ATLANTIS PRESS

Advances in Engineering Research, volume 138

2nd International Conference on Machinery, Electronics and Control Simulation (MECS 2017)

Double color images encryption based on DNA sequences and block permutation

Kunshu Wang[1,a], Xiangjun Wu[2,b,*] and Baoqiang Liu[1,c]

[1]School of Computer and Information Engineering, Henan University, Kaifeng, China

[2]College of Software, Henan University, Kaifeng, China

[a]email: 972454806@qq.com, [b]email: wuxj@henu.edu.cn, [c]email: 190334640@qq.com

*corresponding author

Keywords: double color images; LSS and 2D-LASM; block permutation; DNA sequences.

Abstract: In this paper, we propose a double color images encryption scheme based on Deoxyribonucleic Acid (DNA) sequence, chaotic system and block permutation. Mean Square Error (MSE) of the double images, LSS and 2D-LASM chaotic systems are firstly employed to generate the key streams. Then, divide the plain-images into equal blocks and scramble the blocks using the key streams. Next, encode the permuted images into the DNA matrices according to the DNA encoding rules and a DNA XOR operation is carried out on the DNA matrices. Finally, decode the encrypted the DNA matrices via the DNA decoding rules, and a pixel-level diffusion is further adopted to strengthen the security and sensitivity of the cryptosystem, and the resulting cipher-images are obtained. The experimental results show that the presented encryption algorithm has the advantages of large key space and high security, and fast encryption/decryption speed.

1. Introduction

With the rapid development of information and network technologies, a great many of digital multimedia data are transmitted over the public networks. It is necessary to protect the image information against the unauthorized access, usage, disruption and destruction. In the past years, many encryption methods have been proposed, such as DES, IDEA and RSA etc. However, these traditional encryption schemes are unsuitable for the digital images, because some inherent features of the images such as bulk data capacity, and strong correlation among pixels and high computation complexity [1,2].

Considering the desirable properties of extreme sensitivity to initial conditions and parameters, the close relationship between chaos and cryptography has been revealed [3]. The simple low-dimensional (LD) chaotic systems can be realized conveniently in engineering due to their simple structure, but they also have some drawbacks such as limited/discontinuous chaotic range, the non-uniform data distribution, and the vulnerability to low-computation-cost analysis using iteration and correlation functions [4]. In contrast to the LD chaotic systems, the high-dimensional (HD) chaotic systems have complex structures and chaotic behaviors. Unfortunately, the complex structure and multiple parameters of the HD chaotic systems increase the cost of hardware/software implementations and the computation complexity [5]. Hence, considering the trade-offs between the security and speed, it is essential to design the image encryption algorithms based on the LD chaotic systems/maps with excellent performance.

Since the pioneering work of Adleman [6], DNA computing has gained growing attention from many researchers in various fields such as biology, chemistry, mathematics, computer science etc. Recently, DNA-based image encryption has aroused great interest [7-15]. For example, Liu et al. [7] proposed a color image encryption algorithm based on DNA encoding and the Logistic map. But this cryptosystem is very weak to a chosen-plaintext attack [8]. An image fusion encryption scheme based on DNA sequence operation and Chen hyper-chaotic system was introduced in [9], which can be cracked by the chosen-plaintext attack [10]. To enhance the security of the cryptosystem, Zhang

et al. [11] shuffled the position and value of pixels by using combination of chaotic systems and DNA encoding to implement encryption of digital image and using quaternary DNA encoding instead of binary encoding. Unfortunately, the previous two schemes in [11] are weak against differential attacks. Further, many reported DNA-based image encryption algorithms [12-15] used the DNA sequence operations such as addition, subtraction and XOR to diffuse the DNA-encoded image with absence of DNA-level confusion, which makes the security of the cryptosystems not high enough.

Motivated by the above discussions, in this paper, a double color images encryption based on DNA sequences and block permutation is proposed. Firstly, the key streams are generated from LSS, 2D-LASM and the plain-images. Secondly, two plain-images are decomposed into some blocks, which are shuffled using the key streams. Thirdly, the scrambled images are encoded into the DNA matrices by the DNA encoding rules and a DNA XOR operation is performed on the DNA matrices. Finally, the DNA matrices are decoded by the DNA decoding rules, and the values of the image pixels are modified using the key streams. Thus the resulting cipher-images are obtained.

Then, divide the plain-images into equal blocks and scramble the blocks using the key streams. Next, encode the permuted images into the DNA matrices according to the DNA encoding rules and a DNA XOR operation is carried out on the DNA matrices. Finally, decode the encrypted the DNA matrices via the DNA decoding rules, and a pixel-level diffusion is further adopted to strengthen the security and sensitivity of the cryptosystem, and the resulting cipher-images are obtained. The experimental results demonstrate that our presented method is effective, feasibility and secure.

2. Preliminaries

2.1. DNA sequence encryption

A DNA sequence contains four nucleic acid bases: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T), where ‘A’ and ‘T’, ‘C’ and ‘G’ are complementary pairs, respectively. In the binary system, ‘0’ and ‘1’ are complementary. So the binary numbers ‘00’ and ‘11’, ‘01’ and ‘10’ are also complementary. Thus one can utilize four bases ‘A’, ‘T’, ‘C’, ‘G’ to encode the binary numbers ‘00’, ‘11’, ‘01’, ‘10’. There are total 24 types of DNA coding rules. However, only eight kinds of them which listed in Table 1 satisfying the Watson Crick complementary rule [16]. For an 8-bit image, each pixel value can be firstly transformed into a binary sequence with length 8. Then the binary stream can be encoded to a DNA sequence with length 4 according to the DNA encoding rules in Table 1.

The exclusive or (XOR) operation for DNA sequences are adopted to encrypt and decrypt the digital images. The XOR operation for DNA sequences are executed by the traditional binary XOR. So if we have eight kinds of DNA encoding rules, there also exist eight kinds of DNA XOR rules. For example, according to the DNA encoding Rule 5 in Table 1, the DNA XOR operation is illustrated in Table 2.

Table 1 The encoding and decoding rules for DNA sequences.

	Rule 0	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7
A	11	00	00	01	01	10	10	11
C	10	01	10	00	11	00	11	01
G	01	10	01	11	00	11	00	10
T	00	11	11	10	10	01	01	00

Table 2 The XOR operation for DNA sequences.

XOR	*A*	*C*	*G*	*T*
A	*C*	*A*	*T*	*G*
C	*A*	*C*	*G*	*T*
G	*T*	*G*	*C*	*A*
T	*G*	*T*	*A*	*C*

2.2. Chaotic maps

To overcome the limitations of 1D chaotic systems, the Logistic-Sine system (LSS) and a new 2D Logistic-adjusted-Sine map (2D-LASM) are introduced in [17,18]. The LSS and 2D-LASM have proved to be more suitable for image encryption compared with the Logistic system and the Sine system for their excellent chaotic performance.

In our paper, LSS and 2D-LASM are used to generate the key streams, which can be separately described as follows:

$$X_{n+1} = \mathrm{mod}\left(rX_n\left(1-X_n\right)+(4-r)\sin\left(\pi X_n\right)/4,\ 1\right), \tag{1}$$

$$\begin{cases} Y_{n+1} = \sin\left(\pi\mu\left(Z_n+3\right)Y_n\left(1-Y_n\right)\right) \\ Z_{n+1} = \sin\left(\pi\mu\left(Y_{n+1}+3\right)Z_n\left(1-Z_n\right)\right) \end{cases}, \tag{2}$$

where $r\in(0,\ 4]$, $\mu\in[0,\ 1]$ and $X_n, Y_n, Z_n\in(0,1)$.

2.3. Updating the initial conditions and parameters

In our method, the parameters r, μ and initial conditions X_0, Y_0, Z_0 of LSS and 2D-LASM are used as the secret keys. Compute the Mean Squared Error (MSE) by the following equation:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I_0(i,j)-P_0(i,j)\right]^2, \tag{3}$$

where M and N are the width and height of the test image, respectively.

The parameters and initial conditions of LSS and 2D-LASM can be updated as follows:

$$r' = r - MSE(1)/8^5, \tag{4}$$

$$\mu' = \mu + MSE(2)/8^5, \tag{5}$$

$$X_0' = X_0 - \left(MSE(2)+MSE(1)\right)/8^4, \tag{6}$$

$$Y_0' = Y_0 + \left(MSE(1)+MSE(3)\right)/8^4, \tag{7}$$

$$Z_0' = Z_0 + \left(MSE(1)+MSE(2)-MSE(3)\right)/8^4. \tag{8}$$

2.4. Generating the key streams

Iterate the chaotic maps (1) and (2) for $l+MN$ $(l\geq 1000)$ times by using the updated parameters and initial conditions. To avoid the harmful effect of transitional procedure, we discard the first l numbers and obtain three chaotic sequences with length MN, i.e., $X=\{x_1,x_2,x_3,\cdots\cdots,x_{MN}\}$, $Y=\{y_1,y_2,y_3,\cdots\cdots,y_{MN}\}$ and $Z=\{z_1,z_2,z_3,\cdots\cdots,z_{MN}\}$. To improve the stochastic properties of the chaotic sequences, we process three chaotic sequences as follows:

$$x'_i = 10^{\lambda}\times x_i - \mathrm{Round}\left(10^{\lambda}\times x_i\right), \tag{9}$$

$$y'_i = 10^{\lambda}\times y_i - \mathrm{Round}\left(10^{\lambda}\times y_i\right), \tag{10}$$

$$z'_i = 10^{\lambda}\times z_i - \mathrm{Round}\left(10^{\lambda}\times z_i\right), \tag{11}$$

where the function Round(x) rounds the value of x to the nearest integer, $i=1,2,\cdots,MN$, and

$\lambda \in [5, 14]$. Thus three new pseudorandom sequences X', Y' and Z' are obtained.

In what follows, the key streams $S1$, $S2$, $K1$ and $K2$ are generated by the following formulae:

$$[V1,\ S1] = \text{Sort}\left(X'\left(MN/16+1:2MN/16\right), \text{'Descend'}\right), \tag{12}$$

$$[V2,\ S2] = \text{Sort}\left(X'\left(1:MN/16\right), \text{'Descend'}\right), \tag{13}$$

$$\bar{Y} = \text{Reshape}\left(Y',\ M,\ N\right), \tag{14}$$

$$\bar{Z} = \text{Reshape}\left(Z',\ M,\ N\right), \tag{15}$$

$$K1 = \text{Mod}\left[\text{Round}\left(\text{Abs}\left(\bar{Y}\left(i,j\right)\times 10^{14}\right)\right), 256\right], \tag{16}$$

$$K2 = \text{Mod}\left[\text{Round}\left(\text{Abs}\left(\bar{Z}\left(i,j\right)\times 10^{13}\right)\right), 256\right], \tag{17}$$

where $i = 1,2,\cdots,M$ and $j = 1,2,\cdots,N$. $[\Psi,\Xi] = \text{Sort}(\Gamma,\text{mode})$ sorts the elements of Γ in ascending or descending order, and returns an index matrix Ξ and a sorted sequence Ψ. $\text{Reshape}\left(\Lambda,M,N\right)$ returns the M-by-N matrix whose elements are taken columnwise from Λ. Mod(x, y) returns the remainder after division, Abs(x) returns the absolute value of x.

3. Encryption algorithm

The proposed double color image encryption algorithm can be described in detail as follows:

Step 1: Convert the original plain-images I_0, P_0 into the red, green and blue matrices separately, i.e., IR, IG, IB, PR, PG and PB, where the size of each matrix is $M \times N$. Divide the component matrices IR, IG, IB, PR, PG and PB into non-overlapping blocks with size 4×4, and shuffle these blocks using the key streams $S1$ and $S2$. Then recombine these blocks and get the scrambled matrices IR_1, IG_1, IB_1, PR_1, PG_1 and PB_1.

Step 2: Transform the matrices IR_1, IG_1, IB_1, PR_1, PG_1 and PB_1 into the binary matrices with size $M \times 8N$, respectively. By the random numbers d_1, d_2, d_3 and the DNA encoding rules in Table 1, encode these binary matrices and obtain six DNA sequence matrices IR_2, IG_2, IB_2, PR_2, PG_2 and PB_2 with size $M \times 4N$. The random numbers d_1, d_2 and d_3 are computed as follows:

$$d_1 = \text{Fix}\left(\text{Mod}\left(Z\left(M\right)\times 10^{10}, 8\right)\right), \tag{18}$$

$$d_2 = \text{Fix}\left(\text{Mod}\left(X\left(MN\right)\times 10^{9}, 8\right)\right), \tag{19}$$

$$d_3 = \text{Fix}\left(\text{Mod}\left(Y\left(2N\right)\times 10^{11}, 8\right)\right), \tag{20}$$

where Fix(x) rounds the value of x to the nearest integer toward zero.

Step 3: Perform the DNA XOR operation in Table 2 on the DNA matrices IR_2, IG_2, IB_2, PR_2, PG_2 and PB_2 by the following formulae:

$$\begin{cases} IR_3\left(i,j\right) = \text{OperXOR}\left(IR_2\left(i,j\right), PB_2\left(i,j\right)\right) \\ IG_3\left(i,j\right) = \text{OperXOR}\left(IR_3\left(i,j\right), PG_2\left(i,j\right)\right), \\ IB_3\left(i,j\right) = \text{OperXOR}\left(PB_2\left(i,j\right), IG_3\left(i,j\right)\right) \end{cases} \tag{21}$$

$$\begin{cases} PR_3\left(i,j\right) = \text{OperXOR}\left(IB_3\left(i,j\right), PR_2\left(i,j\right)\right) \\ PG_3\left(i,j\right) = \text{OperXOR}\left(IG_2\left(i,j\right), PR_3\left(i,j\right)\right), \\ PB_3\left(i,j\right) = \text{OperXOR}\left(IB_2\left(i,j\right), PG_3\left(i,j\right)\right) \end{cases} \tag{22}$$

where $i = 1,2,\cdots,M$, $j = 1,2,\cdots,4N$.

Step 4: According to the DNA decoding rules in Table 1 and the random numbers d_1, d_2 and d_3, convert the DNA matrices IR_3, IG_3, IB_3, PR_3, PG_3 and PB_3 into the binary matrices with size $M \times 8N$, respectively. Then transform the binary matrices into the decimal matrices IR_4, IG_4, IB_4, PR_4, PG_4 and PB_4 with size $M \times N$ separately.

Step 5: To further strengthen the security and sensitivity of the image cryptosystem, using the key streams $K1$ and $K2$, a pixel-level diffusion process is carried out as follows:

$$\begin{cases} IR_5(i,j) = PR_4(i,j) \oplus K1(i,j) \\ IG_5(i,j) = IR_4(i,j) \oplus PG_4(i,j), \\ IB_5(i,j) = IG_5(i,j) \oplus PB_4(i,j) \end{cases} \tag{23}$$

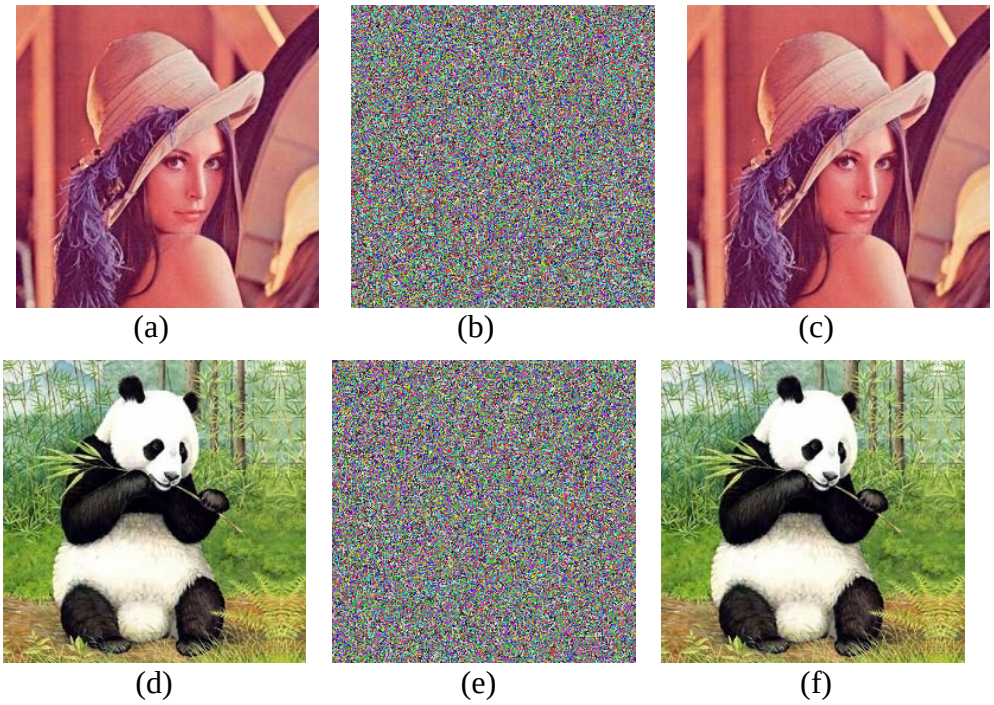
$$\begin{cases} PG_5(i,j) = IR_4(i,j) \oplus K2(i,j) \\ PR_5(i,j) = IG_4(i,j) \oplus PG_5(i,j), \\ PB_5(i,j) = PR_5(i,j) \oplus IB_4(i,j) \end{cases} \tag{24}$$

where $i = 1,2,\cdots,M$, $j = 1,2,\cdots,N$. IR_5, IG_5 and IB_5 are separately the red, green and blue components of the final cipher-image C_1. Similarly, PR_5, PG_5 and PB_5 are separately the red, green and blue components of the final cipher-image C_2.

The proposed image encryption algorithm is a symmetric cryptosystem, so the decryption procedure is similar to that of the encryption process in the reverse order.

4. Experimental results and performance analysis

We do some experiments with double colour images by applying the presented algorithm. The initial conditions and parameters are taken as $r = 3.583980720385628$, $\mu = 0.748590274902131$, $X_0 = 0.289036572890141$, $Y_0 = 0.326432976432178$, $Z_0 = 0.603939084549409$. Fig. 1 shows the encryption and decryption of two color images of Lena and Panda with size 256×256, respectively. We can find that the cipher-images are ambiguous and cannot be recognized correctly. And the decrypted images are identical to the original ones. The results show that the proposed image cryptosystem is feasible and has a satisfactory encryption effect.



(a) (b) (c)

(d) (e) (f)

Figure 1 Experimental results: (a) Plain-image of Lena; (b) Cipher-image of Lena; (c) Decrypted image of Lena; (d) Plain-image of Panda; (e) Cipher-image of Panda; (f) Decrypted image of Panda.

4.1. Key space and sensitivity analysis

In our work, the parameters r, μ and initial conditions X_0, Y_0, Z_0 are regarded as the secret keys. In addition, some other parameters, for example, l and λ, are also considered as the secret keys. Suppose that the computational precision is 10^{-15}. The key space is larger than $10^{78} \approx 2^{259}$. So our proposed algorithm can resist against brute-force attacks.

For convenience, we indicate the original key set as γ. To test the key sensitivity, we slightly change one key, for example, only modify r as $r' = r + 10^{-15}$, and keep other keys unchanged (denote the key set as γ'). Utilizing the key set γ' to encrypt the plain-image in Fig. 1(a) will produce another cipher-image in Fig. 2(a). Corresponding decrypted images using different secret keys are shown in Figs. 2(b) and (c). Fig. 2(c) displays the decrypted images using the wrong keys. As can be clearly seen, even a very slight modification in the secret keys will result in the failure of image decryption. It indicates that the proposed algorithm is high sensitive to the secret keys.

(a) (b) (c)

Figure 2 key sensitive test. (a) Encrypted image of Lena by γ'; (b) Decrypted image of Lena by correct keys; (c) Decrypted image of Lena by wrong keys.

4.2. Information entropy analysis

Information entropy can be calculated by the following formula:

$$H(m)=\sum_{i=0}^{2^n-1}P(m_i)\log_2\frac{1}{P(m_i)}, \tag{25}$$

where 2^n is the total states of the information source m, and $P(m_i)$ denotes the probability of symbol m_i. The theoretical maximum of the information entropy is 8, so the closer the entropy of an encryption algorithm is to 8, the more secure it is. Table 3 provides the entropy values for the plain-images and the cipher-images, respectively. As can be seen clearly, the entropy values of all encrypted images using our proposed algorithm approach closely to the theoretical maximum 8. It implies that the cipher-image is very close to a random source and the probability of information leakage in the encryption process is negligible. Thus the proposed image cryptosystem is secure against the entropy attack.

Table 3 The information entropy of the original and encrypted images.

Encryption algorithms	Test images	Original image			Encrypted images		
		Red	Green	Blue	Red	Green	Blue
The proposed algorithm	Lena	7.2933	7.5812	7.0856	7.9897	7.9888	7.9893
	Panda	7.7118	7.6278	7.7939	7.9901	7.9896	7.9900

4.3. Differential analysis

Generally, an attacker may make a slight change (a single pixel change) in the plain-image and get two cipher-images by using the same encryption algorithm. Then trace the difference between two encrypted images to crack the cryptosystem, called differential attack. Researchers usually use number of pixels change rate (*NPCR*) and unified average changing intensity (*UACI*) as two criterions to examine the performance of resistance against differential attack. *NPCR* and *UACI* are defined as follows:

$$NPCR=\frac{\sum_{i,j}D(i,j)}{M\times N}\times 100\%, \tag{26}$$

$$UACI=\left(\sum_{i,j}\frac{|C'(i,j)-C(i,j)|}{255}\Big/\left(M\times N\right)\right)\times 100\%, \tag{27}$$

$$D(i,j)=\begin{cases}0, & \text{if } C(i,j)=C'(i,j)\\ 1, & \text{otherwise}\end{cases}, \tag{28}$$

where M and N are the width and height of the encrypted image. C and C' are the cipher-images, whose corresponding plain-images have only one pixel difference.

According to Eqs. (26)-(28), we can obtain the average *NPCR* and *UACI* values, i.e., *NPCR*=99.6192% and *UACI*=33.5271%. Obviously, the proposed algorithm is very sensitive to small changes in the plain-text and can well resist the differential attack.

4.4. Statistical analysis

In order to test the correlation between adjacent pixels, we randomly choose 8000 pairs of two neighboring pixels in vertical, horizontal, and diagonal directions from the plain-images and corresponding encrypted images. The correlation coefficients between two adjacent pixels in the plain-images and the cipher-images are shown in Table 4. From Table 4, one can find that the correlation coefficients of the plain-images tend to 1 while those of the encrypted images are nearly zero. So the proposed algorithm greatly weakens correlation between the adjacent pixels in the cipher-images.

Table 4 Correlation coefficients of the plain-images and cipher-images.

Test image	Color component	Correlation direction		
		Horizontal	Vertical	Diagonal
Original Lena image	Red	0.9435	0.9606	0.9383
	Green	0.9365	0.9532	0.9341
	Blue	0.8935	0.9202	0.8889
Encrypted Lena image	Red	-0.0164	0.0094	-0.0062
	Green	-0.0058	0.0214	-0.0039
	Blue	-0.0012	0.0025	-0.0011
Original Panda image	Red	0.9765	0.9744	0.9457
	Green	0.9793	0.9749	0.9480
	Blue	0.9715	0.9750	0.9338
Encrypted Panda image	Red	-0.0007	-0.0055	0.0088
	Green	0.0026	0.0042	0.0084
	Blue	0.0034	-0.0032	-0.0002

5. Conclusions

In this paper, we have designed a double color image cryptosystem based on DNA sequence operation and block permutation. Firstly, both the chaotic systems and the plain-images are employed to generate the key streams for image encryption. Secondly, two plain-images are separated into nonoverlaping blocks, which are permuted using the key streams. Thirdly, the shuffled images are transformed into the DNA matrices by the DNA encoding rules, and the DNA XOR operation is implemented on these DNA matrices. At last, convert the DNA matrices into two intermediate images by the DNA decoding rules, and perform a pixel-level diffusion process on the intermediate images. Simulation results and performance analysis have shown that our proposed encryption algorithm can simultaneously encrypt two plain-images, and has a satisfactory encryption effect and high security.

Acknowledgements

This research was jointly supported by the National Natural Science Foundation of China (Grant Nos 61004006 and 61203094), China Postdoctoral Science Foundation (Grant Nos 2013M530181 and 2015T80396), Program for Science & Technology Innovation Talents in Universities of Henan Province, China (Grant No 14HASTIT042), the Foundation for University Young Key Teacher Program of Henan Province, China (Grant No 2011GGJS-025), Shanghai Postdoctoral Scientific Program (Grant No 13R21410600).

References

[1] Schneier, B. (1995) Cryptography: Theory and Practice. CRC Press, Boca Raton, FL, USA.

[2] Volos, C.K., Kyprianidis, I.M., Stouboulos, I.N. (2013) Image encryption process based on chaotic synchronization phenomena. Signal Processing, 93, 1328-1340.

[3] Kocaerv, L. (2001) Chaos-based cryptography: A brief overview. IEEE Transactions on Circuits and Systems Magzine, 1, 6-21.

[4] Zhou, Y., Bao, L. and Chen, C.L.P. (2014) A new 1D chaotic system for image encryption. Signal Processing, 97, 172-182.

[5] Yaghouti Niyat, A., Hossein Moattar, M. and Niazi Torshiz, M. (2017) Color image encryption based on hybrid hyper-chaotic system and cellular automata. Optics and Lasers in Engineering, 90, 225-237.

[6] Adleman, L.M. (1994) Molecular computation of solutions of combinatiorial problems. Science, 266, 1021-1024.

[7] Liu, H., Wang, X. and Kadir, A. (2012) Image encryption using DNA complementary rule and chaotic maps. Appllied Soft Computing, 12, 1457-1466.

[8] Liu, Y., Tang, J. and Xie, T. (2014) Cryptanalyzing a RGB image encryption algorithm based on DNA encoding and chaos map. Optics and Laser Technology, 60, 111-115.

[9] Zhang, Q., Guo, L. and Wei, X. (2013) A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. Optik, 124, 3596–3600.

[10] Ozkaynak, F. and Yavuz, S. (2014) Analysis and improvement of a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. Nonlinear Dynamics, 78, 1311-1320.

[11] Zhang, Q., Liu, L. and Wei, X. (2014) Improved algorithm for image encryption based on DNA encoding and multi-chaotic maps. International Journal of Electronics and communications (AEÜ), 68, 186-192.

[12] Wang, X., Zhang, H. and Bao, X. (2016) Color image encryption scheme using CML and DNA sequence operations. Biosystems, 144, 18-26.

[13] Enayatifar, R., Abdullah, A.H. and Isnin, I.F. (2014) Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. Optics and Lasers in Engineering, 56, 83-93.

[14] Zhen, P., Zhao, G., Min, L. and Jin, X. (2016) Chaos-based image encryption scheme combining DNA coding and entropy. Multimedia Tools Applications, 75, 6303-6319.

[15] Chai, X., Chen, Y. and Broyde, L. (2017) A novel chaos-based image encryption algorithm using DNA sequence operations. Optics and Lasers in Engineering, 88, 197-213.

[16] Watson, J.D. and Crick, F.H.C. (1953) A structure for deoxyribose nucleic acid. Nature, 171, 737-738.

[17] Zhou, Y., Bao, L. and Chen, C.L.P. (2014) A new 1D chaotic system for image encryption. Signal Processing, 97, 172-182.

[18] Hua, Z., Zhou, Y., Pun, C.M. and Chen, C.L.P. (2015) 2D Sine Logistic modulation map for image encryption. Information Sciences, 297, 80-94.