Multimed Tools Appl (2017) 76:607–629
DOI 10.1007/s11042-015-3064-9

A novel encryption scheme for colored image based on high level chaotic maps

Majid Mollaeefar[1] · **Amir Sharif**[1] · **Mahboubeh Nazari**[2]

Received: 7 March 2015 / Revised: 30 September 2015 / Accepted: 8 November 2015 /
Published online: 17 November 2015

Abstract In the following paper, the novel method for color image encryption has proposed based on high level chaotic maps . We introduced two novel chaotic maps "Cosinus-Arcsinus (*CA*)" and "Sinus-Power Logistic (*SPL*)", which have better chaotic behaviour against other available chaotic maps. Our scheme like other image encryption schemes has two main phases, which are pixel shuffling and pixel diffusion. We made an efficient chaotic permutation method, which is extremely dependent on plain image. The proposed method compared with other availabale permutation methods in pixel shuffling stage has a better performance with a lower computational overhead. Another advantage of our permutation method is less correlation between adjacent pixels in the permuted image, which causes high confusion levels with lower iterations. In pixel diffusion phase, we introduced coupled map based on *SPL* map to change each color component distinctly. Moreover, we used *CA* map to generate three chaotic sequences for deriving initial seeds of coupled map in a random manner. Followed by three chaotic matrix will be created to change pixels color component values. The proposed diffusion phase implies desirable uniformly distributation in color histogram of encrypted image and makes the scheme robust against statistical attacks. In addition, creating secret keys in a large size and high sensitivity to the original pictures leads acceptable results in the average of *NPCR* (99.67), *UACI* (33.45), and resistance against brute-force attack. The experimental results reveal that the new image encryption scheme has the charectristics of 'secure encryption algorithm' such as, large key space, high security and high sensitivity.

Keywords Chaos · Image cryptography · Lyapunov exponent · Security

✉ Amir Sharif
amir-sharif@hotmail.com

Majid Mollaeefar
m.mollaeefar@imamreza.ac.ir

[1] Department of Computer and Information Technology, Imam Reza International University, Mashhad, Iran

[2] Department of Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran

1 Introduction

Indeed, image encrypthion is a method for making illegible information, which transfer through a network like the authorisezd person for the special information. In addition, they can get the basic information (encrypted key). Designs of image encryption methods caused many researchers attend to introducing a safe method for transferring images straightway through internet or Wi-Fi networks. Because of basic difference between images and texts in volume and redundant bits, traditional cryptography methods are not efficient [3, 4]. In the 1989, for the first time the effective method of chaotic had introduced based on the cryptograpgy algoritms. As has been noted, many researchers have done on chaos-based image cryptography [7, 11]. Sensitivity to primary conditions, non-periodic, non-convergence, and controlling parameters are unique properties of chaotic cryptography compared to traditional ones. Other advantages of this type of cryptography are their high speed and having no computational overhead [13]. Real difference and several techniques have been introduced for image cryptography that are stated briefly as followings. (Space domain) and (frequency domain) are two major methods used in most papers. In space domain all pixels forming the image are considered and various methods are implemented directly on these pixels. In the second method, frequency domain is a space such that every proportion of image in image position F, shows the intense amount in image I, vary over a specific distance dependent on F. In fact, in this method we turn to the image as it is. Huang and Nein [9] presented a method for pixel shuffling along with multi-chaos system for image encryption. In 2010, Solac et al. [20] by analyzing a method which introduced earlier at this year find its shortcomings and prove that the scheme was vulnerable against two kind of attacks, then they present a solution for encountering it. During that year, Zhang et al. [26] introduced a method base on Arnold Cat map and coupled chaos for image encryption. Rejinder et al.[12] proposed a method for image cryptography based on improving *DES* and chaos. Zhu et al. [29] proposed a symmetric chaos based image encryption scheme by using of bit-level permutation. They use a simple chaotic map, which named Arnold cat map for bit-level permutation and Logistic map used in diffusion stage. Changing the position and values of pixels are the advantageous of bit-level permutation. Indeed, in bit-level permutation Arnold Cat map and Logistic map has used for determining parameters of the Arnold cat map. Congxu Zhu in [28] proposed a novel image encryption scheme based on hyperchaotic sequences. The hyperchaotic sequences are changed in a way to produce chaotic key streams, which is more suitable for image encryption. The author, in order to improve key sensitivity and plain text sensitivity made final encryption key stream dependent on both of the chaotic key stream and plaintext and this lead to high key sensitivity and plain text sensitivity, which are two important metrics in proving security of the scheme [1, 2]. Wang et al. [21] proposed a fast image encryption method with the help of chaotic maps. In fact, they merged diffusion and permutation phase, which result to acceleration of image encryption scheme. First, image portioned into blocks of pixel, then with help of spatiotemporal chaos, pixel value shuffling and diffusion can run simultaneously. Zhang and Cao in [24] presented an image encryption scheme by using a created chaotic map whose maximal Lyapunov exponent was reached beyond 1 and it was equal to 1.0742. In addition, they used Arnold cat map for permutation phase. Their method has two shortcomings. First, there is no relation between the secret keys and the original image. Indeed, the secret key is not sensitive to the original image, which causes two side effects. First, the secret key may not be changed for various images and it directs a hostile to get a set of basic information to decrypt the encrypted image. Second, using the same sequence for changing all of three-color

components of an image decrease the security. Because of any reason, hostile access to the initial value, at the same time he will be able to recover all the color component values. In this paper, we improve presented scheme in [24] and resolve those shortcomings. As it's obvious, Lyapunov exponent is the main characteristic of chaotic maps. Therefore, the increasing of the Lyapunov exponent, causes to increase the dispersion of output sequence values and better chaotic behavior. Hence, we have made one-dimensional chaotic maps with the Lyapunov exponents 1.38 and 1.518, respectively. These novel chaotic maps in comparison with conventional maps and the ones introduced by Zhang and Cao in [24], have stronger chaotic behaviour. We utilized effective coupled chaotic map to create three Chaotic Matrixes for changing the pixel amounts in each color component, distinctly. In order to improve security of the proposed scheme and key sensitivity, initial state of each color component should be dependant to each other and plain image. By utilizing this procedure, even if an intruder guesses one of the initial state condition values, he wouldn't be able to rebuild part of secret plain image. Regardless of the dependency on the initial values, if hostail guesses one of the initial values corresponding to one of the color components, he/she can retrive the shadow of the plain image. We provide a test to show this concept, which shown in Fig. 8. We use Cosinus-Arcsinus (*CA*) map with these initial states to generate three chaotic sequences as inputs for the coupled map in diffusion phase. Moreover, a new permutation scheme, chaotic-diagonal permutation, proposed the intense dependency on the original image. Indeed, after applying chaotic interception steps we will have new plain. By arranging diagonal layers next to each other, we obtain a vector of pixel values with length $M \times N$ to fill a matrix in rows. This process will be iterated P times with differing interception indices that obtained from *SPL* map to make final permuted matrix (Fig. 9). Simulations and performance evaluations show that this permutation algorithm is more efficient and need less iteration compared with other available permutation methods (Fig. 10). The execution time comparison between our proposed method and other available methods has shown in Table 4, which proved our proposed permutation algorithm speed is satisfying. The output of permutation stage, bitxored with Chaotic Matrixes which are result of coupled chaotic map and make encrypted image. Furthermore, two kinds of keys, cutting key ($CK_{R,\ G,\ B}$) and averaging key ($AK_{R,\ G,\ B}$) for each color components used to create dependencies between the encryption algorithm and the original image. Therefore, the proposed method will be resistant to the chosen-plain text and known-plain text attacks. The total time of encryption algorithm is almost 0.11 second. Briefly, beside presenting two high level chaotic map novels (*CA*, *SPL*), an encryption algorithm which has novelty in both diffusion and permutation phases is proposed. Indeed, in diffusion phase among the presentation of a chaos-based random selection technique, a large key space and high key sensitivity is provided. The overmentioned features cause that the proposed algorithm resistants against brute-force and differential attacks. Moreover, lower computational over head, intenses confusion with lower iterations and high dependency on plain image in chaotic diagonal permutation; and it can also gurantee the performance and security of proposed method. Exprimental results implies that our method has superiority in key space size and convincied results in another part of experimental results such as *NPCR* about 99.67 and *UACI* about 33.45.

The paper structure is as follows: in Section 2, necessary, mathematical basics will be presented. After that proposed method will be described in Section 3. Then we provided experimental results and security analysis in Section 4. Finally, in Section 5, we come into conclusions.

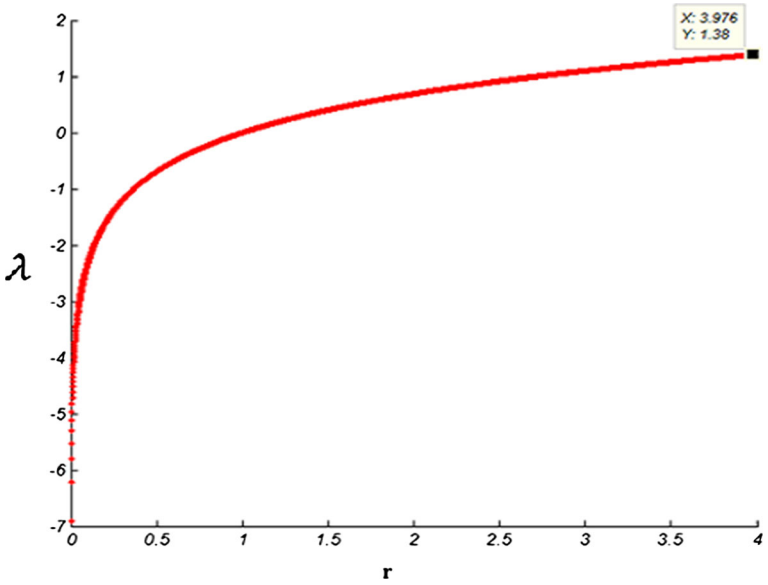


Fig. 1 The Lyapunov exponent of Cosinus-Arcsinus system

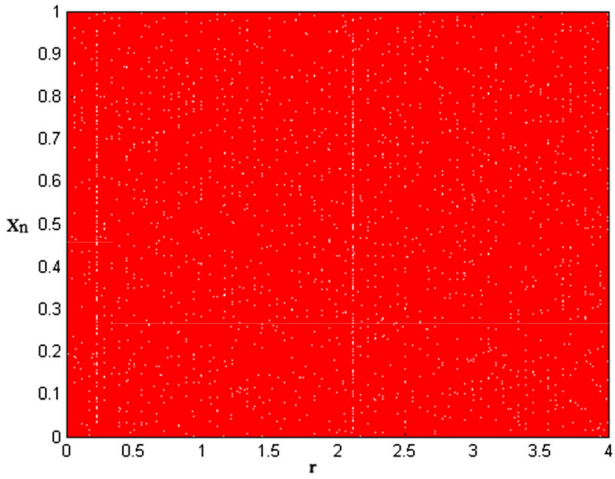
2 High level chaotic maps

2.1 The Cosinus-Arcsinus system

We introduce a new chaotic map on interval [0, 1], with high positive Lyapunov exponent by combination of Cosinus and Arc sinus (*CA*). The mathematical formula of this new chaotic map is as below:

$$X_{n+1} = CA(r, X_n) = \cos^2\left(r \arcsin\left(\sqrt{|X_n|}\right)\right) \tag{1}$$

Fig. 2 The bifurcation diagram of Cosinus-Arcsinus system



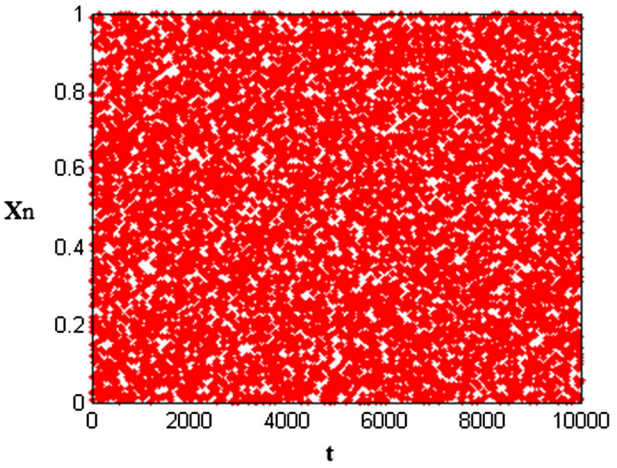


Fig. 3 The spatiotemporal diagram of Cosinus-Arcsinus system

Where r, is a control parameter in a range (0, 4) and X_n is initial state condition, which is in the interval [0, 1]. Moreover, in parameter value r=3.976 Lyapunov exponent has maximum value=λ=1.38. The Lyapunov exponent bifurcation and spatiotemporal diagram are shown in Figs. 1, 2 and 3, respectively.

2.2 The Sinus-power logistic system

48, 0.48]. This map is a combination of Sinus and PowerLogistic that first introduced in [24]. Based on the Lyapunov exponent diagram, this map has positive Lyapunov exponent for $r\epsilon$ (0,3.5). Moreover, in parameter value r=3.465 Lyapunov exponent has maximum value λ= 1.518. The Lyapunov exponent, bifurcation and spatiotemporal diagrams which represented chaotic behaviour of *SPL* map, shown in Figs. 4, 5 and 6, respectively. The mathematical formula of this new chaotic map is as below:

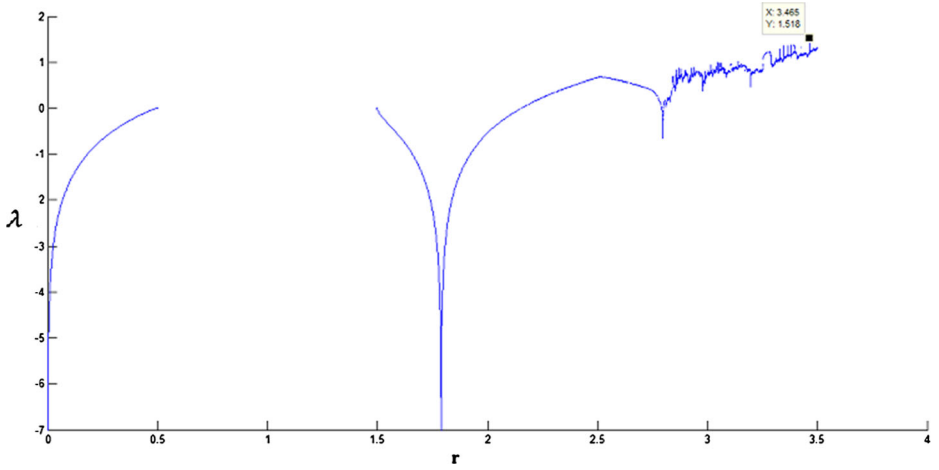


Fig. 4 The Lyapunov exponent of Sinus-Power Logistic System

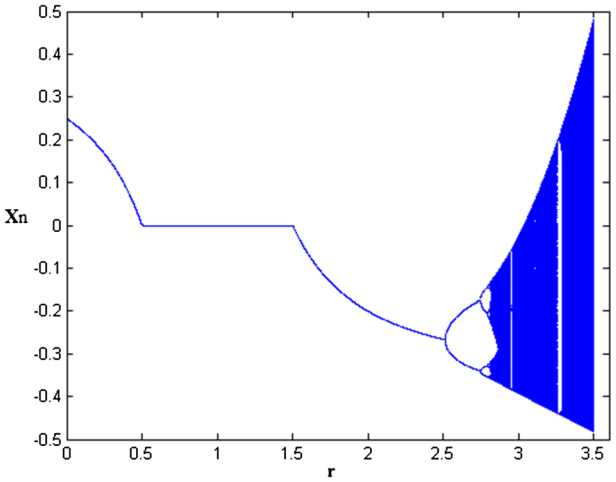


Fig. 5 Bifurcation diagram of Sinus-Power Logistic System

$$X_{n+1} = SPL(r, X_n) = \sin^2\left(r\arcsin\left(\sqrt{|X_n|}\right)\right) + (1-r)2(|X_n|)(1-2|X_n|) \qquad (2)$$

Where r is control parameter in the range (0, 3.5), and X_n is initial state condition with a range (−0.48, 0.48). We provide a comparison between the Lyapunov exponent of new proposed chaotic maps and some available maps in Table 1.

2.3 New coupled chaotic map

For having chaotic dynamics in a spatially extended system, we need to use coupled map. Indeed, a coupled map consists of an ensemble of elements of given ("map") that interact ("couple") with other elements from a suitably chosen set. The dynamic of each element is given by a map. As a consequence, the coupled map is a discrete time multi-dimensional dynamical systems. An important type of coupled map is the coupled map lattice (CML), in which each element is set on a lattice of a given dimension, resulting in a ("map"), discrete space ("lattice"), and continuous state. The mathematical formula of this map is as below:

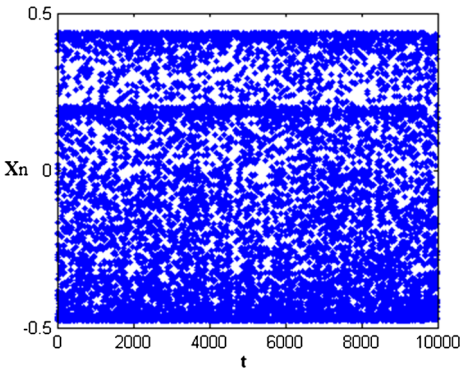


Fig. 6 Spatiotemporal diagram of Sinus-Power Logistic System

Table 1 Comparison between Lyapunov exponents of proposed method with others available chaotic maps

Chaotic maps	Logistic	Tent	Power logistic [24]	*LSS* [22]	Proposed map (*SPL*)	Proposed map (*CA*)
Lyapunov exponent	≅0.68	≅0.69	≅1.07	≅0.7	≅1.52	≅1.38

$$X_{n+1}(j) = (1-\varepsilon)f(X_n(j))\Big) + \frac{\varepsilon}{N-1}\sum_{i\neq j, i\in\{1,\ldots,N\}} f(X_n(i)) \tag{3}$$

Where X_n as the initial state condition is a vector of length N and ε is coupling coefficient, which is in range [0, 1]. The mapping function $f(x)$ is a chaotic map and we use Sinus-Power Logistic in this case, which is defined before in equation (2).

3 Proposed method

3.1 Encryption process

The overall view of encryption process has shown in Fig. 7. At first, we read the plain image and calculate cutting keys ($CK_{R,\ G,\ B}$) and averaging keys ($AK_{R,\ G,\ B}$) values for each color component distinctly. Then, multiply selected initial values by $AK_{R,\ G,\ B}$ to obtain three desired initial states in order to generate three chaotic sequences by using the *CA* map. After that, final eligible sequences with length N for each color component obtained by using a random selection method from previously generated sequences. Next, three Chaotic Matrix C_R, C_G and C_B are built by applying coupled map over to intercept sequences of previous step as initial sequences. Finally, the plain image permuted P times and each color component of the resulting matrix bitxored by its corresponding Chaotic Matrixes C_R, C_G and C_B to obtain an encrypted image.

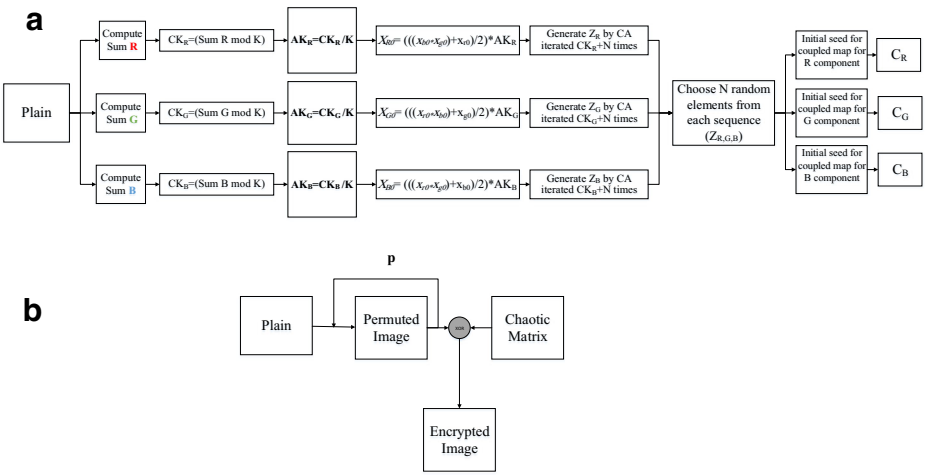


Fig. 7 Image encryption proposed method

Table 2 Parameters used in the encryption phase

Parameter	Definition	The values used in this paper
x_{r0}	Initial point for red component	0.3
x_{g0}	Initial point for green component	0.1
x_{b0}	Initial point for blue component	0.5
P	The iterations of permutation	3
K	A big number for generation intercept and modulation keys	10.001
ε	A control parameter for couple	0.12
r	A control parameter for Cosinus-Arcsines and Sinus-power logistic system	3.976, 3.465

It should be noted that the number of iterations, P, can be fixed when the desired permutation obtained and in our algorithm this number is not too big. Parameters and notation, which are used in this scheme has shown respectively in Tables 2 and 3.

The detailed steps of the proposed image cipher algorithm are as follows:

Step 1 Read plain image with size of $M \times N$, and create the plain matrix from it. $Plain_{ij}$ define the pixel value in the ith row and jth column of Plain, where $1 \le i \le M, 1 \le j \le N$.

Step 2 Initial values and parameters of chaotic maps are set in this step, which are consisting of: x_{r0}, x_{g0}, x_{b0}, r, for CA map, a big integer key K which used for producing cutting and averaging keys (CK, AK), and finally, r,ε values for Coupled map.

Step 3 Cutting and averaging key values calculated by below equations for each color component separately (three cutting keys and three averaging keys):

$$CK_{R,G,B} = \left(\sum_{i=1}^{M} \sum_{j=1}^{N} plain(i,j)_{R,G,B} \bmod K \right) \tag{4}$$

$$AK_{R,G,B} = \frac{CK_{R,G,B}}{K} \tag{5}$$

Table 3 Notation definition

Notation	Definition
M	Size of row
N	Size of column
X_{R0}, X_{G0}, X_{B0}	Modulated initial state conditions for each color component
$CK_{R,\ G,\ B}$	Cutting keys for each color component
$AK_{R,\ G,\ B}$	Averaging keys for each component
$Z_{R,\ G,\ B}$	Sequences generated from initial points for each component
$C_{R,\ G,\ B}$	Chaotic sequences generated from coupled map for each component

Where plain (i, j) is the pixel value in the ith row and jth column of plain. M and N are the size of plain image and K is a big integer number. As it's obvious in the above equations, cutting keys ($CK_{R,G,B}$), calculated based on whole pixels of the plain image, therefore it is dependent on the plain image. Hence if the image changed slightly, the values of these keys and consequently, the values of averaging keys also changed. This fact, made our scheme highly sensitive to its keys and plain image.

Step 4 In order to make initial state conditions dependent on each other and make new modulated initial state conditions we use equations (6) to (8). Where X_{R0}, X_{G0} and X_{B0} are modulated initial state conditions and x_{r0}, x_{g0}, x_{b0} are initial values for each colour component of the image. AK_R, AK_G, AK_B are Averaging keys that are produced in pervious step by using equation (5).

$$X_{R0} = \left(\left(\left(x_{b0} \times x_{g0}\right) + x_{r0}\right)/2\right) \times AK_R \tag{6}$$

$$K_{GO} = \left(\left(\left(x_{ro} \times x_{bo}\right) + x_{go}\right)/2\right) \times AK_G \tag{7}$$

$$X_{BO} = \left(\left(\left(x_{ro} \times x_{go}\right) + x_{bo}\right)/2\right) \times AK_B \tag{8}$$

Remark: As mentioned before, a key point is that our scheme highly sensitive to its keys and plain image. Indeed, we make initial state conditions dependent on plain image and each other, in order to prevent adversary from obtaining a partial image. Otherwise, if an adversary guess one of the initial state condition correctly, he can rebuild partial of plain image and see what is behind of encrypted image, but in this way, because of dependency between initial state conditions he/she wouldn't gain any useful information. (Fig. 8).

Step 5 In this step, initial state conditions which produced from the previous step use as an input for a CA map which iterate $CK_{R,\ G,\ B}+N$ times to produce three chaotic sequences entitled: Z_R, Z_G and Z_B. After that, N elements randomly select from each sequence by using pseudo-code which is provided below:

$$Start\ index_{R,\ G,\ B} = \left(CK_{R,G,B}\right)/2$$

With the help of above pseudo-code, we define the interception index of each color component sequence and start from determining position to select N consecutive elements $\left(\tilde{Z}_R, \tilde{Z}_G, \tilde{Z}_B\right)$. Finally, we multiply the intercepted sequences for each color component ($\tilde{Z}_{R,\ G,\ B}$) by 0.48 to be in the appropriate domain of SPL map, which used in coupled map.

$$\hat{Z}_{R,G,B} = \left(\tilde{Z}_{R,G,B}\right) \times 0.48$$

Then these sequences use as initial state to build $M \times N$ Chaotic Matrix C_R, C_G and C_B by using coupled map in a similar way and briefly denote with $C_{R,\ G,\ B}$ (eq. 3). Now, we build Chaotic Matrix C, row by row. First, The elements of $\hat{Z}_{R,G,B}$ use as zero rows C (0) (initial state conditions) that finally removed. Each row of chaotic matrix, for example $C\ (n+1)$ obtains from previous row, $C\ (n)$, with below formula:

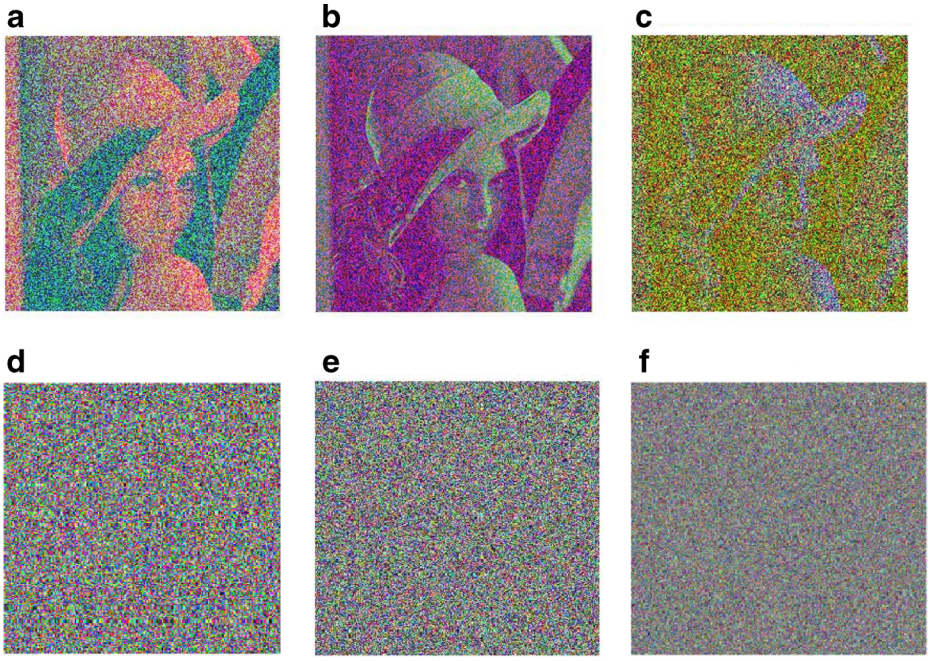


Fig. 8 Effect of no-dependency of initial states to each other and plain image. **a**: Intruder guess value of red component initial state, **b**: Intruder guess value of blue component initial state, **c**: Intruder guess value of red component initial state, **d**: Intruder guess value of red component initial state in proposed method, **e**: Intruder guess value of blue component initial state in proposed method, **f**: Intruder guess value of green component initial state in proposed method

$$C_{n+1}(j) = (1-\varepsilon)SPL(C_n(j)) + \frac{\varepsilon}{N-1}\sum_{i\neq j, i\in\{1,\ldots,N\}} SPL(C_n(i)) \tag{9}$$

Where ϵ is coupling coefficient.

Step 6 The plain image permuted P times. Overall overview of the proposed method for permutation is shown in Fig. 9. In the following, detailed procedure is explained. This method has dependency on plain image based on $CK_{R,\ G,\ B}$ values, which produced in the third step. Based on Fig. 9a, at first, we scroll down plain image row by row from left to right and put the elements in an array. Then we find the position of the first interception index, I_1, as below:

$$I_1 = \mathrm{mod}(SPL(\mathrm{mod})(AK, 0.48)) \times 10^9, M \times N \tag{10}$$

So we put the elements of this position,I_1, to end of this array into the first positions of a new matrix and the remaining elements put over them as shown in Fig. 9b, and build new plain. After that, new plain matrix divided into 3 segments. The first one is the main diagonal of the matrix (D), second part is an upper triangular matrix (T_U) and the last one is a lower triangular matrix (T_L). The first segment puts in VD vector, then, the upper triangular matrix scrolling down diagonally from left to right and put the elements into the vector respectively,

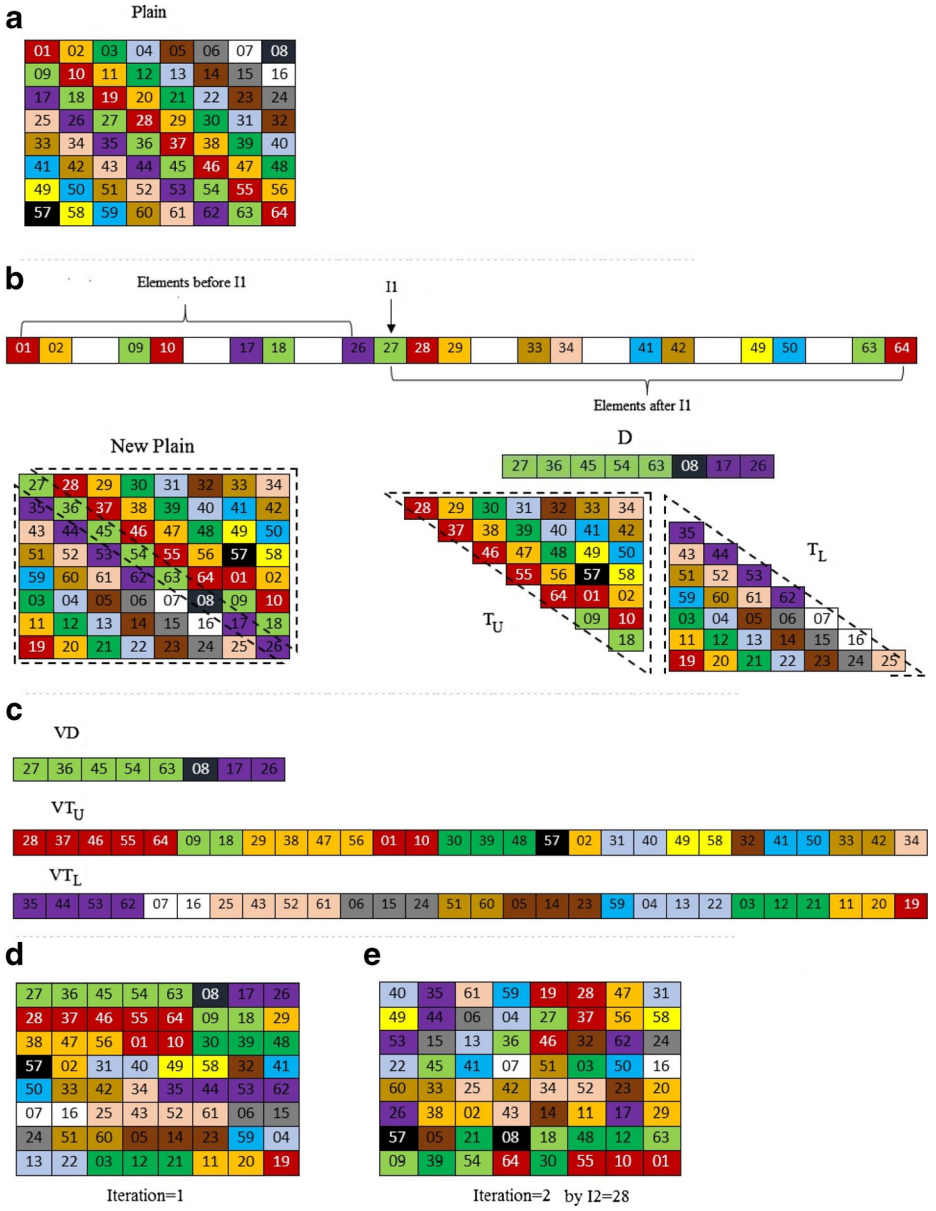


Fig. 9 A sample of permutation algorithm for a matrix 8×8, (**a**) elements in plain image, (**b**) Substituting position of elements based on value of I_1 and I_2, (**c**) Segmentation of new plain matrix into main diagonal (*VD*), upper triangular (VT_U) and lower triangular (VT_L) vector, (**d**) Result of first round permutation, (**e**) Result of second round permutation

which is called VT_U. After doing the same process for the lower triangular matrix VT_L vector obtained as shown in Fig. 9c. Finally, V vector is accomplished by equation (11):

$$V = VD||VT_U||VT_L \tag{11}$$

After that, V vector fill a matrix row by row and rebuild a matrix with a size of original image, as it shown in Fig. 9d for a *8x8* case. This process repeats P times with consider interception indices.

$$I_{j+1} = \mathrm{mod}\left(SPL\left(\mathrm{mod}\left(I_j \times CK, 0.48\right)\right) \times 10^9, M \times N\right) j = 1, \ldots, P \tag{12}$$

In Fig. 9e, the second round permutation result is shown. Where P is the number of iterations of the process which can be fixed when the best result obtained. According to the results, this number is not too large. Indeed, the advantage of this method is high transmogrification with lower iteration and satisfactory speed. The result of permuting Lena gray image with the proposed method, Arnold cat map, Baker map and Standard map with 3 iterations displayed in Fig. 10 and these result show this fact that the power of proposed method is much more than other available algorithms and with only 3 iterations it permutes the Lena image better than overmentioned schemes. Furthermore, we provide execution time and correlation comparisons between proposed method permutation algorithm and other available permutation methods for Lena gray level image that indicated in below (Table 4).

As it's obvious in the above table, our proposed algorithm has a better correlation results among other methods that proved high transmogrification with lower iteration. Indeed, the proposed algorithm disturbs pixel positions in such a way that minimize the correlation between adjacent pixels in horizontal, vertical and diagonal orientation. Although, the speed of Arnold Cat is a little better versus proposed algorithm, but as shown in Fig. 10, with 3 round permutation the result of Arnold Cat map is disappointing, Therefore our algorithm overcome Arnold cat map.

$$\textit{Chaotic Matrix} = \left(\mathrm{round}\left(10^{14} \times \left(C_{R,\,G,\,B}\right)\right)\right) \bmod 256 \tag{13}$$

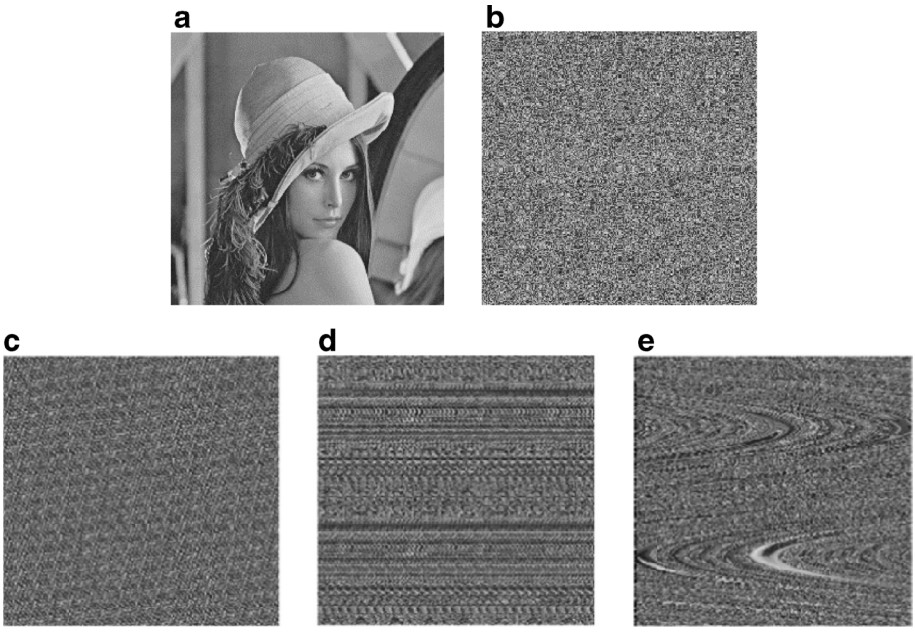


Fig. 10 Compression between proposed permutation method and Arnold cat map, Baker map and Standrad map: (**a**): Lena plain image, (**b**): 3 times iteration with proposed permutation algorithm (**c**): 3 times iteration with Arnold cat map, (**d**): 3 times iteration with Baker map, (**e**): 3 times iteration with Standard map

Table 4 Execution time comparison between proposed permutation method and other available permutation methods

Permutation	Rounds	Correlation of pixels			Time (ms)
		Horizontal	Vertical	Diagonal	
Proposed	3	0.0202	−0.0062	−0.0035	18.6
Arnold cat	3	−0.0273	−0.0626	−0.0106	14.2
Baker map	3	0.0062	0.1687	0.0438	138.6
Standard map	3	0.0477	0.1187	0.0482	187.1

Step 7 Reform the Chaotic Matrixes by (13), in order to put elements in range [0–255].

Where $C_{R,\ G,\ B}$ are chaotic matrixes which produces in step 5, round (x) is a function which gives nearest integer number to x.

Step 8 Finally, Chaotic Matrixes based on Fig. 7b, bitxored by permuted image to obtain the Encrypted Image.

3.2 Decryption process

Decryption phase is the reverse of encryption phase. The reciever party by having appropriate keys (x_{r0}, x_{g0}, x_{b0}, CK_R, CK_G, CK_B, r (CA), r (SPL), K, P and ε) can re-calculate $AK_{R,\ G,\ B}$ based on equation (5). After that, modulated initial state conditions re-build by multiply $AK_{R,\ G,\ B}$ with initial states (x_{r0}, x_{g0}, x_{b0}). We use them as an input for CA map, which iterate $CK_{R,\ G,\ B}+N$ times, this lead to production of chaotic sequences (Z_R, Z_G, Z_B). After that, the third party must use interception process for randomly selection of N elements based on predeterminded methodology that discussed before in step 5 of section 3.1. Finally, distinct Chaotic Matrixes (C_R, C_G, C_B) will be produced. Then, encrypted image de-permuted P times and bitxored with Chaotic Matrixes to achieve plain image.

4 Experimental results and security analysis

In our experiment, we use standard images to test, which consist of Lena, Baboon, Barbara, Peppers, Tree and etc. We use a notebook with windows 7 OS, Intel core i5, 2.40 GHZ CPU, using Matlab R2013b. The initial secret keys are selected without any restriction from predetermind intervals. We use (x_{r0}=0.3, x_{g0}=0.1, x_{b0}=0.5, K=10001, r=3.976, r=3.465 ε= 0.12 and p=3) values in our experiment as stated in Table 2. Moreover, cutting and averaging keys are calculated based on each plain image.

4.1 Key sensitivity

A secure encryption algorithm should be sensitive to all its keys. In order to evaluate this matter for our proposed method, we utilize two concepts, which are called, a number of pixel

Table 5 Quantified Key Sensitivity: *NPCR* and *UACI* test

Image		x_{r0}	x_{g0}	x_{b0}	K	ε	r (*CA*)	r (*SPL*)	Average
Lena	*NPCR*	99.6084	99.5921	99.5743	99.5962	99.6140	99.6038	99.6155	99.6006
	UACI	33.4567	33.4040	33.4708	33.3711	33.3332	33.4482	33.4744	33.4226
Barbara	*NPCR*	99.5967	99.6012	99.6007	99.6262	99.5850	99.6033	99.6119	99.6036
	UACI	33.4993	33.5011	33.5103	33.5623	33.4921	33.4807	33.4406	33.4980
Baboon	*NPCR*	99.5956	99.6145	99.6043	99.5936	99.6033	99.6211	99.6104	99.6061
	UACI	33.4231	33.4878	33.4883	33.4547	33.4379	33.4590	33.3824	33.4476
Tree	*NPCR*	99.6028	99.6185	99.6195	99.6190	99.6140	99.6099	99.6389	99.6191
	UACI	33.4983	33.4637	33.5275	33.5535	33.4131	33.4629	33.5259	33.5138

changing rate (*NPCR*) and unified average changing intensity (*UACI*). In order to calculate these concepts we use equations 14, 15 respectively:

$$NPCR = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(M,N)}{M \times N} \times 100\% \tag{14}$$

$$UACI = \frac{1}{M \times N}\left[\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{|E_1(M,N) - E_2(M,N)}{255}\right] \times 100\% \tag{15}$$

Parameters M, N denotes the size of encrypted image E_1 or E_2. The pixel values in position (m, n) are shown with E_1 (m, n) and E_2 (m, n). The amounts of D (m, n) are determined by the values E_1 (m, n) and E_2 (m, n), such that if E_1 (m, n)=E_2 (m, n) then D (m, n)=0, otherwise it's equal to 1. We do this test for some standard images and the result for them displayed in Table 5. In order to do this test, we should slightly change the values of key, which is equal to 10^{-14} for initial state conditions and for integers K this tiny change equal to 1 [8]. That's clear based on the above table, if we do very trivial change in each key, the result of encryption changed and this can be deduced based on *NPCR* and *UACI* values. Therefor this shows sensitivity to all keys of the proposed method. Moreover, in this part we show the dependency of permutation algorithm on the $CK_{R,\ G,\ B}$ values, as you can see on the below table, if we change a bit $CK_{R,\ G,\ B}$ values, the result of permutation phase will be changed and its effect can be observed from *NPCR* and *UACI* (Table 6).

Table 6 Permutation algorithm sensitivity based on $CK_{R,\ G,\ B}$, just one value (±1) increase/decrease to $CK_{R,\ G,\ B}$

Image	Value of *CK*+1		Value of *CK*-1	
	$NPCR_{Average}$	$UACI_{Average}$	$NPCR_{Average}$	$UACI_{Average}$
Lena	99.6211	33.5113	99.5951	33.3996
Barbara	99.6094	33.4498	99.6195	33.5146
Baboon	99.6190	33.4589	99.5906	33.4688
Tree	99.6262	33.5053	99.6104	33.4771

Table 7 Quantified key space size

Encryption algorithm	Proposed method	Ref [10]	Ref [19]	Ref [27]	Ref [16]	Ref [6]	Ref [18]	Ref [15]	Ref [23]	Ref [17]
Key space size	$>2^{572}$	2^{256}	2^{218}	$\cong 2^{278}$	2^{120}	$>2^{100}$	2^{256}	$>2^{100}$	2^{233}	2^{216}

4.2 Key space

Calculating key space has a vital role in determining the security of the proposed method. Indeed, if the key space is large, the method will be resistant against brute-force attack. In the proposed method, 11 hidden parameters are considered as keys. In software implementation, this parameters utilizes in double format 52 bits [5]. In Table 7, we do a compression between our method and some other methods.

4.3 Plain text sensitivity and differential attack analysis

The proposed method in order to be resistant against differential attack should be intensely dependent to plain image, in other words, it means a very vital change in plain image cause big changes in the ciphered image [22]. Two equations (14), (15) are used for this analysis. The results of this analysis have been shown in Table 8.

As you see in Table 8, the average of *NPCR* is more than 99.67 and the average of *UACI* is greater than 33.45. We provide a comparison between some available and our proposed methods for *NPCR* and *UACI* values of Lena image which shown in Table 9. As obvious based on Chart 1, our proposed method has better values alongside all of availabale methods. Another test which is shown plain-text sensitivity shown in Fig. 11.

Figure 11a is a plain image of couple and Fig. 11b is its corresponding image with one pixel different from the original image (128,128). As it's obvious in Fig. 11c, when we subtract two images, we have a black plane with one white pixel, which is because of differences on plain image, but in Fig. 11f, when we subtract two encrypted images it does not lead to black plane, which is the result of plain text sensitivity.

Table 8 *NPCR*, *UACI* of test images

Image	$NPCR_R$	$NPCR_G$	$NPCR_B$	$UACI_R$	$UACI_G$	$UACI_B$	Average *NPCR*	Average *UACI*
Lena	99.6917	99.6887	99.6704	33.5418	33.5327	33.5164	99.6836	33.5303
Baboon	99.6688	99.6582	99.6795	33.3899	33.5848	33.3953	99.6688	33.4566
Barbara	99.6765	99.6841	99.6856	33.4359	33.6158	33.4495	99.6820	33.5004
Pepper	99.6734	99.6780	99.6627	33.4452	33.4904	33.5966	99.6713	33.5107
Tree	99.6673	99.6643	99.6841	33.2909	33.4836	33.5512	99.6719	33.4419
Girl	99.6810	99.6734	99.7024	33.5140	33.6405	33.5288	99.6856	33.5611
Splash	99.6765	99.6734	99.6978	33.6491	33.5590	33.4425	99.6825	33.5502
House	99.6826	99.6734	99.6795	33.5295	33.3113	33.5130	99.6785	33.4512
Couple	99.6765	99.6749	99.6780	33.3860	33.4235	33.6509	99.6764	33.4868

Table 9 Comparison of $NPCR_{R,\ G,\ B}$ and $UACI_{R,\ G,\ B}$ on Lena image

Algorithm	$NPCR_{Avrage}$	$UACI_{Avrage}$
Proposed algorithm	99.6836	33.5303
Hussain and Gondal's scheme [10] (2014)	99.6123	32.6237
Seyedzadeh and et al.'s scheme [19] (2015)	99.6312	33.5134
Sam et al.'s scheme [17]	99.6027	33.4685
Li and Liu's scheme [14]	99.6836	33.4647
Zhang and Xiao's scheme [25] (2014)	99.6665	33.4245
Wie et al.'s scheme [23] (2014)	99.6260	33.3846
Dong's scheme [6] (2014)	99.6433	33.5892
Seyedzadeh and Mirzakuchaki's scheme [18] (2014)	99.6828	33.4898
Liu et al.'s scheme [15] (2015)	99.6216	33.4158

4.4 Information entropy analysis

In information theory, entropy is a scale for showing randomness in information. This scale can be calculated by using equation (16):

$$H(s) = -\sum_{i=0}^{2^{Q}-1} P(s_i)\log_2 P(s_i) \tag{16}$$

In above equation, $P(s_i)$ is likelihood frequency of symbol $s_i \circ \in s$, Q is the number of bits which used for display symbol s_i and log_2 is base 2 logarithm. The ideal value of entropy is 24. So when the entropy value is less than ideal amount, a degree of predictability may be exist and it will violate security. The Result of this test for Lena image depicted in Table 10.

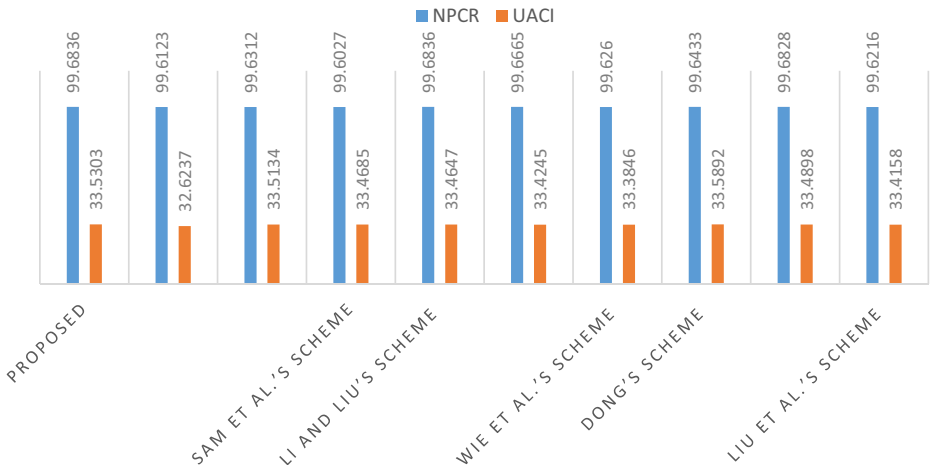


Chart 1 Comparision of *NPCR* and *UACI* on Lena image

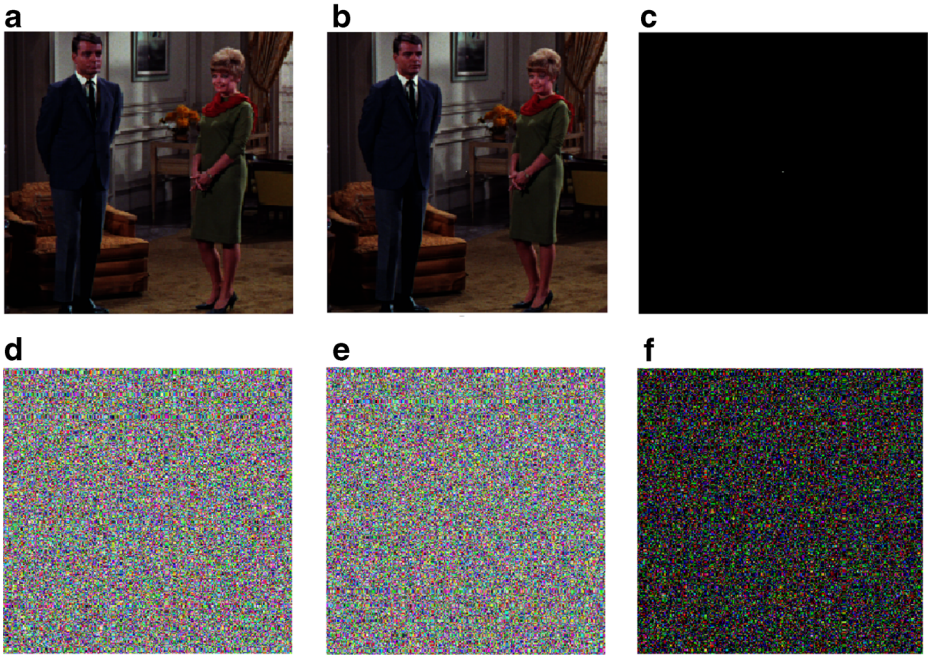


Fig. 11 Plain-text sensitivity: (**a**), (**b**) Couple plain and with one pixel different image, (**d**), (**e**) corresponding encrypted image of (**a**), (**b**). (**c**), (**f**) differential of corresponding plain and encrypted images

4.5 Known-and chosen plaintext analysis

Due to the fact that in proposed method cutting and averaging keys build from plain image, then for each image, the value of these keys will be changed. Hence, we have generated different sequences for each color component distinctly to resist agianst Known and Chosen plaintext attacks.

Table 10 The information entropy of plain/cipher for Lena image

Algorithm	Encrypted image			Average of encrypted image
	Red	Green	Blue	
Proposed Method	7.9972	7.9972	7.9976	7.9973
Hussain and Gondal's scheme [10] (2014)	7.9972	7.9973	7.9971	7.9972
Seyedzadeh and et al.'s scheme [19] (2015)	7.9973	7.9970	7.9971	7.9971
Sam et al.'s scheme [17]	7.9971	7.9969	7.9970	7.9970
Wie et al.'s scheme [23] (2014)	7.9971	7.9969	7.9962	7.9967
Seyedzadeh and Mirzakuchaki's scheme [18] (2014)	7.9973	7.9972	7.9969	7.9971
Dong's scheme [6] (2014)	7.9901	7.9912	7.9921	7.9911
Liu et al.'s scheme [15] (2015)	7.9808	7.9811	7.9814	7.9811
Zhang and Xiao's scheme [25] (2014)	7.9973	7.9968	7.9972	7.9971

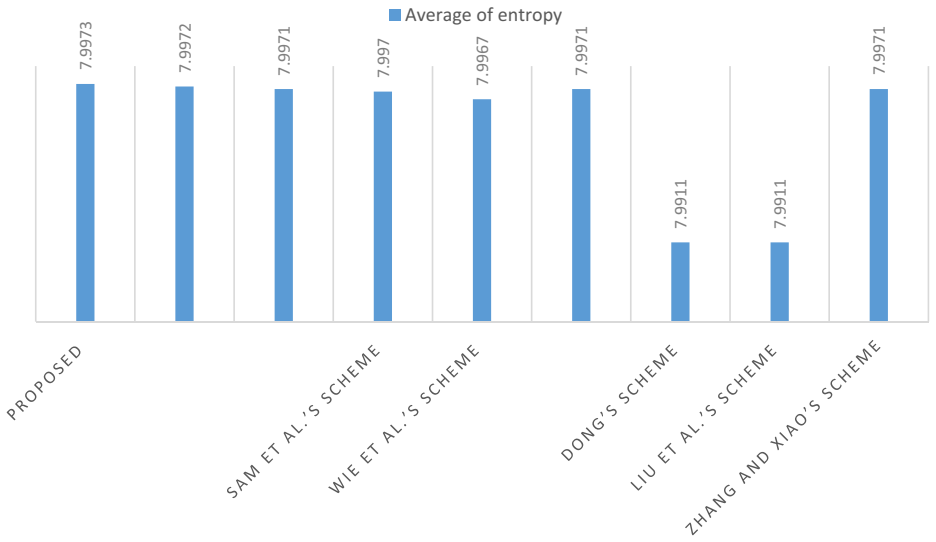


Chart 2 Comparision of the information entropy on lena image

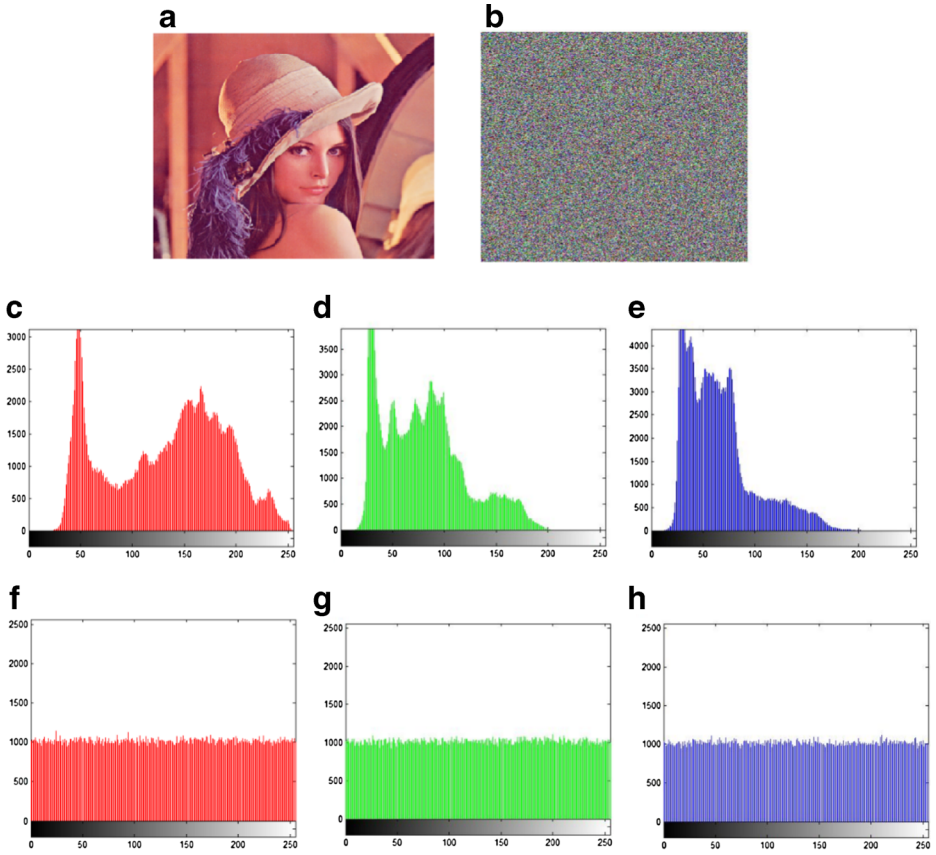


Fig. 12 Image histogram: (**a**) Lena image, (**b**) Lena ciphered, (**c**), (**d**), (**e**): Histogram of the plain image for red, green and blue component, (**f**), (**g**), (**h**): Histogram of ciphered image for *red*, *green* and *blue* component

Table 11 Average correlation of two adjacent pixels in horizontal, vertical, diagonal of Lena

Algorithm	Lena	Orientation		
		Horizontal	Vertical	Diagonal
	Plain	0.9309560	0.9570049	0.9001547
Proposed algorithm	Cipher	−0.0030859	0.0025108	−0.0000938
Hussain and Gondal's scheme [10] (2014)	Cipher	0.0045000	−0.0050000	−0.0025000
Seyedzadeh and et al.'s scheme [19] (2015)	Cipher	0.0035330	0.0029660	0.0015666
Seyedzadeh and Mirzakuchaki's scheme [18] (2014)	Cipher	0.0005950	0.0008390	0.0011240
Liu et al.'s scheme [15] (2015)	Cipher	0.0030000	0.0051000	−0.0030000
Wie et al.'s scheme [23] (2014)	Cipher	0.0044000	0.0034000	0.0020000
Sam et al.'s scheme [17]	Cipher	0.0026000	0.0039000	0.0037000
Li and Liu's scheme [14]	Cipher	0.0075000	0.0129000	0.0011000
Zhang and Xiao's scheme [25] (2014)	Cipher	0.0022326	0.0030550	0.0030146
Dong's scheme [6] (2014)	Cipher	−0.0026000	−0.0058000	−0.0024000

As it is clear based on chart 2, the average of the information entropy for our proposed scheme is the best alongside other proposed methods.

4.6 Image histogram

The encrypted image resistant against statistical attack when the color histogram of image distributed uniformly. The image histogram of Lena plain image and its ciphered are depicted in Fig. 12.

4.7 Correlations of two adjacent pixels

Each pixel in the image is intensely correlated with its adjacent pixels either in horizontal, vertical or diagonal direction. Hence, the proposed method should remove this correlation

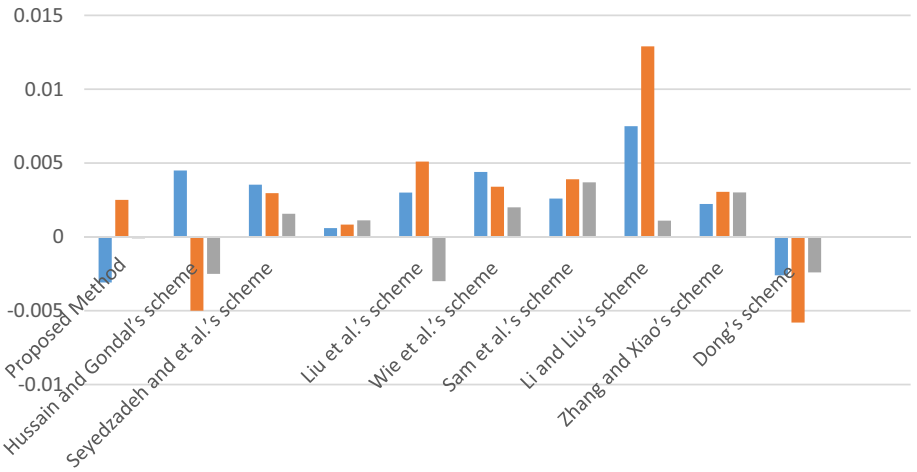


Chart 3 Comparison of correlation coefficient for Lena image

between adjacent pixels in order to be resistant against statistical attack. In an ideal condition, there should be no correlation between adjacent pixels, and its value (correlation coefficient) should be almost 0. In order to calculate CC (correlation coefficient) we use the following equations:

$$r_{xy} = \mathrm{cov}(x,y)/\sqrt{D(x)D(y)} \tag{17}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{18}$$

$$D(x) = \frac{1}{N}\sum\nolimits_{i=1}^{N} (x_i - E(x))^2 \tag{19}$$

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1} (x_i - E(x))(y_i - E(y)) \tag{20}$$

Where x and y are the gray value of two adjacent pixels in the image, cov (x, y) is covariance, D (x) is variance and E (x) is mean. The result of this test for Lena image is shown in Table 11. As it is obvious in chart 3, the calculated correlation coeffeicient values for proposed method are −0.0030859, 0.0025108 and −0.0000938 for horizontal, vertical and diagonal orientation, respectively.

5 Conclusion

To conclude, we proposed a new scheme for image encryption based on two new chaotic maps with high Lyapunov exponents and the new introduced permutation scheme. The advantages of the new permutation scheme is low computational overhead, because its procedure is so simple, and achieve top transmogification in fewer steps versus available permutation algorithms. Furthermore, our permutation algorithm has a convincing speed that is 18.6 ms, although its speed is a little higher than Arnold Cat map, but as it's obvious in Fig. 10, our proposed algorithm disturbs pixels more effectively than Arnold cat map. We use two strong chaotic mappings alongside coupled map for the initial seed production and pixel diffusion. The proposed permutation algorithm used to permute pixels. We use cutting and averaging key to make our approach resistant against known and chosen plain text attack, because they are built from plain images and these are dependant to each other. As it's obvious from experimental results and security analysis, our method has a good performance and good resistant against statistical and differential attack.

References

1. Arshad H, Nikooghadam M (2014) Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems. J Med Syst 38(12):1–12

2. Arshad H, & Nikooghadam M (2014) An efficient and secure authentication and key agreement scheme for session initiation protocol using ECC. Multimed Tools Applic 1–17. doi: 10.1007/s11042-014-2282-x
3. Arshad H & Nikooghadam M (2015) Security analysis and improvement of two authentication and key agreement schemes for session initiation protocol. J Supercomput 1–18. doi: 10.1007/s11227-015-1434-8
4. Arshad H, Teymoori V, Nikooghadam M, & Abbassi H (2015) On the security of a two-factor authentication and key agreement scheme for telecare medicine information systems. J Med Syst 39. doi: 10.1007/s10916-015-0259-6
5. Chen J, Zhou J, Wong K-W (2011) A modified chaos-based joint compression and encryption scheme. Circ Syst II: Express Briefs, IEEE Trans on 58(2):110–114
6. Dong CE (2014) Color image encryption using one-time keys and coupled chaotic systems. Signal Process Image Commun 29(5):628–640
7. Gao H et al (2006) A new chaotic algorithm for image encryption. Chaos, Solitons Fractals 29(2):393–399
8. Huang X (2012) Image encryption algorithm using chaotic Chebyshev generator. Nonlinear Dynam 67(4): 2411–2417
9. Huang C, Nien H (2009) Multi chaotic systems based pixel shuffle for image encryption. Opt Commun 282(11):2123–2127
10. Hussain I, Gondal MA (2014) An extended image encryption using chaotic coupled map and S-box transformation. Nonlinear Dynam 76(2):1355–1363
11. Ismail IA, Amin M, Diab H (2010) A digital image encryption algorithm based a composition of two chaotic logistic maps. IJ Network Sec 11(1):1–10
12. Kaur R and Singh EK (2013) Image encryption techniques: a selected review. IOSR J Comput Eng (IOSR-JCE) e-ISSN 2278–0661
13. Khan M, Shah T (2014) A literature review on image encryption techniques. 3D Res 5(4):1–25
14. Li J, Liu H (2013) Colour image encryption based on advanced encryption standard algorithm with two-dimensional chaotic map. Inform Sec, IET 7(4):265–270
15. Liu H, Kadir A, Gong P (2015) A fast color image encryption scheme using one-time S-Boxes based on complex chaotic system and random noise. Opt Commun 338:340–347
16. SaberiKamarposhti M et al (2014) Using 3-cell chaotic map for image encryption based on biological operations. Nonlinear Dynam 75(3):407–416
17. Sam IS, Devaraj P, Bhuvaneswaran R (2012) An intertwining chaotic maps based image encryption scheme. Nonlinear Dynam 69(4):1995–2007
18. Seyedzadeh SM, Mirzakuchaki S (2012) A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. Signal Process 92(5):1202–1215
19. Seyedzadeh SM, Norouzi B, Mosavi MR. & Mirzakuchaki S (2015) A novel color image encryption algorithm based on spatial permutation and quantum chaotic map. Nonlinear Dynam 1–19
20. Solak E, Rhouma R, Belghith S (2010) Cryptanalysis of a multi-chaotic systems based image cryptosystem. Opt Commun 283(2):232–236
21. Wang Y, Wong KW, Liao X, Chen G (2011) A new chaos-based fast image encryption algorithm. Appl Soft Comput 11(1):514–522
22. Wang Y et al (2011) A new chaos-based fast image encryption algorithm. Appl Soft Comput 11(1):514–522
23. Wei X et al (2012) A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system. J Syst Softw 85(2):290–299
24. Zhang X and Cao Y (2014) A novel chaotic map and an improved chaos-based image encryption scheme. Sci World J
25. Zhang Y, Xiao D (2014) Self-adaptive permutation and combined global diffusion for chaotic color image encryption. AEU-Int J Electron Commun 68(4):361–368
26. Zhang Y., et al (2010) A new image encryption algorithm based on Arnold and coupled chaos maps. in Computer and Communication Technologies in Agriculture Engineering (CCTAE), 2010 International Conference On. IEEE
27. Zhou Y, Bao L, Chen CP (2014) A new 1D chaotic system for image encryption. Signal Process 97:172–182
28. Zhu C (2012) A novel image encryption scheme based on improved hyperchaotic sequences. Opt Commun 285(1):29–37
29. Zhu ZL, Zhang W, Wong KW, Yu H (2011) A chaos-based symmetric image encryption scheme using a bit-level permutation. Inf Sci 181(6):1171–1186

Mollaeefar Majid was born in 1989 in Gonbad-e Kavus, Golestan Province. He received his B.Sc. degree in information technology from Tabari University in June 2013. Nowadays, he study secure communication in Imam Reza International University of Mashhad, attendant for M.Sc. degree. His filed of interest are image processing, cloud security, steganography, covert channel and security.

Sharif Amir was born in 1990 in Birjand, south Khorasan Province. He received his B.Sc. degree in information technology from Birjand University in June 2012. Nowadays, he study secure communication in Imam Reza International University of Mashhad, attendant for M.Sc. degree. His filed of interest are image processing, steganography, cryptography, covert channel and security.

Mahboubeh Nazari received B.Sc. degree from Ferdowsi University of Mashhad, Iran, in 2006, M.Sc. degree from Ferdowsi University of Mashhad, Iran, in 2008 and Ph.D. degree from Ferdowsi University of Mashhad, Iran, in 2013. She is Adjunct professor in Department of Mathematics at Ferdowsi University of Mashhad, Iran. Her research focuses on Dynamical Systems, Chaos theory and it's applications in Cryptography, Network Security and data Security.