Dynamic Parameter-Control Chaotic System

Zhongyun Hua, *Student Member, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*—This paper proposes a general framework of 1-D chaotic maps called the dynamic parameter-control chaotic system (DPCCS). It has a simple but effective structure that uses the outputs of a chaotic map (control map) to dynamically control the parameter of another chaotic map (seed map). Using any existing 1-D chaotic map as the control/seed map (or both), DPCCS is able to produce a huge number of new chaotic maps. Evaluations and comparisons show that chaotic maps generated by DPCCS are very sensitive to their initial states, and have wider chaotic ranges, better unpredictability and more complex chaotic behaviors than their seed maps. Using a chaotic map of DPCCS as an example, we provide a field-programmable gate array design of this chaotic map to show the simplicity of DPCCS in hardware implementation, and introduce a new pseudo-random number generator (PRNG) to investigate the applications of DPCCS. Analysis and testing results demonstrate the excellent randomness of the proposed PRNG.**

***Index Terms*—Chaotic map, dynamic parameter-control chaotic system (DPCCS), field-programmable gate array (FPGA) design, pseudo-random number generator (PRNG).**

I. Introduction

THE "CHAOS" is a kind of complex and irregular behaviors. It was first observed in the meteorology, which describes the connection between the nonperiodicity of climate and the unpredictability of long-term weather forecasting [1]. Generally speaking, chaos is a particular nonlinear dynamics, whose behavior is actually described by deterministic equations [2]. It has clear boundaries and displays sensitivity to initial states [3]. Therefore, the outputs of a chaotic system can be determined only by the given initial states. Any small difference in initial states eventually results in markedly different end states later. This is also known as the butterfly effect [4]. Because chaotic systems have the significant properties of initial state sensitivity, ergodicity, and unpredictability, and these properties can be found quite similar in cryptography, they are popularly used in cryptography including pseudo-random number generator (PRNG) [5], [6], data encryption [7]–[9], image encryption [10], [11], etc. Except for the chaos-based cryptography, chaos theory is also widely studied and explored in many other fields of science and engineering [12], [13]. For example, the chaos synchronization [14], [15] explores the synchronization phenomenon when two or more dissipative chaotic systems are coupled. It has been widely used in secure communication [16]–[18]; the autonomous and nonautonomous chaotic oscillators [19], [20] are two kinds of chaotic systems that have been well studied and used in many applications [20], [21].

With the development of discerning chaos technologies, chaotic systems with poor chaos performance can be easily attacked by different methods [22], which either identify initial states of chaotic systems [23] or estimate chaotic signals [24], [25]. Their corresponding chaos-based applications are then easily broken or attacked [26], such as the chaos-based ciphers [27], [28] and secure chaotic communication [29]. On the other hand, because of the finiteness of the computing precision when a chaotic map is actually implemented, the extremely close states of a chaotic map with poor chaos performance will never overlap theoretically, but in fact they may do [30]. Thus, for a chaotic map with poor chaos performance, its observing states are far below its total number of states and then it can be easily attacked and lose its unpredictability [31], [32]. A chaotic map with good ergodicity can make its outputs separated into more ranges in its phase plane and thus its neighboring states have less probability to overlap. Therefore, developing chaotic maps with better chaos performance becomes attractive.

Recently, numerous efforts are devoted to developing new chaotic systems with better chaos performance. Several chaotic structures using existing chaotic systems have been developed [33]–[35]. Examples include the cross-coupled chaotic oscillators [36], coupled Chua's circuit [37], chaotic Jerk circuit [38], and discrete wheel-switching chaotic system [39]. However, the coupled chaotic maps have slightly improved chaos performance because most of them are generated from a simple linear combination of existing chaotic maps [40], [41]. The chaotic Jerk circuit and discrete wheel-switching chaotic system have notable time delays because they contain either a high-order differential equation [42] or an additional initialization operation [39].

This paper proposes a dynamic parameter-control chaotic system (DPCCS) as a general framework of 1-D chaotic maps. In DPCCS, a chaotic map (control map) is used to dynamically control the parameter of another chaotic map (seed map) to form a new chaotic map. Thus, the seed map generates outputs using a dynamically changed or fixed parameter provided by the control map in each iteration. Any 1-D chaotic map can be used as the control or seed map (or both) in DPCCS. Their different combinations yield a large number of new chaotic maps with

Manuscript received March 30, 2015; revised July 24, 2015; accepted November 22, 2015. Date of publication December 17, 2015; date of current version November 15, 2016. This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/016/2015/A1 and in part by the Research Committee at University of Macau under Grant MYRG2014-00003-FST, Grant MYRG113(Y1-L3)-FST12-ZYC, and Grant MRG001/ZYC/2013/FST. This paper was recommended by Associate Editor H. Zhang.

The authors are with the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: huazyum@gmail.com; yicongzhou@umac.mo).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2504180

excellent chaotic behaviors. The contributions of this paper are as follows.

1) We propose DPCCS as a general chaotic framework. Using a chaotic map to control the parameter of the seed map, the seed map in DPCCS can have a dynamic parameter in each iteration, and thus its outputs are more irregular.
2) A comprehensive analysis of the DPCCS's properties and chaotic behaviors has been performed theoretically and experimentally.
3) Using three existing chaotic maps as the control and seed maps, DPCCS produces nine new chaotic maps. Analysis and comparison results of these maps are provided to show that they are quite sensitive to initial states, and have wider chaotic ranges, better unpredictability and more complex chaotic behaviors than the corresponding seed maps.
4) Using one chaotic map generated by DPCCS, we introduce a field-programmable gate array (FPGA) design of DPCCS to show its simplicity in hardware implementation.
5) We propose a new PRNG using a chaotic map of DPCCS, and evaluate the performance of PRNG using four test standards including the National Institute of Standards and Technology (NIST) SP800-22, TestU01, Diehard statistical, and Federal Information Processing Standards Publication (FIPS PUB) 140-2 tests.

The rest of this paper is organized as follows. Section II briefly reviews three traditional chaotic maps. Section III introduces DPCCS and Section IV provides nine chaotic maps generated by DPCCS. Section V presents an FPGA implementation of DPCCS. Section VI proposes a PRNG using one newly generated chaotic map of DPCCS and Section VII reaches the conclusion.

II. Traditional Chaotic Maps

This section reviews three traditional chaotic maps, the sine, logistic and tent maps. They will be used as the control/seed map in our proposed DPCCS in Section IV.

The sine map is a widely used 1-D chaotic map derived from the sine function. Mathematically, it is defined as

$$x_{n+1} = \mathbb{S}(x) = \mu \sin(\pi x_n) \tag{1}$$

where parameter $\mu \in [0, 1]$. The sine map is chaotic when $\mu \in [0.87, 1]$ (approximately).

By stretching out and folding back an input value into a range of $[0, 1]$, the logistic map generates an output within $[0, 1]$. Its mathematical definition is shown as

$$x_{n+1} = \mathbb{L}(x) = 4\mu x_n(1 - x_n) \tag{2}$$

where its parameter $\mu \in [0, 1]$. When $\mu \in [0.9, 1]$ (approximately), the logistic map is chaotic.

The tent map is a piecewise map, which scales or folds the input value based on its range. Its mathematical representation is defined as

$$x_{n+1} = \mathbb{T}(x) = \begin{cases} 2\mu x_n & \text{for } x_n < 0.5 \\ 2\mu(1 - x_n) & \text{for } x_n \geq 0.5 \end{cases} \tag{3}$$

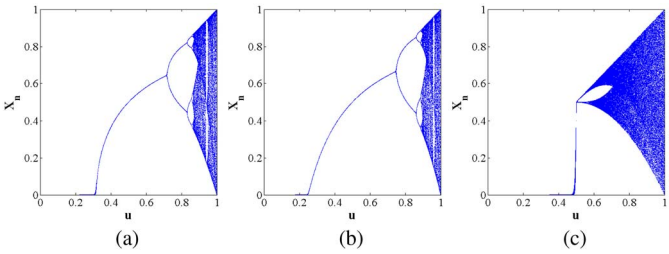

Fig. 1. Bifurcation diagrams of the (a) sine, (b) logistic, and (c) tent maps.

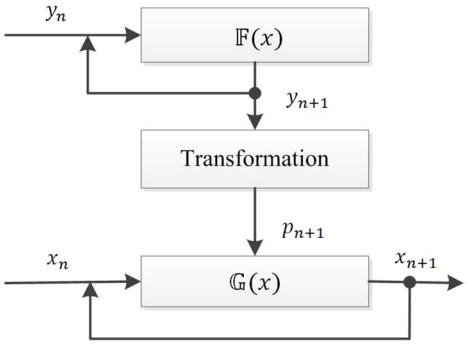

Fig. 2. Structure of DPCCS.

where parameter $\mu \in [0, 1]$. The tent map is chaotic when $\mu \in (0.5, 1)$.

When a dynamic system has chaotic behavior, its output values are irregular. The bifurcation diagrams of the sine, logistic, and tent maps are shown in Fig. 1. The sine and logistic maps have similar chaotic behaviors, which can be seen from their bifurcation diagrams in Fig. 1(a) and (b).

III. DPCCS

This section proposes DPCCS and discusses its properties.

A. DPCCS

To generate a new chaotic map, DPCCS uses a chaotic map to dynamically control the parameter of another chaotic map. Fig. 2 shows the structure of DPCCS. $\mathbb{F}(x)$ and $\mathbb{G}(x)$ are two 1-D chaotic maps. $\mathbb{F}(x)$ is called the control map while $\mathbb{G}(x)$ is the seed map to generate iterative values. p_{n+1} is the control parameter of the seed map $\mathbb{G}(x)$. The transformation process is designed to scale the output of $\mathbb{F}(x)$ into the chaotic range of the $\mathbb{G}(x)$'s parameter such that the seed map $\mathbb{G}(x)$ always has chaotic behaviors. After transformation, $\mathbb{G}(x)$ generates the output values x_{n+1} with a dynamic parameter p_{n+1}. Notice that the parameter of the seed map $\mathbb{G}(x)$ is fixed or dynamically changed in each iteration.

The mathematical definition of DPCCS is shown as

$$x_{n+1} = \mathbb{M}(x) = \mathbb{G}(p_{n+1}, x_n) \tag{4}$$

where x_n are iteration values, p_{n+1} and y_{n+1} are the outputs of the transformation $\mathbb{R}(x)$ and control map $\mathbb{F}(x)$, respectively, as defined by

$$\begin{aligned} p_{n+1} &= \mathbb{R}(y_{n+1}) \\ y_{n+1} &= \mathbb{F}(\mu, y_n) \end{aligned} \tag{5}$$

where μ is the control parameter of DPCCS.

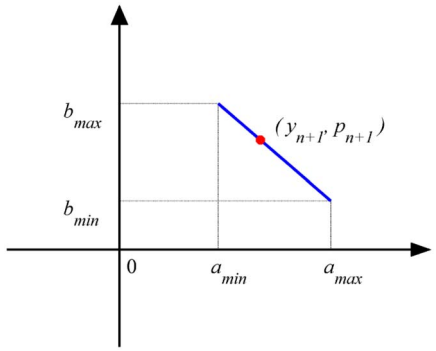

Fig. 3. Transformation $\mathbb{R}(x)$ in DPCCS.

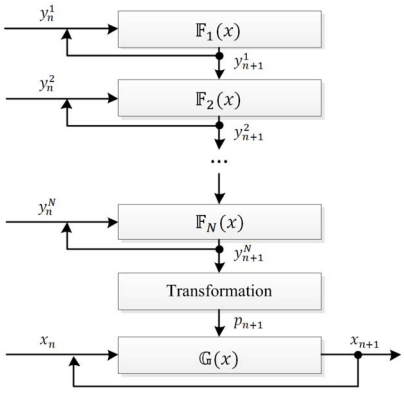

Fig. 4. Extended structure of DPCCS with N control maps.

Because the output range of the control map $\mathbb{F}(x)$ may not match the chaotic range of the seed map $\mathbb{G}(x)$, a transformation $\mathbb{R}(x)$ is designed to map the outputs of $\mathbb{F}(x)$ into the chaotic range of $\mathbb{G}(x)$. Because the control and seed maps are both existing 1-D chaotic maps, researchers have well explored their properties and defined their output and chaotic ranges in literatures. Suppose the output range of $\mathbb{F}(x)$ is $A = [a_{\min}, a_{\max}]$ and the chaotic range of $\mathbb{G}(x)$ is $B = [b_{\min}, b_{\max}]$. Two linear operations, shifting and scaling, are used in transformation $\mathbb{R}(x)$, namely

$$\frac{p_{n+1} - b_{\min}}{b_{\max} - b_{\min}} = \frac{a_{\max} - y_{n+1}}{a_{\max} - a_{\min}}. \tag{6}$$

Thus p_{n+1} can be represented as

$$p_{n+1} = \mathbb{R}(y_{n+1}) = \frac{(a_{\max} - y_{n+1})(b_{\max} - b_{\min})}{a_{\max} - a_{\min}} + b_{\min}. \tag{7}$$

For example, if $\mathbb{F}(x)$ is selected to be the tent map $\mathbb{T}(x)$ and $\mathbb{G}(x)$ to be the sine map $\mathbb{S}(x)$, then $A = [0, 1]$ and $B = [0.87, 1]$. According to (7), the transformation in each iteration is $p_{n+1} = 1 - 0.13y_{n+1}$.

Fig. 3 plots the straightforward relationship between y_{n+1} and p_{n+1}. As can be seen, the transformation process in DPCCS is a linear and one-to-one mapping.

B. Properties

Using one existing chaotic map to control the parameter of another chaotic map, DPCCS in Fig. 2 can achieve many unique properties.

1) Since any chaotic map can be used as the control map $\mathbb{F}(x)$, seed map $\mathbb{G}(x)$, or both, DPCCS offers users the great flexibility to generate a large number of new chaotic maps using different settings of $\mathbb{G}(x)$ and $\mathbb{F}(x)$.
2) $\mathbb{F}(x)$ and $\mathbb{G}(x)$ can be the same or different chaotic maps. When $\mathbb{F}(x)$ and $\mathbb{G}(x)$ are the same chaotic maps, DPCCS can be represented by

$$\begin{cases} x_{n+1} = \mathbb{G}(\mathbb{R}(y_{n+1}), x_n) \\ y_{n+1} = \mathbb{G}(\mu, y_n) \end{cases} \text{ or } \begin{cases} x_{n+1} = \mathbb{F}(\mathbb{R}(y_{n+1}), x_n) \\ y_{n+1} = \mathbb{F}(\mu, y_n). \end{cases} \tag{8}$$

Examples include the tent-control-tent (TCT), logistic-control-logistic (LCL), and sine-control-sine (SCS) maps in Section IV-A. When $\mathbb{F}(x)$ and $\mathbb{G}(x)$ are different chaotic maps, DPCCS is the structure represented by (4) or

$$\begin{cases} x_{n+1} = \mathbb{G}(\mathbb{R}(y_{n+1}), x_n) \\ y_{n+1} = \mathbb{F}(\mu, y_n). \end{cases} \tag{9}$$

Even swapping the settings of $\mathbb{G}(x)$ and $\mathbb{F}(x)$, DPCCS yields different chaotic maps. For example, the tent-control-logistic and logistic-control-tent maps in Section IV-A are completely different chaotic maps. Thus, if L existing 1-D chaotic maps are used, the total number of newly generated chaotic maps can be

$$TN = L^2. \tag{10}$$

3) The newly generated chaotic maps of DPCCS have wider chaotic ranges, more complex chaotic behaviors and better chaos performance than their seed maps. These will be discussed in Section III-C and further verified by experimental results in Section IV-B.
4) The structure of DPCCS can be further extended into two or even more control maps. Fig. 4 shows an extension example of DPCCS with N control chaotic maps. The outputs of kth ($1 \le k \le N - 1$) control map $\mathbb{F}_k(x)$ are used to dynamically control the parameter of the $(k + 1)$th control map $\mathbb{F}_{k+1}(x)$. And the outputs of the Nth control map are used to dynamically control the parameters of the seed map $\mathbb{G}(x)$ to generate iterative values. This extended structure has good chaos performance and complex chaotic behavior because the seed map's parameters are controlled by N chaotic maps. However, it also results in a complex structure, expensive implementation, and difficult performance analysis.

C. Chaotic Behavior Analysis

DPCCS uses a control map $\mathbb{F}(x)$ to dynamically control the parameter of the seed map $\mathbb{G}(x)$ to generate new chaotic maps. This can significantly enhance the chaotic complexity and chaos performance of DPCCS. Here, we use the Lyapunov exponent (LE) [43] to analyze the chaotic behavior of DPCCS.

A dynamic system with chaos nature is unpredictable and sensitive to its initial states [44]. LE denotes the exponential divergence of two extremely close trajectories of a dynamic

system in the phase plane, and it is a widely used indictor to demonstrate chaos. A positive LE value means that no matter how small difference the two trajectories have, they exponentially diverge in each unit time, and will be totally different as the time changes. Therefore, a dynamic system with at least one positive LE value usually has chaotic behavior.

Suppose $\mathbb{G}(x)$ is a differentiable function, LE of the proposed DPCCS in (4) is defined as

$$\lambda_{\mathbb{M}} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \ln\left|\mathbb{G}'(p, x_n)\right|. \tag{11}$$

Because the parameter of DPCCS is controlled by a chaotic map, its parameter can be fixed or dynamically changed in each iteration. Suppose the chaotic range of the seed map $\mathbb{G}(x)$ $[b_{\min}, b_{\max}]$ is continuous, according to chaotic behavior of its control map $\mathbb{F}(x)$, the LE value of DPCCS can be analyzed from the three following ways.

1) When the attractor of $\mathbb{F}(x)$ is a fixed point, namely $\mathbb{F}(x)$ generates a fixed output value. As can be seen from the structure of DPCCS in Fig. 2, this fixed output will be transformed to be a parameter value P in the chaotic range of $\mathbb{G}(x)$. The LE value of DPCCS equals to the LE value of its seed map $\mathbb{G}(x)$ under the parameter P, namely

$$\lambda_{\mathbb{M}} = \lambda_{\mathbb{G}^P} > 0.$$

2) When the attractors of $\mathbb{F}(x)$ are a limit cycle, namely $\mathbb{F}(x)$ has a periodic orbit and its outputs are a finite number of different points, suppose $\{o_i | i = 1, 2, \ldots, k\}$ (k is a finite number) [45]. After transformation, $\{o_i | i = 1, 2, \ldots, k\}$ will be transformed as $\{p_i | i = 1, 2, \ldots, k\}$, which are in the chaotic range of $\mathbb{G}(x)$. Because the k finite outputs of $\mathbb{F}(x)$ are periodic points, when the iteration number N closes to ∞, the number of each point approaches to N/k. Then the LE value of DPCCS can be defined as

$$\lambda_{\mathbb{M}} = \left\{ \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{(N/k)-1} \ln\left|\mathbb{G}'(p_1, x_n)\right| + \cdots + \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{(N/k)-1} \ln\left|\mathbb{G}'(p_k, x_n)\right| \right\}. \tag{12}$$

Because $N \to \infty$ and k is a finite number, then $(N/k) \to \infty$. Thus

$$\begin{aligned} &\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{(N/k)-1} \ln\left|\mathbb{G}'(p_i, x_n)\right| \\ &= \lim_{N\to\infty} \frac{1}{k} \frac{1}{N/k} \sum_{n=0}^{(N/k)-1} \ln\left|\mathbb{G}'(p_i, x_n)\right| \\ &= \frac{1}{k} \left\{ \lim_{(N/k)\to\infty} \frac{1}{N/k} \sum_{n=0}^{(N/k)-1} \ln\left|\mathbb{G}'(p_i, x_n)\right| \right\} \\ &= \frac{1}{k} \lambda_{\mathbb{G}^{p_i}} \end{aligned} \tag{13}$$

where $i = 1, 2, \ldots, k$. Then (12) becomes

$$\begin{aligned} \lambda_{\mathbb{M}} &= \frac{1}{k}\lambda_{\mathbb{G}^{p_1}} + \frac{1}{k}\lambda_{\mathbb{G}^{p_2}} + \cdots + \frac{1}{k}\lambda_{\mathbb{G}^{p_k}} \\ &= \frac{1}{k} \sum_{i=1}^{k} \lambda_{\mathbb{G}^{p_i}}. \end{aligned} \tag{14}$$

Because $p_i (i = 1, 2, \ldots, k)$ are in the chaotic range of $\mathbb{G}(x)$, then $\lambda_{\mathbb{G}^{p_i}} > 0$ for $\forall i \in \{1, 2, \ldots, k\}$. Thus

$$\lambda_{\mathbb{M}} = \frac{1}{k} \sum_{i=1}^{k} \lambda_{\mathbb{G}^{p_i}} > 0.$$

3) When $\mathbb{F}(x)$ has strange attractors, $\mathbb{F}(x)$ is chaotic and its outputs are dynamic and never repeat. In this case, the seed map $\mathbb{G}(x)$ has a different parameter setting provided by $\mathbb{F}(x)$ in each iteration, and thus make the iterative outputs of DPCCS different and unpredictable.

Therefore, when the seed map $\mathbb{G}(x)$ has a continuous chaotic range $[b_{\min}, b_{\max}]$, no matter whether the control map $\mathbb{F}(x)$ is chaotic or not, DPCCS has chaotic behavior and good chaos performance for all parameter settings. However, if the chaotic range of $\mathbb{G}(x)$ is not continuous and $\mathbb{F}(x)$ is not on its chaotic range, DPCCS may loss its chaotic behavior. This occurs when the fixed output(s) of $\mathbb{F}(x)$ happen(s) to be transformed into the nonchaotic ranges of $[b_{\min}, b_{\max}]$, such as the white space in the chaotic ranges of the logistic and sine maps [see Fig. 1(a) and (b)]. These will be further verified by LE results of newly generated chaotic maps of DPCCS in Section IV-B.

IV. Examples and Analysis

Any existing 1-D chaotic map can be used as the control and seed maps in DPCCS. The control and seed maps can be the same or different chaotic maps. According to (10), using the three existing chaotic maps presented in Section II, a total number of nine new chaotic maps can be generated by DPCCS. This section discusses these nine examples of new chaotic maps.

A. Examples of DPCCS

Table I shows the definitions of the nine chaotic maps of DPCCS. When the control and seed maps are selected as the same chaotic maps, the parameter of a chaotic map is controlled by the same chaotic map to generate a new chaotic map, such as the TCT, LCL, and SCS maps. Their control and seed maps are the tent, logistic, and sine maps, respectively. When choosing the control and seed maps as two different chaotic maps, DPCCS is able to generate a large number of different chaotic maps. Exchanging the settings of the control and seed maps, DPCCS yields a different chaotic map. For example, the LCS and SCL maps are two completely different chaotic maps.

Fig. 5 shows bifurcation diagrams of these chaotic maps defined in Table I. As can be seen, for the TCT, TCL, TCS, LCT, and SCT maps in Fig. 5(a)–(e), they have chaotic behaviors in the whole parameter ranges. For the SCL, LCL, LCS, and SCS maps in Fig. 5(f)–(i), their outputs distribute randomly in most parameter settings but their chaotic ranges

TABLE I
NINE NEW CHAOTIC MAPS GENERATED BY DPCCS

Seed map $\mathbb{G}(x)$	Control map $\mathbb{F}(x)$	New chaotic maps	Definition
$\mathbb{T}(x)$	$\mathbb{T}(x)$	Tent-control-Tent (TCT)	$x_{n+1}=\begin{cases}2(0.99-0.5y_{n+1})x_n & \text{for } x_n<0.5\\ 2(0.99-0.5y_{n+1})(1-x_n) & \text{for } x_n\geq 0.5\end{cases}$, where $y_{n+1}=\begin{cases}2\mu y_n & \text{for } y_n<0.5\\ 2\mu(1-y_n) & \text{for } y_n\geq 0.5\end{cases}$
	$\mathbb{L}(x)$	Logistic-control-Tent (LCT)	$x_{n+1}=\begin{cases}2(0.99-0.5y_{n+1})x_n & \text{for } x_n<0.5\\ 2(0.99-0.5y_{n+1})(1-x_n) & \text{for } x_n\geq 0.5\end{cases}$, where $y_{n+1}=4\mu y_n(1-y_n)$
	$\mathbb{S}(x)$	Sine-control-Tent (SCT)	$x_{n+1}=\begin{cases}2(0.99-0.5y_{n+1})x_n & \text{for } x_n<0.5\\ 2(0.99-0.5y_{n+1})(1-x_n) & \text{for } x_n\geq 0.5\end{cases}$, where $y_{n+1}=\mu\sin(\pi y_n)$
$\mathbb{L}(x)$	$\mathbb{T}(x)$	Tent-control-Logistic (TCL)	$x_{n+1}=4(1-0.1y_{n+1})x_n(1-x_n)$, where $y_{n+1}=\begin{cases}2\mu y_n & \text{for } y_n<0.5\\ 2\mu(1-y_n) & \text{for } y_n\geq 0.5\end{cases}$
	$\mathbb{L}(x)$	Logistic-control-Logistic (LCL)	$x_{n+1}=4(1-0.1y_{n+1})x_n(1-x_n)$, where $y_{n+1}=4\mu y_n(1-y_n)$
	$\mathbb{S}(x)$	Sine-control-Logistic (SCL)	$x_{n+1}=4(1-0.1y_{n+1})x_n(1-x_n)$, where $y_{n+1}=\mu\sin(\pi y_n)$
$\mathbb{S}(x)$	$\mathbb{T}(x)$	Tent-control-Sine (TCS)	$x_{n+1}=(1-0.13y_{n+1})\sin(\pi x_n)$, where $y_{n+1}=\begin{cases}2\mu y_n & \text{for } y_n<0.5\\ 2\mu(1-y_n) & \text{for } y_n\geq 0.5\end{cases}$
	$\mathbb{L}(x)$	Logistic-control-Sine (LCS)	$x_{n+1}=(1-0.13y_{n+1})\sin(\pi x_n)$, where $y_{n+1}=4\mu y_n(1-y_n)$
	$\mathbb{S}(x)$	Sine-control-Sine (SCS)	$x_{n+1}=(1-0.13y_{n+1})\sin(\pi x_n)$, where $y_{n+1}=\mu\sin(\pi y_n)$

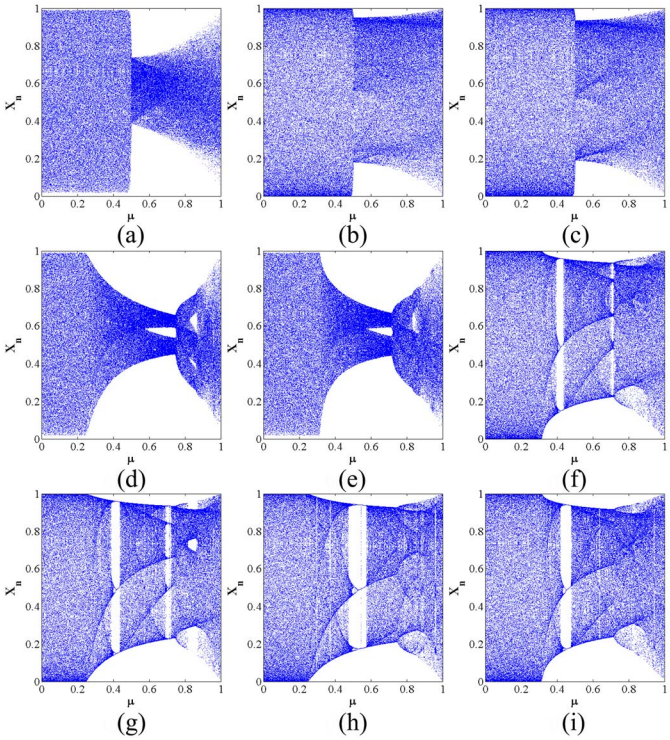

Fig. 5. Bifurcation diagrams of the (a) TCT, (b) TCL, (c) TCS, (d) LCT, (e) SCT, (f) SCL, (g) LCL, (h) LCS, and (i) SCS maps.

are not continuous. This is because their corresponding seed maps, the logistic and sine maps, have discontinuous chaotic ranges and their control maps' outputs happen to be transformed in the nonchaotic ranges of seed maps.

B. Performance Analysis

To analyze chaos performance of DPCCS, here we compare these nine chaotic maps with their corresponding seed maps using LE [43], Kolmogorov entropy (KE) [46] and correlation dimension (CD) [47], [48], and analyze their sensitivity to initial states as well. The analysis and comparison results show that new chaotic maps have better chaos performance than their seed maps.

1) LE: As mentioned in Section III-C, LE is a measure to test whether a dynamic system has chaotic behavior. Fig. 6 plots the LE values of different chaotic maps. As can be seen, only the LCL, SCL, LCS, and SCS maps have chaotic behaviors in most parameter settings while other new chaotic maps have chaotic behaviors in whole parameter ranges. In addition, Table II lists the average LE values of different chaotic maps in their respective chaotic ranges. The new chaotic maps have bigger average LE values than their corresponding seed maps.

A dynamic system with a positive LE value is chaotic and a larger LE value means better chaos performance. Compared with their seed maps, the new chaotic maps generated by DPCCS have wider parameter ranges with positive LE values, and their LE values are bigger in most parameter settings. Thus, these new chaotic maps have more complex chaotic behaviors than their seed maps.

2) KE: In the algorithmic information theory, KE is a notion of entropy that provides a mathematical explication of the randomness of a finite object. It can be used to describe how much extra information is needed to predict the output values of a dynamic system. Mathematically, KE is defined as [46]

$$\mathrm{KE}=\lim_{\tau\to 0}\tau^{-1}\lim_{\varepsilon\to 0}\lim_{m\to\infty}K_{m,\tau}(\varepsilon) \tag{15}$$

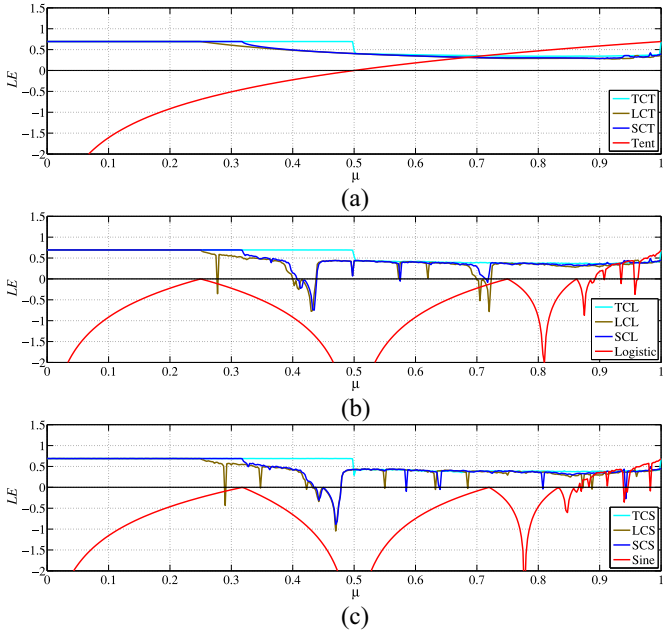

Fig. 6. LE comparisons of the (a) TCT, LCT, SCT, and tent maps, (b) TCL, LCL, SCL, and logistic maps, and (c) TCS, LCS, SCS, and sine maps, respectively.

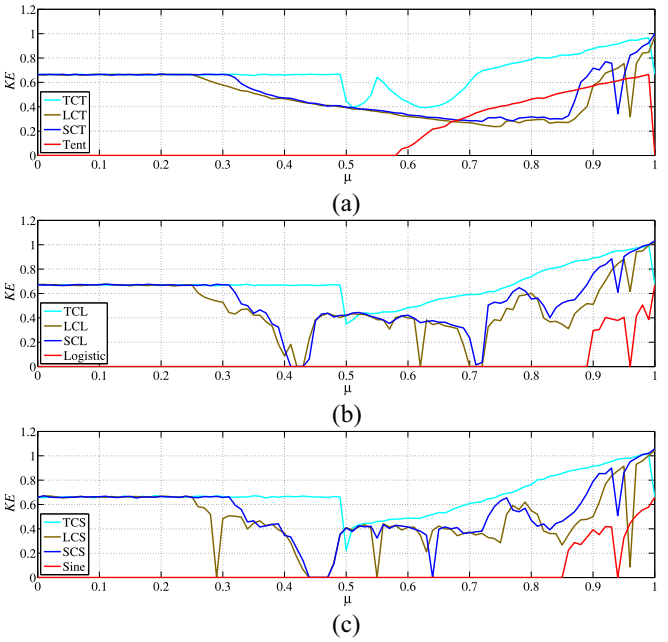

Fig. 7. KE comparisons of the (a) TCT, LCT, SCT, and tent maps, (b) TCL, LCL, SCL, and logistic maps, and (c) TCS, LCS, SCS, and sine maps, respectively.

TABLE II
AVERAGE MEASURE RESULTS OF DIFFERENT CHAOTIC MAPS WITHIN THEIR OWN CHAOTIC RANGES

Chaotic maps	LE	KE	CD
TCT	0.527483	0.671001	1.417556
LCT	0.463243	0.487013	1.083424
SCT	0.474353	0.515105	1.117783
Tent	0.388001	0.345781	0.876437
TCL	0.543375	0.671856	1.188496
LCL	0.470447	0.495264	0.994097
SCL	0.485568	0.544218	1.016035
Logistic	0.385349	0.415624	0.906299
TCS	0.541077	0.675575	1.195133
LCS	0.478923	0.501478	0.991610
SCS	0.493949	0.541542	1.023502
Sine	0.389477	0.437797	0.932510

where m indicates the embedding dimension, and $K_{m,\tau}(\varepsilon)$ is defined by

$$K_{m,\tau}(\varepsilon) = -\sum_{i_1,i_2,\ldots,i_m \le n(\varepsilon)} \Pr(i_1, i_2, \ldots, i_m) \log \Pr(i_1, i_2, \ldots, i_m) \tag{16}$$

where $\phi_{i_1}, \phi_{i_2}, \ldots, \phi_{i_m}$ are m nonoverlapping partitions of the phase plane of a dynamic system, $\Pr(i_1, i_2, \ldots, i_m)$ is the joint probability of the correctly predicting trajectory in partition ϕ_{i_1} at time τ, in partition ϕ_{i_2} at time 2τ, $\ldots$, in partition ϕ_{i_m} at time $m\tau$.

A positive KE value means that extra information is needed to predict the trajectories of a dynamic system and a bigger KE value indicates more information needed. Therefore, a dynamic system with a positive KE value is unpredictable and a bigger KE value means better unpredictability.

In each calculation in our experiments, 12 000 successive states are truncated from a trajectory and the KE value is calculated using the methods in [46]. Fig. 7 plots the KE values of different chaotic maps along with their control parameters. The average KE values of these chaotic maps within their respective chaotic ranges are listed in Table II. From Fig. 7 and Table II, we can observe that all chaotic maps generated by DPCCS have positive KE values in much wider ranges than their seed maps. Moreover, compared with their seed maps, the newly generated chaotic maps have much bigger KE values in most parameter settings, and have larger average KE values in their respective chaotic ranges. These mean that the trajectories of the chaotic maps generated by DPCCS are far more unpredictable.

3) CD: As a type of fractal dimensions, CD describes the dimensionality of the space occupied by a set of points. It can be used to characterize the attractor strangeness (degrees of freedom) of a dynamic system. Its value can be calculated by the Grassberger–Procaccia algorithm [48] with the experimental signals.

For a sequence of experimental signals $\{s_i, |i = 1, 2, \ldots, N\}$ with N points, given an embedding dimension e, the CD values can be calculated as

$$d = \lim_{r\to 0} \lim_{N\to\infty} \frac{\log C_e(r)}{\log r} \tag{17}$$

where $C_e(r)$ is the correlation integral defined as

$$C_e(r) = \lim_{N\to\infty} \frac{1}{[N-(e-1)\zeta][N-(e-1)\zeta-1]} \times \sum_{i=1}^{N-(e-1)\zeta} \sum_{j=i+1}^{N-(e-1)\zeta} \theta\left(r - |\bar{s}_i - \bar{s}_j|\right) \tag{18}$$

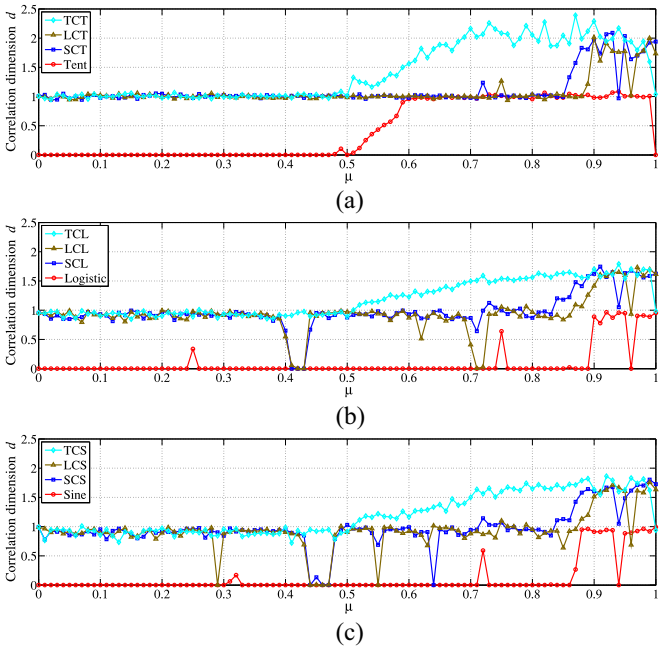

Fig. 8. CD comparisons of the (a) TCT, LCT, SCT, and tent maps, (b) TCL, LCL, SCL, and logistic maps, and (c) TCS, LCS, SCS, and sine maps, respectively.

where $\theta(\omega)$ is the Heaviside step function. $\theta(\omega) = 0$ if $\omega \leq 0$ while $\theta(\omega) = 1$ if $\omega > 0$. ζ is the time delay and for discrete systems, ζ usually equals to 1. The new data sequence $\{\bar{s}_t | t = 1, 2, 3, \ldots\}$ is

$$\bar{s}_t = \left(s_t, s_{t+\zeta}, s_{t+2\zeta}, \ldots, s_{t+(e-1)\zeta}\right)$$
$$t = 1, 2, \ldots, N - (e-1)\zeta.$$

If it exists, d is the slope of the log–log plot of $C_e(r)$ versus r, as defined by

$$d = \lim_{r \to 0} \lim_{N \to \infty} \frac{d[\log C_e(r)]/dr}{d(\log r)/dr}. \tag{19}$$

We use the method in [47] to calculate the CD values of chaotic maps with different parameter settings. The embedding dimension e is set as 2. Fig. 8 shows the CD results of different chaotic maps. For the new chaotic maps generated by DPCCS, they have much bigger CD values than their corresponding seed maps in most parameter settings. The CD values of their seed maps are quite small and close to 0 in most parameter ranges. On the other hand, in their respective chaotic ranges, new chaotic maps of DPCCS have bigger average CD values than their seed maps as shown in Table II. These mean that the outputs of new chaotic maps can occupy a much higher dimensionality in the phase plane, and thus their attractors are more irregular.

4) Sensitivity: A specific feature of chaos is its sensitivity to initial states [3]. For a chaotic map, its trajectories are sensitive to the changes of its initial value and parameter, respectively. This sensitivity can be measured by the correlation of two data sequences as defined

$$Co = \frac{E[(X_t - \mu_X)(Y_t - \mu_Y)]}{\sigma_X \sigma_Y} \tag{20}$$

TABLE III
CORRELATION VALUES OF OUTPUT SEQUENCES OF DIFFERENT CHAOTIC MAPS

Chaotic maps	Correlation of H_1, H_2	Correlation of H_3, H_4
TCT	0.009801	-0.007338
LCT	-0.009917	-0.005691
SCT	0.005562	-0.005910
TCL	-0.007327	-0.003940
LCL	0.004755	-0.005931
SCL	0.000075	-0.008867
TCS	0.002132	0.003972
LCS	-0.021694	-0.004382
SCS	-0.002959	0.002501

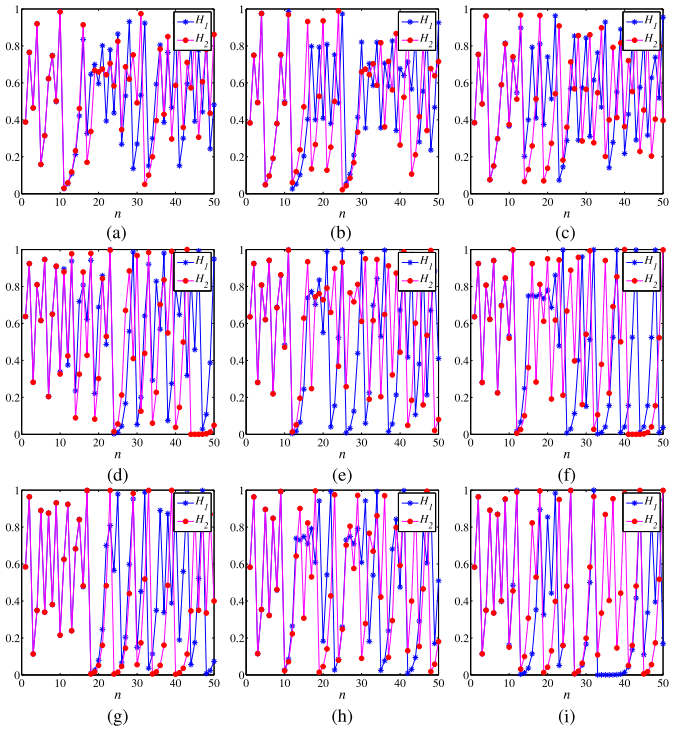

Fig. 9. Two trajectories H_1 and H_2 with initial values 0.20 (H_1) and 0.20001 (H_2) generated by the (a) TCT, (b) LCT, (c) SCT, (d) TCL, (e) LCL, (f) SCL, (g) TCS, (h) LCS, and (i) SCS maps.

where X_t and Y_t are two data sequences generated by a chaotic map with a tiny change in its initial value or parameter, μ and σ are the mean value and standard deviation, $E[.]$ is the expectation function. The correlation value closing to 0 indicates an extremely low relationship of two data sequences and thus the high sensitivity of the chaotic map, and vice versa.

Sequences H_1 and H_2 are generated by applying a tiny change to the initial values of chaotic maps while sequences H_3 and H_4 are generated with a tiny change applied to the parameter. Table III shows the correlation results of nine chaotic maps of DPCCS. As can be seen, the correlation values of sequence pairs H_1 and H_2, H_3 and H_4 of all chaotic maps extremely close to 0. Fig. 9 plots the trajectories of first 50 iteration values of these chaotic maps with a tiny change in initial

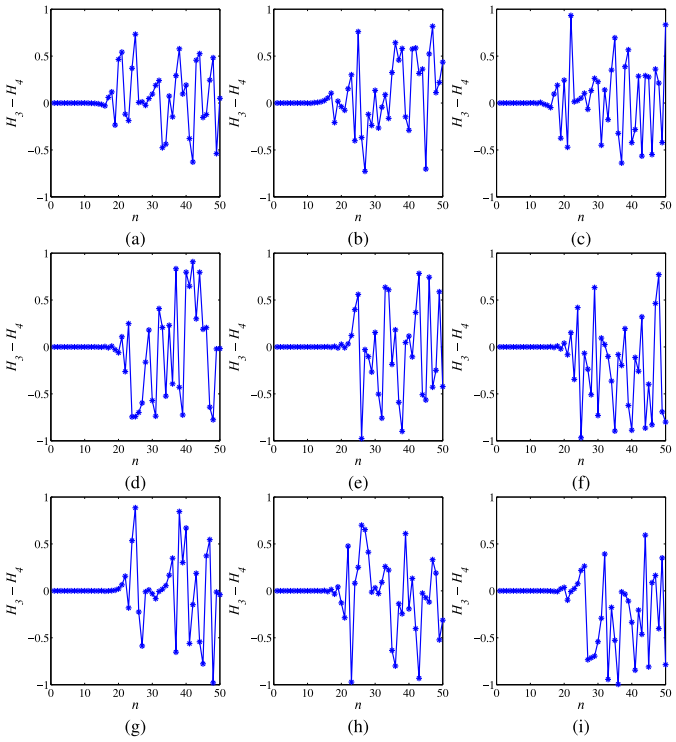

Fig. 10. Differences of two trajectories H_3 and H_4 with parameters 0.9 (H_3) and 0.90001 (H_4) generated by the (a) TCT, (b) LCT, (c) SCT, (d) TCL, (e) LCL, (f) SCL, (g) TCS, (h) LCS, and (i) SCS maps.

values and Fig. 10 plots the differences of two trajectories of first 50 iteration values with a tiny change in parameters of these chaotic maps. With the iteration number increasing, we can observe that the tiny change in initial values or parameters will make the output values significantly different. This further verifies that these new chaotic maps are extremely sensitive to their initial values and parameters.

V. Hardware Implementation

Because DPCCS uses traditional chaotic maps as the control and seed maps, it has simplicity in hardware implementation. This section uses the TCL map as an example to demonstrate the hardware implementation of DPCCS.

A. Circuit Design

The schematic of the TCL map is shown in Fig. 11. It consists of three circuit units including the tent map, transformation, and logistic map units. x_n, y_n, and μ are the circuit inputs while x_{n+1} is the circuit output. The tent map unit composes of the switch, addition, and multiplications. Depending on the data ranges of an input y_n, different operations are used to generate y_{n+1} that is then transformed and multiplied with the input x_n to generate the output x_{n+1}.

B. FPGA Implementation and Simulation

The TCL map circuit in Fig. 11 is implemented using FPGA in the Altium Designer software and the values are represented by 64 bits. The FPGA circuit diagram of the TCL map is depicted in Fig. 12. As can be seen, *ivp* and *ivx* are the initial values of the control and seed maps,

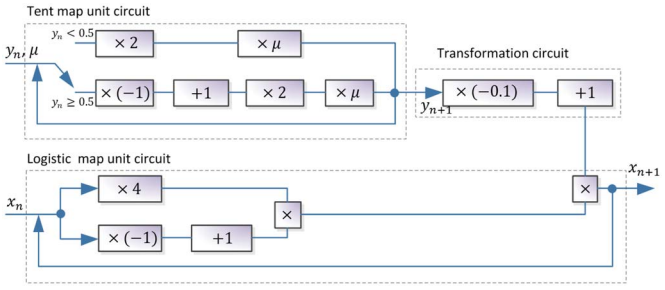

Fig. 11. Schematic of the TCL map.

and u is the control parameter. In this simulation, we set parameters $u[63...0] = 4\,000\,000\,000\,000\,000$ (namely $\mu = 0.25$), $ivx[63...0] = 2\,666\,666\,666\,666\,600$ (namely $x_0 = 0.15$), $ivp[63...0] = 3\,333\,333\,333\,333\,400$ (namely $y_0 = 0.2$). Fig. 13 shows the waveforms of simulation results. The first four outputs are 8141205*BC*01*A*35*BC*, *FEB*20*EB*14*DDF*3*E BD*, 052*DA*43*AAC*78*CC*92, and 1444*D*03558*F*1*D*437, and their corresponding decimal values are 0.504900000000000, 0.994904440200000, 0.020227684330595, and 0.079175007 843282, respectively.

Fig. 14 plots the outputs of the TCL map under the software and hardware environments, which use the MATLAB and PFGA to implement, respectively. As can be seen in Fig. 14(c), under the same initial values and parameter, the TCL outputs in both implementations are almost the same at the beginning. When iteration n increases, the differences between their outputs become bigger. This is because the data precisions in the MATLAB and FPGA implementations are different and the TCL map is extremely sensitive to the changes of its initial settings.

VI. PRNG

The PRNGs are widely used in art, statistics, modern cryptographic systems, and many other scientific areas [49]. The random numbers are usually generated in two ways [50]: one corresponds to the nondeterministic and stochastic physical phenomena [51] while the other generates pseudo-random numbers from the deterministic but chaotic dynamic systems [52]. Because the initial states sensitivity, unpredictability, and ergodicity of chaotic system are analogous to the unpreditability and randomness of pseudo-random numbers [5], [53], the quality of the chaos-based PRNGs is highly dependent on the chaos performance of chaotic systems. As verified in Section IV-B, the new chaotic systems generated by DPCCS have better chaos performance than their seed maps, and thus more appropriate for designing PRNGs. Using the TCL map as an example, this section proposes a chaos-based PRNG, called TCL-PRNG.

A. TCL-PRNG

Suppose $\{X^I | x_i^I, i = 1, 2, \ldots N\}$ and $\{X^{II} | x_i^{II}, i = 1, 2, \ldots, N\}$ are two chaotic sequences generated by the TCL map with different initial values and parameters. Z is the addition of X^I and X^{II}, $\{Z | z_i = x_i^I + x_i^{II}\}$. TCL-PRNG is then defined as

$$Q_i = \sum_{i=1}^{k} M_i \mod 2 \tag{21}$$

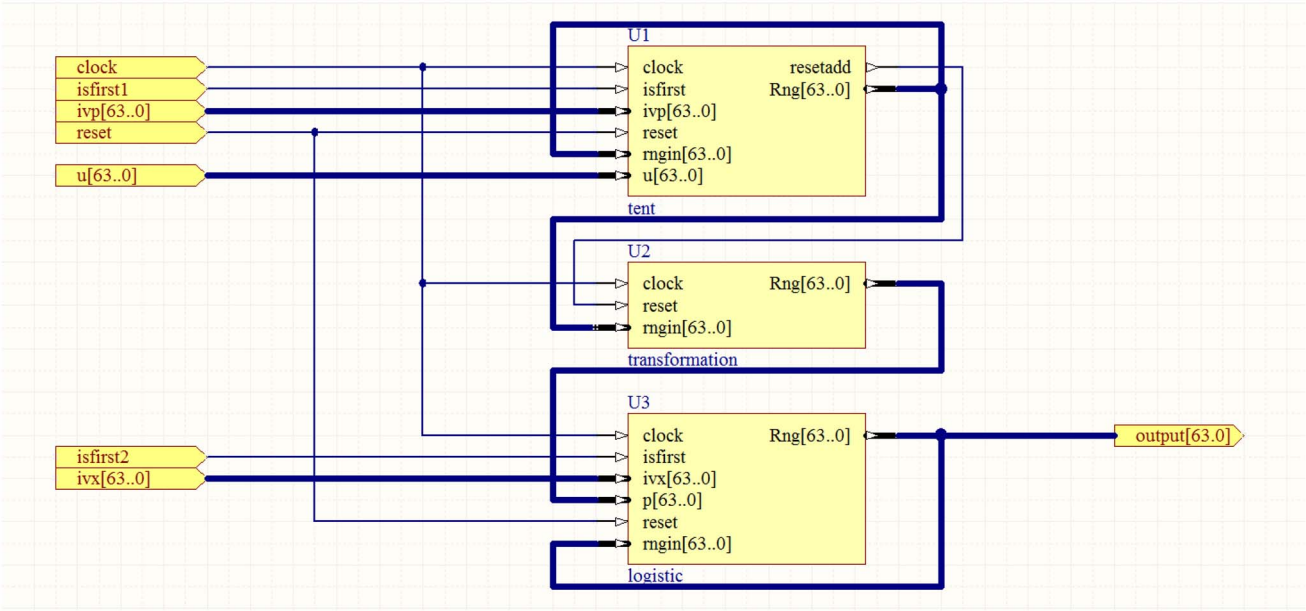

Fig. 12. FPGA circuit diagram of the TCL map.

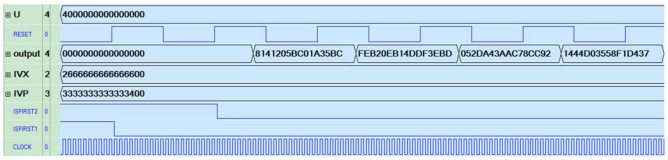

Fig. 13. Waveforms of TCL map in the FPGA implementation.

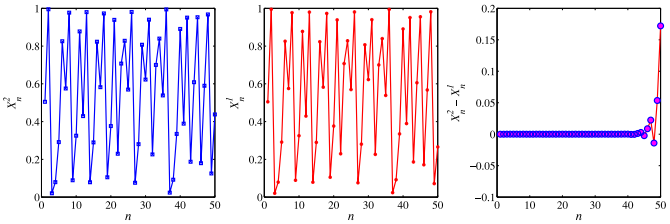

Fig. 14. Outputs of the TCL map with the first 50 iterations in the (a) MATLAB and (b) FPGA implementations using parameter $\mu = 0.25$ and initial values $x_0 = 0.15$, $y_0 = 0.2$. (c) Differences between (a) and (b).

where

$$k = \left\lfloor z_i \times 2^3 \right\rfloor + 2$$
$$M = T\lfloor z_i \rceil_{1:k} \tag{22}$$

where $T\lfloor m \rceil_{1:k}$ is a truncation function to fetch the decimal numbers from the 1st to kth decimal positions in a float number m.

B. Performance Evaluation

A PRNG should have the ability to generate long pseudo-random numbers with good randomness. Here, we analyze the periodicity of TCL-PRNG and use four test suits, the NIST SP800-22 [54], TestU01 [55], Diehard statistical [56], and FIPS PUB 140-2 [57] tests to evaluate its randomness.

TABLE IV
EXAMPLE OF HOW THE ADDITION OPERATION CAN ENLARGE THE CYCLE LENGTH OF A CHAOTIC ORBIT

Chaotic orbit												Cycle length
X^I	1	2	1	2	1	2	1	2	1	2	1	2	**2**
X^{II}	1	2	3	1	2	3	1	2	3	1	2	3	**3**
$Z = X^I + X^{II}$	2	4	4	3	3	5	2	4	4	3	3	5	**6**

1) Periodicity: When a chaotic map is directly used as a PRNG, its output sequences are usually used as the pseudo-random numbers. Due to the limited computing precision of the computer and the digitalization of the output chaotic sequence, the generated numbers of a PRNG may loss their randomness and become cyclic [30]. The proposed TCL-PRNG generates the pseudo-random numbers by adding two chaotic sequences of the TCL map with different settings of initial values and parameters. This significantly enlarges the cycle length of the output sequence and increases the randomness of TCL-PRNG. A straightforward example is shown in Table IV. As can be seen, the cycle lengths of two sequences X^I and X^{II} are 2 and 3, respectively. However, the period of $Z = X^I + X^{II}$ is 6, which is much larger than those of sequences X^I and X^{II}. Thus, the pseudo-random numbers generated by TCL-PRNG have a significantly large cycle length and high quality of randomness.

2) NIST SP800-22 Test Suit: Among all test standards, the NIST Statistical Test Suite, SP 800-22 [54], is a cogent and all-sided method to test the randomness of a binary sequence. It has 15 subtests that are designed to find the nonrandom area in a binary sequence in all aspects. The length of the testing binary sequence is suggested as 10^6.

The significance level in the NIST SP800-22 Test Suit is preset as $\alpha = 0.01$ and the sample size of binary sequences should be not less than the inverse of the significance

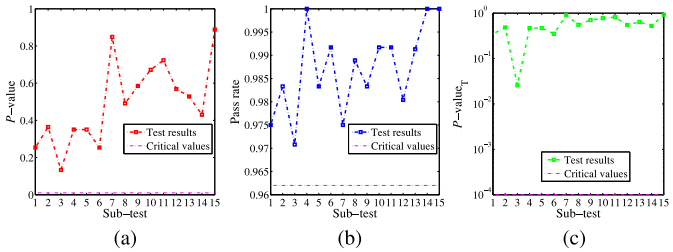

Fig. 15. Randomness test results of TCL-PRNG using the NIST SP800-22 Test Suite. (a) *P*-value interpretation. (b) Pass rate interpretation. (c) *P*-value$_T$ interpretation.

TABLE V
P-VALUE$_T$ RESULTS OF DIFFERENT PRNGS

PRNGs	Pass No.	Pass rate
New CI [39] ($m^n = y^n \ mod$ N)	15/15	100.00%
New CI [39] (no mark)	8/15	53.33%
Ole CI [39]	11/15	73.33%
New CI [39] (g_1)	15/15	100.00%
New CI [39] (g_2)	15/15	100.00%
DWSCS-PRNG [39]	15/15	100.00%
TCL-PRNG	15/15	100.00%

level (α^{-1}). In our experiments, 120 binary sequences with 10^6 bits are generated by TCL-PRNG. Each subtest will generate a *P*-value.

The test results can be interpreted in three ways. They are *P*-value, pass rate and *P*-value$_T$ interpretations. The *P*-value interpretation is to test whether the generated *P*-values are within [0.01, 1]. A PRNG passes the test if the *P*-values are in this range. The pass rate interpretation is to calculate the pass rate of *P*-value that indicates the proportion of sequences passing the *P*-value test among all testing sequences. In our experiments, the proportion of a subtest falling into range of [0.9628, 1] is considered to pass the test. Each subtest will also generate a *P*-value for each binary sequence. The *P*-value$_T$ interpretation is a statistic of the distribution of these *P*-values and its calculation method is mentioned in [54]. *P*-value$_T \geq$ 0.0001 means that the sample sequences are uniform to pass the corresponding subtest.

Fig. 15 shows the test results of each subtest for these three interpretations. As can be seen, 120 binary sequences generated by TCL-PRNG can pass all the *P*-value, pass rate and *P*-value$_T$ interpretations of all subtests.

Table V compares the *P*-value$_T$ interpretation results of TCL-PRNG with other PRNGs reported in [39]. The results show that TCL-PRNG and several existing PRNGs can pass all 15 subtests.

3) TestU01 Test: The TestU01 [55] is a widely accepted standard that also offers a collection of utilities for the empirical statistical tests of a PRNG. There are six predefined batteries of tests in the TestU01 and our experiments select three binary sequence test batteries, the Rabbit, Alphabit, and BlockAlphabit, to test the quality of TCL-PRNG. The Rabbit applies 38 different statistical tests while the Alphabit and BlockAlphabit apply 17 different statistical tests.

TABLE VI
TESTU01 RESULTS OF BINARY SEQUENCES GENERATED BY TCL-PRNG

Test batteries \ Lengths	2^{20} bits	2^{25} bits	2^{30} bits
Rabbit	38/38	38/38	38/38
Alphabit	17/17	17/17	17/17
BlockAlphabit	17/17	17/17	17/17

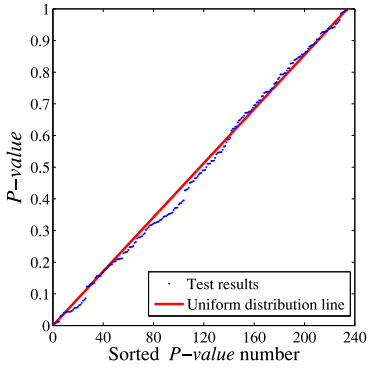

Fig. 16. Diehard statistical test results of TCL-PRNG.

In each statistical test, the generated *P*-value within range of [0.001, 0.999] is considered to pass the tests.

The software package in our experiment is TestU01-1.2.3. Three binary sequences that are generated by TCL-PRNG with lengths of 2^{20}, 2^{25}, and 2^{30} bits are applied with the tests. The results are shown in Table VI. As can be seen, all binary sequences generated by TCL-PRNG with different lengths can pass all subtests. This means that TCL-PRNG is able to generate pseudo-random numbers with high quality of randomness.

4) Diehard Statistical Test Suit: Diehard statistical suit [56] is one classical and powerful test suit that includes 15 statistical subtests for number sequences. It can satisfy the testing requirements for a big-size random sequence and have the ability to detect the defects. A total number of 234 *P*-values will be generated in the Diehard test. The testing sequence will be considered to be truly independent and random to pass the test if all these *P*-values are uniformly distributed in range of [0, 1). If a testing sequence obtains six or more *P*-values of 0 or 1, the sequence is considered to fail the test. The testing sequence is with size 11 468 800 byte.

In our test, a sequence of 11 468 800 byte is generated by the proposed TCL-PRNG and then measured by the Diehard statistical test. The 234 generated *P*-values are sorted in an increasing order and then plotted in Fig. 16. As can be seen, the red straight line is the uniform distribution line on [0, 1) and all sorted *P*-values are distributed on or nearby this red line. This means that the random sequence generated by TCL-PRNG is a truly random sequence that can pass the Diehard statistical test.

5) FIPS PUB 140-2 Test: FIPS PUB 140-2 test [57] is a practical and widely accepted test standard that was also developed by NIST to test the randomness of binary sequences. It uses 20 000-bit sequences as the input and includes four

TABLE VII
FIPS PUB 140-2 TEST RESULTS OF TCL-PRNG

Sub-tests		Accepted intervals	Test Results	
Monobit		(9725, 10275)	10002	
Poker		(2.16, 46.17)	12.8960	
Runs			All zeros	All ones
	Run $= 1$	(2315, 2685)	2505	2509
	Run $= 2$	(1114, 1386)	1216	1201
	Run $= 3$	(527, 723)	632	656
	Run $= 4$	(240, 384)	324	299
	Run $= 5$	(103, 209)	165	167
	Run ≥ 6	(103, 209)	147	158
Long run		[0, 0]	0	0

subtests, namely the Monobit, Poker, Runs, and Long run tests. The Monobit test counts the number of ones in the 20 000-bit sequence and the number must be in the range of (9725, 10275) to pass the test. The Poker test first divides the 20 000-bit sequence into 5000 continuous 4-bit segments. A 4-bit segment can be 16 possible values and $f(i)$ is used to represent the number of i $(0 \leq i \leq 15)$ in the 5000 continuous 4-bit segments. An evaluation value W can be calculated as

$$W = \frac{16}{5000}\left(\sum_{i=0}^{15}[f(i)]^2\right) - 5000$$

W within (2.16, 46.17) is considered to pass the test. A run is defined as the maximum number of continuous bits of all ones or zeros in the 20 000-bit sequence. The Runs test considers that different lengths of runs for either continuous ones or zeros should be within the corresponding intervals specified in the third column of Table VII. The Long run test thinks that a run of length 26 or more for the continuous zeros or ones should be 0.

Table VII shows the test results of the pseudo-random numbers generated by TCL-PRNG. As can be see, all the test results are within the corresponding accepted intervals to pass the tests.

VII. CONCLUSION

This paper has introduced a general framework of producing 1-D chaotic maps called DPCCS. It uses a chaotic map (control map) to dynamically control the parameter of another chaotic map (seed map). The seed map generates output chaotic values with a fixed or dynamic parameter setting in each iteration. Any chaotic map can be used as the control/seed map (or both) in DPCCS. Using different chaotic maps as the control and seed maps, DPCCS is able to yield a completely different chaotic map. To demonstrate the feasibility of DPCCS, we have provided nine chaotic maps generated by DPCCS. The performance analysis and comparison results have shown that these nine chaotic maps are sensitive to their initial states, and they have wider chaotic ranges and more complex chaotic behaviors, and are more unpredictable than their seed maps.

To show the simplicity of the proposed DPCCS in hardware implementation, we have introduced an FPGA implementation of the TCL map. To show the applications of DPCCS, we have introduced a TCL-PRNG. Performance evaluations have been conducted using the NIST SP800-22, TestU01, Diehard statistical, and FIPS PUB 140-2 tests. The experiment results have shown that the proposed TCL-PRNG has high quality of randomness.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valued comments and suggestions, which greatly helped to improve the quality of this paper.

REFERENCES

[1] E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.*, vol. 20, no. 2, pp. 130–141, 1963.
[2] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential Equations, Dynamical Systems, and an Introduction to Chaos*, vol. 60. Amsterdam, The Netherlands: Academic, 2004.
[3] M. R. Frank, L. Mitchell, P. S. Dodds, and C. M. Danforth, "Standing swells surveyed showing surprisingly stable solutions for the Lorenz'96 model," *Int. J. Bifurcat. Chaos*, vol. 24, no. 10, 2014, Art. ID 1430027-14.
[4] H.-G. Chou, C.-F. Chuang, W.-J. Wang, and J.-C. Lin, "A fuzzy-model-based chaotic synchronization and its implementation on a secure communication system," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 2177–2185, Dec. 2013.
[5] S.-L. Chen, T. T. Hwang, and W.-W. Lin, "Randomness enhancement using digitalized modified logistic map," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 12, pp. 996–1000, Dec. 2010.
[6] Y. Wu, Z. Hua, and Y. Zhou, "n-dimensional discrete cat map generation using Laplace expansions," *IEEE Trans. Cybern.*, 2015, to be published.
[7] X. Wang, X. Wang, J. Zhao, and Z. Zhang, "Chaotic encryption algorithm based on alternant of stream cipher and block cipher," *Nonlin. Dyn.*, vol. 63, no. 4, pp. 587–597, 2011.
[8] Y. Zhou, Z. Hua, C.-M. Pun, and C. L. P. Chen, "Cascade chaotic system with applications," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2001–2012, Sep. 2015.
[9] K.-W. Wong, Q. Lin, and J. Chen, "Simultaneous arithmetic coding and encryption using chaotic maps," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 2, pp. 146–150, Feb. 2010.
[10] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D sine logistic modulation map for image encryption," *Inf. Sci.*, vol. 297, pp. 80–94, Mar. 2015.
[11] X. Wang, L. Teng, and X. Qin, "A novel colour image encryption algorithm based on chaos," *Signal Process.*, vol. 92, no. 4, pp. 1101–1108, 2012.
[12] Z.-P. Wang and H.-N. Wu, "On fuzzy sampled-data control of chaotic systems via a time-dependent Lyapunov functional approach," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 819–829, Apr. 2015.
[13] C. Shao *et al.*, "Recovering chaotic properties from small data," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2545–2556, Dec. 2014.
[14] H. Zhang, D. Liu, and Z. Wang, *Controlling Chaos: Suppression, Synchronization and Chaotification*. London, U.K.: Springer, 2009.
[15] H. Zhang, T. Ma, G.-B. Huang, and C. Wang, "Robust global exponential synchronization of uncertain chaotic delayed neural networks via dual-stage impulsive control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 3, pp. 831–844, Jun. 2010.
[16] J. Kalomiros, C. Hilas, and S. G. Stavrinides, "Chaotic synchronization of a secure system based on one-dimensional iterated maps," in *Proc. 4th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, 2015. [Online]. Available: http://mocast.physics.auth.gr/images/Papers/PAPER_40F.pdf
[17] A. Dwivedi, A. K. Mittal, and S. Dwivedi, "Adaptive synchronisation of diffusionless Lorenz systems and secure communication of digital signals by parameter modulation," *IET Commun.*, vol. 6, no. 13, pp. 2016–2026, Sep. 2012.
[18] K.-Y. Lian, T.-S. Chiang, C.-S. Chiu, and P. Liu, "Synthesis of fuzzy model-based designs to synchronization and secure communications for chaotic systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 1, pp. 66–83, Feb. 2001.
[19] S. G. Stavrinides, N. F. Karagiorgos, K. Papathanasiou, S. Nikolaidis, and A. N. Anagnostopoulos, "A digital nonautonomous chaotic oscillator suitable for information transmission," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 12, pp. 887–891, Dec. 2013.

[20] S. Ergün, "A high-speed truly random number generator based on an autonomous chaotic oscillator," in *Proc. IEEE Asia Pac. Conf. Circuits Syst. (APCCAS)*, Ishigaki, Japan, 2014, pp. 217–220.
[21] S. Ergün and S. Özoğuz, "Truly random number generators based on a non-autonomous chaotic oscillator," *AEU Int. J. Electron. Commun.*, vol. 61, no. 4, pp. 235–242, 2007.
[22] A. N. Srivastava and S. Das, "Detection and prognostics on low-dimensional systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 1, pp. 44–54, Jan. 2009.
[23] X. Wu, H. Hu, and B. Zhang, "Parameter estimation only from the symbolic sequences generated by chaos system," *Chaos Soliton. Fract.*, vol. 22, no. 2, pp. 359–366, 2004.
[24] C. Ling, X. Wu, and S. Sun, "A general efficient method for chaotic signal estimation," *IEEE Trans. Signal Process.*, vol. 47, no. 5, pp. 1424–1428, May 1999.
[25] Z. Zhu and H. Leung, "Identification of linear systems driven by chaotic signals using nonlinear prediction," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 2, pp. 170–180, Feb. 2002.
[26] D. Arroyo, R. Rhouma, G. Alvarez, S. Li, and V. Fernandez, "On the security of a new image encryption scheme based on chaotic map lattices," *Chaos Interdiscipl. J. Nonlin. Sci.*, vol. 18, no. 3, 2008, Art. ID 033112.
[27] G. Álvarez, F. Montoya, M. Romera, and G. Pastor, "Cryptanalysis of an ergodic chaotic cipher," *Phys. Lett. A*, vol. 311, nos. 2–3, pp. 172–179, 2003.
[28] A. Skrobek, "Cryptanalysis of chaotic stream cipher," *Phys. Lett. A*, vol. 363, nos. 1–2, pp. 84–90, 2007.
[29] T. Yang, L.-B. Yang, and C.-M. Yang, "Cryptanalyzing chaotic secure communications using return maps," *Phys. Lett. A*, vol. 245, no. 6, pp. 495–510, 1998.
[30] V. Patidar, K. K. Sud, and N. K. Pareek, "A pseudo random bit generator based on chaotic logistic map and its statistical testing," *Informatica (Slovenia)*, vol. 33, no. 4, pp. 441–452, 2009.
[31] H. Hu, Y. Xu, and Z. Zhu, "A method of improving the properties of digital chaotic system," *Chaos Soliton. Fract.*, vol. 38, no. 2, pp. 439–446, 2008.
[32] K. J. Persohn and R. J. Povinelli, "Analyzing logistic map pseudorandom number generators for periodicity induced by finite precision floating-point representation," *Chaos Soliton. Fract.*, vol. 45, no. 3, pp. 238–245, 2012.
[33] J. Lü, X. Yu, and G. Chen, "Generating chaotic attractors with multiple merged basins of attraction: A switching piecewise-linear control approach," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 2, pp. 198–207, Feb. 2003.
[34] R. Bose and S. Pathak, "A novel compression and encryption scheme using variable model arithmetic coding and coupled chaotic system," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 4, pp. 848–857, Apr. 2006.
[35] C. Shen, S. Yu, J. Lü, and G. Chen, "A systematic methodology for constructing hyperchaotic systems with multiple positive Lyapunov exponents and circuit implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 3, pp. 854–864, Mar. 2014.
[36] S. Ozoguz, A. S. Elwakil, and S. Ergun, "Cross-coupled chaotic oscillators and application to random bit generation," *IEE Proc. Circuits Devices Syst.*, vol. 153, no. 5, pp. 506–510, Oct. 2006.
[37] Z. Duan, J.-Z. Wang, and L. Huang, "Input and output coupled nonlinear systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 3, pp. 567–575, Mar. 2005.
[38] J. C. Sprott, "A new chaotic jerk circuit," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 4, pp. 240–243, Apr. 2011.
[39] Y. Wu, Y. Zhou, and L. Bao, "Discrete wheel-switching chaotic system and applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 12, pp. 3469–3477, Dec. 2014.
[40] A. Prasad, "Amplitude death in coupled chaotic oscillators," *Phys. Rev. E*, vol. 72, no. 5, 2005, Art. ID 056204.
[41] S.-L. Bu and I.-M. Jiang, "Estimating the degree distribution in coupled chaotic oscillator networks," *Europhys. Lett.*, vol. 82, no. 6, 2008, Art. ID 68001.
[42] B. Zhang, S. Xu, Y. Li, and Y. Chu, "On global exponential stability of high-order neural networks with time-varying delays," *Phys. Lett. A*, vol. 366, nos. 1–2, pp. 69–78, 2007.
[43] C. Shen, S. Yu, J. Lü, and G. Chen, "Designing hyperchaotic systems with any desired number of positive Lyapunov exponents via a simple model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 8, pp. 2380–2389, Aug. 2014.
[44] C. Werndl, "What are the new implications of chaos for unpredictability?" *Brit. J. Philos. Sci.*, vol. 60, no. 1, pp. 195–220, 2009.
[45] Wikipedia. *Attractor*. [Online]. Available: http://en.wikipedia.org/wiki/Attractor, accessed Jul. 15, 2015.
[46] P. Grassberger and I. Procaccia, "Estimation of the Kolmogorov entropy from a chaotic signal," *Phys. Rev. A*, vol. 28, no. 4, pp. 2591–2593, 1983.
[47] A. M. Albano, J. Muench, C. Schwartz, A. I. Mees, and P. E. Rapp, "Singular-value decomposition and the Grassberger–Procaccia algorithm," *Phys. Rev. A*, vol. 38, no. 6, pp. 3017–3026, 1988.
[48] P. Grassberger and I. Procaccia, "Characterization of strange attractors," *Phys. Rev. Lett.*, vol. 50, no. 5, pp. 346–349, 1983.
[49] C. Guyeux, Q. Wang, and J. M. Bahi, "Improving random number generators by chaotic iterations application in data hiding," in *Proc. Int.Conf. Comput. Appl. Syst. Model. (ICCASM)*, vol. 13. Taiyuan, China, 2010, pp. V13-643–V13-647.
[50] X. Fang *et al.*, "Noise and chaos contributions in fast random bit sequence generated from broadband optoelectronic entropy sources," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 3, pp. 888–901, Mar. 2014.
[51] X. Li, A. B. Cohen, T. E. Murphy, and R. Roy, "Scalable parallel physical random number generator based on a superluminescent LED," *Opt. Lett.*, vol. 36, no. 6, pp. 1020–1022, 2011.
[52] L. Cao, L. Min, and H. Zang, "A chaos-based pseudorandom number generator and performance analysis," in *Proc. Int. Conf. Comput. Intell. Security*, vol. 1. Beijing, China, 2009, pp. 494–498.
[53] B. Schneier, *Applied Cryptography*. New York, NY, USA: Wiley, 2004.
[54] L. E. Bassham, III, *et al.*, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Inf. Technol. Lab., Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Tech. Rep. SP 800-22 Rev. 1a., 2010.
[55] P. L'Ecuyer and R. Simard, "TestU01: A C library for empirical testing of random number generators," *ACM Trans. Math. Soft.*, vol. 33, no. 4, 2007, Art. ID 22.
[56] G. Marsaglia. *Diehard Statistical Tests*. [Online]. Available: http://stat.fsu.edu/pub/diehard/, accessed Jul. 15, 2015.
[57] *Security Requirements for Cryptographic Modules*, NIST Standard FIPS PUB 140-2, 2001.

Zhongyun Hua (S'14) received the B.S. degree from Chongqing University, Chongqing, China, and the M.S. degree from the University of Macau, Macau, China, in 2011 and 2013, both in software engineering. He is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, University of Macau.

His current research interests include chaos-based applications, multimedia security, and signal/image processing.

Yicong Zhou (M'07–SM'14) received the B.S. degree from Hunan University, Changsha, China, in 1992, and the M.S. and Ph.D. degrees from Tufts University, Medford, MA, USA, in 2008 and 2010, respectively, all in electrical engineering.

He is currently an Assistant Professor with the Department of Computer and Information Science, University of Macau, Macau, China. He has authored/co-authored over 90 papers, including 14 IEEE TRANSACTION papers, six most downloaded/popular papers in corresponded journals, and one highly cited paper within the top 1% of published papers in the Information Sciences Institute database up to 2015. His current research interests include chaotic systems, multimedia security, image processing and understanding, and machine learning.

Dr. Zhou was a recipient of the Third Prize of Macau Natural Science Award in 2014. He is a member of the International Society for Optical Engineers and the Association for Computing Machinery.