Journal of Applied Analysis and Computation
Volume 5, Number 2, May 2015, 197–209
Website:http://jaac-online.com/
doi:10.11948/2015018

A METHOD FOR IMAGE ENCRYPTION BASED ON FRACTIONAL-ORDER HYPERCHAOTIC SYSTEMS*

Jianbin He[1,†], Simin Yu[1] and Jianping Cai[2]

Abstract By using sequences generated from fractional-order hyperchaotic systems, a color image encryption scheme is investigated. Firstly, a plain image, which is known to users in advance, is chosen as a secret key to confuse the original image. Then, the confused image is encrypted by the sequences generated from the fractional-order hyperchaotic systems. With this simple encryption method, we can get an encrypted image that is fully scrambled and diffused. For chaos-based image cryptosystems, this encryption scheme enhances the security and improves the effectiveness. Furthermore, the cryptosystem resists the differential attack. Experiments show that the algorithm is suitable for image encryption, and some statistical tests are provided to show the high security in the end.

Keywords Fractional order, hyperchaotic system, image encryption, security analysis.

MSC(2000) 34C28, 94A60, 68P30.

1. Introduction

Chaos is a very interesting phenomenon in nonlinear science, thanks to its sensitivity to initial values and parameters. In 1963, the chaotic Lorenz system was proposed, then many chaotic systems and hyperchaotic systems were investigated by scholars, such as the Rössler system, Lü system, Chen system, high-dimensional hyperchaotic systems, and multi-wing hyperchaotic systems [6,7,17]. In recent years, chaos-based encryption has attracted much attention due to its potential advantage of good confusion and diffusion properties in commercial, military and medical applications.

Methods of chaos-based image encryption have been widely studied in the past. As we know, the ciphered images should be greatly different from the original images, for example, the pixel values are changed pseudo-randomly and the correlation coefficient drops near to zero. Determining whether a system can have good resistance to attacks, it can be measured by means of NPCR (number of pixels change rate) and UACI (unified average changing intensity) [1]. One of the key points

†The corresponding author. Email address: jbh2012yml@126.com (J. He)
[1]College of Automation, Guangdong University of Technology, 510006 Guangzhou, China
[2]College of Mathematics and Statistics, Minnan Normal University, 363000 Zhangzhou, China
*This work was supported by the National Natural Science Foundation of China (Grant No: 61172023) and the Natural Science Foundation of Fujian Province (Grant No: 2014J01018).

in image encryption is to change the positions or pixel values based on some low-dimensional chaotic systems, such as the cat map and the logistic map, which are used to generate pseudo-random sequences. Accordingly, encryption based on hyperchaotic systems will be a typical and useful method due to its good results. Mazloom et al. [10] proposed a novel chaos-based image encryption algorithm to encrypt color images by using a coupled nonlinear chaotic map. For getting higher security and higher complexity, they employ the image size and color components into the cryptosystem, thereby significantly increasing the resistance to known and chosen plaintext attacks. Liu and Wang [8] designed a triple color image encryption scheme based on chaos where the Lorenz system was employed to generate four pseudo-random sequences, and the 256-bit hash value came from three images was applied to produce the initial values and parameters for the Lorenz system. In [15], a new color image encryption algorithm was investigated based on the chaotic sequences generated from two fractional-order hyperchaotic systems.

However, recent cryptanalysis works have demonstrated that some chaos-based image cryptosystems are insecure against various attacks, and have been broken easily [9, 13]. The weaknesses in these insecure algorithms include insensitiveness to the changes of the plain image and weak secret keys. Compared to integer-order chaotic systems, the dynamics of fractional-order chaotic systems are more complex, because fractional derivatives have complex geometrical interpretation for their nonlocal character and high nonlinearity [12, 15]. In addition, the derivative order of fractional-order chaotic systems can be used as secret keys. Based on a combination of fractional-order hyperchaotic systems, we propose a method of image encryption in this paper.

Specially, a plain image is employed to scramble the pixel values of the original image. Then, the scrambled image is encrypted once again by using sequences generated from the combination of fractional-order hyperchaotic systems. Similarly, the decryption algorithm is the reverse of the encryption. The encrypted image can be deciphered successfully when the user obtains the correct keys. In the end, the results of several experiments show that the scheme of image encryption is effective, and some security tests are provided to show its high security for image encryption and transmission.

The paper is organized as follows. In Section 2, the definition of fractional-order derivative and some fractional-order hyperchaotic systems are introduced. The proposed encryption scheme is investigated in Section 3. Experimental results are performed in Section 4. Some security analyses are performed and discussed in Section 5. Finally, Conclusions are drawn in Section 6.

2. Systems description

In general, hyperchaotic systems have more complex behavior. So, fractional-order hyperchaotic systems are employed to design the cryptosystem. Riemann-Liouville definition and the predictor-corrector algorithm are used in computing the fractional-order differential equations. There are several definitions of fractional derivatives [3, 11], and the Riemann-Liouville definition is given by

$$\frac{d^\alpha f(t)}{dt^\alpha} = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_0^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau,$$

where $\Gamma(\bullet)$ is the gamma function, and $\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt, n-1 \le \alpha \le n$.

The fractional-order hyperchaotic Lü, Lorenz and Chen systems are combined to generate pseudo-random sequences. In this subsection, they will be described separately. The hyperchaotic Lü equation can be described as follows [2]:

$$\begin{cases} \dfrac{d^\alpha x}{dt^\alpha} = a_1(y-x)+z, \\ \dfrac{d^\alpha y}{dt^\alpha} = -xz+c_1 y, \\ \dfrac{d^\alpha z}{dt^\alpha} = -b_1 z+xy, \\ \dfrac{d^\alpha w}{dt^\alpha} = d_1 w+xz, \end{cases} \tag{2.1}$$

where x, y, z, w are the state variables and a_1, b_1, c_1, d_1 are the parameters. It has a hyperchaotic attractor when $a_1 = 36$, $b_1 = 3$, $c_1 = 20$, $d_1 = -0.4$, $\alpha = 0.95$. The hyperchaotic attractor is shown in Figure 1.

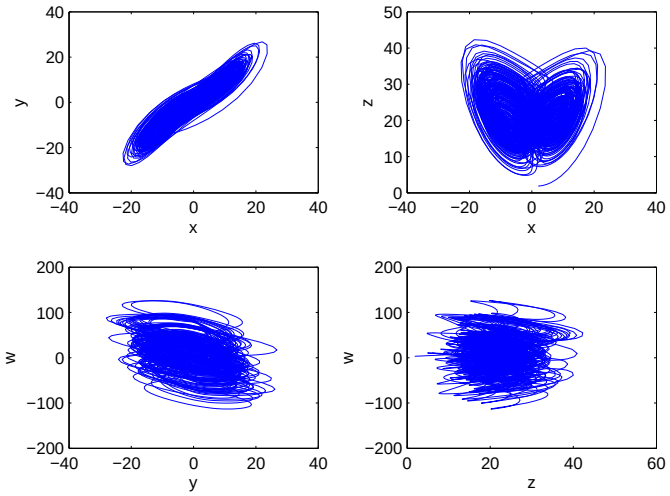


Figure 1. The attractor of the hyperchaotic Lü system.

Similarly, the fractional-order hyperchaotic Lorenz system is given by

$$\begin{cases} \dfrac{d^\beta x}{dt^\beta} = a_2(y-x)+w, \\ \dfrac{d^\beta y}{dt^\beta} = -xz+c_2 x-y, \\ \dfrac{d^\beta z}{dt^\beta} = -b_2 z+xy, \\ \dfrac{d^\beta w}{dt^\beta} = r_2 w-xz, \end{cases} \tag{2.2}$$

where $a_2 = 10, b_2 = 8/3, c_2 = 28, d_2 = -1, \beta = 0.95$, and system (2.2) exhibits a hyperchaotic behavior, as described in [16]. Furthermore, the fractional-order

hyperchaotic Chen system is in the form of

$$\begin{cases} \dfrac{d^\gamma x}{dt^\gamma} = a_3(y-x) + w, \\ \dfrac{d^\gamma y}{dt^\gamma} = -xz + b_3 x - c_3 y, \\ \dfrac{d^\gamma z}{dt^\gamma} = -d_3 z + xy, \\ \dfrac{d^\gamma w}{dt^\gamma} = r_3 w + xz, \end{cases} \tag{2.3}$$

and, when $a_3 = 35, b_3 = 7, c_3 = 12, d_3 = 0.5, \gamma = 0.95$, it has a hyperchaotic attractor [4].

3. Encryption scheme based on the combination of fractional-order hyperchaotic systems

3.1. Generation of the variable initial values and parameters

Chaotic systems are sensitive to the initial values and parameters. In order to improve the security of a designed scheme, the key point is to keep the initial values dynamic, which associates to the original image.

SHA-256 is a widely used cryptographic hash function with 256-bit hash value in cryptography [8]. The 256-bit hash value generated from the original image is divided into 8-bit blocks k_i $(i = 1, 2, \ldots, 32)$, so they are used to regenerate the initial values and parameters. For example, the new initial values and parameters are given as follows:

$$\begin{cases} x_0^{'} = x_0 + \dfrac{k_1 \oplus k_2 \oplus k_3 \oplus k_4}{256}, & y_0^{'} = y_0 + \dfrac{k_5 \oplus k_6 \oplus k_7 \oplus k_8}{256}, \\ z_0^{'} = z_0 + \dfrac{k_9 \oplus k_{10} \oplus k_{11} \oplus k_{12}}{256}, & w_0^{'} = w_0 + \dfrac{k_{13} \oplus k_{14} \oplus k_{15} \oplus k_{16}}{256}, \\ a_0^{'} = a_0 + \dfrac{k_{17} \oplus k_{18} \oplus k_{19} \oplus k_{20}}{256}, & b_0^{'} = b_0 + \dfrac{k_{21} \oplus k_{22} \oplus k_{23} \oplus k_{24}}{256}, \\ d_0^{'} = d_0 + \dfrac{k_{25} \oplus k_{26} \oplus k_{27} \oplus k_{28}}{256}, & r_0^{'} = r_0 + \dfrac{k_{29} \oplus k_{30} \oplus k_{31} \oplus k_{32}}{256}, \end{cases} \tag{3.1}$$

where $x_0^{'}, y_0^{'}, z_0^{'}, w_0^{'}, a_0^{'}, b_0^{'}, d_0^{'}, r_0^{'}$ are the new initial values and parameters, respectively.

3.2. Design of the encryption scheme

In this subsection, a color image encryption scheme is investigated. A color image could be decomposed into three colors (RGB). Assume that the image size is $w \times h$ and the value of each pixel is an integer in the interval $[0, 255]$.

First, a plain image ('I') is employed to scramble the original image ('O') by the exclusive OR operation instead of encrypting an image using a chaotic sequences directly. The detailed encryption algorithm is described as follows:

Step 1. There are a variety of options on image ('I'), but it must be different from the image ('O'), where the size of images ('I') is also $w \times h$. Image ('I') can be

converted into I_R, I_G, I_B, so the size of matrix ('I_R, I_G, I_B') is $w \times h$ too. Therefore, the image ('I') is used to scramble the pixel values of image ('O'). The encryption procedure is according to the formula

$$\begin{cases} P_R(i,j) = O_R(i,j) \oplus I_R(i,j), \\ P_G(i,j) = O_G(i,j) \oplus I_G(i,j), \\ P_B(i,j) = O_B(i,j) \oplus I_B(i,j), \end{cases} \tag{3.2}$$

where $i = 1, 2, \ldots, w; j = 1, 2, \ldots, h$, and $\oplus$ is the exclusive OR operator. After the operations of '*XOR*' and '*mod*', all the pixel values of image ('O') are changed, therefore we get a scrambled image ('P').

Step 2. We need to generate the pseudo-random sequences from hyperchaotic systems by using the initial values and parameters in (2.1)(2.2)(2.3). The corresponding pseudo-random sequences are shown as follows:

$$\begin{aligned} X^k &= [x_1, x_2, \ldots, x_n], & Y^k &= [y_1, y_2, \ldots, y_n], \\ Z^k &= [z_1, z_2, \ldots, z_n], & W^k &= [w_1, w_2, \ldots, w_n], \end{aligned}$$

where $k = 1, 2, 3$, $n = w \times h + l$, and w, h stand for the width and height of image respectively, and l is the discarded former values of each sequence.

Step 3. The pseudo-random sequences generated from the fractional-order hyperchaotic Lü system are preprocessed according to formula

$$\begin{cases} X_i^k = ((X_{i+l}^k - [X_{i+l}^k]) \times 10^5) \mod 256, \\ Y_i^k = ((Y_{i+l}^k - [Y_{i+l}^k]) \times 10^5) \mod 256, \\ Z_i^k = ((Z_{i+l}^k - [Z_{i+l}^k]) \times 10^5) \mod 256, \\ W_i^k = ((W_{i+l}^k - [W_{i+l}^k]) \times 10^5) \mod 256, \end{cases} \tag{3.3}$$

where $i = 1, 2, \ldots, w \times h$, and [ ] is the rounding integer of X^k, Y^k, Z^k, W^k. So, the new pseudo-random sequences X^1, Y^1, Z^1, W^1 with length of $w \times h$ are integers in the interval [0,256].

Similarly, the pseudo-random sequences generated from the fractional-order hyperchaotic Lorenz and Chen systems are preprocessed according to the formula (3.3), therefore, we get random sequences X^2, Y^2, Z^2, W^2 and X^3, Y^3, Z^3, W^3, which have good stochastic properties.

Step 4. From Step 3, we can now obtain a random sequence via a combination of the random sequences. Select any three random sequences from X^1, Y^1, Z^1, W^1, X^2,Y^2,Z^2,W^2 and X^3,Y^3,Z^3,W^3, and combine them into new sequences C_1,C_2,C_3. For instance, one combination is given by

$$\begin{cases} C_R = (X^1 + Y^2 + Z^3) \mod 256, \\ C_G = (Y^1 + Z^2 + W^3) \mod 256, \\ C_B = (Z^1 + W^2 + X^3) \mod 256. \end{cases} \tag{3.4}$$

Furthermore, the new pseudo-sequences C_R, C_G, C_B are rearranged into three matrices $C_R^{'}, C_G^{'}, C_B^{'}$, respectively, whose sizes are also $w \times h$.

Step 5. Modify the pixel values for RGB color components of the image ('P') by the formulas

$$\begin{cases} P_R^{'}(i,j) = P_R(i,j) \oplus C_R^{'}(i,j), \\ P_G^{'}(i,j) = P_G(i,j) \oplus C_G^{'}(i,j), \\ P_B^{'}(i,j) = P_B(i,j) \oplus C_B^{'}(i,j). \end{cases} \tag{3.5}$$

The pixel values of image ('P') are completely diffused by the pseudo-sequences $C_R^{'}, C_G^{'}, C_B^{'}$, so that we get the resulting encryption image ('$P^{'}$') via the exclusive OR operation.

The flowchart of the encryption process is shown in Figure 2.

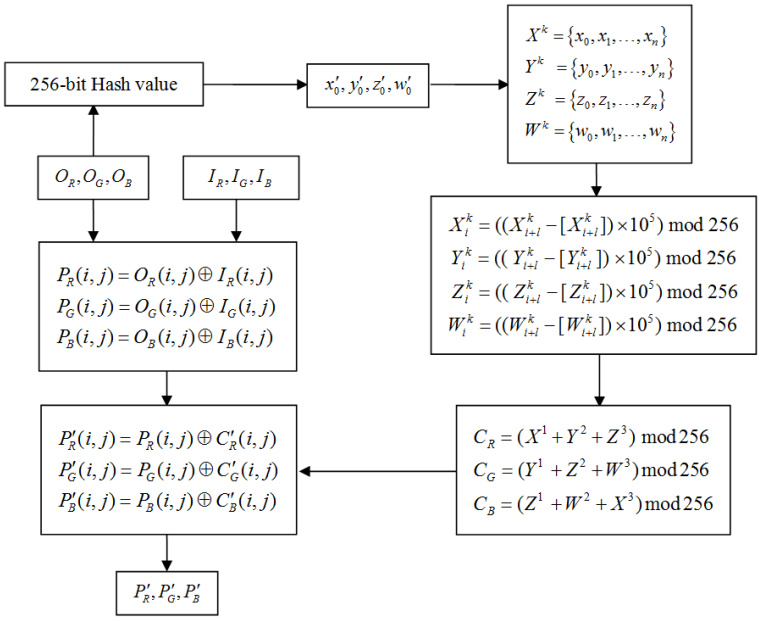


Figure 2. The procedure of image encryption.

Remark 3.1. For the initial values of fractional-order hyperchaotic systems, they are usually set by the user. In Subsection 3.1, SHA-256-bit hash function is employed to generate new initial values, which are unique. Therefore, the secret key is dynamic and relates to the original image. Accordingly, the proposed scheme has a good effect on resisting the differential attack.

Remark 3.2. The fractional-order hyperchaotic systems are used to produce four pseudo-random sequences. However, the color image has three RGB colors. So, we only need to choose any three sequences generated from X^k, Y^k, Z^k, W^k ($k = 1, 2, 3$). From the perspective of cryptography, this would expand the key space of the scheme.

Remark 3.3. In Step 1, a plain image ('I') is randomly chosen to scramble the pixel values of the original image, and it is known in advance.

3.3. Design of the decryption scheme

At the receiver side, the image ('$P^{'}$') is decrypted by the same pseudo-random sequences generated from a combination of fractional-order Lü, Lorenz and Chen hyperchaotic system. The decryption procedure is similar to the encryption process, with the reversed order, and the decryption procedure consists of the following two steps:

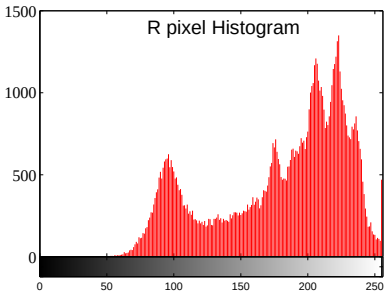
Step 1. We need to get a correct key including initial values and parameters, and use it to generate pseudo-random sequences from the combined fractional-order hyperchaotic systems. According to the encryption steps, reprocess the pseudo-sequences, then decipher the image ('$P^{'}$'). Thus, we get the deciphered image ('P').

Step 2. The plain image ('I') is known, so the original image ('O') can be recovered according to the reverse algorithm of the encryption in Step 1 if the secret key is correct.

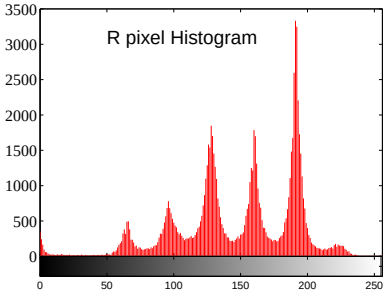
However, it can not be deciphered correctly if initial values or parameters are incorrect. More about security analyses and key sensitivities will be discussed in Section 5.

(a) The original image 'O'

(b) The image 'I'



(c) The histogram of 'O'



(d) The histogram of 'I'

Figure 3. The images and the corresponding histograms

4. Experimental results

This experiment is based on the platform of Matlab R2011a, and the results of simulations show the feasibility of the proposed scheme.

The variables initial values of the fractional-order hyperchaotic Lü system are set by

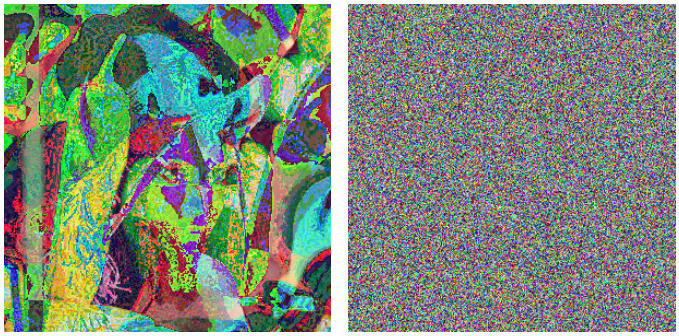
$$x_0^1 = 1.2157, y_0^1 = 3.0436, z_0^1 = 1.8573, w_0^1 = 2.9458.$$

Meanwhile, the variables initial values of the hyperchaotic Lorenz and Chen system are chosen as:

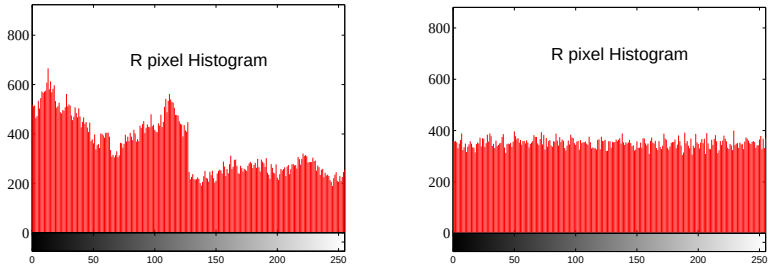
$$x_0^2 = 2.2254,\ y_0^2 = 5.5124,\ z_0^2 = 2.2458,\ w_0^2 = 1.2408,$$
$$x_0^3 = 0.5124,\ y_0^3 = 2.5278,\ z_0^3 = 0.8192,\ w_0^3 = 4.5281.$$

The familiar 'Lena.bmp' is chosen as the original image ('O'), and the plain image ('I') is 'Pepper.bmp'. They are shown in Figure 3 respectively, whose sizes are set as 300×300.

In Step 1, the image 'Lena.bmp' is confused as shown in Figure 4(a), which is blurred and confused. Then, the image 'P' is also encrypted as described in the previously encryption Steps 2-5. Finally, we get the resulting encrypted image. The resulting encrypted images and histograms are shown in Figure 4. Without any encryption, the image histograms are different and uneven. However, after the encryption of Steps 2-5, the histogram becomes much evenly as shown in Figure 4 (d).



(a) The first encrypted image (b) The resulting encrypted image



(c) The histogram of 'P' (d) The histogram of 'P''

Figure 4. The encrypted images and corresponding histograms

5. Security analyses

Here, several security tests are employed to verify the effectiveness of the new image encryption scheme.

5.1. Key space analysis

The high sensitiveness to initial conditions and parameters has a significant advantage in secure communications. It would provide a large enough key space against the brute force attacks.

In this encryption algorithm, system initial values, parameters values, derivative order of the fractional-order hyperchaotic systems, and the plain image ('I') are secret keys. For the initial conditions $x_0^2, y_0^2, z_0^2, w_0^2$ of the hyperchaotic Lorenz system, if the precision is 10^{-14}, the key space size will be 10^{56}. In our scheme, the initial values key space size is above $10^{14\times4\times3}$, the parameter key space size is above $10^{14\times3}$, and the derivative order can also create a larger key space. So, the total key space is more than $S \approx 10^{14\times4\times3} = 10^{210}$, which is large enough to resist brute-force attacks.

5.2. Key sensitivity analyses

We know that the secret key sensitivity play a decisive role on the key space, so the high sensitiveness of initial values will enlarge the key space and improve the security of the encryption scheme. One mismatch digit of initial values is used to decipher the encrypted image, for instance, using the initial value $x_0^1 = 1.21570000000001$, which is a little bigger than $x_0^1 = 1.2157$, and the other initial values keep matched. The deciphered image is shown in Figure 5 (a), from which one couldn't find any useful information. Therefore, if the secret keys value is changed a little, then the decrypted image is absolutely different from the original image. Meanwhile, if one of the initial values $x_0^i, y_0^i, z_0^i, w_0^i$ $(i = 1, 2, 3)$ is mismatched, then the deciphered image is blurred such that one can't get any useful information.

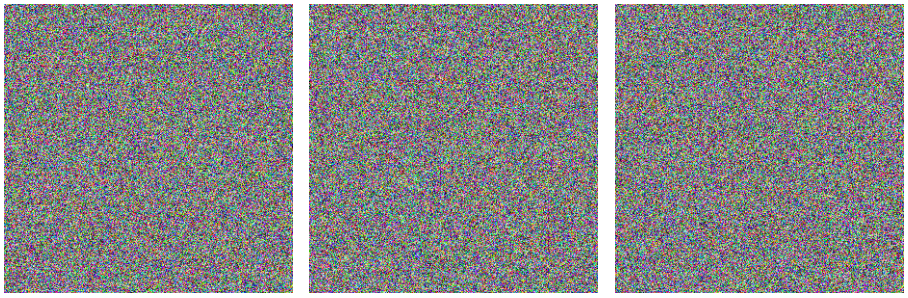
Similarly, choose one parameter, whose value is a little bigger than the standard parameters of equations (2.1) (2.2) (2.3), such as $a = 35.00000000000001$, while the others keep matched. Simulations show that the original images couldn't be recovered by using a wrong secret key, and Figure 5 (b) is an example for the results. In addition, if the order $\alpha = 0.96$, the recovered image shown in Figure 5(c) is blurred too.

Table 1. Differences between cipher images produced by slightly different keys

Decryption keys						Differences
x_1=1.2157	y_1=3.0436	z_1=1.8573	w_1=2.9258	a=35	α=0.95	ratio (%)
$+10^{-14}$	0	0	0	0	0	93.8211
0	$+10^{-14}$	0	0	0	0	93.9370
0	0	$+10^{-14}$	0	0	0	93.4130
0	0	0	$+10^{-14}$	0	0	93.8562
0	0	0	0	$+10^{-14}$	0	99.6325
0	0	0	0	0	0.01	99.5825

The differences between the corresponding ciphered images are computed and

given in Table 1. The results obviously demonstrate that the ciphered images exhibit no similarity and there isn't significant correlation that could be observed from the differential images.



(a) Incorrect initial value (b) Incorrect parameter (c) Incorrect derivative order

Figure 5. The deciphered images with incorrect secret keys

5.3. Differential attack

A good secure communication should resist all kinds of attacks, such as differential attack. The criteria of NPCR and UACI are used to examine the encryption scheme in resisting the differential attack. The formulae of calculating NPCR and UACI are given as follows [1]:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%,$$
$$UACI = \frac{1}{W \times H}[\frac{\sum_{i,j} |C_0(i,j) - C_1^{'}(i,j)|}{255}] \times 100\%,$$

where W and H represent the width and height of the image, $C_0(i,j)$ and $C_1{}'(i,j)$ are respectively the ciphered image before and after one pixel of the plain image is changed, and $D(i,j)$ are defined by:

$$D(i,j) = \begin{cases} 1, & \text{if } C_0(i,j) \neq C_1^{'}(i,j) \ , \\ 0, & \text{if } C_0(i,j) = C_1^{'}(i,j). \end{cases} \tag{5.1}$$

The test results are shown in Table 2. From Figure 4(b), the NPCR is over 99% and the average UACI is over 33%, so the encryption scheme is very sensitive with respect to the little change in the plain image. In our scheme, a simple plain image ('I') is employed to scramble the pixel values of the original image. Compared with the NPCR, to some extent, the effects of encryption are better than the method proposed by Hussain et al. [5]. Although the images used in this scheme are different from other references, test results may be largely identical with only minor differences.

5.4. Statistical analysis

The correlation coefficients of images can be used to measure the effectiveness of an encryption algorithm. In order to calculate the correlation coefficient of two

Table 2. NPCR and UACI of ciphered images by changing their original images by one bit.

Images	NPCR(%)				UACI(%)			
	R	G	B	Ave.	R	G	B	Ave.
Figure 4(a)	0.0017	0.0008	0.0015	0.0013	0.0004	0.0004	0.0004	0.0004
Figure 4(b)	99.6100	99.6311	99.6177	99.6196	33.3403	33.2045	33.2496	33.2648
Ref. [5]	94.6836	95.6835	98.6810	—	33.4647	34.5048	35.4999	—

adjacent pixels, we randomly select 5000 pairs of adjacent pixels to calculate it according to the following formula:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \quad \text{where } cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)),$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2.$$

Table 3 shows the correlation coefficients of the original image ('O') and the ciphered image ('P'). The correlation coefficients of the original image are nearly 1, implying that the image has good relevancy. After encryption, the correlation coefficients are close to zero, implying that the ciphered image has been well encrypted.

Table 3. Correlation coefficients of adjacent pixels in different images

		Vertical	Horizontal	Diagonal
Plain Lena Figure 3(a)	Red	0.9769	0.9522	0.9318
	Green	0.9745	0.9552	0.9370
	Blue	0.9531	0.9117	0.8767
Ciphered Lena Figure 4(a)	Red	0.4512	0.5611	0.4825
	Green	0.6397	0.6737	0.5959
	Blue	0.4311	0.5184	0.4151
Ciphered Lena Figure 4(b)	Red	-0.0242	0.0018	0.0171
	Green	0.0241	0.0094	-0.0074
	Blue	-0.0076	-0.0115	0.0121

5.5. Information entropy

In 1949, Shannon firstly introduced the information entropy, which was a mathematical property that reflects the randomness and the unpredictability of an information source [14]. The entropy of information $H(s)$ is defined as

$$H(s) = -\sum_{i=0}^{2^N-1} P(s_i)\log_2 P(s_i),$$

where s is the source, N is the number of bits to represent the symbol s_i, and $P(s_i)$ is the probability of the symbol s_i. For a truly random source consisting of 2^N symbols, the entropy is N. Therefore, for an effective cryptosystem, the entropy of the ciphered image with 256 gray levels should ideally be 8. The information entropies are calculated and listed in Table 4.

From Table 4, where three different images are chosen as test objects, the results show that the information entropies could approach the theoretical value 8. So, information leakage in the encryption procedure could be negligible and the proposed algorithm is secure against entropy analysis.

Table 4. Entropies of the original images and encryption results

	Plain image	Ciphered images	
	Original image	First encrypted images	Resulting encrypted images
lena	7.2898	7.9268	7.9979
Baboon	2.8413	7.5589	7.9976
Barb	7.7340	7.9185	7.9978

6. Conclusions

In this paper, an image encryption method is proposed. The original image is scrambled by using a known image. Then, image is encrypted once again by the pseudo-random sequences generated from the combined fractional-order hyperchaotic systems. Experiments demonstrate that the scheme is suitable for image encryption with the same size in batches. Some security tests are discussed to show the effectiveness of the new scheme. The key space is large enough to resist brute force attacks. For chaos-based secure communications, in the future, we will investigate some better algorithms and use other high-dimensional hyperchaotic systems.

Acknowledgements

The authors thank the referees for their valuable comments and suggestions.

References

[1] G. Chen, Y. Mao and C. Chui, *A symmetric image encryption scheme based on 3D chaotic cat maps*, Chaos, Solitons & Fractals, 21(2004), 749-761.

[2] A. Chen, J. Lu, J. Lü and S. Yu, *Generating hyperchaotic Lü attractor via state feedback control*, Physica A: Statistical Mechanics and its Applications, 364(2006), 103-110.

[3] K. Diethelm, N.J. Ford and A.D. Freed, *A predictor-corrector approach for the numerical solution of fractional differential equations*, Nonlinear Dyn, 29(2002), 3-22.

[4] A. S. Hegazia and A. E. Matouk, *Dynamical behaviors and synchronization in the fractional order hyperchaotic Chen system*, Applied Mathematics Letters, 24(2011), 1938-1944.

[5] I. Hussain, T. Shah and M. A. Gondal, *Image encryption algorithm based on PGL(2,GF(28)) S-boxes and TD-ERCS chaotic sequence*, Nonlinear Dyn, 70(2012), 181-187.

[6] E. N. Lorenz, *Deterministic nonperiodic flow*, J Atmospheric Sciences, 20(1963), 130-141.

[7] J. Lü and G. Chen, *A new chaotic attractor coined*, Int J Birurcat Chaos, 12(2002), 659-661.

[8] H. Liu and X. Wang, *Triple-image encryption scheme based on one-time key stream generated by chaos and plain images*, Journal Systems and Software, 86(2013), 826-834.

[9] C. Li, Y. Liu, T. Xie and M. Chen, *Breaking a novel image encryption scheme based on improved hyperchaotic sequences*, Nonlinear Dyn, 73(2013), 2083-2089.

[10] S. Mazloom and A. M. Eftekhari-Moghadam, *Color image encryption based on Coupled Nonlinear Chaotic Map*, Chaos, Solitons & Fractals, 42(2009), 1745-1754.

[11] I. Podlubny, *Fractional Differential Equations*, Academic Press, New York, 1999.

[12] F. Riewe, *Mechanics with fractional derivatives*, Physical Review E, 55(1997), 3581-3592.

[13] E. Solak, C. Cokal, O. Yildiz and T. Biyikoglu, *Cryptanalysis of Fridrichs chaotic image encryption*, Int J Bifur Chaos, 20(2010), 1405-1413.

[14] C. Shannon, *Communication theory of secrecy systems*, Bell Syst Tech J, 18(1949), 656-715.

[15] X. Wu, *A Color Image Encryption Algorithm Using the Fractional-order Hyperchaotic Systems, Chaos-Fractals Theories and Applications (IWCFTA)*, 2012 Fifth International Workshop on IEEE, (2012), 196-201.

[16] X. Wang and J. Song, *Synchronization of the fractional order hyperchaos Lorenz systems with activation feedback control*, Commun Nonlinear Sci Numer Simul. 14(2009), 3351-3357.

[17] S. Yu, J. Lü, X. Yu and G. Chen, *Design and implementation of grid multiwing hyperchaotic Lorenz system family via switching control and constructing super-heteroclinic loops*, IEEE Transactions on Circuits and Systems, 59(2012), 1015-1028.