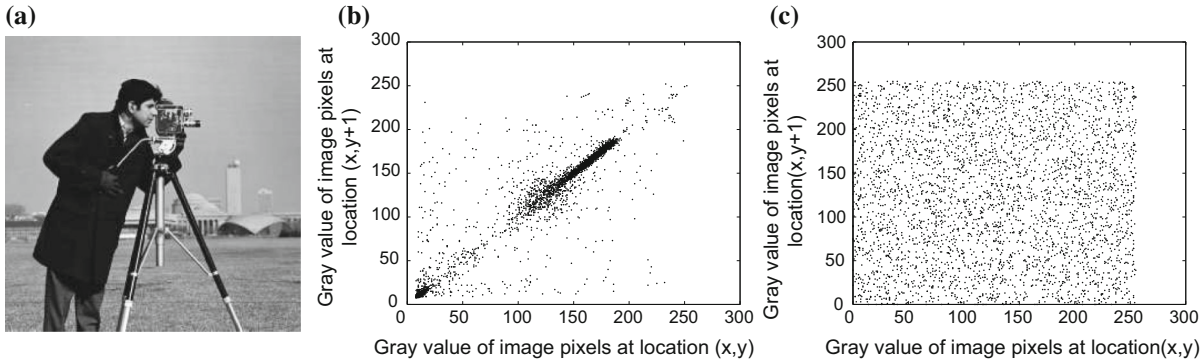

Fig. 8 Test of correlation: **a** Cameraman image, **b** correlation distribution in the plain-image, **c** correlation distribution in the cipher-image

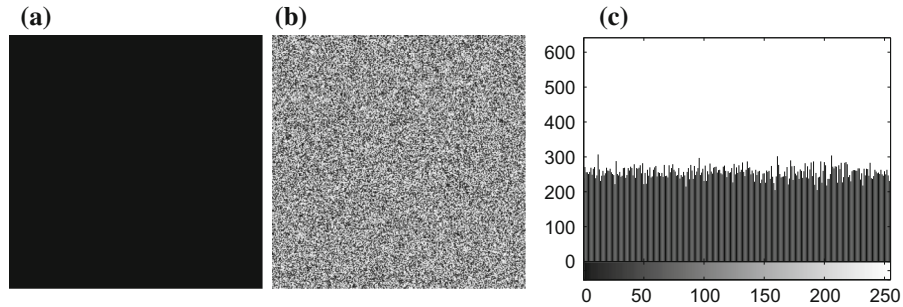

Fig. 9 Test for black image **a** plain-image, **b** cipher-image, **c** histogram of the cipher-image

Pixel values distribution of an image can be displayed in histogram plane. To resist the statistical attack, an uniform distribution of gray values in the cipher-image should be achieved even for encrypting black image. Figure 9 shows the test of black image in size of 256×256. Therefore, the proposed method can resist efficiently all kinds of statistical attacks.

5.4 Randomness analysis using NIST

NIST 800-22 test can evaluate the true-random and pseudorandom numbers for cryptographic application [28]. For a ideal encryption algorithm, the outputs of P value for the cipher-image should fall into the interval [0, 1]. In this section, we randomly select two images, i.e., fruits and seaside as shown in Fig. 10a, b

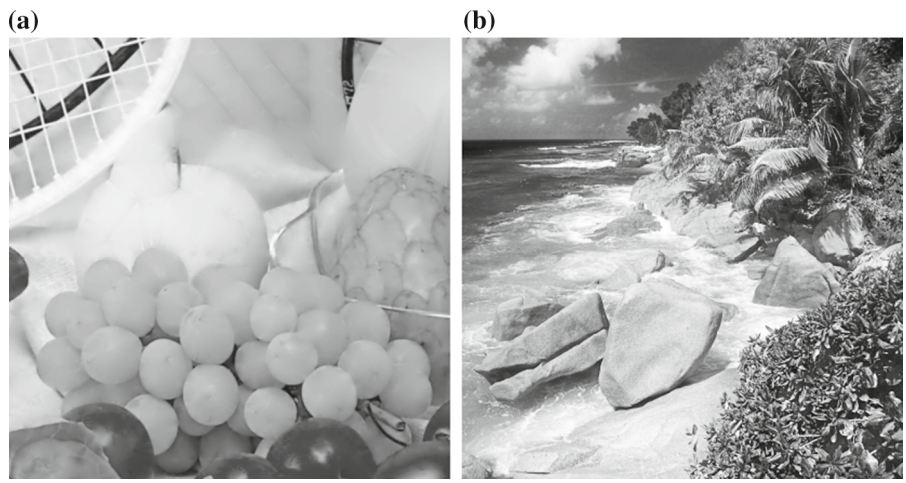

Fig. 10 Images for randomness test: **a** Fruits, **b** Seaside

Table 2 Randomness test for the cipher-image of fruits image

Test items	P value	Results
Frequency	0.657933	Pass
Block frequency	0.816537	Pass
Cumulative sums	0.289667	Pass
Runs	0.181557	Pass
Longest run	0.851383	Pass
Rank	0.816537	Pass
FFT	0.798139	Pass
Non-overlapping template	0.419021	Pass
Overlapping template	0.834308	Pass
Universal	0.924076	Pass
Approximate entropy	0.759756	Pass
Random excursions	0.392456	Pass
Random excursions variant	0.585209	Pass
Serial	0.108791	Pass
Linear complexity	0.678686	Pass

Table 3 Randomness test for the cipher-image of seaside image

Test items	P value	Results
Frequency	0.955835	Pass
Block frequency	0.366918	Pass
Cumulative sums	0.834308	Pass
Runs	0.816537	Pass
Longest run	0.071177	Pass
Rank	0.739918	Pass
FFT	0.964295	Pass
Non-overlapping template	0.924076	Pass
Overlapping template	0.350485	Pass
Universal	0.759756	Pass
Approximate entropy	0.897763	Pass
Random excursions	0.437274	Pass
Random excursions variant	0.534146	Pass
Serial	0.080519	Pass
Linear complexity	0.816537	Pass

respectively, from Google image database. After applying our method, the results for P value are given in Tables 2 and 3 accordingly. From these results, we can conclude that the proposed algorithm can pass the randomness test.

5.5 Information entropy analysis

To measure the expected value of a message and the unpredictability of information content, information entropy (IE) is usually taken to test the strength of a designed encryption algorithm, which is defined in Eq. (13) for a received message m.

$$\mathrm{IE}(m) = \sum_{j=0}^{2^L-1} p(m_j) log_2 \frac{1}{p(m_j)} \tag{13}$$

where L is the length of pixel value in binary number (for a gray image, $L = 8$), $p(m_j)$ represents the probability of the occurrence of symbol m_j, and log_2 denotes the base 2 algorithm. The results for different images are given in Table 4 after using our method. All the values are close to theoretical value 8. Therefore, the proposed image encryption algorithm is secure against information entropy attack.

Table 4 Information entropy for different cipher-images

Cipher-images	Lena	Boat	Baboon	Fruits	Seaside
IE	7.990	7.991	7.992	7.992	7.992

Table 5 The speed performance

Images	Ref. [9]	Ref. [20]	Ref. [22]	Ref. [27]	Ours
256×256	0.1170 s	0.4680 s	0.3198 s	0.1092 s	0.0156 s
512×512	0.4290 s	1.8876 s	1.2324 s	0.3666 s	0.0624 s

5.6 Comparisons

The speed performance for the whole encryption process (time cost for the decryption process is the same due to the symmetric structure) is one of the rules to evaluate the efficiency of any designed algorithm. Table 5 lists some comparisons of different encryption methods when encrypting different image sizes, of which shows the faster speed (unit: second) of our scheme. Additionally, Table 6 gives us another comparison of iteration rounds for different algorithms used to reach certain security level. So, the proposed scheme is fast and efficient for real-time image communications. In additional, we also do some comparisons with reference [11] which uses the sorting operation. Table 7 shows the results by testing different images under a new computation environment (Matlab R2011b on a platform of Windows 7 with an Intel(R) Core(TM) i3-2350, 3.40 GHz CPU). Here, for an image in size of 512×512, 0.2059 second is spent compared with

Table 6 The rounds of iteration

Methods	Permutation round	Diffusion round
Ref. [2]	4	4
Ref. [6]	2	2
Ref. [20]	2	2
Ref. [22]	1	2
Ref. [24]	2	2
Ours	1	3

Table 7 Comparisions with Ref. [11]

Methods	Ref. [11]	Ours
Time cost for encrypting an image of size 256×256	0.0671 s	0.0421 s
Time cost for encrypting an image of size 512×512	0.2059 s	0.1529 s
Time cost for encrypting an image of size 1024×1024	0.8377 s	0.5678 s
Key space	10^{57}	10^{56}
Dependence on plain-image in permutation stage	No	Yes
Dependence on plain-image in masking stage	No	Yes
Dependence on plain-image in final permutation stage	Yes	–
The Use of sorting function	Yes	No

0.210 second cost in [11] due to a better computational platform.

6 Conclusions

A novel image encryption algorithm has been proposed in this paper, in which a new wave-line-based permutation is designed with SHA-3 function. To avoid the known-plaintext attack, the keystream is generated dependent on the plain-image. In the stage of diffusion, we adopt the blocking method to save time consumption. Due to the updating system for the initial conditions in Arnold map (SHA-3 function is used again), any one-bit change in pixel of the plain-image will influence significantly the whole cipher-image. Because the algorithm enhances the relationship of any two pixels in the permuted image, the presented method can resist efficiently the chosen-plaintext attack. Various types of common security analysis have showed the outstanding performance of the proposed image encryption algorithm.

Acknowledgments The authors would like to thank the three anonymous reviewers for valuable comments which are very useful in improving the quality of this manuscript. The work described in this paper was fully supported by the National Natural Science Foundation of China (No. 11301091), the Natural Science Foundation of Guangdong Province of China (No. 2015A030313614), the Project of Enhancing School With Innovation of Guangdong Ocean University of China (No. Q14217), and the Science & Technology Planning Project of Zhanjiang City of China (Nos. 2015B01051, 2015B01098).

References

1. Matthews, R.: On the derivation of a chaotic encryption algorithm. Cryptologia **13**, 29–42 (1989)
2. Wong, K.W., Kwok, B.S.H., Law, W.S.: A fast image encryption scheme based on chaotic standard map. Phys. Lett. A **372**, 2645–2652 (2008)
3. Cheng, H., Li, X.B.: Partial encryption of compressed image and videos. IEEE Trans. Signal Process. **48**, 2439–2451 (2000)
4. Hua, Z.Y., Zhou, Y.C., Pun, C.M., Chen, C.L.P.: 2D Sine Logistic modulation map for image encryption. Inf. Sci. **297**, 80–94 (2015)
5. Hammami, S.: State feedback-based secure image cryptosystem using hyperchaotic synchronization. ISA Trans. **54**, 52–59 (2015)
6. Wang, Y., Wong, K.W., Liao, X.F., Chen, G.R.: A new chaos-based fast image encryption algorithm. Appl. Soft Comput. **11**, 514–522 (2011)
7. Amin, M., Faragallah, O.S., El-Latif, A.A.A.: A chaotic block cipher algorithm for image cryptosystems. Commun. Nonlinear Sci. Numer. Simul. **15**, 3484–3497 (2010)
8. Zhou, Y.C., Bao, L., Chen, C.L.P.: Image encryption using a new parametric switching chaotic system. Signal Process. **93**, 3039–3052 (2013)
9. Norouzi, B., Seyedzadeh, S.M., Mirzakuchaki, S., Mosavi, M.R.: A novel image encryption based on hash function with only two-round diffusion process. Multimed. Syst. **20**, 45–64 (2014)
10. Liu, G.Y., Li, J., Liu, H.J.: Chaos-based color pathological image encryption scheme using one-time keys. Comput. Biol. Med. **45**, 111–117 (2014)
11. Fouda, J.S.A.E., Effa, J.Y., Sabat, S.L., Ali, M.: A fast chaotic block cipher for image encryption. Commun. Nonlinear Sci. Numer. Simul. **9**, 578–588 (2014)
12. Sen Teh, J., Samsudin, A., Akhavan, A.: Parallel chaotic hash function based on the shuffle-exchange network. Nonlinear Dyn. **81**, 1067–1079 (2015)
13. Özkaynak, F., Özer, A.B., Yavuz, S.: Cryptanalysis of a novel image encryption scheme based on improved hyperchaotic sequences. Opt. Commun. **285**, 4946–4948 (2012)
14. Zhu, C.X.: A novel image encryption scheme based on improved hyperchaotic sequences. Opt. Commun. **285**, 29–37 (2012)

15. Wang, X.Y., Liu, L.T.: Cryptanalysis of a parallel sub-image encryption method with high-dimensional chaos. Nonlinear Dyn. **73**, 795–800 (2013)
16. Mirzaei, O., Yaghoobi, M., Irani, H.: A new image encryption method: parallel sub-image encryption with hyper chaos. Nonlinear Dyn. **67**, 557–566 (2012)
17. Solak, E., Çokal, C., Yildiz, O.T., Biyikiğlu, T.: Cryptoanalysis of Fridrich's chaotic image encryption. Int. J. Bifurcat. Chaos. **20**, 1405–1413 (2010)
18. Farajallah, M., Fawaz, Z., El Assad, S., Déforges, O.: Efficient image encryption and authentication scheme based on chaotic sequences. In: The 7th International Conference on Emerging Security Information, Systems and Technologies, pp.150–155 (2013)
19. Luo, Y.L., Du, M.H., Liu, J.X.: A symmetrical image encryption scheme in wavelet and time domain. Commun. Nonlinear Sci. Numer. Simul. **20**, 447–460 (2015)
20. Ye, R.S.: A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism. Opt. Commun. **284**, 5290–5298 (2011)
21. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak sponge function family. http://keccak.noekeon.org
22. Ye, G.D., Wong, K.W.: An efficient chaotic image encryption algorithm based on a generalized Arnold map. Nonlinear Dyn. **69**, 2079– 2087 (2012)
23. Deng, S.J., Zhan, Y.P., Xiao, D., Li, Y.T.: Analysis and improvement of a hash-based image encryption algorithm. Commun. Nonlinear Sci. Numer. Simul. **16**, 3269–3278 (2011)
24. Wong, K.W., Kwok, B.S.H., Yuen, C.H.: An efficient diffusion approach for chaos-based image encryption. Chaos Solitons Fractals **41**, 2652–2663 (2009)
25. Alvarez, G., Li, S.J.: Some basic cryptographic requirements for chaos based cryptosystems. Int. J. Bifurcat. Chaos. **16**, 2129–2151 (2006)
26. Murillo-Escobar, M.A., Cruz-Hernández, C., Abundiz-Pérez, F., López-Gutiérrez, R.M., Acosta Del Campo, O.R.: A RGB image encryption algorithm based on total plain image characteristics and chaos. Signal Process. **109**, 119–131 (2015)
27. Eslami, Z., Bakhshandeh, A.: An improvement over an image encryption method based on total shuffling. Opt. Commun. **286**, 51–55 (2013)
28. Pareschi, F., Rovatti, R., Setti, G.: On statistical tests for randomness included in the NIST SP800-22 test suite and based on the binomial distribution. IEEE Trans. Inf. Forensics Secur. **7**, 491–505 (2012)