n-Dimensional Discrete Cat Map Generation Using Laplace Expansions

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

***Abstract*—Different from existing methods that use matrix multiplications and have high computation complexity, this paper proposes an efficient generation method of n-dimensional (nD) Cat maps using Laplace expansions. New parameters are also introduced to control the spatial configurations of the nD Cat matrix. Thus, the proposed method provides an efficient way to mix dynamics of all dimensions at one time. To investigate its implementations and applications, we further introduce a fast implementation algorithm of the proposed method with time complexity $O(n^4)$ and a pseudorandom number generator using the Cat map generated by the proposed method. The experimental results show that, compared with existing generation methods, the proposed method has a larger parameter space and simpler algorithm complexity, generates nD Cat matrices with a lower inner correlation, and thus yields more random and unpredictable outputs of nD Cat maps.**

***Index Terms*—Arnold's cat map, cryptography, Laplace expansion, n-dimensional (nD) Cat map.**

I. Introduction

As a type of dynamical systems closely related to natural processes, chaotic systems are quite sensitive to their initial conditions. This indicates that their trajectories are exactly decided by their initial states. Any small difference in their initial states yields significantly different outcomes after long time system evolution [1]–[3]. This is commonly referred as butterfly effect [4]. Examples of chaotic systems include the logistic map [5], Chua's circuit [6], and Chen–Lee system [7]. With significant properties of initial state sensitivity, ergodicity, and unpredictability, chaotic systems have been widely reported in various subjects including mathematics, physics, computer sciences, biology, engineering, economics, robotics, geology, neuron science, and chemistry [8]–[11]. Among all chaos-based applications, cryptography is the most popular one [12]–[18]. This is because many properties of chaotic systems can be found similar in cryptography [19], [20]. For example, the initial condition sensitivity and ergodicity of a chaotic system are analogous to the diffusion and confusion properties [21] in cryptography, and the deterministic dynamics in a chaotic system corresponds to the deterministic pseudorandomness in cryptography [22], [23].

Among various developed chaotic maps, the Cat map is a 2-D chaotic map named after Vlamimir Arnold [24], [25]. Except for the common properties of chaotic systems, the Cap map has many unique properties.

1) It is in the integer form and can be easily adapted to an arbitrary finite precision [24], which benefits to the difference between the precision infiniteness of chaotic systems and precision finiteness of cryptosystems [26].
2) It is invertible and its inverse is also in the integer form because its Cat matrix has determinant 1.
3) It is area preserving [24], indicating that the Cat map can be directly used as a permutation function, which is the fundamental building blocks for cryptography.
4) It is topologically transitive [27] and thus it is chaotic with high randomness.
5) It is an Anosov diffeomorphism and structurally stable [25], indicating that small perturbations in the system do not affect the qualitative behaviors of the map's trajectory, and thus the Cat map itself can resist a certain level of noise.

With these significant properties, the Cat map has been studied in both theory and practice and used in different subjects. Examples include the Cat map period distributions [28], [29], Cat map properties [30], Quantum Cat maps [31]–[34], Cat maps in other domains [35], and Cat map-based pseudorandom number generator (PRNG) [36], image encryption [24], and watermarking [18].

A Cat map usually contains a transformation matrix called the Cat matrix. Several Cat map generation methods have been developed to construct different Cat matrices such that the generated Cat maps have high randomness and large parameter spaces. According to the dimensions of the Cat matrices, these generation methods can be roughly classified into 2-D [24], [25], 3-D [24], [37]–[39], and n-dimensional (nD) Cat map generation methods [40]–[42]. A 2-D Cat matrix is a 2×2 matrix containing four elements, while a 3-D Cat matrix is a 3×3 matrix with nine elements. When being used in a cryptosystem, they commonly have a parameter space less than the required size to resist brute-force attacks [43], and thus they have to be used with additional components to ensure security [24]. On the other hand, an nD Cat map with a parametric nD Cat matrix seems to have a quite large

Manuscript received July 9, 2015; accepted September 21, 2015. Date of publication October 26, 2015; date of current version October 13, 2016. This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/017/2012/A1, and in part by the Research Committee at the University of Macau under Grant MYRG2014-00003-FST, Grant MRG017/ZYC/2014/FST, Grant MYRG113(Y1-L3)-FST12-ZYC, and Grant MRG001/ZYC/2013/FST. This paper was recommended by Associate Editor J. Cao. *(Corresponding author: Yicong Zhou.)*

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

Z. Hua and Y. Zhou 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.2015.2483621

parameter space. However, because of using a large number of matrix multiplications and dependency relationships among the elements in the Cat matrix, existing *n*D Cat map generation methods produce *n*D Cat matrices with highly correlated matrix elements, which may downgrade the security of cryptography applications [40]. A detailed literature review on these methods is given in Section II.

In this paper, we propose a new *n*D Cat map generation method using Laplace expansions to iteratively construct an *n*D Cat matrix. Its properties are discussed and theoretical analysis is provided. The main contributions of this paper can be summarized as follows.

1) To overcome the high computation complexity of existing methods that are rooted in matrix multiplications, we propose a new *n*D Cat matrix generation method that is the first time to use Laplace expansions to efficiently generate *n*D Cat matrices.
2) We further propose new parameters to control the spatial configurations of an *n*D Cat matrix. This provides an efficient way to mix dynamics of all dimensions at one time.
3) We introduce a new fast algorithm to implement the proposed method. It can reduce the time complexity from $O(n^5)$ to $O(n^4)$.
4) We provide theoretical analysis and comprehensive performance evaluations of the proposed method with respect to the parameter space, algorithm complexity, matrix element correlation, Shannon entropy, and Kolmogorov entropy. The comparison results show that the proposed method have a larger parameter space, less computation complexity, more independent, and random matrix elements and can generate outputs with better randomness and unpredictability than existing methods.
5) To investigate the real-world application of the proposed method, we introduce a new PRNG using the Cat maps generated by the proposed method. The randomness quality of the proposed PRNG is evaluated using two test standards.

The rest of this paper is organized as follows. Section II briefly reviews the existing Cat map generation methods, Section III introduces our *n*D Cat map generation method using Laplace expansions, Section IV analyzes the proposed method, Section V compares the proposed method with existing methods, Section VI proposes a new PRNG and evaluates its performance, and Section VII concludes this paper.

II. Existing Cat Map Generation Methods

This section briefly reviews the parametric Arnold's, 3-D, and *n*D Cat map generation methods, and discusses their properties.

All discrete Cat maps can be represented as a general form

$$Y(t+1) = \mathbf{C}_*^{n\mathrm{D}} Y(t) \bmod N \tag{1}$$

where $\mathbf{C}_*^{n\mathrm{D}}$ is an *n*D Cat matrix generated by the method $*$ and $|\mathbf{C}_*^{n\mathrm{D}}| = 1$, $Y(t) = [y_1(t), y_2(t), \ldots, y_n(t)]^T$, and $Y(t+1) = [y_1(t+1), y_2(t+1), \ldots, y_n(t+1)]^T$ are the input and output of the *n*D Cat map. N is the number of finite states. Based on the way how to construct the *n*D Cat matrix $\mathbf{C}_*^{n\mathrm{D}}$, there exist several Cat map generation methods.

A. Arnold's Cat Map Generation Method

The original Cat map called the Arnold's Cat map is a 2-D Cat map. Its Cat matrix $\mathbf{C}_{\mathsf{A}}^{2\text{-D}}$ is defined by [25]

$$\mathbf{C}_{\mathsf{A}}^{2\text{-D}} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}. \tag{2}$$

In practice, the Arnold's Cat matrix $\mathbf{C}_{\mathsf{A}}^{2\text{-D}}$ is commonly extended into a parametric form for cryptography applications [24] and defined as follows:

$$\mathbf{C}_{\mathsf{para.}}^{2\text{-D}} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \tag{3}$$

where a and b are integers. Obviously, $\left|\mathbf{C}_{\mathsf{para.}}^{2\text{-D}}\right| = 1$ and $\mathbf{C}_{\mathsf{para.}}^{2\text{-D}}$ is indeed a Cat matrix with area preserving, invertibility, and other properties.

B. 3-D Cat Map Generation Methods

To enhance the security level of the Cat-map-based cryptography and other applications, several 3-D Cat map generation methods were developed.

Lian *et al.* [37] extended $\mathbf{C}_{\mathsf{para.}}^{2\text{-D}}$ into the $x-z$ and $y-z$ planes, and then proposed a parametric 3-D Cat matrix $\mathbf{C}_{\mathsf{L}}^{3\text{-D}}$, as

$$\mathbf{C}_{\mathsf{L}}^{3\text{-D}} = \begin{bmatrix} 1 & 0 & a \\ bc & 1 & abc+c \\ bcd+b & d & abcd+ab+cd+1 \end{bmatrix} \tag{4}$$

where a, b, c, and d are integer parameters.

Chen *et al.* [24] constructed a parametric 3-D Cat matrix by firstly extending $\mathbf{C}_{\mathsf{para.}}^{2\text{-D}}$ with respect to the x, y, and z directions and then mixing all coupled Cat matrices using matrix multiplications. Consequently, their 3-D Cat matrix $\mathbf{C}_{\mathsf{C}}^{3\text{-D}}$ is the product of three extended Cat matrices of pairwise-coupled dynamics as

$$\begin{aligned} \mathbf{C}_{\mathsf{C}}^{3\text{-D}} &= \mathbf{C}_{\mathsf{C}}^{x}\mathbf{C}_{\mathsf{C}}^{y}\mathbf{C}_{\mathsf{C}}^{z} \\ &= \begin{bmatrix} a_x a_z b_y + 1 & a_z \\ b_z + a_x b_y (a_z b_z + 1) & a_z b_z + 1 \\ b_y (a_x b_x + 1) & b_x \end{bmatrix} \\ &\quad \begin{bmatrix} a_y + a_x a_z (a_y b_y + 1) \\ a_y b_z + a_x (a_y b_y + 1)(a_z b_z + 1) \\ (a_x b_x + 1)(a_y b_y + 1) \end{bmatrix} \end{aligned} \tag{5}$$

where a_x, b_x, a_y, b_y, a_z, and b_z are six integer control parameters.

Also based on $\mathbf{C}_{\mathsf{para.}}^{2\text{-D}}$, Liu *et al.* [38] introduced a parametric 3-D Cat matrix $\mathbf{C}_{\mathsf{U}}^{3\text{-D}}$ by mixing dynamics on all three dimensions into the extended dimension as

$$\mathbf{C}_{\mathsf{U}}^{3\text{-D}} = \begin{bmatrix} 1 & a & 0 \\ b & ab+1 & 0 \\ c & d & 1 \end{bmatrix} \tag{6}$$

where parameters a, b, c, and d are integers.

Recently, Pan and Li [39] also introduced their parametric 3-D Cat matrix adding two control parameters c and d, but in a formulation different from Lian *et al.*'s [37] in (4). In particular, Pan's 3-D Cat matrix $\mathbf{C}_{\mathsf{P}}^{\text{3-D}}$ is defined as follows:

$$\mathbf{C}_{\mathsf{P}}^{\text{3-D}} = \begin{bmatrix} 1 & a & c \\ b & ab+1 & bc \\ d & abcd & cd+1 \end{bmatrix} \tag{7}$$

where a, b, c, and d are integer control parameters.

C. nD Cat Map Generation Methods

Besides the efforts on generating 3-D Cat matrices, high-dimensional Cat map generation methods have also been explored.

Falcioni *et al.* [40] proposed their nD Cat matrix $\mathbf{C}_{\mathsf{F}}^{n\text{D}}$ by replacing parameters in $\mathbf{C}_{\mathsf{para.}}^{\text{2-D}}$ with matrix ones as defined

$$\mathbf{C}_{\mathsf{F}}^{n\text{D}} = \begin{bmatrix} \mathbf{I}_{m\times m} & \mathbf{P}_a \\ \mathbf{P}_b & \mathbf{P}_b\mathbf{P}_a + \mathbf{I}_{m\times m} \end{bmatrix} \tag{8}$$

where $\mathbf{I}$ is the mth order identity matrix with $1 \le m \le n$, and $\mathbf{P}_a$ and $\mathbf{P}_b$ are arbitrary parameter matrices of size $m \times (n-m)$ and $(n-m) \times m$, respectively. It is easy to verify that

$$\begin{aligned} \left|\mathbf{C}_{\mathsf{F}}^{n\text{D}}\right| &= |\mathbf{I}| \cdot \left|(\mathbf{P}_b\mathbf{P}_a + \mathbf{I}) - \mathbf{P}_b\mathbf{I}^{-1}\mathbf{P}_a\right| \\ &= |\mathbf{I}| \cdot |\mathbf{I}| \\ &= 1. \end{aligned} \tag{9}$$

When $m = 1$ and $n = 2$, this nD Cat matrix becomes the parametric 2-D Cat matrix $\mathbf{C}_{\mathsf{para.}}^{\text{2-D}}$ in (3).

Tang and Tang [41] proposed an nD Cat map generation method by coupling 2-D dynamics developed by Chen *et al.* [24]. It firstly couples dynamics in any two of the nDs, and then mixes all these coupled matrices together as

$$\mathbf{C}_{\mathsf{T}}^{n\text{D}} = \mathbf{C}_{\mathsf{T}_{1,2}}^{n\text{D}}\mathbf{C}_{\mathsf{T}_{1,3}}^{n\text{D}}\cdots\mathbf{C}_{\mathsf{T}_{1,n}}^{n\text{D}}\mathbf{C}_{\mathsf{T}_{2,3}}^{n\text{D}}\cdots\mathbf{C}_{\mathsf{T}_{n-1,n}}^{n\text{D}} \tag{10}$$

where each coupled Cat matrix $\mathbf{C}_{\mathsf{T}_{\mathsf{p,q}}}^{n\text{D}}$ $(q > p)$ is an identical matrix except for elements $\mathbf{C}_{\mathsf{T}_{\mathsf{p,q}}}^{n\text{D}}(p,q) = a_{pq}, \mathbf{C}_{\mathsf{T}_{\mathsf{p,q}}}^{n\text{D}}(q,p) = b_{pq}$ and $\mathbf{C}_{\mathsf{T}_{\mathsf{p,q}}}^{n\text{D}}(q,q) = a_{pq}b_{pq} + 1$. In total, $\mathbf{C}_{\mathsf{T}}^{n\text{D}}$ is a mixed matrix of $\binom{n}{2}$ nD pairwise-coupled Cat matrices. Since $|\mathbf{C}_{\mathsf{T}_{\mathsf{p,q}}}^{n\text{D}}| = 1$, then $|\mathbf{C}_{\mathsf{T}}^{n\text{D}}| = 1$.

Starting with $\mathbf{C}_{\mathsf{N}}^{\text{2-D}} = \mathbf{C}_{\mathsf{para.}}^{\text{2-D}}$, Nance [42] recursively constructed his nD Cat matrix $\mathbf{C}_{\mathsf{N}}^{n\text{D}}$ using matrix union, which is defines as

$$\mathbf{C}_{\mathsf{N}}^{n\text{D}} = \mathbf{B}_1\mathbf{B}_2\cdots\mathbf{B}_n = \prod_{p=1}^{n}\mathbf{B}_p \tag{11}$$

where the pth basis Cat matrix $\mathbf{B}_p$ $(1 \le p \le n)$ is obtained by expanding $\mathbf{C}_{\mathsf{N}}^{(n-1)\text{D}}$ with respect to the pth diagonal elements as

$$\mathbf{B}_p = \begin{bmatrix} \mathbf{C}_{\mathsf{N_I}}^{(n-1)\text{D}} & \mathbf{0}_{p\times 1} & \mathbf{C}_{\mathsf{N_{II}}}^{(n-1)\text{D}} \\ \mathbf{0}_{1\times p} & 1 & \mathbf{0}_{1\times(n-p-1)} \\ \mathbf{C}_{\mathsf{N_{III}}}^{(n-1)\text{D}} & \mathbf{0}_{(n-p-1)\times 1} & \mathbf{C}_{\mathsf{N_{IV}}}^{(n-1)\text{D}} \end{bmatrix}$$

where $\mathbf{0}$ denotes the zero matrix and $\mathbf{C}_{\mathsf{N_I}}^{(n-1)\text{D}}$, $\mathbf{C}_{\mathsf{N_{II}}}^{(n-1)\text{D}}$, $\mathbf{C}_{\mathsf{N_{III}}}^{(n-1)\text{D}}$, and $\mathbf{C}_{\mathsf{N_{IV}}}^{(n-1)\text{D}}$ are four submatrices of $\mathbf{C}_{\mathsf{N}}^{(n-1)\text{D}}$ partitioned with respect to the pth row and pth column, that is

$$\mathbf{C}_{\mathsf{N}}^{(n-1)\text{D}} = \begin{bmatrix} \mathbf{C}_{\mathsf{N_I}}^{(n-1)\text{D}} & \mathbf{C}_{\mathsf{N_{II}}}^{(n-1)\text{D}} \\ \mathbf{C}_{\mathsf{N_{III}}}^{(n-1)\text{D}} & \mathbf{C}_{\mathsf{N_{IV}}}^{(n-1)\text{D}} \end{bmatrix}.$$

Since all $|\mathbf{B}_p| = 1$, then $|\mathbf{C}_{\mathsf{N}}^{n\text{D}}| = 1$ and $\mathbf{C}_{\mathsf{N}}^{n\text{D}}$ is indeed an nD Cat matrix.

D. Discussion

When given the number of finite states, the trajectory of the Arnold's Cat map can be predicted. The parameter space of the parametric Arnold's Cat matrix $\mathbf{C}_{\mathsf{para.}}^{\text{2-D}}$ is obviously too small for cryptography applications. The 3-D Cat matrices generated by above-mentioned methods have the largest parameter space of M^6 (M is the number of possible values of each matrix parameter) attained by $\mathbf{C}_{\mathsf{C}}^{\text{3-D}}$. Taking $M = 256$ as an example, then $(256)^6 = 2^{48} \ll 2^{128}$, which is far smaller than the required size to resist brute-force attacks [43]. This implies that applications directly using these parametric 3-D Cat matrices are insecure. Consequently, these 3-D Cat matrices are commonly used with additional components [24].

The parametric nD Cat matrices have many matrix elements. However, because of many matrix multiplications involved in their generation methods, their computation complexity is high and their matrix elements are strongly correlated with each other. Correlated elements are difficultly coprime even with coprime parameters. For a Cat map system, coprime elements often result in a long period of outputs [28], which is desired for many applications [44]. Also, when used in cryptography applications, correlated elements might reduce the actual key space of the security system and thus it is vulnerable for the brute-force attacks [40]. This is because with correlated elements, the actual number of distinctive Cat matrices that can be generated could be less than the parameter space where parameters are commonly used as keys.

III. Proposed Method

This section presents the proposed nD Cat map generation method in detail. First, a theorem for the nD Cat matrix is introduced. Based on this theorem, we then give a lemma of extending a Cat matrix from $(n-1)$D to nD. Finally, we propose our method to generate nD Cat matrices.

A. Theorem for nD Cat Matrix

An nD Cat matrix is an $n \times n$ square matrix with the constraint that its determinant is 1. First of all, we introduce Theorem 1 that allows us to construct an nD Cat matrix conditionally.

Theorem 1: For an $n \times n$ square matrix $\mathbf{X}$, if an element X_{ij} with a cofactor $A_{ij}^X = 1$ satisfies the relation

$$X_{ij} = (-1)^{i+j}\left(1 - \sum_{k=1,k\neq i}^{n} X_{kj}(-1)^{k+j}A_{kj}^X\right) \tag{12}$$

for all other elements and their cofactors, then $|\mathbf{X}| = 1$ and $\mathbf{X}$ is an nD Cat matrix.

Appendix A provides a detailed proof of Theorem 1. From Theorem 1, if an element in an $n \times n$ square matrix has a cofactor of value 1 and the relation (12) is satisfied, the determinant of this square matrix is constant 1 and thus it is an *n*D Cat matrix.

B. Extending (n − 1)D Cat Matrix to nD

From Theorem 1, it can be known that when some dependency relationships are satisfied, an *n*D Cat matrix can be constructed. Lemma 1 gives a definition of extending an $(n-1)$D Cat matrix into an *n*D one.

Lemma 1: If an $n \times n$ matrix $\mathbf{X}$ extended from an $(n-1)$D Cat matrix $\mathbf{C}^{(n-1)\mathrm{D}}$ as the submatrix associated with the element X_{ij}, and its elements and submatrices satisfy the relation in (12), the $n \times n$ matrix $\mathbf{X}$ is an *n*D Cat matrix.

Proof: Since $\mathbf{C}^{(n-1)\mathrm{D}}$ is the submatrix associated with element X_{ij}, then

$$A_{ij}^{X} = \left|\mathbf{C}^{(n-1)\mathrm{D}}\right| = 1.$$

Because (12) is also satisfied, $\mathbf{X}$ is then an *n*D Cat matrix as Theorem 1 states. ■

Lemma 1 indicates that an *n*D Cat matrix $\mathbf{C}^{n\mathrm{D}}$ can be generated in a parametric way, where the parameters consist of a given $(n-1)$D Cat matrix $\mathbf{C}^{(n-1)\mathrm{D}}$, the location of the element X_{ij} associated with $\mathbf{C}^{(n-1)\mathrm{D}}$, and $2(n-1)$ additional matrix elements. Fig. 1 illustrates the procedures of extending an $(n-1)$D Cat matrix to *n*D. As can be seen, two random spatial parameters (i, j) divide the $(n-1)$D Cat matrix [Fig. 1(a)] into four parts: 1) $\mathbf{C}_{UL}^{(n-1)\mathrm{D}}$; 2) $\mathbf{C}_{UR}^{(n-1)\mathrm{D}}$; 3) $\mathbf{C}_{LL}^{(n-1)\mathrm{D}}$; and 4) $\mathbf{C}_{LR}^{(n-1)\mathrm{D}}$, and then add new elements on the *i*th row and *j*th column expect $\mathbf{C}_{i,j}^{n\mathrm{D}}$, whose value is computed using (12). The generated *n*D Cat matrix is shown in Fig. 1(d).

C. Proposed nD Cat Map Generation Method

Suppose 1-D Cat matrix $\mathbf{C}^{1\text{-D}} = [1]$[1], any Cat matrix with a desired dimension can be constructed using Lemma 1 and $\mathbf{C}^{1\text{-D}}$. The proposed *n*D Cat matrix generation method using Laplace expansions is defined as follows:

$$\mathbf{C}_{\mathsf{Ours}}^{n\mathrm{D}} = \mathbb{L}(\mathbf{I}, \mathbf{J}, \mathbf{P}, \mathbf{Q}) \tag{13}$$

where $\mathbf{I} = \{i_1, i_2, \ldots, i_{n-1}\}$ and $\mathbf{J} = \{j_1, j_2, \ldots, j_{n-1}\}$ are two integer lists and $i_k, j_k \in [1, k+1]$ for all $k \in [1, n-1]$; $\mathbf{P} = \{\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^{n-1}\}$ and $\mathbf{Q} = \{\mathbf{Q}^1, \mathbf{Q}^2, \ldots, \mathbf{Q}^{n-1}\}$ are two vector lists and $\mathbf{P}^k$ and $\mathbf{Q}^k$ are two k-length vectors for all $k \in [1, n-1]$. Algorithm 1 describes the detailed procedures of our proposed *n*D Cat map generation method. Fig. 2 shows an example of generating a 6-D Cat matrix. All dependent elements in each Cat matrix can be determined via (12). For example, $\mathbf{C}_{2,2}^{2\text{-D}}$ in Fig. 2(b) can be calculated by (14) and $\mathbf{C}_{2,2}^{3\text{-D}}$ in Fig. 2(c) can be calculate by (15)

$$C_{2,2}^{2\text{-D}} = ab + 1 \tag{14}$$

$$C_{2,2}^{3\text{-D}} = abcd + cd + ef - bce - adf + 1. \tag{15}$$

[1]We just use $\mathbf{C}^{1\text{-D}}$ as a dummy case for generating high-dimensional Cat matrices, because it involves no dynamics mixing.

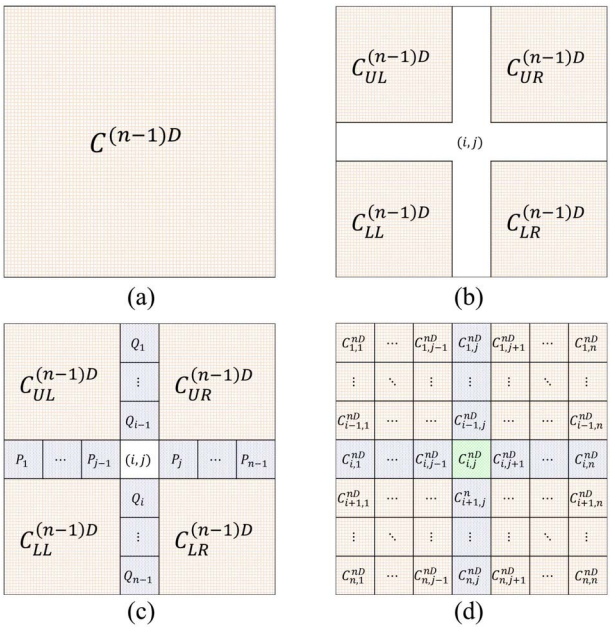

Fig. 1. Extending an $(n-1)$D Cat matrix to *n*D. (a) $(n-1)$D Cat matrix $\mathbf{C}^{(n-1)\mathrm{D}}$. (b) Extending matrix with respect to the element located at (i, j). (c) Adding new elements on the *i*th row or the *j*th column except $C_{i,j}^{n\mathrm{D}}$. (d) Computing the element $C_{i,j}^{n\mathrm{D}}$ using (12) and generating new *n*D Cat matrix $\mathbf{C}^{n\mathrm{D}}$.

In the rest of this paper, when generating an *n*D Cat matrix using our proposed method, $\mathbf{C}_{\mathsf{ours}^*}^{n\mathrm{D}}$ indicates that all parameters in $\mathbf{I}$ and $\mathbf{J}$ are restricted to be constant 2 (Fig. 2) and all parameters in $\mathbf{P}$ and $\mathbf{Q}$ are randomly determined, while $\mathbf{C}_{\mathsf{ours}}^{n\mathrm{D}}$ denotes that all parameters in $\mathbf{I}$, $\mathbf{J}$, $\mathbf{P}$, and $\mathbf{Q}$ are randomly determined.

IV. Method Discussion

Because most parameters to generate an *n*D Cat matrix using Laplace expansions are highly independent, users have great flexibility to select different parameters to generate a large number of *n*D Cat matrices. This section discusses the parameter settings and proposes a computation efficiency improvement method for generating *n*D Cat matrices with Laplace expansions.

A. Parameter Settings

In the proposed *n*D Cat map generation method in Algorithm 1, the parameters in $\mathbf{I}$ and $\mathbf{J}$ control the locations of the dependent elements and they are called the spatial configuration parameters (SCPs), and the parameters in $\mathbf{P}$ and $\mathbf{Q}$ are newly added independent matrix elements in each iteration of the matrix expansion and they are called matrix entries parameters (MEPs). Two SCPs are needed to generate an *n*D Cat matrix $\mathbf{C}^{n\mathrm{D}}$ from $\mathbf{C}^{(n-1)\mathrm{D}}$. Each SCP is an integer within range of $[1, n]$ and thus has n possible choices [Fig. 1(b)]. Therefore, when generating $\mathbf{C}^{n\mathrm{D}}$ from $\mathbf{C}^{1\text{-D}}$, the total number of SCPs $\mathbb{N}_{\mathrm{SCP}}$ and different choices $\mathbb{C}_{\mathrm{SCP}}$ can be calculated as

$$\begin{aligned}\mathbb{N}_{\mathrm{SCP}} &= 2(n-1)\\ \mathbb{C}_{\mathrm{SCP}} &= n^2 \times (n-1)^2 \times \cdots \times 2^2\\ &= (n!)^2.\end{aligned}$$

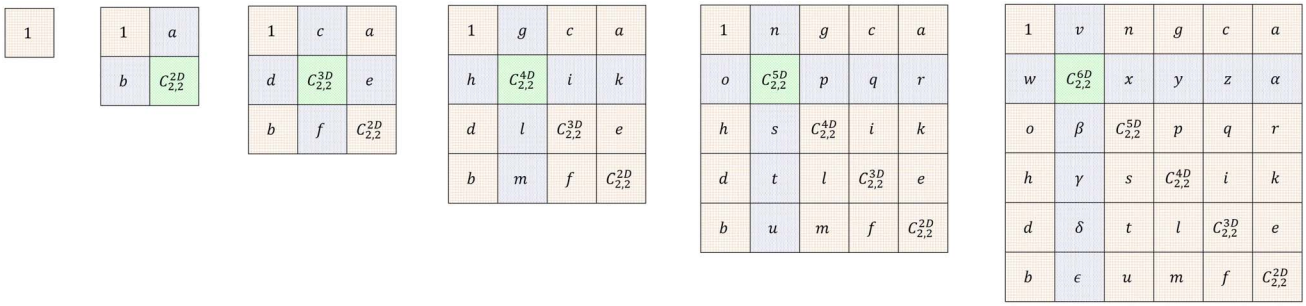

Fig. 2. Example of generating a 6-D Cat matrix. (a) 1-D Cat matrix $\mathbf{C}^{\text{1-D}}$ (dummy). (b) 2-D Cat matrix $\mathbf{C}^{\text{2-D}}$. (c) 3-D Cat matrix $\mathbf{C}^{\text{3-D}}$. (d) 4-D Cat matrix $\mathbf{C}^{\text{4-D}}$. (e) 5-D Cat matrix $\mathbf{C}^{\text{5-D}}$. (f) 6-D Cat matrix $\mathbf{C}^{\text{6-D}}$.

Algorithm 1 Proposed nD Cat Map Generation Method $\mathbf{C}^{n\text{D}}_{\text{Ours}} = \mathbb{L}(\mathbf{I}, \mathbf{J}, \mathbf{P}, \mathbf{Q})$

Input: $\mathbf{I} = \{i_1, i_2, \ldots, i_{n-1}\}$ is an vector with $i_k \in [1, k+1]$ for all $k \in [1, n-1]$
Input: $\mathbf{J} = \{j_1, j_2, \ldots, j_{n-1}\}$ is an vector with $j_k \in [1, k+1]$ for all $k \in [1, n-1]$
Input: $\mathbf{P} = \{\mathbf{P}^1, \mathbf{P}^2, \ldots, \mathbf{P}^{n-1}\}$ is a vector list and $\mathbf{P}^k$ is a k-length vector for all $k \in [1, n-1]$
Input: $\mathbf{Q} = \{\mathbf{Q}^1, \mathbf{Q}^2, \ldots, \mathbf{Q}^{n-1}\}$ is a vector list and $\mathbf{Q}^k$ is a k-length vector for all $k \in [1, n-1]$

1: $\mathbf{C}^{1D} = [1]$;
2: **for** $k = 1$ to $n-1$ **do**
3: Extend $\mathbf{C}^{kD}$ into a $(k+1) \times (k+1)$ matrix $\mathbf{C}^{(k+1)D}$ by placing its partitions $\mathbf{C}^{kD}_{UL}$, $\mathbf{C}^{kD}_{UR}$, $\mathbf{C}^{kD}_{LL}$ and $\mathbf{C}^{kD}_{LR}$ with respect to the associated element location (i_k, j_k);
4: Set elements in the i_k-th row and j_k-th column in the extended matrix with respect to the parameter vectors $\mathbf{P}^k$ and $\mathbf{Q}^k$ as follows

$$\mathbf{C}^{(k+1)D}_{i_k,m} = \begin{cases} \mathbf{P}^k_m & \text{if } m < j_k \\ \mathbf{P}^k_{m-1} & \text{if } m > j_k \end{cases} \text{ and } \mathbf{C}^{(k+1)D}_{m,j_k} = \begin{cases} \mathbf{Q}^k_m & \text{if } m < i_k \\ \mathbf{Q}^k_{m-1} & \text{if } m > i_k \end{cases};$$

5: Calculate element $\mathbf{C}^{(k+1)D}_{i_k,j_k}$ using (12).
6: **end for**

Ensure: nD Cat matrix $\mathbf{C}^{n\text{D}}$ with $\left|\mathbf{C}^{n\text{D}}\right| = 1$

When extending an $(n-1)$D Cat matrix $\mathbf{C}^{(n-1)\text{D}}$ to $\mathbf{C}^{n\text{D}}$, $2(n-1)$ independent matrix elements are needed [Fig. 1(c)]. Thus, the total number of MEPs can be computed as

$$\begin{aligned} \mathbb{N}_{\text{MEP}} &= 2(n-1) + 2(n-2) + \cdots + 2 \\ &= n^2 - n. \end{aligned}$$

If we assume that all MEPs are selected from a symbol set with M possible values, then the total number of different choices to form $(n^2 - n)$ MEPs is $\mathbb{C}_{\text{MEP}} = M^{n^2-n}$. Finally, due to the independency of SCPs and MEPs, the whole parameter space is simply the production of their individual parameter spaces, namely $\mathbb{C}_{\text{S}} = \mathbb{C}_{\text{SCP}}\mathbb{C}_{\text{MEP}}$, which can be seen in Table I. Thus users have great flexility to choose different parameter settings to generate a large number of different nD Cat matrices.

TABLE I
PARAMETER SPACES OF nD CAT MAPS GENERATED USING LAPLACE EXPANSIONS

Dimensions	$\mathbb{C}_{\text{SCP}}$	$\mathbb{C}_{\text{MEP}}$	$\mathbb{C}_{\text{S}}$
2	4	M^2	$4M^2$
3	36	M^6	$36M^6$
4	576	M^{12}	$576M^{12}$
5	14400	M^{20}	$14400M^{20}$
...	...	...	...
n	$(n!)^2$	M^{n^2-n}	$(n!)^2 M^{n^2-n}$

B. Computation Efficiency Improvement

Computing the unknown element in (12) requires to perform n multiplications and n additions and to evaluate the determinants of $(n-1)$ matrices with size of $(n-1) \times (n-1)$. Calculating the determinant of an $n \times n$ matrix requires the computation complexity of order $O(n^3)$ when using some methods like the Gaussian elimination, lower-upper (LU) matrix decomposition, or QR decomposition (Q is an orthogonal matrix and R is an upper triangular matrix) [45]. Thus, computing the unknown element by (12) costs the computation complexity of order $O(n^4)$. Extending an $(n-1)$D Cat matrix to nD also requires the computation complexity of order $O(n^4)$ because it needs to compute only the unknown element in (12). Therefore, generating an nD Cat matrix from the 1-D Cat matrix $\mathbf{C}^{\text{1-D}}$ requires the computation complexity of order $O(n^5)$. In order to further improve the computation efficiency, we introduce Theorem 2.

Theorem 2: In the context of (12), X_{ij} can be alternatively computed as

$$X_{ij} = (-1)^{i+j}(1 - |\mathbf{G}|) \tag{16}$$

with the aid of an auxiliary matrix $\mathbf{G}$, whose element $G_{ij} = 0$ and all other elements are identical to those in $\mathbf{X}$.

Appendix B provides the mathematical proof of Theorem 2. As a result, computing the unknown element in an extended Cat matrix using (16) instead of (12) only requires to compute two additions and the determinant of an $n \times n$ matrix. Its computation complexity is of order $O(n^3)$. Thus, generating an nD Cat matrix needs the computation complexity of order $O(n^4)$. In this way, we can successfully reduce the computation complexity of the proposed method for one magnitude.

C. Chaotic Behavior Analysis

The "chaos" phenomenon is difficult to exactly define and qualitatively measure since everyone has a different viewpoint about what the chaos is. The Lyapunov exponent (LE) [46] is to describe the exponential divergence of two close trajectories. It can be used to measure whether a dynamical system has chaotic behaviors. With a positive LE value, the difference of two close trajectories of a dynamical system will exponentially increase, and ultimately make the two trajectories totally different. Thus, a dynamical system with a positive LE value can be considered as chaotic. For high dimension dynamical systems, they have more than one LE value and the largest LE (lgtLE) value is an indictor of its chaotic behaviors.

An nD discrete dynamical system can be defined by

$$X(t+1) = \mathbb{F}(X(t)) = \begin{pmatrix} \mathbb{A}^1 \\ \mathbb{A}^2 \\ \cdots \\ \mathbb{A}^n \end{pmatrix} X(t) \tag{17}$$

where $X(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T$ and $X(t+1) = [x_1(t+1), x_2(t+1), \ldots, x_n(t+1)]^T$ are the input and output. If $\mathbb{A}^i$ $(i = 1, 2, \ldots, n)$ are differentiable functions, the Jacobian matrix $\mathbf{J}$ of the system $\mathbb{F}(\cdot)$ can be defined as follows:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \mathbb{A}^1}{\partial x_1} & \frac{\partial \mathbb{A}^1}{\partial x_2} & \cdots & \frac{\partial \mathbb{A}^1}{\partial x_n} \\ \frac{\partial \mathbb{A}^2}{\partial x_1} & \frac{\partial \mathbb{A}^2}{\partial x_2} & \cdots & \frac{\partial \mathbb{A}^2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \mathbb{A}^n}{\partial x_1} & \frac{\partial \mathbb{A}^n}{\partial x_2} & \cdots & \frac{\partial \mathbb{A}^n}{\partial x_n} \end{pmatrix}. \tag{18}$$

Therefore, in each iteration, $\mathbf{J}(t)$ corresponds to the iteration output $X(t)$. If $\mathbf{J}(t)$ has m eigenvalues, given by $\mu_i(t)$, the n LE values of the system $\mathbb{F}(\cdot)$ are defined as follows:

$$\lambda_j = \lim_{k \to \infty} \left\{ \frac{1}{k} \sum_{t=1}^{k} \ln \mu_j(t) \right\} \tag{19}$$

where $j = 1, 2, \ldots, n$. Chaotic behaviors of the system $\mathbb{F}(\cdot)$ is demonstrated by the lgtLE value among λ_j.

To analyze the chaotic behaviors of the nD Cat maps generated by the proposed method, we perform the following experiment: for our nD Cat map generation method, we set $M = 2$ (MPEs are restricted to $\{0, 1\}$) and randomly generate 5×2^n nD Cat maps for $n \in \{3, 4, \ldots, 10\}$. The lgtLE values of these nD Cat maps are calculated. Fig. 3 plots the mean lgtLE values for different dimensions n. As can be seen, the mean lgtLE values of nD Cat maps are all positive and they become bigger when the dimension n increases. This means that the proposed method can generate nD Cat maps with complex chaotic behaviors.

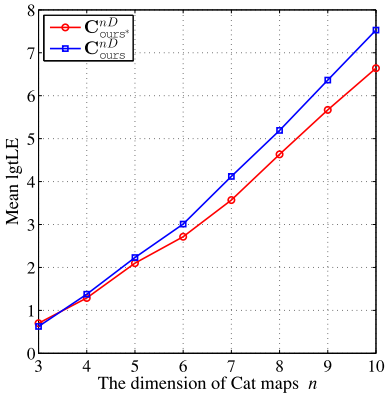

Fig. 3. Mean lgtLE values of nD Cat maps generated by the proposed method.

TABLE II
PARAMETER SPACES OF DIFFERENT nD CAT MAP GENERATION METHODS

nD Cat matrices	Parameter Numbers		Parameter Spaces[2]
	$\mathbb{N}_{\text{SCP}}$	$\mathbb{N}_{\text{MEP}}$	$\mathbb{C}_{\text{S}}$
$\mathbf{C}^{nD}_{\text{T}}$ [41]	0	$n^2 - n$	M^{n^2-n}
$\mathbf{C}^{nD}_{\text{F}}$ [40][1]	1	$n^2/2$	$nM^{n^2/2}$
$\mathbf{C}^{nD}_{\text{N}}$ [42]	0	2	M^2
$\mathbf{C}^{nD}_{\text{ours}^*}$	0	$n^2 - n$	M^{n^2-n}
$\mathbf{C}^{nD}_{\text{ours}}$	$2(n-1)$	$n^2 - n$	$(n!)^2 M^{n^2-n}$

[1] Results are given with respect to the maximum parameter space.
[2] Results are estimated when MEPs are from M possible values.

V. PERFORMANCE EVALUATIONS AND COMPARISONS

In this section, the performance of the proposed nD Cat map generation method is compared with several other methods.

A. Parameter Space

The computation precision of implementing a chaotic map in software or hardware is finite, implying that the chaotic complexity of a chaotic map in real numbers will inevitably degrade [47]. Consequently, the parameter space of an nD Cat map generation method is always limited. Moreover, the parameter space is also closely related to the total number of distinctive Cat matrices, which is known as the security key space in cryptography applications [24], [37]. Therefore, an nD Cat map generation method with sufficiently large parameter space is desired to against the brute-force attacks.

Table II shows the parameter spaces of different nD Cat map generation methods. It is clear that the proposed method has the largest parameter space in all listed methods. This is because, compared with other methods that use only the matrix entries as the parameters, the proposed method introduces parameters to control the spatial configurations of an nD Cat matrix. These spatial parameters can significantly increase the parameter space of an nD Cat matrix.

TABLE III
ALGORITHM COMPLEXITY OF DIFFERENT nD CAT MAP GENERATION METHODS

nD Cat matrices	Operations			Time complexity[4]					Memory cells
	Element addition	Matrix multiplication[2]	Determinant[3]	3D	5D	10D	100D	Order	
$\mathbf{C}_{\mathsf{T}}^{nD}$ [41]	$n(n-1)/2$	$(n(n-1)/2-1)\,O_M(n)$	0	57	1135	44045	$100^{4.8473}$	$O(n^5)$	$2n^2$
$\mathbf{C}_{\mathsf{F}}^{nD}$ [40][1]	$(n/2)^2$	$O_M(n/2)$	0	6	22	150	$100^{2.5528}$	$O(n^3)$	n^2
$\mathbf{C}_{\mathsf{N}}^{nD}$ [42]	0	$\sum_{k=3}^{n}(k-1)O_M(k)$	0	54	746	22300	$100^{4.6532}$	$O(n^5)$	$2n^2+(n-1)^2$
$\mathbf{C}_{\mathsf{ours}*}^{nD}$,$\mathbf{C}_{\mathsf{ours}}^{nD}$	$2(n-1)$	0	$\sum_{k=2}^{n}O_D(k)$	39	232	3042	$100^{3.7033}$	$O(n^4)$	n^2

[1] Time complexity is given with respect to its maximum complexity.
[2] $O_M(k)$ denotes the time complexity of multiplying two $k\times k$ matrices.
[3] $O_D(k)$ denotes the time complexity of computing the determinant of a $k\times k$ matrix.
[4] We simply consider $O_M(k)=O_D(k)=O(k^3)$, although $O_M(k)=O(k^{2.373})$ in [48] and $O_D(k)=O(k^{2.376})$ in [45].

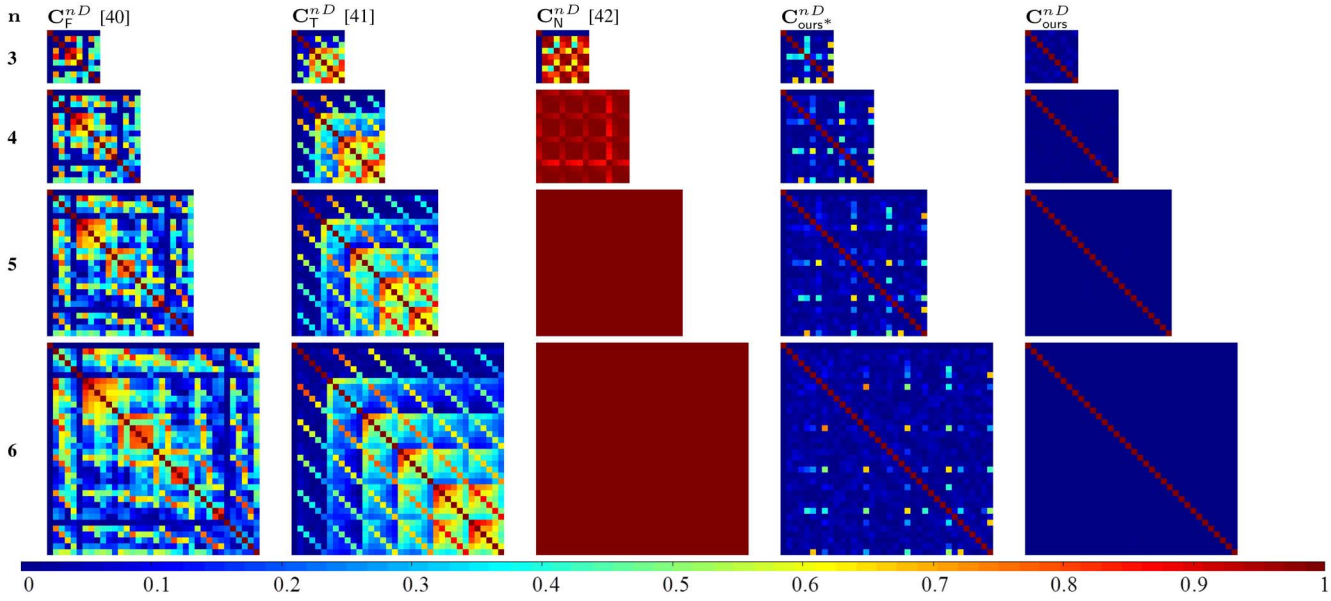

Fig. 4. Matrix element correlation analysis of different nD Cat map generation methods.

B. Algorithm Complexity

Because of the high complexity of generating a multidimensional Cat matrix, the computation and memory costs are quite important factors to evaluate whether an nD Cat map generation method is suitable for the applications. Table III lists the algorithm complexity of different nD Cat map generation methods. As can be seen, for generating an nD Cat matrix, the proposed method requires n^2 memory cells and its computation complexity is $O(n^4)$, which is one magnitude less than the Tang and Tang's [41] and Nance's [42] methods.

C. Matrix Element Correlation

In the process of the nD Cat matrix generation, the settings of matrix elements are totally decided by the generation method. The Cat matrices generated by different methods have different degrees of inner correlations between their matrix elements. As discussed in Section II-D, an nD Cat matrix with low inner correlation is commonly preferred for many applications.

To analyze the correlations of matrix elements in nD Cat matrices generated by different methods, we perform the following correlation analysis: for each nD Cat map generation method $*$, we randomly generate 1000 Cat matrices $\mathbf{C}_{*_1}^{n\mathrm{D}}, \mathbf{C}_{*_2}^{n\mathrm{D}}, \ldots, \mathbf{C}_{*_{1000}}^{n\mathrm{D}}$ for $M=4$ (MEPs are restrict to $\{0,1,2,3\}$). Suppose p and q are two different indexes ($1\le p,q\le n^2$), the correlation coefficient (CE) between pth and qth matrix elements in $\mathbf{C}_{*}^{n\mathrm{D}}$ is defined as follows:

$$\mathrm{CE}^{\mathbf{C}_{*}^{n\mathrm{D}}}(p,q)=\frac{E\left[\left(S_p-\mu_{S_p}\right)\left(S_q-\mu_{S_q}\right)\right]}{\sigma_{S_p}\sigma_{S_q}} \tag{20}$$

where μ and σ are the mean value and standard deviation and S_p and S_q are sequences of matrix elements in the form of

$$\begin{aligned} S_p&=\left\{\mathbf{C}_{*_1}^{n\mathrm{D}}[p],\ldots,\mathbf{C}_{*_{1000}}^{n\mathrm{D}}[p]\right\}\\ S_q&=\left\{\mathbf{C}_{*_1}^{n\mathrm{D}}[q],\ldots,\mathbf{C}_{*_{1000}}^{n\mathrm{D}}[q]\right\}. \end{aligned} \tag{21}$$

They contain the pth and qth elements in these 1000 randomly generated matrices, respectively. Thus, $\mathrm{CE}^{\mathbf{C}_{*}^{n\mathrm{D}}}(p,q)$ is the correlation coefficient between the sequences S_p and S_q. The CE value closing to 0 means low correlation between S_p and S_q while closing to 1 means high correlation.

Fig. 4 plots the magnitudes of these correlation coefficients for 3-D, 4-D, 5-D, and 6-D Cat matrices generated by various methods. As can be seen for each color matrix, the values in the pth row denote the correlation coefficients between the pth matrix element and all the n^2 matrix elements, namely $\mathrm{CE}^{\mathbf{C}_{*}^{n\mathrm{D}}}(p,1), \mathrm{CE}^{\mathbf{C}_{*}^{n\mathrm{D}}}(p,2),\ldots,\mathrm{CE}^{\mathbf{C}_{*}^{n\mathrm{D}}}(p,n^2)$, because $\mathrm{CE}^{\mathbf{C}_{*}^{n\mathrm{D}}}(p,q)=\mathrm{CE}^{\mathbf{C}_{*}^{n\mathrm{D}}}(q,p)$ and $\mathrm{CE}^{\mathbf{C}_{*}^{n\mathrm{D}}}(p,p)=1$,

TABLE IV
STATISTICS OF nD CAT MATRIX ELEMENTS THAT COULD BE CONTROLLED BY DIFFERENT NUMBERS OF MEPS IN DIFFERENT METHODS

# of matrix elements with	nD Cat matrices				
	$\mathbf{C}_{\mathsf{T}}^{n\,D}$ [41]	$\mathbf{C}_{\mathsf{F}}^{n\,D}$ [40][1]	$\mathbf{C}_{\mathsf{N}}^{n\,D}$ [42]	$\mathbf{C}_{\mathsf{ours}*}^{n\,D}$	$\mathbf{C}_{\mathsf{ours}}^{n\,D}$
Constant 0	0	$m^2 - m$	0	0	0
Constant 1	1	m	0	1	0
1 MEP	$n-1$	$2m(n-m)$	0	n^2-n	0
1+ MEP	n^2-n	$(n-m)^2$	n^2	$n-1$	n^2

[1] m is a parameter in $\mathbf{C}_{\mathsf{F}}^{n\,D}$ with $1 \le m \le n-1$ (see Eq. (8) for details).

all magnitudes are on the main diagonal symmetry and the magnitudes on the main diagonal line are 1, marked as red. It is easy to see that $\mathbf{C}_{\mathsf{ours}*}^{n\mathsf{D}}$ generated with specific spatial configurations have patterns in their element correlation plots, while $\mathbf{C}_{\mathsf{ours}}^{n\mathsf{D}}$ generated with randomly determined spatial configurations have CE values close to 0. Besides, it is noticeable that multidimensional Cat matrices generated by the proposed method have smaller CE values, and thus have less correlated matrix elements than those of other methods.

All these methods have their own characteristics in generating multidimensional Cat matrices. For example, if the generated multidimensional Cat matrices always contain a constant element in a fixed position, the correlation between this element and any other elements is always 0. On the other hand, the more elements controlled by a single MEP, the less correlation between two Cat matrix elements. For the multidimensional Cat matrices generated by various methods, Table IV lists the statistics of their elements that could be controlled by different numbers of MEPs. It is clear that $\mathbf{C}_{\mathsf{ours}*}^{n\mathsf{D}}$ generated by the proposed method with fixed spatial parameters has the largest number of elements controlled by one MEP, implying that most of matrix elements in the nD Cat matrices are independent. Considering spatial parameters, no element in $\mathbf{C}_{\mathsf{ours}}^{n\mathsf{D}}$ is a constant and all of them are determined in a more complicated way involving both SCPs and MEPs. Consequently, for the Cat matrices generated by the proposed method, any two of their elements are uncorrelated as shown in Fig. 4.

D. Shannon Entropy

Due to the finite precision of digital devices, the nD Cat maps are commonly used in discrete forms in real-world applications as shown in (1). In (1), $Y(t)$ and $Y(t+1)$ are observations of the outputs of the nD Cat map with respect to time t and $t+1$. Each observation has n channels and each channel has N states. Thus observation $Y(t)$ has the change to attain a finite number N^n of states.

The randomness of the discrete nD Cat map can be measured by the Shannon entropy [17] defined as follows:

$$H = -\sum_{k=1}^{N^n} \Pr(k) \log_2 \Pr(k) \tag{22}$$

where $\Pr(k)$ denotes the probability of seeing an observation $Y(t)$ in the kth state. Then we have $0 \le H \le n \log_2 N$ and a bigger Shannon entropy value indicates better randomness.

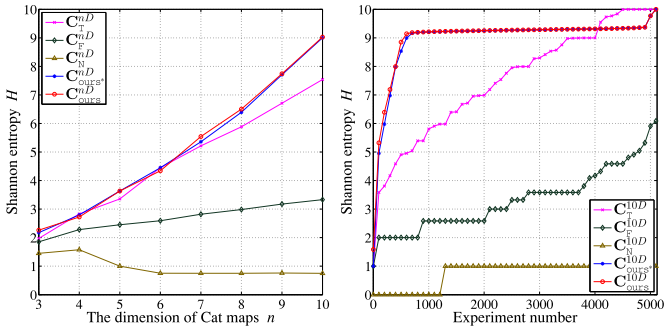

Fig. 5. Shannon entropy values of nD Cat maps for $N = 2$. (a) Mean of Shannon entropy values of nD Cat maps versus map dimension n. (b) Sorted Shannon entropy scores of 10-D Cat maps for each generation method.

$H_{\min} = 0$ implies that the system is completely predictable and $H_{\max} = n \log_2 N$ when if and only if all observations absolutely uniformly distribute over the N^n states.

Two experiment groups of Shannon entropy analysis are performed to evaluate the randomness of various nD Cat map generation methods. The first group of experiments is to analyze the Shannon entropy values against the Cat map dimension n by fixing $N = 2$. For each Cat map generation method, we randomly generate $5N^n$ nD Cat maps and compute the Shannon entropy values of these nD Cat maps for $n \in \{3, 4, \ldots, 10\}$. The observation time series is of length 2^{n+1} with respect to the initial state $Y(0) = \mathbf{1}_{n\times 1}$.

The mean Shannon entropy results of different methods are given in Fig. 5(a). As can be seen, the values of all generation methods increase as the dimension n increases except for the Cat maps generated by the Nance's [42] method. The low randomness of the Nance's method is caused by its small parameter space and high matrix element correlations, which can be seen in Table II and Fig. 4, respectively. The nD Cat maps generated by the proposed methods achieve high Shannon entropy values for all test dimensions. Fig. 5(b) shows the Shannon entropy values of 10-D Cat maps generated by different methods in the ascending order. Our methods $\mathbf{C}_{\mathsf{ours}*}^{n\mathsf{D}}$ and $\mathbf{C}_{\mathsf{ours}}^{n\mathsf{D}}$ have bigger Shannon entropy values than other methods in the first 4000 10-D Cat maps. Tang and Tang's [41] method can achieve the biggest Shanon entropy values when the number of 10-D Cat maps is over 4000. At last, our proposed and Tang and Tang's [41] methods can achieve the same biggest Shannon entropy values. Fig. 5(b) also shows that the outputs of 10-D Cat maps generated by our methods $\mathbf{C}_{\mathsf{ours}*}^{n\mathsf{D}}$ and $\mathbf{C}_{\mathsf{ours}}^{n\mathsf{D}}$ have similar randomness. This is because their MEPs are randomly selected from the same data set. Table V lists the maximum Shannon entropy values for nD Cat maps generated by different methods with various dimensions. It is noticeable that the proposed and Tang and Tang's [41] methods are able to generate nD Cat maps with high Shannon entropy values that are quite close to the theoretical maximum Shannon entropy values $H_{\max} = n \log_2 N = n$.

The second group of experiments is to analyze the Shannon entropy values against the state number N by fixing the dimension $n = 3$. The state number $N = \{2, 3, \ldots, 10\}$ and for each

TABLE V
MAXIMUM SHANNON ENTROPY VALUES OF nD CAT MAPS GENERATED BY DIFFERENT METHODS WITH $N = 2$

Dimensions		**nD Cat matrices**				
n	$H_{\max}$	$\mathbf{C}_{\mathrm{T}}^{nD}$ [41]	$\mathbf{C}_{\mathrm{F}}^{nD}$ [40]	$\mathbf{C}_{\mathrm{N}}^{nD}$ [42]	$\mathbf{C}_{\mathrm{ours}*}^{nD}$	$\mathbf{C}_{\mathrm{ours}}^{nD}$
3	3	2.781	2.561	2.000	2.781	2.781
4	4	3.890	3.000	2.000	3.890	3.890
5	5	4.945	3.579	1.000	4.945	4.945
6	6	5.973	4.579	1.000	5.973	5.973
7	7	6.986	4.584	1.000	6.986	6.986
8	8	7.993	5.904	1.000	7.993	7.993
9	9	8.997	5.907	1.000	8.997	8.997
10	10	9.998	6.907	1.000	9.998	9.998

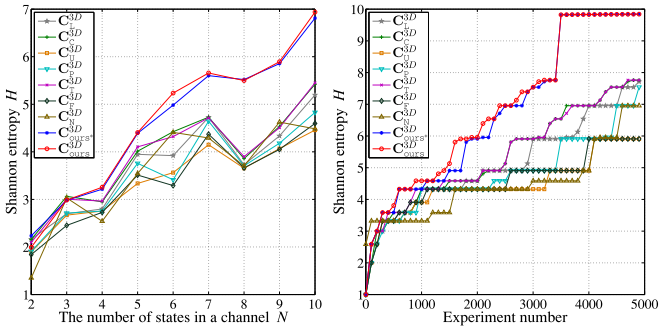

Fig. 6. Shannon entropy of 3-D Cat maps. (a) Mean of Shannon entropy values of 3-D Cat maps versus the number of channel states N. (b) Sorted Shannon entropy scores of 3-D Cat maps with $N = 10$ for each generation method.

TABLE VI
MAXIMUM SHANNON ENTROPY VALUES OF 3-D CAT MAPS GENERATED BY DIFFERENT METHODS

3D Cat matrices	**Number of finite states N**								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$H_{\max}$	3.000	4.755	6.000	6.966	7.755	8.422	9.000	9.510	9.966
$\mathbf{C}_{\mathrm{L}}^{3D}$ [37]	2.781	3.697	3.807	4.954	6.507	5.833	4.807	5.285	7.762
$\mathbf{C}_{\mathrm{C}}^{3D}$ [24]	2.781	3.697	3.807	4.954	6.507	5.833	4.807	5.285	7.762
$\mathbf{C}_{\mathrm{U}}^{3D}$ [38]	2.561	3.582	3.585	4.907	5.170	5.807	4.585	5.170	5.907
$\mathbf{C}_{\mathrm{P}}^{3D}$ [39]	2.561	3.697	3.585	4.954	6.285	5.833	4.585	5.285	7.539
$\mathbf{C}_{\mathrm{T}}^{3D}$ [41]	2.781	3.697	3.807	4.954	6.507	5.833	4.807	5.285	7.762
$\mathbf{C}_{\mathrm{F}}^{3D}$ [40]	2.561	3.582	3.585	4.907	5.170	5.807	4.585	5.170	5.907
$\mathbf{C}_{\mathrm{N}}^{3D}$ [42]	2.000	3.697	3.585	4.954	5.700	5.833	4.585	5.285	6.954
$\mathbf{C}_{\mathrm{ours}*}^{3D}$	2.781	3.697	5.805	6.786	7.605	8.288	8.875	9.389	9.854
$\mathbf{C}_{\mathrm{ours}}^{3D}$	2.781	4.496	5.805	6.801	7.616	8.314	8.875	9.426	9.855

3-D Cat map generation method, randomly generating $5N^3$ 3-D Cat maps and computing their Shannon entropy values using an observed time series of length N^4 with respect to initial state $Y(0) = \mathbf{1}_{3\times 1}$. The mean Shannon entropy values of different methods are shown in Fig. 6(a). It is clear that on average the 3-D Cat maps generated by the proposed method have higher mean Shannon entropy values than other methods. Fig. 6(b) shows the Shannon entropy values of 3-D Cat maps generated by different methods in the ascending order with $N = 10$. Compared with other listed generation methods, the proposed method has the fastest rate to achieve its highest Shannon entropy values.

Table VI lists the maximum Shannon entropy values of 3-D Cat maps generated by different methods with different state numbers N. As can be seen, the proposed method outperforms all other listed generation methods. Its generated 3-D Cat maps can achieve the highest Shannon entropy values for $N = \{2, 3, \ldots, 10\}$ that are extremely close to the theoretical maximum scores. These further verify that the Cat maps generated by the proposed method are more stable to generate random-like outputs.

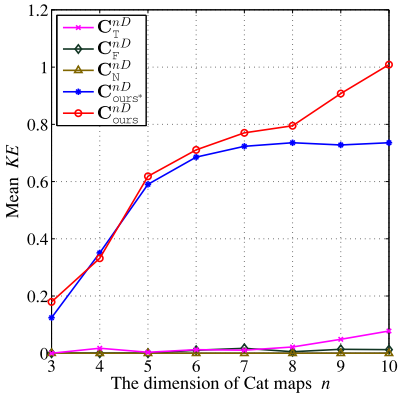

Fig. 7. Mean Kolmogorov entropy values of nD Cat maps generated by different methods.

E. Kolmogorov Entropy

The Kolmogorov entropy [49] can measure whether extra information is needed to predict the trajectory of a dynamical system, which is defined as follows:

$$\mathrm{KE} = \lim_{\tau\to 0} \tau^{-1} \lim_{\varepsilon\to 0} \lim_{m\to\infty} K_{m,\tau}(\varepsilon) \tag{23}$$

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

$$K_{m,\tau}(\varepsilon) = -\sum_{i_1,\ldots,i_m \le n(\varepsilon)} p(i_1,\ldots,i_m)\log p(i_1,\ldots,i_m). \tag{24}$$

Divide the phase plane into m nonoverlapping partitions $\phi_{i_1},\ldots,\phi_{i_m}$, then $p(i_1,\ldots,i_m)$ is the probability of correctly predicted trajectory in partition ϕ_{i_1} at time τ, in partition ϕ_{i_2} at time $2\tau,\ldots,$ in partition ϕ_{i_m} at time $m\tau$. Positive Kolmogorov entropy value indicates chaotic behaviors of the dynamical system and a bigger value means better unpredictability.

For different nD Cat map generation methods, we investigate their Kolmogorov entropy values against the Cat map dimension n. When generating nD Cat matrices, the number of possible values is set as $M = 4$ (MPEs are restricted to $\{0, 1, 2, 3\}$). For each generation method with different dimensions n, we randomly generate $10n$ Cat maps and then obtain trajectories of these nD Cat maps with the number of finite states $N = 4$ and initial value $Y(0) = \mathbf{1}_{n\times 1}$. For each iteration output $Y(t) = [y_1(t), y_2(t), \ldots, y_n(t)]^T$, we convert it into float number within the range of $[0, 1]$ by

$$\mathbf{y}_t = \frac{\sum_{i=1}^{n} 4^{i-1} y_i(t)}{4^n}. \tag{25}$$

The generated trajectories are with length of 12 000 and we use the method proposed in [50] to calculate their Kolmogorov entropy values. Fig. 7 plots the mean values of Kolmogorov entropy results of different generation methods.

TABLE VII
MAXIMUM KOLMOGOROV ENTROPY VALUES OF nD CAT MAPS GENERATED BY DIFFERENT METHODS

Dimensions	nD Cat matrices				
n	$\mathbf{C}_{\mathrm{T}}^{nD}$ [41]	$\mathbf{C}_{\mathrm{F}}^{nD}$ [40]	$\mathbf{C}_{\mathrm{N}}^{nD}$ [42]	$\mathbf{C}_{\mathrm{ours}*}^{nD}$	$\mathbf{C}_{\mathrm{ours}}^{nD}$
3	0.000	0.000	0.000	0.765	0.868
4	0.693	0.000	0.000	0.759	0.769
5	0.063	0.000	0.000	0.761	0.753
6	0.416	0.223	0.000	0.767	0.755
7	0.128	0.575	0.000	0.754	1.931
8	0.401	0.337	0.000	0.756	1.981
9	0.275	0.406	0.000	0.758	1.933
10	0.359	0.223	0.000	0.756	1.932

As can be seen, the nD Cat maps generated by the proposed methods can achieve positive mean Kolmogorov entropy values and the values become bigger as the dimension n increases. For other methods, their randomly generated nD Cat maps achieve quite small mean Kolmogorov entropy values that are close to 0. This is because the trajectories of the discrete Cat map are periodic [29] and its period is small when the elements of its Cat matrix are highly correlated. Table VII lists the maximum Kolmogorov entropy values of nD Cat maps generated by different methods with various dimensions n. It is obvious that the proposed method is able to generate nD Cat maps with bigger Kolmogorov entropy values than other methods.

VI. PROPOSED PSEUDORANDOM NUMBER GENERATOR

To show the effectiveness of the nD Cat maps generated by our proposed method, as an example, this section introduces a PRNG using the 5-D Cat maps generated by the proposed method.

Here, we set $N = 1$ and the initial value $Y(0)$ as a float number in (1). Given $\mathbf{I}$, $\mathbf{J}$, $\mathbf{P}$, and $\mathbf{Q}$, we use (13) to produce a 5-D Cat matrix $\mathbf{C}_{\mathrm{ours}}^{5\text{-D}}$. Suppose $\{Y(t)|t = 1, 2, \ldots\}$ is the output of $\mathbf{C}_{\mathrm{ours}}^{5\text{-D}}$ with the initial value $Y(0)$. $Y(t)$ is with size of 5×1. The proposed PRNG is defined as follows:

$$T(t) = \mathrm{Bin}\Big[\sum Y(t)\Big]_{31:32} \tag{26}$$

where $\mathrm{Bin}[\alpha]_{31:32}$ is a function to convert the float number α into a 52-bit binary stream using the IEEE 754 standard [51] and then fetch its 31st and 32nd digital numbers.

A. Diehard Statistical Test

Diehard statistical test suit [52] is a battery of statistical tests that is widely used for measuring the quality of a PRNG. It contains 15 subtests and shows good performance for data sequences with a large size. These 15 empirical subtests are designed to find the nonrandomness areas in a large-size data sequence and total 234 p-values are generated. For a random number sequence with high quality, these p-values are expected to randomly distribute to pass the Diehard statistical test. If six or more p-values with 0 or 1 are obtained, a data sequence is considered to fail the test. The size of the test sequence is suggested as 11 468 800 byte.

Fig. 8 plots the Diehard statistical test results of a random number sequence generated by the proposed PRNG.

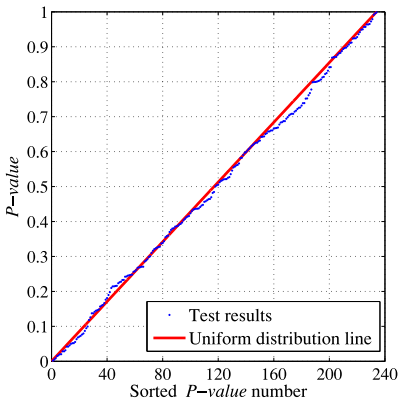

Fig. 8. Diehard statistical test results.

TABLE VIII
TESTU01 RESULTS FOR BINARY SEQUENCES WITH DIFFERENT LENGTHS

Lengths \ Test batteries	*Rabbits*	*Alphabit*	*BlockAlphabit*
2^{20}	37/38	17/17	17/17
2^{25}	38/38	17/17	17/17
2^{30}	38/38	17/17	17/17

TABLE IX
PERFORMANCE COMPARISONS OF DIFFERENT nD CAT MAP GENERATION METHODS

Comparison items	nD Cat matrices				
	$\mathbf{C}_{\mathrm{T}}^{nD}$ [41]	$\mathbf{C}_{\mathrm{F}}^{nD}$ [40]	$\mathbf{C}_{\mathrm{N}}^{nD}$ [42]	$\mathbf{C}_{\mathrm{ours}*}^{nD}$	$\mathbf{C}_{\mathrm{ours}}^{nD}$
Parameter space	2	4	5	2	1
Algorithm complexity	5	1	4	2	2
Matrix element correlation	3	4	5	2	1
Shannon entropy	3	4	5	2	1
Kolmogorov entropy	3	4	5	2	1

The sorted 234 p-values are all distributed on or nearby the uniform distribution line (the red line in Fig. 8). This means that the generated random number sequence can pass the Diehard statistical test.

B. TestU01 Test

TestU01 is a random number test package that provides convinced empirical description for the quality of the PRNG [53]. It contains six test batteries and three of them (Rabbits, Alphabit, and BlockAlphabit) are used to test the randomness of binary sequences. Rabbits applies 38 subtests while Alphabit and BlockAlphabit apply 17 subtests. Each subtest will generate a p-value and the random number sequence is considered to pass the subtest if the generated p-value is within the range of [0.001, 0.999].

In our experiment, binary sequences with different lengths are generated by the proposed PRNG and tested by Rabbits, Alphabit, and BlockAlphabit. The test results are shown in Table VIII. As can be seen, binary sequences with different lengths can pass almost all the subtests except for one subtest in Rabbits.

VII. Conclusion

In this paper, we have introduced an efficient method of generating nD Cat maps using Laplace expansions. It can iteratively generate any dimension Cat matrix from a 1-D Cat matrix. Unlike existing nD Cat map generation methods, which only consider MEPs to control the matrix elements, the proposed method also introduces SCPs to control the spatial configurations.

Theoretical analysis and experiment evaluations have been done to show that the proposed nD Cat map generation method can combine the dynamics of different dimensions and generate an nD Cat matrix with $(n^2 - n)$ independent matrix elements. The performance of the proposed method has been compared with other nD Cat map generation methods and the comparison results are shown in Table IX. As can be seen, the proposed method can achieve the best performance in the parameter space, matrix element correlation, Shannon entropy, Kolmogorov entropy, and the second-best performance in algorithm complexity. To investigate its real-world applications, we have proposed a new PRNG using the 5-D Cat maps generated by the proposed method. Our future work will explore its applications in secure communication and data encryption.

Acknowledgment

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

Appendix A
Proof of Theorem 1

Here proves Theorem 1 in Section III-A.

Proof: Compute the determinant of an $n \times n$ matrix $\mathbf{X}$ using Laplace expansions

$$|\mathbf{X}| = \sum_{p=1}^{n} X_{pq}(-1)^{p+q} A_{pq}^{X} = X_{ij}(-1)^{i+j} A_{ij}^{X} + \sum_{k=1,k\neq i}^{n} X_{kj}(-1)^{k+j} A_{kj}^{X}.$$

Because $A_{ij}^{X} = 1$ and $\mathbf{X}$ satisfies

$$X_{ij} = (-1)^{i+j}\left(1 - \sum_{k=1,k\neq i}^{n} X_{kj}(-1)^{k+j} A_{kj}^{X}\right).$$

Then

$$|\mathbf{X}| = X_{ij}(-1)^{i+j} A_{ij}^{X} + 1 - X_{ij}(-1)^{i+j} = X_{ij}(-1)^{i+j} 1 + 1 - X_{ij}(-1)^{i+j} = 1.$$

Therefore, $\mathbf{X}$ is an nD Cat matrix. ■

Appendix B
Proof of Theorem 2

Here proves Theorem 2 in Section IV-B.

Proof: The determinant of matrix $\mathbf{G}$ can be expanded as

$$|\mathbf{G}| = \sum_{p=1}^{n} G_{pq}(-1)^{p+q} A_{ij}^{G} = G_{ij}(-1)^{i+j} A_{ij}^{G} + \sum_{k=1,k\neq i}^{n} G_{kj}(-1)^{k+j} A_{kj}^{G}.$$

Because $G_{ij} = 0$ and for arbitrary elements G_{pq} in auxiliary matrix $\mathbf{G}$ with either $p \neq i$ or $q \neq j$, we have $G_{pq} = X_{pq}$, the above equation can be rewrote as

$$|\mathbf{G}| = 0(-1)^{i+j} A_{ij}^{G} + \sum_{k=1,k\neq i}^{n} X_{kj}(-1)^{k+j} A_{kj}^{X} = \sum_{k=1,k\neq i}^{n} X_{kj}(-1)^{k+j} A_{kj}^{X}.$$

Substitute this equation to (12), we then obtain (16). ■

References

[1] 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, pp. 1430027–14, 2014.

[2] 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.

[3] S. Y. Li, L. M. Tam, S. E. Tsai, and Z. M. Ge, "Novel fuzzy modeling and synchronization of chaotic systems with multinonlinear terms by advanced Ge-Li fuzzy model," *IEEE Trans. Cybern.*, to be published, 2015

[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. Forensic Security*, vol. 8, no. 12, pp. 2177–2185, Dec. 2013.

[5] R. Law, D. J. Murrell, and U. Dieckmann, "Population growth in space and time: Spatial logistic equations," *Ecology*, vol. 84, no. 1, pp. 252–262, 2003.

[6] N. Boccara, "Van der pol oscillator," *Essentials of Mathematica: With Applications to Mathematics and Physics*. New York, NY, USA: Springer, 2007, pp. 505–508.

[7] H.-K. Chen and C.-I. Lee, "Anti-control of chaos in rigid body motion," *Chaos Solitons Fractals*, vol. 21, no. 4, pp. 957–965, 2004.

[8] 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.

[9] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Cambridge, MA, USA: Perseus, 2001.

[10] H.-J. Stöckmann, *Quantum Chaos: An Introduction*. New York, NY, USA: Cambridge Univ. Press, 2006.

[11] M. Shen, W.-N. Chen, J. Zhang, H. S.-H. Chung, and O. Kaynak, "Optimal selection of parameters for nonuniform embedding of chaotic time series using ant colony optimization," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 790–802, Apr. 2013.

[12] 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.

[13] 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, Dec. 2010.

[14] Y. Zhou, L. Bao, and C. L. P. Chen, "Image encryption using a new parametric switching chaotic system," *Signal Process.*, vol. 93, no. 11, pp. 3039–3052, 2013.

[15] S.-L. Chen, 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.

[16] Z. Li, K. Li, C. Wen, and Y. C. Soh, "A new chaotic secure communication system," *IEEE Trans. Commun.*, vol. 51, no. 8, pp. 1306–1312, Aug. 2003.
[17] 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.
[18] S. Chen and H. Leung, "Ergodic chaotic parameter modulation with application to digital image watermarking," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1590–1602, Oct. 2005.
[19] T. Habutsu, Y. Nishio, I. Sasase, and S. Mori, *A Secret Key Cryptosystem by Iterating a Chaotic Map*. Berlin, Germany: Springer, 1991, pp. 127–140.
[20] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcat. Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998.
[21] D. Xiao, X. F. Liao, and S. J. Deng, "One-way hash function construction based on the chaotic map with changeable-parameter," *Chaos Solitons Fractals*, vol. 24, no. 1, pp. 65–71, 2005.
[22] V. Patidar, K. K. Sud, and N. K. Pareek, "A pseudo random bit generator based on chaotic logistic map and its statistical testing," *Inform. Slovenia*, vol. 33, no. 4, pp. 441–452, 2009.
[23] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, pp. 172–182, Apr. 2014.
[24] 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.
[25] J. Ford, G. Mantica, and G. Ristow, "The Arnol'd cat: Failure of the correspondence principle," *Phys. D Nonlin. Phenom.*, vol. 50, no. 3, pp. 493–520, 1991.
[26] L. Kocarev and S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*, vol. 354. Berlin, Germany: Springer, 2011.
[27] X. Tong and M. Cui, "Image encryption with compound chaotic sequence cipher shifting dynamically," *Image Vis. Comput.*, vol. 26, no. 6, pp. 843–850, 2008.
[28] G. Gaspari, "The Arnold cat map on prime lattices," *Phys. D Nonlin. Phenom.*, vol. 73, no. 4, pp. 352–372, 1994.
[29] F. Chen, K.-W. Wong, X. Liao, and T. Xiang, "Period distribution of the generalized discrete Arnold cat map for $N = 2^e$," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 3249–3255, May 2013.
[30] I. Antoniou, B. Qiao, and Z. Suchanecki, "Generalized spectral decomposition and intrinsic irreversibility of the Arnold Cat Map," *Chaos Solitons Fractals*, vol. 8, no. 1, pp. 77–90, 1997.
[31] S. Weigert, "Quantum chaos in the configurational quantum cat map," *Phys. Rev. A*, vol. 48, no. 3, pp. 1780–1798, 1993.
[32] A. Lakshminarayan and N. Balazs, "On the quantum cat and sawtooth maps—Return to generic behaviour," *Chaos Solitons Fractals*, vol. 5, no. 7, pp. 1169–1179, 1995.
[33] S. P. Kuznetsov, "Disheveled Arnold's cat and the problem of quantum-classic correspondence," *Phys. D Nonlin. Phenom.*, vol. 137, no. 3, pp. 205–227, 2000.
[34] S. V. Neshveyev, "On the K-property of quantized Arnold cat maps," *J. Math. Phys.*, vol. 41, no. 4, pp. 1961–1965, 2000.
[35] A. M. F. Rivas, M. Saraceno, and A. M. Ozorio de Almeida, "Quantization of multidimensional cat maps," *Nonlinearity*, vol. 13, no. 2, p. 341, 2000.
[36] L. Barash and L. N. Shchur, "Periodic orbits of the ensemble of Sinai-Arnold cat maps and pseudorandom number generation," *Phys. Rev. E*, vol. 73, no. 3, 2006, Art. ID 036701.
[37] S. Lian, Y. Mao, and Z. Wang, "3D extensions of some 2D chaotic maps and their usage in data encryption," in *Proc. 4th Int. Conf. Control Autom.*, Montreal, QC, Canada, 2003, pp. 819–823.
[38] H. Liu, Z. Zhu, H. Jiang, and B. Wang, "A novel image encryption algorithm based on improved 3D chaotic cat map," in *Proc. 9th Int. Conf. Young Comput. Sci.*, Hunan, China, 2008, pp. 3016–3021.
[39] T. G. Pan and D. Y. Li, "A new algorithm of image encryption based on 3D Arnold Cat," in *Adv. Eng. Forum*, vol. 1, pp. 183–187, Sep. 2011.
[40] M. Falcioni, L. Palatella, S. Pigolotti, and A. Vulpiani, "Properties making a chaotic system a good pseudo random number generator," *Phys. Rev. E*, vol. 72, no. 1, 2005, Art. ID 016220.
[41] K. W. Tang and W. K. S. Tang, "A chaos-based secure voice communication system," in *Proc. IEEE Int. Conf. Ind. Technol.*, Hong Kong, 2005, pp. 571–576.
[42] J. Nance, "Periods of the discretized Arnold Cat Map and its extension to n dimensions," [Online]. Available: http://arxiv.org/pdf/1111.2984v1.pdf, 2013.
[43] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcat. Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.
[44] N. J. Corron and S. D. Pethel, "Control of long-period orbits and arbitrary trajectories in chaotic systems using dynamic limiting," *Chaos Interdiscipl. J. Nonlin. Sci.*, vol. 12, no. 1, pp. 1–7, 2002.
[45] J. A. Storer, *An Introduction to Data Structures and Algorithms*. Basel, Switzerland: Springer, 2001.
[46] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Physica*, vol. 16, no. 3, pp. 285–317, 1985.
[47] K. J. Persohn and R. J. Povinelli, "Analyzing logistic map pseudo-random number generators for periodicity induced by finite precision floating-point representation," *Chaos Solitons Fractals*, vol. 45, no. 3, pp. 238–245, 2012.
[48] A. M. Davie and A. J. Stothers, "Improved bound for complexity of matrix multiplication," *Proc. Roy. Soc. Edin. Sect. A Math.*, vol. 143, no. 2, pp. 351–369, 2013.
[49] P. Faure and H. Korn, "A new method to estimate the Kolmogorov entropy from recurrence plots: Its application to neuronal signals," *Phys. D Nonlin. Phenom.*, vol. 122, no. 1, pp. 265–279, 1998.
[50] J. A. C. Gallas, "Structure of the parameter space of the Hénon map," *Phys. Rev. lett.*, vol. 70, no. 18, p. 2714, 1993.
[51] *IEEE Standard for Floating-Point Arithmetic*, IEEE Standard 754-2008, 2008.
[52] G. Marsaglia. *Diehard Statistical Tests*. [Online]. Available: http://www.stat.fsu.edu/pub/diehard/, accessed Sep. 22, 2015.
[53] P. L'Ecuyer and R. Simard, "Testu01: A C library for empirical testing of random number generators," *ACM Trans. Math. Softw.*, vol. 33, no. 4, p. 22, 2007.

Yue Wu (S'08–M'12) received the B.E. degree in telecommunication engineering from the Huazhong University of Sciences and Technologies, Hubei, China, in 2001, the M.Sc. degree in applied mathematics from the University of Toledo, Toledo, OH, USA, in 2008, and the Ph.D. degree in electrical engineering from Tufts University, Medford, MA, USA, in 2012.

He was a Scientist with the Raytheon BBN Technologies, Cambridge, U.K. He is currently a Computer Scientist with the Information Sciences Institute, University of Southern California Viterbi School of Engineering. His current research interests include information security, image processing, pattern recognition, and their real applications.

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

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 Institute for Scientific Information (ISI) 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 price of the Macau Natural Science Award in 2014. He is a member of the International Society for Optical Engineers and the Association for Computing Machinery.