ORIGINAL ARTICLE

A pseudorandom number generator based on piecewise logistic map

Yong Wang · Zhaolong Liu · Jianbin Ma · Haiyuan He

Received: 22 April 2015 / Accepted: 3 November 2015 / Published online: 19 November 2015

Abstract In order to overcome the disadvantages of logistic map in designing chaos-based cipher, the piecewise logistic map (PLM) is presented. Some properties related to cryptography of the PLM, such as ergodicity, Lyapunov exponent, and bifurcation, are analyzed and compared with the logistic map. From the view of cryptography, the PLM owns better properties than the logistic map. Then, a novel pseudorandom number generator (PRNG) based on the PLM is proposed. Since the cryptographic properties of the PLM are enhanced, the presented PRNG achieves a tradeoff between efficiency and security. Both performance analysis and simulation test confirm that our scheme is simple, secure, and efficient, with high potential to be adopted as a stream cipher for secure communication.

Keywords Piecewise logistic map · Pseudorandom number generator · Chaos · Secure communication

Y. Wang (✉) · Z. Liu · J. Ma
College of Computer Science and Technology,
Chongqing University of Posts and Telecommunications,
Chongqing 400065, China
e-mail: wangyong_cqupt@163.com

Y. Wang · H. He
Key Laboratory of Electronic Commerce and Logistics
of Chongqing, Chongqing University of Posts
and Telecommunications, Chongqing 400065, China

1 Introduction

Over the past 10 years, the chaotic map has been attracting more and more interests from researchers in the field of cryptography. It has been widely applied to secure communication and encryption algorithms. In the encryption algorithm, the chaotic map is usually used as the core component to generate pseudorandom sequences. Then, the plain message is masked or encrypted by the sequences from chaotic maps, which is the general idea of designing chaos-based cipher [1–7]. Thus, the cryptographic properties of the sequences generated from chaotic map are very important to the security of encryption algorithms. The logistic map is one of the popular chaotic map used in chaos-based cryptography, which has been widely used in block cipher [6,7], stream cipher [8,9] and Hash function [10,11]. Although logistic map owns some advantages from chaotic system, it still has some inherent problem from the view of cryptography, such as uneven density probability distribution. The security problems of logistic map are presented and some encryption schemes based on logistic map are cryptanalyzed [12,13]. Thus, it is necessary to improve the cryptographic properties of logistic map. In this paper, the enhanced form of logistic map, i.e., the piecewise logistic map (PLM) is presented. Some cryptographic properties of this chaotic map are analyzed by numeric method and compared with the logistic map. The results show that the PLM has better cryptographic performance than the logistic map. Furthermore, to improve the density probability distribution of the PLM, the piecewise logistic map with variable parameter (PLMVP) is suggested. These researches on enhancing the cryptographic performance of the chaotic map benefit the designing of pseudorandom number generator and encryption algorithm.

In stream cipher, the pseudorandom number generator (PRNG) is one of the most important components. Recently, some PRNGs are proposed based on chaotic systems [14–18]. To achieve high speed, a simple chaotic map, such as tent map or logistic map, is iterated to generate pseudorandom numbers. Although one-dimensional chaotic system has the advantages of high-level efficiency and simplicity, there are fundamental drawbacks in this chaotic cryptosystem, such as small key space and weak security [12,19]. Most of the PRNGs based on chaos are obtained directly by sampling the trajectory of the chaotic map. In this case, some information of the chaotic map is probably exposed, which leads to some loopholes. Moreover, methods for predicting chaotic time series have been suggested [20–22]. To prevent attackers from breaking the PRNG by predicting the chaotic series, complex chaotic system should be considered. In Refs. [23,24], high-dimensional chaos and spatiotemporal chaos are applied to design PRNG. Owing to making full use of the traits of a complex system, the algorithms satisfy the security requirement of PRNG. However, more computation is needed in these algorithms, when producing pseudorandom numbers. Therefore, further studying PRNG on the trade-off between efficiency and security is needed.

In this paper, the PRNG is proposed based on the PLM, which is an enhanced version of logistic map. To break the relation between the chaotic sequences and the pseudorandom numbers, some special operations, such as substitution and feedback, are employed in our scheme. Theory analysis and simulation tests both confirm that the proposed PRNG is simple, secure, and efficient. The rest of the paper is organized as follows. Section 2 analyzes the logistic map and its cryptographic properties. In Sect. 3, the PLM is defined and its properties related to cryptography are discussed. The PLMVP with better uniform density probability is presented in Sect. 4. The new PRNG and its performance are discussed in Sects. 5 and 6, respectively. Finally, conclusion is drawn in Sect. 7.

2 Logistic map

The logistic map is a simple chaotic map often used to describe the growth of biological population. The logistic system has very complex dynamic behavior, which has been widely applied to data security and secure communication [3,4,25]. The logistic map is a discrete-time dynamic system, being mathematically expressed as

$$x_{n+1} = f(x_n) = \mu x_n (1 - x_n) \tag{1}$$

where $x_0 \in (0, 1)$ is the state value and μ is the control parameter. When $\mu \in [3.57, 4]$, the logistic map is in chaotic status.

2.1 Ergodicity of logistic map

In statistics, the ergodicity describes a random process for which the time average of one sequence of events is the same as the ensemble average. The ergodicity can be expressed by the statistics distribution of state value. In Fig. 1, the logistic map is plotted for different μ. It can be seen from Fig. 1 that the maximum state value of the logistic map is smaller than 1 when $\mu < 4$. Moreover, the maximum state value of logistic map becomes smaller with the decrease of μ. It means that the ergodicity of logistic map in interval [0, 1] becomes better with the increase of μ.

To further demonstrate the ergodicity changes of logistic map with the control parameter, the following numeric experiment is done:

(1) Fix the value of control parameter μ and randomly set the initial value x_0.
(2) Iterate the logistic map for 10,000 times.
(3) The distribution of the logistic state values is plotted.

The experiment is repeated for different μ and all results are shown in Fig. 2. According to Fig. 2, we can see that the logistic map has the best uniform distribution and ergodicity when μ is 4. The results as shown in Fig. 2 also confirm that the ergodicity of logistic map in interval [0, 1] becomes better with the increase of μ.

2.2 Bifurcation of logistic map

In the dynamic system, the bifurcation diagram shows the qualitative changes of a system as a map of parameter. The period-doubling bifurcation of the logistic map is depicted in Fig. 3. Since the map is surjection in the

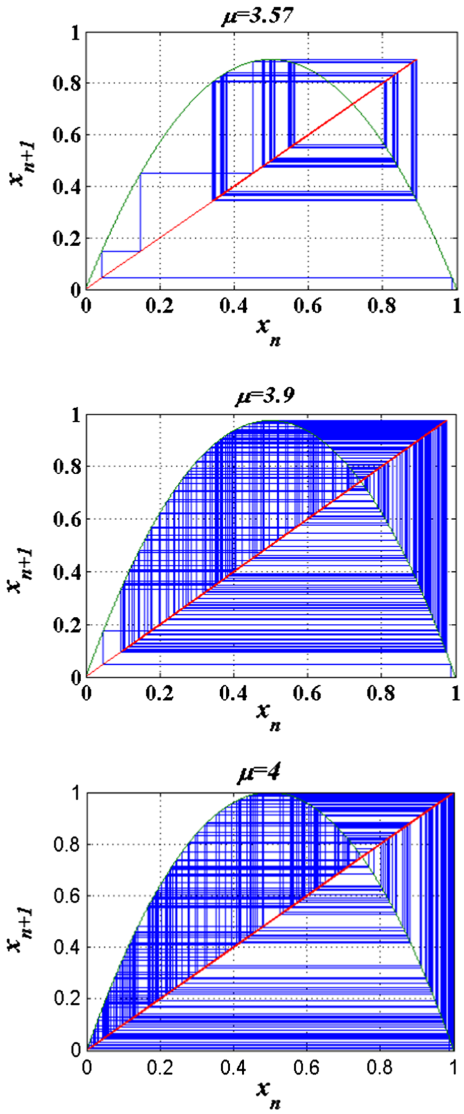


Fig. 1 Chaotic attractors of the logistic map for different μ

intervals (0, 1), the separation speed of period-doubling orbit of the logistic map is the fastest when μ is 4.

2.3 Lyapunov exponent of logistic map

For dynamical systems, the Lyapunov exponent characterizes the velocity of evolution between two near trajectories. For function $x_{n+1} = f(x_n)$, its Lyapunov exponent λ is defined as follows [26–28]

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_n)| \tag{2}$$

A quantitative measurement of chaotic behavior of the function is given by the positive values of Lyapunov exponent (i.e., $\lambda > 0$). The Lyapunov exponent of the logistic map is calculated by quantitative method under the condition that the control parameter μ is changed from 3 to 4 with step 0.01. The distribution of Lyapunov exponents is shown in Fig. 4. Based on Fig. 4, it can be seen that the Lyapunov exponent of the logistic map begins to be greater than 0 when $\mu > 3.57$. Meanwhile, there are some points less than 0 when $\mu \in [3.57, 3.9]$. Most Lyapunov exponents become positive and their values are greater when $\mu \in [3.9, 4]$. Thus, the logistic map has more stable chaotic behavior when $\mu \in [3.9, 4]$.

2.4 Density probability of logistic map

The density function of logistic map for $\mu = 4$ is illustrated by numeric simulation in Fig. 5 [29]. Obviously, the density distribution of the logistic map is not uniform, which means that the sequence directly generated by this chaotic map probably has no enough good random properties.

3 The PLM

According to the analysis in Sect. 2, the logistic map has some potential problems, such as mediocre ergodicity and uneven density probability, when it is applied to designing encryption algorithms. In order to further improve the cryptographic properties, we present an enhanced version of the logistic map, i.e., the PLM, which is defined by

$$\begin{aligned} x_{j+1} &= \mathrm{PLM}(x_j) \\ &= \begin{cases} N^2\mu x_j\left(\frac{1}{N} - x_j\right), & 0 < x_j < \frac{1}{N} \\ 1 - N^2\mu\left(x_j - \frac{1}{N}\right)\left(\frac{2}{N} - x_j\right), & \frac{1}{N} < x_j < \frac{2}{N} \\ \vdots & \\ N^2\mu\left(x_j - \frac{i-1}{N}\right)\left(\frac{i}{N} - x_j\right), & \frac{i-1}{N} < x_j < \frac{i}{N} \\ 1 - N^2\mu\left(x_j - \frac{i}{N}\right)\left(\frac{i+1}{N} - x_j\right), & \frac{i}{N} < x_j < \frac{i+1}{N} \\ N^2\mu\left(x_j - \frac{N-2}{N}\right)\left(\frac{N-1}{N} - x_j\right), & \frac{N-2}{N} < x_j < \frac{N-1}{N} \\ 1 - N^2\mu\left(x_j - \frac{N-1}{N}\right)(1 - x_j), & \frac{N-1}{N} < x_j < 1 \\ x_j + \frac{1}{100N}, & x_j = 0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N-1}{N} \\ x_j - \frac{1}{100N}, & x_j = 1 \end{cases} \end{aligned} \tag{3}$$

where $x_j \in (0, 1)$ is the state value of PLM, $\mu \in (0, 4]$ is the control parameter, and N is the segment number of the PLM. For example, when $N = 4$ and $\mu = 4$, the PLM can be expressed in the form

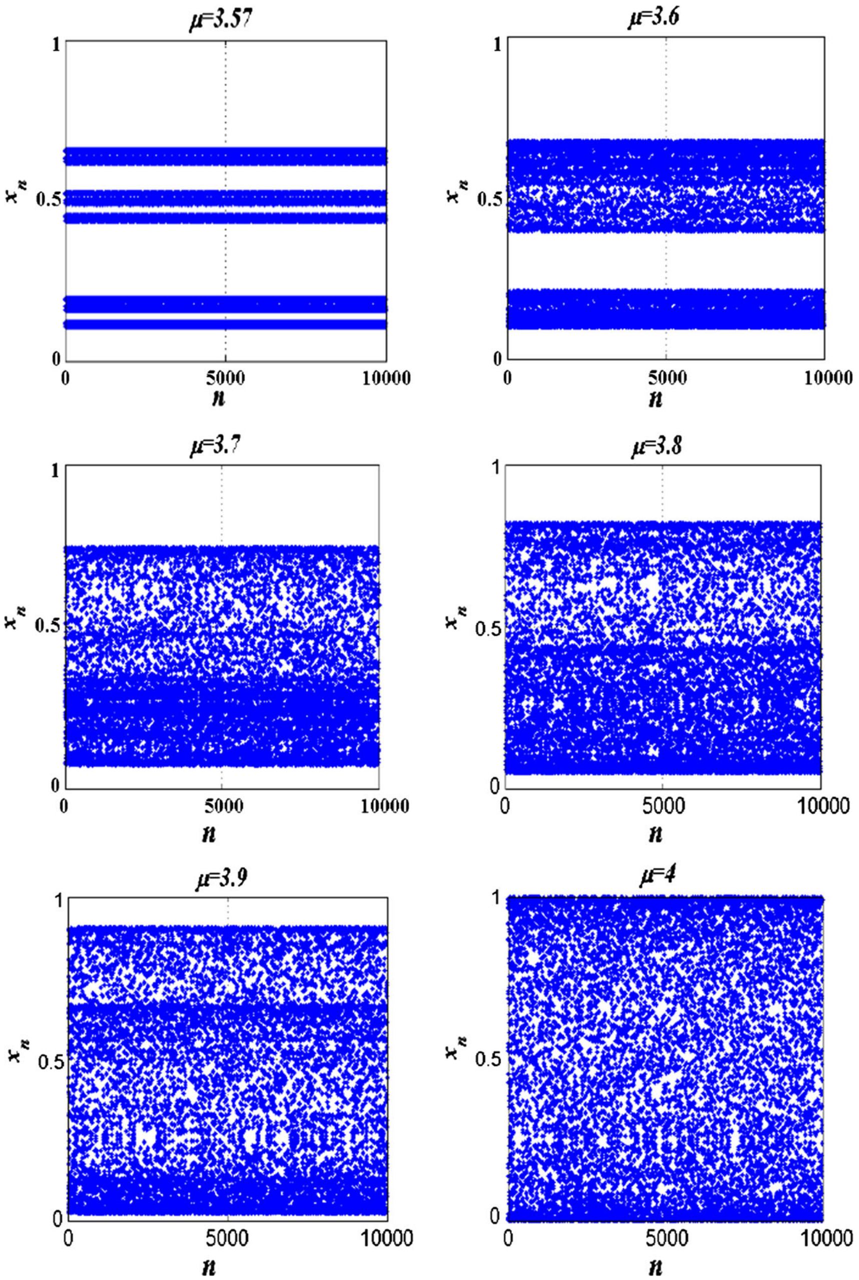


Fig. 2 The state value distribution of logistic map for different μ

$$x_{j+1}=\mathrm{PLM}(x_j)=\begin{cases} 64x_j\left(\frac{1}{4}-x_j\right), & 0<x_j<\frac{1}{4}\\ 1-64\left(x_j-\frac{1}{4}\right)\left(\frac{1}{2}-x_j\right), & \frac{1}{4}<x_j<\frac{1}{2}\\ 64x_j\left(x_j-\frac{1}{2}\right)\left(\frac{3}{4}-x_j\right), & \frac{1}{2}<x_j<\frac{3}{4}\\ 1-64x_j\left(x_j-\frac{3}{4}\right)(1-x_j), & \frac{1}{4}<x_j<1\\ x_j+\frac{1}{400}, & x_j=0,\frac{1}{4},\frac{1}{2},\frac{3}{4}\\ x_j-\frac{1}{400}, & x_j=1 \end{cases} \tag{4}$$

and plotted in Fig. 6.

To compare the PLM with the logistic map, we have performed the following test: For the PLM with fixed $\mu = 4$, set $N = 4$ and randomly select an initial state value of PLM. Then, the PLM is iterated for 30 times. Finally, the state values are plotted in Fig. 7. Repeat performing this test with the same initial value but different N. Moreover, the test result of the logistic map is also illustrated in Fig. 7. It can be seen from Fig. 7 that the sequences generated by the PLM and the logis-

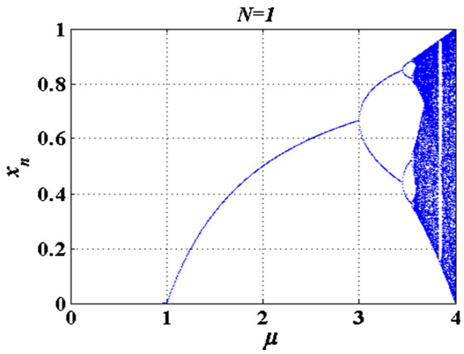


Fig. 3 Bifurcation diagram of the logistic map

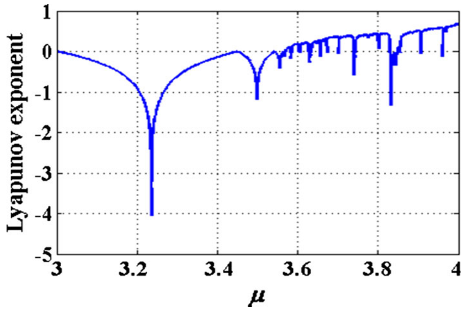


Fig. 4 The Lyapunov exponents of the logistic map

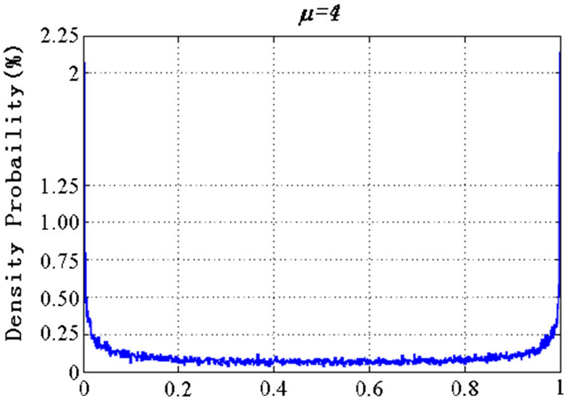


Fig. 5 Density probability distribution of the logistic map

tic map are total different. Furthermore, we change the initial state value and repeat the comparison mentioned above. The same result is obtained, which means that the PLM is not related to the logistic map and it is a new map.

3.1 Ergodicity of the PLM

Here, we give some simulation test to show the ergodicity changes of the PLM. The curves of the PLM for different μ are shown in Fig. 8, where $N = 4$. It can be seen that the PLM occupies the whole intervals (0, 1) when $\mu \in [2, 4]$. As we know from Fig. 1, the logistic map can achieve the same result when μ is only 4. Moreover, the state value distributions of the PLM with different μ and N are obtained by the same method in Sect. 2.1. They are shown in Fig. 9. According to Fig. 9, it can be concluded that: (1) The sequences generate by the piecewise logistic map with different N has good randomness when $\mu \in [2, 4]$. (2) Compared with Fig. 2, the value range of μ corresponding to good ergodicity is extended with the increase of N, which means that the ergodicity of PLM is improved.

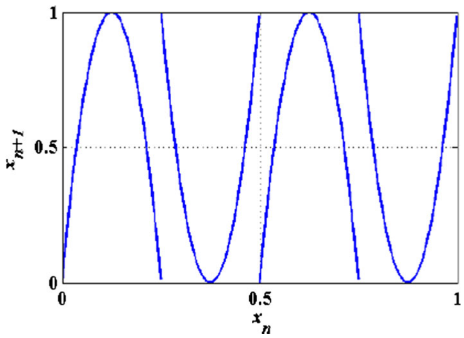


Fig. 6 The piecewise logistic map when $N = 4$ and $\mu = 4$

3.2 Bifurcation of the PLM

In this section, we discussed the period-doubling bifurcation of the PLM with different N. With numeric method, the bifurcation diagrams of the PLM for different N are plotted in Fig. 10, where the initial value is a random value. In Fig. 10, the control parameter μ is shown on the horizontal axis and the vertical axis shows the possible long-term status values of PLM. It is clear by comparing Fig. 10 with Fig. 3 that the value of μ corresponding to the starting point of the PLM falling into chaos is smaller than that of logistic map. Furthermore, the value of μ corresponding to the starting point of the PLM being chaotic becomes smaller with the increase of N.

3.3 Lyapunov exponent of the PLM

Since μ and N both affect the Lyapunov exponent, only one parameter is varied each time with the other parameter fixed, so as to observe the influence of the varying

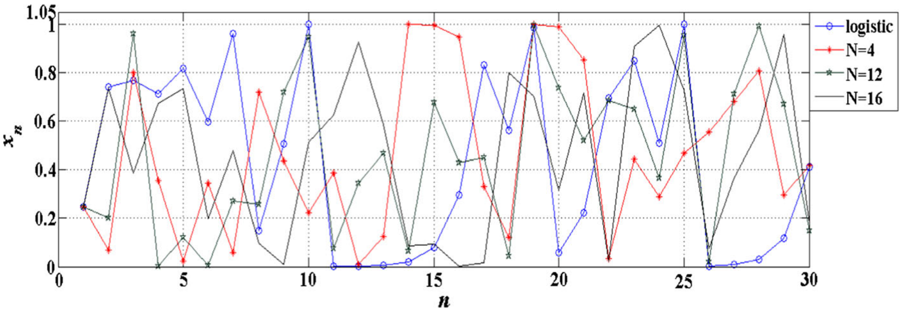


Fig. 7 The state values of the PLM and the logistic map

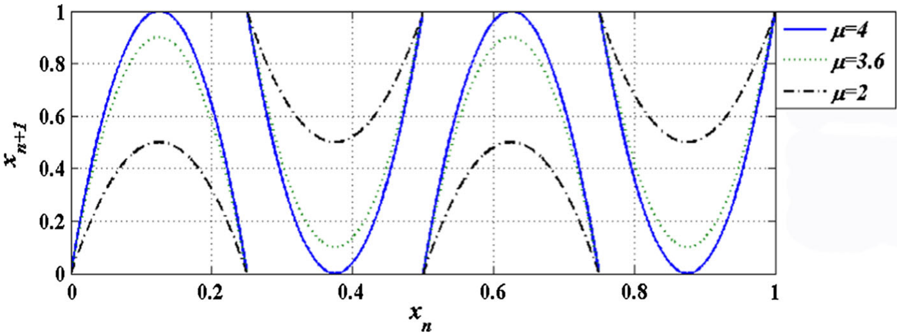


Fig. 8 The curves of the PLM for different μ

parameter on the Lyapunov exponent. The Lyapunov exponents with various μ or N are shown in Figs. 11 and 12, respectively.

According to the result of numerical analysis in Figs. 4, 11 and 12, some conclusions are drawn below:

(1) The Lyapunov exponent of the PLM is increased with the increase of N when μ is fixed.
(2) With the increase of N, the interval of μ corresponding to the positive Lyapunov exponent becomes wider.
(3) Compared with the logistic map, the PLM has greater Lyapunov exponent when the parameter μ is set as the same value.

3.4 The density probability of the PLM

By the method in the Sect. 2.4, we analyze the density probability of the PLM for different N and μ. All of them have almost the same result as the logistic map when μ is 4. Moreover, the density probability of the PLM is also not uniform, and the intervals corresponding to the higher density probability are different when μ is changed. Here, we only give the density probability of the PLM with $N = 64$ and $\mu = 1, 2, 3, 4$ as an example, which is shown in Fig. 13.

4 The PLM with variable μ

Since the density distribution of the PLM is not uniform, the numbers directly generated by this chaotic map is also uneven. It means that the sequences from the PLM probably has no good enough random properties. To remedy this problem, we present a PLMVP, whose parameter μ is changed when each iteration of the piecewise logistic map is completed. The PLMVP is defined as $h(x_0, \mu_0, s, m)$, where $x_0 \in (0, 1)$ and $\mu_0 \in (m, 4-m)$ are the initial state value and the initial control parameter value, respectively; s is the step value of changing control parameter μ; m is the parameter used to control the scope of μ. To get good chaotic properties, we suggest $m \in (0.01, 0.1)$. The pseudo codes in Fig. 14 illustrate the iteration process of the PLMVP. By controlling the condition in the pseudocodes, we can obtain as many chaotic state values from PLMVP as we want.

The density probability of the PLMVP is got and shown in Fig. 15. Here, x_0 and μ_0 are randomly set and s, m are 0.001 and 0.01, respectively. By comparing Fig. 15 with Figs. 5 and 13, it is obvious that the PLMVP has much more uniform density probability distribution than that of PLM and logistic map.

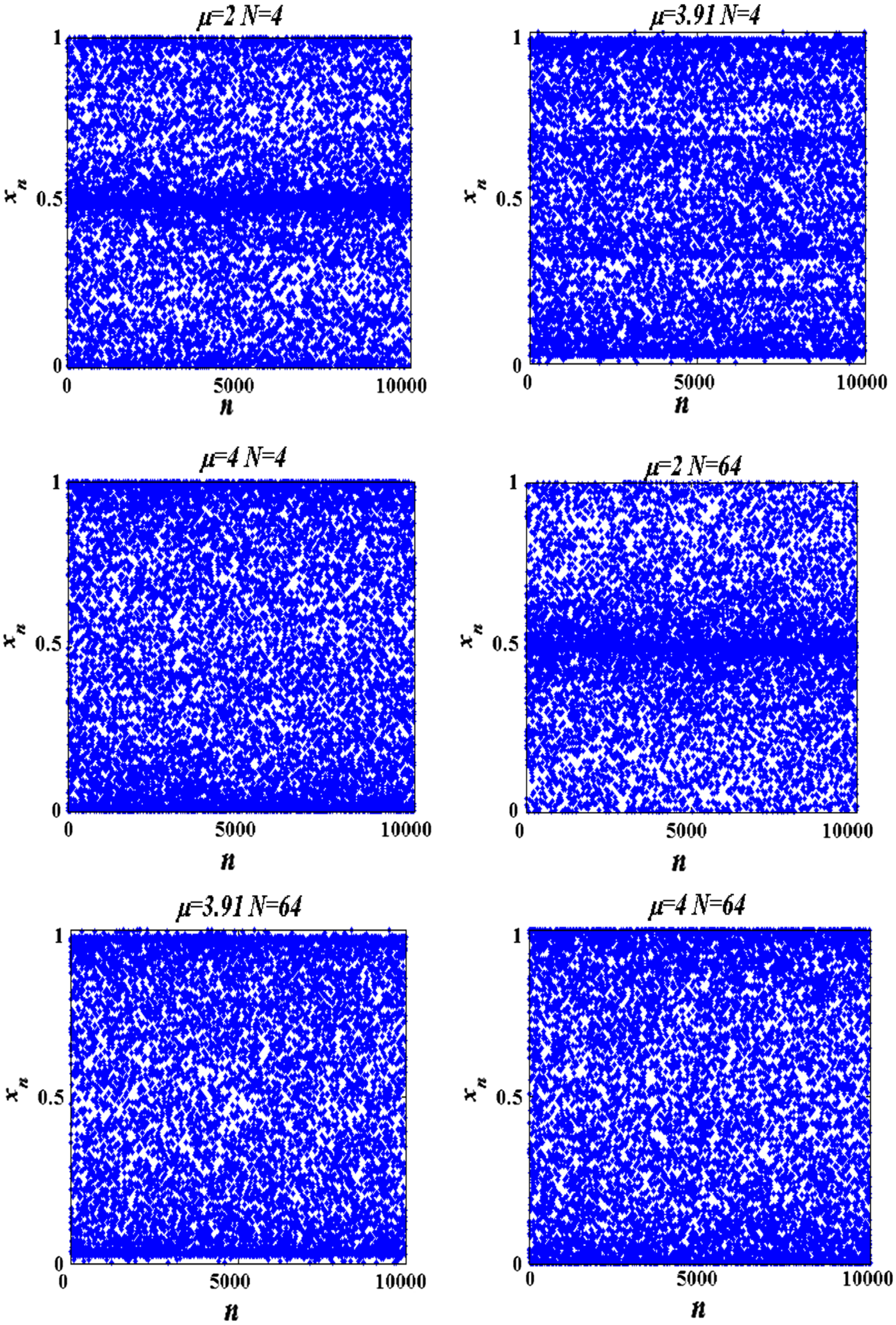


Fig. 9 The distribution of state value of piecewise logistic map

5 The proposed pseudorandom number generator

5.1 Notations

A pseudorandom number generator (PRNG), also known as a deterministic random number generator, is an algorithm or a device for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers [8]. The PRNG is very important in applications such as simulation electronic games and cryptography. We will present our PRNG algorithm based on piecewise logistic map. To

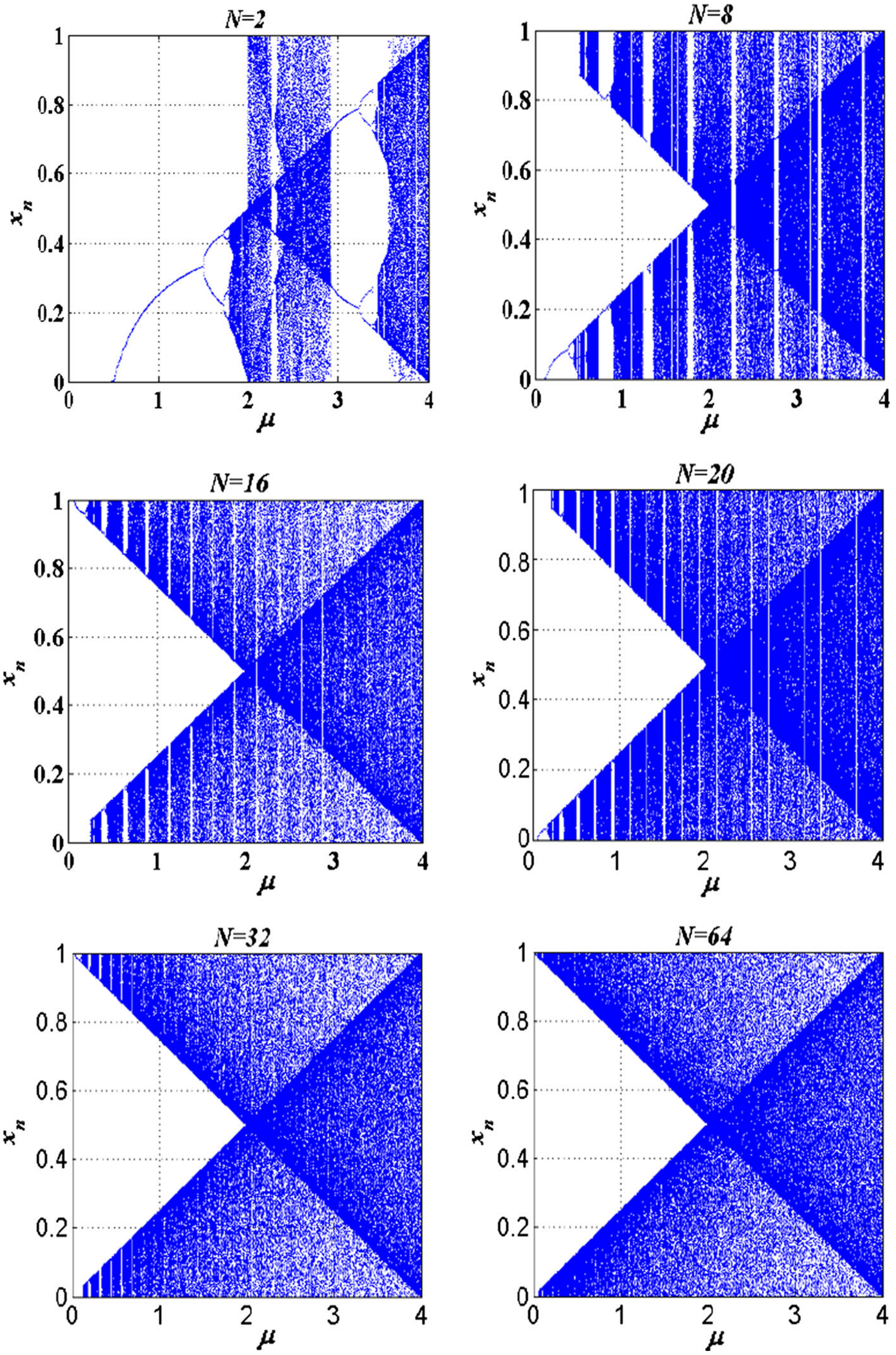


Fig. 10 Bifurcation diagrams of the PLM

the convenience of description and understanding, the notations in our algorithm are firstly listed in Table 1.

5.2 The algorithm

Cryptographic applications require the output not to be predictable from earlier outputs. Thus, the sequence generated by any PRNG must have the following characteristics: (1) The output of the PRNG has good statistical properties; (2) For any initial values, the PRNG generated the sequence has no shorter periods; (3) The correlation of successive values is poor in the sequence. Base on the PLM, a novel pseudorandom number generator is proposed. The whole process of generating

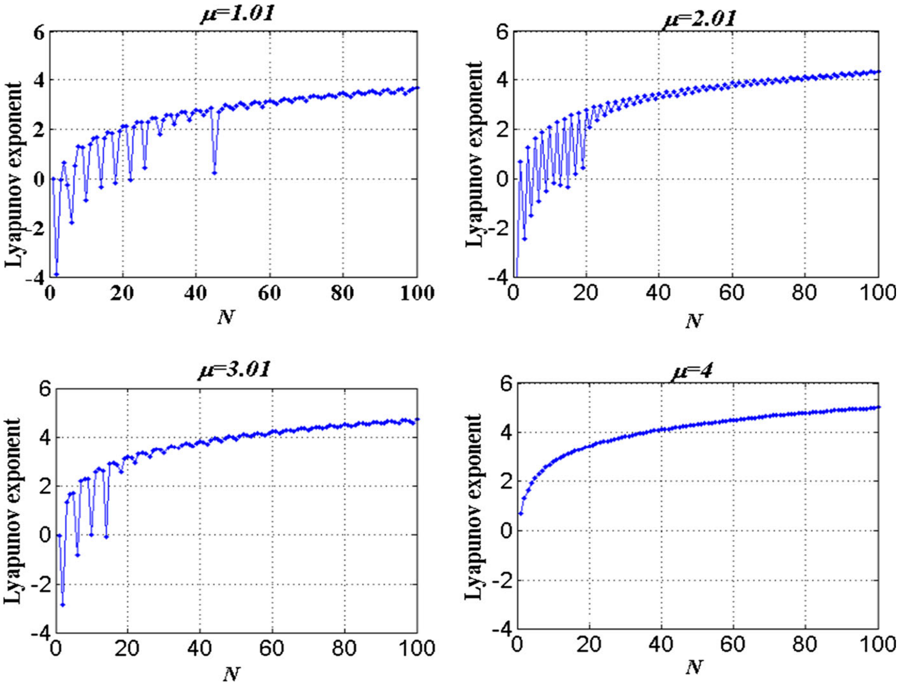


Fig. 11 Lyapunov exponents with variable N

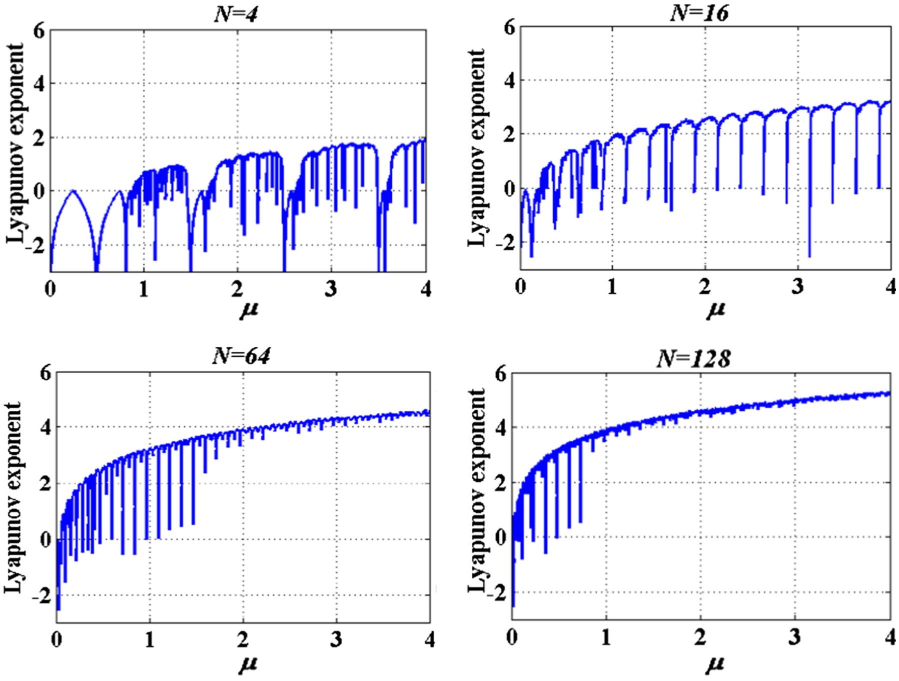


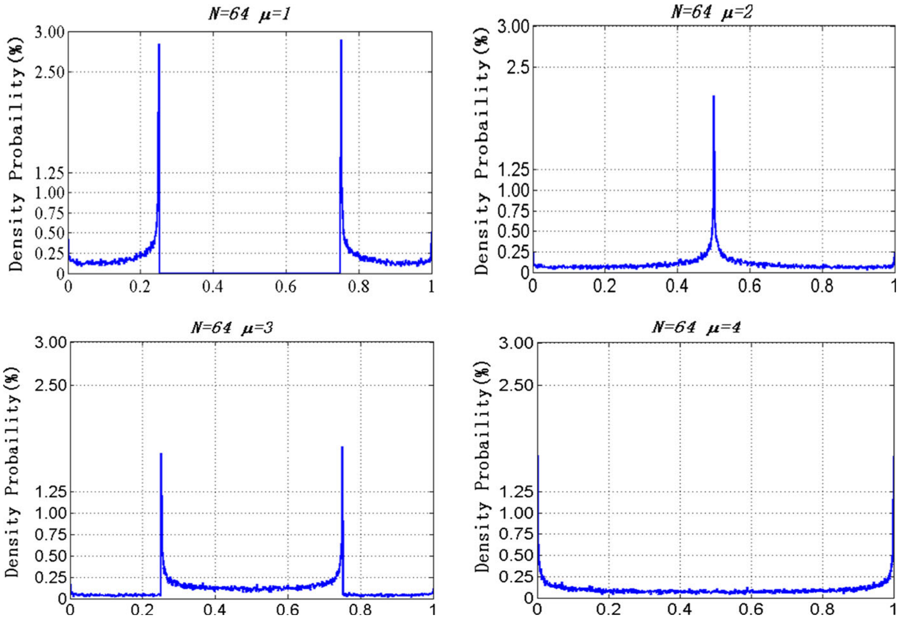
Fig. 12 Lyapunov exponents with variable μ

pseudorandom numbers is described as follows and an illustration is given in Fig. 16.

Step 1 Choose the PLM with $N = 64$ as the chaotic map to generate pseudorandom numbers. Then, initialize the control parameter μ of the PLM and set the initial values of parameters R and m.

Step 2 Iterate the PLM once. Then, transform the current status value of PLM to the corresponding binary format and extract 8 bits (9th to 16th

Fig. 13 Density probability of the PLM



```
Initialization: $x = x_0$, $\mu = \mu_0$
While (condition) do
$x \leftarrow PLM(x)$
$\mu \leftarrow \mu + s$
If $\mu > 4 - m$ then
        $\mu \leftarrow \mu - (4 - m)$
    End If
    If $\mu < m$ then
            $\mu \leftarrow m$
    End If
End While
```

Fig. 14 The pseudocode for implementing the PLMVP

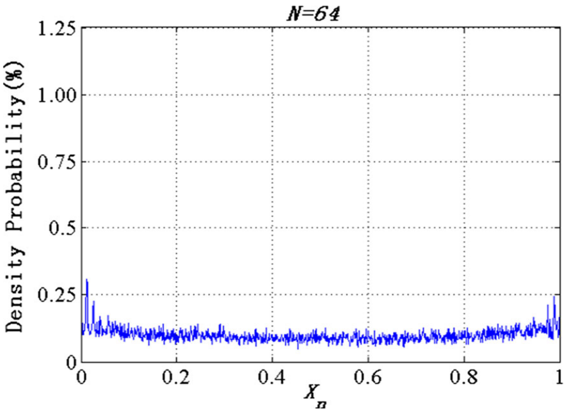


Fig. 15 Density probability distribution of piecewise logistic map

bits after the decimal point) to obtain an integer K.

Step 3 Generate an 8-bit output number according to the following equation

$$P = K \oplus S(R) \tag{5}$$

where function $S(x)$ returns the substitution value of x according to the S-box of AES [30], which is expressed in hexadecimal form in Table 2.

Step 4 Update the value of register R, i.e., $R = K$.

Step 5 By using P as a feedback, adjust the value of μ according to the following steps

(i) $\mu = \mu + P/256$
(ii) if $\mu > (4 - m)$ then $\mu = \mu - (4 - m)$
(iii) if $\mu < m$ then $\mu = m$

Step 6 To get rid of the transient effect of the chaotic map, the first 1024 generated numbers is omitted. If enough pseudorandom numbers have already been generated, stop this pseudorandom number generator; otherwise, go to Step 3 to generate the next 8-bit pseudorandom number.

Table 1 Notations in the proposed algorithm

Symbols	Meanings	Symbols	Meanings
PLM	Piecewise logistic map in Eq. (3)	μ	$\mu \in (0, 4]$ is the control parameter of PLM
N	The segment number of the PLM	K	8-Bit integer value extracted from PLM
R	8-Bit integer value of register	m	A floating-point number used to adjust μ
$S(R)$	The substitution value of R	P	8-Bit output number after each iteration of PLM

Fig. 16 Scheme of the proposed PRNG

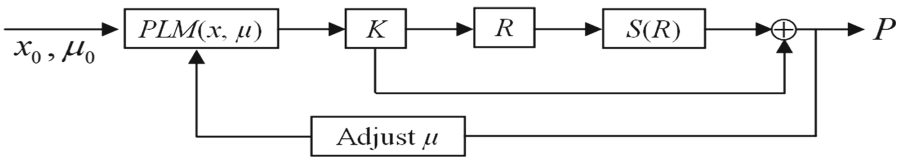


Table 2 AES S-box: substitution values for byte *xy* (in hexadecimal format)

x	*Y*															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	fl	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	la	lb	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	dl	00	ed	20	fc	bl	5b	6a	cb	be	39	4a	4c	58	cf
E	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	be	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	3d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	lc	a6	b4	c6	e8	dd	74	If	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	cl	Id	9e
e	el	f8	98	11	69	d9	8e	94	9b	le	87	e9	ce	55	28	df
f	3c	al	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

6 Performance analysis of the proposed PRNG

6.1 Advantages of the scheme

The PLM is the core component in the proposed PRNG. According to the analysis in Sects. 2 and 3, we may conclude that using the PLM can bring more advantages than using the logistic map. Firstly, the Lyapunov exponent of PLM is greater than that of logistic map when μ is the same. It means that the separation rate of the PLM trajectories is larger than that of logistic trajectories. Thus, compared with the logistic map, the PLM is more sensitive to the status value and parameter. The orbit of the PLM is more unstable and its sequence is more complex. Obviously, these properties of the PLM are very useful to designing PRNG.

In the second place, the PLM has wider interval of μ corresponding to its chaotic status. As we know, the control parameter is usually employed as the secrete key in the chaos-based cipher. Therefore, the PLM owns larger key space.

Finally, to overcome the problem of uneven density probability, the operation of adjusting μ is specially added in our scheme. The value of μ is changed after the

PLM finishes each iteration, which makes the PLM act as the PLMVP. Thus, the PLM in the presented PRNG scheme can generate the more uniform sequence.

6.2 Statistical testing

In order to gain the confidence that the proposed PRNG is cryptographically secure, the statistical testing should be performed to verify the randomness of the sequences generated by our scheme. As we know, NIST suite of statistical test is one of the most popular options available for analyzing the randomness of PRNGs. In the following subsection, we briefly introduce the tests of NIST suite and give our test strategy [31].

6.2.1 The NIST tests suite

The US NIST statistical test suite provides 15 statistical tests to detect deviations of a binary sequence from randomness. All the 15 statistical tests are briefly introduced as follows. A more detailed description for them could be found in [31].

Frequency test (FT): The purpose of this test is to determine whether the number of ones and zeros in a sequence are approximately the same.

Frequency test within a block (FBT): The purpose of this test is to determine whether the frequency of ones is an M-bit block is approximately $M/2$. The default value for M is set to 128.

Cumulative sum test (CST): The purpose of the test is to determine whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of that cumulative sum for random sequences.

Runs test (RT): The runs test is used to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence, where a run is an uninterrupted sequence of identical bits.

Test for the longest run of ones in a block (LROBT): The purpose of this test is to determine whether the length of the longest run of ones within the tested sequence is as expected for a random sequence.

Binary matrix rank test (BMRT): The purpose of this test is to check for linear dependence among fixed length substrings of the original sequence.

Discrete Fourier transform test (DFTT): The purpose of this test is to detect periodic features in the tested sequence that would indicate a deviation from the assumption of randomness.

Non-overlapping template matching test (NTMT): The purpose of this test is to reject sequences that exhibit too many occurrences of a given non-periodic pattern.

Overlapping template matching test (OTMT): The purpose of this test is to reject sequences that show deviations from the expected number of runs of ones of a given length.

Maurer's universal statistical test (MUST): The purpose of the test is to detect whether or not the sequence can be significantly compressed without loss of information.

Approximate entropy test (AET): The purpose of the test is to compare the frequency of overlapping blocks of two consecutive/adjacent lengths (l and $l+1$) against the expected result for a random sequence. The default length of each block is set to 10 bits.

Random excursions test (RET): The purpose of this test is to determine if the number of visits to a state within a random walk exceeds what one would expect for a random sequence.

Random excursions variant test (REVT): The purpose of this test is to detect deviations from the expected number of occurrences of various states in the random walk.

Serial test (ST): The purpose of this test is to determine whether the number of occurrences of the 2-mm bit overlapping patterns is approximately the same as would be expected for a random sequence. The default length of each block is set to 16 bits.

Linear complexity test (LCT): The purpose of this test is to determine whether or not the sequence is complex enough to be considered random, where the linear complexity is determined by the Berlekamp–Massey algorithm.

6.2.2 Testing results analysis

The NIST test suite is a statistical package for testing the randomness of bit sequences, which include 15 tests in edition 2.2. The sequence is random if it passes the 15 tests index, whereas the sequence is insufficient. For each test, a *P_value* is computed from the binary sequence. If the *P_value* is greater than a predefined threshold α, then the sequence would be considered to be random with a confidence of $1-\alpha$ and the sequence passes the test successfully. Otherwise, the sequence fails this test.

In our experiment, α is set to 0.01, which means a sequence passed the test is considered as random with 99% confidence. 2000 different sequences, each of 1,000,000 bits, are generated by the proposed PRNG. Then, we calculate the P values corresponding to each sequence for all the 15 tests of NIST suite. The test results (i.e., the *P_values*) are listed in Table 3. The proportion of the sequences passing this particular statistical test is also calculated and compared with the range of acceptable proportion. According to Ref. [8,29], the range of acceptable proportion is [0.9833245, 0.9966745]. It can be seen from Table 3 that over 98.60% sequences pass all randomness test and the average value is 99.08%. It is clear that proportions for each test lies inside the confidence interval. Thus, the sequence generated by the proposed scheme has good random properties with respect to all the 15 tests of NIST suite.

Moreover, for each test, the distribution of P values for the 2000 pseudorandom number sequences has been examined by the method in Ref. [8]. The interval (0, 1) is divided into 10 equal subintervals and the count of P values appearing in each subinterval is displayed in Fig. 17. It is clear that the distribution of P values is uniform.

6.3 Correlation evaluation

A correlation coefficient is generally used to measure the dependence and meaning statistical relationships between two or more random variables or observed data values. For two data sequences, the correlation coefficient is defined by

$$\mathrm{cov}(x, y) = E\{(x - E(x))(y - E(y))\} \tag{6}$$

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{7}$$

where x and y are the numbers in two data sequences and $E(x) = \frac{1}{M}\sum_{i=1}^{M} x_i$, $D(x) = \frac{1}{M}\sum_{i=1}^{M}(x_i - E(x))^2$.

To test the correlation performance of our scheme, we generate some pseudorandom sequences with a size of 2000 numbers according to the following conditions:

(1) Fix μ, m, R and randomly set 1000 initial values of x_0, then produce 1000 sequences.
(2) Fix x_0, m, R and randomly set 1000 values of μ, then produce 1000 sequences;
(3) Fix x_0, μ, R and randomly set 1000 values of m, then produce 1000 sequences;
(4) Fix x_0, μ, m and set R from 0 to 255, then produce 256 sequences.

Finally, the correlation coefficients between each pair of the 3256 produced sequences are computed and the distributions of correlation coefficients are shown in Fig. 18. The line graphs show that all the correlation coefficients are very small and close to zero. It means that the present PRNG has good correlation performance.

6.4 Security analysis

6.4.1 Key space

A good PRNG should be sensitive to the key (or seed) and the key space should be sufficiently large to make brute-force attack infeasible. In the proposed scheme, the key consists of the initial status value x_0, the parameter μ, m and R, where we set $x_0 \in (0, 1), \mu \in (0.01, 3.99), m \in (0.01, 0.1)$ and $R \in [0, 255]$. Suppose the precision of a floating-point number is 10^{-13}. Then, x can be any one among those 10^{13} possible values. Similarly, μ and m can be any values in the range of $(3.99-0.01) \times 10^{13}$ values and $(0.1-0.01) \times 10^{13}$ values, respectively. Meanwhile, R is arbitrarily chosen from 255 possible values. Therefore, the key space is about 9.17×10^{40}, which satisfies the general requirement of resisting brute-force attack. According to the IEEE floating-point standard [32], the computational precision of the 64-bit double-precision numbers is about 10^{-15}. Therefore, a sufficiently large key space is guaranteed in the proposed PRNG for practical applications.

6.4.2 Key sensitivity

Key sensitivity refers that only a tiny difference in the key causes a substantial changes in the output. The following key sensitivity tests have been performed:

Case 1 x_0 is changed from 0.123456789 to $x_0 + \Delta$;
Case 2 μ is changed from 0.987654321 to $\mu + \Delta$;
Case 3 m is changed from 0.012345678 to $m + \Delta$;
Case 4 R is changed from 4 to 5

Table 3 The NIST tests results

No.	Statistical test	Count of sequences with P value $\geq$0.01 (success)	Count of sequences with P value $<$0.01 (failure)	P value corresponding to the goodness of fit (P value)	Proportion of sequences passing the test
1	FT	1979	21	0.518106	0.9895
2	FBT	1973	27	0.203894	0.9865
3	CST (forward)	1979	21	0.473064	0.9895
	CST (reverse)	1978	22	0.573875	0.9890
4	RT	1982	18	0.967382	0.9910
5	LROBT	1982	18	0.492436	0.9910
6	BMRT	1985	15	0.070737	0.9925
7	DFTT	1972	28	0.093157	0.9860
8	NTMT*	1980.277	19.723	0.473206	0.9901
9	OTMT	1981	19	0.245490	0.9905
10	MUST	1980	20	0.307077	0.9900
11	AET	1977	23	0.146152	0.9885
12	*RET (the sample size = 1253)*				
	(1) $x = -4$	1239	14	0.783019	0.9888
	(2) $x = -3$	1241	12	0.063137	0.9904
	(3) $x = -2$	1245	8	0.605501	0.9936
	(4) $x = -1$	1241	12	0.037566	0.9904
	(5) $x = 1$	1243	10	0.808515	0.9920
	(6) $x = 2$	1237	16	0.247255	0.9872
	(7) $x = 3$	1237	16	0.937919	0.9872
	(8) $x = 4$	1244	9	0.216485	0.9928
13	*REVT (the sample size = 1253)*				
	(1) $x = -9$	1247	6	0.459717	0.9952
	(2) $x = -8$	1247	6	0.675372	0.9952
	(3) $x = -7$	1245	8	0.841226	0.9936
	(4) $x = -6$	1242	11	0.574081	0.9912
	(5) $x = -5$	1245	8	0.348546	0.9936
	(6) $x = -4$	1244	9	0.849373	0.9928
	(7) $x = -3$	1238	15	0.817260	0.9880
	(8) $x = -2$	1241	12	0.775337	0.9904
	(9) $x = -1$	1238	15	0.880145	0.9880
	(10) $x = 1$	1242	11	0.022361	0.9912
	(11) $x = 2$	1244	9	0.154109	0.9928
	(12) $x = 3$	1242	11	0.686955	0.9912
	(13) $x = 4$	1242	11	0.820143	0.9912
	(14) $x = 5$	1242	11	0.887349	0.9912
	(15) $x = 6$	1241	12	0.658765	0.9904
	(16) $x = 7$	1240	13	0.802608	0.9896
	(17) $x = 8$	1245	8	0.191273	0.9936
	(18) $x = 9$	1246	7	0.721371	0.9944
14	ST 1	1984	16	0.522100	0.9920
	ST 2	1983	17	0.487561	0.9915
15	LCT	1979	21	0.575932	0.9895

The result marked with an asterisk is the average value

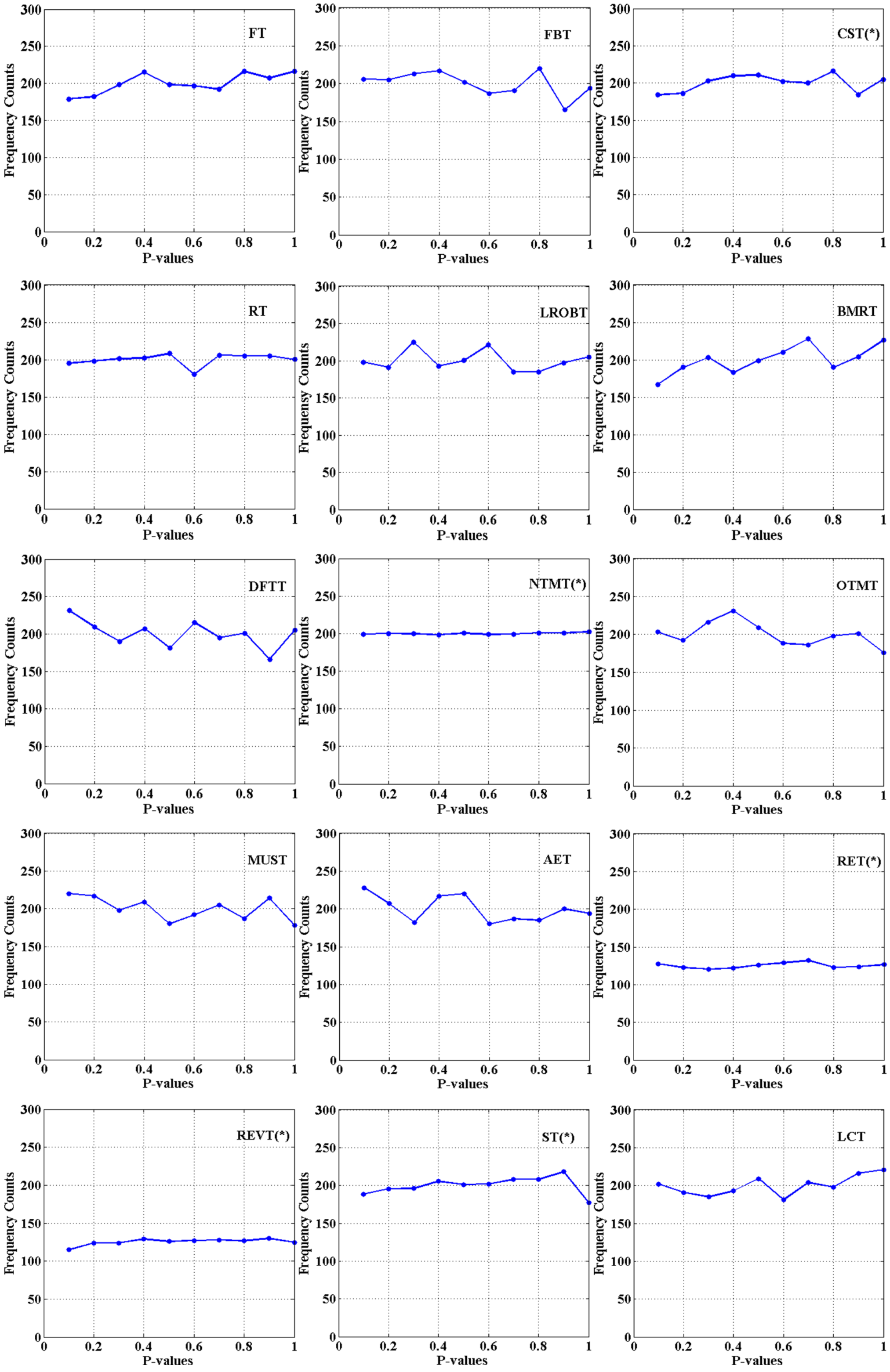


Fig. 17 Polygon of P values for the tests of NITST suite (*Note* the result marked with an *asterisk* is the average value)

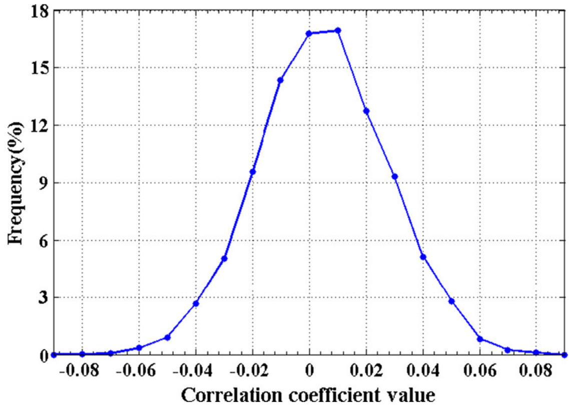


Fig. 18 Distribution of correlation coefficients on interval $[-0.08, 0.08]$

where $\Delta = 2^{-44}$. Under these four different cases, the sequences with 2000 pseudorandom numbers are generated and compared with those of the original cases. The first 20 numbers of each case are plotted in Fig. 19. It is clear from Fig. 19 that the sequences are completely different.

Moreover, the correlation coefficients between the origin sequences and their corresponding sequences with a tiny difference in key are calculated according to Eqs. (6) and (7). The results are shown in Table 4. We can see that all the correlation coefficients are very small, which implies that no detectable correlation exists between the original sequence and the four sequences. Therefore, the proposed scheme possesses high key sensitivity.

6.4.3 Differential attack

The differential attack is generally applied to cryptanalyze block ciphers, which studies the effect of tiny changes in the plaintexts on their corresponding ciphertexts. However, there is no input plaintext for PRNG. We employ the method in Ref. [16] to test the resistance of differential attack. The following analysis is performed on the initial seeds which are the keys for the pseudorandom number generator.

Table 4 Correlation coefficients between the original sequence and the four sequences

	Case 1	Case 2	Case 3	Case 4
Correlation coefficient	0.026	0.025	0.023	0.025

(1) Four sequences (S_1, S_2, S_3 and S_4) with 50,000,000 pseudorandom numbers are generated.

(2) The seeds of presented PRNG are changed according to the operations in Sect. 6.4.2. Then, another four sequences (S_1', S_2', S_3' and S_4') with same size are obtained.

(3) The average absolute difference between sequence pairs (S_1, S_1'), (S_2, S_2'), (S_3, S_3') and (S_4, S_4') are calculated and shown in Table 5. The average absolute difference between two sequences is defined as [16,33]

$$d = \frac{1}{M}\sum_{i=1}^{M} |t(e_i) - t(e_i')| \quad (8)$$

where e_i and e_i' are the ith ASCII character of the original and the new pseudorandom sequences, respectively. Function $t(x)$ converts the entries to their equivalent decimal values. If e_i and e_i' are two independent variables and the corresponding ASCII values have the uniform distribution, it is can be proved that the ideal value of aver-

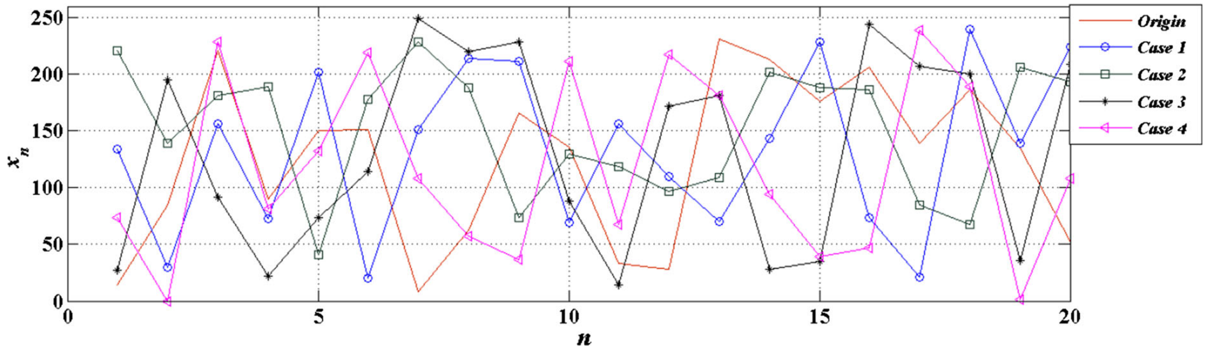


Fig. 19 The sequences of under different conditions

Table 5 The average absolute difference (d) in different cases

	Case 1 (S_1, S'_1)	Case 2 (S_2, S'_2)	Case 3 (S_3, S'_3)	Case 4 (S_4, S'_4)
d	85.3363	85.3191	85.3314	85.3306

age absolute difference is two-third of their mean value, which in this case is 85.333.

We can see from Table 5 that all the average absolute differences in the four cases are very close to the ideal value 85.333, which means that the proposed PRNG has high resistance against differential attack.

6.5 Efficiency analysis

In addition to security, the running speed becomes an important factor for practical applications. In the proposed PRNG, PLM is the core component. When iterating the PLM, we firstly need to determine the subinterval which the state value is in, then, calculate the next state value of the PLM. Since determining the subinterval of the state value is not a time-consuming operation, the computation of iterating the PLM is similar to that of iterating logistic map. As we know, the logistic map is usually used in chaos-based cipher for its high efficiency and good cryptographic properties. Thus, like the logistic map, the PLM also possesses the merit of high efficiency.

In the proposed scheme, the 8-bit number is generated by each iteration of the PLM. There are five addition/subtraction operations and seven multiplication/division operations in each iteration of the PLM. Besides the iteration of PLM, substitution and XOR are the other two main operations in the process of generating 8-bit pseudorandom number. The basic operations of generating 8-bit pseudorandom number are listed in Table 6. It is clear that the proposed scheme needs only 18 simple computation operations to output each 8-bit number, which guarantees its high efficiency.

6.6 Comparison analysis

Recently, some pseudorandom number generators based on chaotic maps have been presented [8,16–18,24]. Owing to make full use of cryptographic property of chaotic system, all of these algorithms have ideal statistical performance and satisfies the requirement of resisting statistical attacks. Due to some homophyly between the proposed algorithm and them, the efficiency of our algorithm is compared with these of algorithms in Refs. [8,24]. In Ref. [8], the PRNG is based on logistic map. After each iteration, only one bit is generated. To get an 8-bit pseudorandom number, the logistic map must be iterated for eight times. The numbers of performing each basic operation to obtain an 8-bit number is counted and also listed in Table 6. Three algorithms of generating pseudorandom numbers are presented based on spatiotemporal chaos in Ref. [24]. In the spatiotemporal chaos, the local chaotic maps are first iterated. Then, the new state values are calculated according to the cou-

Table 6 The numbers of basic operations to generate an 8-bit number

	Our scheme	Ref. [8]	Ref. [24] ($L = 64$)		
			Method 1	Method 2 ($P = 40$)	Method 3 ($P = 8$)
Number of addition/subtraction	5	16	24	0.6	24
Number of multiplication/division	7	32	32	8.8	96
Number of module	0	0	0	8	64
Number of substitution	1	0	0	0	0
Number of exclusive OR (XOR)	1	0	0	0	56
Number of compare	2	8	8	0	0
Converting floating-point to char	2	0	0	8	64
Total	18	56	64	25.4	304

Table 7 The running speed of generating pseudorandom numbers

	Running speed (MByte/s)
Our scheme	20.5383
Ref. [8]	9.5980
Ref. [24] ($L = 64$)	
Scheme 1	8.5470
Scheme 2 ($P = 40$)	12.7408
Scheme 3 ($P = 8$)	1.3675

pling relationship between the local chaotic maps. Although the spatiotemporal chaos has more complex dynamic behavior, iterating the spatiotemporal chaos requires much more computational effort than iterating a simple chaotic map. Similarly, the numbers of each basic operation used to generating an 8-bit number is calculated and shown in Table 6. It is clear that the total number of basic operation in our scheme is smaller than those of the algorithms in Refs. [8,24], which means that the proposed scheme has higher efficiency.

To further verify the efficiency results, our scheme and other algorithms in Refs. [8,24] are implemented by using Visual C++ 6.0 and run on a computer with 330GH Intel Core i3-2120 CPU and 4G RAM. The running speeds of these algorithms are listed in Table 7. The results confirm that the proposed scheme not only owns higher efficiency but also satisfies the requirement of practical application.

7 Conclusion

In this paper, the piecewise logistic map (PLM) is proposed, which enhances some properties related to cryptography. The PLM has better ergodicity property and greater Lyapunov exponent than the logistic map, and extends the value range of μ corresponding to chaotic status. In order to achieve more uniform density probability, we suggested that the variable control parameter should be employed when iterating the PLM. A PRNG is proposed based on PLM with $N = 64$. The operations, such as substitution and XOR, are used in the process of generating numbers, which not only lower the correlation of two adjacent chaotic state values but also enhance the security of pseudorandom number sequence. The generated sequences based on our scheme are rigorously tested by the NIST suite. The tests results show that the proposed PRNG has perfect statistics performance. Moreover, the security and efficiency analyses have confirmed that the proposed algorithm satisfy all the performance requirements of a PRNG. It is practical and reliable, with high potential to be adopted in the design of stream cipher.

Acknowledgments The work described in this paper was supported by grants from the National Natural Science Foundation of China (No. 61472464), the National Social Science Foundation of China (No. 14CTQ026), the Chongqing Research Program of Application Foundation and Advanced Technology (Nos. cstc2013jcyjA40017, cstc2014jcyjA40028), the Foundation of Chongqing Education Committee (No. KJ120506).

References

1. Pareek, N.K., Patidar, V., Sud, K.K.: Image encryption using chaotic logistic map. Image Vis. Comput. **24**, 926–934 (2006)
2. Mandal, M.K., Banik, G.D., Chattopadhyay, D.: An image encryption process based on chaotic logistic map. IETE Tech. Rev. **441**, 441–452 (2009)
3. Wang, Y., Wong, K.W., Liao, X., Chen, G.: A new chaos-based fast image encryption algorithm. Appl. Soft Comput. J. **11**(1), 514–522 (2011)
4. Sethi, N., Sharma, D.: A novel method of image encryption using logistic mapping. Int. J. Comput. Sci. Eng. **1**, 115–119 (2012)
5. Cheng, H., Huang, C., Ding, Q., Chu, S.C.: An efficient image encryption scheme based on ZUC stream cipher and chaotic logistic map. Adv. Intell. Syst. Comput. **298**, 301–310 (2014)
6. Sam, I.S., Devaraj, P., Bhuvaneswaran, R.S.: Transformed logistic block cipher scheme for image encryption. Commun. Comput. Inf. Sci. **132**, 70–78 (2011)
7. LjupcoKocarev, GoceJakimoski: Logistic map as a block encryption algorithm. Phys. Lett. A **289**, 199–206 (2001)
8. Pareek, N.K., Patidar, V., Sud, K.K.: A pseudo random bit generator based on chaotic logistic map and its statistical testing. Informatica **33**, 441–552 (2009)
9. Pellicer-Lostao, C., López-Ruiz, R.: Pseudo-random bit generation based on 2D chaotic maps of logistic type and its applications in chaotic cryptography. Comput. Sci. Appl. **5073**, 784–796 (2008)
10. Wang, L.: Research of one way hash function based on logistic mapping. Comput. Eng. Des. **27**, 774–776 (2006)
11. Rani, P.J., Rao, M.S., Bhavani, S.D.: Design of secure chaotic hash function based on logistic and tent maps. Commun. Comput. Inf. Sci. **196**, 43–52 (2011)
12. Li, C.Q., Xie, T., Liu, Q., Cheng, G.: Cryptanalyzing image encryption using chaotic logistic map. Nonlinear Dyn. **78**(2), 1545–1551 (2014)
13. Li, C.Q., Li, S.J., Asim, M., Juana, N., Gonzalo, A., Chen, G.R.: On the security defects of an image encryption scheme. Image Vis. Comput. **27**(9), 1371–1381 (2009)

14. Francois, M., Grosges, T., Barchiesi, D.: Pseudo-random number generator based on mixing of three chaotic maps. Commun. Nonlinear Sci. Numer. Simul. **4**, 887–895 (2014)
15. Nian-Sheng, L.: Pseudo-randomness and complexity of binary sequences generated by the chaotic system. Commun. Nonlinear Sci. Numer. Simul. **16**, 761–768 (2011)
16. Akhshani, A., Akhavan, A., Mobaraki, A., Lim, S.C., Hassan, Z.: Pseudo random number generator based on quantum chaotic map. Commun. Nonlinear Sci. Numer. Simul. **19**, 101–111 (2014)
17. Chiang, Y.T., Wang, H.S., Wang, Y.N.: A chaotic-based pseudo-random bit generator for navigation applications. Appl. Mech. Mater. **311**, 99–104 (2013)
18. Wang, X.Y., Qin, X.: A new pseudo-random number generator based on CML and chaotic iteration. Nonlinear Dyn. **2**, 1589–1592 (2012)
19. Li, S.J., Mou, X.Q., Cai, Y.L.: Pseudo-random bit generator based on couple chaotic systems and its application in stream-ciphers cryptography. In: Progress in Cryptology—INDOCRYPT 2001. Lecture Notes in Computer Science, vol. 2247, pp. 316–329 (2001)
20. Maguire, L.P., Roche, B., McGinnity, T.M., McDaid, L.J.: Predicting a chaotic time series using a fuzzy neural network. Inf. Sci. **112**, 125–136 (1998)
21. Short, K.M.: Unmasking a modulated chaotic communications scheme. Int. J. Bifurc. Chaos **6**(2), 367–375 (1996)
22. Short, K.M.: Steps toward unmasking secure communications. Int. J. Bifurc. Chaos **4**(4), 959–977 (1994)
23. Hu, H., Liu, L., Ding, N.: Pseudorandom sequence generator based on the Chen chaotic system. Comput. Phys. Commun. **184**, 765–768 (2013)
24. Li, P., Li, Z., Halang, W.A., Chen, G.: A multiple pseudorandom-bit generator based on a spatiotemporal. Phys. Lett. A **349**, 467–573 (2006)
25. Jiulun, F., Xuefeng, Z.: Piecewise logistic chaotic map and its performance analysis. Acta Electron. Sin. **37**(4), 720–725 (2009)
26. Francois, M., Grosges, T., Barchiesi, D., Erra, R.: A new pseudo-random number generator based on two chaotic maps. Informatica **24**(2), 181–197 (2013)
27. Aurell, E., Boffetta, G., Crisanti, A., Paladin, G., Vulpiani, A.: Predictability in the large: an extension of the concept of Lyapunov exponent. J. Phys. A Math. Gen. **30**, 1–26 (1997)
28. Wolf, A., Swift, J.B., Swinney, H.L., Vastano, J.A.: Determining Lyapunov exponents from a time series. Phys. D Nonlinear Phenom. **16**, 285–317 (1985)
29. Hao, B.L.: Starting with Parabolas—An Introduction to Chaotic Dynamics. Shanghai Scientific and Technological Education Publishing House, Shanghai (1993). (in Chinese)
30. Announcing the advanced encryption standard (AES). Federal Information Processing Standards Publication 197. 2001.2
31. A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications. NIST Special Publication 800-22 rev1a. 2010
32. IEEE Standard for Floating-Point Arithmetic. IEEE Computer Society. 2008.8
33. Akhavan, A., Samsudin, A., Akhshani, A.: Hash function based on piecewise nonlinear chaotic map. Chaos Solitons Fractals **42**, 1046–1053 (2009)