Technology and Health Care 28 (2020) S303–S309
DOI 10.3233/THC-209030
IOS Press

Study on an efficient hyper-chaos-based image encryption scheme using global bit permutation

Jianfeng Zhang, Zhiying Lu* and Min Li
School of Electrical and Information Engineering, Tianjin University, Tianjin, China

Abstract.
BACKGROUND: The chaotic system with low dimensions has a low security compared to the high-dimensional chaotic system. Furthermore, major pixel-level permutations merely transform the pixel position and cannot change the intensity distribution of the original image. Bit-level permutation could change the intensity distribution, as it devotes more time to conduct bit-level computation.
OBJECTIVE: In this study, we present a more efficient image encryption approach based on hyper-chaos and a global bit cycle shift (HC-GBCS).
METHODS: According to the input image we adopted the SHA-256 secure hash algorithm to obtain the initial key, which served as the premier parameter of the chaotic system. Then we employed a 4D hyper-chaotic system for generating the chaotic series, on which we utilized global bit permutation to enhance the security of the encryption system. Finally, the diffusion process was conducted by using the generated chaotic series extended with a logistic map.
RESULTS: Experimental results and analysis reveal that the presented approach encrypts plain images effectively and achieves high security and stability.
CONCLUSION: The proposed method can deal with the problems inherently existing in encryption methods utilizing low-dimensional chaotic map. Furthermore, global bit permutation can transform the pixel distribution of plain images and enhance the cryptosystem security.

Keywords: Hyper-chaos, global bit cycle shift, SHA-256, encryption

1. Introduction

With the increasing popularity of network and information technology, the information transfer via Internet is increasingly common. As a greatly efficient approach of privacy and security protection, image encryption has drawn considerable attention and plenty of algorithms utilizing various techniques have been explored in the past decades. Due to the fact that images are inherently bulk data with a high correlation among pixels and significant data redundancy, existing encryption methods cannot figure out real-time image encryption tasks well. Thus, amounts of chaos-based encryption methods achieved inspiring results in amounts of real tasks [1].

The fundamental structure of chaos-based approaches consists of two procedures: permutation and diffusion phases. In recent years, pixel-level based permutation methods have achieved significant results

*Corresponding author: Zhiying Lu, School of Electrical and Information Engineering, Tianjin University, Tianjin, China. E-mail: luzy@tju.edu.cn.

in many image encryption tasks [2–4]. Liu et al. [5] utilized one-time keys and two reliable chaotic maps, and combined MD5 algorithm with existing cycle encryption. Wang et al. [6] proposed an encryption method which associated a 4D Lorenz hyper-chaotic system with gene recombination.

To deal with the drawback inherently existing in approaches utilizing pixel-level permutation, various approaches utilizing bit-level permutation have been proposed. Zhu et al. [7] developed an algorithm that uses a logistic map and an Arnold map separately for diffusion and permutation. Li et al. [8] reported a five-dimensional multi-wing hyper-chaotic system and the permutation process divided the image into blocks, and each block is multiplied by a constant matrix. Moreover, DNA sequence has been used for image encryption [9,10]. Unfortunately, this kind of algorithm has the same weakness as the ones using pixel-level permutation.

In order to solve the above-described problems, this study presents an efficient real-time encryption algorithm on the basis of hyper-chaos and a global bit cycle shift (HC-GBCS). Theoretical and numerical analyses indicate that the proposed method resists different attacks and owns low time complexity. In addition, the encryption test on finger-vein images shows that the proposed method has a significant performance in practical application.

The remainder of our study is summarized as follows. The basic theory of the 4D hyper-chaotic system is introduced in Section 2. In Section 3, the proposed approach is introduced in detail. Experimental results and performance analysis are presented in Section 4, and conclusions are drawn in Section 5.

2. Hyper-chaotic system

Although traditional low-dimensional chaotic systems are highly efficient, they suffer from inadequate key spaces and low security. On the contrary, high-dimensional hyper-chaotic systems own more positive Lyapunov exponents, more complicated and unpredictable dynamic characteristics, and higher randomness [11]. The 4D hyper-chaotic system adopted in our study can be given as follows [12]:

$$\begin{cases} \dot{x}_1 = ax_1 - x_2x_3 \\ \dot{x}_2 = x_1x_3 - bx_2 \\ \dot{x}_3 = cx_1x_2 - dx_3 + gx_1x_4 \\ \dot{x}_4 = kx_4 - hx_2 \end{cases} \tag{1}$$

$$(54x^2 + 12)\ln(3x^x + 2) + 36x^2 - \cos^2 x - (\cos^2 x - \sin^2 x)\ln(\sin x) \tag{2}$$

where $x_1 - x_4$ are state variables and $a - k$ are constant values. When $a = 8, b = 40, c = 2, d = 14, g = 5, h = 0.2$ and $k = 0.05$, the above system is hyper-chaotic.

3. Proposed image encryption system

This paper presents a novel real-time image encryption algorithm based on hyper-chaos and a global bit cycle shift (HC-GBCS). First, the plain image is used to generate a 256-bit external secret key K through the SHA-256 secure hash algorithm. Then a 4D hyper-chaos system which is initialized by using secret key K is introduced to generate the chaotic sequence. Then the image is shuffled by a pixel-level permutation and decomposed into eight binary bitplane images. Finally, global bit permutation and diffusion operations are conducted to change the image intensity distribution and further enhance the security.

3.1. Generation of the secret key and chaotic sequence

Step 1. Let $M \times N$ denote the size of input image. SHA [13] is used for the input image to produce the external key sequence K. This is divided into 32 groups, that is, $K = k_1, k_2, \ldots, k_{32}$.

Step 2. The parameters of the chaotic system (1) are computed by Eq. (3):

$$\begin{aligned} x_1 &= \mathrm{mod}((k_1 \oplus k_5 \oplus ... \oplus k_{29})/256 + a_1, 1) \\ y_1 &= \mathrm{mod}((k_2 \oplus k_6 \oplus ... \oplus k_{30})/256 + a_2, 1) \\ z_1 &= \mathrm{mod}((k_3 \oplus k_7 \oplus ... \oplus k_{31})/256 + a_3, 1) \\ w_1 &= \mathrm{mod}((k_4 \oplus k_8 \oplus ... \oplus k_{32})/256 + a_4, 1) \end{aligned} \tag{3}$$

where $a_1 - a_4$ are the given keys.

Step 3. The chaotic system (1) is solved numerically using the fourth-order Runge-Kutta method, and the former L_1 values are discarded to eschew side effects. Each chaotic sequence consists of $L = \max\{M, N\}$ elements. The four chaotic series are denoted as X, Y, Z, W.

3.2. Image encryption

3.2.1. Pixel-level permutation process

Step 1. Series X and Y are utilized to obtain new sequences $X1$ and $Y1$ according to:

$$\begin{aligned} X1 &= \mathit{abs}(X + Y * 2) - \mathit{floor}(\mathit{abs}(X + Y * 2)) \\ Y1 &= \mathit{abs}(X * 2 + Y) - \mathit{floor}(\mathit{abs}(X * 2 + Y)) \end{aligned} \tag{4}$$

Step 2. Chaotic series $X1$ and $Y1$ are arranged in ascending order to generate two new sequences. Both rows and columns are scrambled based on the respective index values. The scrambled image is denoted as K_1.

After the permutation process, the correlations among neighboring pixels of the original image have been broken but the image histogram has not changed, so the scrambled image requires further processing.

3.2.2. Global bit cycle shift (GBCS)

The pixel value of a grayscale image is an integer ranged from 0 to 255, which could be converted into a binary series with eight bits, and the ith bit in the series per pixel value is extracted to obtain the ith bitplane image. The GBCS process can be described as the joint processing of the eight bitplane images. The procedure changes both the pixel position and the pixel value.

Step 1. The image K_1 is decomposed into eight bitplanes, denoted as $\mathit{bitimg}\{i\}, i = 1, 2 \ldots, 8$.

Step 2. Bitplane image cycle shift.

The lower-order bits are shifted using the values of the higher-order bits. We select four numbers from the chaotic sequence Y, taking the last number in the sequence and those at intervals of 50 numbers prior to the last value in the sequence. The four numbers are sorted in ascending order, and the index values are employed to confuse the lower-order bits according to:

$$\mathit{bitimg}\{i\} = \mathit{cirshift}(\mathit{bitimg}\{i\}, \mathit{sum}(\mathit{bitimg}\{I(i) + 4\})), i = 1, 2, 3, 4 \tag{5}$$

where $\mathit{cirshift}\{a, b\}$ represents the rotation of sequence a to the right by b bits and $I(i)$ denotes the index of $\mathit{bitimg}\{i\}$.

Step 3. The scrambled bitplanes are merged and the acquired image is denoted as K_2.

After the GBCS process, the image histogram has changed, but the distribution is not sufficiently

uniform. This means that the statistical properties of the original image have not completely changed, so a further diffusion process is necessary.

3.2.3. Diffusion process

The operation of diffusion is an extremely crucial procedure which transforms the pixel distribution and arouses an avalanche effect in case a pixel value is changed.

Step 1. Reshape the image matrix K_2 to a one-dimensional column vector Q.

Step 2. The first round of diffusion is as follows.

Step 2.1 Diffuse the first pixel value $Q(1)$ according to:

$$P(1) = Q(1) \oplus Z2(\textit{end}) \oplus Z2(1) \tag{6}$$

Step 2.2 Set $k = 2$ and process the kth pixel value according to:

$$P(k) = Q(k) \oplus Z2(k) \oplus P(k-1) \oplus Z2(k-1) \tag{7}$$

Step 2.3 Set $k = k + 1$ and repeat Step 2.2 until $k = M \times N$.

Step 2.4 Process the first pixel again as $P(1) = P(1) \oplus P(\textit{end})$.

Step 3. For the second round of diffusion, we use the sequence $W2$ and perform the same diffusion operation as in Step 2 on the vector P to obtain the column vector R.

Step 4. The encrypted image is acquired by transforming R into an image with size $M \times N$.

The decryption algorithm is the reverse procedure of the above steps.

4. Experimental results and security analyses

Various experiments are conducted to analyze and verify the effectiveness of the presented method. In the experiments, except the natural images with size 256×256, finger-vein images are extra led in to demonstrate that our proposed algorithm has unique advantages in the security field of the real world. In addition, Wangs algorithm (GR-HC) [6], Zhus algorithm (BLP) [7], Lis algorithm (PL-BL) [8], Tengs algorithm (CML) [14], Chens algorithm (LUT) [15] and another Chens algorithm (DSVSM) [16] are compared with our proposed algorithm.

4.1. Key space

To ensure that the cipher text cannot be revealed by a brute-force attack, the encryption approach should own a large key space. In our presented HC-GBCS method, the secret keys consist of two parts: the original key K produced using SHA-256 hash algorithm, and five given keys $a_1 - a_5$ introduced in the hyper-chaotic system and logistic map. Since the parameter K is a binary number with a length of 32 bytes and the precision of an eight-bit binary number is 2^8, the precision of K is 2^{256} and the five given keys are $H_{a1} = H_{a2} = H_{a3} = H_{a4} = H_{a5} = 10^{16}$. Thus, the whole key space H is calculated as:

$$H = KH_{a1}H_{a2}H_{a3}H_{a4}H_{a5} = 2^{256} \times 10^{80} \approx 2^{521} \tag{8}$$

To ensure the reliability of an encryption system, the key space should be larger than 2^{100} to resist exhaustive attacks [16]. Thus, we can see that the proposed HC-GBCS method owns a sufficient key space to withstand various kinds of exhaustive attacks.

4.2. Key sensitivity

An efficient encryption method should be sensitive to the secret key for resisting different attacks. The

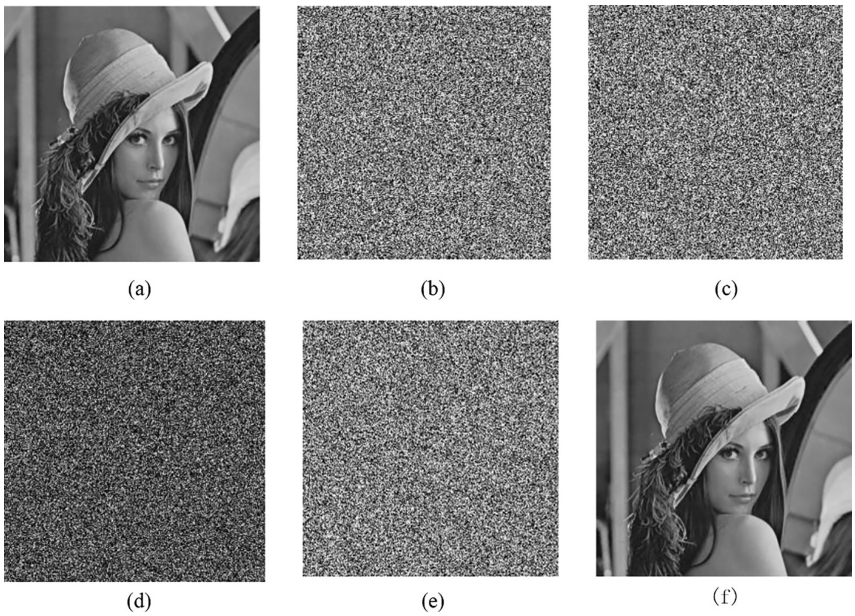

Fig. 1. Analysis of key sensitivity: (a) Lena image, (b) Encrypted image (a1 = 0.2), (c) Encrypted image (a1 = 0.200000000000001), (d) Difference between (b) and (c), (e) Decryption of image (b) with a1 = 0.200000000000001, (f) Decryption of image (b) with a1 = 0.2.

plain image could not be recovered even if the key has a slight difference. The Lena image with size 256×256 is utilized to verify the sensitivity of the presented encryption method. From Fig. 1b and d, we can see that the cipher-image is totally different when the key a_1 is increased by only 10^{-16}. Furthermore, as shown in Fig. 1e and f, the cipher-image is decrypted incorrectly when it adopts a secret key which is subtly different from the initial key. We can conclude that even a subtle difference in the decryption key will cause the encrypted image to be unrecoverable. Hence, the presented encryption method is greatly sensitive to the key.

4.3. Correlation analysis

The correlation among neighboring pixels refers to similarity degree between nearby pixels. Generally, correlation among neighboring pixels along the horizontal, vertical and diagonal orientations is always high for most natural images. Thus, an excellent encryption system should significantly reduce the correlation among neighboring pixels in the cipher-image.

We randomly selected 5000 pairs of neighboring pixels along three orientations from 40 natural images and finger-vein images. We then calculated the correlation coefficients of neighboring pixels using the following formula:

$$\rho_{xy} = \frac{cov(x, y)}{\sqrt{D(x)D(y)}} \tag{9}$$

where x, y are pixel values of two neighboring pixels in an image, $D(x)$, $D(y)$, $cov(x, y)$ denote the corresponding variance and covariance, respectively.

The mean values of correlation coefficients with different algorithms are listed in Table 1, which indicates that the correlation coefficients of plain images tends to 1, whereas those of cipher-images tends

Table 1
Comparison results of correlation coefficients

		Natural images			Finger-vein images		
	Methods	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Plain-image		0.9012	0.7603	0.9302	0.9992	0.9960	0.9945
	HC-GBCS	**−0.0037**	**0.0081**	**−0.0020**	**−0.0062**	**0.0057**	**0.0036**
	GR-HC	−0.0169	0.0137	0.0023	−0.0015	0.0051	0.0174
Cipher-image	LBP	0.0186	−0.0145	0.0125	−0.0018	−0.0271	−0.0176
	PL-BL	0.0094	−0.0155	−0.0053	−0.0078	0.0095	0.0210
	CML	0.0081	0.0340	0.0104	0.0035	0.0068	0.0203
	LUT	−0.0833	−0.0103	0.0108	0.2212	0.0048	−0.0061
	DSVSM	−0.0020	−0.0117	0.0178	−0.0119	0.0037	−0.0107

Table 2
Packet loss and packet error comparison results

	Packet loss			Salt-and-pepper noise			Gaussian noise		
	1/16	1/8	1/4	0.01	0.03	0.05	0.0001	0.0003	0.0005
HC-GBCS	**0.5727**	**0.4381**	**0.2798**	**0.7191**	**0.5362**	**0.4316**	**0.3206**	**0.2365**	**0.2010**
GR-HC	0.4885	0.3346	0.2078	0.7449	0.5066	0.3909	0.1080	0.1083	0.1084
BLP	0.7481	0.6406	0.5184	0.8655	0.6850	0.5688	0.4021	0.3098	0.2711
PL-BL	0.5966	0.4086	0.2634	0.7998	0.5699	0.4478	0.2189	0.1571	0.1320
CML	0.1948	0.1808	0.1537	0.1774	0.1335	0.1070	0.1433	0.1227	0.1098
LUT	0.4823	0.3393	0.2199	0.8017	0.5723	0.4470	0.1125	0.1121	0.1126
DSVSM	0.5555	0.3886	0.2412	0.8004	0.5705	0.4505	0.3209	0.2366	0.2016

to 0 in three directions. Therefore, it demonstrates that the presented encryption method has excellent confusion and diffusion traits.

4.4. Packet loss and packet error analysis

We simulated the corruption of information that occurs during actual network transmission by adding salt-and-pepper noise with different densities (0.01, 0.03, and 0.05, respectively), and Gaussian noise with mean 0 and different variances (0.0001, 0.0003, and 0.0005, respectively) to 40 encrypted images, and then decrypted them.

Structural Similarity (SSIM) is an indicator to measure the similarity of two images [17], the average values of SSIM values are shown in Table 2. The proposed algorithm outperforms other algorithms except BLP, and obviously, it is robust to the packet loss and pack error.

5. Conclusions

This study presented a hyper-chaos-based image encryption scheme utilizing global bit cycle shift. The algorithm utilizes a 4D hyper-chaotic system whose preliminary parameters are produced by SHA-256 secure hash algorithm, and adopts global bit permutation to enhance the security of the encryption system.

The system can deal with the problems inherently existing in encryption methods utilizing low-dimensional chaotic map. Furthermore, global bit permutation can transform the pixel distribution of plain images and enhance the cryptosystem security. The amount of experiments and corresponding analysis, including key space analysis, key sensitivity analysis, correlation analysis, and packet loss and packet error analysis, validate that the presented approach is secure and reliable for image encryption.

In addition, the encryption test on finger-vein images show that the proposed method has a significant performance in practical application.

Conflict of interest

None to report.

References

[1] Abanda Y, Tiedeu A. Image encryption by chaos mixing. Image Processing Let. 2016; 10(10): 742-750.
[2] Wang X, Guo K. A new image alternate encryption algorithm based on chaotic map. Nonlinear Dynamics. 2014; 76(4): 1943-1950.
[3] Zhang X, Zhao Z. Chaos-based image encryption with total shuffling and bidirectional diffusion. Nonlinear Dynamics. 2014; 75(1-2): 319-330.
[4] Hua Z, Zhou Y, Pun CM, et al. 2D sine logistic modulation map for image encryption. Information Sciences. 2015; 297: 80-94.
[5] Liu H, Wang X. Color image encryption based on one-time keys and robust chaotic maps. Computers & Mathematics with Applications. 2010; 59(10): 3320-3327.
[6] Wang X, Zhang HL. A novel image encryption algorithm based on genetic recombination and hyper-chaotic systems. Nonlinear Dynamics. 2015; 83(1-2): 333-346.
[7] Zhu ZL, Zhang W, Wong KW, et al. A chaos-based symmetric image encryption scheme using a bit-level permutation. Information Sciences. 2011; 181(6): 1171-1186.
[8] Li Y, Wang C, Chen H. A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. Optics & Lasers in Engineering. 2017; 90: 238-246.
[9] Liu H, Wang X, Kadir A. Image encryption using DNA complementary rule and chaotic maps. Applied Soft Computing. 2012; 12(5): 1457-1466.
[10] Huang X, Ye G. An image encryption algorithm based on hyper-chaos and DNA sequence. Multimedia Tools & Applications. 2014; 72(1): 57-70.
[11] Salman SM, Elsadany AA. On the bifurcation of Marotto's map and its application in image encryption. Journal of Computational & Applied Mathematics. 2018; 328: 177-196.
[12] Dadras S, Momeni HR. Four-scroll hyperchaos and four-scroll chaos evolved from a novel 4D nonlinear smooth autonomous system. Physics Letters A. 2010; 374(11-12): 1368-1373.
[13] Guesmi R, Farah MAB, Kachouri A, et al. A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2. Nonlinear Dynamics. 2016; 83(3): 1123-1136.
[14] Teng L, Wang X. A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive. Optics Communications. 2012; 285(20): 4048-4054.
[15] Chen JX, Zhu ZL, Fu C, et al. An efficient image encryption scheme using lookup table-based confusion and diffusion. Nonlinear Dynamics. 2015; 81(3): 1151-1166.
[16] Chen JX, Zhu ZL, Fu C, et al. A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism. Communications in Nonlinear Science & Numerical Simulation. 2015; 20(3): 846-860.
[17] Oh J, Hwang H. Feature enhancement of medical images using morphology-based homomorphic filter and differential evolution algorithm. International Journal of Control Automation & Systems. 2010; 8(4): 857-861.