

Article

A Pseudorandom Number Generator Based on the Chaotic Map and Quantum Random Walks

Wenbo Zhao ¹, Zhenhai Chang ², Caochuan Ma ^{2,*} and Zhuozhuo Shen ²

¹ School of Electronic Information and Electrical Engineering, Tianshui Normal University, Tianshui 741000, China

² School of Mathematics and Statistics, Tianshui Normal University, Tianshui 741000, China

* Correspondence: ccma@tsnu.edu.cn

Abstract: In this paper, a surjective mapping that satisfies the Li–Yorke chaos in the unit area is constructed and a perturbation algorithm (disturbing its parameters and inputs through another high-dimensional chaos) is proposed to enhance the randomness of the constructed chaotic system and expand its key space. An algorithm for the composition of two systems (combining sequence based on quantum random walks with chaotic system’s outputs) is designed to improve the distribution of the system outputs and a compound chaotic system is ultimately obtained. The new compound chaotic system is evaluated using some test methods such as time series complexity, autocorrelation and distribution of output frequency. The test results showed that the new system has complex dynamic behavior such as high randomness, unpredictability and uniform output distribution. Then, a new scheme for generating pseudorandom numbers is presented utilizing the composite chaotic system. The proposed pseudorandom number generator (PRNG) is evaluated using a series test suites such as NIST sp 800-22 soft and other tools or methods. The results of tests are promising, as the proposed PRNG passed all these tests. Thus, the proposed PRNG can be used in the information security field.

Keywords: Li–Yorke chaos; perturbation algorithm; composition of two systems; PRNG



Citation: Zhao, W.; Chang, Z.; Ma, C.; Shen, Z. A Pseudorandom Number Generator Based on the Chaotic Map and Quantum Random Walks. *Entropy* **2023**, *25*, 166. <https://doi.org/10.3390/e25010166>

Academic Editors: Yongpan Sheng, Hao Wang and Yixiang Fang

Received: 10 December 2022

Revised: 6 January 2023

Accepted: 10 January 2023

Published: 13 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Chaos theory is a conspicuous area in the researches of mathematics and dynamic system and has attracted many researchers for nearly fifty years [1]. A chaotic dynamic system has the special nonlinear dynamics characteristics that can be regarded as a random motion, and its motion trail is characterized by sensitivity of the initial value and the initial parameter, unpredictability and ergodicity. Therefore, chaos theory is comprehensively applied in engineering fields of the communication, signal processing, etc. [1–4]. Especially in the information security field, many designs of safety algorithms based on the chaotic map are proposed, such as the block cipher S-box, the key generator in stream cipher, and the construction of Hash compression function, etc.

One of the most significant components of an information security system is the random number generator. Random number generator (RNG) are widely applied in many fields such as Artificial intelligence, Digital communications, System testing, Statistical simulation, Software development and Crypto-system [5–8]. In different application fields, RNG has diverse properties and these properties include: a sequence generated by a RNG has any weakness in statistics; attackers can not predict the leading sequence or the following sequence; a sequence can be generated or predicted as the internal state value is known. In view of the forgoing premises, random number generators are divided into two categories: true random number generator (TRNG) and pseudorandom number generator (PRNG). True random number generators are usually based on the phenomena of the true world and the physical process. However, TRNG has some disadvantages, such as slow speed, high cost and over dependence on hardware. Accordingly, most practical

application systems choose the pseudorandom number generator, especially in network information security system (cryptographical system). A Cryptographical system requires that a PRNG with little wasted memory can generate a sequence of long period and generate unpredictable data quickly. A complex cryptosystem possesses two main operations: Diffusion and Confusion. Chaotic system has many characteristics: ergodicity, sensitivity to initial conditions and structural complexity of dynamic system. These properties are equivalent to the confusion, diffusion and algorithm complexity in traditional cryptosystem. Therefore, many Chaotic-maps-based PRNGs have been put forward. Chaotic maps such as logistic mapping and its variant, quantum logistic map, one dimension piecewise linear map and tinkerbell map have been widely used in PRNGs [1,9–12].

Most application depend on the performance of the original chaotic system, that is to say, chaos in ideal state. However, in the practical system operation, original chaos system may lead to arise problems such as short cycle, nonergodicity and decreased complexity, which will make application systems lose their original characteristics like long-term unpredictability, etc.; thus, a cryptosystem based on the original chaotic map may be successfully attacked [4,13,14]. Security of the analyzed PRNG is much lower than expected and it should be used with caution [14]. Even some security problems can allow attackers to completely crack and analyze the cryptosystems, getting the secret data and secret keys. It is critically necessary to improve chaos power performance degeneration and further optimize the chaos. Common methods of improving chaos power performance degeneration include [4]: high precision, approaches of the connection of multiple chaos systems, and methods of the disturbance, etc. Among those, the method of the disturbance can improve the performance (prolong the cycle and enhance the complexity) of the chaos greatly if constructed rationally. We usually hope to get the chaos map with uniform output; it is necessary to further optimize the output distribution. A brilliant simple solution to optimize the output distribution can be chosen, which is to combine chaotic outputs with another pseudorandom signals.

On the basis of the fact that ring graph quantum random walks (QRWs) are prone to generate the pseudorandom sequence with uniform distribution, the system output distribution can be improved by mixing original system outputs and QRWs outputs together. QRWs is a quantum corresponding scene of the classical random walk. For the widespread applications of the classical random walk in fields of physics, biology, computer science and finance, etc. [15]. Hence in the future, QRWs probably become tools for many applications, and it may appear lots of information security algorithms based on the QRWs [16–19]. In literature [18], Y. Yang and Q. Zhao constructed a novel PRNG based on QRWs. The present QRWs-based PRNG has some advantages such as better statistical complexity and recurrence, whose normalized Shannon entropy are close to 1. Thus, it is indicated that outputs of PRNG based on QRWs distribute uniformly. Therefore, it is a good method by simulating QRWs to construct a “stochastic” system.

In conclusion, if a PRNG based on the chaotic map is to be designed, the original chaotic system is should not be used directly. A perturbation algorithm (makeing use of a high dimensional chaotic system to disturb the inputs and parameters of the original system) should be applied to enhance randomness of chaotic system and expand its key space. The devise of combining the chaotic system with a sequence based on QRWs will be further improved output distribution. A predicted outcome is that a compound chaotic system with large key space, high randomness and high uniform output distribution can be obtained.

Inspired by reasons discussed above, we are motivated to search for a novel compound chaotic system with complex dynamic behavior and design a PRNG based on compound chaotic system to meet the needs of practical applications.

The rest of this paper is arranged as following: In Section 2, we constructed a surjective chaotic map that satisfies Li–Yorke chaos condition in unit region; in Section 3, we used a discrete two-dimensional chaotic system to disturb the parameters and inputs of the constructed system and combined its outputs with the sequence generated by the quantum

random walk, thus obtaining a compound chaotic system with complex behavior and nearly uniform distribution; in Section 4, a new scheme for generating pseudorandom numbers is presented utilizing the composite chaotic system, and the security and randomness of the proposed PRNG are analyzed and tested roundly; in Section 5, the research results are summarized.

2. A Internal Randomness System is Constructed in Unit Region

The parameter equations of conic curve in unit region are given:

$$\begin{cases} x(t) = \frac{2\omega_1 t(1-t)x_1 + \omega_2 t^2}{\omega_0(1-t)^2 + 2\omega_1 t(1-t) + \omega_2 t^2}, \\ y(t) = 2 \frac{\omega_1 t(1-t)y_1}{\omega_0(1-t)^2 + 2\omega_1 t(1-t) + \omega_2 t^2}. \end{cases} \tag{1}$$

When $t \in [0, 1]$, the two ends of the curve are $(0, 0)$ and $(1, 0)$. The shape of the curve is determined by ω_0, ω_1 and ω_2 , and the bump and height of the curve are determined by (x_1, y_1) . The curve is shown in Figure 1(1).

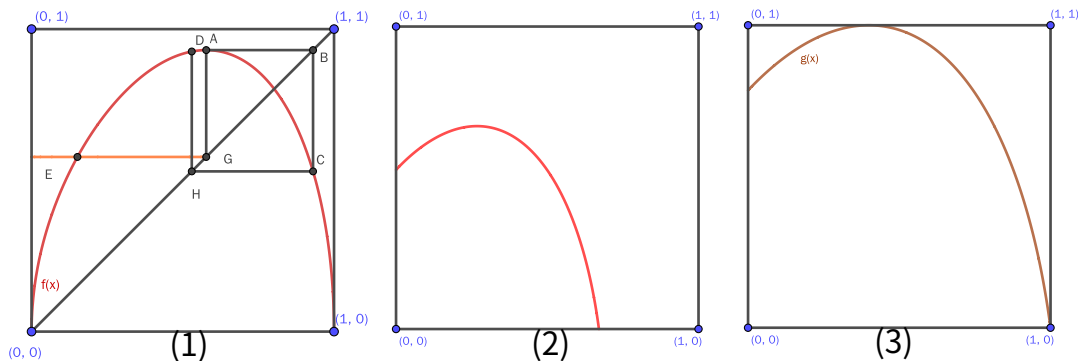


Figure 1. Conical section in unit region.

As shown in Figure 1(1): point A is the maximum value of the curve, the ordinate of point A is the same as the abscissa of point C, the abscissa of point A is the same as the abscissa of point E, and the abscissa of point C is the same as the abscissa of point D. Point A is mapped to point D through two times of recursion. According to the conclusion in literature [20], map (1) satisfied the general conditions for Li–Yorke chaos as long as the ordinate of point D is greater than the ordinate of point E. The system satisfying conditions for Li–Yorke chaos should be constructed from the explicit form of the curve because explicit form is more understandable than implicit expression. There are three types of conic curves, namely, parabola, ellipse and hyperbola. In this paper, we choose an ellipse curve for researching and focus on constructing a chaotic system.

Let f be an elliptic curve. Because the two endpoints of the curve are $(0, 0)$ and $(1, 0)$ respectively, the curve equation can be given:

$$f(x) = a\sqrt{-x^3 + x}, \tag{2}$$

calculate the maximum value of curve $D(x_{max}, y_{max})$, we have

$$\begin{cases} x_{max} = \frac{\sqrt{3}}{3}, \\ y_{max} = f(x_{max}). \end{cases}$$

If Equation (2) satisfied the following conditions (3):

$$\begin{cases} 1 \geq y_{max} > 0, \\ f^2(y_{max}) > x_{max}, \end{cases} \tag{3}$$

system (2) is Li-Yorke chaos as $(1.4690 < a \leq 1.6110)$ by calculation.

Chaotic map satisfying conditions (3) is transformed to surjective map by isometric scaling, and it does not change chaotic characteristics. So, curve (2) is first shifted to the left and down $m = f(y_{max})$, as shown in Figure 1(2). Then, the map is magnified by $t = 1/(y_{max} - f(y_{max}))$ times and obtains a surjective chaotic map in unit region. The surjective map is shown in Figure 1(3), and the expression is as the following:

$$g(x) = t \left(a \sqrt{\frac{1}{t}x + m} - \left(\frac{1}{t}x + m \right)^3 - m \right). \tag{4}$$

Lyapunov Exponent, Trajectory Iteration Diagram and Bifurcation Diagram

Lyapunov exponent is a main quantitative index of chaotic analysis by reason that it is used to describe the local stability of the trajectory of the dynamic system. In general, as the system is chaotic, the Lyapunov exponent is positive. The calculation of Lyapunov exponent is by using the definition method, and the evaluating expression is as the following:

$$\begin{cases} g^n(x) = g(g^{n-1}(x)), \\ LE = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln \left| \frac{dg^n(x)}{dx} \right|_{x=x_i}. \end{cases} \tag{5}$$

The Lyapunov exponent of system (4) calculated by Formula (5) is shown in Figure 2. Figure 2 shows the Lyapunov exponent of system (4) for different control parameter a . According to Figure 2, the system (4) can exhibit chaotic behavior for $1.52 \leq a \leq 1.6$.

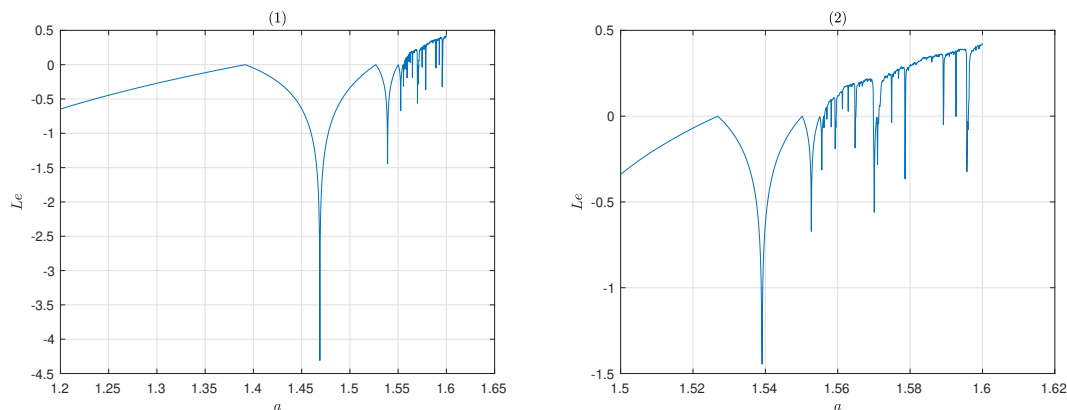


Figure 2. Lyapunov exponent of the map (4).

Bifurcation diagram of map (4) is shown in Figure 3. From Figure 3, the results indicate that as parameter a gradually increases from 1.5 to 1.61, chaos phenomena appears. In a certain range of values of the control parameter, $1.59 \leq a \leq 1.60$, full chaotic behavior can be seen with Figures 2 and 3.

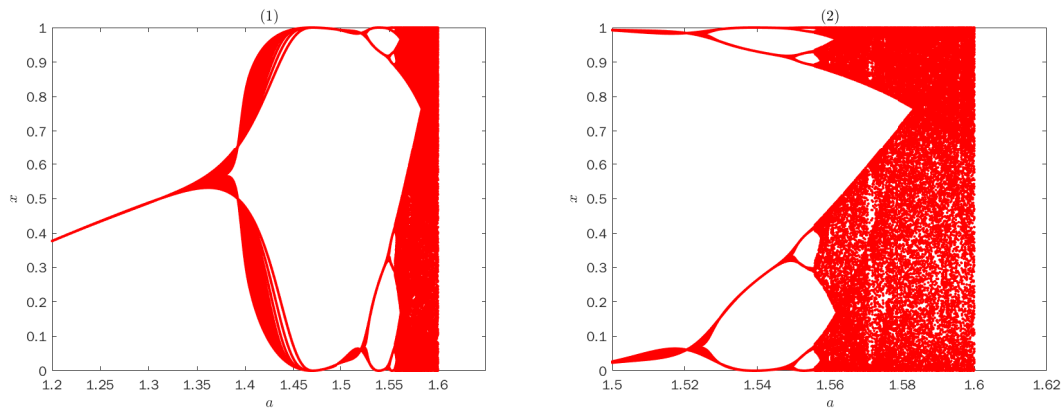


Figure 3. Bifurcation diagram of the map (4).

For a chaotic system, it will be found that the iterative trajectory of the system will present chaotic state as giving an initial value and analyzing its output sequence. The iterative trajectory of system (4) is shown in Figure 4. It can be judged from the Figure 4 that as $a = 1.59$, the system has obvious chaos characteristics; as $a = 1.44$, the system takes on periodic oscillation state; as $a = 1.38$, the system converges to a stable state.

To facilitate the simplification of the established chaotic system, let $\alpha = (a - 1.56) \times 20 + 1$, then Equation (4) can be rewritten as:

$$g(x) = t \left(a(\alpha) \sqrt{\frac{1}{t}x + m - \left(\frac{1}{t}x + m\right)^3} - m \right), \quad a(\alpha) = \frac{\alpha - 1}{20} + 1.56. \quad (6)$$

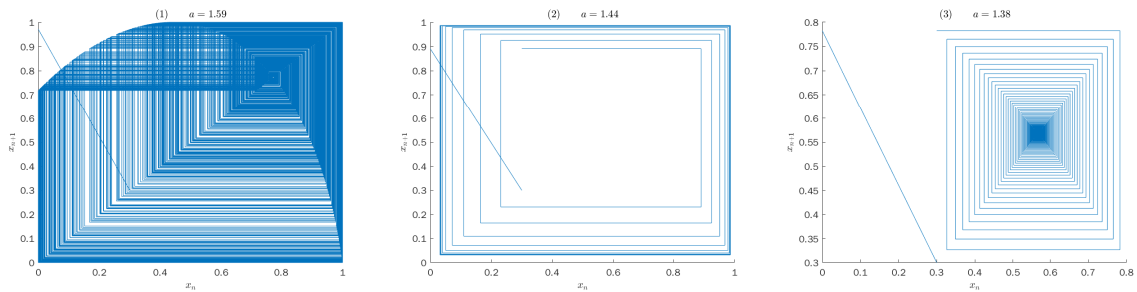


Figure 4. Iterative trajectory of map (4).

3. Design and Performance Analysis of a New Compound Chaotic System

In practical applications, a chaotic system with large key space, complex dynamic behaviors and nearly uniform distribution is generally required. As a chaotic system has been in operation of digital systems with finite precision, the dynamic performance can deteriorate. So, to improve more the dynamic performance of the chaotic system (4) and overcome the short period of chaotic sequence caused by the finite precision effect, a mechanism needs to be designed.

Firstly, we choose perturbation method [4,21], that is, known two-dimensional chaotic map outputs are used to perturb the constructed system parameters and inputs; Then, we research the quantum random walk on the ring graph under control of two-dimensional chaos, and the outputs of the perturbed system are merged into outputs of the quantum random walk. A new compound chaotic system with complex behavior is obtained and the output sequence generated by new system distributes uniformly in the whole state space.

3.1. Optimization Algorithm Based on Two-Dimensional Chaotic Map and Quantum Random Walk

The discretization of (6) is derived as the following:

$$x_{n+1} = t \left(a(x) \sqrt{\frac{1}{t} x_n + m - \left(\frac{1}{t} x_n + m \right)^3} - m \right). \tag{7}$$

Call equation (7) as the *Ecsys*; then, we select a two-dimensional hyperchaotic system to disturb the parameters and inputs of *Ecsys* while controlling the quantum random walk.

3.1.1. Two-Dimensional Hyper-Chaotic System

The general expression of two-dimensional hyperchaotic system is as the following:

$$\begin{aligned} u_{n+1} &= k_{11} + k_{12}u_n + k_{13}u_n^2 + k_{14}v_n + k_{15}v_n^2 + k_{16}u_nv_n, \\ v_{n+1} &= k_{21} + k_{22}u_n + k_{23}u_n^2 + k_{24}v_n + k_{25}v_n^2 + k_{26}u_nv_n. \end{aligned} \tag{8}$$

Limit the coefficients in Equation (8) and make most of them zero, a simplified two-dimensional chaotic system can be obtained ultimately. Let $k_{11} = k_{12} = k_{13} = k_{16} = k_{21} = k_{23} = k_{25} = k_{26} = 0$, that is

$$\begin{aligned} u_{n+1} &= k_{14}v_n + k_{15}v_n^2, \\ v_{n+1} &= k_{22}u_n + k_{24}v_n, \end{aligned} \tag{9}$$

where $k_{15} = -1.55, k_{22} = -1.1$ and $k_{24} = 0.1$. We take k_{14} as the control parameter.

In order to discuss chaotic characteristics of system (9) caused by the variation of parameter k_{14} , a modified version of Marotto’s theorem is first presented in literature [22]. A discrete dynamical system is as the following:

$$X_{n+1} = F(X_n), n \geq 0, \tag{10}$$

where $F : \mathcal{X} \rightarrow \mathcal{X}$ is the mapping, and $(\mathcal{X}, \|\cdot\|)$ is the Banach space.

Theorem 1. Let $z \in R^n$ be a fixed point of the mapping $F : R^n \rightarrow R^n$. Assume that

- a. F is continuously differentiable in some fields of z and the absolute values of all eigenvalues of $DF(z)$ are greater than 1. Thus, there exists a normal number r and a norm of $\|\cdot\|$, so that F can expand on $\bar{B}_r(z)$ under $\|\cdot\|$, $\bar{B}_r(z)$ is a closed sphere of space $(R^n, \|\cdot\|)$ centered on z ;
- b. z is the return-expansion fixed point of F , that is, it exists a point $x_0 \in B_r(z)$ and positive integer m such that $F^m(x_0) = z(x_0 \neq z)$, where $\bar{B}_r(z)$ is the opening ball of space $(R^n, \|\cdot\|)$ centered on z . F is continuous and differentiable in a field of x_0, x_1, \dots, x_{m-1} and $\det DF(x_j) \neq 0 (0 \leq j \leq m - 1)$, where $x_j = F(x_{j-1})$.

Then, the system (10) is chaotic in the sense of Li–Yorke.

The value of k_{14} is discussed below for system (9), as it satisfies theorem (1). A fixed point of system (9) is $O = (0, 0)$, we can define the following norm:

$$\|(u, v)\| = \sqrt{u^2 + v^2}.$$

Let $h = (h_1, h_2)$, where $h_1(u, v) = k_{14}v - 1.55v^2$ and $h_2(u, v) = -1.1u + 0.1v$. It is obvious that h is continuously differentiable in R^2 , and its Jacobian matrix is

$$Dh(u, v) = \begin{pmatrix} 0 & k_{14} - 3.1v \\ -1.1 & 0.1 \end{pmatrix}.$$

For the fixed point $O = (0, 0)$, we assume that the absolute values of all eigenvalues of the matrix $Dh(O)$ are greater than 1, that is, system (9) may be chaotic when

$k_{14} < -1.023532631$ or $k_{14} > 1.023532631$. The Lyapunov exponent of the system is shown in Figure 5(1–2), and the bifurcation diagram of the system (8) is shown in Figure 5(3–6). It can be seen that when parameter k_{14} increases gradually from 0.9 to 1.47, the system gradually enters a complex chaotic state. Without losing generality, let $k_{14} = 1.55$ and analyze whether system (9) with the fixed point $O = (0, 0)$ satisfies the Theorem 1.

It is shown that $\|h(x) - h(y)\| \geq 1.1\|x - y\|$ for all $x, y \in \bar{B}_r(O)$ where $r = 5.556897 \times 10^{-163}$. Therefore, O is an expansion fixed point of h in $\bar{B}_r(O)$. After the calculation, there is a point $x_0 = (6.253681 \times 10^{-163}, -1.485254 \times 10^{-162}) \neq O, x_0 \in \bar{B}_r(O)$ and a positive integer $m = 1290$ to $h^m(x_0) = O$. It can be obtained $\frac{\partial h_1}{\partial u}, \frac{\partial h_1}{\partial v}, \frac{\partial h_2}{\partial u}$ and $\frac{\partial h_2}{\partial v}$ are continuous in $B_r(O)$, because that $\frac{\partial h_1}{\partial u} = 0, \frac{\partial h_1}{\partial v} = 1.5 - 2.6v, \frac{\partial h_2}{\partial u} = -1.1$ and $\frac{\partial h_2}{\partial v} = 0.1$. Then, h is continuously differentiable at x_i and $\det Dh(x_i) \neq 0, 0 \leq i \leq m$, according to theorem 1, the fixed point O is the return-expansion fixed point of h , that is, h is Li-Yorke chaos. It can be inferred that map (9) is chaotic in the sense of Li-Yorke as $k_{14} = 1.55$.

According to the above theoretical analysis and simulation, system (9) has complex dynamic behavior when the parameter $k_{14} \in [1.47, 1.57]$. It can be used as a disturbance source to *Ecsys*.

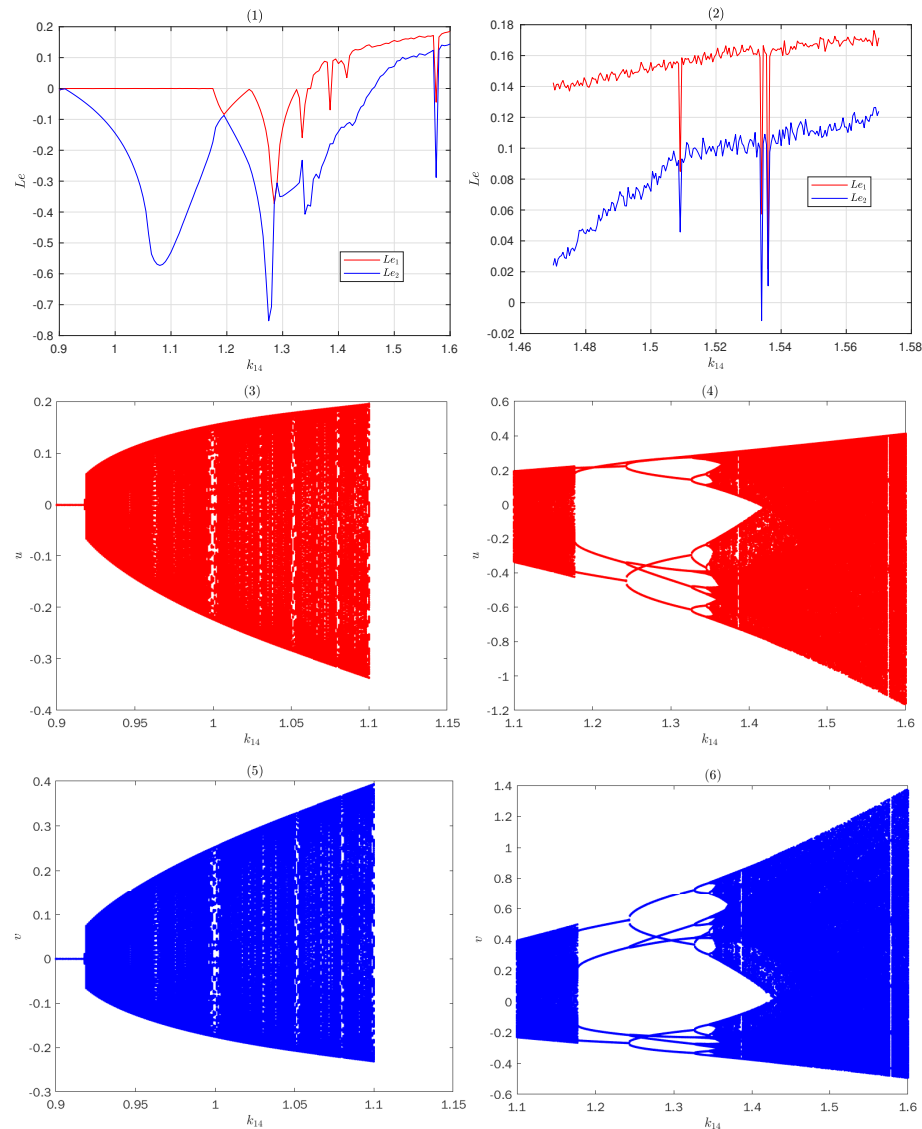


Figure 5. Lyapunov exponent and Bifurcuresation diagram of sys (9) for k_{14} .

3.1.2. Sequence Generation Algorithm Based on Quantum Random Walk

Let G be n -nodes and undirected graph. It is a n -cycle graph, that is, the degree of each node is 2. Then, the quantum random walk in G contains two quantum systems: Walker and Coin. Walker is an N -dimensional Hilbert space H_p , whose location of the ground state is $\{|i\rangle, i \in \{0, 1, 2, \dots, N\}\}$. Any position of Walker can be represented as $\sum_i k_i |i\rangle$, and $\sum_i |k_i|^2 = 1$. Coin is a two-dimensional Hilbert space H_c , whose ground state is $\{|0\rangle, |1\rangle\}$. Then the state of any Coin can be expressed as $a|0\rangle + b|1\rangle$, and $|a|^2 + |b|^2 = 1$. The joint state of Walker and Coin is $H_t = H_p \otimes H_c$, and the evolution of the joint state is accomplished by using coin operation and position movement.

1. The coin operator \hat{C}_θ is as following:

$$\hat{C}_\theta = \cos \theta |0\rangle\langle 0| + \sin \theta |0\rangle\langle 1| + \sin \theta |1\rangle\langle 0| - \cos \theta |1\rangle\langle 1|.$$

Let the position shift operator be $\hat{S}_{(forward,back)}$, and the expression is as following:

$$\hat{S} = \sum_i (|i + forward(mod\ n)\rangle\langle i| \otimes |0\rangle\langle 0| + |i - back(mod\ n)\rangle\langle i| \otimes |1\rangle\langle 1|),$$

where *forward* means that Walker goes right steps as the state of the coin is $|0\rangle$, and *back* means the steps to left as $|1\rangle$. So, each step of the quantum random walk can be written as

$$\hat{U}_{(\theta, forward, back)} = \hat{S} \cdot (\hat{C} \otimes I). \tag{11}$$

Assuming that the initial state of the system is $|\varphi(0)\rangle$, and after t steps, according to Equation (11), the joint state is

$$|\varphi(t)\rangle = \hat{U}^t |\varphi(0)\rangle = (\hat{S} \cdot (\hat{C} \otimes I))^t |\varphi(0)\rangle. \tag{12}$$

Then, the probability of stopping at point $|v\rangle$ in graph G after step t is

$$P_t(v|\varphi(0)) = |\langle(v,0)|\varphi(t)\rangle|^2 + |\langle(v,1)|\varphi(t)\rangle|^2, \tag{13}$$

and the limiting distribution π of stopping at point $|v\rangle$ is

$$\pi = \lim_{T \rightarrow \infty} \bar{P}_T(v|\varphi(0)) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} P_t(v|\varphi(0)). \tag{14}$$

In order to design a sequence generation algorithm, the following theorem based on Theorem 3.6 and Theorem 4.1 in literature [23] is given.

Theorem 2. *Let U be a coined quantum walk on the n -cycle graph, with n odd, and with the Hadamard transform as the coin. Then the limiting distribution π is uniform over the nodes of the graph, independent of the initial state $|\varphi(0)\rangle$.*

According to Theorem 2, if quantum random walk is based on n -cycle graph G , the number of vertices in graph G is n , n is an odd number, and the number of iterations t is relatively large, (14) is close to uniform distribution. A vector $(\theta, forward, back, n, i_0, c_0)$ is setup, where c_0 is the initial state of coin and i_0 is the initial position of the bludger. It can be seen from (13) that there is a nonlinear map between the probability distribution $\mathbf{P}_t = [P_t(v_1|\varphi(0)), P_t(v_2|\varphi(0)), \dots, P_t(v_n|\varphi(0))]$ and initial state $|\varphi(0)\rangle = (i_0, c_0)$. According to (12) and (13), a uniformly distributed sequence can be generated. Hence, a sequence generation Algorithm 1 with high sensitivity to initial conditions is proposed following:

Algorithm 1 Sequence generator algorithm.

Input: $(\theta, faward, back, n, i_0, c_0)$
 Output: *AllOutputSeq*

1. $T_{max} = 2n, AllOutputSeq = \emptyset;$
2. *for* $T = 1 : T_{max}$
 - $t = 1, |\varphi(0)\rangle = (i_0, c_0),$
 - while* $(t \leq T)$
 - $|\varphi(t)\rangle = \hat{U}|\varphi(t-1)\rangle$
 - end;*
 - $|\varphi(T)\rangle = \sum_1^{n_1} k_i(x_i, 0) + \sum_1^{n_2} h_j(y_j, 1),$
 - $S_1 = \{(k_1, x_1, 0), (k_2, x_2, 0) \dots (k_{n_1}, x_{n_1}, 0)\},$
 - $S_2 = \{(h_1, y_1, 1), (h_2, y_2, 1) \dots (h_{n_2}, y_{n_2}, 1)\},$
 - $RanSeq = \{(p_1, x_1), (p_2, x_2) \dots (p_{n_1}, x_{n_1}) | p = k^2\}$
 - fors*₂ $= 1 : length(S_2)$
 - fors*₁ $= 1 : length(S_1)$
 - if* $(x_{s_1} == y_{s_2})$
 - $(p_{s_1}, x_{s_1}) = (p_{s_1} + h_{s_2}^2, x_{s_1}),$
 - break,*
 - endif*
 - endfor*
 - if* $(s_1 == length(S_1))$
 - $RanSeq+ = (h_{s_2}^2, y_{s_2}),$ ultimately
 - endif*
 - endfor,*
 - $OutputSeq = \{\pm p_i + \frac{x_i}{n} | i = 1, 2 \dots length(RanSeq)\},$
 - $AllOutputSeq+ = OutputSeq,$
 - endfor.*
 - 3. *return AllOutputSeq*

3.1.3. Optimization Scheme and a Compound Chaotic System

The optimization block diagram is shown in Figure 6, where u of system (9) is used to disturb the input x of *Ecsys* system, and v is used to disturb the control parameter α . A name “*Qusys*” is given by the sequence based on Algorithm 1. (u, v) and output x of *Ecsys* controlled parameters (θ, c_0, n, i_0) of *Qusys* and other parameters of *Qusys* are fixed. The outputs of *Ecsys* merged with outputs of *Qusys*, and y_j is combined output. So, a new compound chaotic system is ultimately obtained.

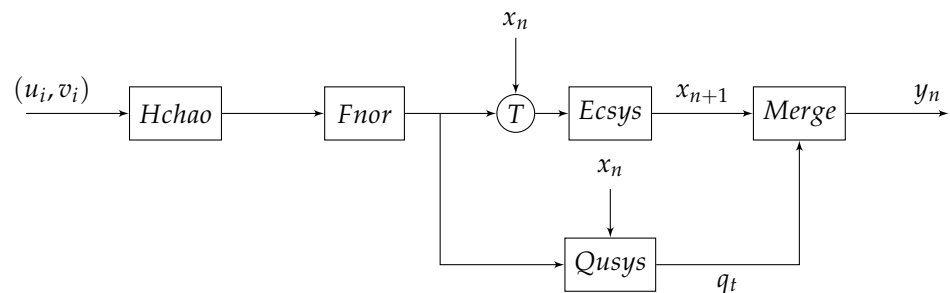


Figure 6. Block diagram of the optimization scheme.

Detailed description of perturbation algorithm is as following. A simple normalized processing function is given by

$$Fnor(x) = (x - x_{min}) / (x_{max} - x_{min}).$$

Apparently, $Fnor(x)$ can adjust the value of the output (u, v) of system (9) to the unit region $(0, 1)$. Two disturbance functions are constructed, and named, respectively, T_1 and T_2 . The functions is designed as the following:

$$\begin{cases} T_1(u_{i+1}, x_n) = Fnor(u_{i+1}) * \beta + x_n * (1 - \beta), \\ T_2(v_{i+1}) = 1.5 + Fnor(v_{i+1}) * \gamma, \end{cases}$$

where $\beta \in (0, 1)$ and $\gamma \in (0, 0.5)$ are two control parameters. After being perturbed, a chaotic system can be obtained:

$$x_{n+1} = t \left(a(T_2(v_{i+1})) \sqrt{\frac{1}{t} T_1(u_{i+1}, x_n) + m - \left(\frac{1}{t} T_1(u_{i+1}, x_n) + m \right)^3 - m} \right), \tag{15}$$

where $a(c(v_{i+1})) = \frac{T_2(v_{i+1})-1}{20} + 1.56$. Parameter settings of *Qusys* are as following:

$$\begin{aligned} \theta &= Fnor(u) \cdot \frac{\pi}{3}, \\ n &= INT(Fnor(v) \cdot 20) + 11, \\ i_0 &= INT(x \cdot 20)(mod \ n), \\ forward &= INT(Fnor(v) \cdot 20)(mod \ n), \\ back &= INT(Fnor(u) \cdot 20)(mod \ n), \\ c_0 &= 1, \end{aligned}$$

where $INT(\cdot)$ is an Integral function. Sequence $\{x_j\}$ merged with sequence $\{q_j\}$ in a nonlinear way, and the ultimate output is

$$y_j = \eta |\cos(j)| x_j + (1 - \eta |\cos(j)|) q_j, \tag{16}$$

where η is a proportion parameter and $\eta \in (0, 0.2)$.

3.1.4. The Digital Compound Chaotic System Expression

If a realized chaotic system executes on a digital device and precision of the digital device is S bit, a quantization function $B_S(x)$ is defined to analyze dynamical behavior in digital chaotic systems. Each process of specific calculation shall be quantized; for instance, $y = x + z$ is quantified to an expression of the following form:

$$y = B_S(B_S(x) + B_S(z)).$$

In order to present a digital chaotic system expression, we have left out some details and the expression of system (6) is:

$$x_{n+1} = B_S \left(t \left(a(\alpha) \sqrt{\frac{1}{t} x_n + m - \left(\frac{1}{t} x_n + m \right)^3 - m} \right) \right). \tag{17}$$

The expression of the disturbed system (15) is as following:

$$\begin{aligned} x_{n+1} &= B_S \left(t \left(a(B_S(T_2(v_{i+1}))) \right. \right. \\ &\quad \left. \left. \cdot \sqrt{\frac{1}{t} B_S(T_1(u_{i+1}, x_n)) + m - \left(\frac{1}{t} B_S(T_1(u_{i+1}, x_n)) + m \right)^3 - m} \right) \right). \end{aligned} \tag{18}$$

According to Equation (16), the digital compound chaotic system is as following:

$$y_{n+1} = B_S(\eta|\cos(n+1)|)x_{n+1} + B_S(1 - \eta|\cos(n+1)|)B_S(q_{n+1}). \quad (19)$$

3.2. Performance Evaluation of the Compound Chaotic System

3.2.1. Analysis under Finite Precision

For performance assessment, we have implemented the original chaotic system (17) and the compound chaotic system in the simulation environment. The digital device is assumed to be an S -bit machine, and the quantization function B_S is expressed as $B_S(x) = \lceil x \cdot 2^S \rceil / 2^S$, where $\lceil x \rceil$ represents an integer less than or equal to x . Trajectories of different chaotic systems are shown in Figure 7. According to Figure 7(2), with finite precision 8 bits, trajectory of the digital *Ecsys* fall into periodic motion after several iterations. According to Figure 7(3), for the digital compound chaotic system, periodic motion do not occur as 8 bits. So, the improved system still maintains strong random characteristics and good chaotic dynamics performance.

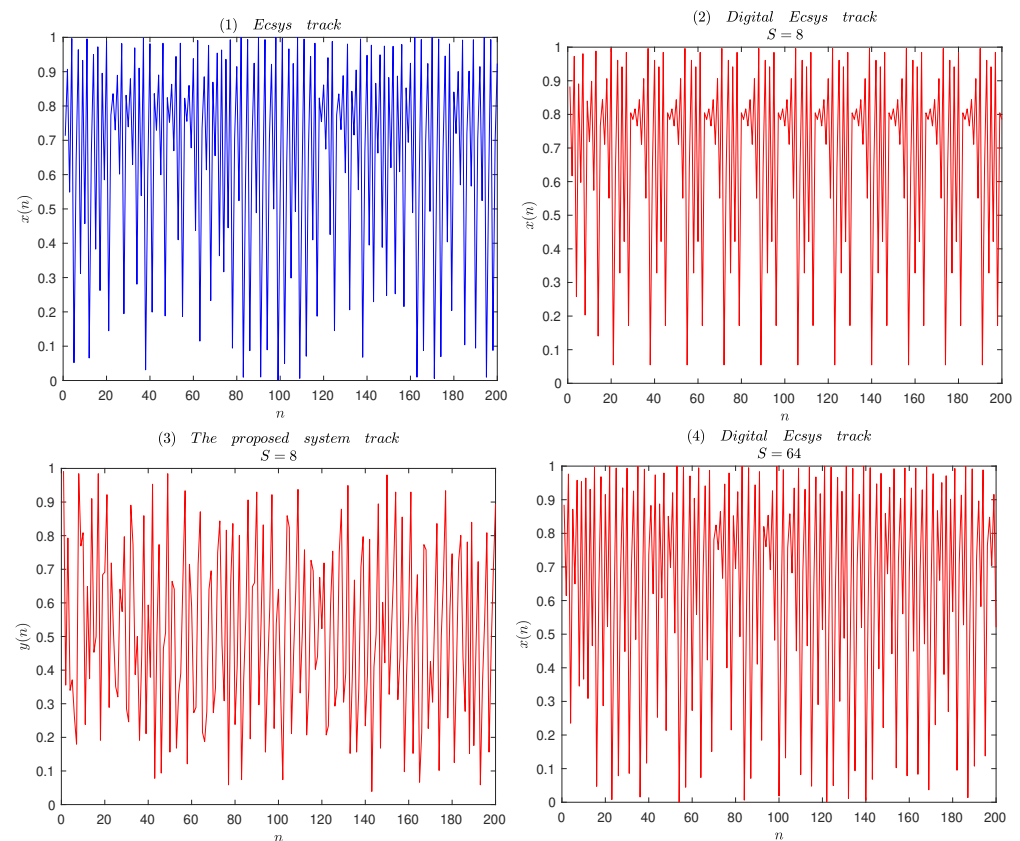


Figure 7. Trajectories of different chaotic systems.

3.2.2. Time Series Complexity

Approximate Entropy is to evaluate the system complexity from sequence generated by chaotic system [24]. For a sequence, the greater the approximate entropy, the higher the complexity. In literature [24], parameters for approximate entropy calculation are recommended that: mode dimension ($m = 2$), similarity tolerance ($r = 2$). The approximate entropy values of the sequence generated by the chaotic maps are calculated and shown in Figure 8.

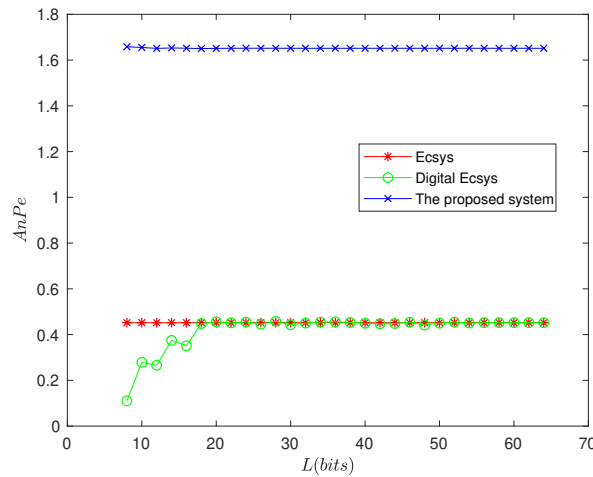


Figure 8. Approximate entropy.

From Figure 8, we can see that the approximate entropy values generated by the digital compound chaotic system are the largest ones among the three maps in the cases of different precisions of the digital device. In particular, low-precision digital device does not affect the new system dynamics performance; so, there is no need for additional precision compensation technology support in practical applications.

3.2.3. Histogram Analysis

Histogram is a significant feature in analysis to the sequence generated by chaotic system. For a good chaotic system for encryption algorithm, the output chaotic sequence distributes uniformly in the whole state space. For the digital chaotic system, the same initial values are setup as different finite precisions, such as 8 bits and 64 bits. Sequence with length of 2424 numbers is generated respectively, and the distribution of the sequence is statistically analyzed. The statistical results are shown in Figure 9. It can be obtained from Figure 9(1–3): The distributions of original *Ecsys* system and digital system are mainly concentrated in the region (0.9, 1.0), uneven with 64 bits precision. From Figure 8, one can see that the proposed system output distributes uniformly in the total region (0, 1) with low precision. So, The proposed system and digital proposed system can both resist statistical attack.

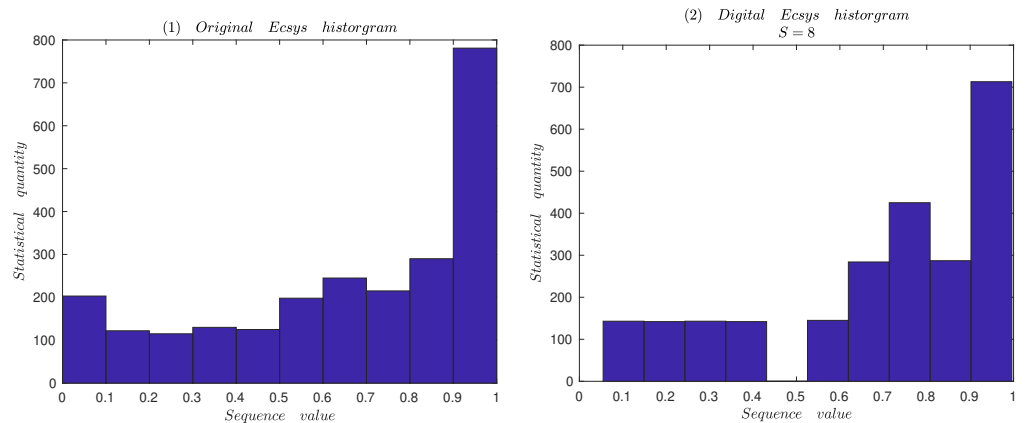


Figure 9. Cont.

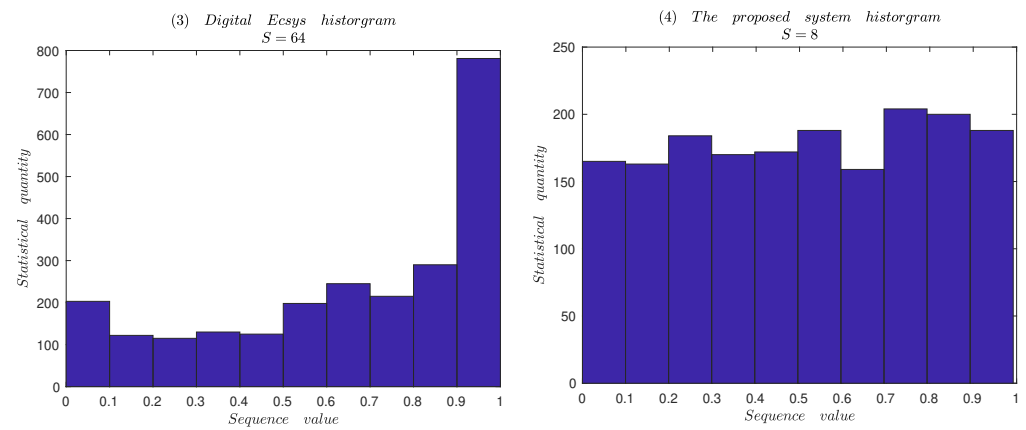


Figure 9. Histograms of the sequences generated by the digital chaotic system.

3.2.4. Autocorrelation Analysis

Autocorrelation is used to measure the relation between current value and past values of the same element. For sequence generated by a chaotic system, it is measuring value between own and its own shifted. It determines the presence of any repetitive patterns of bits.

If a sequence $\{y_1, y_2, \dots, y_N\}$, the autocorrelation function for lag k is as following:

$$r_k = \frac{1}{N} \sum_{i=1}^{N-k} (y_i - \mu)(y_{i+k} - \mu), \tag{20}$$

where μ, σ are the mean and the standard deviation of the sequence. The autocorrelation of sequence generated by different chaotic systems is calculated for 2500 shifts in left and is plotted in Figure 10. It can be seen from Figure 10 that autocorrelation value of the original system is in the range of $(-0.1, 0.1)$, while the digital compound chaotic system with finite precision is in the range of $(-0.05, 0.05)$. Therefore, it can be concluded that the proposed compound chaotic system has lower autocorrelation and better correlation analysis attacks.

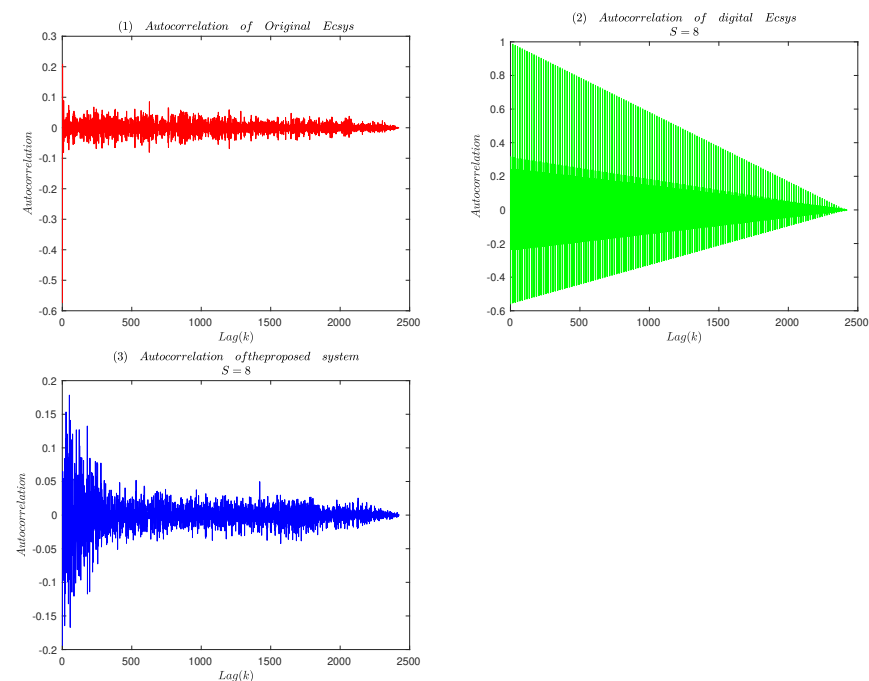


Figure 10. Autocorrelation analysis.

4. Design and Performance Analysis of Pseudo Random Number Generator

It is a requirement for cryptographic applications to construct pseudorandom number generator based on chaotic system [25]. However, due to the lack of strict security analysis, the PRNG based on the original chaos often has some security vulnerabilities [14,26]. If the algorithm for PRNG is reasonably designed or the PRNG is designed based on the proposed system with more chaotic behavior, resistance against the finite precision effect, and larger key space, the PRNG should have good random characteristics, security and effectiveness.

4.1. Design of PRNG Based on the Proposed Compound Chaotic System

A PRNG can be used for any application if the PRNG has some properties such as good statistical properties, long cycle length, larger key space, etc. In order to achieve a fast throughput and make easier hardware or soft implementation, mechanism with p bit accuracy is adopted. The steps of algorithm for generating pseudo-random numbers are as following:

1. Import the keys: initialize $(u_0, v_0), k_{14}$ and x_0 , which are the control parameters and initial conditions as shown in Figure 6.
2. Iterate the proposed compound chaotic map 1000 times and the output y_i is discarded, where i end at 999.
3. Generate and output the random number z_n using the following equation from the y_n :

$$z_n = [y_n \times 10^m - \text{floor}(y_n \times 10^m)] \times 2^p,$$

where p is the length of the corresponding binary random number and n start at 1000. The expression $y_n \times 10^m - \text{floor}(y_n \times 10^m)$ excludes m most effective numbers, which makes it more complex and uniform. The value of m is determined by p , and the proposed values are listed in Table 1.

Table 1. The suggested values of p and m .

p	1	4	8	16	32
m	≥ 3	≥ 2	≥ 2	≥ 2	≥ 1

4.2. Analysis and Test of Security for the Proposed PRNG

It's essential for a random number generator to perform all necessary analyses and tests. There are several fundamental analysis and tests to verify the randomness, security and availability of proposed algorithm. The pseudorandomness of sequence generated by RNG is mainly through recurrence plots analysis, information entropy, and random evaluation software to test, etc. The following excerpt is that under $p = 2$ and $m = 2$ conditions, some security characteristics of the PRNG are comprehensively analyzed or tested and the test results with the fine-grained trace are carried out.

4.2.1. Key Space Analysis

Random number generator is mainly used to generate key and from an encryption point of view, the size of the key should not be less than 2^{128} to provide a high level of security [27]. We select (u_0, v_0, k_{14}, x_0) as the key set. These parameters should be selected from the control parameters and initial conditions of the chaotic region, which depends on the system bifurcation diagram. Because bifurcation diagram is used to describe mutations in system dynamics. Thus, a set of keys are given based on the bifurcation diagram of system (4), including: $u_0 \in [-0.9, 0.3], v_0 \in [-0.4, 1.2], k_{14} \in [1.47, 1.57]$ and $x_0(0.0, 1.0)$. If the computational accuracy of the actual system applied is 10^{16} , the range of the secret key space can be roughly calculated as following:

$$(6 \times 10^{15})(8 \times 10^{15})(10^{15})(10^{16}) = 4.8 \times 10^{62} \simeq 2^{208}.$$

It can be concluded that the size of the key space is sufficient to resist all kinds of violent attacks.

4.2.2. Correlation Analysis

For a PRNG, the correlation coefficients between sequences produced with nearby keys are computed according to the method in [28]. For two sequences $S_1 = [x_1, x_2, \dots, x_N]$ and $S_2 = [y_1, y_2, \dots, y_N]$, coefficients are calculated as following:

$$Cor(S_1, S_2) = \frac{\sum_{i=1}^N (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\left[\sum_{i=1}^N (x_i - \bar{x})^2\right]^{1/2} \left[\sum_{i=1}^N (y_i - \bar{y})^2\right]^{1/2}}, \tag{21}$$

where \bar{x} and \bar{y} are the mean values of x and y respectively. Correlation is strong between two sequences for $Cor(S_1, S_2) \simeq 1$ and no or very small correlation corresponds to $Cor(S_1, S_2) \simeq 0$.

Let $p = 32$ and $m = 2$; the correlation coefficients test are performed as following:

1. $x_{01} = 0.205001025, u_{01} = 0.102772828$ and $v_{01} = 0.118667888$, a sequence with 10^6 numbers is generated. If the initial condition changes small ($x_{02} = x_{01} + 0.0000000000000001$) and the others remain unchanged, a new sequence will be generated.
2. Let the initial condition u is changed ($u_{02} = u_{01} + 0.0000000000000001$) and the others remain unchanged, a new sequence with the same length will be generated.
3. Let the initial condition v is changed ($v_{02} = v_{01} + 0.0000000000000001$) and the others remain unchanged, a new sequence with the same length will be generated.

Correlation coefficients data are calculated by formula (21) and listed in Table 2. By analyzing the data in Table 2, the following conclusions can be drawn: there is no correlation between the generated sequences produced the proposed PRNG is sensitive to small changes in all initial conditions.

Table 2. Correlation coefficients of three pairs of pseudo random sequences.

		<i>Cor</i>
$x_{01} = 0.205001025$	$x_{02} = 0.2050010250000001$	−0.00290
$u_{01} = 0.102772828$	$u_{02} = 0.2050010250000001$	−0.00230
$v_{01} = 0.102772828$	$v_{02} = 0.2050010250000001$	−0.00016

4.2.3. Recurrence Plots Analysis

A powerful tool is given in [29] for visualization and analysis of recurrence, called recurrence plot (RP). As analyzing nonlinear time series by RP, phase space reconstruction is the first step. The phase space reconstruction is carried out by selecting appropriate time delay τ and embedding dimension m . For a time sequence $\{x_n, n = 1, 2, \dots, N\}$, a set of m dimensional vectors is obtained after phase space reconstruction:

$$X(n) = (x(n), x(n + \tau), \dots, x(n + (m - 1)\tau)), \tag{22}$$

$$n = 1, 2, \dots, N - (m - 1)\tau.$$

The distance between m -dimensional vector $X(i)$ and $X(j)$ at two moments is r_{ij} , that is:

$$r_{ij} = \|X(i) - X(j)\|.$$

It can be defined as the following recursive matrix form:

$$R_{ij}(\xi) = \theta(\xi - r_{ij}), i, j = 1, 2, \dots, N - (m - 1)\tau,$$

where θ is Heaviside function and ξ is threshold. Heaviside function is expressed as following:

$$\theta(x) = \begin{cases} 1, & x = \xi > r_{ij} > 0, \\ 0, & x = \xi \leq r_{ij} < 0. \end{cases}$$

For the recursive matrix, it means that the states at time i and time j are obviously different (obviously similar) when $R_{ij} = 0$ ($R_{ij} = 1$). The corresponding RP can be drawn according to the recursive matrix. RP can intuitively reflect the movement rule and trend of time sequences. In the actual calculation, the threshold ξ is 0.1 times of the standard deviation of the time series [30].

RP can directly show the motion rule of dynamical systems. However, RP cannot quantify the system characteristics because of the small-scale structure [31]. Recurrence quantification analysis (RQA) precisely quantifies these characteristics. RQA is proposed by Webber and Zbilut in the literature [32]. RQA is a quantitative analysis of the sequence by extracting structural feature quantities by analyzing the detailed structure of the RP. The main feature quantities are recursive rate (RR), a measure for determinism (DET), layered degrees (LAM), trapping time (TT) and the average diagonal line length (L). Small values of feature quantities for dynamical system represent a process with weakly correlated and chaotic behaviors.

The recursion rate represents the proportion of adjacent vectors in RP and it is the percentage of recursion points in the total number of points:

$$RR = \frac{1}{N^2} \sum_{i,j=1}^N R_{i,j}(\xi),$$

where N is the number of points on the abscissa of the RP. The histogram $DLS(\xi, l)$ of the diagonal with length l in RP is as following:

$$DLS(\xi, l) = \sum_{i,j=1}^N (1 - R_{i-1,j-1}(\xi)) (1 - R_{i+l,j+l}(\xi)) \prod_{k=0}^{l-1} R_{i+k,j+k}(\xi).$$

DET is the ratio of recurrence points that form diagonal structures and it is given by

$$DET = \frac{\sum_{l=l_{min}}^N l \cdot DLS(\xi, l)}{\sum_{l=1}^N l \cdot DLS(\xi, l)},$$

where l_{min} is the minimum length requirement (length threshold) and value is generally 2. In the RP and during l time steps, a diagonal line with length L means that a segment of system trajectory is close to another at different moments. The average diagonal length is the average time that two segments are close to each other, which can be interpreted as the average prediction time

$$L = \frac{\sum_{l=l_{min}}^N l \cdot DLS(\xi, l)}{\sum_{l=l_{min}}^N DLS(\xi, l)}.$$

For calculating the number of vertical line segments $VLS(v)$ with length v , a method is as following:

$$VLS(v) = \sum_{i,j=1}^N (1 - R_{i,j}(\xi)) (1 - R_{i,j+v}(\xi)) \prod_{k=0}^{v-1} R_{i,j+k}(\xi).$$

Laminar degree (LAM) is the ratio between the recurrence points forming the vertical structures and the entire set of recurrence points, and it can be computed,

$$LAM = \frac{\sum_{v=v_{min}}^N v \cdot VLS(v)}{\sum_{v=1}^N v \cdot VLS(v)},$$

where v_{min} is the threshold of vertical line length (generally 2). Capture time (TT) is the average length of the vertical line segment in RP, and it is expressed as following:

$$TT = \frac{\sum_{v=v_{min}}^N v \cdot VLS(v)}{\sum_{v=v_{min}}^N VLS(v)}$$

Capture time (TT) is used to estimate the average duration of a system in a particular state.

Figure 11 shows RQA measures as parameter k_{14} is changed, the threshold $\zeta = 0.1\sigma$ (σ as the standard deviation of the sequence), $l_{min} = v_{min} = 2$, and the initial conditions ($x_0 = 0.205001025, u_0 = 0.102772828, v_0 = 0.118667888$). Processes with uncorrelated or weakly correlated and stochastic or chaotic behaviors cause none or short diagonals, whereas deterministic processes cause longer diagonals and less single, isolated recurrence points [18]. It can be seen from Figure 11 that RR typical measurement value is 0.5, DET value is 0.65, LAM value is 0.03 and TT value is 3. Obviously, these values are small and the proposed PRNG has good randomness.

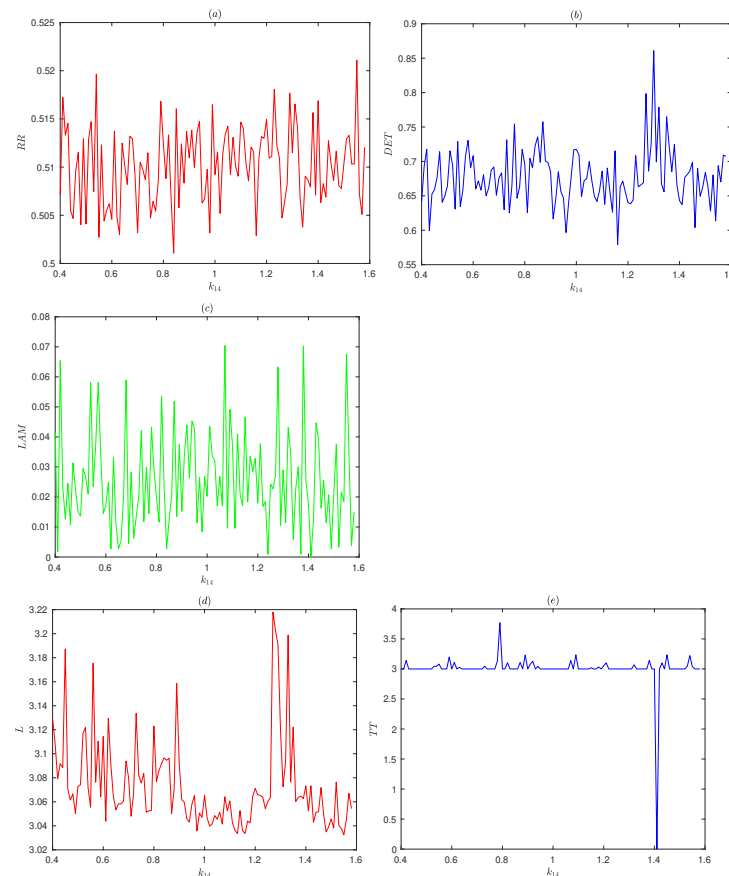


Figure 11. RQA measures for the proposed PRNG.

4.2.4. Information Entropy

One of the most important concepts in information theory is entropy, which is first introduced by Shannon [33]. It reflects the uncertainty and randomness of each information system. Entropy measures the unpredictability of a sequence generated by a PRNG. For a sequence $S = \{x_1, x_2, \dots, x_n\}$, the definition of information entropy $H(S)$ is as follows:

$$H(S) = \sum_{i=1}^N p(a_i) \log_2 \frac{1}{p(a_i)}, \tag{23}$$

where number a_i has the probability of $p(a_i)$ to occur in sequence S . In actual calculation, a corresponding character sequence is generated for every 8 bits by original binary sequence.

The information entropy is calculated according to Formula (23). For a sequence of bytes, the ideal value of information entropy is 8.

Sequences of varying lengths and the different initial conditions are generated by the proposed PRNG, and their entropy values are shown in Table 3. The values of the different initial conditions with $k_{14} = 1.55$ are: S_1 ($x_0 = 0.105001025$, $u_0 = 0.201772828$, $v_0 = 0.218667888$), S_2 ($x_0 = 0.105001025$, $u_0 = 0.201772828$, $v_0 = 0.218667888$) and S_3 ($x_0 = 0.505001025$, $u_0 = 0.251772828$, $v_0 = 0.418667088$). Sequences with lengths of 5000, 10000, 20000, 100000 and 10^7 are generated. Table 3 shows that each information entropy value is close to the ideal value 8.

Table 3. Information entropy.

Len_{Seq}	5000	10000	20000	10^5	10^7
$H(S_1)$	7.9904252203	7.9953919349	7.9975813926	7.9994779223	7.9999951973
$H(S_2)$	7.9899963471	7.9949233070	7.9974845070	7.9994868500	7.9999952355
$H(S_3)$	7.9895789685	7.9947618385	7.9970912252	7.9995475969	7.9999954916

4.2.5. Statistical Complexity Measure

Complexity is a measure of off-equilibrium ‘order’ [18]. Statistical complexity measure (SCM) is proposed as quantifiers of the degree of physical structure in a signal [34]. SCM can be used to study the complex structure hidden in chaotic system. In literature [35], the statistical complexity of the presented algorithm is calculated. The probability distribution P is associated with the time series generated by the dynamical system. The intensive SCM ($C_j[P]$) can be considered as a quantity that characterizes the probability distribution P which not only quantifies the randomness but also presents the structure. $C_j[P]$ is defined based on information entropy as following:

$$C_j[P] = Q_j[P, P_e] \cdot H_S[P], \tag{24}$$

where $H_S[P]$ is the entropic measure and Q_j is “disequilibrium”. $H_S[P]$ and Q_j are defined in [36]. Q_j is given by

$$Q_j[P, P_e] = Q_0 \cdot \{H[(P + P_e)/2] - H[P]/2 - H[P_e]/2\},$$

where Q_0 is a normalization constant that reads

$$Q_0 = -2 \left\{ \left(\frac{N+1}{N} \right) \ln(N+1) - 2 \ln(2N) + \ln N \right\}^{-1}.$$

$H_S[P]$ is defined as following:

$$H_S[P] = H[P] / H_{max}, \tag{25}$$

where $H(P)$ is the Shannon entropy. For an extremely good PRNG based on chaotic system, it can be expected that “no attractor” will be reconstructed. It will be quite reasonable to obtain a homogeneity cloud of points with a tendency to fill the d-dimensional space [35]. Consequently, the associated permutation probability distribution will be $P \simeq P_e$ [9]. So, in the case of a PRNG, the “ideal” values are $H_S[P] \simeq 1$ and $C_j[P] \simeq 0$.

If entropy H_S and the intensive statistical complexity C_j are as functions of the number of 8 bits-words, then $H_{max} = 8$, $N = 256$ and $P_e = \{1/N, 1/N, \dots, 1/N\}$. Based on the calculations mentioned above, the normalized entropy H_s and the intensive statistical complexity C_j as functions of the number of 8 bits are shown in Figure 12. It can be obtained from the Figure 12 that C_j and H_s tend to 0 and 1, respectively. So, the proposed PRNG is successfully verified by the statistical complexity and the normalized Shannon entropy.

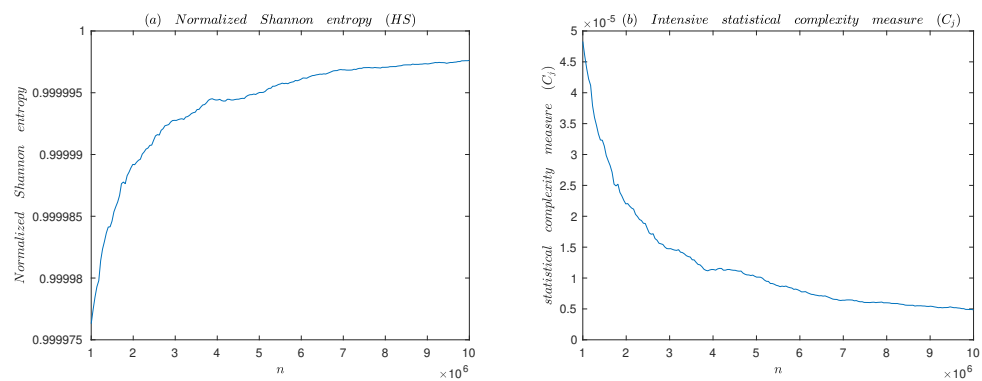


Figure 12. Normalized Shannon entropy H_s and intensive SCMC $_j$ for the proposed PRNG.

4.2.6. Degree of Non-Periodicity

Wavelet analysis is a valuable tool for the study of dynamic systems. The Scale index technique and the windowed Scale index are based on the continuous wavelet transform and the wavelet multi-resolution analysis [37,38]. The tools are designed to measure the degree of non-periodicity through its wavelet scalogram, allowing to quantify how much chaotic a signal is [38].

In order to detect and study nonperiodicity in sequences generated by PRNG, we can regard the PRNG as a continuous function $f \in L^2(\mathbb{R})$, where f defines the time interval at a finite time interval $I = [a, b]$ and I is large enough [18]. The Continuous Wavelet Transform (CWT) of f at time u and scale s is defined as following:

$$Wf(u, s) := \langle f, \psi_{u,s} \rangle = \int_{-\infty}^{+\infty} f(t) \psi_{u,s}^*(t) dt,$$

and it provides the frequency details of the function corresponding to scale s and time location u . The scalogram of f at a given scale s is given by

$$S(s) = \|Wf(u, s)\| = \left(\int_{-\infty}^{+\infty} |Wf(u, s)|^2 du \right)^{\frac{1}{2}}.$$

$S(s)$ is the energy of the CWT of f at scale s . The scalogram is a useful tool for studying a signal because it can detect the most representative scales or frequencies. The inner scalogram of f at a scale s can be defined by:

$$S^{inner}(s) = \|Wf(u, s)\|_{J(s)} = \left(\int_{c(s)}^{d(s)} |Wf(u, s)|^2 du \right)^{\frac{1}{2}},$$

where $J(s) = [c(s), d(s)] \subseteq I$ is the maximal subinterval in I for which the support of $\psi_{u,s}$ is included in I for all $u \in J(s)$. Let l be the length of $\psi_{u,s}$ and $b - a \gg sl$ must also be satisfied. Since the length of $J(s)$ depends on the scale s , the values of the inner scalogram do not be compared at different scales. In order to avoid this problem, the inner scalogram should be normalized as follows:

$$\bar{S}^{inner}(s) = \frac{S^{inner}(s)}{(d(s) - c(s))^{\frac{1}{2}}}.$$

In [38], the new Scale index of f in the scale interval $[s_0, s_1] \subseteq I$ is given by the quotient

$$i_{scale} = \frac{\bar{S}^{inner}(s_{min})}{\bar{S}^{inner}(s_{max})},$$

where $s_{max} \in [s_0, s_1]$ is the maximal scale such that $\mathcal{S}^{inner}(s_{max}) \geq \mathcal{S}^{inner}(s)$ for all $s \in [s_0, s_1]$, and $s_{min} \in [s_{max}, 2s_1]$ is the smallest scale such that $\mathcal{S}^{inner}(s_{min}) \leq \mathcal{S}^{inner}(s)$ for all $s \in [s_{max}, 2s_1]$. From its definition, the scale index i_{scale} is such that $0 \leq i_{scale} \leq 1$, and it can be interpreted as a measure of the degree of nonperiodicity of the signal [37,38].

Let haar wavelet be mother wavelet function to calculate the Scale index i_{scale} , where $\Delta s = \Delta t = 0.05s$, $s_0 = 1$ and $s_1 = 20$. Figure 13 shows the Scale index analysis of the proposed PRNG, Henon map and the logistic map. It is apparent from comparison of Figures 12 and 13, that the Scale index of the proposed PRNG is higher than other two chaotic maps. Thus, the generated sequence of the proposed PRNG is highly nonperiodic.

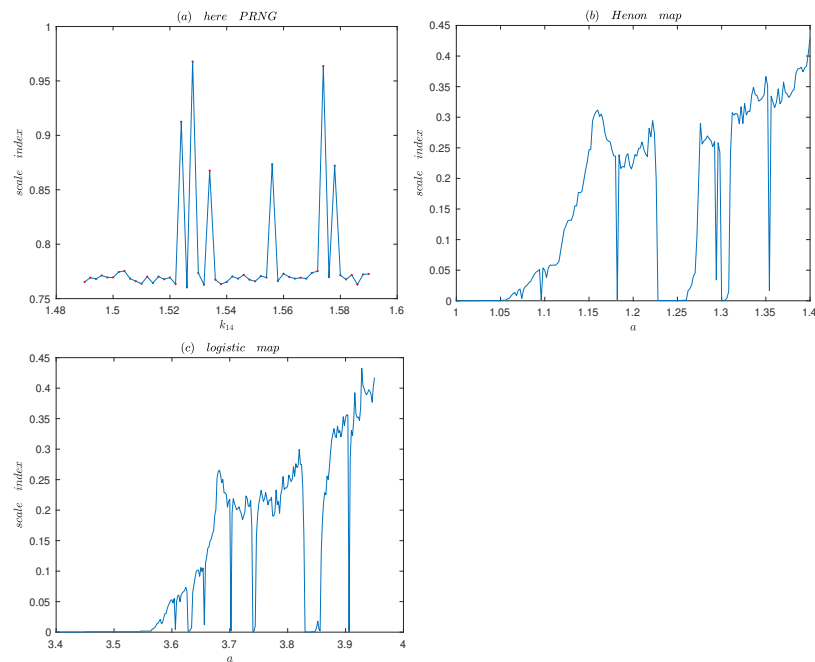


Figure 13. The Scale index of the Henon map, the logistic map and the proposed PRNG.

The windowed Scale index is appropriate for nonstationary time series whose characteristics change over time [38]. It is based on the windowed scalogram and the scale index. The windowed scalogram of a time series f is given by [39]:

$$\mathcal{WS}_\tau(t, s) = \left(\int_{t-\tau}^{t+\tau} |Wf(u, s)|^2 du \right)^{\frac{1}{2}},$$

where f is centered at time t with radius τ . The windowed scale index of a time series f centered is defined as

$$wi_{scale, \tau}(t) = \frac{\mathcal{WS}_\tau(t, s_{min})}{\mathcal{WS}_\tau(t, s_{max})},$$

where s_{max} is the smallest scale such that $\mathcal{WS}_\tau(t, s_{max}) \geq \mathcal{WS}_\tau(t, s)$ for all $s \in [s_0, s_1]$, and s_{min} is the smallest scale such that $\mathcal{WS}_\tau(t, s_{min}) \leq \mathcal{WS}_\tau(t, s)$ for all $s \in [s_{max}, 2s_1]$.

In general, if $[a, b]$ is the support of f , $\tau = (b - a)/20$ is a good choice[38]. Figure 14 shows the windowed scale index analysis of the proposed PRNG ($k_{14} = 1.55$), Henon map ($b = 0.3, a = 1.155$) and the logistic map ($a = 3.88$). As can be seen in Figure 14, the windowed Scale index clearly shows the evolution over time. Through comparative analysis, it can be obtained that window scale index of the sequence generated by the proposed PRNG changes over time mainly above 0.75, which is much higher than the other two classical chaos maps. Thus, the generated sequence of the proposed PRNG is highly nonperiodic over time.

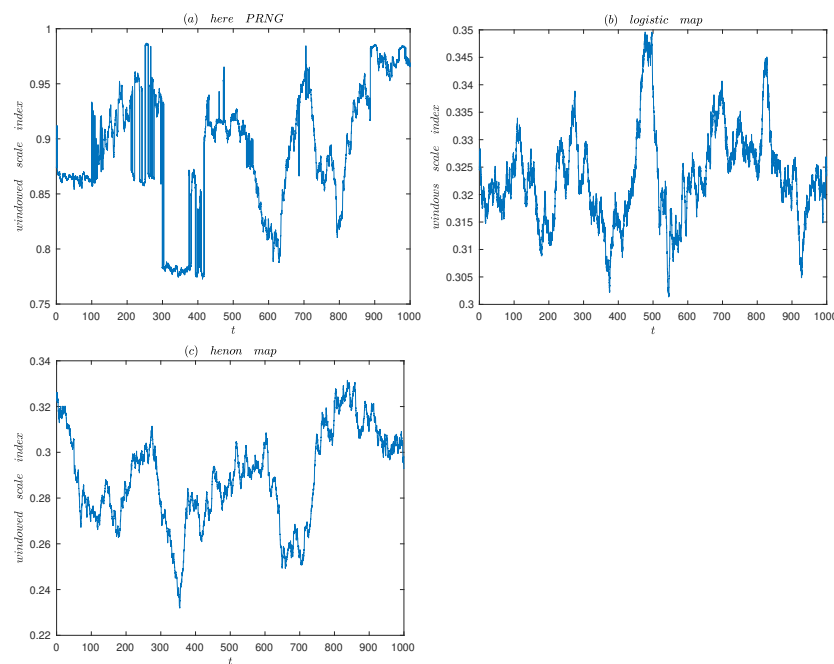


Figure 14. Windowed Scale index of the Henon map, the logistic map and the proposed PRNG.

4.2.7. Differential Attack

Differential attack is that the effect of corresponding ciphertexts is analyzed as small changes on the plaintext. For a PRNG, it is applied the same analysis on the initial seeds which are at the same time keys because there is no plaintext. “Bit Change Rate (BCR)” is carried out to ensure the resistance of the proposed PRNG against the differential attack.

Bit Change Rate (BCR) criterion is defined as

$$BCR(S_1, S_2) = \frac{Bit_{Diff}[S_1, S_2]}{N} \times 100\%, \tag{26}$$

where $S_1(S_2)$ is the sequence generated on the initial seed “seed₁” (“seed₂”) and N is the generated sequence length and $\|seed_1 - seed_2\| < \epsilon$, which represents the small change between seed keys. $Bit_{Diff}[S_1, S_2]$ is the number of different bits in S_1 and S_2 . If the measure of BCR for the two sequences is close to 50, it indicates the two sequences are almost completely different. The results of BCR for sequences with 10^7 bit lengths, is displayed with small changes (10^{-15}) on initial conditions in Table 4. From Table 4, it can be clearly seen that the BCR values are close to 50 percent. Hence, the proposed PRNG is sensitive to the change of seeds, and it can be concluded that the presented PRNG is highly resistive against differential attack.

Table 4. Key sensitivity evaluation based on Bit Change Rate (BCR).

		<i>BCR</i>
$x_{01} = 0.105001025$	$x_{02} = 0.105001025000001$	49.9994
$u_{01} = 0.201772828$	$u_{02} = 0.201772828000001$	50.0209
$v_{01} = 0.218667888$	$v_{02} = 0.218667888000001$	50.0014

4.2.8. Random Tests

To examine the randomness of sequence generated by presented PRNG, NIST SP800-22 is carried out. The soft test suit includes 17 independent statistical tests, which focus on a sort of different types of nonrandomness in sequence. This software mainly uses performance indicator p -value which determined the random performance of the sequence. If the p -value of sequence is higher than the threshold α (the significance level), it means

that the sequences pass the test. In our tests, a bit sequence is generated, which had the length of 100×10^6 bits and the bit sequence was divided into 100 subsequences. α is 0.01, which implies that the sequence can be inferred to be random with 99% probability if it passes the test.

By this way, results from all statistical tests are given in Table 5. From Table 5, the results of sequence generated by proposed PRNG are all “success”. Hence, the proposed PRNG successfully passed the NIST SP800-22 tests.

Table 5. Randomness test by NIST SP800-22 for the PRNG.

<i>Test Name</i>	<i>p-Value</i>	<i>Pass Rate</i>	<i>Results</i>
Frequency	0.657933	98/100	Pass
Block Frequency (m = 128)	0.051942	100/100	Pass
Cumulative Sums (Forward)	0.224821	98/100	Pass
Cumulative Sums (Reverse)	0.455937	98/100	Pass
Runs	0.514124	100/100	Pass
Longest Run of Ones	0.262249	100/100	Pass
Rank	0.955835	98/100	Pass
FFT	0.883171	98/100	Pass
Non-Overlapping Templates (m = 9, B = 000000001)	0.867692	100/100	Pass
Overlapping Templates (m = 9)	0.574903	100/100	Pass
Universal	0.474986	99/100	Pass
Approximate Entropy (m = 10)	0.574903	100/100	Pass
Random-Excursions (data3)	0.422034	67/67	Pass
Random-Excursions Variant Serial (data5)	0.922036	67/67	Pass
Serial Test 1 (m = 16)	0.883171	99/100	Pass
Serial Test 2 (m = 16)	0.350485	100/100	Pass
Linear complexity (M = 500)	0.275709	99/100	Pass

4.2.9. Speed Performance Analysis

Speed of data generated by PRNG is an important factor for evaluating the performance. For the proposed PRNG, the time cost is measured in the running environment: Centos 7, Intel I5-6300U CPU, 4GB RAM and MATLAB 2018a software framework. We measured the time cost in the running environment. We set the parameters and 100 bit sequence are generated, each of which is 500,000 bits in length. Table 6 compares the speed of the proposed PRNG in terms of the number of bits generated with other PRNG schemes. Compared with other PRNG schemes, the proposed PRNG is fast enough for practical application.

Table 6. Speed comparison

PRNG	Mean Time Cost
Proposed PRNG	0.3514
Ref. [21]	3.8989
Ref. [40]	108.1568
Chebyshev map	0.8056
Ref. [41]	2.2114
LFSR with discrete chaotic map [42]	3.4530

5. Conclusions and Discussion

In a summary, we proposed a new scheme with perturbation and mixture together to optimize the one-dimensional chaotic map self-constructed. We obtain a compound chaotic system using this scheme and the new system is evaluated using some test methods such as time series complexity, autocorrelation and distribution of output frequency. The test

results showed that the new system has high randomness and the system can operate well in the environment of low precision equipment. Thus, it can be concluded that the new scheme can be used to design a compound chaotic system with complex dynamic behavior. When using the compound chaotic system in a PRNG, designer can easily achieve high security and good quality of the random bit sequences.

A pseudorandom number generator (PRNG) is specifically designed based on the new chaos. The PRNG merely relies on the equations used in compound chaotic map. The algorithm is not complex, which does not impose high hardware requirement and thus speed is fast. The proposed PRNG exhibits excellent security property in terms of quantifiers based on information theory, recurrence plots, nonperiodicity, correlation analysis, differential attack and NIST tests. By these test results, we conclude that our PRNG is a reliable PRNG and it can generate highly available random numbers for various applications in computer science.

Author Contributions: Conceptualization, C.M. and W.Z.; formal analysis and investigation, C.M. and W.Z.; discussion and suggestion, C.M. and Z.C.; writing—original draft preparation, C.M. and W.Z.; writing—review and editing, C.M., W.Z., Z.C. and Z.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (No. 12161077), Natural Science Foundation of Gansu Province (No. 21JR7RE172, 22JR11RE193) and the innovation Fund Project of University in Gansu Province (No. 2021B-218, 2021B-219, 2021QB-109).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PRNG	Pseudorandom number generator
RP	Recurrence plot
CWT	Continuous Wavelet Transform
BCR	Bit Change Rate
SCM	Statistical complexity measure

References

- Jafari Barani, M.; Ayubi, P.; Yousefi Valandar, M.; Irani, B.Y. A new Pseudo random number generator based on generalized Newton complex map with dynamic key. *J. Inf. Secur. Appl.* **2020**, *53*, 102509. [\[CrossRef\]](#)
- Lambić, D. S-box design method based on improved one-dimensional discrete chaotic map. *J. Inf. Telecommun.* **2018**, *2*, 181–191. [\[CrossRef\]](#)
- Valandar, M.Y.; Barani, M.J.; Ayubi, P.; Aghazadeh, M. An integer wavelet transform image steganography method based on 3D sine chaotic map. *Multimed. Tools Appl.* **2019**, *78*, 9971–9989. [\[CrossRef\]](#)
- Liu, L.; Hu, H.; Deng, Y. An analogue–digital mixed method for solving the dynamical degradation of digital chaotic systems. *IMA J. Math. Control Inf.* **2015**, *32*, 703–716. [\[CrossRef\]](#)
- James, F.; Moneta, L. Review of High-Quality Random Number Generators. *Comput. Softw. Big Sci.* **2020**, *4*, 2. [\[CrossRef\]](#)
- Jiang, N.; Dong, X.; Hu, H.; Ji, Z.; Zhang, W. Quantum Image Encryption Based on Henon Mapping. *Int. J. Theor. Phys.* **2019**, *58*, 979–991. [\[CrossRef\]](#)
- Yu, F.; Li, L.; Tang, Q.; Cai, S.; Song, Y.; Xu, Q. A Survey on True Random Number Generators Based on Chaos. *Discret. Dyn. Nat. Soc.* **2019**, *2019*, 2545123. [\[CrossRef\]](#)
- Wang, J.; Zhang, M.; Tong, X.; Wang, Z. A chaos-based image compression and encryption scheme using fractal coding and adaptive-thresholding sparsification. *Phys. Scr.* **2022**, *97*, 105201. [\[CrossRef\]](#)
- 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.* **2014**, *19*, 101–111. [\[CrossRef\]](#)

10. Lambić, D.; Janković, A.; Ahmad, M. Security Analysis of the Efficient Chaos Pseudo-random Number Generator Applied to Video Encryption. *J. Electron. Test. Theory Appl. (JETTA)* **2018**, *34*, 709–715. [[CrossRef](#)]
11. EL-Latif, A.A.A.; Abd-El-Atty, B.; Venegas-Andraca, S.E. Controlled alternate quantum walk-based pseudo-random number generator and its application to quantum color image encryption. *Phys. A Stat. Mech. Its Appl.* **2020**, *547*, 123869. [[CrossRef](#)]
12. Stoyanov, B.; Kordov, K. Novel secure pseudo-random number generation scheme based on two Tinkerbell maps. *Adv. Stud. Theor. Phys.* **2015**, *9*, 411–421. [[CrossRef](#)]
13. Li, C.; Lo, K. Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Process.* **2011**, *91*, 949–954. [[CrossRef](#)]
14. Ahmad, M.; Alam, M.Z.; Ansari, S.; Lambić, D.; AlSharari, H.D. Cryptanalysis of an image encryption algorithm based on PWLCM and inertial delayed neural network. *J. Intell. Fuzzy Syst.* **2018**, *34*, 1323–1332. 3. [[CrossRef](#)]
15. Venegas-Andraca, S. Quantum walks: A comprehensive review. *Quantum Inf. Process.* **2012**, *11*, 1015–1106. [[CrossRef](#)]
16. Dernbach, S.; Mohseni-Kabir, A.; Pal, S.; Towsley, D. Quantum Walk Neural Networks for Graph-Structured Data. In Proceedings of the Complex Networks and Their Applications VII, Cambridge, UK, 11–13 December 2018; Aiello, L.M., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L.M., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 182–193.
17. Rigovacca, L.; Di Franco, C. Two-walker discrete-time quantum walks on the line with percolation. *Sci. Rep.* **2016**, *6*, 22052. [[CrossRef](#)]
18. Yang, Y.; Zhao, Q. Novel pseudo-random number generator based on quantum random walks. *Sci. Rep.* **2016**, *6*, 20362. [[CrossRef](#)]
19. Ge, B.; Luo, H. Image Encryption Application of Chaotic Sequences Incorporating Quantum Keys. *Int. J. Autom. Comput.* **2020**, *17*, 123–138. [[CrossRef](#)]
20. Yu, W.B.; Yang, L.Z. Chaos analysis of the conic in planar unit area. *Acta Phys. Sin.* **2013**, *62*, 79–86. [[CrossRef](#)]
21. Liu, Y.; Luo, Y.; Song, S.; Cao, L.; Liu, J.; Harkin, J. Counteracting Dynamical Degradation of Digital Chaotic Chebyshev Map via Perturbation. *Int. J. Bifurc. Chaos* **2017**, *27*, 1750033. [[CrossRef](#)]
22. Shi, Y.; Chen, G. Discrete chaos in Banach spaces. *Sci. China Ser. A Math.* **2005**, *48*, 222–238. [[CrossRef](#)]
23. Aharonov, D.; Ambainis, A.; Kempe, J.; Vazirani, U. Quantum Walks On Graphs. In Proceedings of the 33rd ACM Symposium on Theory of Computing, Crete, Greece, 6–8 July 2001; pp. 50–59. [[CrossRef](#)]
24. Pincus, S. Approximate entropy as a measure of system complexity. *Proc. Natl. Acad. Sci. USA* **1991**, *88*, 2297–2301. [[CrossRef](#)] [[PubMed](#)]
25. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151. [[CrossRef](#)]
26. Hu, H.; Liu, L.; Ding, N. Pseudorandom sequence generator based on the Chen chaotic system. *Comput. Phys. Commun.* **2013**, *184*, 765–768. [[CrossRef](#)]
27. Ecrypt II Yearly Report on Algorithms and Keysizes. 2012. Available online: <http://www.ecrypt.eu.org/documents/D.SPA.20.pdf> (accessed on 9 December 2022).
28. Pareek, N.K.; Patidar, V.; Sud, K.K. Diffusion–substitution based gray image encryption scheme. *Digit. Signal Process.* **2013**, *23*, 894–901. [[CrossRef](#)]
29. Eckmann, J.P.; Kamphorst, S.O.; Ruelle, D. Recurrence Plots of Dynamical Systems. *Europhys. Lett. (EPL)* **1987**, *4*, 973–977. [[CrossRef](#)]
30. Marwan, N.; Wessel, N.; Meyerfeldt, U.; Schirdewan, A.; Kurths, J. Recurrence-plot-based measures of complexity and their application to heart-rate-variability data. *Phys. Rev. E* **2002**, *66 Pt 2*, 026702. [[CrossRef](#)]
31. Marwan, N.; Carmen Romano, M.; Thiel, M.; Kurths, J. Recurrence plots for the analysis of complex systems. *Phys. Rep.* **2007**, *438*, 237–329. [[CrossRef](#)]
32. Webber, C.; Zbilut, J. Dynamical assessment of physiological systems and states using recurrence plot strategies. *J. Appl. Physiol.* **1994**, *76*, 965–973. [[CrossRef](#)]
33. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [[CrossRef](#)]
34. Martin, M.; Plastino, A.; Rosso, O. Statistical complexity and disequilibrium. *Phys. Lett. A* **2003**, *311*, 126–132. . [[CrossRef](#)]
35. Larrondo, H.A.; González, C.M.; Martín, M.T.; Plastino, A.; Rosso, O.A. Intensive statistical complexity measure of pseudorandom number generators. *Phys. A Stat. Mech. Its Appl.* **2005**, *356*, 133–138. [[CrossRef](#)]
36. Lamberti, P.W.; Martin, M.T.; Plastino, A.; Rosso, O.A. Intensive entropic non-triviality measure. *Phys. A Stat. Mech. Its Appl.* **2004**, *334*, 119–131. [[CrossRef](#)]
37. Benítez, R.; Bolós, V.J.; Ramírez, M.E. A wavelet-based tool for studying non-periodicity. *Comput. Math. Appl.* **2010**, *60*, 634–641. [[CrossRef](#)]
38. Bolós, V.J.; Benítez, R.; Ferrer, R. A New Wavelet Tool to Quantify Non-Periodicity of Non-Stationary Economic Time Series. *Mathematics* **2020**, *8*, 844. [[CrossRef](#)]
39. Bolós, V.J.; Benítez, R.; Ferrer, R.; Jammazi, R. The windowed scalogram difference: A novel wavelet tool for comparing time series. *Appl. Math. Comput.* **2017**, *312*, 49–65. [[CrossRef](#)]
40. Cao, L.; Luo, Y.; Qiu, S.; Liu, J. A perturbation method to the tent map based on Lyapunov exponent and its application. *Chin. Phys. B* **2015**, *24*, 100501. [[CrossRef](#)]

41. Krishnamoorthi, S.; Jayapaul, P.; Dhanaraj, R.K.; Rajasekar, V.; Balusamy, B.; Islam, S.H. Design of pseudo-random number generator from turbulence padded chaotic map. *Nonlinear Dyn.* **2021**, *104*, 1627–1643. [[CrossRef](#)]
42. Alhadawi, H.S.; Zolkipli, M.F.; Ismail, S.M.; Lambić, D. Designing a pseudorandom bit generator based on LFSRs and a discrete chaotic map. *Cryptologia* **2019**, *43*, 190–211. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.