Multimedia Tools and Applications (2021) 80:27445–27469
https://doi.org/10.1007/s11042-021-10979-w

A novel image encryption algorithm based on least squares generative adversarial network random number generator

Zhenlong Man[1,2] · Jinqing Li[1,2] · Xiaoqiang Di[1,2,3] · Xu Liu[1,2] · Jian Zhou[1,2] · Jia Wang[1,2] · Xingxu Zhang[1,2]

Received: 31 August 2020 / Revised: 4 January 2021 / Accepted: 30 April 2021 /
Published online: 20 May 2021

Abstract

In cryptosystems, the generation of random keys is crucial. The random number generator is required to have a sufficiently fast generation speed to ensure the size of the keyspace. At the same time, the randomness of the key is an important indicator to ensure the security of the encryption system. The chaotic random number generator has been widely used in cryptosystems due to the uncertainty, non-repeatability, and unpredictability of chaotic systems. However, chaotic systems, especially high-dimensional chaotic systems, have slow calculation speed and long iteration time. This caused a conflict between the number of random keys and the speed of generation. In this paper, we introduce the Least Squares Generative Adversarial Networks(LSGAN)into random number generation. Using LSGAN's powerful learning ability, a novel learning random number generator is constructed. Six chaotic systems with different structures and different dimensions are used as training sets to realize the rapid and efficient generation of random numbers. Experimental results prove that the encryption key generated by this scheme can pass all randomness tests of the National Institute of Standards and Technology (NIST). Hence, our result shows that LSGAN has the potential to improve the quality of the random number generators. Finally, the results are successfully applied to the image encryption scheme based on selective scrambling and overlay diffusion, and good results are achieved.

Keywords Random number generator · Least squares generative adversarial networks · NIST test · Chaotic system · Image encryption

✉ Jinqing Li
lijinqing@cust.edu.cn

✉ Xiaoqiang Di
dixiaoqiang@cust.edu.cn

[1] School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China

[2] Jilin Province Key Laboratory of Network and Information Security, Jilin, China

[3] Information Center of Changchun University of Science and Technology, Changchun, China

1 Introduction

With the rapid development of network communication and multimedia technology, more and more digital images are stored, copied, and transmitted through various types of third-party platforms or unsecured channels [5]. Therefore, image security has become a hot topic [44]. There are a variety of means to protect image security (such as steganography [1, 2], watermark [4, 43], encryption [11, 12, 45, 54, 55]. etc.), among which image encryption is a more common and effective tool. Image encryption is usually divided into two stages: the scrambling stage and the diffusion stage. The scrambling stage is to change the relative position of pixels in the image, while the diffusion phase is to change the specific value of pixels in the image. Both scrambling and diffusion need to be controlled by the key. Considering the importance of an encryption key, it is a difficult task to generate encryption key with good randomness [44].

In recent years, many schemes of generating random numbers based on chaos have been proposed, but they have little help to improve the security of encryption algorithms [20, 46, 52, 53, 59]. Chai x et al. proposed an image encryption algorithm based on DNA sequence operation and chaotic system. This paper mainly uses a two-dimensional logistic adjusted sine map (2D-LASM) to generate encryption key [6]. Hua Z et al. proposed a cosine-transform-based chaotic system (CTBCS). Two chaotic maps were used as seed maps to make the CTBCS have more complex dynamic behaviors and then generate encryption keys with better randomness [18]. Gong l et al. proposed an effective image compression and encryption algorithm based on chaotic systems and compressed sensing. In this scheme, the logistic map and 1D cascade map are used as key generators [14]. The commonness of these algorithms is that they all use low dimensional chaotic systems as key generators, which may lead to the possibility that the generated keys are not random enough, so that attackers can take advantage of them. In contrast, the key generated by our scheme can make the encryption key completely random.

In fact, many encryption schemes based on chaos have disadvantages [9, 14, 16, 22, 23, 42]. Short cycle length is one of the important problems of chaotic keystream generators, which results from the finite precision of computers [24, 35]. To solve the problem of randomness and insufficient security of keystream generated by the chaotic system, many scholars have developed several effective encryption algorithms by combining chaotic systems with other methods [10, 25, 37, 47, 57], such as breadth-first search [57], DAN coding [25, 47], edge password [37], elliptic curve [10] and so on.

After an in-depth study, we have found that the image encryption algorithm also has a promoted requirement of random numbers. According to the definition in [3, 36], three attributes can be considered to evaluate the randomness of the sequence [26]:

1. It appears to be random, which means that it can pass all statistical tests of randomness.
2. Unpredictability, even if the corresponding algorithm and hardware are given, it is still unable to predict the next random bit by calculation.
3. It is impossible to replicate reliably. Even if the generator runs twice with exactly the same input, two completely unrelated random sequences can be obtained.

For the first time, we combine LSGAN with the chaotic system as a random number generator. The random numbers generated by this scheme satisfy the above three conditions at the same time. Therefore, the encryption key we use is completely random.

The basic idea of GAN comes from a two person zero-sum game in game theory. It consists of a generator and a discriminator and is trained through confrontation learning.

Its purpose is to estimate the potential distribution of data samples and generate new data samples [30]. Since its first launch in 2014 [27], GAN has become a hot topic in the field of computer vision [48], natural language processing [28], malicious attack detection [8], and data generation [58]. We are very happy to see it introduced into the field of security [19, 21, 32, 60]. Because of the randomness and difference of GAN training results, we take the chaotic sequence as real data, so that the generator can produce data samples similar to and different from the chaotic sequence distribution to meet our need. Then, through NIST, histogram, and entropy analysis, we further prove the effectiveness of the random number generator. Finally, the generated random number is mapped to the image encryption key. Through the analysis of the ciphertext image, we can find that security has been significantly improved.

For an excellent image encryption algorithm, security and efficiency are equally important. Many existing encryption schemes are difficult to achieve a good balance between security performance and encryption efficiency [13, 29, 50, 54]. The main factor affecting the efficiency is that there will be some repeated pixel displacements during scrambling (for example, two identical pixel values are exchanged), which increases the time cost [31]. To improve the security of the algorithm, researchers have proposed a dynamic diffusion algorithm [53] and adaptive diffusion algorithm [7, 17], but they all traverse all pixels in turn to change the size of their values, so the encryption efficiency is not ideal. Therefore, we draw lessons from this mode and design a selective scrambling method and a superposition diffusion method.

To solve the above problems, we design a new random number generator to solve the problem of insufficient security and randomness of the encryption key. Firstly, we take the chaotic sequences generated by six chaotic systems as the training set of GAN in turn. By adjusting the parameters and training time of the GAN system, we finally generate random numbers that meet our needs. Secondly, to improve the scrambling effect, a selective scrambling method is designed, which scrambles the pixel regions with important information and the edge pixels in the image. Finally, a superposition diffusion method is proposed. Compared with the traditional dynamic diffusion method, each pixel block is used as the diffusion unit in the coverage diffusion method, while each pixel is used as the diffusion unit in the traditional method, so the coverage diffusion can improve the efficiency of the diffusion operation. Simulation results and security analysis show that the random number generated by this scheme can pass all the tests in NIST, and the efficiency of key generation is improved. The encryption system can effectively resist the common violent attacks, differential attacks, and shear attacks. In the performance test of pixel correlation and information entropy, compared with other literature, each index has certain advantages.

The rest of this paper is structured as follows. Section 2 briefly introduces the related work and the method of generating the security key. Section 3 introduces the randomness and security analysis of generating random numbers in detail. Section 4 introduces the encryption algorithm. Section 5 analyses the performance of the proposed algorithm through various security tests. Section 6 winds up the paper.

2 Random numbers generator with LSGAN

We propose a random number generator that combines LSGAN and chaotic systems as shown in Fig. 1. LSGAN is closer to real data than Generative Adversarial Networks(GAN) (The random number generated by LSGAN is used as the encryption key in the encryption algorithm). We choose six chaotic systems, including low dimensions and high dimensions. The chaotic sequence generated by a chaotic system is used as the training set. We try to make the data generated by the generator infinitely close to the chaotic sequence, and then test the randomness of the generated data. Through performance analysis, we can prove that our idea is correct. At the same time, we can test the randomness of the generated data. When a large number of random numbers are needed, the efficiency of LSGAN is much better than that of a chaotic system.

2.1 LSGAN

GAN is a deep learning model and is one of the most promising methods for unsupervised learning on complex distributions in recent years. The model produces good output through the mutual game learning of two modules in the framework: the generative model and the discriminant model.

$$\begin{aligned}&\min_G \max_D V(G, D) = \\ &E_{x\sim Pdata^{(x)}}[\log D(x)] + E_{Z\sim P_{Z(Z)}}[\log(1 - D(G(Z)))]\end{aligned} \tag{1}$$

In the formula, $E_{x\sim Pdata}$ is the distribution of the real input data, log $D(x)$ is the judgment value of the discriminator, log $[(1 - D(z))]$ is the judgment value of the generated data. Through the continuous game of the maximum and minimum values, the G-Network model and D-Network model are circularly and alternately optimized until the two models reach the Nash equilibrium; x is the training set, $G(z)$ is the data generated by the G-Network model, and $D(x)$ is the D-Network model for determining whether the real data and the training z is the noise of input G network model,$E_{Z\sim P_{Z(Z)}}$ is the distribution of noise data.

Conventional GAN uses S-type cross entropy loss function. As described in reference 1, when updating the generator, this loss function will cause the problem of vanishing gradients for the samples that are on the correct side of the decision boundary, but are still far from the real data, Therefore, the least square method (LSGAN) is proposed. Suppose A-B coding

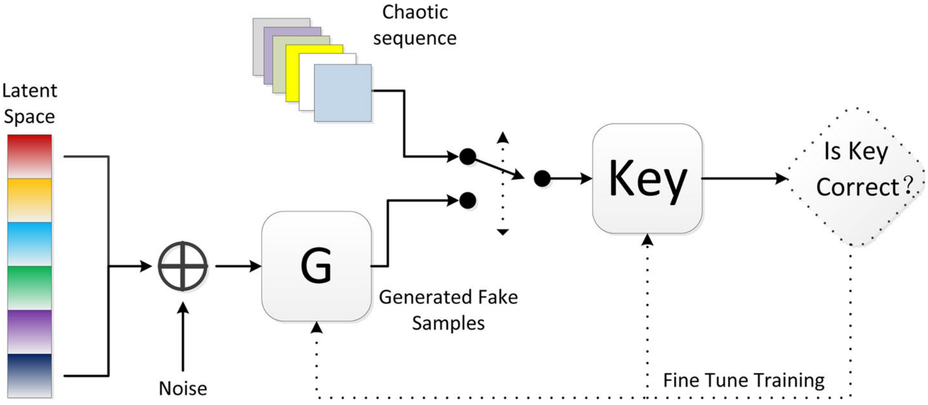

Fig. 1 Block diagram of random number generation scheme. The latent space shows where the noise is located in the potential space. Noise is the random number generated by the systems, the chaotic sequence is obtained by the iterative chaotic system

scheme is used for the identifier, where A and B are labels of pseudo data and real data, respectively. Then, the objective function of LSGAN can be defined as following:

$$\begin{cases} \min_D V_{LSGAN}(D) = \\ \frac{1}{2}E_{x\sim Pdata^{(x)}}[(D(x)-b)^2] + \frac{1}{2}E_{Z\sim P_{Z(Z)}}[D(G(Z))-a)^2] \\ \min_D V_{LSGAN}(D) = frac12E_{Z\sim P_{Z(Z)}}[D(G(Z))-c)^2] \end{cases} \tag{2}$$

where c denotes the value that G wants D to believe for fake data. The benefits of LSGAN can be drawn from two aspects [38]. First of all, unlike conventional GAN, LSGAN does minor damage to long samples located on the right side of the decision boundary. Even if they are correctly classified, LSGAN will punish those samples. When the generator is updated, the discriminator's parameters are fixed, that is, the decision boundary is set. As a result, the penalty causes the generator to generate samples toward the decision boundary. On the other hand, the decision boundary should span all aspects of the real data for successful GAN learning. Otherwise, the learning process will be saturated. Therefore, moving the generated samples to the decision boundary will make them closer to the real data. Second, penalizing the samples that are far from the decision boundary can generate more gradients when up-dating the generator, which in turn reduces the problem of vanishing gradients. This allows LSGAN to perform more stable during the learning process.

2.2 Data set preparation

Six chaotic systems are selected, and the generated pseudo-random number is used as the training set of GAN.

(1) QCNN [39]:

$$\begin{cases} \dot{g}_1 = -2a_1\sqrt{1-g_1{}^2}\sin h_1 \\ \dot{h}_1 = (-b_1(g_1-g_2)+2a_1\cos h_1)/(\sqrt{1-g_1{}^2}) \\ \dot{g}_2 = -2a_2\sqrt{1-g_2{}^2}\sin h_2 \\ \dot{h}_2 = (-b_2(g_2-g_1)+2a_2g_2\cos h_2)/(\sqrt{1-g_2{}^2}) \end{cases} \tag{3}$$

Where: g_1, g_2 is the polarizability; h_1, h_2 is the quantum phase; a_1, a_2 is the proportional coefficient of the energy between the points in each cell; b_1, b_2 is the weighted influence factor for the difference between the polarizability of adjacent cells. When $a_1 = a_2 = 0.28$, $b_1 = 0.7$, and $b_2 = 0.3$, the system is in a chaotic state.

(2) 4D hyperchaotic system [16]:

$$\begin{cases} \dot{x}_1 = \delta_1(x_2 - x_1) \\ \dot{x}_2 = \delta_2 x_1 + \delta_3 x_2 - x_1x_3 + x_4 \\ \dot{x}_3 = x_2^2 - \delta_4 x_3 \\ \dot{x}_4 = -\delta_5 x_1 \end{cases} \tag{4}$$

Let $x = [x_1; x_2; x_3; x_4]$ represent the state vector of the system. When $\delta_1 = 27.5, \delta_2 = 3, \delta_3 = 19.3, \delta_4 = 2.9, \delta_5 = 3$, the system is hyperchaotic. Lyapunov exponents are $\lambda 1 = 1.6170, \lambda_2 = 0.1123, \lambda_3 = 0, \lambda_4 = -12.8245$. The system has two positive Lyapunov exponents, indicating that it is a hyperchaotic system at this time.

(3) Fractional Chen hyperchaotic system [33]:

$$\begin{cases} \frac{d^{\alpha}}{dt^{\alpha}} y_1 = \omega_1(y_2 - y_1) + y_4 \\ \frac{d^{\alpha}}{dt^{\alpha}} y_2 = \omega_2 y_1 - y_1 y_3 + \omega_3 y_2 \\ \frac{d^{\alpha}}{dt^{\alpha}} y_3 = y_1 y_2 - \omega_3 y_4 \\ \frac{d^{\alpha}}{dt^{\alpha}} y_4 = y_2 y_3 + \omega_5 y_4 \end{cases} \tag{5}$$

Where $\omega_1, \omega_2, \omega_3, \omega_4,$ and $\omega_5$are positive parameters. When $\omega_1 = 35, \omega_2 = 3, \omega_3 = 12, \omega_4 = 7$ and $\omega_5 = 0.58$, two positive Lyapunov exponents of hyperchaotic systems are given by $\lambda_1 = 0.2104$ and $\lambda_2 = 0.126$. The prediction time of hyperchaotic systems is usually shorter than that of chaotic systems, so the security of hyperchaotic systems is higher.

(4) 3D hyperchaotic system [34]:

$$\begin{cases} \dot{z}_1 = -\beta_1 z_1 + z_2 z_3 \\ \dot{z}_2 = \beta_2 z_2 - z_1 z_3 - z_3 \\ \dot{z}_3 = -\beta_3 z_3 + z_2^3 \end{cases} \tag{6}$$

Parameters β_1, β_2 and β_3 are real constants, when $\beta_1 = 3$, $\beta_2 = 5$ and $\beta_3 = 10$, the system is chaotic. Lyapunov exponents are $\lambda_1 = 0.03, \lambda_2 = -0.01$ and $\lambda_3 = -7.78$. There are positive exponents in Lyapunov exponent, so the system has chaotic characteristics.

(5) Lorenz chaotic system [31]:

$$\begin{cases} \frac{dq}{dt} = -\varphi_1(q - w) \\ \frac{dw}{dt} = -qr + \varphi_2 q - w \\ \frac{dr}{dt} = -qw - \varphi_3 r \end{cases} \tag{7}$$

where φ_1, φ_2 and φ_3 are the parameters of the system. When $\varphi_1 = 10$ and $\varphi_3 = 8/3$, the system turns into the chaotic state if $\varphi_2 > 24.74$. When $\varphi_2 = 28$, the system turns into the best chaotic state.

(6) Henon chaotic system [51]:

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n \end{cases} \tag{8}$$

where,$n = 2, 3, , w, a > 0, b > 0$, and $x, y \in R^w$; w refers to the dimension, and a and b refer to the control parameters. When $w = 2$, the system becomes the famous Henon map.

2.3 Generation of random numbers

Firstly, the six chaotic systems are used to generate six sets of 400 × 100 normalized chaotic sequences $HD1$, $HD2$, $HD3$,

$HD4$, $HD5$ and $HD6$, which are the training set of the LSGANs.

$$\begin{cases} HD1 = mod(HD1, 1) \\ HD2 = mod(HD2, 1) \\ HD3 = mod(HD3, 1) \\ HD4 = mod(HD4, 1) \\ HD5 = mod(HD5, 1) \\ HD6 = mod(HD6, 1) \end{cases} \tag{9}$$

Secondly, LSGAN is iterated t times to generate 6 groups of random numbers $G1$, $G2$, $G3$, $G4$, $G5$, $G6$ with a size of 800×100. The encryption keys K_{G1}, K_{G2}, K_{G3}, K_{G4}, K_{G5} and K_{G6} are obtained after conversion.

$$\begin{cases} K_{G1} = mod(G1 \times 10^5, 256) \\ K_{G2} = mod(G2 \times 10^5, 256) \\ K_{G3} = mod(G3 \times 10^5, 256) \\ K_{G4} = mod(G4 \times 10^5, 256) \\ K_{G5} = mod(G5 \times 10^5, 256) \\ K_{G6} = mod(G6 \times 10^5, 256) \end{cases} \tag{10}$$

3 Security analysis of random number generation scheme

3.1 Entropy analysis

It can be seen from Fig. 2a that the random numbers are evenly distributed and have good randomness. After the random number is converted into a key, it can be seen in Fig. 2b that the data distribution is relatively flat, and the entropy of the image is directly related to the histogram of the image, so the smoother histogram has a higher entropy. The entropy H (m) can be calculated as follows:

$$H(s) = -\sum_{i=0}^{2^N-1} R(s_i) \log_2 R(s_i) \tag{11}$$

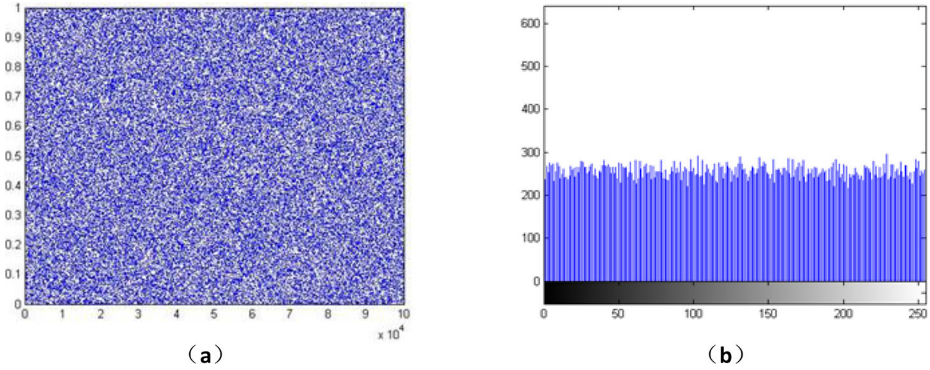

Fig. 2 Distribution of random number and histogram

Where $N = 256$, $R(s_i)$ is the probability of s_i. Ideally, entropy should be 8, which indicates that information is uncertain. Through the measurement, we get an entropy value of 7.9986 that is very close to the ideal value, which proves the feasibility of the scheme.

3.2 Randomness analysis

A test suite is a collection of statistical randomness test that is designed to test the randomness properties of sequences. NIST test suite [14] has been used in this study to assess randomness. Table 1 shows the results of NIST test when the chaotic sequences generated by different chaotic systems are used as real data. It can be seen from the data in Table 1 that the training results of high-dimensional chaotic system or low-dimensional chaotic system can pass NIST test when the chaotic sequence is used as the training set. Table 2 also lists the situation of using chaotic system to generate random number, using other software generated random number as training set and using chaotic sequence as training set. As can be seen from the data in Table 2, only our proposed scheme can pass all NIST tests.

3.3 Efficiency analysis

In order to test the efficiency of random numbers generation by LSGAN and chaotic system, we tested the time of generating 80000, 160000, 320000 and 1600000 random numbers respectively. As shown in Table 3, the time of random numbers generated by LSGAN is obviously better than that of the random numbers generated by the chaotic system. With the increasing amount of random number data, the advantage of LSGAN is more obvious. It shows that the random numbers generated by our method can improve encryption efficiency. At the same time, we find that the random numbers generated by QCNN, 4D hyperchaotic system and Lorenz chaotic system iterating 100 times in LSGAN can pass all NIST tests, while fractional Chen hyperchaotic system, 3D chaotic system and Henon chaotic system need to iterate 500 times in LSGAN. It shows that when the chaotic sequence generated by different chaotic systems are used as training sets, different iteration times are needed to pass all NIST tests.

4 Encryption algorithm

We propose an image encryption scheme based on LSGAN and a chaotic system to generate secure keys. The scheme uses LSGAN to generate the key of image encryption, and encrypts the image through the designed selective scrambling method and superposition diffusion method to obtain the cipher-image with good security. The scheme solves the problems of key randomness and low key generation efficiency. The overall block diagram of the encryption scheme is shown in Fig. 3, and the encryption flow chart is shown in Fig. 4.

Step 1. The input is an original image P in which M and N express its the width and height of the image, respectively.

Step 2. Four groups of security keys K1, K2, K3 and K4 with the size of 256×256 are generated using the method described in Part 2.

Step 3. Divide P into 16 equal-sized blocks, $B_i(i = 1, 2, 3, 16)$. As shown in Fig. 5, there are 12 inter-block exchanges, so 12 pairs of coordinate blocks $Ind_i(x, y)$ are generated by

Table 1 NIST test: P-value (1) is the real data set generated by QCNN chaotic system, P-value(2) is the real data set generated by 4D hyperchaotic system, P-value(3) is the real data set generated by Fractional Chen hyperchaotic system, P-value(4) is the real data set generated by 3D hyperchaotic system, P-value(5) is the real data set generated by Lorenz chaotic system, P-value(6) is the real data set generated by Henon chaotic system

Test name		P-value(1)	P-value(2)	P-value(3)	P-value(4)	P-value(5)	P-value(6)	result
approximate entropy		0.4363	0.4800	0.9299	0.3289	0.9983	0.1701	success
block-frequency		0.6474	0.3084	0.8880	0.8279	0.8863	0.7633	success
cumulative sums forward		0.5968	0.1856	0.3526	0.1705	0.9757	0.3751	success
FFT		0.1883	0.3972	0.0173	0.6869	0.1166	0.1841	success
frequency test		0.5397	0.4344	0.2038	0.1043	0.7092	0.3268	success
linear complexity		0.9967	0.1003	0.3864	0.3207	0.6388	0.405	success
long runs of ones		0.0417	0.6963	0.1602	0.2325	0.5087	0.3119	success
no overlapping templates		0.1059	0.8145	0.3532	0.4513	0.7065	0.4498	success
overlapping templates		0.0861	0.3525	0.8859	0.1914	0.2194	0.107	success
rank		0.8850	0.6378	0.5344	0.7205	0.3301	0.6293	success
runs		0.9497	0.6772	0.2484	0.9896	0.3254	0.9022	success
serial	1	0.875	0.1614	0.9509	0.0509	0.8459	0.9983	success
serial	2	0.9469	0.0327	0.6722	0.4824	0.7946	0.9975	success
universal		0.9022	0.6809	0.1163	0.7181	0.0163	0.1888	success
random excursions	X=-4	0.1284	0.0105	0.4739	0.6382	0.6555	0.6098	success
	X=-3	0.1406	0.4557	0.2245	0.3761	0.8990	0.2244	success
	X=-2	0.7013	0.0210	0.9279	0.3854	0.4302	0.9636	success
	X=-1	0.2029	0.2648	0.3125	0.2491	0.2173	0.1853	success
	X=1	0.6736	0.9775	0.1340	0.3928	0.6350	0.8933	success
	X=2	0.1585	0.5471	0.0473	0.7091	0.3970	0.2102	success
	X=3	0.3081	0.0774	0.1470	0.9706	0.3140	0.1776	success
	X=4	0.6997	0.4331	0.4057	0.2969	0.2433	0.3732	success

Table 1 (continued)

Test name		P-value(1)	P-value(2)	P-value(3)	P-value(4)	P-value(5)	P-value(6)	result
random excursions variant	X=-9	0.7113	0.8445	0.5124	0.9534	0.8064	0.1335	success
	X=-8	0.8955	0.9871	0.4521	0.906	0.8158	0.0835	success
	X=-7	0.9213	0.7825	0.5466	0.774	0.7793	0.0266	success
	X=-6	0.5861	0.5930	0.6152	0.4121	0.8490	0.0267	success
	X=-5	0.1401	0.9256	0.8548	0.2196	0.6520	0.0454	success
	X=-4	0.0476	0.2847	0.9237	0.1901	0.4657	0.0556	success
	X=-3	0.0517	0.133	0.6103	0.0869	0.2761	0.1795	success
	X=-2	0.1094	0.1961	0.4005	0.1483	0.1597	0.8399	success
	X=-1	0.4607	0.3508	0.2545	0.7543	0.1124	0.5313	success
	X=1	0.6471	0.7558	0.0181	0.4131	0.1819	0.3025	success
	X=2	0.4718	0.8294	0.0220	0.8894	0.1692	0.3075	success
	X=3	0.3688	0.2969	0.0370	0.8632	0.0814	0.3191	success
	X=4	0.2991	0.1354	0.0312	0.9854	0.2048	0.2807	success
	X=5	0.4711	0.0620	0.0155	0.8222	0.6093	0.2536	success
	X=6	0.5345	0.0373	0.0156	0.9942	0.5533	0.1958	success
	X=7	0.6985	0.0317	0.0366	0.9946	0.3495	0.2361	success
	X=8	0.9685	0.0250	0.0364	0.7845	0.1462	0.309	success
	X=9	0.8242	0.0142	0.0246	0.8978	0.0757	0.2031	success

Table 2 Comparison of NIST test results, (1) is the QCNN chaotic system, (2) is the 4D hyperchaotic system, (3) is the Fractional Chen hyperchaotic system,(4) is the 3D hyperchaotic system, (5) is the Lorenz chaotic system, (6) is the Henon chaotic system, (7) is the Pseudo random number, (8) is the our scheme

Test name		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
approximate entropy		success	success	fail	fail	success	success	success	success
block-frequency		success	success	success	success	success	success	success	success
cumulative sums forward		success	success	success	success	success	fail	success	success
FFT		success	fail	success	success	fail	success	fail	success
frequency test		success	success	success	fail	success	success	success	success
linear complexity		success	success	success	success	success	success	success	success
long runs of ones		success	success	success	success	success	success	success	success
no overlapping templates		success	fail	success	success	fail	success	success	success
overlapping templates		fail	success	success	success	success	success	success	success
rank		success	success	success	success	success	success	success	success
runs		success	success	success	success	success	success	success	success
serial	1	success	success	success	success	success	success	success	success
serial	2	success	success	success	success	success	success	success	success
universal		success	success	fail	success	success	success	fail	success
random excursions	X=-4	success	success	success	fail	success	success	success	success
	X=-3	success	success	success	success	success	success	success	success
	X=-2	success	success	success	success	success	success	success	success
	X=-1	success	success	success	success	success	success	success	success
	X=1	success	success	success	success	success	success	success	success
	X=2	success	success	success	success	success	success	success	success
	X=3	success	success	success	success	success	success	success	success
	X=4	success	success	success	success	success	success	success	success

Table 2 (continued)

Test name		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
random excursions variant	X=-9	success	success	success	success	success	fail	success	success
	X=-8	success	success	success	success	success	success	success	success
	X=-7	success	success	success	success	success	success	success	success
	X=-6	success	success	success	success	success	success	success	success
	X=-5	success	success	success	success	success	success	success	success
	X=-4	success	success	success	success	success	fail	success	success
	X=-3	success	success	success	success	success	success	success	success
	X=-2	success	success	success	success	success	success	success	success
	X=-1	success	success	success	success	success	success	success	success
	X=1	success	success	success	success	success	success	success	success
	X=2	success	success	success	success	success	success	success	success
	X=3	success	success	success	success	success	success	success	success
	X=4	success	success	success	success	success	success	success	success
	X=5	success	success	success	success	success	success	success	success
	X=6	success	success	success	success	success	success	success	success
	X=7	success	success	success	success	success	success	success	success
	X=8	success	success	success	success	success	success	success	success
	X=9	success	success	success	success	success	success	success	success

Table 3 Random numbers generation time of chaotic system and LSGAN

	Key generation time(s)							
Random numbers	QCNN	4D hyper chaotic	Fractional Chen hyperchaotic	3D chaotic	Lorenz	Henon	GANs (100 times)	GANs (500 times)
80000	0.508	4.048	8.798	40.877	3.705	6.914	0.734	2.102
160000	1.001	8.256	17.431	73.153	6.865	16.797	0.861	2.698
320000	1.971	16.505	34.985	150.811	13.452	30.099	0.974	3.21
1600000	10.179	84.373	172.388	693.101	62.685	152.235	1.581	5.923

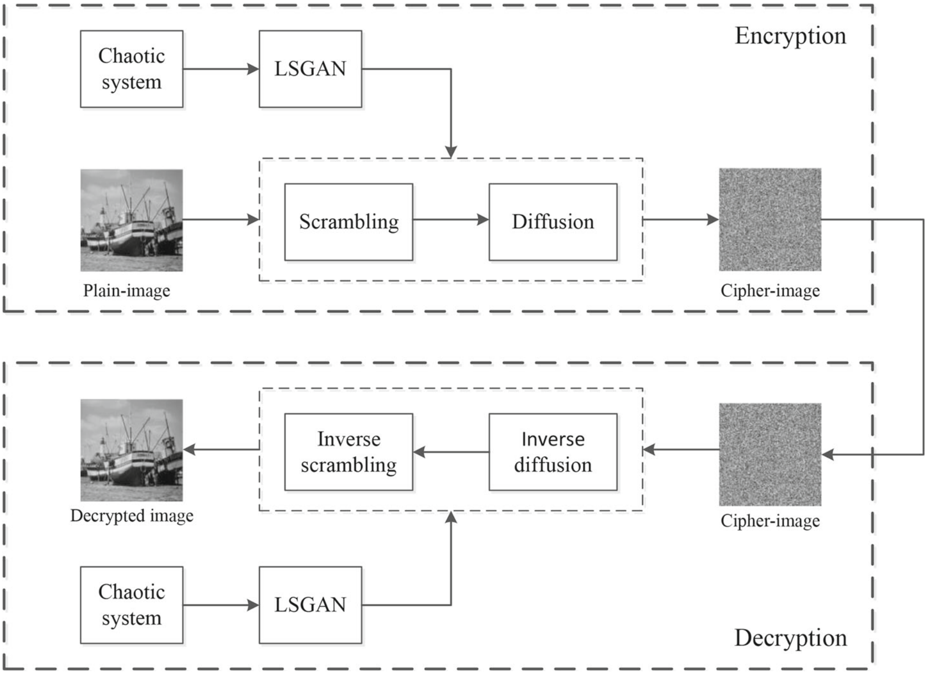

Fig. 3 Image encryption flow chart

using the security keys K1 and K2 to control the inter-block pixel exchange.

$$\begin{cases} Ind_i(x) = \sim ismember(Ind_i(mod(ceil(\frac{M\times N}{i}), 64)), K1) \\ Ind_i(y) = \sim ismember(Ind_i(mod(ceil(\frac{M\times N}{i}), 64)), K2) \end{cases} \tag{12}$$

Step 4. Design a selectieve scrambling rule.$B_1 \leftrightarrow B_{11}$, $B_2 \leftrightarrow B_{10}$, $B_3 \leftrightarrow B_{11}$, $B_4 \leftrightarrow B_{10}$, $B_5 \leftrightarrow B_7$, $B_6 \leftrightarrow B_8$, $B_6 \leftrightarrow B_{14}$, $B_6 \leftrightarrow B_{16}$, $B_7 \leftrightarrow B_{13}$, $B_7 \leftrightarrow B_{15}$, $B_9 \leftrightarrow B_{11}$, $B_{10} \leftrightarrow B_{12}$. The principle of pixel exchange between blocks is controlled by random coordinate blocks $Ind_i(x, y)$. The scrambled image is P'.

Step 5. Use $K1$ and $K2$ to construct a global coordinate Q_x and Q_y, by which the whole P' is scrambled and the pixel correlation is further reduced.

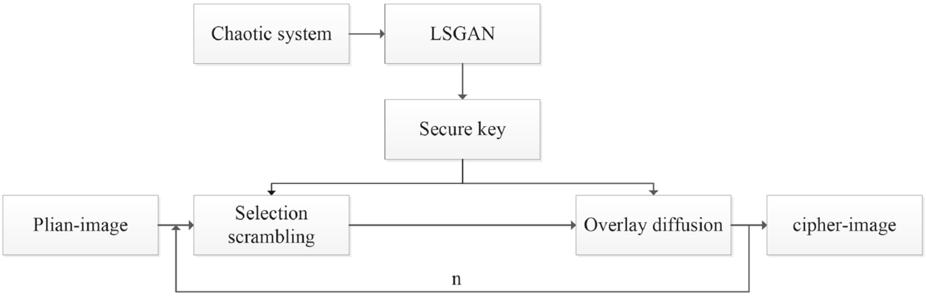

Fig. 4 Image encryption flow chart

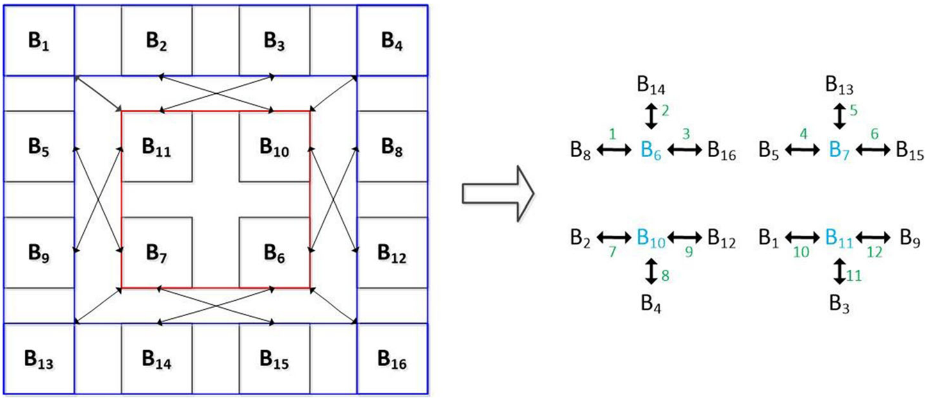

Fig. 5 Schematic diagram of selective scrambling method

$$\begin{cases} Q_x(i) =\sim ismember(Q_x(i), mod(K1, 256)) \\ Q_y(j) =\sim ismember(Q_y(j), mod(K2, 256)) \end{cases} \tag{13}$$

$$\begin{cases} \gamma = P'(Q_x(i), mod(K1, 256)) \\ P'(Q_x(i), Q_y(j)) = P'(i, j) \\ P'(i, j) = \gamma \end{cases} \tag{14}$$

Step 6. Dynamic overlay diffusion: Firstly, K3, K4 and the scrambled image P' are transformed into a matrix of 16 × 4096 size, and the 16 × 16 pixel block is taken as an overlay block. Then K3 and K4 are used alternatively to do XOR operation with P', as shown in the Fig. 6, and the cipher-image Cp is finally obtained.

$$\begin{cases} Dp(i) = bitxor(K3(i), P'(i)) \\ Cp(i) = bitxor(Dp(i), K4(i)) \end{cases} \tag{15}$$

Where i denotes the location of the current block, i = 1,2,3,...256.

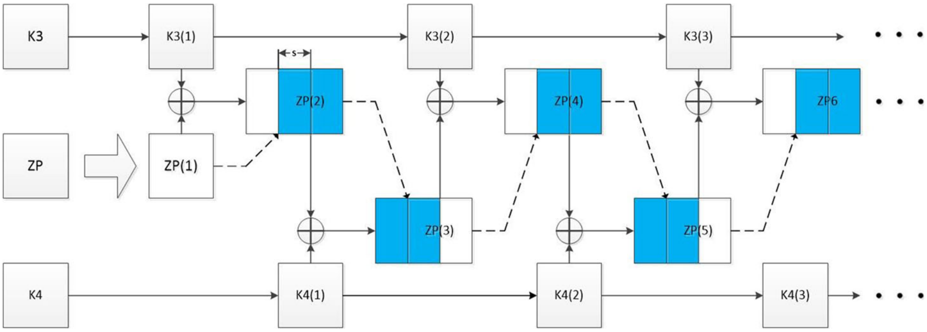

Fig. 6 Schematic diagram of overlay diffusion method

5 Experimental simulation and performance analysis

In this section, to prove the security of our scheme, the 256 × 256 traditional 8-bit grayscale images are used as the original images. Common security analyses, including statistical data and common attacks and so on, are also made and compared. Figure 7a, d and g are plain-images, (b) (e) (h) are cipher-images, (c) (f) (i) are decrypted images.

5.1 Key space

It is important for an image encryption algorithm to have a large enough security key space to resist the brute force attacks. Taking a 4-D hyperchaotic system as an example, only the initial value and parameters of the chaotic system are considered, if the length of every subkey is set to 16 decimals, the key space of our algorithm will be $10^{16\times 4} = 2^{192} > 2^{100}$ [40]. As a result, it is large enough to resist brute force attacks.

5.2 Information Entropy

In many image processing processes, information entropy is often used to measure the randomness of a noise map. Its entropy value $H(m)$ can be calculated by the following equation [49]:

$$H(m) = -\sum\nolimits_{i=0}^{2^N-1} p(m_i)\log_2 p(m_i) \tag{16}$$

Where N is the gray level, $p(m_i)$ is the probability of m_i. The entropy should ideally be 8 for a cipher-image with 256 gray levels, which indicates that the information is uncertain. Therefore, the information entropy of the encrypted image with high security should be close to 8. Table 4 lists the results of the information entropy of the cipher-image. Compared with the algorithms in reference [10–12], we can see that in terms of information entropy, the proposed scheme is better, and the pixel distribution of the encrypted image is more random.

5.3 Histogram analysis

The histogram is one of the important criteria for measuring the performance of an image encryption algorithm. It can describe the number of pixels corresponding to the gray level and the frequency of occurrence of each gray level, showing the distribution law of image pixels. As can be seen from Fig. 8, the histograms of all cipher-images are uniformly distributed, so the strong regularity of the pixels of plain-images is not brought into the cipher-images. This means that the attacker can’t get any useful statistical information from the cipher-image to attack the algorithm. The scheme can provide a good confusion effect for encrypted images.

5.4 Correlation of two adjacent pixels

Because of the characteristics of digital images, plain-images usually show a certain degree of correlation between two adjacent pixels. In order to assess the correlation between adjacent pixels,4000 pairs of adjacent pixels (x_i, y_i) are randomly selected from the plain-image

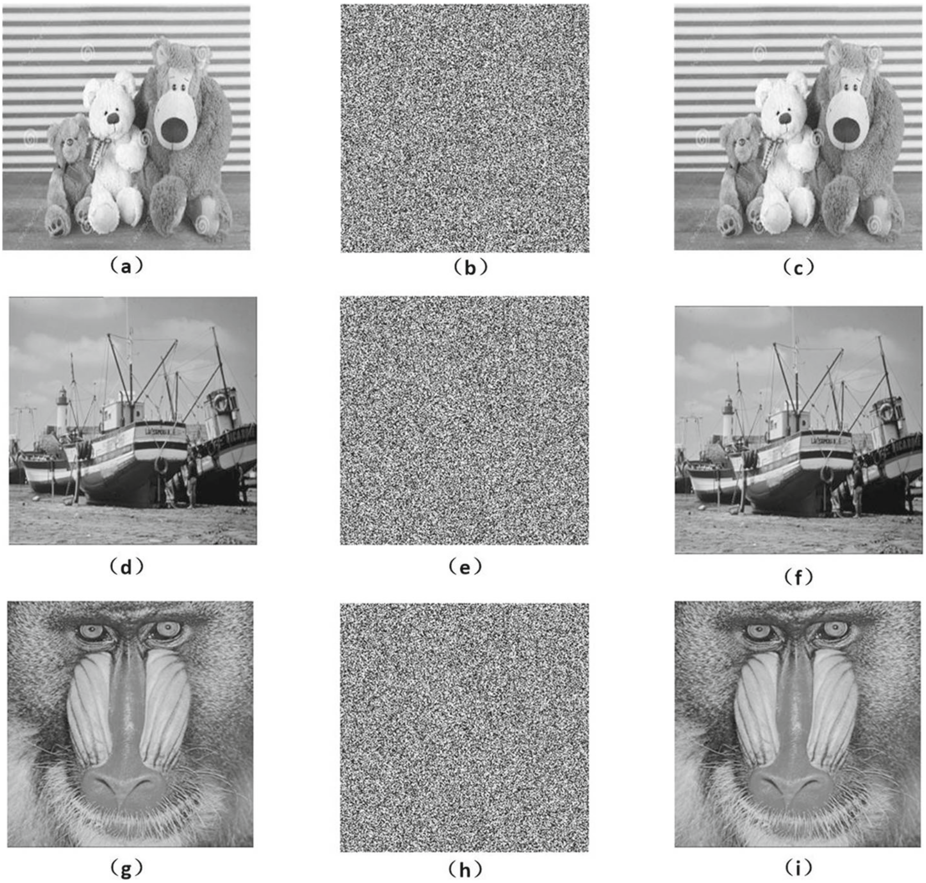

Fig. 7 Experimental results: **a** plain-image of ‘bear’, **b** cipher-image of ‘bear’, **c** decryption image of ‘bear’, **d** plain-image of ‘boat’, **e** cipher-image of ‘boat’, **f** decryption image of ‘boat’, **g** plain-image of ‘baboon’, **h** cipher-image of ‘baboon’ and **i** decryption image of ‘baboon’

and cipher-image, and their correlation coefficients are calculated.

$$\begin{aligned} r_{xy} &= \frac{cov(x,y)}{\sqrt{D(x)D(y)}} \\ E(x) &= \frac{1}{S}\sum_{i=1}^{S} x_i \\ D(x) &= \frac{1}{S}\sum_{i=1}^{S}(x_i - E(x))^2 \\ cov(x,y) &= \frac{1}{S}\sum_{i=1}^{S}(x_i - E(x))(y_i - E(y)) \end{aligned} \tag{17}$$

Table 4 Information entropy for the encryption

	‘bear’	‘boat’	‘baboon’	[11]	[12]	[10]
Entropy	7.9975	7.9975	7.9977	7.9912	7.9972	7.9973

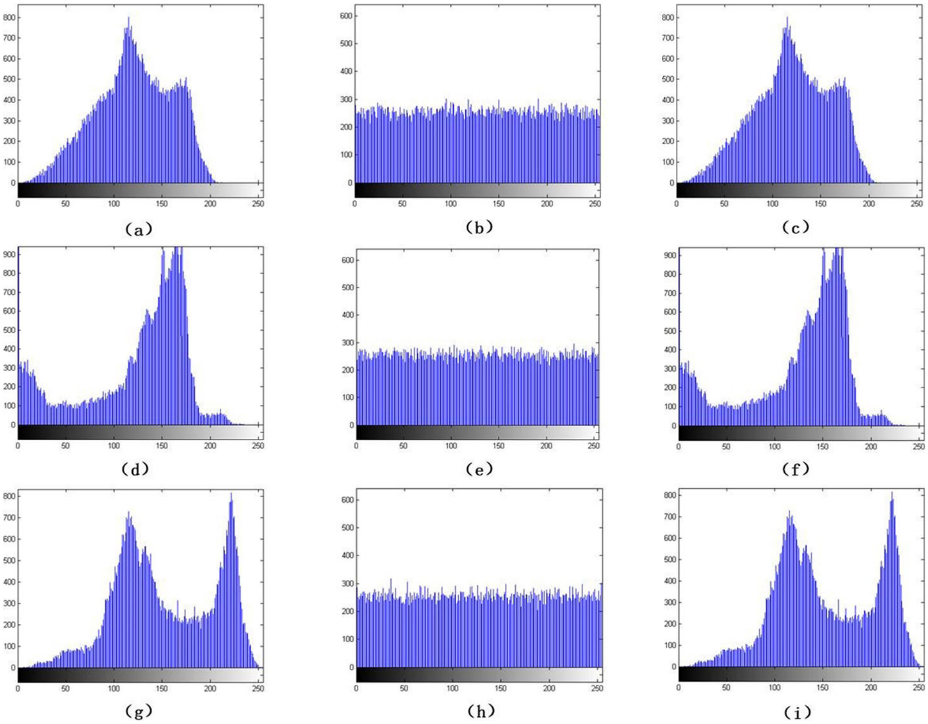

Fig. 8 Histogram analysis: **a**, **d**, **f** are histograms of the plain-image of 'bear', 'boat' and 'baboon' **b**, **d**, **j** are histograms of the cipher-image of 'bear', 'boat' and 'baboon', **c**, **e**, **i** are histograms of the decryption image of 'bear' , 'boat' and 'baboon'

Where x and y represent the gray value of two adjacent pixels, Correlation coefficient performances of the proposed scheme have been presented in Fig. 9 and Table 5, from which we can see that the correlation coefficients of the secret image have been reduced dramatically, while the correlation coefficients of the plain image are close to 1. Literature [45, 54, 55] provide a comparative algorithm. It is obvious that adjacent pixels of plain-image has strong correlation, while those of cipher-image have low correlation. This further proves that the scheme can effectively resist statistical attacks.

Table 5 Correlation coefficients of the cipher-images

	'bear'	'boat'	'baboon'	[54]	[55]	[45]
Horizontal	7.9975	7.9975	7.9977	7.9972	7.9973	7.9912
Vertical	0.0015	−0.0113	0.0286	−0.00209	0.0093	0.00964
Vertical	−0.0134	0.0056	0.0042	−0.01618	0.0159	0.01963
Diagonal	−0.0008	−0.0004	0.0216	0.0178	0.0097	0.01963

Table 6 MSE and PSNR for the cipher-image

	bear	boat	baboon	[46]	[10]	[51]
MSE	8798.2	8482.4	6988.5	–	–	6252.83
PSNR	8.6868	8.8456	9.6457	29.79	8.9448	0.17003

5.5 Peak signal-to-noise ratio

The difference between plain-image and cipher-image is measured by mean square error (MSE). MSE is defined as:

$$MSE = \frac{\sum_i \sum_j (P(i,j)-C(i,j))^2}{T} \times 100\% \tag{18}$$

T represents the number of pixels in an encrypted image. The larger the value of MSE, the greater the difference between the encrypted image and the original image, and the better the image encryption effect.

Peak signal-to-noise ratio (PSNR) is a ratio between plain-image and cipher-image. PSNR is defined as:

$$PSNR = 10log_{10}(\frac{I_{max}^2}{MSE}) \tag{19}$$

max is the maximum pixel value of plain-image. In a good encryption algorithm, PSNR should be as low as possible, which indicates that the encrypted image are more randomness [41]. The MSE and PSNR values of the cipher-image are shown in Table 6, and references [10, 46, 51] are used for comparison.

5.6 Differential attack

Sometimes, attackers make a tiny change in the original plain image, and then encrypts both the original plain image and the changed plain image by the same encryption scheme, and try to find out the relationship between plain image and its cipher image by comparing the two encrypted images. The number of pixels change rate (NPCR) [26] and the unified average changing intensity (UACI) [15] are two criteria for analyzing differential attacks.

$$\begin{aligned} NPCR &= \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\% \\ UACI &= \frac{1}{M \times N}[\sum_{i,j} \frac{|C_1(i,j)-C_2(i,j)|}{255}] \times 100\% \end{aligned} \tag{20}$$

Where $C_1(i, j)$ and $C_2(i, j)$ are the two encrypted images mentioned above and $D(i, j)$ is computed as

$$D(i,j) = \begin{cases} 0, & C_1(i,j) = C_2(i,j) \\ 1, & C_1(i,j) \neq C_2(i,j) \end{cases} \tag{21}$$

The ideal NPCR value and UACI value of the cipher-image are 99.6093% and 33.4653% respectively [56]. Table 7 lists the calculations of NPCR and UACI and compares them

Table 7 UACI and NPCR performances

	'bear'	'boat'	'baboon'	[53]	[52]	[10]
NPCR	99.6063	99.6033	99.6124	99.62	99.71	99.62
UACI	33.4512	33.4619	33.4544	33.53	33.45	33.41

Table 8 Speed test

Algorithm	Ours scheme	[53]	[52]	[51]
Speed(s)	0.9437	1.001	3.06	9.89

with those of other algorithms [10, 52, 53]. A conclusion can be drawn that the proposed algorithm has good property in resisting differential attacks.

5.7 Speed analysis

Speed of the encryption algorithm is also considered as a crucial factor. The proposed GAN key generation scheme reduces the time of key generation greatly, and the designed scrambling and diffusion algorithm also improves the encryption efficiency. Compared with other encryption schemes [51–53] in Table 8, our algorithm shows some advantages in speed. At the same time, due to the symmetric structure, the time cost of encryption and decryption are the same. Therefore, the proposed algorithm shows that it can can be used for real-time transportation.

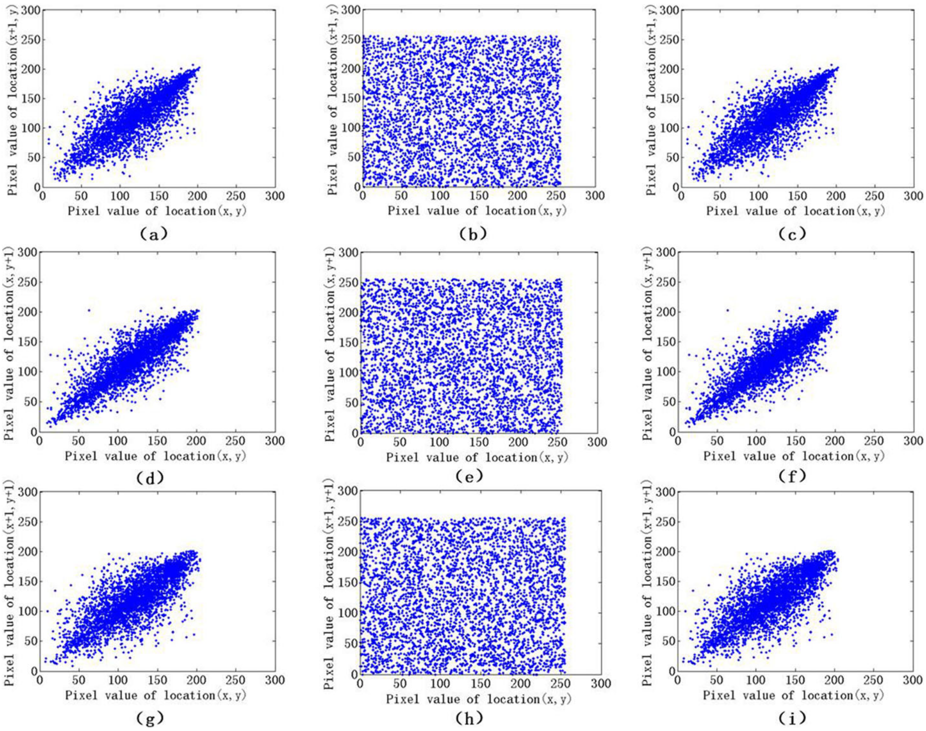

Fig. 9 The correlation plots of 'baboon' image: **a** horizontal correlation of plain-image, **b** horizontal correlation of cipher-image, **c** horizontal correlation of decrypted image, **d** vertical correlation of plain-image, **e** vertical correlation of cipher-image, **f** vertical correlation of decrypted image, **g** diagonal correlation of plain-image, **h** diagonal correlation of cipher- image, **i** diagonal correlation of decrypted image

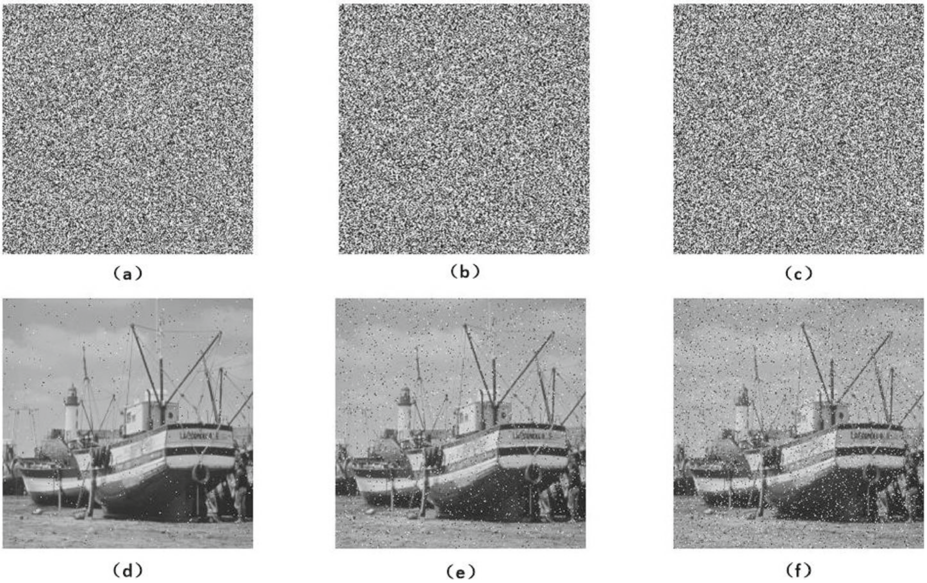

Fig. 10 The images on the first row are the cipher-image with 0.01, 0.05 and 0.1 salt and pepper noise respectively. The images on the second row are decrypted results of corresponding encrypted images

5.8 Robustness against noise

Practically, some noise may be added through the transmission of encrypted images from the transmitter to the receiver. This can lead to an inevitable error, causing difficulties in

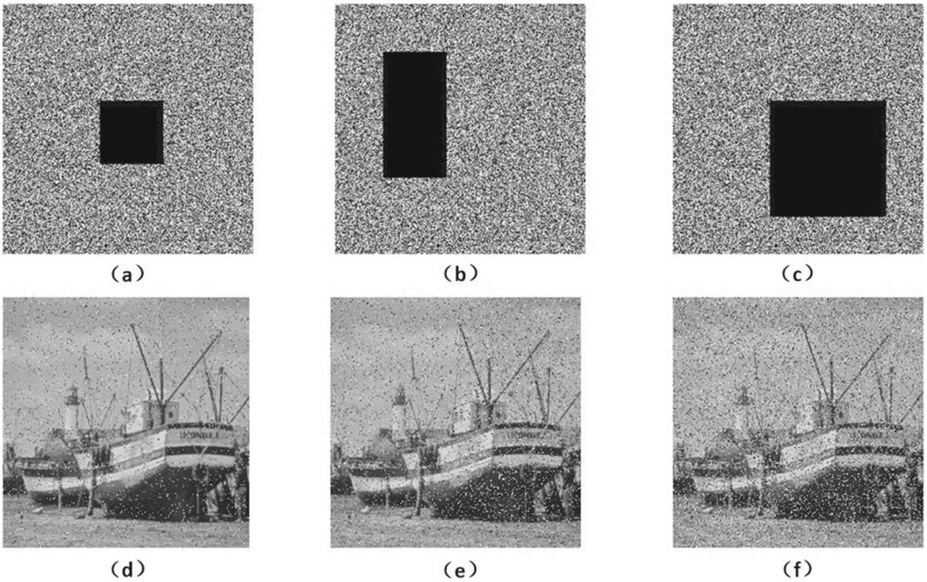

Fig. 11 The images on the first row are the encrypted image after cutting 64 × 64, 64 × 128 and 128 × 128 respectively. Those images on the second row are the decrypted results of corresponding encrypted images

decryption. A good encryption algorithm should be able to resist noise attacks. Salt and Pepper noise is added to the ciphered image of Boat, with different densities. The decrypted image of each case is shown in Fig. 10, which validates the efficiency of the system against noise attacks. Therefore, our algorithm has good robustness and can efficiently resist noise attacks.

5.9 Robustness against cropping

An effective cryptosystem must take information loss into account. To evaluate its robustness of resisting cropping attacks, parts with 64 × 64, 64 × 128 and 128 × 128, are deleted from the cipher-image ’boat’. As shown in Fig. 11a-d, the decrypted plain images continue to be meaningful. Hence, our method is robust against this kind of attacks.

Through a series of performance analysis and anti-attack detection, it can be seen from the key space, information entropy, histogram, adjacent pixel correlation, peak signal-to-noise ratio and speed test results that its security is considerable. From the test results of differential attack, cropping attack and noise attack, it has been found that its anti-attack ability has certain advantages over the similar literature.

6 Conclusion

This paper innovatively introduces LSGAN into image encryption. The good learning ability of LSGAN model is used to obtain the encryption key. Through NIST test, efficiency test, entropy and histogram analysis, it is proved that the random number generation scheme proposed in this paper can provide strong guarantee for image security, and expand the new idea of image security research. In order to improve the security of encrypted image, a scrambling method and a coverage diffusion algorithm are designed. The experimental results show that the scheme has a large key space and an average entropy of 7.9976. The anti-attack ability of the scheme is also verified by differential attack, cut attack and noise attack. In the future, we will improve the current model and other shortcomings, try to make the encrypting and decrypting parties generate keys synchronously, and reduce the number of keys transmitted on the network.

Acknowledgements This work was supported in part by China’s Natural Science Foundation Project(20190201188JC) of the Science and Technology Department of Jilin Province and China’s Science and Technology Project ($JJKH$20181137KJ) for the 13th Five-Year Plan of the Education Department of Jilin Province.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abdulla AA (2015) Exploiting similarities between secret and cover images for improved embedding efficiency and security in digital steganography. University of Buckingham
2. Abdulla AA, Sellahewa H, Jassim SA (2014) Stego quality enhancement by message size reduction and fibonacci bit-plane mapping. in International Conference on Research in Security Standardisation. Springer
3. Babbage S et al (2009) ECRYPT yearly report on algorithms and keysizes
4. Brunk H, Rogers E, Hannigan BT (2002) Adjusting an electronic camera to acquire a watermarked image. Google Patents
5. Cao W, Mao Y, Zhou Y (2020) Designing a 2D infinite collapse map for image encryption. Signal Processing, pp 107457
6. Chai X et al (2019) A novel image encryption scheme based on DNA sequence operations and chaotic systems. Neural Comput Appl 31(1):219–237
7. Chen J et al (2018) Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption. Signal Process 142:340–353
8. De Bernardi M, Khouzani M, Malacaria P (2018) Pseudo-Random Number Generation Using Generative Adversarial Networks. in Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer
9. Di X et al (2017) A semi-symmetric image encryption scheme based on the function projective synchronization of two hyperchaotic systems. PloS one 12(9):e0184586
10. El-Latif AAA, Niu X (2013) A hybrid chaotic system and cyclic elliptic curve for image encryption. AEU-Int J Electron Commun 67(2):136–143
11. Fathi-Vajargah B, Kanafchian M, Alexandrov V (2018) Image encryption based on permutation and substitution using Clifford Chaotic System and logistic map. J Comput 13(3):309–326
12. Farwa S et al (2019) Fresnelet approach for image encryption in the algebraic frame (Retraction of Vol 334, Pg 343, 2018). ELSEVIER SCIENCE INC 360 PARK AVE SOUTH, NEW YORK, NY 10010-1710 USA
13. Fouda JAE et al (2014) A fast chaotic block cipher for image encryption. Commun Nonlinear Sci Numer Simul 19(3):578–588
14. Gong L et al (2019) An image compression and encryption algorithm based on chaotic system and compressive sensing. Opt Laser Technol 115:257–267
15. Ghebleh M, Kanso A, Noura H (2014) An image encryption scheme based on irregularly decimated chaotic maps. Signal Process Image Commun 29(5):618–627
16. Hao-Xiang W et al (2010) Nonlinear feedback control of a novel hyperchaotic system and its circuit implementation. Chin Phys B 19(3):030509
17. Hua Z, Yi S, Zhou Y (2018) Medical image encryption using high-speed scrambling and pixel adaptive diffusion. Signal Process 144:134–144
18. Hua Z, Zhou Y, Huang H (2019) Cosine-transform-based chaotic system for image encryption. Inf Sci 480:403–419
19. Huang C et al (2017) Context-aware generative adversarial privacy. Entropy 19(12):656
20. Jin C, Liu H (2017) A color image encryption scheme based on arnold scrambling and quantum chaotic. IJ Netw Secur 19(3):347–357
21. Ke Y et al (2019) Generative steganography with Kerckhoffs' principle. Multimed Tools Appl 78(10):13805–13818
22. Liu Y, Tong X, Ma J (2016) Image encryption algorithm based on hyper-chaotic system and dynamic S-box. Multimed Tools Appl 75(13):7739–7759
23. Lin J et al (2020) An image encryption method based on logistic chaotic mapping and DNA coding. In: MIPPR 2019, Remote Sensing Image Processing, Geographic Information Systems, and Other Applications. International Society for Optics and Photonics
24. Liu H, Wang X (2010) Color image encryption based on one-time keys and robust chaotic maps. Comput Math Appl 59(10):3320–3327
25. Li X, Zhou C, Xu N (2018) A secure and efficient image encryption algorithm based on DNA coding and spatiotemporal chaos. IJ Netw Secur 20(1):110–120
26. Liu H, Kadir A, Sun X (2017) Chaos-based fast colour image encryption scheme with true random number keys from environmental noise. IET Image Process 11(5):324–332
27. Liang X et al (2017) Dual motion gan for future-flow embedded video prediction. In: Proceedings of the IEEE International Conference on Computer Vision
28. Liu J et al (2020) Recent advances of image steganography with generative adversarial networks. IEEE Access 8:60575–60597

29. Liu H et al (2018) Chaos based adaptive double-image encryption scheme using hash function and S-boxes. Multimedia Tools and Applications 77(1):1391–1407
30. Mahajan P (2020) Recent Advances in Generative Adversarial Networks: An Analysis along with its outlook. in 2020 10th International Conference on Cloud Computing. Data Science & Engineering, Confluence. IEEE
31. Mao Y, Chen G, Lian S (2004) A novel fast image encryption scheme based on 3D chaotic baker maps. Int J Bifurcat Chaos 14(10):3613–3624
32. Parker AT, Short KM (2001) Reconstructing the keystream from a chaotic encryption scheme. IEEE Trans Circ Syst I: Fund Theory Appl 48(5):624–630
33. Peng J et al (2020) Image Encryption Based on Fractional-order Chen Hyperchaotic System. In: 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA). IEEE
34. Qi-Ling HE et al (2017) A New Chaotic System and its Linear Feedback Synchronization. Journal of Chengdu University of Information Technology
35. Stinson DR, Paterson M (2018) Cryptography: theory and practice. CRC press
36. Schneier B (2007) Applied cryptography: protocols, algorithms, and source code in C. Wiley
37. Sreelaja N, Sreeja N (2016) An image edge based approach for image password encryption. Secur Commun Netw 9(18):5733–5745
38. Sun D et al (2017) A New Mimicking Attack by LSGAN. In: 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE
39. Sen W et al (2007) Chaotic phenomena in Josephson circuits coupled quantum cellular neural networks. Chin Phys 16(9):2631
40. 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
41. SK NK, HS SK, Panduranga H (2012) Encryption approach for images using bits rotation reversal and extended hill cipher techniques. Int J Comput Appl 59(16)
42. Taiyong L et al (2017) A novel image encryption algorithm based on a Fractional-Order hyperchaotic system and DNA computing. Complexity 2017:1–13
43. Venkatesan R, Jakubowski M, Jayram TS (2003) Technique for watermarking an image and a resulting watermarked image. Google Patents
44. Wang X-Y, Li Z-M (2019) A color image encryption algorithm based on Hopfield chaotic neural network. Opt Lasers Eng 115:107–118
45. Wang X-Y, Gu S-X, Zhang Y-Q (2015) Novel image encryption algorithm based on cycle shift and chaotic system. Opt Lasers Eng 68:126–134
46. Wang X, Zhu X, Zhang Y (2018) An image encryption algorithm based on Josephus traversing and mixed chaotic map. IEEE Access 6:23733–23746
47. Wang X-Y et al (2018) A novel color image encryption scheme using DNA permutation based on the Lorenz system. Multimed Tools Appl 77(5):6243–6265
48. Wang X et al (2018) KDGAN: Knowledge distillation with generative adversarial networks in Advances in Neural Information Processing Systems
49. Wang X, Gao S (2020) Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. Inf Sci 539:195–214
50. Wang X, Wang Q, Zhang Y (2015) A fast image algorithm based on rows and columns switch. Nonlinear Dyn 79(2):1141–1149
51. Wei-Bin C, Xin Z (2009) Image encryption algorithm based on Henon chaotic system in 2009, International Conference on Image Analysis and Signal Processing. IEEE
52. Wu X et al (2017) A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps. IEEE Access 5:6429–6436
53. Xu L et al (2017) A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion. Opt Lasers Eng 91:41–52
54. Yang H et al (2010) A fast image encryption and authentication scheme based on chaotic maps. Commun Nonlinear Sci Numer Simul 15(11):3507–3517
55. Ye G, Huang X (2016) A secure image encryption algorithm based on chaotic maps and SHA-3. Secur Commun Netw 9(13):2015–2023
56. Ye G et al (2018) A Chaotic Image Encryption Algorithm Based on Information Entropy. Int J Bifurcat Chaos 28(01):1850010
57. Yin Q, Wang C (2018) A new chaotic image encryption scheme using breadth-first search and dynamic diffusion. Int J Bifurcat Chaos 28(04):1850047

58. Yin D, Yang Q (2018) GANS based density distribution privacy-preservation on mobility data. Security and Communication Networks 2018
59. Zhang M, Tong X (2015) A new algorithm of image compression and encryption based on spatiotemporal cross chaotic system. Multimed Tools Applx 74(24):11255–11279
60. Zhang R, Dong S, Liu J (2019) Invisible steganography via generative adversarial networks. Multimed Tools Appl 78(7):8559–8575

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.