Entropy **2015**, *17*, 181-196; doi:10.3390/e17010181

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Article

An Image Encryption Scheme Based on Hyperchaotic Rabinovich and Exponential Chaos Maps

Xiaojun Tong [1,*], Yang Liu [1], Miao Zhang [1], Hui Xu [1] and Zhu Wang [2]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Weihai, 264209, China; E-Mails: liuyang@hitwh.edu.cn (Y.L.); zhangmiaozm209@126.com (M.Z.); banmianstudent@163.com (H.X.)

[2] School of Information and Electrical Engineering, Harbin Institute of Technology, Weihai, 264209, China; E-Mail: wangzhu@hit.edu.cn

* Author to whom correspondence should be addressed; E-Mail: tong_xiaojun@163.com; Tel.: +86-130-6118-1039; Fax: +86-0631-568-7750.

Academic Editors: Guanrong Chen, C.K. Michael Tse, Mustak E. Yalcin, Hai Yu and Mattia Frasca

Received: 31 October 2014 / Accepted: 24 December 2014 / Published: 8 January 2015

Abstract: This paper proposes a new four-dimensional hyperchaotic map based on the Rabinovich system to realize chaotic encryption in higher dimension and improve the security. The chaotic sequences generated by Runge-Kutta method are combined with the chaotic sequences generated by an exponential chaos map to generate key sequences. The key sequences are used for image encryption. The security test results indicate that the new hyperchaotic system has high security and complexity. The comparison between the new hyperchaotic system and the several low-dimensional chaotic systems shows that the proposed system performs more efficiently.

Keywords: exponential chaos; hyperchaos; image encryption; pseudo-random sequence

PACS Codes: 05.45.−a; 05.45.Gg

1. Introduction

In recent years, chaos [1–4] has been used widely in encryption schemes. In 1979, Rossler [5] described the first hyperchaos system. Based on some classical systems, many scholars have made new development [6,7], but even so, generating a new hyperchaotic system is still a challenge.

Based on the Chen system, Jia [8] constructed a four-dimensional hyperchaos by adding one dimension and changing xy to y^2. Based on the Rossler system, Deng [9] constructed a new four dimensional hyperchaotic Rossler system by adding a feedback control. Based on the Lü system, Shen [10] added one new nonlinear term and Pang [11] added two nonlinear terms to construct the four-dimensional hyperchaos. Huang *et al*. constructed a four-dimensional non-linear dynamics system [12] based on the features of nonlinear parts of Qi attractor and Chen attractor. Zhang [13] constructed a Qi unified hyperchaos system by adding linear feedback control and new dimensions in the Qi chaos system. In addition, some three-dimensional chaotic systems, such as chaotic financial system [14] and Rabinovich system [15], are proposed and proved. These three-dimensional systems provide a reference and basis for future studies.

In all the chaotic systems, the hyperchaotic system has two or more than two positive Lyapunov exponents. To generate a hyperchaotic system, it needs at least four dimensions for the integer order continuous autonomous system. Chaotic sequences of hyperchaotic system are more dependent on the parameters and the initial conditions, so its dynamic behaviors are more difficult to predict and the chaos attractor is more complex. Diffusion and confusion can be carried out simultaneously in several dimensional spaces. Therefore, hyperchaotic system has a distinct advantage over low dimensional chaos.

2. Design and Dynamic Behavior Analysis of New Hyperchaotic System

In this section, a new hyperchaotic system is constructed and proved to be hyperchaotic. Then the dynamic behaviors of this new hyperchaotic system are presented. Finally, the exponential chaotic map is introduced. The proposed hyperchaotic system will be combined with the exponential chaotic map to design a pseudorandom number generator (PRNG) in Section 3.

2.1. Design of New Hyperchaotic System

Rabinovich system, closely associated with Lorenz chaotic system, is described as follows:

$$\begin{cases} \dot{x} = hy - ax + yz \\ \dot{y} = hx - by - xz \\ \dot{z} = -dz + xy \end{cases} \tag{1}$$

The dynamic properties of this system and that of the Lorenz system are similar, but they are not topology equivalent. When $a = 4$, $b = d = 1$, $4.84 < h < h_0$ (where h_0 ($\geq$4.92) is the value of one of the features), the system is chaotic. By adding a new parameter w to the Rabinovich system, a new hyperchaotic system is constructed. It is:

$$\begin{cases} \dot{x} = hy - ax + yz \\ \dot{y} = hx - by - xz \\ \dot{z} = -dz + xy + w^2 \\ \dot{w} = xy + cw \end{cases} \tag{2}$$

where xy is fed back into the new parameter w. The appearance of chaos attractor is controlled by the parameters.

As we know, hyperchaotic system must meet the following conditions:

(1) The dimension of the phase space of an autonomous system is at least four.
(2) There are two equations at least to increase the instability of the system. The two equations have one nonlinear term at least, respectively.
(3) The system has two or more than two positive Lyapunov exponents. Moreover, the sum of the four Lyapunov exponents is less than zero.
(4) The Lyapunov dimension of the system is a fraction.

For the proposed system, the first two conditions are satisfied obviously. Now we consider the last two conditions. When $a = 4$, $b = -0.5$, $d = 1$, $h = 8.1$ and $c = -2.2$, the four Lyapunov exponents of the proposed system, calculating by Wolf algorithm, are $\lambda_{L1} = 1.090046$, $\lambda_{L2} = 0.012243$, $\lambda_{L3} = -3.105106$ and $\lambda_{L4} = -4.697183$. Thus the system meets the third condition. The Lyapunov dimension can be calculated by Kaplan-Yorke conjecture:

$$D_l = k + \frac{1}{\left|\lambda_{L,k+1}\right|}\sum_{i=1}^{k}\lambda_{L,i} = 2 + \frac{\lambda_{L1} + \lambda_{L2}}{\left|\lambda_{L3}\right|} = 2 + \frac{1.090046 + 0.012243}{\left|-3.105106\right|} = 2.35499. \quad (3)$$

Then the system meets the fourth condition. So the proposed system is a hyperchaotic system.

Lyapunov exponent characterizes the separation rate of infinitesimally close trajectories. The bigger the Lyapunov exponent is, the faster the trajectories separate is. Moreover, the maximal Lyapunov exponent characterizes the typical dynamic speciality of a system. We have compared the Lyapunov exponents of the proposed hyperchaotic system and that of other hyperchaotic systems. The results are shown in Table 1.

Table 1. Lyapunov exponents comparison.

Hyperchaotic system	λ_{L1}	λ_{L1}	λ_{L1}	λ_{L1}
Proposed	1.090046	0.012243	−3.105106	−4.697183
Rössler system	0.112	0.019	0	−25.188
Reference [16]	0.648	0.153	0	−38.468

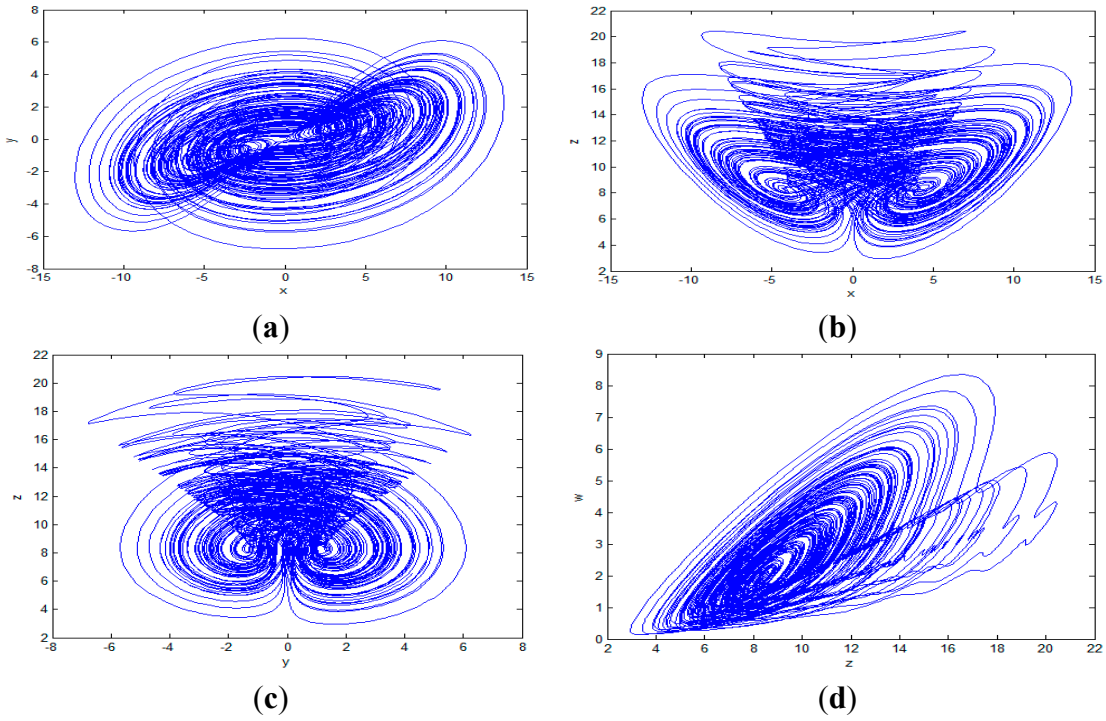
Obviously, the maximal Lyapunov exponent of the proposed hyperchaotic system is bigger than that of other systems. So the proposed system has better dynamic characteristics.

2.2. Dynamic Behavior Analysis of New Chaotic System

2.2.1. Dissipation and Existence of Hyperchaotic Attractor

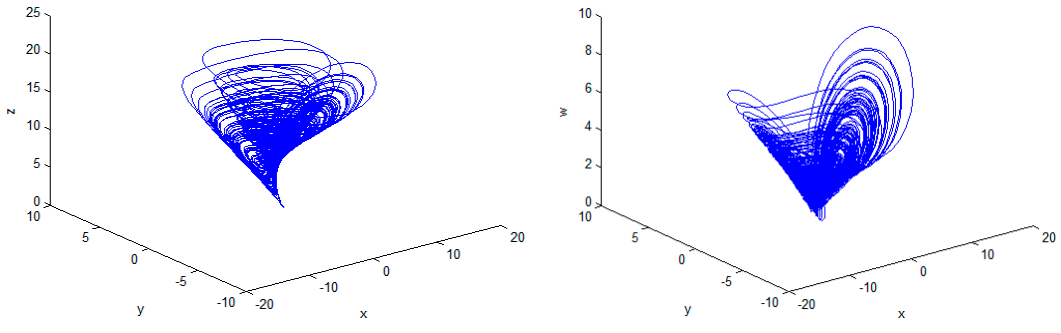
In Equation (2), when $a = 4$, $b = -0.5$, $d = 1$, $h = 8.1$ and $c = -2.24$, there is $\nabla V = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} + \frac{\partial \dot{w}}{\partial w} = -a - b - d + c = -6.7 < 0$. That is, the system is a dissipative system. For the volume element of which initial volume is V_0. When $t \to \infty$, it exponentially converges to 0 with the rate, $a + b + d - c$, along the system track. That is, the track curve of the system will eventually be fixed to an attractor. When $a = 4$, $b = -0.5$, $d = 1$, $h = 8.1$, $c = -2.2$ and the initial value is [0.8, 0.3, 10.1, 4.5], two dimensional phase planes of the chaotic attractors of the hyperchaotic Rabinovich system is shown in Figure 1.

Figure 1. Phase planes of new Rabinovich system. (**a**) *x*-*y* phase plane; (**b**) *x*-*z* phase plane; (**c**) *y*-*z* phase plane; (**d**) *z*-*w* phase plane.



It is shown that the proposed Rabinovich map has clear strange attractor in two-dimensional planes. Numerical simulation proves the existence of the chaotic attractor. The *x*-*y*-*z* phase planes of the chaotic attractors of the proposed Rabinovich system is shown in Figure 2a. The *x*-*y*-*w* phase planes of the chaotic attractors are shown in Figure 2b. From Figures 1 and 2, we can see the strange attractors of the proposed system clearly.

Figure 2. (a) *x*-*y*-*z* phase plane; **(b)** *x*-*y*-*w* phase plane.



2.2.2. Equilibrium and Stability

When $a = 4$, $b = -0.5$, $d = 1$, $h = 8.1$, $c = -2.2$, set $\dot{x} = \dot{y} = \dot{z} = \dot{w} = 0$, we can find the equilibrium points of the system. In the equilibrium point $P_0(0,0,0,0)$, Jacobi matrix is:

$$J_0 = \begin{bmatrix} -a & h & 0 & 0 \\ h & -b & 0 & 0 \\ 0 & 0 & -d & 0 \\ 0 & 0 & 0 & -c \end{bmatrix}. \quad (4)$$

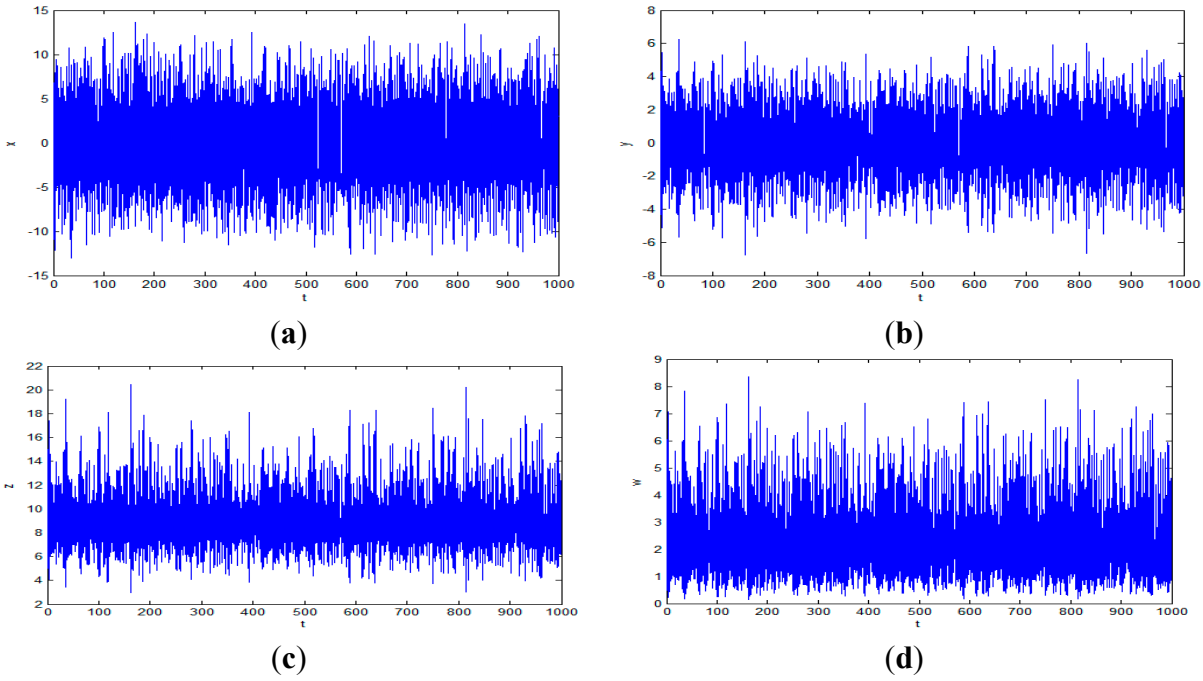
The eigenvalues of the Jacobi matrix are −2.2, −1, 76.775 and −80.275. If there is at least one eigenvalue of the Jacobi matrix of which the real part is greater than 0, then the equilibrium state is unstable. Calculations shows that the equilibrium point $P_0(0,0,0,0)$ is unstable. For the remaining eight equilibrium points, each equilibrium point has at least one eigenvalue of which the real part is greater than 0. Therefore, the equilibrium points are all unstable.

2.2.3. Non-periodic Flow

Using fourth-order Runge-Kutta method, we can get the chaotic sequences of the proposed system with step size 0.01. When $a = 4$, $b = -0.5$, $d = 1$, $h = 8.1$ and $c = -2.2$, we can get time responses of the four variables. The four variables of the proposed system change with time t are shown in Figure 3. As can be seen from this figure, each variable changes on a certain range over time. Obviously, the change of variable shows a disorderly and unsystematic characteristic, not a periodic characteristic.

Theoretical analysis and numerical simulation show that the proposed system has the following characteristics: qualified dimensionality is greater than or equal to 4; there are two equations at least which have at least one nonlinear term; it has dissipative structure and two positive Lyapunov exponents; equilibrium points are unstable; strange attractors can be observed clearly on three dimensional phase planes. Therefore, the proposed system is a hyperchaotic system.

Figure 3. State variables of proposed system change with time t. (**a**) Change of parameter x; (**b**) Change of parameter y; (**c**) Change of parameter z; (**d**) Change of parameter w.



2.3. Exponential Chaos Algorithm

A low-dimensional chaotic map is just a simple iteration. Its iterative process is too singular, with less initial values and parameters. It has a stable periodic window, but the chaotic sequence generation speed is faster than in a hyperchaotic system, so we combine the proposed hyperchaotic map and the one-dimensional exponential chaos map together to generate key-stream sequences to get better performance. The exponential chaos map [17] is shown as follows:

$$x_{n+1} = index^{x_n} (\text{mod} \quad 1). \tag{5}$$

where the variable $x \in [0, 1]$ and the parameter *index* is an arbitrary real number greater than 1.

Let *index* = 200, x_0 = 0.05 and the iteration number be 10,000. The chaotic sequence values generated by the exponential chaotic map are shown in Figure 4. Moreover, the sequence is even-distributed in the interval [0, 1]. The histogram of the sequence distribution is shown in Figure 5.

Figure 4. Chaotic sequence values of exponential chaotic map.

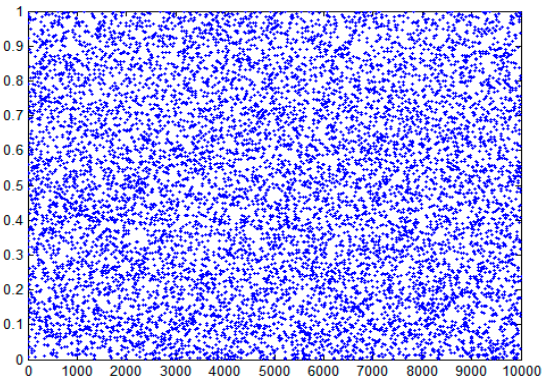
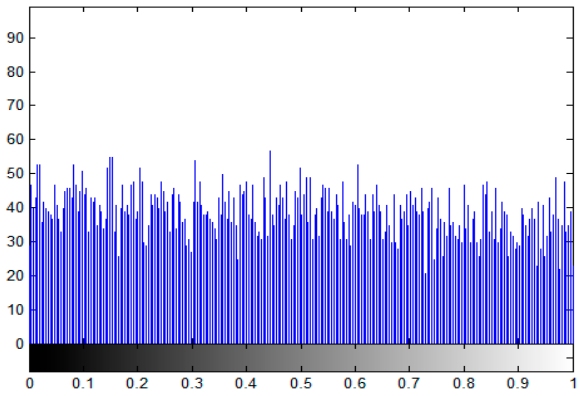


Figure 5. Histogram of the exponential chaotic sequence.



3. Design of Pseudo-random Sequence Based on Hyperchaotic Map

In order to improve encryption security, the pseudo-random sequence is designed affected by plaintext:

(1) Chaotic sequences preprocessing

Let the four chaotic sequences obtained by the hyperchaotic map be $\{x_n\}$, $\{y_n\}$, $\{z_n\}$ and $\{w_n\}$, respectively. The sensitivity of chaotic trajectories to the initial conditions is the typical characteristic of chaos. It is also a cause of applying chaos to encryption, but some values at the beginning don't meet the sensitivity to the initial conditions, so to get better randomness, the previous N_0 numbers of the four sequences are discarded. The new sequences are denoted still as $\{x_n\}$, $\{y_n\}$, $\{z_n\}$ and $\{w_n\}$, respectively. Here we set $N_0 = 100$.

(2) Determining the size of chaotic sequence

The plain image is cut into 256×256 blocks. Then the size of each chaotic sequence $\{\tau_n\}$ ($\tau = x, y, z, w$) need to be $256 \times 256/4$, so the total length of the sequence for each block is 256×256 when the four sequences are put together. When the size of the plain image is $M \times N$, then the final size of the chaotic sequence is $M \times N$. The sequence obtained in this stage is denoted by $\{z_n\}$ ($n = 1, 2, M \times N$).

(3) Chaotic sequence normalization

Normalize the four chaotic sequences as follows:

$$z'_n = \frac{z_n - z_{\min}}{z_{\max} - z_{\min}} \tag{6}$$

where $z_{\max}$ denotes the maximum value and the minimum value of the sequence $\{z_n\}$, respectively. The sequence obtained in this stage is denoted by $\{z'_n\}$.

(4) Exponential chaos processing

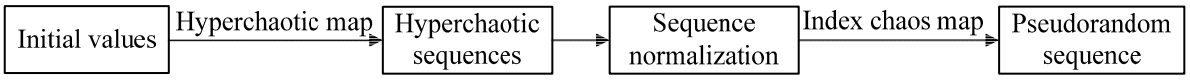
The values in the sequence $\{z'_n\}$ are put into the exponential chaos map in Equation (5). Here we set the parameter *index* = 200 in Equation (5). This parameter, *index*, is used as a key. The iteration number, *itera*, of each value in the sequence $\{z_n\}$ is:

$$itera = ceil(50 \times z'_n) + 10, \tag{7}$$

where $ceil(\cdot)$ is the rounded up function.

At this point, we get a pseudo-random sequence, *Rand*. The size of this sequence is $M \times N$. And this sequence values in [0, 1]. The generation process of the pseudo-random sequence, *Rand*, is shown in Figure 6:

Figure 6. Pseudo-random sequence generation process.



4. Design of Image Encryption Algorithm Based on Hyperchaotic Map

4.1. Encryption Algorithm

Image encryption schemes are usually implemented in two steps: the first step is image scrambling, and the second step is pixel substitution. Image scrambling is carried out in the spatial domain. An image can be described by the position and the pixel value. Image scrambling is to change the position

relationship between the pixels in the image. It changes the image from an original digital image into a noise-polluted image. Pixel substitution is implemented by XOR or other operation with other sequence or matrix to change original image pixel value.

(1) Image scrambling

In image scrambling stage, the chaotic sequence used in scrambling is selected from the pseudo-random sequence *Rand* generated by the pseudo-random sequence generator.

(A) Plain image preprocessing

Let the size of the plain image be $Mini \times Nini$. The plain image is cut into 256×256 blocks. The shortfall is complemented with pixel value 1. Let the size of the complemented plain image be $M \times N$. Then the size of the cipher image is $M \times N$ too. So the size of the original image, $Mini \times Nini$, needs to be sent to the receiver. Thus the receiver can know which are the plain pixels and which are the padded pixels. The following operations are implemented in each 256×256 block.

(B) Scrambling in block

Scrambling in block is carried out as follows: fetch the E different values $\{C_i | i = 0, 1, 2,\ldots, E - 1\}$ from the chaotic sequence sequentially. Sort the sequence $\{C_i | i = 0, 1, 2,\ldots, E - 1\}$, and we get the sorted sequence $\{P_i | i = 0, 1, 2,\ldots, E - 1\}$ and an index sequence $\{T_i | i = 0, 1, 2,\ldots, E - 1\}$, where the symbol T_i is the position index of which C_i is in P. According to the index sequence T, moves the T_i row to the i-th row. Then take the F different values $\{D_j | j = 0, 1, 2,\ldots, F - 1\}$ from the chaotic sequence sequentially. Sort the sequence $\{D_j | j = 0, 1, 2,\ldots, F - 1\}$, and we get the sorted sequence $\{Q_j | j = 0, 1, 2,\ldots, F - 1\}$ and an index sequence $\{S_j | j = 0, 1, 2,\ldots, F - 1\}$, where the symbol S_j is position index of which D_j is in Q. According to the index sequence S, move the S_j column to the j-th column. Column scrambling is carried out.

(C) Chaos value selection rule

The selected chaotic sequences used to sort are the same sequences, so the selection rule is important. To improve the security, the selection rule is designed to relate to the plain image. The first value, *csd*1, selected from the chaotic sequence used to scramble the rows of the image is obtained as follows:

$$csd1 = \mathrm{mod}[P(1,1) \times 100, P(ceil(M / 2), ceil(N / 2))] + 5 \tag{8}$$

where $P(i, j)$ denotes the pixel value of the plain image at the position (i, j) and $ceil(\cdot)$ indicates the rounded up function.

The first value, *csd*2, selected from the chaotic sequence used to scramble the columns of the images is obtained as follows:

$$csd2 = \mathrm{mod}[P(2,2) \times 100, P(ceil(M / 4), ceil(N / 4))] + 5 \tag{9}$$

Both *csd*1 and *csd*2 are used as the keys. To make the receiver able to decrypt the cipher image, *csd*1 and *csd*2 must be sent to the receiver. After scrambling, we get the resulting image M.

(2) Pixel value substitution

(A) Pseudo-random sequence pretreatment

The pseudo-random sequence generator produces a pseudo-random sequence *Rand*. The values in the sequence belong to the interval [0, 1]. Then the pseudo-random sequence *Rand* is transformed into a new sequence *Rand*Image used in pixel substitution:

$$Rand\text{Image} = fix((1000 \times Rand(i)) \bmod 256) \tag{10}$$

In Equation (9), $fix(\cdot)$ indicates the truncating toward zero.

(B) Pixel value substitution process

We also need the custom key *MC* used to encrypt the pixel values at the position (1,1) in the scrambled image *M*, from which the new value generated is used in diffusion. Equation (11) shows that:

$$C(1) = (M(1,1) + Rand\text{Image}(1) + MC) \bmod 256 \tag{11}$$

In Equation (11), $M(1,1)$ is a pixel value of each sub-block after scrambling at the position (1,1).

When $i \geq 2$, the converting Equation (12) is as follows:

$$C(i) = (M(ki,kj) + Rand\text{Image}(i) + C(i-1)) \bmod 256 \tag{12}$$

where $M(ki,kj)$ is a pixel value of the image *M* in the *i*-th position after scrambling which is sorted to a one-dimensional array based on the order of row and line. The resulting array *C* is arranged into a matrix which is the final cipher-image.

4.2. Decryption Algorithm

All the keys are plugged into the proposed hyperchaotic system and the exponential chaos map to get the sequence *Rand* used in scrambling and the sequence *Rand*Image used in substitution.

According to the encryption algorithm, the decryption algorithm is shown below:

$$\begin{cases} M(1,1) = (C(1) - MC - Rand\text{Image}(1)) \bmod 256, & i = 1 \\ M(ki,kj) = (C(i) - C(i-1) - Rand\text{Image}(i)) \bmod 256, & i \geq 2 \end{cases} \tag{13}$$

where $C(i)$ is a pixel value of cipher-image in the *i*-th position after scrambling which is sorted to a one-dimensional array based on the order of row and line. By taking this step, we can get the image *M* after scrambling. Next, the padding out plain image is obtained by making *M* and pseudo-random sequence *Rand* used in scrambling to sort in reverse order. According to the size of the original image, we get rid of the redundant pixels to get the plain image.

5. Encryption Test and Security Analysis

The following tests are realized by MATLAB software on an Intel Core 2 Duo 2.4 GHz PC.

5.1. Encryption Test

The first original image as shown in Figure 7a in experiment is the well-known Lena image. The size of the image is 256×256, the grayscale is $L = 256$. The encryption result is displayed as Figure 7b.

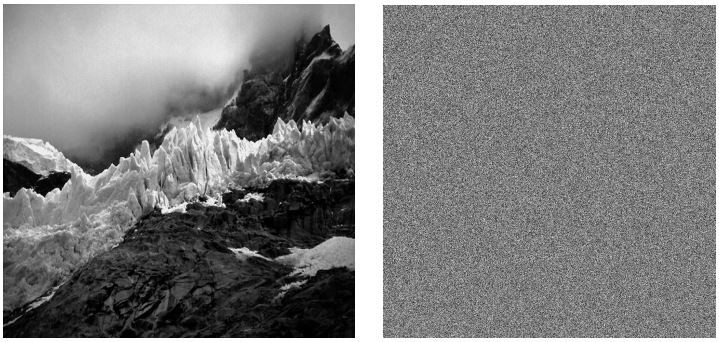
The second original image is the Jokul image, as shown in Figure 8a. Its size is 900×960, the grayscale is $L = 256$. The encryption result is displayed as Figure 8b. From the encryption results as shown in Figure 7b and Figure 8b, we cannot obtain any information about the plain-images. That is,

the algorithm passes the subjective test. The appraisal method of image encryption consists of a subjective test and an objective measuring test. Here, we use the subjective test to measure the encryption effect.

Figure 7. (**a**) Lena image before encryption; (**b**) Lena image after encryption.

(**a**) (**b**)

Figure 8. (**a**) Jokul image before encryption; (**b**) Jokul image after encryption.



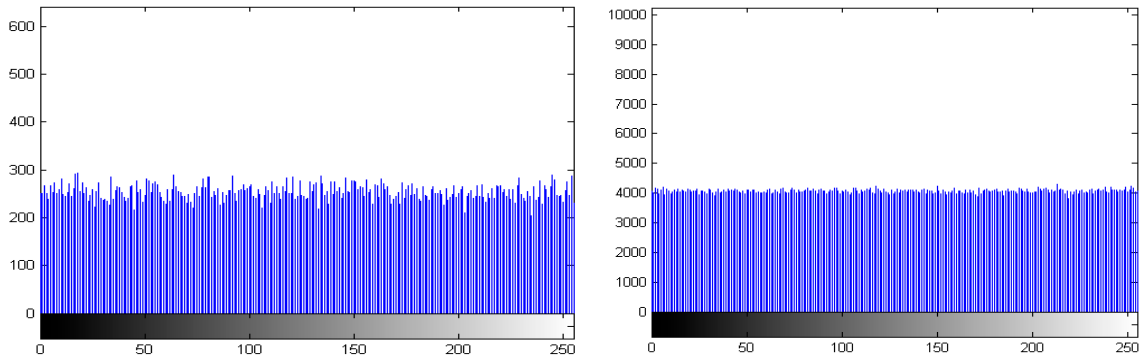
(**a**) (**b**)

5.2. Security Tests

(1) Histogram analysis

The histograms of the encrypted images are shown in Figure 9a and Figure 9b, respectively. From the results, we can see that the pixel distribution of the cipher-images is fairly uniform, which can greatly reduce the correlation between the pixel values.

Figure 9. (**a**) Histogram of cipher-image of Lena; (**b**) Histogram of cipher-image of Jokul.



(**a**) (**b**)

(2) NIST SP 800-22 Tests

In 2003, the United States National Institute of Standards and Technology issued "Special Publication 800-22" (SP 800-22) for cryptographic random and pseudorandom number statistical tests. This is one of the most extensively inspection standards by far. It is well-known that NIST SP 800-22 tests are applied for 0-1 sequences, so the cipher-image can be regarded as a binary data stream file. Thus the cipher-image is tested as 0-1 sequences. The test results of the cipher-image of Lena are shown in Table 2.

Table 2. NIST test results.

Test Item	**P-Value**	**Result**
Frequency Test	0.1455	Pass
Frequency Test within a Block (m = 12,000)	0.2095	Pass
Runs Test	0.4811	Pass
Test for the Longest Run of Ones in a Block (M = 10,000，N = 100)	0.3943	Pass
Binary Matrix Rank Test	0.8510	Pass
Discrete Fourier Transform Test	0.7120	Pass
Non-overlapping Template Matching Test (m = 8)	0.4064	Pass
Overlapping Template Matching Test (m = 8, M = 65,536)	0.0459	Pass
Maurer's Test (L = 6，Q = 640，K = 86,741)	0.8468	Pass
Linear Complexity Test (N = 256)	0.9018	Pass
Serial Test	0.2110, 0.8431	Pass
Approximate Entropy Test (m = 8)	0.8184	Pass
Cumulative Sums Test (Positive)	1.5922	Pass
Cumulative Sums Test (Reverse)	1.5936	Pass
Random Excursions Test	0.3757, 0.8900, 0.5554, 0.0838, 0.1223, 0.5888, 0.8276, 0.7656	Pass
Random Excursions Variant Test	0.3819, 0.4292, 0.5932, 0.6232, 0.7172, 0.9851, 0.8424, 0.3047, 0.2560, 0.2773, 0.2786, 0.2600, 0.1978, 0.3740, 0.6551, 0.8911, 0.8483, 0.7193	Pass

As can be seen from Table 2, the cipher-text sequence can pass all the tests. It can be said that the cipher-text sequence is a pseudo-random sequence. The randomness of the cipher-image is good.

(3) Key space analysis

The key space for a good encryption scheme should be big enough to resist brute-force attacks. From Figures 3–7, the point sequence in each dimension will eventually return to a safe range and this range is the suggested range of initial value. The range of the sequence $\{x_n\}$ is about [−10, 10], the range of the sequence $\{y_n\}$ is about [−6, 6], the range of the sequence $\{z_n\}$ is about [4, 20], the range of the sequence $\{w_n\}$ is about [0, 8]. We assume the key spaces of the four variables are K_1, K_2, K_3 and K_4, respectively. Then the total key space of this hyperchaotic system is $K_1 * K_2 * K_3 * K_4$. In addition, there are other two key parameters, namely, *csd*1 and *csd*2 used in scrambling, the custom key *MC* and

the parameter *index* in the exponential chaos map. The custom key *MC* is a positive integer. The parameter *index* is greater than 1, so the key space of the proposed scheme is big enough to resist the brute-force attacks.

(4) Differential attack analysis

The attacker may seek to observe variations of the ciphertext in the tiny variations of the plaintext to find the correlation between the plaintext and the ciphertext. If a tiny change in the original image can lead to a great change in the cipher image, then the algorithm can effectively resist these differential attacks. Generally, the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) can be used to describe the ability to resist the differential attack. Their definitions are as follows:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%, \tag{14}$$

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\%, \tag{15}$$

where W and H are the width and height of the image, respectively. $C_1(i, j)$ and $C_2(i, j)$ are the corresponding pixels of two images. If $C_1(i, j) = C_2(i, j)$, then $D(i, j) = 0$, otherwise $D(i, j) = 1$. The ideal values of NPCR and UACI are 99.61% and 33.46%, respectively.

A pixel is selected randomly from the original image. The corresponding ciphertexts of this new plain image and the original image can be obtained by the proposed algorithm, respectively. In this way, 500 tests are implemented and the corresponding values of NPCR and UACI can be obtained. Thus we can get the average values of NPCR and UACI. The results are shown in Table 3.

Table 3. NPCR and UACI values.

Image	NPCR	UACI
Lena	99.49%	33.32%
Jokul	99.44%	33.28%

From Table 3, we can see that the values of NPCR and UACI are close to the ideal values. It means that one bit difference of the plain image can diffuse to the whole cipher image, and we can conclude that the algorithm can resist differential attacks.

(5) Key sensitivity analysis

A secure encryption scheme should be sensitive to the key. Very tiny differences between the initial values will lead to the completely different cipher-images. In this test, the chaotic sequences of the proposed system are obtained by the fourth-order Runge-Kutta algorithm with step size 0.01. The results are shown in Figure 10. Figure 10(a) and Figure 10(b) show the cipher images C_1 and C_2 with the keys K_1 [0.8, 0.3, 10.1, 4.5] and K_2 [0.8 + 0.00000000001, 0.3, 10.1, 4.5], respectively. Figure 10c shows the decrypted result of C_1 with the right key K_1. Figure 10d shows the decrypted result of C_1 with the wrong key K_2. In spite of tiny difference between K_1 and K_2, C_1 cannot be decrypted correctly.

Figure 10. (**a**) Cipher image C_1 with K_1; (**b**) Cipher image C_2 with K_2; (**c**) Decrypted result of C_1 with K_1; (**d**) Decrypted result of C_1 with K_2.

(**a**) (**b**) (**c**) (**d**)

To evaluate the key sensitivity further, we test the values of NPCR and UACI between C_1 and C_2, respectively. As an example, Table 4 shows the test results of the image "Lena".

Table 4. NPCR and UACI values.

Image	NPCR	UACI
Lena	99.37%	33.45%

Based on the above analysis, it can be concluded that the proposed encryption scheme is sensitive to the key.

(6) Information entropy analysis

Image information entropy can measure the distribution of image gray values. The more uniform the gray value distribution is, the bigger the information entropy is. The less information of the original image can be obtained from the gray value distribution of the cipher-image by the attacker, the higher security the encryption algorithm has. Image information entropy is defined as:

$$H = -\sum_{i=1}^{256} p_i \log p_i , \tag{17}$$

where p_i is the probability of the gray value.

The ideal value of the cipher information entropy is 8. The information entropy of the cipher-image for Lena generated by the proposed algorithm is 7.9893. The information entropy of the cipher-image for Jokul generated by the proposed algorithm is 7.9920. They are both close to the ideal value, so to the ciphertext attackers, the cipherimage pixels are statistically independent of each other, so it is difficult to decrypt the cipertext.

5.3. Analysis of Chaotic Maps with Others

The maximum Lyapunov exponent of Rabinovich system is 0.1459, while the maximum Lyapunov exponent of the new hyperchaotic Rabinovich system is 1.090046. The bigger the Lyapunov exponent is, the faster the trajectories separate and the wider the corresponding separatrix of the chaotic region is, so the dynamic behaviors of the new system are better than that of the original system.

The typical one-dimensional chaotic system is the Logistic map, it is:

$$x_{k+1} = \mu x_k (1 - x_k), \tag{18}$$

where $3.569 \le \mu \le 4.0$, $x_k \in (0, 1)$.

In addition, we choose Cube map as a reference, it is:

$$x_{k+1} = \lambda x_k - x_k^3, \tag{19}$$

where $2.59 \le \lambda \le 3.0$, $x_k \in [-2, 2]$.

In our design, the total number of target points of chaotic map is 256×256. The following experiment is processed on computer with 2.7 GHz CPU, 2 GB memory with Windows XP operation system. We generate 256×256 points sequences using the proposed hyperchaotic system, the Logistic map and the cube map, respectively. The run times are shown in Table 5.

Table 5. Comparison of chaotic sequence generation speed.

Chaos map	Generate sequence time (s)
Proposed hyperchaotic map	0.3419
Logistic map	3.1559
Cube map	3.1520

Table 5 shows that the speed of the proposed map is faster than several low-dimensional chaotic systems. Appling the chaotic sequence generated by the proposed system to encrypt images is more efficient.

6. Conclusions

In this paper, we construct a new four-dimensional hyperchaotic system by adding a nonlinear term to the Rabinovich system. Then we analyze the basic dynamic characteristics of the proposed system. The chaotic sequences of the proposed system are generated by the Runge-Kutta method. These sequences are put into the exponential chaos map to generate the key sequence. A selection rule related to the plain image is designed to select the different sub-segment from the key sequence. The key sequence is associated with the plaintext to scramble and diffuse the pixels of the plain image. The security analysis, including histogram, randomness, information entropy and key sensitive, shows that the proposed system has good security and complexity. Moreover, the key space is big enough to resist the brute-force attack. The comparison with several low-dimensional chaotic systems shows that the system has more efficiency.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (60973162), the Natural Science Foundation of Shandong Province of China (ZR2009GM037, ZR2014FM026), the Science and Technology of Shandong Province, China (2013GGX10129, 2010GGX10132, 2012GGX10110), the Soft Science of Shandong Province, China (2012RKA10009), the National Cryptology Development Foundation of China (No. MMJJ201301006), Foundation of Science and

Technology on Information Assurance Laboratory (No. KJ-14-005) and the Engineering Technology and Research Center of Weihai Information Security.

Author Contributions

Xiaojun Tong, Yang Liu and Miao Zhang conceived and carried out the project. Hui Xu and Zhu Wang wrote the manuscript. All authors have read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Wang, X.; Wang, Q. A novel image encryption algorithm based on dynamic S-boxes constructed by chaos. *Nonlinear Dyn.* **2014**, *75*, 567–576.
2. Hu, H.; Liu, L.; Ding, N. Pseudorandom sequence generator based on the Chen chaotic system. *Computer Phys. Commun.* **2013**, *184*, 765–768.
3. Wang, X.; Zhang, W.; Guo, W.; Zhang, J. Secure chaotic system with application to chaotic ciphers. *Inf. Sci.* **2013**, *221*, 555–570.
4. Qi, G.Y.; Du, S.; Chen, G.R.; Chen, Z.; Yuan, Z. On a four-dimensional chaotic system. *Chaos Solitons Fractals* **2005**, *23*, 1671–1682.
5. Rossler, O.E. An equation for hyperchaos. *Phys. Lett. A* **1979**, *71*, 155–157.
6. Si, G.Q.; Cao, H.; Zhang, Y.B. A new four-dimensional hyperchaotic Lorenz system and its adaptive control. *Chin. Phys. B* **2011**, *20*, 229–237.
7. Zhu, C. A novel image encryption scheme based on improved hyperchaotic sequences. *Optics Commun.* **2012**, *285*, 29–37.
8. Jia, L.X.; Dai, H.; Hui, M. A new four-dimensional hyperchaotic Chen system and its generalized synchronization. *Chin. Phys. B* **2010**, *19*, 125–135.
9. Deng, Y.; Wang, G.Y.; Yuan, F. A new image encryption algorithm based on improved Rossler chaotic sequence. *J. Hangzhou Dianzi Uni.* **2011**, *31*, 9–12.
10. Shen, M.; Liu, J. A new hyperchaotic system and its circuit simulation by electronic workbench. *J. Chongqing Uni. Posts Telecommun.* **2010**, *22*, 71–74.
11. Pang, S.Q.; Liu, Y.J. A new hyperchaotic system from the Lü system and its control. *J. Comput. Appl. Math*. **2011**, *235*, 2775–2789.
12. Huang, S.H.; Tian, L.X. Dynamical analysis and anti-synchronization for a new four dimensional hyperchaotic system. *J. Circuits Syst.* **2011**, *16*, 74–76.
13. Zhang, W.Q. Analysis of Properties of Qi Unified Hyperchaotic System and Its Realization. *J. Nanjing Inst. Technol.* **2011**, *9*, 1–5.
14. Jian, J.G.; Deng, X.L.; Wang, J.F. Globally Exponentially Attractive Set and Synchronization of a Class of Chaotic Finance System. In *Advances in Neural Networks—ISNN 2009*, Proceedings of 6th International Symposium on Neural Networks, Wuhan, China, 26–29 May 2009; Lecture Notes in Computer Science; Volume 5551; Springer: Berlin/Heidelberg, Germany, 2009; pp. 253–261.

15. Pikovski, A.S.; Rabinovich, M.I.; Trakhtengerts, V.Y. Onset of stochasticity in decay confinement of parametric instability. *Soviet Phys. JEPT* **1978**, *47*, 715–719.
16. Wang, J.Z.; Chen, Z.Q.; Yuan, Z.Z. The generation of a hyperchaotic system based on a three-dimensional autonomous chaotic system. *Chin. Phys.* **2006**, *15*, 1216–1225.
17. Li, C.G.; Han, Z.Z.; Zhang, H.R. A sequence based on the gray Exponential Chaotic Image Encryption Algorithm. *Comput. Eng. Appl.* **2002**, *38*, 16–17.