Discrete Wheel-Switching Chaotic System and Applications

Yue Wu, *Student Member, IEEE*, Yicong Zhou, *Member, IEEE*, and Long Bao, *Student Member, IEEE*

***Abstract*—This paper introduces a discrete wheel-switching chaotic system (DWSCS). Changing the controlling sequence of the wheel switch allows DWSCS to generate a large number of new chaotic sequences from a set of existing chaotic maps (seed maps). Simulations and analysis demonstrate the DWSCS's characteristics and chaotic behaviors. An FPGA design of DWSCS shows its effectiveness in hardware implementation. We also propose a pseudo-random number generator using DWSCS whose excellent performance has been shown in experiments and comparisons.**

***Index Terms*—Chaotic system, FPGA, pseudo-random number generator.**

I. Introduction

OVER THE PAST decades, many research interests have been put in the study of nonlinear dynamical systems [1]–[5]. Chaotic behaviors have been observed in a variety of systems including electrical circuits, lasers, oscillating chemical reactions, and fluid dynamics. Applications of chaos in economics [6], finance [7] and engineering [8]–[13] offer several opportunities of improvement. Especially, in electrical engineering, the chaotic system has been used in communications [14]–[16], random number generators [17], and encryption systems [18]–[22].

Chaotic systems are well-known for their random-like behaviors and high sensitivity to initial values. For a chaotic system, any two close sets of initial values generate random-like trajectories which diverge markedly. However, little information of the system may be sufficient to deduce the distribution of its output states [23]. For example, the artificial neural network has been demonstrated to identify chaotic systems [24], [25]. Other efforts are found in estimating parameters and initial values in existing chaotic maps [26]–[28]. These weaknesses has great impact of its applications in pseudo-random generators and encryption systems. For the chaos-based pseudo-random generator, its performance mainly depends on the chaotic properties of utilized chaotic maps/systems. Some existing chaotic maps can be attacked by some methods. This means that the corresponding pseudo-random generator is not desired. Designing good chaotic systems and generators becomes necessary.

For image encryption, chaotic ciphers for digital data encryption are claimed to be secure, but many of them are actually not [29]. The reason is that the utilized chaotic system is easy to be attacked. Due to a number of tools available for identifying and estimating chaotic systems, an attacker might significantly reduce the complexity of finding the encryption keys of a chaos based cipher. Recently, researches in cryptanalysis have already found approaches to break analog chaos-based secure communications [30] and chaotic ciphers [31], [32]. Many attacks rely on the knowledge of the chaotic system, i.e. the chaotic map and encryption mechanism. A good encryption method, however, should not rely on its structures [33]. In order to solve this dilemma, many high dimensional (three dimensional and four dimensional) chaotic systems were developed [34], [35]. Though more complicated chaotic systems sets more difficulties for attackers, they also asks for much more computational costs. Meanwhile, the security of a such high dimensional chaotic system might lose its invulnerability when faster computers are available. Hence, designing new chaotic systems with simple structures and good performance is vital to the chaos-based ciphers.

In this paper, we introduce a new chaotic system using a wheel switch. The system uses a set of predetermined chaotic maps as seeds to generate new chaotic sequences according to the controlling sequence of the wheel switch. Changing the controlling sequence yields different new chaotic sequences. The proposed system provides an alternative solution to the dilemma encountered in chaotic ciphers. It ensures security using the controlling sequence, which is unknown to attackers. We implements the system in the FPGA (field-programmable gate array) platform. The simulation results show that the generated chaotic sequences have more advanced structures and complicated chaotic behaviors than its seed chaotic maps. To investigate its applications, this paper also proposes a new pseudo-random generator. Its application for image encryption has been discussed in our previous work in [36].

The remainder of the paper is organized as follows: Section II reviews three traditional discrete chaotic maps; Section III introduces the discrete wheel-switching chaotic system (DWSCS); Section IV provides discussions of its chaotic property and a case study of its behaviors; Section V shows the FPGA design of a specific wheel-switching chaotic system. To investigate the applications of DWSCS, we use it for designing a pseudo-random number generator in Section VI. Finally, Section VII reaches a conclusion and discusses our future directions.

Manuscript received January 27, 2014; revised May 01, 2014, June 12, 2014, and June 24, 2014; accepted June 29, 2014. Date of publication August 11, 2014; date of current version November 21, 2014. This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/017/2012/A1 and by the Research Committee at University of Macau under Grants MYRG2014-00003-FST, MRG017/ZYC/2014/FST, MYRG113(Y1-L3)-FST12-ZYC and MRG001/ZYC/2013/FST. This paper was recommended by Associate Editor M. Frasca. *(Corresponding author: Y. Zhou.)*

Y. Wu is with the Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155 USA (e-mail: ywu03@ece.tufts.edu).

Y. Zhou and L. Bao are with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@umac.mo; baolonghnu@gmail.com).

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/TCSI.2014.2336512

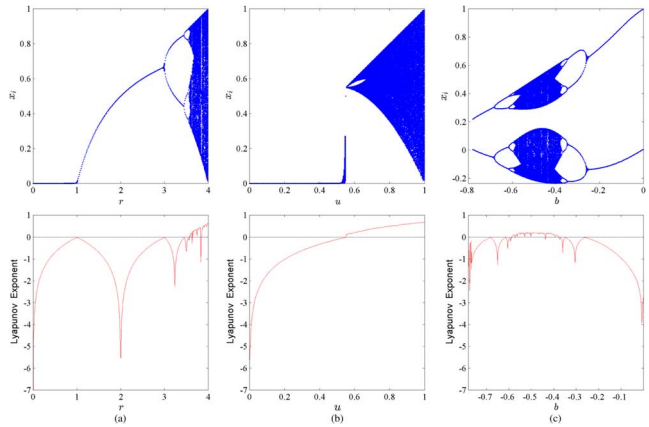


Fig. 1. The bifurcation diagrams and Lyapunov Exponent of the (a) Logistic map ($u \in [0, 4]$); (b) Skew Tent map ($c = 0.55$, and $u \in [0, 1]$); and (c) Gauss map ($a = -5$ and $b \in [-0.78, 0]$).

II. Background

We briefly review three chaotic maps that will be later used. For more details, one may refer to literatures [2], [37]–[39].

A. Logistic Map

The Logistic map $\mathbb{L}$ is a polynomial mapping of degree two introduced by a biologist Robert May in 1976 [2]. It is defined in (1), where $x_i \in [0, 1]$ and represents the population at year i and x_0 represents the initial population at year 0; r is a positive number and represents the combined rate for reproduction and starvation [37].

$$x_{i+1} = \mathbb{L}(x_i) = r x_i (1 - x_i) \tag{1}$$

The bifurcation diagram and Lyapunov Exponent of the Logistic map are given in Fig. 1(a). Its chaotic range is $r \in [3.57, 4]$ (approximately). This map is frequently used as an example of how complex chaotic behaviors can arise from a simple nonlinear dynamic equation.

B. Tent Map

Mathematically, the Skew Tent map $\mathbb{T}$ is a real value map defined by (2), where $\mathbb{1}$ is the indicator function that is of value 1 for $x \in I$ and of value 0 otherwise; parameter u controls the height of the map; and parameter $c \in [0, 1]$ controls the center of the map. When $c = 0.5$, it becomes a classic Tent map with chaotic behaviors for $u \in [0.5, 1]$ [38]. The bifurcation diagram and Lyapunov Exponent of the Skew Tent map are given in Fig. 1(b).

$$x_{i+1} = \mathbb{T}(x_i) = u \left(\mathbb{1}_{[0,c)} \frac{x_i}{c} + \mathbb{1}_{[c,1]} \frac{1 - x_i}{1 - c} \right) \tag{2}$$

C. Gauss Map

The Gauss map, also known as Gaussian map [39] or mouse map, is named after Carl Friedrich Gauss. It is a nonlinear iterated map defined by the Gaussian function in (3), where a and b are parameters controlling the width and height of the Gaussian curve, respectively.

$$x_{i+1} = \mathbb{G}(x_i) = \exp\left(-a x_i^2\right) + b \tag{3}$$

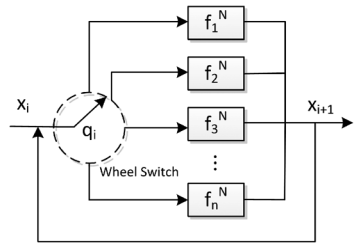


Fig. 2. The sketch diagram of DWSCS.

Due to two associated parameters, the Gauss map shows complicated dynamics. It includes all features that appear in the Logistic map and also contains its own features such as period undoublings and bistability [39]. Its chaotic behaviors are determined by parameters $a \in [0, 8]$, $b \in [-1, 1]$. The bifurcation diagram and Lyapunov Exponent of the Gauss map are given in Fig. 1(c).

III. Discrete Wheel-Switching Chaotic System

The new Discrete Wheel-Switching Chaotic System (DWSCS) consists of a controllable wheel switch and a set of normalized seed chaotic maps. Fig. 2 illustrates the DWSCS's sketch diagram [36]. In particular, DWSCS utilizes the wheel switch to select one of the normalized chaotic maps in each iteration. The DWSCS's output sequence is iteratively defined by (4), where x_i is the ith input of DWSCS $\mathbb{W}$; q_i is the ith element in the controlling sequence Q; and $f_{q_i}^{\mathbb{N}}$ denotes the q_ith normalized seed chaotic function.

$$x_{i+1} = \mathbb{W}(x_i) = f_{q_i}^{\mathbb{N}}(x_i) \tag{4}$$

As one may notice, DWSCS is fully determined by: (1) the set of normalized seed chaotic maps, i.e. $f_1^{\mathbb{N}}, f_2^{\mathbb{N}}, \ldots, f_n^{\mathbb{N}}$; and (2) the controlling sequence of the wheel switch, i.e. $Q = \{q_1, q_2, \ldots, q_i, \ldots, q_m\}$ with $q_i \in \{1, \cdots, n\}$. Although the seed chaotic maps can be chosen from the continuous chaotic functions, this paper focuses on the discrete chaotic functions similar to those presented in Section II.

The process for constructing a DWSCS can be done in three steps, namely: 1) initialization; 2) normalization; and 3) generation. Details of these three steps and the chaotic nature of a DWSCS are discussed below.

A. Initialization

Because these seed chaotic maps may be mismatched in dimensions, parameters and domains, the objective of the initialization stage is to coordinate the seed chaotic maps, $F = \{f_1, f_2, \ldots, f_n\}$, with a set of map parameters, $P_F = \{P_{f_1}, P_{f_2}, \ldots, P_{f_n}\}$, where f_i and P_{f_i} are the ith seed chaotic function and its parameter set, respectively.

There are two types of variables in the proposed DWSCS, namely the local and global variables. The local variable(s) will be used only within the associated seed maps. The global variable(s) will be used in throughout DWSCS in all iterations and passed from one seed map to another. It is worthy noting that DWSCS has at least one global variable to be passed through all seed chaotic maps.

DWSCS requires its parameters selected in the way to ensure that all seed maps have excellent chaotic behaviors. In this way,

even for the worst case of a controlling sequence, i.e. this sequence is not random-like at all, we still ensure a DWSCS is chaotic.

B. Normalization

The objective of the normalization stage is to ensure the domain and range of an initial seed map are both normalized into the unit interval [0,1]. In this way, no matter how we cascade these seed maps, there is no mismatching between one map's range and its succeeder's domain.

Without loss of generality, let $y = f(x)$ be a seed chaotic map with a bounded domain $\mathcal{D} = [d_{\min}, d_{\max}]$ and a bounded range $\mathcal{R} = [r_{\min}, r_{\max}]$, with real $d_{\min}, d_{\max}, r_{\min}$, and $r_{\max}$. Our aim is to find two normalization transforms $\mathbb{N}^{\mathcal{D}}$ for the f's domain and $\mathbb{N}^{\mathcal{R}}$ for the f's range, such that the resulting normalized function (5) has both its domain and range in [0,1],

$$\widetilde{y} = f^{\mathbb{N}}(\widetilde{x}) = \mathbb{N}^{\mathcal{R}} \circ f \circ \mathbb{N}^{\mathcal{D}}(\widetilde{x}) = \mathbb{N}^{\mathcal{R}}\left(f\left(\mathbb{N}^{\mathcal{D}}(\widetilde{x})\right)\right) \quad (5)$$

Though one may find $\mathbb{N}^{\mathcal{D}}$ and $\mathbb{N}^{\mathcal{R}}$ transforms in different means, we adopt the widely used *shifting and scaling* strategy. In particular, we find these two transforms by simply applying shifting and scaling functions as shown in (6) and (7).

$$x = \mathbb{N}^{D}(\widetilde{x}) = \widetilde{x}(d_{\max} - d_{\min}) + d_{\min} \quad (6)$$

$$\widetilde{y} = \mathbb{N}^{R}(y) = \frac{y - r_{\min}}{r_{\max} - r_{\min}} \quad (7)$$

In this way, we make sure that each seed map has been properly normalized to the unit interval. These normalized maps $F^{\mathbb{N}} = \{f_1^{\mathbb{N}}, f_2^{\mathbb{N}}, \ldots, f_n^{\mathbb{N}}\}$ then can be safely cascaded without any conflict.

C. Generation

As described in Fig. 2, at each iteration, DWSCS picks one normalized seed map $f_{q_i}^{\mathbb{N}}(\cdot)$, according to the controlling sequence at the ith moment. This implies that our controlling sequence has the same length as the output sequence. To improve efficiency, we actually repeatedly use a controlling sequence in a circular way. In other words, once we reach the tail of our controlling sequence, we restart at its head.

Let Q^* be a base controlling sequence with a length of l, we expand Q^* to a length m controlling sequence Q ($m \gg l$) by reusing Q^* in a periodic way, as defined by.

$$q_i = q^*_{\mathrm{mod}(i-1,l)+1} \quad (8)$$

where q_i is the ith controlling element in Q used in DWSCS, while q_j^* is the jth controlling element in Q^*.

For example, if $l = 3$ and $Q^* = [q_1^*, q_2^*, q_3^*]$, then $q_4 = q^*_{\mathrm{mod}(4-1,3)+1} = q_1^*$. The normalized seed map $f_{q_1}^{\mathbb{N}}(\cdot)$ will be selected in the 4th iteration. In this way, with a finite length controlling sequence Q^* we are able to produce an arbitrary length of the controlling sequence Q and a discrete chaotic sequence X.

D. Example

Suppose we want to construct a DWSCS by using the three mentioned chaotic maps: the Logistic map in (1), the Skew Tent map in (2), and the Gauss map in (3). In other words, we have these three seed maps as $f_1 = \mathbb{L}$, $f_2 = \mathbb{T}$ and $f_3 = \mathbb{G}$.

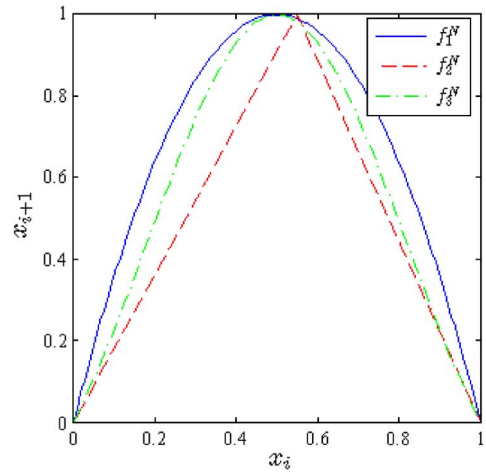


Fig. 3. Three examples of the normalized seed maps.

Because these seed maps are of one-dimensional map, our global variable for the resulting DWSCS is simply the variable in the first dimension of these seed maps. Next, we choose valid parameters of each seed map which ensure chaotic behaviors of the corresponding map. For instance, we choose $r = 4$ for the Logistic map, $u = 1$ and $c = 0.55$ for the Skew Tent map, and $a = -5$ and $b = -0.5$ for the Gauss map. As a result, we have

$$\begin{cases} f_1 : x_{i+1} = 4x_i(1 - x_i) \\ f_2 : x_{i+1} = \mathbb{1}_{[0,.55)}\frac{x_i}{0.55} + \mathbb{1}_{[.55,1]}\frac{1-x_i}{0.45} \\ f_3 : x_{i+1} = \exp\left(-5x_i^2\right) - 0.5 \end{cases} \quad (9)$$

It is noticeable that f_1 and f_2 are already in range of [0,1] for $x_i \in [0,1]$ and thus they are already normalized. However, f_3 is unlike the previous two maps. Though f_3 is valid for an arbitrary real x, we could only consider a fixed domain, say $x \in [-0.5, 0.5]$. As a result, we know its corresponding range is $[\exp(-5/4) - 0.5, 0.5]$. According to the normalization scheme introduced in (6) and (7), we have the normalized transforms for the domain and range of f_3 as given in (10) and (11), respectively.

$$x = \mathbb{N}^{D}(\widetilde{x}) = \widetilde{x} - 0.5 \quad (10)$$

$$\widetilde{y} = \mathbb{N}^{R}(y) = \frac{y + 0.5 - \exp(-5/4)}{1 - \exp(-5/4)} \quad (11)$$

Finally, we obtain three normalized maps as follows

$$\begin{cases} f_1^{\mathbb{N}} : x_{i+1} = 4x_i(1 - x_i) \\ f_2^{\mathbb{N}} : x_{i+1} = \mathbb{1}_{[0,.55)}\frac{x_i}{0.55} + \mathbb{1}_{[.55,1]}\frac{1-x_i}{0.45} \\ f_3^{\mathbb{N}} : x_{i+1} = \frac{\exp\left(-5(x_i-.5)^2\right) - \exp(-5/4)}{1 - \exp(-5/4)} \end{cases} \quad (12)$$

As one can see in Fig. 3, $F^{\mathbb{N}} = \{f_1^{\mathbb{N}}, f_2^{\mathbb{N}}, f_3^{\mathbb{N}}\}$ all have ranges and domains in [0,1]. We are now worry-free to construct a DWSCS with respect to an arbitrary controlling sequence Q.

IV. New Chaotic Maps Generated by DWSCS

In this section, we focus on the chaotic behaviors of a DWSCS. In particular, we first show that a DWSCS is chaotic as long as all of its seed maps are chaotic. We then investigate the chaotic behaviors of a DWSCS via a series of bifurcation diagrams of a DWSCS with different controlling sequences.

A. DWSCS is Chaotic

It is known that "*no universally accepted mathematical definition of the term 'chaos' exists*" [40]–[42]. We adopt a widely accepted working definition given in [43], which defines a chaotic system by using the Lyapunov exponent: "*Any system containing at least one positive Lyapunov exponent is defined to be chaotic, with the magnitude of the exponent reflecting the time scale on which system dynamics become unpredictable.*" The Lyapunov Exponents (LE) can be defined as follows.

Definition 1: The Lyapunov exponent for a discrete time system $x_{i+1} = \mathbb{H}(x_i; P_{\mathbb{H}})$ is the degree of divergence for the two trajectories starting at very close initial points as time goes to infinity. [40]

Mathematically, the LE of system $\mathbb{H}(\cdot)$, denoted as $\lambda_{\mathbb{H}}$, can be written as follows [40].

$$\lambda_{\mathbb{H}} = \lim_{k\to\infty} \left\{ \frac{1}{k} \ln \left| \frac{\mathbb{H}^{(k)}(x_0+\epsilon_0) - \mathbb{H}^{(k)}(x_0)}{\epsilon_0} \right| \right\} \tag{13}$$

where ϵ_0 denotes a very small positive quantity close to zero.

Intuitively, LE of a system reflects the average exponential rates of divergence of convergence of nearby orbits in space. When LE $\lambda_{\mathbb{H}} > 0$, it means that two trajectories starting at very close initial points diverge as time goes to infinity and it is a chaotic system. It is also well known that when $\mathbb{H}(\cdot)$ is a weakly differentiable function, then (13) has the alternative form

$$\lambda_{\mathbb{H}} = \lim_{k\to\infty} \left\{ \frac{1}{k} \sum_{i=0}^{k} \ln |\mathbb{H}'(x_i)| \right\} \tag{14}$$

It is demonstrable that a DWSCS system with all chaotic seed maps is also a chaotic system regardless of its controlling sequence. We scratch this demonstration by using mathematical induction. Let us start with a DWSCS with a base controlling sequence Q^* of length $l = 1$. It is clear that in this case the proposed DWSCS degrades to one of its chaotic seed maps and thus it is chaotic.

Now let us examine a DWSCS with a base controlling sequence of length $l = 2$. It is not difficult to see that the resulting DWSCS is equivalent to a system cascading two chaotic seed maps allowing duplicates.

Let $\mathbb{W}(x) = \mathbb{R} \circ \mathbb{S}(x)$ be such a DWSCS with both chaotic seed maps $\mathbb{R}(\cdot)$ and $\mathbb{S}(\cdot)$ weakly differentiable. When two seed chaotic maps $\mathbb{R}(\cdot)$ and $\mathbb{S}(\cdot)$ are identical, $\mathbb{W}(\cdot)$ is also identical to the map $\mathbb{R}(\cdot)$, and thus it is chaotic. When these two seed maps are distinctive, we have

$$\mathbb{W}'(x_i) = \mathbb{R}'\left(\mathbb{S}(x_i)\right) \mathbb{S}'(x_i) \tag{15}$$

After several steps of derivations as given in (16), one can see LE of the resulting DWSCS, i.e. $\lambda_{\mathbb{W}}$, is greater than zero because $\lambda_{\mathbb{R}} > 0$ and $\lambda_{\mathbb{S}} > 0$.

$$\begin{aligned}
\lambda_{\mathbb{W}} &= \lim_{k\to\infty} \left\{ \frac{1}{k} \sum_{i=0}^{k} \ln |\mathbb{W}'(x_i)| \right\} \\
&= \lim_{n\to\infty} \left\{ \frac{1}{k} \sum_{i=0}^{k} \ln \left|\mathbb{R}'(\mathbb{S}(x_i)\right| + \ln |\mathbb{S}'(x_i)| \right\} \\
&= \lim_{k\to\infty} \left\{ \frac{1}{k} \sum_{i=0}^{k} \ln \left|\mathbb{R}'(\mathbb{S}(x_i)\right| \right\} + \lim_{k\to\infty} \left\{ \frac{1}{k} \sum_{i=0}^{k} \ln |\mathbb{S}'(x_i)| \right\} \\
&= \lambda_{\mathbb{R}} + \lambda_{\mathbb{S}} > 0
\end{aligned} \tag{16}$$

Assume that a DWSCS with a controlling sequence of length $l = k$ is chaotic. It is not difficult to see that a DWSCS with a controlling sequence of length $l = k + 1$ is also chaotic. As we showed in the case $l = 2$, a DWSCS with $l = k + 1$ length controlling sequence is also equivalent to a system cascading $k + 1$ chaotic seed maps allowing duplicates. Considering the first k chaotic seed maps as a whole, the DWSCS with $l = k + 1$ length controlling sequence is nothing but the cascading of 1 new chaotic map and the chaotic system composed of k chaotic seed maps. Because both of these two parts are chaotic, the resulting DWSCS is also chaotic.

In this way, we show that a DWSCS with an arbitrary controlling sequence is also chaotic.

B. A Case Study of DWSCS Behaviors

In this section, we study the behaviors of a DWSCS with three seed maps, namely the Logistic, Skew Tent, and Gauss maps. An advantage of using these concrete maps is that we are now able to easily illustrate the DWSCS's behaviors.

To show different behaviors of a DWSCS, we use the method of bifurcation diagram [44], which depicts a system's behavior with respect to its continuous parameter. Though the only parameter we have in a DWSCS is its discrete controlling sequence, we are still able to plot the behavior of a DWSCS by imposing a continuous system parameter. Specifically, we define a system parameter p, and represent the parameter r in the Logistic map, parameter u in the Tent map and parameter b in the Gauss map as follows

$$\begin{cases} r = p \\ u = \frac{p}{4} \\ b = \frac{0.78(p-4)}{4} \end{cases} \tag{17}$$

In other words, we do not fix the parameters of the seed maps, but change them accordingly. It is worthwhile noting that we do this chaining step only to pretend a continuous DWSCS parameter, so that we can plot the system behaviors via bifurcation diagrams.

In summary, we observe several different types of behaviors of DWSCS according to Qs: DWSCS

1) with shift-equivalent Qs has similar dynamic behaviors.
2) with shift-inequivalent Qs has dissimilar dynamic behaviors.
3) with random Qs has more dynamic behaviors as the Q's length increases.

First, we say two controlling sequences Q^{a*} and Q^{b*} are shift-equivalent if and only if we have some integers k and l, such that

$$q_i^{a*} = q_{\mathrm{mod}(i+k,l)+1}^{b*} \tag{18}$$

for all controlling elements in Q^{a*} or Q^{b*}. For example, $Q^{a*} = [1, 2, 1]$ and $Q^{b*} = [2, 1, 1]$ is shift-equivalent, because there exist $k = 1$ and $l = 3$ to make (18) hold. However, $Q^{c*} = [2, 1, 2]$ is neither a shift-equivalent sequence of Q^{a*} nor Q^{b*}.

1) DWSCS With Shift-Equivalent Qs: Fig. 4 shows the results of shift-equivalent Qs of length 3. In particular, each row in Fig. 4 shows the bifurcation diagrams and Lyapunov Exponent plots of the DWSCS with shift-equivalent Qs. It is noticeable that both bifurcation diagrams and Lyapunov Exponent plots of these shift-equivalent Qs are quite similar to each other. However, Qs used in different rows are shift-inequivalent and lead DWSCS to different dynamic behaviors.

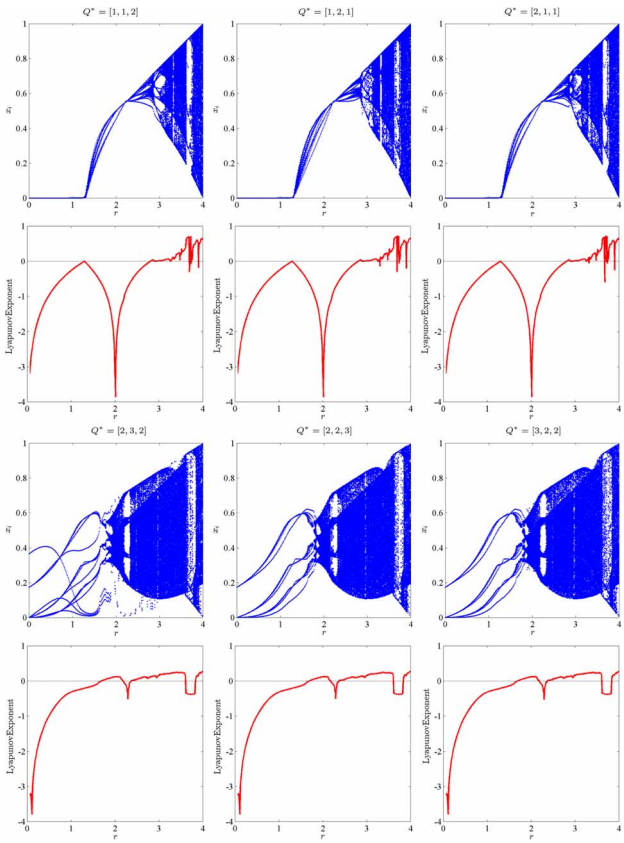


Fig. 4. DWSCS with shift-equivalent Q^*.

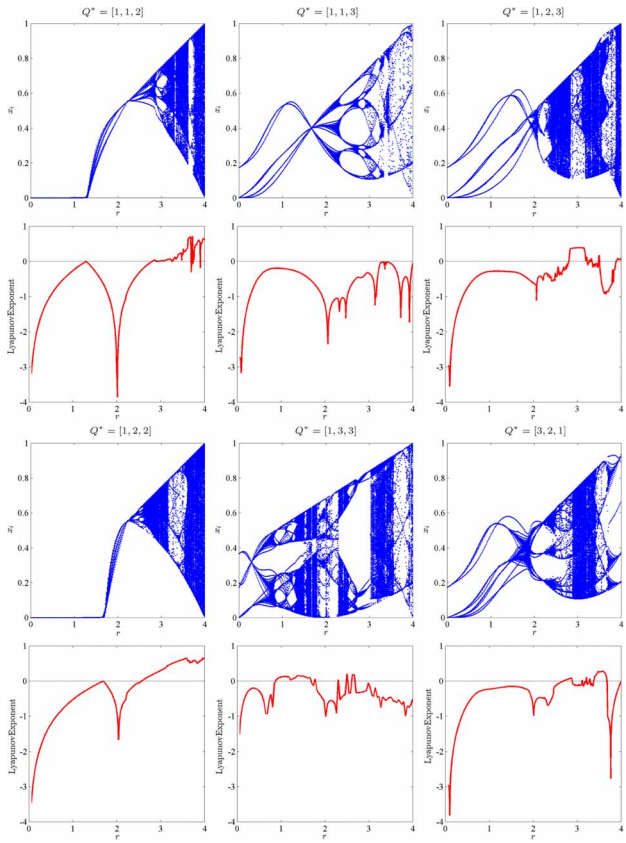


Fig. 5. DWSCS with shift-inequivalent Q^*.

2) DWSCS With Shift-Inequivalent Qs: Fig. 5 shows the bifurcation diagrams and Lyapunov Exponent plots of the DWSCS with shift-inequivalent Q^*s of length 3. As one can see, neither of two maps listed in Fig. 5 lead to similar dynamic behaviors. These results are not beyond our expectations. This is because we use a controlling sequence to repeatedly generate the outputs. When two controlling sequences are shift-equivalent, one may regard these two DWSCSs as an identical system but with different initial values. For example, let one DWSCS-I of $Q^{\mathrm{I}*} = [1, 1, 2]$ and another DWSCS-II of $Q^{\mathrm{II}*} = [1, 2, 1]$ both start with an initial value x_0. According to (18), we have $q^{\mathrm{I}}_{i+1} = q^{\mathrm{II}}_i$ for all $i \geqslant 1$. This means we will generate the identical output of DWSCS-I by feeding a new initial value $x'_0 = f_1^{\mathbb{N}}(x_0)$ in DWSCS-II.

3) DWSCS With Random Qs: Fig. 6 shows bifurcation diagrams of DWSCS with randomly generated Qs of different lengths. As one may see, as the length of the randomly generated Q increases, behaviors of a resulting DWSCS are more dynamic. One may notice that the bifurcation diagram of a DWSCS tends to a limit case as the length Q increases. In order to compare the bifurcation diagram of a DWSCS with a long Q, we put the bifurcation diagrams of chaotic seed maps together in Fig. 6(d). It is obvious that the resulting DWSCS with a long Q has a much wider dynamic range than the superposition of all three chaotic seed maps. This simply means that a DWSCS is effective to introduce additional dynamic behaviors.

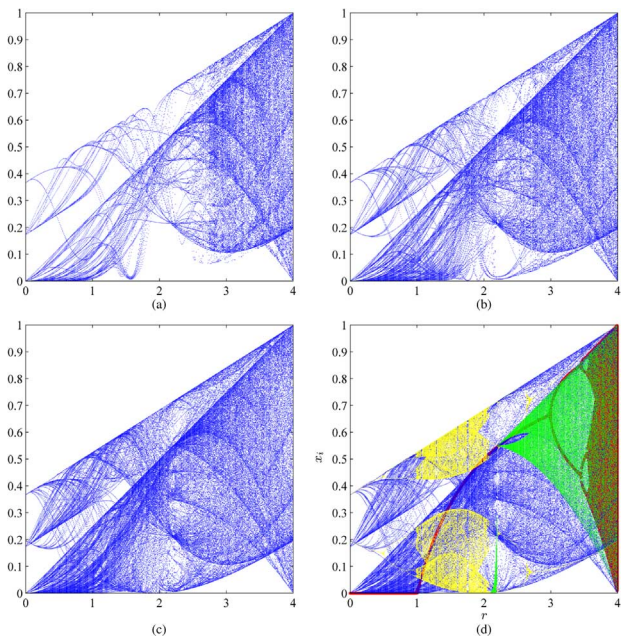


Fig. 6. DWSCS with random generated Qs of length: (a) 81, (b) 243, (c) 2187, and (d) 2187 (Red: Logistic map; Green: Tent map; Yellow: Gauss map; Blue: DWSCS), respectively.

V. FPGA Implementation of DWSCS

This section presents an FPGA implementation of DWSCS in the VHDL language. Again, we select the Logistic, Skew Tent, and Gauss maps as seed maps, but one may choose other chaotic maps instead.

A. FPGA Design

As shown in Fig. 7(a), the DWSCS circuit consists of the wheel switch and three normalized chaotic map (seed map) units, including the Logistic, Gauss, and Tent map units. x_i and

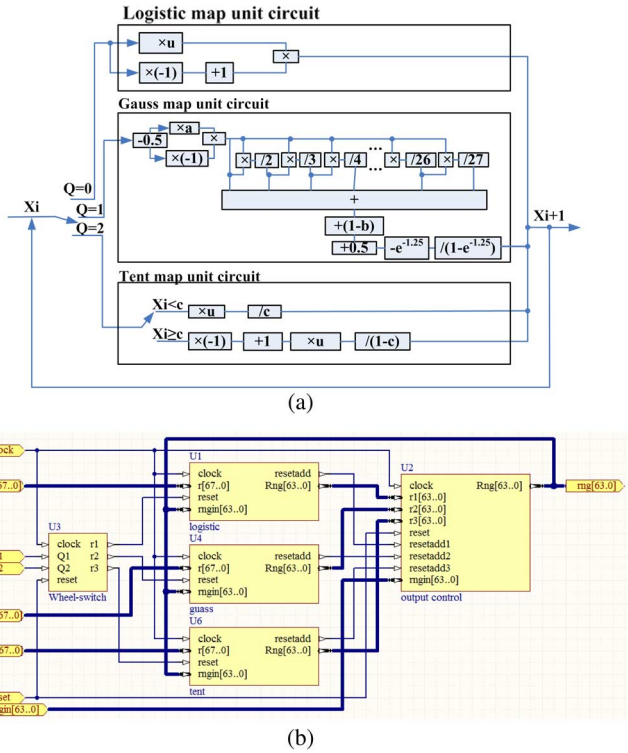


Fig. 7. The FPGA implementation of DWSCS: (a) The schematic diagram; (b) the FPGA structure.

x_{i+1} are the input and output of DWSCS. The wheel switch is used to select one of three seed maps in each iteration. The Logistic map unit is composed of the multiplication and addition circuits. The Tent map circuit contains the multiplication, addition and switch units. Its switch unit chooses different sub-circuits to produce the output x_{i+1} according to the comparison of x_i and c. The Gauss map unit is slightly different. In particular, we adopt the Taylor series to approximate e^x as defined by (3).

$$e^x \approx 1 + w_1 + w_2 + w_3 + \ldots\ldots + w_n \tag{19}$$

where,

$$w_n = \prod_{i=1}^{n} \frac{x}{i} = \left(\prod_{i=1}^{n-1} \frac{x}{i}\right) \frac{x}{n} = w_{n-1} \frac{x}{n}$$

Then, we can calculate w_n, recursively. This significantly reduces computational cost. Theoretically, the Taylor series could have an infinite length, n. We set $n = 27$ to balance the tradeoff between the execution time and approximation accuracy. Notice that one should normalize the Gauss map properly as we previously discussed in Section III-B.

B. Simulation Results

The FPGA implementation of DWSCS in the VHDL language is done with the Altium Designer software. The FPGA scheme structure is shown in Fig. 7(b). In this design, parameters of three seed maps $r[67\ldots0]$, $u[67\ldots0]$ and $a[67\ldots0]$ are represented by 68 bits. Combining two binary sequences $Q1$ and $Q2$ together can generate the control sequence Q. Here, all parameters are extended to the range of [0,4] based on (17). Fig. 8 plots the first 100 output points of two chaotic sequences that are generated by the circuit in Fig. 7(b) under different lengths of the base control sequence Q^*. These demonstrate that DWSCS is easy to implement in hardware.

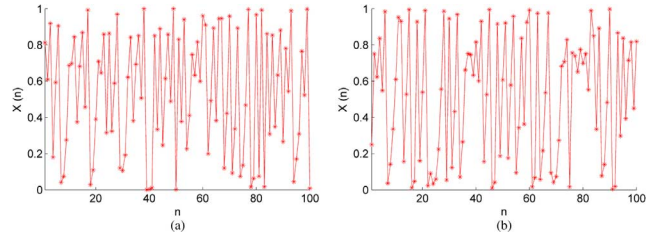


Fig. 8. The FPGA simulation results of DWSCS with the base control sequence Q^* of length (a) 3 and (b) 100, respectively.

TABLE I
15 SUBTESTS IN THE NIST SP 800-22 TEST

Subtest	Description
Frequency (Monobit)	percentage of zeros and ones in a sequence
Frequency within a Block	percentage of zeros and ones within an M-bit block
Runs	total number of runs (uninterrupted sequence of identical bits)
Longest-Run-of-Ones in a Block	the longest run of ones within an M-bit block
Binary Matrix Rank	rank of disjoint sub-matrices
Discrete Fourier Transform (Spectral)	peak heights after Discrete Fourier Transform
Non-overlapping Template Matching	probability of occurrences of pre-specified target strings
Overlapping Template Matching	probability of occurrences of pre-specific target strings
Maurer's "Universal Statistical"	number of bits between matching patterns
Linear Complexity	length of a linear feedback shift register
Serial	frequency of all possible overlapping M-bit patterns
Approximate Entropy	frequency of overlapping blocks of two adjacent lengths
Cumulative Sums (Cusums)	cumulative sum of adjusted $(-1, +1)$ digits
Random Excursions	number of cycles which have K visits in a cumulative sum random walk
Random Excursions Variant	total number of times that a particular state is visited in a cumulative sum random walk

VI. DWSCS-BASED PSEUDO-RANDOM NUMBER GENERATOR

Random numbers have an essential role in simulation and cryptography. Because a perfect random number is generated only by the nondeterministic physical phenomena, the pseudo-random numbers are often be used to approximate it in the real applications. Due to the close relationship between chaotic sequences and pseudo-random numbers, chaotic systems are the ideal candidates for the pseudo-random number generator (PRNG) [29]. This section proposes a new DWSCS-based pseudo-random number generator (DWSCS-PRNG) and evaluates its performance.

A. DWSCS-PRNG

The definition of DWSCS-PRNG is shown in (20).

$$s_i = \left\lfloor \left(x_i \times 10^{\lfloor x_i \times 10 \rfloor + 5} \right) \bmod 2 \right\rfloor \tag{20}$$

where, $S = \{s_1, s_2, \ldots, s_n\}$ is the pseudo-random number sequence, $X = \{x_1, x_2, \ldots, x_n\}$ is the output chaotic sequence of DWSCS in range of [0,1], $\lfloor \cdot \rfloor$ is the floor operator to obtain

TABLE II
THE COMPARISON RESULTS OF THE NIST SP 800-22 TEST

Test	New CI [46] $(m^n = y^n mod N)$	New CI [46] (no mark)	Old CI [47]	New CI [46] $(g_1())$	New CI [46] $(g_2())$	DWSCS-PRNG
Frequency (Monobit)	Pass	Pass	Pass	Pass	Pass	Pass
Frequency within a Block	Pass	**Fail**	**Fail**	Pass	Pass	Pass
Runs	Pass	Pass	Pass	Pass	Pass	Pass
Longest Run of Ones in a Block	Pass	Pass	Pass	Pass	Pass	Pass
Binary Matrix Rank	Pass	**Fail**	Pass	Pass	Pass	Pass
Discrete Fourier Transform (Spectral)	Pass	**Fail**	**Fail**	Pass	Pass	Pass
Non-overlapping Template Matching	Pass	Pass	Pass	Pass	Pass	Pass
Overlapping Template Matching	Pass	**Fail**	**Fail**	Pass	Pass	Pass
Maurer's "Universal Statistical"	Pass	Pass	Pass	Pass	Pass	Pass
Linear Complexity	Pass	Pass	Pass	Pass	Pass	Pass
Serial	Pass	**Fail**	Pass	Pass	Pass	Pass
Approximate Entropy	Pass	**Fail**	Pass	Pass	Pass	Pass
Cumulative Sums (Cusums)	Pass	**Fail**	**Fail**	Pass	Pass	Pass
Random Excursions	Pass	Pass	Pass	Pass	Pass	Pass
Random Excursion Variant	Pass	Pass	Pass	Pass	Pass	Pass
Success	15/15	8/15	11/15	15/15	15/15	15/15

the largest integer toward zero, and mod is the modula operation.

B. Statistical Analysis

To evaluate the quality of a DWSCS-PRNG, a convincing way is to perform the NIST SP 800-22 test published by the Information Technology Laboratory in National Institute of Standards and Technology in USA [45]. This test mainly assesses the quality of hardware or software based pseudo-random number generators in cryptographic applications which require a stronger randomness than other applications. In the software package of the NIST SP 800-22 test,[1] there are 15 subtests for evaluating different aspects of the random binary sequence. Table I provides the detail descriptions of these 15 tests.

In our experiment, 100 binary sequences with a size of 1 000 000 bits are generated by DWSCS-PRNG. We also compare DWSCS-PRNG with five existing methods in [46], [47]. They belong to a family of chaotic iteration (CI) based random generators proposed by Jacques Bahi. CI is a good iterative tool satisfying the topological chaotic property. It can be utilized to generate unpredictable pseudo-random numbers.

Table II shows the comparison results. As can be seen, two CI-based generators fail in all 15 subtests of the NIST SP 800-22 test while DWSCS-PRNG passes all of them. This proves the high quality of DWSCS-PRNG.

C. Security Analysis

Because DWSCS-PRNG has potential applications in the security fields, here we evaluate its security performance in terms of the brute-force attack, sensitivity test, and linear complexity.

1) Brute-Force Attack: The brute-force attack is a common attack in which one intends to guess the security key of an algorithm by exhaustively searching its key space. To withstand the brute-force attack, the algorithm should have a sufficiently large key space.

[1]The C code is located in http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html.

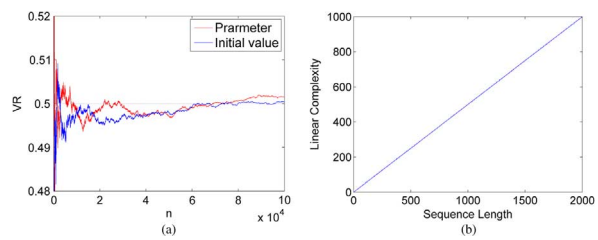


Fig. 9. Sensitivity and Linear complexity tests: (a) The sensitivity test with a tiny change (10^{-14}) in parameter (r) and initial value (x_0), respectively; (b) The linear complexity results.

The key space of DWSCS-PRNG is dependent on the length (L) of the controlling sequence and the number (N) of chaotic seed maps. Its key space is N^L, or equivalently $L\log_2(N)$ bits. For example, if DWSCS has 4 seed maps and its controlling sequence has a length of $L = 64$, its key space is 4^{64}, which is known to be large enough to resist the brute-force attack. Furthermore, one may include parameters of DWSCS, such as parameters and initial conditions of all seed maps in DWSCS, to further increase the key space of DWSCS-PRNG.

2) Sensitivity Test: A PSNG should be sensitive to its initial conditions. This ensures that any tiny change of the conditions yields a completely different random sequence. Here, we perform the sensitivity test to DWSCS-PRNG.

Fig. 9(a) shows the results of the sensitivity test of DWSCS-PRNG to its initial value and parameter, respectively. The x axis indicates the length (n) of the sequence. The y axis is the variance ratio (VR) defined in (21).

$$VR = \frac{\mathrm{Ham}(S_1, S_2)}{n} \tag{21}$$

where S_1 and S_2 are two binary sequences with the length of n generated by DWSCS-PRNG and $\mathrm{Ham}(\cdot)$ is a function to obtain the Hamming distance between two binary sequences.

From the results in Fig. 9(a), the variance ratio is close to or even larger than 0.5 after several iterations. This indicates that a tiny variation (10^{-14}) in the parameter or initial value(s) leads to a totally different pseudo-random number sequence. Thus,

DWSCS-PRNG is extremely sensitive to its initial value(s) and parameter(s).

3) Linear Complexity: Generally, attackers intend to utilize the linear feedback shift register (LFSR) to re-generate and analyze a pseudo-random sequence. To withstand this attack, a PRNG should ensure the attackers' difficulty of designing this LFSR. The linear complexity is the size in bits of LFSR and is introduced to evaluate the level of this design difficulty. the larger the value of the linear complexity is, the more difficulty of the LFSR design the attacker has. Fig. 9(b) plots the linear complexity values of a pseudo-random number sequence with length of 2000 bits which is generated by DWSCS-PRNG. Here, we use the Berlekamp-Massey algorithm [48] to calculate the values of the linear complexity. As one may observe, the curve of the linear complexity is extremely close to the ideal line $y = x$. This demonstrates that the DWSCS-PRNG has a high linear complexity. It is a sign of high security level of a PRNG [49].

VII. Conclusion

This paper has introduced a new wheel-switching chaotic system (DWSCS). Theoretically, any set of chaotic maps (called seed maps) can be used in the proposed DWSCS. It provides an easy and inexpensive way to produce new chaotic maps from existing chaotic maps. Using different parameters or seed maps, DWSCS is able to generate a large number of new chaotic maps. Even if the seed maps are determined and fixed, DWSCS can still produce distinctive chaotic maps using different wheel-switching control sequences. The newly generated maps may have longer periodic orbits than the corresponding seed maps. We have demonstrated that using only three simple seed maps allows DWSCS to generate new complicated and unpredictable chaotic maps.

To demonstrate that DWSCS is easy to implement in hardware, an FPGA design of DWSCS has been proposed. To investigate its applications, DWSCS has been used to design a pseudo-random number generator and an image security system. Simulations and analysis have proved that DWSCS shows excellent performance in both applications. The new pseudo-random number generator has passed all statistical tests in the NIST testing package.

Because different chaotic maps have different nonchaotic regions, an initial value may lead to a chaotic trajectory in one seed map but a periodic trajectory in another seed map. This is the reason why linking different seed maps properly improves the randomness of a trajectory. Alternatively, it is clear that changing map parameters, e.g. r in a Logistic map, may also lead to different nonchaotic regions. This is one of our future directions to explore.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valued comments and suggestions which helped to improve the manuscript.

References

[1] H. Dimassi and A. Loria, "Adaptive unknown-input observers-based synchronization of chaotic systems for telecommunication," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 4, pp. 800–812, 2011.

[2] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[3] H. Jia, Z. Chen, and G. Qi, "Chaotic characteristics analysis and circuit implementation for a fractional-order system," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 3, pp. 845–853, 2014.

[4] Y. Li, G. Chen, and W. Tang, "Controlling a unified chaotic system to hyperchaotic," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 4, pp. 204–207, 2005.

[5] S. Yu, J. Lu, G. Chen, and X. Yu, "Design and implementation of grid multiwing butterfly chaotic attractors from a piecewise lorenz system," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 10, pp. 803–807, 2010.

[6] C. Kyrtsou and W. C. Labys, "Detecting positive feedback in multivariate time series: The case of metal prices and US inflation," *Physica A, Stat. Mech. Its Appl.*, vol. 377, no. 1, pp. 227–229, 2007.

[7] C. Kyrtsou and M. Terraza, "Is it possible to study chaotic and arch behaviour jointly? Application of a noisy mackeycglass equation with heteroskedastic errors to the paris stock exchange returns series," *Comput. Econ.*, vol. 21, no. 3, pp. 257–276, 2003.

[8] S. Kozic and M. Hasler, "Low-density codes based on chaotic systems for simple encoding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 2, pp. 405–415, 2009.

[9] A. Loria and A. Zavala-Rio, "Adaptive tracking control of chaotic systems with applications to synchronization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 9, pp. 2019–2029, 2007.

[10] Y. Zhou, K. Panetta, S. Agaian, and C. L. P. Chen, "(n, k, p)-gray code for image systems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 515–529, 2013.

[11] X. Yi, "Hash function based on chaotic tent maps," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 6, pp. 354–357, 2005.

[12] K. Satish, T. Jayakar, C. Tobin, K. Madhavi, and K. Murali, "Chaos based spread spectrum image steganography," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 587–590, 2004.

[13] L. Wang, X. Min, and G. Chen, "Performance of simo fm-dcsk uwb system based on chaotic pulse cluster signals," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 9, pp. 2259–2268, 2011.

[14] M. Willsey, K. Cuomo, and A. Oppenheim, "Quasi-orthogonal wideband radar waveforms based on chaotic systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 3, pp. 1974–1984, 2011.

[15] O. Hugues-Salas and K. Shore, "An extended kalman filtering approach to nonlinear time-delay systems: Application to chaotic secure communications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 9, pp. 2520–2530, 2010.

[16] J. C. Pizolato, M. A. Romero, and L. Goncalves Neto, "Chaotic communication based on the particle-in-a-box electronic circuit," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 4, pp. 1108–1115, 2008.

[17] F. Pareschi, G. Setti, and R. Rovatti, "Implementation and testing of high-speed cmos true random number generators based on chaotic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 12, pp. 3124–3137, 2010.

[18] J. Chen, J. Zhou, and K.-W. Wong, "A modified chaos-based joint compression and encryption scheme," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 2, pp. 110–114, 2011.

[19] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, pp. 172–182, 2014.

[20] 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, 2010.

[21] W. Yue, Y. Zhou, S. Agaian, and J. P. Noonan, "A symmetric image cipher using wave perturbations," *Signal Process.*, vol. 102, pp. 122–131, 2014.

[22] Y. Zhou, W. Cao, and C. P. Chen, "Image encryption using binary bitplane," *Signal Process.*, vol. 100, pp. 197–207, 2014.

[23] 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, 2002.

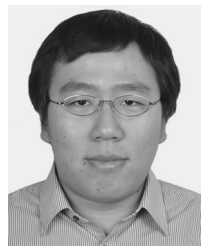
[24] G. Chen, Y. Chen, and H. Ogmen, "Identifying chaotic systems via a Wiener-type cascade model," *IEEE Control Syst. Mag.*, vol. 17, no. 5, pp. 29–36, 1997.

[25] M. Xu, G. Chen, and Y.-T. Tian, "Identifying chaotic systems using Wiener and Hammerstein cascade models," *Math. Comput. Model.*, vol. 33, no. 4–5, pp. 483–493, 2001.

[26] 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.

[27] 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, 1999.

[28] H. E. Papadopoulos and G. W. Wornell, "Maximum-likelihood estimation of a class of chaotic signals," *IEEE Trans. Inf. Theory*, vol. 41, no. 1, pp. 312–317, 1995.
[29] S. Li, X. Mou, and Y. Cai, *Pseudo-Random Bit Generator Based on Couple Chaotic Systems and Its Applications in Stream-Cipher Cryptography*, ser. Lecture Notes in Computer Science. Berlin/Heidelberg, Germany: Springer, 2001, vol. 2247, pp. 316–329.
[30] 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.
[31] G. Álvarez, F. Montoya, M. Romera, and G. Pastor, "Cryptanalysis of an ergodic chaotic cipher," *Phys. Lett. A*, vol. 311, no. 2–3, pp. 172–179, 2003.
[32] A. Skrobek, "Cryptanalysis of chaotic stream cipher," *Phys. Lett. A*, vol. 363, no. 1–2, pp. 84–90, 2007.
[33] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, 1949.
[34] T. Gao and Z. Chen, "A new image encryption algorithm based on hyper-chaos," *Phys. Lett. A*, vol. 372, no. 4, pp. 394–400, 2008.
[35] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons, Fractals*, vol. 21, no. 3, pp. 749–761, 2004.
[36] Y. Wu, J. P. Noonan, and S. Agaian, "A wheel-switch chaotic system for image encryption," in *Proc. 2011 Int. Conf. Syst. Sci. Eng. (ICSSE)*, pp. 23–27.
[37] J. Thompson and H. Stewart, *Nonlinear Dynamics and Chaos*. Hoboken, NJ, USA: Wiley, 2002.
[38] M. Hasler and Y. Maistrenko, "An introduction to the synchronization of chaotic systems: Coupled skew tent maps," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 44, no. 10, pp. 856–866, 1997.
[39] S. Lynch, *Dynamical Systems With Applications Using MAPLE*. New York: Springer, 2009.
[40] J. B. Dingwell, "Lyapunov exponents," in *Wiley Encyclopedia of Biomedical Engineering*. Hoboken, NJ, USA: Wiley, 2006.
[41] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Boulder, CO, USA: Westview Press, 2001.
[42] J. Banks, J. Brooks, G. Cairns, G. Davis, and P. Stacey, "On devaney's definition of chaos," *Amer. Math. Monthly*, vol. 99, no. 4, pp. 332–334, 1992.
[43] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining lyapunov exponents from a time series," *Physica D, Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, 1985.
[44] P. Glendinning, *Stability, Instability and Chaos*. Cambridge, U.K.: Cambridge Univ. Press, 1994.
[45] I. Lawrence, E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, and S. Vo, SP 800-22 rev. 1a. A Statistical Test Suite for Random and Pseudo-Random Number Generators for Cryptographic Applications, National Institute of Standards and Technology (NIST), 2010.
[46] J. Bahi, X. Fang, C. Guyeux, and Q. Wang, "Randomness quality of ci chaotic generators: Applications to internet security," in *Proc. 2010 Int. Conf. Evolving Internet (INTERNET)*, pp. 125–130.
[47] Q. Wang, C. Guyeux, and J. Bahi, "A novel pseudo-random number generator based on discrete chaotic iterations," in *Proc. 2009 Int. Conf. Evolving Internet (INTERNET)*, pp. 71–76.
[48] A. Canteaut, "Berlekamp-Massey algorithm," in *Encyclopedia of Cryptography and Security*. New York: Springer, 2011, p. 80.
[49] J. Gutierrez, I. E. Shparlinski, and A. Winterhof, "On the linear and nonlinear complexity profile of nonlinear pseudorandom number generators," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 60–64, 2003.

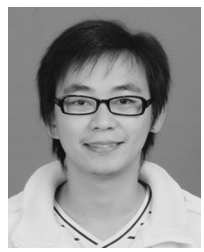


Yue Wu received his B.E. degree in telecommunication engineering from Huazhong University of Science and Technology, Wuhan, China, the M.S. degree in applied mathematics from the University of Toledo, Toledo, OH, USA, and the Ph.D. degree in electrical engineering from Tufts University, Medford, MA, USA.

Dr. Wu is currently a Scientist at Raytheon BBN Technologies, Cambridge, MA, USA. His research interests include signal processing, information security, and pattern recognition.

Yicong Zhou (M'07) received his B.S. degree from Hunan University, Changsha, China, and his M.S. and Ph.D. degrees from Tufts University, Medford, MA, USA, all degrees in electrical engineering.

Dr. Zhou is currently an Assistant Professor in the Department of Computer and Information Science at University of Macau, Macau, China. His research interests focus on multimedia security, image/signal processing, pattern recognition, and medical imaging.



Long Bao (S'12) received his B.S. and M.S. degrees from Hunan University, Changsha, China, both in electrical engineering.

Mr. Bao is currently a research Assistant in the Department of Computer and Information Science at University of Macau, Macau, China. His research interests are multimedia security and signal/image processing.