Journal of Pure and Applied Mathematics: Advances and Applications
Volume 6, Number 1, 2011, Pages 1-29

A NEW CHAOTIC CRYPTOSYSTEM BASED ON MULTIPLE ONE-DIMENSIONAL CHAOTIC MAPS

XINGYUAN WANG and YISONG TAN

School of Electronic & Information Engineering
Dalian University of Technology
Dalian 116024
P. R. China
e-mail: wangxy@dlut.edu.cn

Abstract

In this paper, the authors propose a new chaotic block cryptosystem based on multiple one-dimensional chaotic maps. In the proposed cryptosystem, the authors establish map-to-task relations between the multiple chaotic maps and the tasks, which must be performed in the encryption rounds. The encryption process is based on these map-to-task relations. The authors enhance the security of the cryptosystem by changing the map-to-task relations constantly and applying plaintext and ciphertext feedback in the encryption process. Experiments and theoretic analysis show that the proposed cryptosystem possesses high performance and security.

1. Introduction

Chaos is characterized by ergodicity, random-like behaviours, sensitive dependence on control parameters, and initial conditions. These properties are analogous to the confusion and diffusion properties of a

2010 Mathematics Subject Classification: 34C28, 94A60.

Keywords and phrases: chaos, cryptosystem, plaintext, ciphertext, security, encryption, decryption.

Received May 11, 2011

© 2011 Scientific Advances Publishers

good cryptosystem. Consequently, chaos can be used in cryptosystem. In recent years, a lot of researchers has researched many chaotic cryptology algorithms and proposed many chaotic cryptosystem [7, 8]. Systematic theory of chaotic cryptology is being established gradually [6, 9].

In 1998, Baptista [2] proposed a chaotic cryptosystem based on ergodicity of chaos. Baptista's cryptosystem is based on a simple one-dimensional logistic map. In Baptista's cryptosystem, message text is encrypted as the number of iterations applied in the chaotic map in order to reach the region corresponding to that text. Baptista's cryptosystem has some flaws. Álvarez et al. [1] analyzed the security flaws of Baptista's cryptosystem and proposed four methods to attack it. Other researchers [3, 11, 12] also analyzed the drawbacks of Baptista's cryptosystem and proposed some schemes to improve it. In 2003, Li et al. [4] pointed out that a chaotic system implemented in a computer with finite computing precision will encounter problem of dynamical degradation, which induces many weak keys to cause large information leaking of the plaintext in a chaotic cryptosystem. Many researchers try to avoid the affection of this problem by designing cryptosystems based on multiple chaotic maps or high dimensional chaotic maps, for example, Pareek et al. [5] proposed a cryptosystem based on multiple one-dimensional chaotic maps. However, Wei et al. [10] analyzed Pareek et al.'s cryptosystem and proposed a method to attack it.

In this paper, the authors propose a new chaotic block cryptosystem based on multiple one-dimensional chaotic maps. In the proposed cryptosystem, a block of the plaintext or the ciphertext is a byte. In every encryption round of the encryption process, several tasks must be performed by the multiple chaotic maps to encrypt the current plaintext block. In every encryption round, one chaotic map can perform only one task. We establish map-to-task relations between the maps and the tasks to determine which task will be performed by which chaotic map in an encryption round. To enhance the security of the proposed cryptosystem,

we change the map-to-task relations constantly in the overall encryption process. Consequently, in different encryption rounds, one task is usually performed by different chaotic maps. By randomly changing the relations between the tasks and the chaotic maps, we can make the relation between the plaintext and the ciphertext more complicated, which leads to higher security of the proposed cryptosystem. To further enhance the security of the cryptosystem, we apply plaintext and ciphertext feedback in the encryption process. Experiments and theoretic analysis show that the proposed cryptosystem possesses high performance and security.

2. Description of the Proposed Cryptosystem

2.1. Five one-dimensional chaotic maps

We denote the five one-dimensional chaotic maps used in the proposed cryptosystem as the following equations:

$$x_{n+1} = f_1(x_n, \lambda), \tag{1}$$

$$y_{n+1} = f_2(y_n, \mu), \tag{2}$$

$$z_{n+1} = f_3(z_n, \nu), \tag{3}$$

$$w_{n+1} = f_4(w_n, \gamma), \tag{4}$$

$$u_{n+1} = f_5(u_n, \alpha), \tag{5}$$

where x, y, z, w, and u denote the state variables of the five chaotic maps, respectively. λ, μ, ν, γ, and α denote the control parameters of the five chaotic maps, respectively.

2.2. Two encryption operations

Two encryption operations, which can be represented as follows are used by the proposed cryptosystem:

Encryption operation 1:

$$C_i = (P_i + \lfloor STATE_{mask} \times 10^{14} \rfloor) \bmod 256. \quad (6)$$

Encryption operation 2:

$$C_i = (\lfloor STATE_{mask} \times 10^{14} \rfloor \bmod 256) \oplus P_i, \quad (7)$$

where C_i denotes the i-th block of the ciphertext, P_i denotes the i-th block of the plaintext, $STATE_{mask}$ denotes the state variable of the chaotic map, which is used to mask P_i. In every encryption round of the encryption process, we randomly choose one of the above two encryption operations to mask the plaintext block, i.e., in the encryption process, the encryption operation used to mask the plaintext blocks is changed constantly. In this way, we can make the relation between the plaintext and ciphertext more complicated, which leads to higher security of the cryptosystem.

2.3. Four tasks

In every encryption round of the proposed cryptosystem, four tasks must be performed to encrypt the current plaintext block. The four tasks are listed as follows:

(a) Calculate the length of the right cyclic shift, which is applied to the plaintext block to be encrypted.

(b) Calculate the number of iterations applied in the chaotic map, which is used to mask the plaintext block to be encrypted.

(c) Determine which of the two encryption operations is used to mask the current plaintext block.

(d) Mask the current plaintext block with state variable.

2.4. A map-to-task table

Among the five chaotic maps, f_1 is used to generate initial conditions for the other four chaotic maps. In an encryption round, f_2, f_3, f_4, and

f_5 are used to perform the four tasks introduced in the previous subsection (Subsection 2.3). In an encryption round, one chaotic map can perform only one task. To determine which task should be performed by which chaotic map, we construct map-to-task relations between the chaotic maps and the tasks. For example, in an encryption round, if there is a map-to-task relation represented as the following Table 1, then we can know that in the encryption round, task(a), task(b), task(c), and task(d) will be performed by f_2, f_3, f_4, and f_5, respectively.

Table 1. A map-to-task relation

task(a)	task(b)	task(c)	task(d)
f_2	f_3	f_4	f_5

We can easily know that, there are 24 possible relations between the four chaotic maps and the four tasks. We construct a map-to-task table as follow to store the 24 different map-to-task relations (Table 2).

Table 2. Map-to-task table, which contain the 24 map-to-task relations

Row index	task(a)	task(b)	task(c)	task(d)
1	f_2	f_3	f_4	f_5
2	f_2	f_3	f_5	f_4
3	f_2	f_4	f_3	f_5
4	f_2	f_4	f_5	f_3
5	f_2	f_5	f_3	f_4
6	f_2	f_5	f_4	f_3
7	f_3	f_2	f_4	f_5
8	f_3	f_2	f_5	f_4
9	f_3	f_4	f_2	f_5
10	f_3	f_4	f_5	f_2
11	f_3	f_5	f_2	f_4
12	f_3	f_5	f_4	f_2
13	f_4	f_2	f_3	f_5
14	f_4	f_2	f_5	f_3
15	f_4	f_3	f_2	f_5
16	f_4	f_3	f_5	f_2
17	f_4	f_5	f_2	f_3
18	f_4	f_5	f_3	f_2
19	f_5	f_2	f_3	f_4
20	f_5	f_2	f_4	f_3
21	f_5	f_3	f_2	f_4
22	f_5	f_3	f_4	f_2
23	f_5	f_4	f_2	f_3
24	f_5	f_4	f_3	f_2

In an encryption round, we randomly choose one of the 24 map-to-task relations from the map-to-task table and use it. Consequently, in different encryption rounds, one task is usually performed by different chaotic maps. By randomly changing the relations between the tasks and the chaotic maps, we can make the relation between the plaintext and the ciphertext more complicated. In this way, the security of the cryptosystem can be enhanced.

For convenience, we denote the maps which performed task(a), task(b), task(c), and task(d), respectively, as f_{shift_len}, f_{iter_time}, $f_{operate}$, and f_{mask}, respectively.

2.5. Encryption process

(i) Iterate chaotic map f_1 200 times from its initial condition to make the trajectory of the map fall into its chaotic attractor. Iterate f_1 50 more times , get the new value of the state variable and make it as the initial condition of f_2; iterate f_1 50 more times, get the new value of the state variable and make it as the initial condition of f_3; iterate f_1 50 more times, get the new value of the state variable and make it as the initial condition of f_4; iterate f_1 50 more times, get the new value of the state variable and make it as the initial condition of f_5.

(ii) Iterate f_2, f_3, f_4, and f_5 200 times from their initial condition, respectively, to make their trajectories fall into their chaotic attractor.

(iii) For the first block of the plaintext, we use the first row of the map-to-task table (Table 2) to determine the map-to-task relation of the first encryption round, i.e., f_2 is f_{shift_len}, f_3 is f_{iter_time}, f_4 is $f_{operate}$, and f_5 is f_{mask}.

Perform task (a):

Iterate f_{shift_len}, get the new value of its state variable (denote it as $STATE_{shift\_len}$) and calculate

$$LEN = \lfloor STATE_{shift\_len} \times 10^{14} \rfloor \bmod 8, \tag{8}$$

where LEN denotes the length of the cyclic right shift, which will be applied to the current plaintext block.

Perform task (b):

Iterate f_{iter_time}, get the new value of its state variable (denote it as $STATE_{iter\_time}$) and calculate

$$IT = \lfloor STATE_{iter\_time} \times 10^{14} \rfloor \bmod I + 10, \tag{9}$$

where IT denotes the number of iterations applied to f_{mask} before masking the current plaintext block and I is an integer constant. If the value of I is bigger, the cryptosystem will be more secure, but the encryption speed of the cryptosystem will be lower.

Perform task (c):

Iterate $f_{operate}$, get the new value of its state variable (denote it as $STATE_{operate}$) and calculate

$$OP = \lfloor STATE_{operate} \times 10^{14} \rfloor \bmod 2 + 1. \tag{10}$$

If $OP = 1$, we will choose encryption operation 1 to mask the current plaintext block, otherwise, we will choose encryption operation 2.

(iv) Permute the current plaintext block P_i with right cyclic shift LEN bits and get the resulting block P_i^*, i.e.,

$$P_i^* = P_i >> LEN \text{ bits}, \tag{11}$$

where ">>" denotes the right cyclic shift operation.

Perform task (d):

Iterate f_{mask} IT times, get the new value of its state variable (denote it as $STATE_{mask}$). If $OP = 1$, we choose encryption operation 1 to mask P_i^*, i.e., we encrypt P_i^* as

$$C_i = (P_i^* + \lfloor STATE_{mask} \times 10^{14} \rfloor) \bmod 256, \tag{12}$$

otherwise, we choose encryption operation 2 to mask P_i^*, i.e., we encrypt P_i^* as

$$C_i = (\lfloor STATE_{mask} \times 10^{14} \rfloor \bmod 256) \oplus P_i^*. \tag{13}$$

(v) If all of the blocks of the plaintext have been encrypted, output the ciphertext, otherwise, we calculate

$$ROW = (\lfloor P_{i-1} + C_{i-1} + STATE_{mask} \times 10^{14} \rfloor) \bmod 24 + 1, \tag{14}$$

where ROW is used to determine which map-to-task relation will be used in the next encryption round, i.e., map-to-task relation in the ROW-th row of the map-to-task table (Table 2) will be used in the next encryption round.

(vi) After new f_{shift_len}, f_{iter_time}, $f_{operate}$, and f_{mask} are determined, we iterate f_{group}, f_{iter_time}, and $f_{operate}$, respectively, get new values of their state variables (denote them as $STATE_{shift\_len}$, $STATE_{iter\_time}$, and $STATE_{mask}$, respectively) and calculate

Perform task (a):

$$LEN = \lfloor P_{i-1} + C_{i-1} + STATE_{shift\_len} \times 10^{14} \rfloor \bmod 8. \tag{15}$$

Perform task (b):

$$IT = \lfloor P_{i-1} + C_{i-1} + STATE_{iter\_time} \times 10^{14} \rfloor \bmod I + 10. \tag{16}$$

Perform task (c):

$$OP = \lfloor P_{i-1} + C_{i-1} + STATE_{operate} \times 10^{14} \rfloor \bmod 2 + 1. \tag{17}$$

where P_{i-1} denotes the previous plaintext block, C_{i-1} denotes the ciphertext of P_{i-1}. Plaintext and ciphertext feedback are used in this step, which will enhance the diffusion effect of the encryption process and consequently enhance the security of the cryptosystem.

(vii) Repeat steps (iv)-(vi) until all plaintext blocks are encrypted.

2.6. Decryption process

The decryption process is the reverse process of the encryption process, but in the decryption process, we must replace equations

$$C_i = (P_i^* + \lfloor STATE_{mask} \times 10^{14} \rfloor) \bmod 256, \tag{18}$$

and

$$C_i = (\lfloor STATE_{mask} \times 10^{14} \rfloor \bmod 256) \oplus P_i^*, \tag{19}$$

with equations

$$P_i^* = (C_i + 256 - (\lfloor STATE_{mask} \times 10^{14} \rfloor \bmod 256)) \bmod 256, \tag{20}$$

and

$$P_i^* = (\lfloor STATE_{mask} \times 10^{14} \rfloor \bmod 256) \oplus C_i, \tag{21}$$

respectively.

2.7. Secret key

The proposed cryptosystem is a symmetry cryptosystem, i.e., the encryption key and the decryption key are the same. The secret key of the proposed cryptosystem is composed of the initial condition x_0 of f_1, the five control parameters λ, μ, ν, γ, and α of the five chaotic maps, and integer I in Equations (9) and (16).

3. Experiments and Analysises

In the following experiments, f_1 is a logistic map:

$$x_{n+1} = \lambda x_n (1 - x_n), \quad x_n \in (0, 1), \tag{22}$$

f_2 is a logistic map:

$$y_{n+1} = \mu y_n(1 - y_n), \quad y_n \in (0, 1), \tag{23}$$

f_3 is a sine map:

$$z_{n+1} = \nu \sin(\pi z_n), \quad z \in (0, 1), \tag{24}$$

f_4 is a tent map:

$$w_{n+1} = \begin{cases} \gamma w_n, & \text{if } w_n < 0.5, \\ \gamma(1 - w_n), & \text{if } w_n \geq 0.5, \end{cases} \tag{25}$$

and f_5 is a cubic map:

$$u_{n+1} = \alpha u_n(1 - u_n^2), \quad u \in (0, 1). \tag{26}$$

The values of control parameters of f_1, f_2, f_3, f_4, and f_5 are $\lambda = 3.99992636836375$, $\mu = 3.99263254852698$, $\nu = 0.99451673564973$, $\gamma = 1.97346154279534$, and $\alpha = 2.59431678431627$, respectively. Under this condition, the five maps are chaotic. The initial condition of f_1 is $x_0 = 0.87372345564312$.

We implement the cryptosystem with Visual C++ 6.0 and run it in a personal computer with a Pentium-IV 1.40GHz Celeron CPU, 256MB memory, 80GB hard-disk capacity, and Windows XP operation system.

3.1. Encryption and decryption experiments

The proposed cryptosystem has nothing to do with the files' internal memory and encoding format, therefore it is fit for encrypt all kinds of files, which is an advantage of it. We use the proposed cryptosystem to encrypt a text string "this is a new chaotic cryptosystem", a text file of 2.58K and an index image "Lena" of 512×512 pixels. The following Figure 1 presents the ASCII distribution of the string "this is a new chaotic cryptosystem" and its corresponding ciphertext. Figure 2 presents

the ASCII distribution of the plaintext of the text file and its corresponding ciphertext. Figure 3 presents the plain image, ciphered image, and recovered image of "Lena". Figure 4 presents the gray scale distribution of the plain image and the ciphered image.

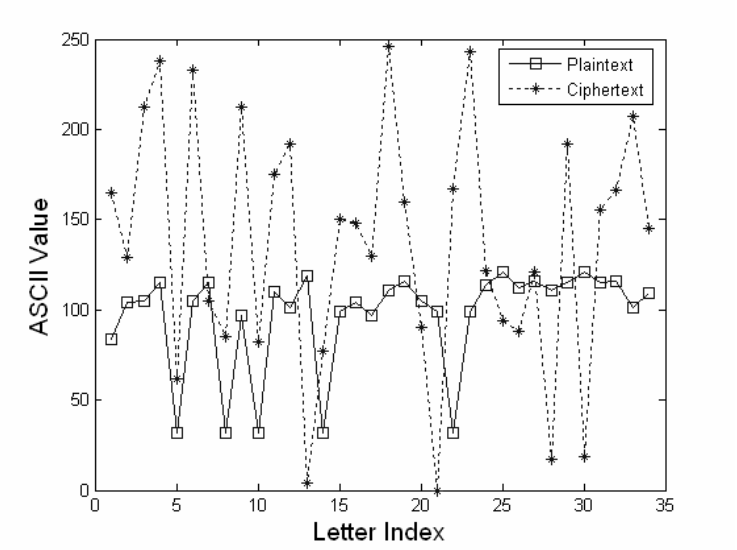
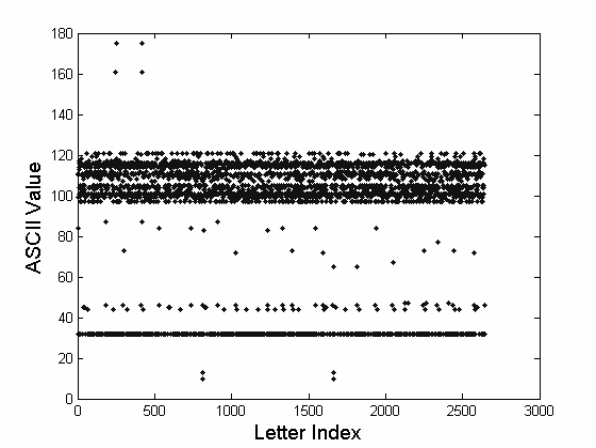
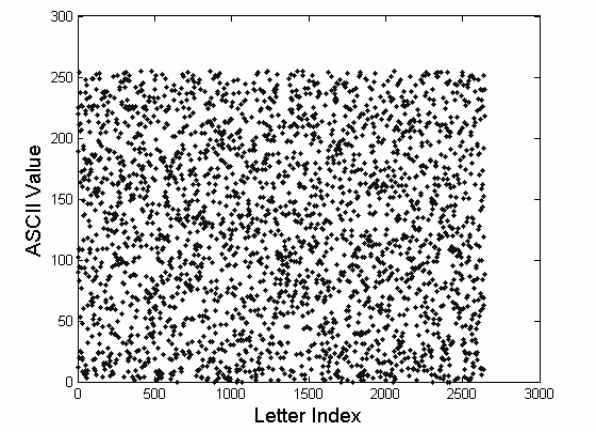


Figure 1. ASCII distribution of string "this is a new chaotic cryptosystem" and its ciphertext.



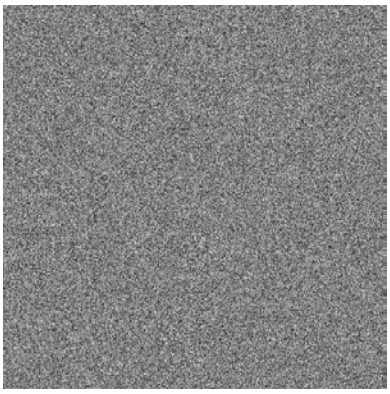
(a) ASCII distribution of plaintext.



(b) ASCII distribution of ciphertext.

Figure 2. ASCII distribution of the plaintext and ciphertext of the text file of 2.58K.

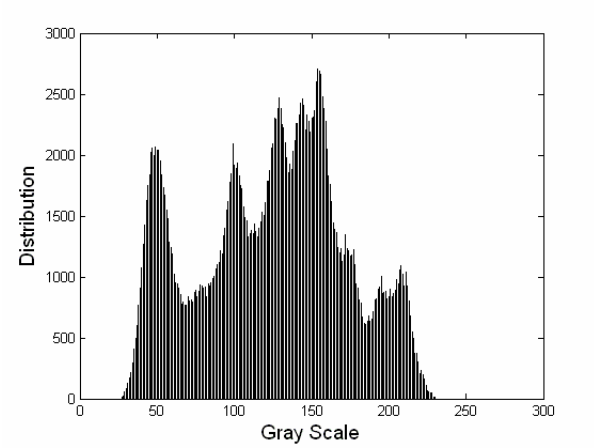
(a) Plain image



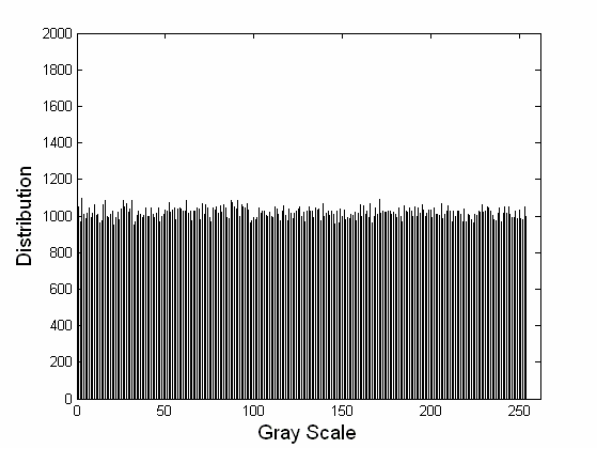
(b) Ciphered image

(c) Recovered image

Figure 3. Plain image, ciphered image, and recovered image of "Lena".



(a) Gray scale distribution of the plain image.



(b) Gray scale distribution of the ciphered image.

Figure 4. Gray scale distribution of the plain image and the ciphered image of "Lena".

3.2. Analysis of the plaintexts and ciphertexts

Figure 1 shows that the ASCII distribution of the string "this is a new chaotic cryptosystem" is entirely different from that of its ciphertext. Figure 2 shows that the ASCII codes of the plaintext mostly distribute in [30, 120], while the ASCII codes of the ciphertext evenly distribute in [0, 255]. Figure 3 shows that the generated ciphered image is not understandable, while the recovered image is the same as the plain image. We can see from Figure 4 that the gray scale distribution of the plain image is not even, while the gray scale distribution of the ciphered image is quite even. The even ASCII code or gray scale distribution of the ciphertexts can hide the statistic information of the plaintexts effectively. Consequently, it is very difficult for the attacker to break the ciphertext by using statistic method.

The authors also analyze the correlation coefficients of the plain image and the ciphered image of "Lena". The authors randomly select 1000 pairs of two-adjacent pixels (in vertical, horizontal, and diagonal direction) from the plain image and the ciphered image, respectively. Then, the correlation coefficients of the plain image and ciphered image are calculated with the following equations:

$$r_{xy} = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{27}$$

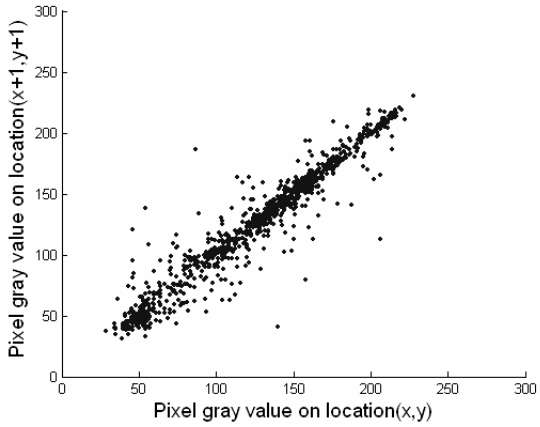
$$\operatorname{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \tag{28}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i, \tag{29}$$

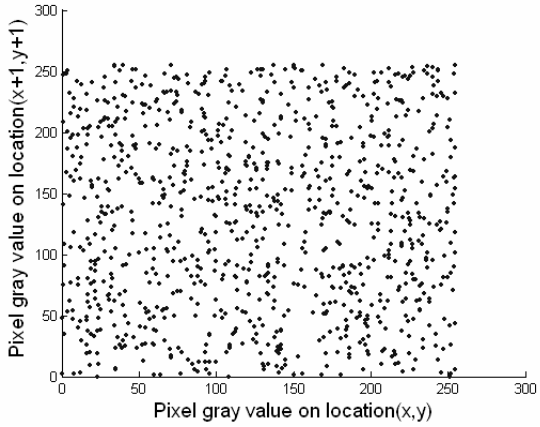
$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2, \tag{30}$$

where x and y denote the gray-scale values of two adjacent pixels in the image, $E(x)$ and $D(x)$ are the estimation of mathematical expectations and estimation of variance of x, respectively, and $\operatorname{cov}(x, y)$ is the estimation of covariance between x and y. Figure 5 presents the correlation analysis result of two-horizontally-adjacent pixels of the plain

image and the ciphered image. Table 3 presents the correlation coefficients of two-adjacent pixels of the plain image and the ciphered image. Figure 5 and Table 3 show that the correlation coefficients of the plain image are very big and the correlation coefficients of the ciphered image are very small. We can see that the cryptosystem can achieve a desirable encryption result.



(a) Correlation analysis of plain image (correlation coefficients = 0.966216).



(b) Correlation analysis of ciphered image (correlation coefficients = 0.056345).

Figure 5. Correlation analysis of two-horizontally-adjacent pixels of the plain image and the ciphered image.

Table 3. Correlation coefficients of two-adjacent pixels of the plain image and the ciphered image

Direction	Plain	Ciphered
Horizontal	0.966216	0.056345
Vertical	0.992372	0.035912
Diagonal	0.985614	0.036248

3.3. Sensitive dependence on secret keys

We perform two experiments in this subsection. In the two experiments, we let $\lambda = 3.99992636836375$, $\mu = 3.99263254852698$, $\nu = 0.99451673564973$, $\gamma = 1.97346154279534$, and $\alpha = 2.59431678431627$. In the first experiment, we encrypt the string "this is a new chaotic cryptosystem" with the proposed cryptosytem twice. In the first encryption, let $x_0 = 0.9$ and in the second encryption, $x_0 = 0.90000000000001$. We got two different ciphertexts in this experiment. The following Figure 6 presents the difference between the two ciphertexts.

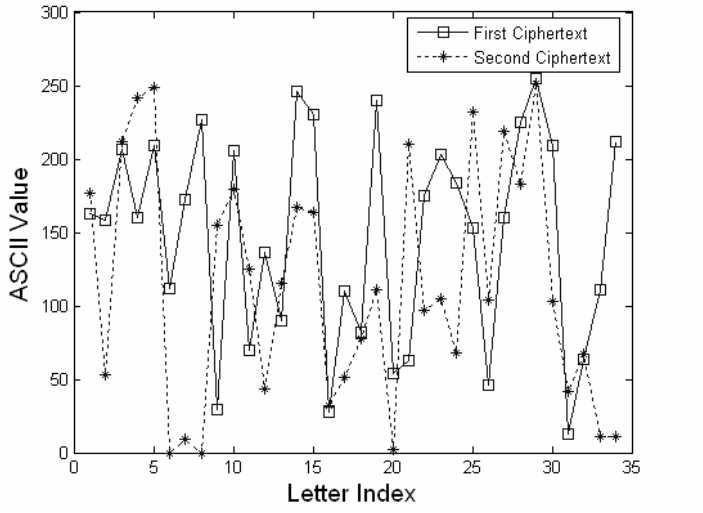


Figure 6. Difference between the two ciphertext with x_0 changes 10^{-14}.

In the second experiment, we encrypt the string "this is a new chaotic cryptosystem" with the cryptosystem with $x_0 = 0.8$. Then, we decrypt the generated ciphertext with the correct control parameters and the wrong initial condition $x_0 = 0.80000000000001$. Figure 7 presents the difference between the plaintext and the recovered text.

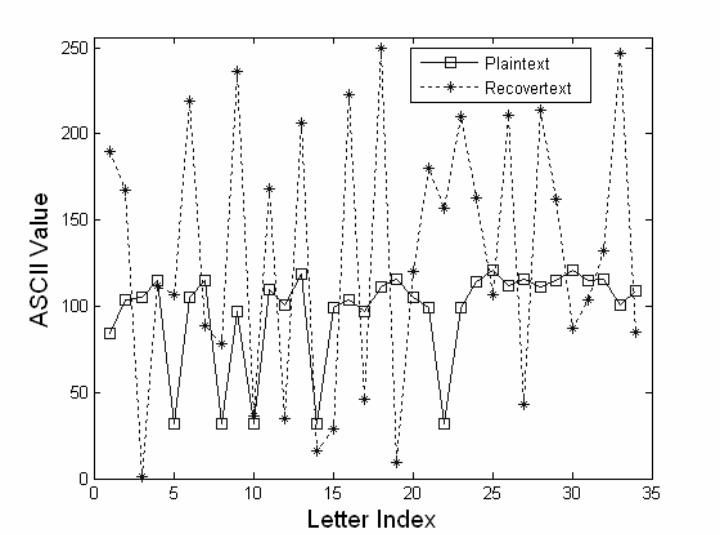


Figure 7. Difference between the plaintext and the recovered text with x_0 changes 10^{-14}.

Figure 6 shows that for a plaintext, slight change (10^{-14}) of x_0 will lead to completely different encryption results. Figure 7 shows that slight change (10^{-14}) of x_0 will lead to the failure of decryption. We can see that the proposed cryptosystem is very sensitive to the change of x_0. We have performed other experiments, which prove that the proposed cryptosystem is also very sensitive to the five control parameters. We can see that the proposed cryptosystem is very sensitive to the change of the secret key. Consequently, the proposed cryptosystem can resist brute-force attack effectively.

3.4. Differential attack

We use *NPCR* and *UACI* to test the influence of one-pixel change on the whole image "Lena". *NPCR* (number of pixels change rate) stands for the number of pixels change rate, while one pixel of plain image changed. *UACI* (unified average changing intensity) stands for the average intensity of differences between the plain image and ciphered image. The bigger *NPCR* is, the more sensitive for the cryptosystem to the pixels change of plain image. The bigger *UACI* is, the more effective for the cryptosystem to resist differential attack. We assume that there is a two ciphered images $(C_1 \text{ and } C_2)$, whose corresponding plain images have only one-pixel difference. The formulas to calculate *NPCR* and *UACI* are:

$$NPCR = \frac{\sum_{i,j} D(i,\ j)}{W \times H} \times 100\%, \tag{31}$$

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,\ j) - C_2(i,\ j)|}{255}\right] \times 100\%, \tag{32}$$

where W and H represent the width and height of the image, respectively. For the pixel at position $(i,\ j)$, if $C_1(i,\ j) \neq C_2(i,\ j)$, then $D(i,\ j) = 1$; otherwise, $D(i,\ j) = 0$. We have calculated *NPCR* and *UACI* of the image "Lena" encrypted by the proposed cryptosystem and got the results *NPCR* = 99.6251% and *UACI* = 33.2875%, respectively. The results show that the proposed cryptosystem can resist differential attack effectively.

3.5. Key space

The secret key of the proposed cryptosystem is composed of six double-floating numbers and one integer. The precision of the six double-floating numbers is 10^{-14}. We can easily know that the key space of the proposed cryptosystem is larger than 10^{90}. The key space is large enough to resist brute-force attack.

3.6. Information entropy

Information entropy is the most important feature of randomness. Let m denotes the information source, we can calculate information entropy by using the following formula:

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i) \log_2 \frac{1}{p(m_i)}, \tag{33}$$

where $p(m_i)$ denotes the probability of symbol m_i. Assume that there are 2^8 states of the information source and they appear with the same probability. According to Equation (33), we can easily calculate the ideal information entropy $H(m) = 8$, which shows that the information is random. Hence, the information entropy of the ciphertext generated by a good cryptosystem should be close to 8. The closer it gets to 8, the less possible for the cryptosystem to divulge information. We calculate the information entropy of the ciphertext of the text file encrypted in Subsection 3.1 according to Equation (33) and get the result $H(m) = 7.998623$. We also calculate the information entropy of the ciphered image of "Lena" and get $H(m) = 7.997889$. We can see that the information entropy of the ciphertext generated by the proposed cryptosystem is close to the ideal value 8, so the probability of accidental information leakage is very little, which leads to high security of the cryptosystem.

3.7. Speed experiments

The following Table 4 presents the comparison of encryption speed of the proposed cryptosystem and Pareek et al.'s cryptosystem [5]. We can see from Table 4 that the encryption speed of the proposed cryptosystem is higher than that of Pareek et al.'s cryptosystem. So, the proposed cryptosystem is more suitable for practical application.

Table 4. Encryption speed of the proposed cryptosystem and Pareek et al.'s cryptosystem

File type	Size of plaintext (KB)	Encryption time of the proposed cryptosystem Min-Max (Mean)	Encryption time of Pareek et al.'s cryptosystem Min-Max (Mean)	Size of ciphertext (KB)
Text file (*.txt)	30	0.094s-0.125s (0.113s)	0.122s-0.276s (0.186s)	30
	90	0.266s-0.292s (0.271s)	0.324s-0.503s (0.368s)	90
	240	0.672s-0.703s (0.682s)	0.865s-1.869s (1.239s)	240
Document file (*.doc)	30	0.109s-0.135s (0.118s)	0.123s-0.246s (0.167s)	30
	90	0.265s-0.281s (0.274s)	0.357s-0.503s (0.435s)	90
	240	0.688s-0.704s (0.695s)	0.986s-1.965s (1.221s)	240
Image file (*.bmp)	192	0.593s-0.625s (0.606s)	0.701s-1.162s (1.101s)	192
	768	2.172s-2.403s (2.189s)	3.110s-4.102s (3.674s)	768
	3072	8.469s-8.916s (8.491s)	10.365s-15.341s (12.136s)	3072
Music file (*.mp3)	1382	3.859s-3.922s (3.878s)	5.236s-9.352s (7.306s)	1382
	2437	6.656s-6.692s (6.663s)	9.1686s-14.953s (11.652s)	2437

4. Possible Improvements

There are some possible improvement schemes of the proposed cryptosystem.

(i) Applying more chaotic maps

Using more chaotic maps can make the encryption process more complicated and lead to bigger key space of the cryptosystem, so we can obtain higher security of the cryptosystem.

(ii) Applying high dimensional chaotic maps

The trajectories of high dimensional chaotic maps are more complicated than that of the one-dimensional chaotic maps, so using high dimensional chaotic maps will lead to better encryption results. However, using high dimensional maps may offset the speed of the cryptosystem.

(iii) Exchanging the rows of the map-to-task table constantly

We can change the content of the map-to-task table constantly during the encryption process to make the encryption process more complicated. For example, we can constantly exchange the content of the two rows, which are most recently used after encrypting a plaintext block. In this way, the order of the map-to-tasks relations stored in the map-to-task table is constantly changed, which make the encryption process more complicated and lead to higher security. Exchanging two rows of the map-to-task table is not time consuming, so this improvement scheme will not affect the speed of the cryptosystem significantly.

5. Comparisons

5.1. Compared with Pareek et al.'s cryptosystem and its improved version

In 2005, Pareek et al. [5] proposed a chaotic cryptosystem based on multiple one-dimensional chaotic maps. Wei et al. [10] analyzed the security flaw of Pareek et al.'s cryptosystem and presented a known plaintext attack method to attack it. Furthermore, Wei et al. remedied

Pareek et al.'s cryptosystem so that the improved cryptosystem can resist the proposed attack. In this subsection, we compare our cryptosystem with Pareek et al.'s cryptosystem and its improved version.

5.1.1. Compared with Pareek et al.'s cryptosystem

Wei et al. [10] pointed out that in Pareek et al.'s cryptosystem, the keystream sequence depended only on the secret key, but not on the plaintext. The plaintext-independent keystream caused Pareek et al.'s cryptosystem vulnerable to known plaintext attack. More information about this security flaw can be obtained in literature [10]. We can observe from Equations (14)-(17) of this paper that plaintext feedback is used in our proposed cryptosystem. Therefore, the keystream sequence of our cryptosystem depends on both the secret key and the plaintext. Consequently, our proposed cryptosystem can resist the known plaintext attack proposed by Wei et al. successfully.

5.1.2. Compared with improved version of Pareek et al.'s cryptosystem

Wei et al. [10] improved Pareek et al.'s original cryptosystem. In the improved cryptosystem, keystream sequence depends on both the secret key and the plaintext. Consequently, the improved version of Pareek et al.'s cryptosystem can resist the known plaintext attack proposed by Wei et al.. More information about this improved cryptosystem can be obtained in literature [10]. In this subsection, we compare our proposed cryptosystem with the improved version of Pareek et al.'s cryptosystem in terms of speed and flexibility.

In the improved version of Pareek et al.'s cryptosystem, before encrypting every plaintext block, the cryptosystem iterates the selected chaotic map many times, usually larger than 100. Although so many iterations can enhance the security of the cryptosystem, they cost a lot of time and limit the speed of the cryptosystem. In our proposed cryptosystem, by controlling the value of the integer I, which is used in Equations (9) and (16), we can control the number of iterations applied in map f_{mask}. The value of I is usually less than 50. Although the number of iterations of our proposed cryptosystem is less than that of the improved Pareek et al.'s cryptosystem, by using the map-to-task table and

randomly selecting encryption operations, the proposed cryptosystem can also achieve desirable encryption result and high security (see Section 3). Consequently, the speed of our proposed cryptosystem is higher than that of the improved Pareek et al.'s cryptosystem.

The following Table 5 presents the comparison between the proposed cryptosystem and the improved Pareek et al.'s cryptosystem in terms of encryption speed. The experimental conditions are: CPU: Pentium-IV 1.40GHz Celeron; Memory: 256MB; Hard-disk capacity: 80GB; Operation system: Windows XP. We can observe from Table 5 that the encryption speed of the proposed cryptosystem is higher than that of the improved Pareek et al.'s cryptosystem.

Table 5. Encryption speed of the proposed cryptosystem and the improved Pareek et al.'s cryptosystem

File type	Size of plaintext (KB)	Encryption time of the proposed cryptosystem Min-Max (Mean)	Encryption time of the improved Pareek et al.'s cryptosystem Min-Max (Mean)
Text file (*.txt)	30	0.094s-0.125s (0.113s)	0.148s-0.296s (0.201s)
	90	0.266s-0.292s (0.271s)	0.353s-0.520s (0.402s)
	240	0.672s-0.703s (0.682s)	0.899s-1.907s (1.283s)
Document file (*.doc)	30	0.109s-0.135s (0.118s)	0.150s-0.299s (0.207s)
	90	0.265s-0.281s (0.274s)	0.369s-0.526s (0.439s)
	240	0.688s-0.704s (0.695s)	1.002-1.976s (1.349s)
Image file (*.bmp)	192	0.593s-0.625s (0.606s)	0.753s-1.199s (1.147s)
	768	2.172s-2.403s (2.189s)	3.198s-4.135s (3.802s)
	3072	8.469s-8.916s (8.491s)	10.429s-15.98s (12.964s)
Music file (*.mp3)	1382	3.859s-3.922s (3.878s)	5.423s-9.893s (7.885s)
	2437	6.656s-6.692s (6.663s)	9.264s-15.301s (12.101s)

Of course, we can make the value of I bigger to achieve higher security at the expense of some efficiency. Integer I is a part of the secret key of our proposed cryptosystem, and its value can be easily controlled by the users of our proposed cryptosystem according to their need. By controlling the value of I, the speed and security level of the proposed cryptosystem can be easily customized by the users of the proposed cryptosystem. In this respect, our proposed cryptosystem is more flexible than the improved Pareek et al.'s cryptosystem.

5.2. Compared with traditional cryptography

A lot of traditional cryptosystems, such as DES, RSA, and AES, have been proposed in recent decades. Our proposed cryptosystem is a chaotic block cryptosystem instead of a traditional block cryptosystem. Compared with traditional block cryptosystems, such as DES and AES, the proposed cryptosystem has some advantages.

(i) Large key space

At present, the key space of a secure cryptosystem should be larger than 2^{100} in order to resist brute-force attack effectively. The key space of DES is 2^{56}. Obviously, the key space of DES is not large enough, which is a disadvantage of DES. In 1997, DES (56bit) was broken by brute-force attack. In 1998, a special computer named "Deep Crack" broke DES successfully in only 56 hours. The key space of AES can be as large as 2^{256}.

The secret key of the proposed cryptosystem is mainly composed of some double-floating numbers. These double-floating numbers possess high precision, which makes the proposed cryptosystem possess large key space. The key space of our proposed cryptosystem is larger than 10^{90} (see Subsection 3.5). We can see that the key space of the proposed cryptosystem is even larger than that of AES. Therefore, the proposed cryptosystem can resist brute-force attack effectively.

(ii) Simple design and realization

The design of the proposed cryptosystem is simple. The proposed cryptosystem is mainly composed of some simple one-dimensional chaotic maps and a simple map-to-task table. Therefore, the proposed cryptosystem can be designed and realized easily. The map-to-task table used in the proposed cryptosystem can randomly change the tasks of the chaotic maps, which can enhance the confusion effect of the proposed cryptosystem and avoid the affection of dynamical degradation of single simple one-dimensional chaotic map. That is to say, the existence of the map-to-task table enhances the security of the proposed cryptosystem without loosing the simplicity of the cryptosystem.

Most traditional block cryptosystems, such as DES and AES, use S-box (substitution box, S-box). A good S-box must comply with some standards in order to achieve desirable encryption result. Therefore, an S-box must be designed carefully to insure the security of the traditional block cryptosystem, which use this box. Consequently, the design and realization of these traditional cryptosystems will need more effort.

(iii) Flexibility

As we can see in Subsection 5.1.2, the user of the proposed cryptosystem can customized the balance point between the speed and security level of the proposed cryptosystem by controlling the value of integer I, which is a part of the secret key. This flexibility is an advantage of the proposed cryptosystem, which most traditional cryptosystems do not have.

The chaotic maps used by the proposed cryptosystem are not fixed, i.e., the actual equations corresponding to maps f_1, f_2, f_3, f_4, and f_5 are not fixed. We can flexibly choose different chaotic maps for the proposed cryptosystem according to our need. We can observe from Section 4 of this paper that by slightly changing the usage of the map-to-task table, the proposed cryptosystem can be improved easily. That is to say, the map-to-task table can also make the proposed cryptosystem more flexible and extensible.

6. Conclusion

In this paper, the authors propose a new chaotic block cryptosystem based on multiple one-dimensional chaotic maps. The proposed cryptosystem applies map-to-task relations in the encryption process. The authors enhance the security of the cryptosystem by changing the map-to-task relationships constantly and applying plaintext and ciphertext feedback in the encryption process. Experiments and theoretic analysis show that the proposed cryptosystem possesses high performance and security. Some improvement schemes of the proposed cryptosystem are proposed too. The authors also compare the proposed cryptosystem with other two chaotic cryptosystems, which are also based on multiple one-dimensional chaotic maps. Moreover, the authors compare the proposed cryptosystem with traditional cryptography. These comparisons show that the proposed cryptosystem possesses some advantages, such as high speed, simple realization, and flexibility. With these desirable properties, the proposed cryptosystem is suitable for practical use such as the secure transmission of secret files over public data communication network.

Acknowledgement

This research is supported by the National Natural Science Foundation of China (Nos. 60573172, 60973152), the Superior University Doctor Subject Special Scientific Research Foundation of China (No. 20070141014), and the Natural Science Foundation of Liaoning province (No. 20082165).

References

[1] G. Álvarez, F. Montoya and M. Romera et al., Cryptanalysis of an ergodic chaotic cipher, Physics Letters A 311(2-3) (2003), 172-179.

[2] M. S. Baptista, Cryptography with chaos, Physics Letter A 240(1) (1998), 50-54.

[3] F. J. Huang and Z. H. Guan, A modified method of a class of recently presented cryptosystems, Chaos, Solitons and Fractals 23(5) (2005), 1893-1899.

[4] S. J. Li, X. Q. Mou and Y. L. Cai et al., On the security of a chaotic encryption scheme: Problems with computerized chaos in finite computing precision, Computer Physics Communications 153(1) (2003), 52-58.

[5] N. K. Pareek, V. Patidar and K. K. Sud, Cryptography using multiple one-dimensional chaotic maps, Commun. Nonlinear Sci. Numer. Simul. 10(7) (2005), 715-723.

[6] X. Y. Wang, Chaos in the Complex Nonlinearity System, pp. 1-68, Electronics Industry Press, Beijing, 2003.

[7] X. Y. Wang and X. J. Wang, Design of chaotic pseudo-random bit generator and its applications in stream-cipher cryptography, International Journal of Modern Physics C 19(5) (2008), 813-820.

[8] X. Y. Wang and Q. Yu, A block encryption algorithm based on dynamic sequences of multiple chaotic systems, Commun. Nonlinear Sci. Numer. Simul. 14(2) (2009), 574-581.

[9] X. Y. Wang and C. H. Yu, Cryptanalysis and improvement on a cryptosystem based on a chaotic map, Comput. Math. Appl. 57(3) (2009), 476-482.

[10] J. Wei, X. F. Liao and K. W. Wong et al., Cryptanalysis of a cryptosystem using multiple one-dimensional chaotic maps, Commun. Nonlinear Sci. Numer. Simul. 12(5) (2007), 814-822.

[11] W. K. Wong, L. P. Lee and K. W. Wong, A modified chaotic cryptographic method, Computer Physics Communications 138(3) (2001), 234-236.

[12] K. W. Wong, A combined chaotic cryptographic and hashing scheme, Physics Letters A 307(5-6) (2003), 292-298.

■