Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 121359, 12 pages
http://dx.doi.org/10.1155/2015/121359

Research Article

A Generalized Stability Theorem for Discrete-Time Nonautonomous Chaos System with Applications

Mei Zhang, Danling Wang, Lequan Min, and Xue Wang

School of Mathematics and Physics, University of Science and Technology Beijing, Beijing 100083, China

Correspondence should be addressed to Lequan Min; minlequan@sina.com

Received 17 April 2015; Accepted 7 July 2015

Academic Editor: Ricardo Aguilar-López

Firstly, this study introduces a definition of generalized stability (GST) in discrete-time nonautonomous chaos system (DNCS), which is an extension for chaos generalized synchronization. Secondly, a constructive theorem of DNCS has been proposed. As an example, a GST DNCS is constructed based on a novel 4-dimensional discrete chaotic map. Numerical simulations show that the dynamic behaviors of this map have chaotic attractor characteristics. As one application, we design a chaotic pseudorandom number generator (CPRNG) based on the GST DNCS. We use the SP800-22 test suite to test the randomness of four 100-key streams consisting of 1,000,000 bits generated by the CPRNG, the RC4 algorithm, the ZUC algorithm, and a 6-dimensional CGS-based CPRNG, respectively. The numerical results show that the randomness performances of the two CPRNGs are promising. In addition, theoretically the key space of the CPRNG is larger than 2^{1116}. As another application, this study designs a stream avalanche encryption scheme (SAES) in RGB image encryption. The results show that the GST DNCS is able to generate the avalanche effects which are similar to those generated via ideal CPRNGs.

1. Introduction

Chaos, characterized by its deterministic, unpredictable features and extremely sensitive dependence on initial conditions, stems from nonlinear systems (e.g., see [1–3]). During the last three decades, chaos theory has been developed to model complex nature and social phenomena by using quite simple mathematical models. Thus it has captured much attention of the scientific community for predicting the behavior of systems in the real world.

Nonautonomous discrete systems were introduced in [4]; as we can see, they also appear connected to some nonautonomous difference equations (e.g., see [5, 6]). A lot of natural questions concerned the nonautonomous dynamical systems; therefore the research of the nonautonomous system is recently very intensive (e.g., see [7–9]). However, there is no much research on discrete nonautonomous chaotic systems.

Mathematically chaos synchronization (CS) means that the trajectories of two different chaotic systems exhibit identical phenomena with time evolution. Synchronization phenomenon is a kind of typical collective behaviors that could be found in many physical, biological, and engineering systems (e.g., see [10–16]).

Chaos generalized synchronization (CGS) means that, with time evolution, the trajectories of two different chaotic systems tend to become identical with respect to a transformation in a specific domain. Therefore CGS has more general meaning than CS. The study of CGS has also attracted much attention (e.g., see [17–25]).

Generally speaking, there are different methods such that two systems achieve generalized synchronization such as design control laws to force coupled systems to satisfy a prescribed functional relation [26–28]. In series papers [29–32], we have studied the general representations of two systems to achieve GS.

Since the pioneer work of Pecora and Carroll on CS secure communication [33], CS and CGS have been used as new tools in secure communications and have been used as designs of pseudorandom number generators (e.g., [17–25, 34–37]). Research along this line is promising.

This paper extends the concept of the generalized synchronization [38] to generalized stability (GST) for discrete-time nonautonomous chaos system (DNCS), and then we propose a corresponding GST theorem. Using the GST theorem helps design a novel nonautonomous chaotic discrete

system and construct a chaos-based pseudorandom number generator (CPRNG). We test the randomness of the CPRNG, the RC4 algorithm, the ZUC algorithm [39], and a 6-dimensional CGS-based CPRNG2 [40] by the SP800-22 test suite of the INST [41], respectively. At last, as an application, by using the CPRNG and the SAES a RGB image has been encrypted in communication.

The rest of this paper is organized as follows. Section 2 introduces the definition and the theorem of GST. Section 3 presents a novel 4-dimensional nonautonomous chaotic discrete system and an 8-dimensional GST system and simulates the dynamic behaviors of the GST system. Section 4 designs the CPRNG and makes the statistic tests for the CPRNG, the RC4 algorithm, the ZUC algorithm, and the 6-dimensional CGS-based CPRNG2, respectively. An image encryption example of the CPRNG and the SAES is introduced in Section 5. Finally, Section 6 presents some concluding remarks.

2. Definition and Theorem of GST

A point of view states that two events with relationship of cause and effect might be described via CGS for two systems. Motivated by CGS, let us introduce the concept of GST.

Definition 1. Consider two systems

$$\mathbf{X}(k+1) = F(\mathbf{X}(k), k), \tag{1}$$

$$\mathbf{Y}(k+1) = G(\mathbf{Y}(k), \mathbf{X}(k), k), \tag{2}$$

where

$$\begin{aligned}
&\mathbf{X}(k) = (x_1(k), \ldots, x_n(k))^T,\\
&\mathbf{Y}(k) = (y_1(k), \ldots, y_m(k))^T, \quad m \le n,\\
&F(\mathbf{X}(k), k) = (f_1(\mathbf{X}(k), k), \ldots, f_n(\mathbf{X}(k), k))^T,\\
&G(\mathbf{Y}(k), \mathbf{X}(k), k)\\
&\quad = (g_1(\mathbf{Y}(k), \mathbf{X}(k), k), \ldots, g_m(\mathbf{Y}(k), \mathbf{X}(k), k))^T.
\end{aligned} \tag{3}$$

If there exists a transformation $H : \mathbb{R}^n \times \mathbb{Z}^+ \to \mathbb{R}^m$, where

$$H(\mathbf{X}(k), k) = (h_1(\mathbf{X}(k), k), \ldots, h_m(\mathbf{X}(k), k))^T, \tag{4}$$

for $\forall \epsilon > 0$, there exist $\delta_1 > 0$ and $\delta_2 > 0$ such that all trajectories of (1) and (2) with initial conditions $(\mathbf{X}(0), \mathbf{Y}(0)) \in B(\mathbf{X}_0, \delta_1) \times B(\mathbf{Y}_0, \delta_2) \subset \mathbb{R}^n \times \mathbb{R}^m$ satisfy

$$\|H(\mathbf{X}(k), k) - \mathbf{Y}(k)\| < \epsilon, \quad k = 1, 2, \ldots. \tag{5}$$

Then the systems in (1) and (2) are said to be in GST with respect to the transformation $H(\mathbf{X}(k), k)$. System (1) is called the driving system; system (2) is said to be the driven system.

In order to construct the novel DNCS with the GST property, we present the following.

Theorem 2. *Let $\mathbf{X}, \mathbf{Y}, \mathbf{X}_m, F(\mathbf{X}, k)$, and $G(\mathbf{Y}, \mathbf{X}, k)$ be defined by (3); $\mathbf{X}_m = (x_1(k), \ldots, x_m(k))^T$. Suppose that*

$$H(x_1, x_2, \ldots, x_n, k) = (y_1, y_2, \ldots, y_m)^T. \tag{6}$$

If two systems (1) and (2) are in GST via the transformation $H(\mathbf{X}, k)$, if and only if, the function $G(\mathbf{Y}, \mathbf{X}, k)$ given in (2) has the following form:

$$G(\mathbf{Y}, \mathbf{X}, k) = H(F(\mathbf{X}, k), k+1) - q(\mathbf{X}_m, \mathbf{Y}, k), \tag{7}$$

and the function

$$\begin{aligned}
&q(\mathbf{X}_m, \mathbf{Y}, k)\\
&\quad = (q_1(\mathbf{X}_m, \mathbf{Y}, k), q_2(\mathbf{X}_m, \mathbf{Y}, k), \ldots, q_m(\mathbf{X}_m, \mathbf{Y}, k))^T
\end{aligned} \tag{8}$$

guarantees that the zero solution of the following error equation is stable on the open set $B(\mathbf{X}_0, \delta_1) \times B(\mathbf{Y}_0, \delta_2)$ defined by Definition 1:

$$\begin{aligned}
\mathbf{e}(k+1) &= H(\mathbf{X}(k+1), k+1) - \mathbf{Y}(k+1)\\
&= q(\mathbf{X}_m, \mathbf{Y}, k).
\end{aligned} \tag{9}$$

Proof. Denote

$$G(\mathbf{Y}, \mathbf{X}, k) - H(F(\mathbf{X}, k), k+1) = -q(\mathbf{X}_m, \mathbf{Y}, k). \tag{10}$$

Then

$$\begin{aligned}
\mathbf{e}(k+1) &= H(\mathbf{X}(k+1), k+1) - \mathbf{Y}(k+1)\\
&= q(\mathbf{X}_m, \mathbf{Y}, k).
\end{aligned} \tag{11}$$

Therefore, two dynamic systems (1) and (2) are in GST via the transformation H, if and only if the function $q(\mathbf{X}_m, \mathbf{Y}, k)$ makes the zero solution of the error equation (9) stable. This completes the proof. □

3. A Novel GST DNCS

The construction of the novel GST DNCS is divided into 3 steps: (1) construct a chaotic system $\mathbf{X}(k+1) = F(\mathbf{X}(k), k)$; (2) introduce a transformation H; (3) construct a system $\mathbf{Y}(k+1) = H(F(\mathbf{X}, k), k+1) - q(\mathbf{X}_m, \mathbf{Y}, k)$ such that $q(\mathbf{X}_m, \mathbf{Y}, k)$ guarantees the zero solution of the error equation (9) stable.

Step 1. Construct a novel 4-dimensional nonautonomous chaotic system as follows:

$$\begin{aligned}
x_1(k+1) &= 1.675 x_1(k) \cos(x_1(k)) \sin(x_2(k))\\
&\quad - \sin(x_1(k)) \cos(x_3(k))\\
&\quad + \cos(x_2(k)) \sin(x_3(k)),\\
x_2(k+1) &= 5.125 \sin(x_2(k))\\
&\quad - 2 \cos(x_3(k)) \sin(x_1(k))\\
&\quad + \frac{k}{k+1} \sin(x_3(k)),\\
x_3(k+1) &= \sin(x_1(k) + x_2(k)) + \sin(x_3(k)),\\
x_4(k+1) &= \cos(x_1(k) + x_4(k)).
\end{aligned} \tag{12}$$

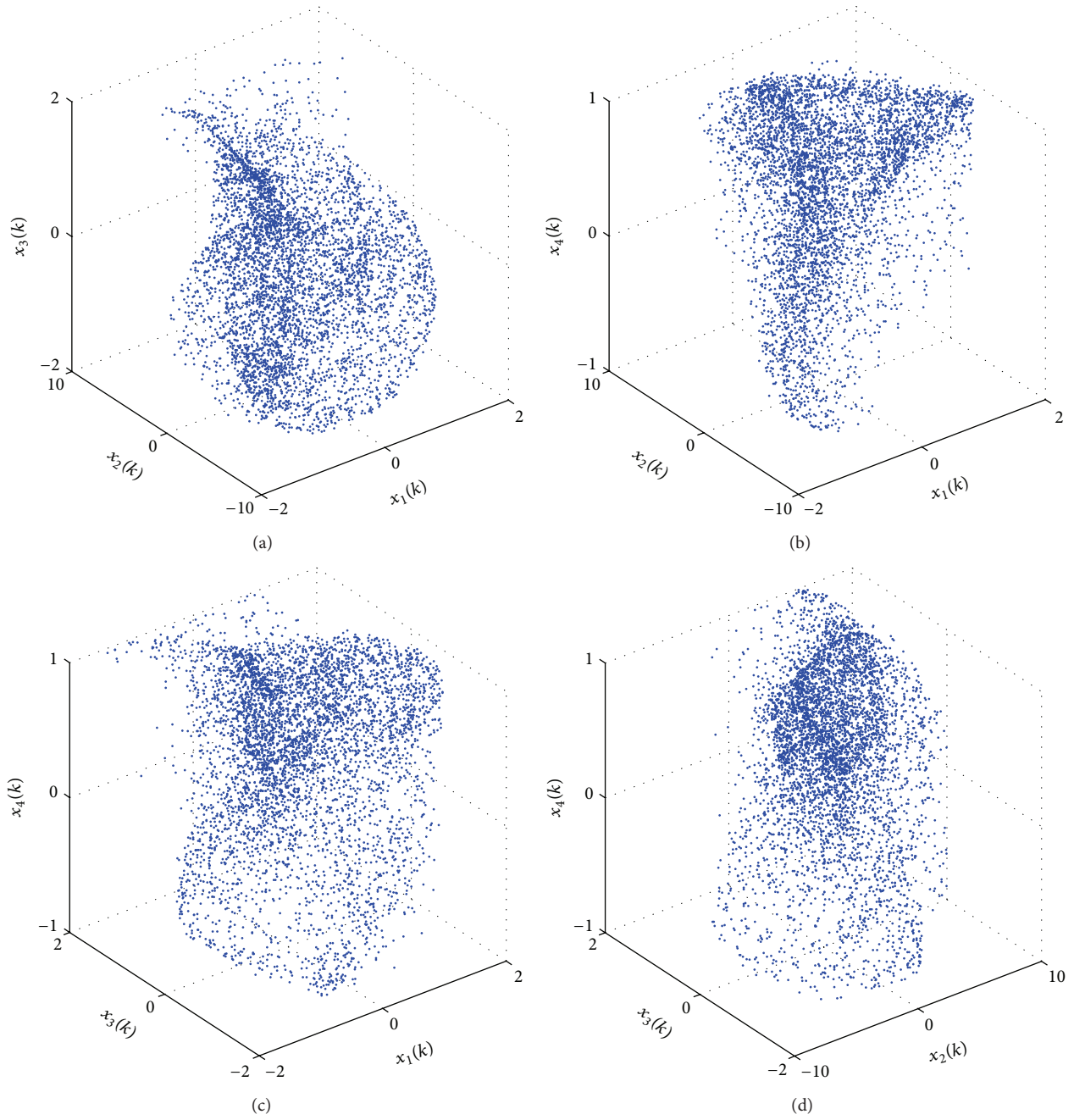

FIGURE 1: Chaotic trajectories of variables: (a) $x_1(k)$–$x_2(k)$–$x_3(k)$, (b) $x_1(k)$–$x_2(k)$–$x_4(k)$, (c) $x_1(k)$–$x_3(k)$–$x_4(k)$, and (d) $x_2(k)$–$x_3(k)$–$x_4(k)$.

The calculated Lyapunov exponents of system (12) are $\{0.9608, 0.0433, -0.5614, -0.7245\}$; therefore, system (12) is a chaotic system.

Now, select the following initial condition:

$$\mathbf{X}(0) = (0.32, 0.2, 0.1, 0.12)^T. \tag{13}$$

The chaotic orbits of the state variables x_1, x_2, x_3, x_4 for the first 5000 iterations are shown in Figure 1. The evolution of state variables, k–$x_1(k)$, k–$x_2(k)$, k–$x_3(k)$, and k–$x_4(k)$, is shown in Figure 2. Observe that the dynamic behaviors of the chaotic map demonstrate chaotic attractor characteristics.

Step 2. Let

$$A = \begin{pmatrix} -1 & 2 & -5 & 1 \\ 2 & -5 & -4 & 2 \\ -4 & -2 & 4 & 3 \\ 1 & 2 & 3 & -2 \end{pmatrix}, \tag{14}$$

which is an invertible matrix. Define a transformation $H : \mathbb{R}^4 \times \mathbb{Z}^+ \to \mathbb{R}^4$ as follows:

$$H(\mathbf{X}, k) = A\mathbf{X} + \phi(k), \tag{15}$$

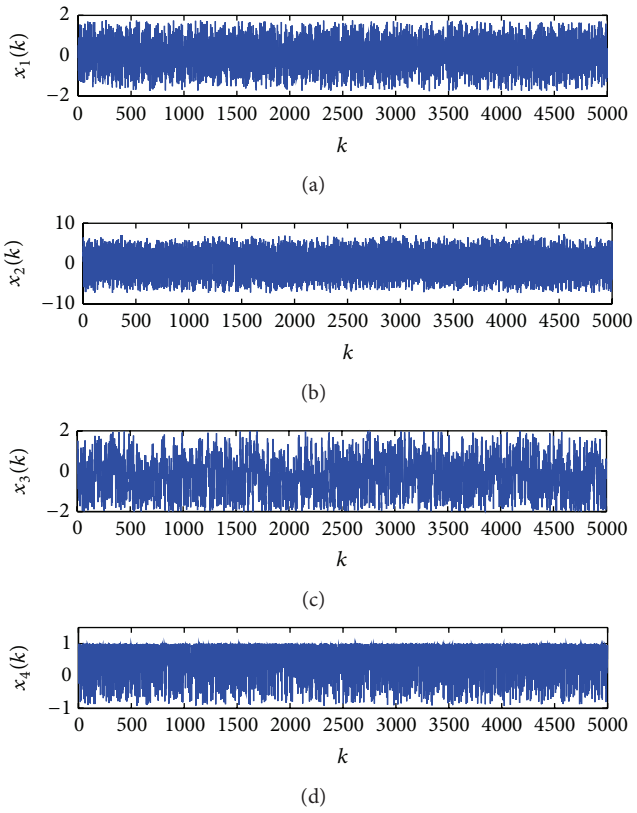

FIGURE 2: The evolution of state variables: (a) $k-x_1(k)$, (b) $k-x_2(k)$, (c) $k-x_3(k)$, and (d) $k-x_4(k)$.

where

$$\phi(k) = \begin{pmatrix} \arctan(2\pi k) \\ \arctan(2\pi k) \\ \arctan(2\pi k) \\ \arctan(2\pi k) \end{pmatrix}. \tag{16}$$

Step 3. First, we need the following lemma.

Lemma 3 (see [42]). *If there exists a positive definite function $v(\mathbf{e}, k)$ such that $\Delta v_{(2,1)}(\mathbf{e}, k)$ is negative semidefinite, then the system $\mathbf{e}(k+1) = f(\mathbf{e}(k), k)$ is zero solution stable.*

Then, we construct system (2). Let

$$\begin{aligned} \mathbf{Y}(k+1) &= G(\mathbf{Y}(k), \mathbf{X}(k), k) \\ &= H(F(\mathbf{X}, k), k+1) - q(\mathbf{X}_m, \mathbf{Y}, k) \\ &= H(\mathbf{X}(k+1), k+1) - q(\mathbf{X}_m, \mathbf{Y}, k), \end{aligned} \tag{17}$$

where

$$\begin{aligned} &q(\mathbf{X}_m, \mathbf{Y}, k) \\ &= \left(\frac{e_2(k)}{1+e_1(k)^2}, \frac{e_1(k)}{1+e_2(k)^2}, \frac{e_3(k)}{5}, \frac{e_4(k)}{5} \right)^T, \end{aligned} \tag{18}$$

$$\begin{aligned} \mathbf{e}(k) &= (e_1(k), e_2(k), e_3(k), e_4(k))^T \\ &= H(\mathbf{X}(k), k) - \mathbf{Y}(k) = q(\mathbf{X}_m, \mathbf{Y}, k). \end{aligned} \tag{19}$$

Now we prove that $q(\mathbf{X}_m, \mathbf{Y}, k)$ guarantees that error equation (19) will be zero solution stable.

In fact, define a positive definite function

$$v(\mathbf{e}, k) = e_1(k)^2 + e_2(k)^2 + e_3(k)^2 + e_4(k)^2, \tag{20}$$

and then

$$\begin{aligned} \Delta v_{(2,1)}(\mathbf{e}, k) &= v(\mathbf{e}(k+1), k+1) - v(\mathbf{e}(k), k) \\ &= e_1(k+1)^2 + e_2(k+1)^2 + e_3(k+1)^2 \\ &\quad + e_4(k+1)^2 \\ &\quad - \left(e_1(k)^2 + e_2(k)^2 + e_3(k)^2 + e_4(k)^2 \right) \\ &= \left(\frac{e_2(k)}{1+e_1(k)^2} \right)^2 + \left(\frac{e_1(k)}{1+e_2(k)^2} \right)^2 \\ &\quad + \left(\frac{e_3(k)}{5} \right)^2 + \left(\frac{e_4(k)}{5} \right)^2 \\ &\quad - \left(e_1(k)^2 + e_2(k)^2 + e_3(k)^2 + e_4(k)^2 \right) \\ &= \left(\frac{1}{\left(1+e_2(k)^2\right)^2} - 1 \right) e_1(k)^2 \\ &\quad + \left(\frac{1}{\left(1+e_1(k)^2\right)^2} - 1 \right) e_2(k)^2 \\ &\quad - \frac{24e_3(k)^2}{25} - \frac{24e_4(k)^2}{25} \le 0. \end{aligned} \tag{21}$$

Therefore, $\Delta v_{(2,1)}(\mathbf{e}, k)$ is a negative semidefinite function. According to Lemma 3, (19) is zero solution stable.

Hence we conclude, from Theorem 2, that the constructed dynamic systems (12) and (17) are in GST with respect to transformation (15) for any initial value $(\mathbf{X}(0), \mathbf{Y}(0)) \in B(\mathbf{X}_0, \delta_1) \times B(\mathbf{Y}_0, \delta_2) \subset \mathbb{R}^4 \times \mathbb{R}^4$.

Now we choose (22) as the initial condition:

$$\mathbf{Y}(0) = A\mathbf{X}(0). \tag{22}$$

The first 5000 iterations of variable components of $\mathbf{Y}$ are obtained as shown in Figure 3. The evolution of state variables, $k-y_1(k)$, $k-y_2(k)$, $k-y_3(k)$, and $k-y_4(k)$, is shown in Figure 4. The simulation result shows that the system has chaotic attractor characteristics. And Figure 5 shows that $\mathbf{X}$ and $\mathbf{Y}$ are rapidly in GST with respect to transformation $H(\mathbf{X}, k) = A\mathbf{X} + \phi(k)$.

In summary, both theory and numerical simulation shows that the constructed dynamic systems (12) and (17) are in GST.

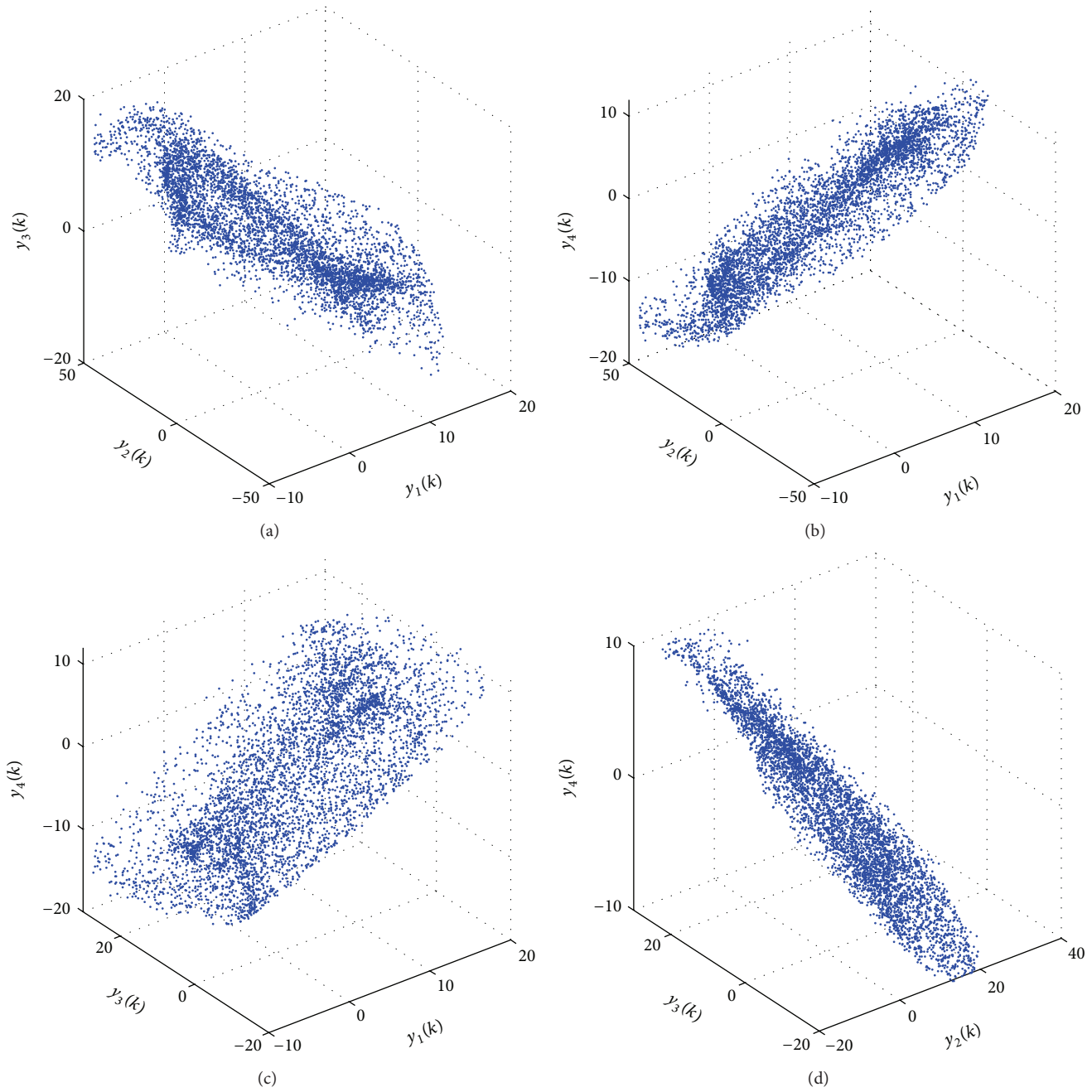

FIGURE 3: Chaotic trajectories of variables: (a) $y_1(k)-y_2(k)-y_3(k)$, (b) $y_1(k)-y_2(k)-y_4(k)$, (c) $y_1(k)-y_3(k)-y_4(k)$, and (d) $y_2(k)-y_3(k)-y_4(k)$.

4. Chaotic Pseudorandom Number Generator and Test

4.1. Pseudorandom Number Generator. Denote

$$\begin{aligned} \mathbf{X}_i &= \{x_i(k) \mid k = 1, 2, \ldots, N\}, \\ \mathbf{Y}_i &= \{y_i(k) \mid k = 1, 2, \ldots, N\}, \end{aligned} \tag{23}$$

where x_i's and y_i's are defined by (12) and (17). First introduce a transformation $T_1 : \mathbb{R} \to \{0, 1, \ldots, 2^{16} - 1\}$, which transforms the chaotic streams of GST systems (23) into keystreams. Denote

$$\mathbf{S} = \mathbf{X}_1 + \mathbf{Y}_2, \tag{24}$$

and then T_1 is defined by

$$T_1(\mathbf{S}) = \mathrm{mod}\left(\mathrm{round}\left(\frac{L(\mathbf{S} - \min(\mathbf{S}))}{(\max(\mathbf{S}) - \min(\mathbf{S}))}\right), 2^{16}\right), \tag{25}$$

where $L = 10^{15}$.

Now we can design a CPRNG based on transformation (24) and (25) and the GST systems (12) and (17). The seeds

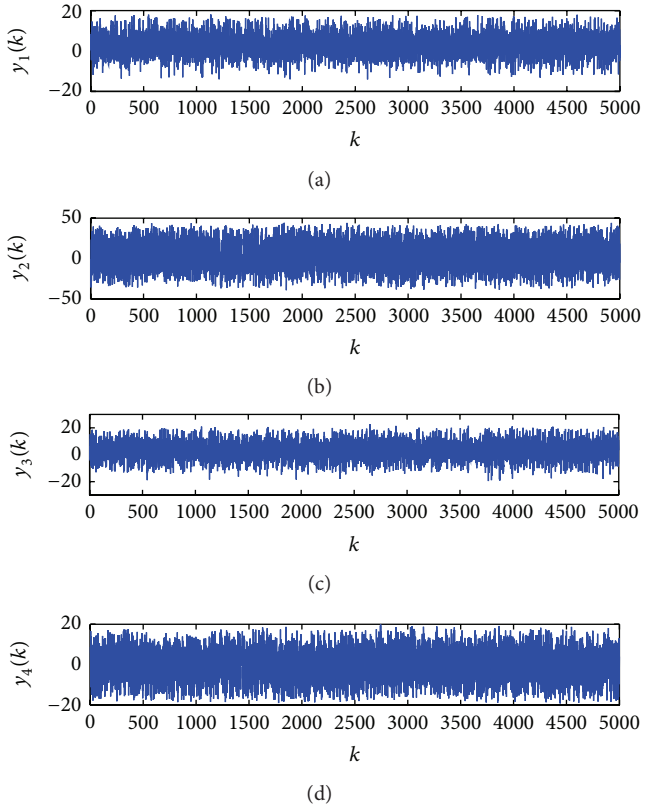

FIGURE 4: The evolution of state variables: (a) $k - y_1(k)$, (b) $k - y_2(k)$, (c) $k - y_3(k)$, and (d) $k - y_4(k)$.

of the CPRNG are the initial conditions of the GST systems, which can be chosen via random number generators. Therefore the output keystreams of the CPRNG can be obtained via (25) and the GST systems (12) and (17).

4.2. Randomness Test. The NIST SP800-22 test suite [41] consists of 15 statistical tests (see the first column in Table 2) that were developed to test the randomness of binary sequences produced by either hardware or software based cryptographic random or pseudorandom number generators [41]. Each statistical test is formulated to test a specific *null hypothesis* (H_0): the sequence being tested is random. A significance level (α) can be chosen for the tests. If P value $\geq \alpha$, then the null hypothesis is accepted; that is, the sequence appears to be random. Typically, α is chosen in the range $[0.001, 0.01]$. NIST SP800-22 test suite is stricter than the FIPS140-2 test suite NIST [43]. A binary sequence which has passed all tests of FIPS140-2 test suite may not pass all tests in the NIST SP800-22 test suite.

In order to test the pseudorandomness of the CPRNG, we transform the "16-bit" stream defined by (25) to the $\{0, 1\}$ bit stream as follows.

Construct a transform $T_2 : \{0, 1, \ldots, 2^{16} - 1\} \to \{0, 1\}$, which is defined by

$$T_2 = T_{22} \circ T_{21} \tag{26}$$

```
N=20000;
K=randi([0 254],1,255);
S=[0:255-1];j=0;
for i=1:255
       j=mod(j+S(i)+K(i),255);
            Sk=S(j+1);
               S(j+1)=S(i);
                    S(i)=Sk;
end
      C=zeros(1,N);j=0;i=0;k=1;
for l=1:N/8
        i=mod(i+1,255);
           j=mod(j+S(i+1),255);
               Sk=S(j+1);
                  S(j+1)=S(i+1);
                     S(i+1)=Sk;
 C(l)=S(mod(S(j+1)+S(i+1),255)+1);
end
 C=(dec2bin(C))';
     C=C(:);
        C=bin2dec(C);
```

ALGORITHM 1

s.t., for all $\mathbf{y} \in \{0, 1, \ldots, 2^{16} - 1\}^N$,

$$T_{21}(\mathbf{y}) = \text{dec2bin}(\mathbf{y}). \tag{27}$$

Let $\mathbf{z} = \text{dec2bin}(\mathbf{Y})$; then

$$T_{22}(\mathbf{z}) = \mathbf{z}(:), \tag{28}$$

where dec2bin and $\mathbf{z}(:)$ are both Matlab commands. And the transformation $T : \mathbb{R}^4 \to \{0, 1\}$ is defined via

$$T = T_2 \circ T_1. \tag{29}$$

Now the SP800-22 test is used to check 100 keystreams randomly generated, respectively, by CPRNG with perturbed randomly initial condition $\mathbf{X}(0)$, $\mathbf{Y}(0)$, and the parameters of matrix (14) in the range $|\epsilon| \in [10^{-16}, 10^{-2}]$. The results are listed in the 2nd column of Tables 1 and 2.

The ZUC algorithm, a stream cipher designed by Chinese cryptologists, is accepted by the 3GPP LTE as the international encryption standard for the 4G mobile communication. Now, the SP800-22 test is used to test the 100 keystreams randomly generated by the ZUC algorithm program (see Appendix A in [39]). The statistical test results are shown in the 4th column of Tables 1 and 2.

In paper [40], a nonautonomous discrete CGS system based CPRNG has been proposed. The SP800-22 statistic test results of the CPRNG2 are listed in the 5th column of Tables 1 and 2 (also see Tables I and II in [40]).

The well-known RC4 was designed by Rivest of the RSA Security in 1987, which has been widely used in popular protocols such as Secure Sockets. The RC4 algorithm as PRNG can be designed via the Matlab commands as shown in Algorithm 1.

Here, "randi([0 254], 1, 255)" generates a vector of uniformly distributed random integers $\{0, 1, \ldots, 254\}$ of dimension 255; "mod" means taking modulus after division;

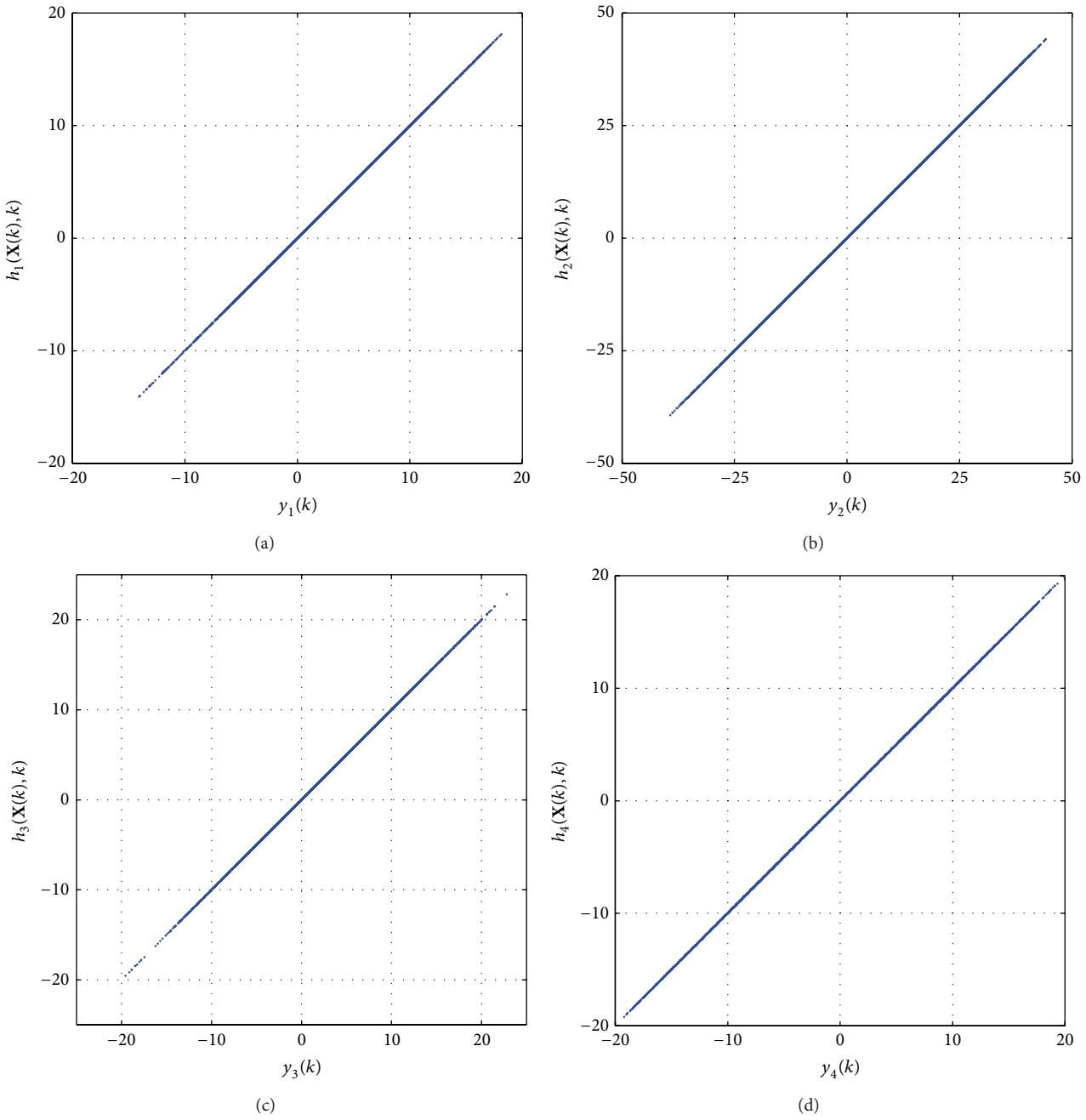

FIGURE 5: The state variables $\mathbf{X}$ and $\mathbf{Y}$ are in GST: (a) $y_1(k) - h_1(\mathbf{X}(k),k)$, (b) $y_2(k) - h_2(\mathbf{X}(k),k)$, (c) $y_3(k) - h_3(\mathbf{X}(k),k)$, and (d) $y_4(k) - h_4(\mathbf{X}(k),k)$.

"zeros$(1,N)$" is a zero row vector of dimension N. Consequently, the RC4 algorithm-based PRNG is designed. Then, the SP800-22 test suite is used to test 100 keystreams randomly generated by the RC4 PRNG. The statistical test results are listed in the 3rd column of Tables 1 and 2.

Now, we compare all test results shown in Tables 1 and 2. It can be clearly observed, from the statistical properties of the pseudorandomness of the sequences generated via the four PRNGs, that all the CPRNGs have satisfactory, indeed very promising, randomness properties.

4.3. Key Space. The key set parameters of CPRNG includes the initial condition $\mathbf{X}(0)$, $\mathbf{Y}(0)$, and the matrix $A = (\alpha_{i,j})$. It can be proved that if the perturbation matrix $\Delta = (\delta_{i,j})$ satisfies $|\delta_{i,j}| < 0.07419$, the matrix $A + \Delta$ is still invertible. Therefore the CPRNGs have 4 + 4 + 16 key parameters denoted by

$$\mathbf{K}_s = \{k_1, k_2, \ldots, k_{24}\}. \tag{30}$$

Let the key set be perturbed by

$$\mathbf{K}_s(\Delta) = \mathbf{K}_s + [\delta_1, \delta_2, \ldots, \delta_{24}], \tag{31}$$

Table 1: The calculated mean p-values of SP800-22 statistical tests for the 100 binary sequences with length 10^6 produced by the CPRNG proposed in this paper, the RC4 PRNG, the ZUC algorithm and the CPRNG2, respectively. Select a significance level $\alpha = 0.01$.

Statistical Test	Mean p-value			
	CPRNG	RC4	ZUC	CPRNG2
(1) Frequency	0.5254	0.49598	0.46669	0.51593
(2) Block Frequency	0.4764	0.47781	0.48780	0.47206
(3) Runs	0.5667	0.46958	0.45937	0.50736
(4) Long Runs of Ones	0.5144	0.53504	0.45351	0.55381
(5) Binary Matrix Rank	0.5020	0.50302	0.47611	0.49911
(6) Spectral DFT	0.5219	0.47094	0.50207	0.49296
(7) Non-overlapping Template	0.5037	0.49385	0.50045	0.50023
(8) The Overlapping Template	0.4812	0.50478	0.46822	0.49787
(9) Maurer s Universal Test	0.5084	0.4878	0.45006	0.50524
(10) Linear Complexity	0.5150	0.51639	0.46828	0.52894
(11) Serial ($m = 5, \nabla\Psi_m^2$)	0.4721	0.47546	0.4837	0.50094
Serial ($m = 5, \nabla^2\Psi_m$)	0.4726	0.48377	0.50556	0.55179
(12) Approximate Entropy	0.5158	0.48344	0.45022	0.53289
(13) Cumulative Sums +1	0.5133	0.45873	0.46031	0.47830
Cumulative Sums −1	0.5040	0.47298	0.47543	0.50262
(14) Random Excursions	0.3481	0.31615	0.29159	0.35605
(15) Random Excursions Variant	0.3367	0.30332	0.29350	0.34412

Table 2: The acceptance rates of SP800-22 statistical tests for the 100 binary sequences with length 10^6 produced by the CPRNG proposed in this paper, the RC4 PRNG, the ZUC algorithm, and the CPRNG2, respectively. Here select a significance level $\alpha = 0.01$.

Statistical test	Acceptance rate (%)			
	CPRNG	RC4	ZUC	CPRNG2
(1) Frequency	100	98	100	100
(2) Block frequency	100	98	100	99
(3) Runs	100	98	100	100
(4) Long runs of ones	99	97	99	99
(5) Binary matrix rank	99	97	99	100
(6) Spectral DFT	100	98	99	97
(7) Nonoverlapping template	97–100	94–98	96–100	95–100
(8) The overlapping template	100	97	100	100
(9) Maurer s universal test	99	97	100	100
(10) Linear complexity	99	98	98	99
(11) Serial ($m = 5, \nabla\Psi_m^2$)	98	98	98	98
Serial ($m = 5, \nabla^2\Psi_m$)	97	96	99	100
(12) Approximate entropy	99	98	99	99
(13) Cumulative sums +1	99	98	98	100
Cumulative sums −1	100	98	98	100
(14) Random excursions	65–68	57-58	57-58	66–69
(15) Random excursions variant	66–68	56–58	56–58	68-69

where

$$10^{-16} \le \left|\delta_i\right| \le 10^{-2}, \quad i = 1, \ldots, 24. \tag{32}$$

First, we compare the difference between the keystream $S = T_1(\mathbf{S})$ including 10^6 codes' length generated by key set (30) and keystreams S_p's generated by the perturbed key set (31). SV represents statistic values, DC represents different codes, and CC represents correlation coefficients. The comparison results are shown in third column in Table 3. Observe that the average percentage of different codes is 50.001% and the mean correlation coefficient is 0.00071329. It is very closed to the ideal different value 50% and ideal correlation coefficient 0.

Table 3: The statistic data for the percentages of the codes of the keystream variations between S and S_p's, as well as S and S_m's.

Item	SV	S_p's	S_m's
DC	Min	49.854%	49.886%
	Mean	50.001%	49.995%
	Max	50.082%	50.136%
CC	Min	5.4235×10^{-5}	0
	Mean	0.00071329	2.8295×10^{-7}
	Max	0.0029163	2.8295×10^{-5}

Then, let us compare the same keystream S with the 100 keystreams S_m's generated by the function of Matlab command randi($[0\ \ 1], 1, 10^6$). The comparison results are shown in fourth column in Table 3. Observe that the average percentage of different codes is 49.995% and the mean correlation coefficient is 2.8295×10^{-7}. The results may suggest that the keystream S has no significant correlations with the perturbed keystreams S_p's and the streams S_m's. Furthermore, the key space of the CPRNG is larger than $10^{14\times24} > 2^{1116}$, which will help to increase security.

5. Simulations on Avalanche Image Encryption

This study designs a stream encryption scheme with avalanche effect.

Definition 4 (see [44]). Let $P = \{p_1, p_2, \ldots, p_n\}$ be a binary keystream with d-bit segments generated by a PRNG, $M = \{m_1, m_2, \ldots, m_n\}$ a binary plaintext stream, and $C = \{c_1, c_2, \ldots, c_n\}$ a ciphertext stream. Then the stream encryption scheme with avalanche effect is described as follows:

(1) The ciphertext:

$$C = E(M, P) \tag{33}$$

satisfying

$$c_i = \begin{cases} p_i & \text{if } m_i = 0, \\ \sim p_i & \text{if } m_i = 1, \end{cases} \tag{34}$$

where $\sim p_i$ is defined to be the bit string obtained by replacing all 0s in p_i with 1s and all 1s in p_i with 0s.

(2) The corresponding decrypted plaintext:

$$M = E^{-1}(C, P) \tag{35}$$

satisfying

$$m_i = \begin{cases} 0 & \text{if } c_i = p_i, \\ 1 & \text{if } c_i \neq p_i. \end{cases} \tag{36}$$

Definition 5 (see [44]). A PRNG, S, which generates d-bit keystreams, is called an ideal PRNG, if S has the following properties:

(1) The period of any keystream generated by the PRNG is larger than 2^d. Its seed space and key space are both larger than 2^{512}.

(2) In one period of a pseudorandom keystream generated by the PRNG, the distribution of different d-bit segments in the keystream is homogenous. That is, if the period $p = n \times 2^d$, then the number of each different d-bit segment is equal to n. If the period p is not an integer multiple of 2^d, then the difference between the numbers of different d-bit segments is at most one.

(3) The two keystreams P_1, P_2 generated by any two different seeds have $(2^d - 1)/2^d \times 100\%$ different d-bit segments.

Now, we use the CPRNG to investigate SAES in RGB image Beach with 250×140 pixels as shown in Figure 6(a).

Simulations are implemented and the procedures are described as follows.

(1) Transform the image Beach to a binary plaintext stream

$$M = \{m_1, m_2, \ldots, m_n\}, \tag{37}$$

where $n = 250 \times 140 \times 3 \times 8$.

(2) Use the CPRNG with initial conditions (13) and (22) to generate a keystream

$$P = \{p_1, p_2, \ldots, p_{n+1000}\}. \tag{38}$$

(3) Drop the first 1000 iterative values, use the formula (33) and (34) to encrypt the plaintext stream M, and then obtain a ciphertext

$$C = \{c_1, c_2, \ldots, c_n\}. \tag{39}$$

(4) Use formula (35) and (36) to decrypt the ciphertext; then obtain a decrypted plaintext image

$$\overline{M} = E^{-1}(C, P) \tag{40}$$

without errors (see Figure 6(b)).

(5) Randomly disturb the initial conditions (13), (22) and matrix (14) for 1000 times in the range $|\epsilon| \in [10^{-16}, 10^{-2}]$; then obtain keystreams

$$P_i, \quad i = 1, 2, \ldots, 1000. \tag{41}$$

(6) Use $\{P_1, \ldots, P_{1000}\}$ to decrypt the ciphertext; then obtain the decrypted plaintext

$$\overline{M_i} = E^{-1}(C, P_i), \quad i = 1, \ldots, 1000. \tag{42}$$

(7) Change $\overline{M_i}$ to RGB images.

After changing $\overline{M_i}$ to RGB images, all images become almost pure white images. There are total of 840000 $\{0, 1\}$ codes in each decrypted image. Among the decrypted images, the minimum number of 0s in the decrypted is 3 and the maximum one is 27. Let $I_{i,j}$ denote the jth image having number "i" of 0 codes.

The first five decrypted images with minimum zero codes are shown in Figures 6(c)–6(g). Figure 6(c) is the decrypted image with only three 0 codes. Figures 6(d)–6(g) are the decrypted image with four 0 codes.

The last five decrypted images with maximum zero codes are shown in Figures 6(h)–6(l). Figures 6(h) and 6(i) are the decrypted image with twenty-four 0 codes. Figures 6(j) and 6(k) are the decrypted image with twenty-five 0 codes. Figure 6(l) is the decrypted image with twenty-seven 0 codes.

As can be seen, the percentages of the numbers of "1" codes in the 1000 decrypted images are within the range [99.997%, 99.999%], which are very closed to the ideal value $(2^{16} - 1)/2^{16} \times 100\% = 99.998\%$.

In summary, the simulation shows that the CPRNG and the SAES to encrypt RGB images are able to generate encrypted images with significant avalanche effects.

Some statistic data of the norms between the original keystream S_0 and the keystream $S_{i,j}$ used in the above ten decrypted images are listed in Table 4, respectively. The comparison results show that in the norms between the original image and the corresponding decrypted images there are no significant correlations.

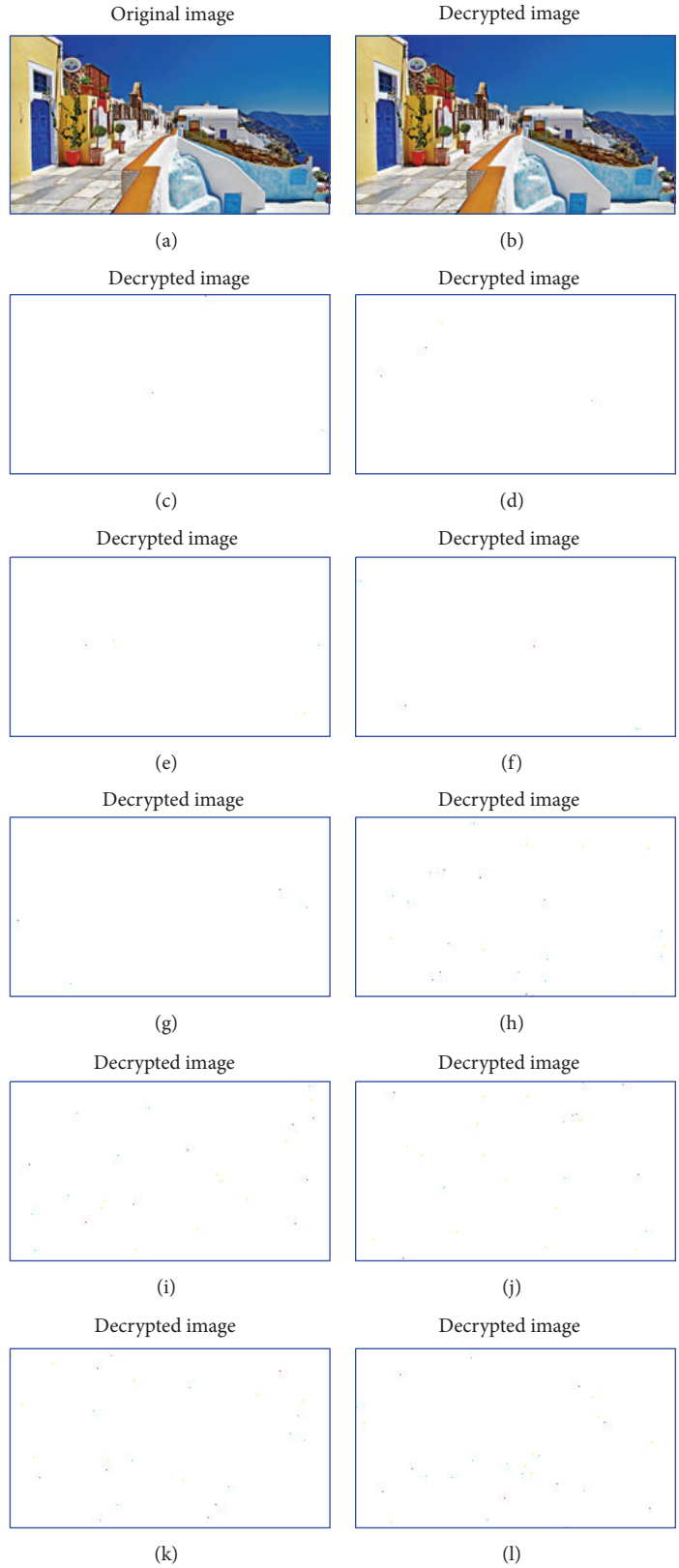

FIGURE 6: (a) Original image beach. (b) Decrypted image without error. Ten decrypted images via keystreams generated with slightly perturbed initial conditions and system parameters in the range $[10^{-16}, 10^{-2}]$: (c) $I_{3,1}$, (d) $I_{4,1}$, (e) $I_{4,2}$, (f) $I_{4,3}$, (g) $I_{4,4}$, (h) $I_{24,1}$, (i) $I_{24,2}$, (j) $I_{25,1}$, (k) $I_{25,2}$, and (l) $I_{27,1}$.

Table 4: Differences between the original keystream S_0 and the keystreams $S_{i,j}$, measured by norm $\|S_0 - S_{i,j}\|$.

	$\|S_0 - S_{i,j}\|$ ($\times 10^{-5}$)									
	$S_{3,1}$	$S_{4,1}$	$S_{4,2}$	$S_{4,3}$	$S_{4,4}$	$S_{24,1}$	$S_{24,2}$	$S_{25,1}$	$S_{25,2}$	$S_{27,1}$
S_0	0.27697	0.27976	0.27644	0.26644	0.29565	0.27747	0.26163	0.27994	0.31517	0.29068

6. Conclusions

The main results of this paper are concluded as follows.

(1) This study first introduces the definition of GST in DNCS, which is an extension for chaos generalized synchronization in DNCS, and then proposes a constructive GST theorem, which provides a general representation for GST DNCS.

(2) As an example, a novel 4-dimensional nonautonomous discrete chaotic map and an 8-dimensional GST system are constructed, whose trajectories display chaotic attractor characteristics.

(3) This 8-dimensional GST system is applied as a chaotic pseudorandom number generator (CPRNG). The SP800-22 tests show that the randomness of the CPRNG is similar to RC4 PRNG, ZUC PRNG, and CPRNG2 [40].

(4) The key space of the CPRNG is larger than 2^{1116}, which is large enough against brute-force attacks.

(5) An image encryption example on the CPRNG with the SAES is given. It shows that the CPRNG is able to generate significant avalanche effects. The numerical simulations also show that the CPRNG is a qualified candidate for SAES.

Furthermore, to prevent attacks, one can consider a "one-time-pad" scheme: let X be a set in the seed space (initial conditions) of the CPRNG, and assume that Alice and Bob share a one-to-one map $f : X \to X$. Before each communication, Alice randomly selects an element $x \in X$ and sends it to Bob. Then, they both use $f(x)$ as the seed for one-time encryption.

In summary, based on generalized synchronization methods, GST and corresponding theorem are introduced. They have been successfully used to design CPRNG with sound randomness for practical applications. It is expected that the GST theory deserves further detailed investigations on its theoretical analyses and numerical simulations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and thank Longjie Hao for computer program assistance. This project is supported by the National Natural Science Foundation of China (Grant nos. 61074192 and 61170037) and Fundamental Research Funds for the Central Universities no. FRF-TP-14-068A2.

References

[1] G. Chen and X. Dong, *From Chaos to Order: Methodologies, Perspectives, and Applications*, 1998.

[2] E. Ott, *Chaos in Dynamical Systems*, Cambridge University Press, Cambridge, UK, 2002.

[3] J. C. Sprott, *Chaos and Time-Series Analysis*, Oxford University Press, Oxford, UK, 2003.

[4] S. Kolyada and L. Snoha, "Topological entropy of nonautonomous dynamical systems," *Random & Computational Dynamics*, vol. 4, no. 2-3, pp. 205–233, 1996.

[5] S. Eleydi, "Nonautonomous difference equations: open problems and conjectures," *Fields Institute Communications*, no. 42, pp. 423–428, 2004.

[6] S. Elaydi and R. J. Sacker, "Nonautonomous Beverton-Holt equations and the Cushing-Henson conjectures," *Journal of Difference Equations and Applications*, vol. 11, no. 4-5, pp. 337–346, 2005.

[7] G. Froyland, S. Lloyd, and N. Santitissadeekorn, "Coherent sets for nonautonomous dynamical systems," *Physica D*, vol. 239, no. 16, pp. 1527–1541, 2010.

[8] Y. Zhang, S. Gao, and Y. Liu, "Analysis of a nonautonomous model for migratory birds with saturation incidence rate," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 4, pp. 1659–1672, 2012.

[9] S. Balasuriya and B. J. Binder, "Nonautonomous analysis of steady Korteweg-de Vries waves under nonlocalised forcing," *Physica D: Nonlinear Phenomena*, vol. 285, pp. 28–41, 2014.

[10] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physical Review Letters*, vol. 64, no. 8, pp. 821–825, 1990.

[11] J. Daafouz and G. Millerioux, "Poly-quadratic stability and global chaos synchronization of discrete time hybrid systems," *Mathematics and Computers in Simulation*, vol. 58, no. 4–6, pp. 295–307, 2002.

[12] Z.-M. Ge and J.-K. Lee, "Chaos synchronization and parameter identification for gyroscope system," *Applied Mathematics and Computation*, vol. 163, no. 2, pp. 667–682, 2005.

[13] J. Hu, J. Ma, and J. Lin, "Chaos synchronization and communication of mutual coupling lasers ring based on incoherent injection," *Optik*, vol. 121, no. 24, pp. 2227–2229, 2010.

[14] X. Zhao, Z. Li, and S. Li, "Synchronization of a chaotic finance system," *Applied Mathematics and Computation*, vol. 217, no. 13, pp. 6031–6039, 2011.

[15] J. Yang and F. Zhu, "Synchronization for chaotic systems and chaos-based secure communications via both reduced-order and step-by-step sliding mode observers," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 4, pp. 926–937, 2013.

[16] M. Faieghi, S. K. M. Mashhadi, and D. Baleanu, "Sampled-data nonlinear observer design for chaos synchronization: a Lyapunov-based approach," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 7, pp. 2444–2453, 2014.

[17] Y. Zou and J. Zhu, "Control linear and nonlinear generalized synchronization of chaotic systems," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 1397–1402, Guangzhou, China, August 2005.

[18] F. A. Breve, L. Zhao, M. G. Quiles, and E. E. N. Macau, "Chaotic phase synchronization and desynchronization in an oscillator network for object selection," *Neural Networks*, vol. 22, no. 5-6, pp. 728–737, 2009.

[19] G. Zhang, Z. Liu, and Z. Ma, "Generalized synchronization of different dimensional chaotic dynamical systems," *Chaos, Solitons & Fractals*, vol. 32, no. 2, pp. 773–779, 2007.

[20] J. Hu, Y. Han, and L. Zhao, "Synchronizing chaotic systems using control based on a special matrix structure and extending to fractional chaotic systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 1, pp. 115–123, 2010.

[21] A. Margheri and R. Martins, "Generalized synchronization in linearly coupled time periodic systems," *Journal of Differential Equations*, vol. 249, no. 12, pp. 3215–3232, 2010.

[22] Z. Yuan, Z. Xu, and L. Guo, "Generalized synchronization of two unidirectionally coupled discrete stochastic dynamical systems," *Chinese Physics B*, vol. 20, no. 7, Article ID 070503, 2011.

[23] A. A. Koronovskii, O. I. Moskalenko, and A. E. Hramov, "Generalized synchronization in complex networks," *Technical Physics Letters*, vol. 38, no. 10, pp. 924–927, 2012.

[24] S. K. Agrawal and S. Das, "Function projective synchronization between four dimensional chaotic systems with uncertain parameters using modified adaptive control method," *Journal of Process Control*, vol. 24, no. 5, pp. 517–530, 2014.

[25] X. P. Yang, L. Q. Min, and X. Wang, "A cubic map chaos criterion theorem with applications in generalized synchronization based pseudorandom number generator and image encryption," *Chaos*, vol. 25, no. 5, Article ID 053104, 2015.

[26] G. Peng, Y. Jiang, and F. Chen, "Generalized projective synchronization of fractional order chaotic systems," *Physica A*, vol. 387, no. 14, pp. 3738–3746, 2008.

[27] X. Wu, W. X. Zheng, and J. Zhou, "Generalized outer synchronization between complex dynamical networks," *Chaos*, vol. 19, no. 1, Article ID 013109, 9 pages, 2009.

[28] C. L. Li, J. B. Xiong, and W. Li, "A new hyperchaotic system and its generalized synchronization," *Optik*, no. 125, pp. 575–579, 2014.

[29] H. Zang, L. Min, and G. Zhao, "A generalized synchronization theorem for discrete-time chaos system with application in data encryption scheme," in *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS '07)*, pp. 1325–1329, IEEE, Kokura, Japan, July 2007.

[30] T. Liu, Y. Ji, L. Min, G. Zhao, and X. Qin, "Generalized synchronization theorem for non-autonomous differential equation with application in encryption scheme," in *Proceedings of the International Conference on Computational Intelligence and Security (CIS '07)*, pp. 972–976, IEEE, Harbin, China, December 2007.

[31] Y. Ji, T. Liu, and L. Min, "Generalized chaos synchronization theorems for bidirectional differential equations and discrete systems with applications," *Physics Letters A*, vol. 372, no. 20, pp. 3645–3652, 2008.

[32] L. Min and G. Chen, "Generalized synchronization in an array of nonlinear dynamic systems with applications to chaotic CNN," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 23, no. 1, Article ID 1350016, 2013.

[33] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physical Review Letters*, vol. 64, no. 8, pp. 821–824, 1990.

[34] K. Murali and M. Lakshmanan, "Secure communication using a compound signal from generalized synchronizable chaotic systems," *Physics Letters A*, vol. 241, no. 6, pp. 303–310, 1998.

[35] X. Wu, H. Hu, and B. Zhang, "Parameter estimation only from the symbolic sequences generated by chaos system," *Chaos, Solitons & Fractals*, vol. 22, no. 2, pp. 359–366, 2004.

[36] H.-K. Chen, "Global chaos synchronization of new chaotic systems via nonlinear control," *Chaos, Solitons and Fractals*, vol. 23, no. 4, pp. 1245–1251, 2005.

[37] S. Banerjee and A. Roy Chowdhury, "Lyapunov function, parameter estimation, synchronization and chaotic cryptography," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 5, pp. 2248–2254, 2009.

[38] T. Liu, L. Min, and L. Cao, "Generalized synchronization of non-autonomously discrete time chaotic system," in *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS '10)*, pp. 786–790, July 2010.

[39] ETSI/SAGE Specification, *Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3*, Document 2: ZUC Specification; Version: 1.5, 2011.

[40] T. Liu and L. Min, "Design of non-autonomous chaotic generalized synchronization based pseudorandom number generator with application in avalanche image encryption," in *Proceedings of the 17th IEEE International Conference on Computational Science and Engineering*, pp. 576–582, Chengdu, China, 2014.

[41] R. Rukhin and J. Soto, "A statistical test suite for random and pseudorandom number generator for cryptographic applications," NIST Special Publication, 2001.

[42] M. Q. Wang and L. Wang, "On stability of discrete dynamical system," *Chinese Quarterly Journal of Mathematics*, vol. 2, no. 3, pp. 12–30, 1987.

[43] NIST, "Security requirements for cryptographic modules," FIPS PUB 140-2, NIST, Gaithersburg, Md, USA, 2001.

[44] L. Min and G. Chen, "A novel stream encryption scheme with avalanche effect," *The European Physical Journal B*, vol. 86, article 459, 2013.