Natural Computing (2023) 22:477–491
https://doi.org/10.1007/s11047-022-09930-1

Enumeration of maximal cycles generated by orthogonal cellular automata

Luca Mariot[1]

Accepted: 13 November 2022 / Published online: 26 November 2022

Abstract
Cellular automata (CA) are an interesting computational model for designing pseudorandom number generators (PRNG), due to the complex dynamical behavior they can exhibit depending on the underlying local rule. Most of the CA-based PRNGs proposed in the literature, however, suffer from poor diffusion since a change in a single cell can propagate only within its neighborhood during a single time step. This might pose a problem especially when such PRNGs are used for cryptographic purposes. In this paper, we consider an alternative approach to generate pseudorandom sequences through *orthogonal CA* (OCA), which guarantees a better amount of diffusion. After defining the related PRNG, we perform an empirical investigation of the maximal cycles in OCA pairs up to diameter $d = 8$. Next, we focus on OCA induced by linear rules, giving a characterization of their cycle structure based on the rational canonical form of the associated Sylvester matrix. Finally, we devise an algorithm to enumerate all linear OCA pairs characterized by a single maximal cycle, and apply it up to diameter $d = 16$ and $d = 13$ for OCA respectively over the binary and ternary alphabets.

Keywords Cellular automata · Latin squares · Pseudorandom number generators · Multipermutation · Sylvester matrices · Polynomials

Mathematics Subject Classification 05B15 · 68Q80 · 37B15 · 11T06

1 Introduction

Consider the following game: we are given a $N \times N$ square, where each cell is labelled by a pair of numbers (i, j) with $i, j \in \{1, \ldots, N\} = [N]$. Moreover, we assume that each of the N^2 pairs in the Cartesian product $[N] \times [N]$ occurs exactly once as a label in the square. Our only move is to choose an initial cell; after that, we read the corresponding label (i, j), and use it as the new row and column coordinates of the cell where to jump next. The process is then iterated until we jump back to the initial cell, which is granted by the assumption that the cells' labeling is a permutation of $[N] \times [N]$. The goal of the game is to achieve the highest score, defined as the number of distinct cells visited before returning to the initial one. Figure 1 depicts an example of 4×4 square where choosing any initial cell except the top left one always yields the highest score, which is 15 in this case.

Given the above rules, "winning" the game depends on two factors: (1) the cycle structure of the permutation that defines the cells' labeling, and (2) the initial cell where we start from. Clearly, there is a trade-off between these two aspects: the more the cells' labeling permutation is composed of few cycles having a large length, the less the position of the initial cell matters to reach a high score. Figure 1 represents an extreme case, where the permutation is made only of a single large cycle of length $2^N - 1$ and a fixed point.

Suppose now that we add a further constraint on the labels: beside representing a permutation of the Cartesian product $[N] \times [N]$, we also require that the two projections are *Latin squares* of order N. This means that if we consider only the left (respectively, the right) coordinate of each label, we obtain a square where each number from 1 to N occurs exactly once in each row and column. This is indeed the case of the square in Fig. 1, with the Latin

✉ Luca Mariot
luca.mariot@ru.nl

[1] Digital Security Group, Radboud University, PO Box 9010, 6500 GL Nijmegen, The Netherlands

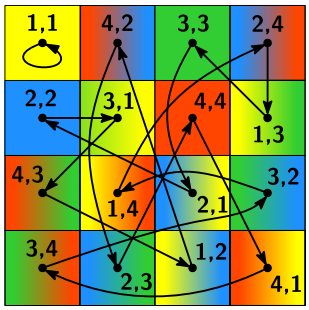


Fig. 1 Example of square game. Choosing any initial cell other than the one at the top left corner gives a maximum score of 15

squares corresponding to the left and right coordinates depicted in Fig. 2.

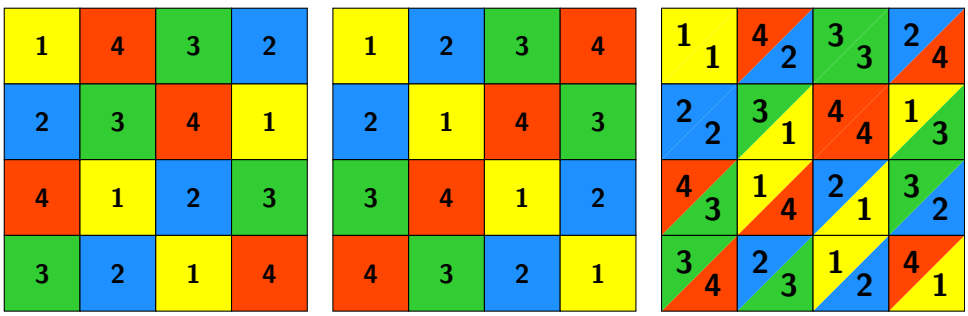
Pairs of Latin squares of this kind (that is, whose superposition gives a permutation over the Cartesian product of possible entries) are also called *orthogonal*.

Although the game described above seems quite detached from any real-world setting at a first glance, there are several applications for it in *cryptography*, particularly in the context of *pseudorandom number generators* (PRNGs). Indeed, the initial cell can be thought of as the seed of a PRNG, with the generated keystream being the sequence of labels encountered along the path where the seed lies. A desirable property in PRNGs is to generate sequences of large periods, which is related to the game's goal of reaching a high score. The fact that the cells' labels define a permutation further ensures that the dynamics of the game is *invertible*, which is useful in the context of *block ciphers* for decryption purposes. Finally, having a permutation defined by a pair of orthogonal Latin squares guarantees a certain amount of *diffusion*, a paramount property for stream and block ciphers to frustrate statistical attacks. As a matter of fact, orthogonal Latin squares correspond to a particular kind of *multipermutation*, which are a useful cryptographic primitive when designing the diffusion layer of a block cipher (Vaudenay 1994).

The aim of this paper is to investigate the dynamics of the above game when the two orthogonal Latin squares are defined by *Cellular Automata* (CA). In general, CA represent an attractive approach to design PRNGs for cryptographic purposes, for a twofold reason. First, CA can exhibit a very complex dynamical behavior depending on the underlying local rule, which can be exploited to generate pseudorandom sequences that are hard to predict. Second, the shift-invariance that characterizes CA lends itself to very efficient implementations, both in hardware and software.

Wolfram was the first researcher to propose the use of one-dimensional CA to generate pseudorandom sequences for Vernam-like stream ciphers (Wolfram 1985). His idea was to initialize a CA with a random configuration (representing the PRNG's seed) and then iterate the CA for many time steps, taking the trace of the CA's central cell as a pseudorandom keystream. According to Wolfram's claims, the unpredictability of the keystream stemmed from the chaotic dynamics induced by the CA, equipped with rule 30. Unfortunately, later research showed that Wolfram's PRNG is in fact very weak, showing attacks to both recover the initial configuration of the CA (Meier and Staffelbach 1991) and invert its iterations (Koç and Apohan 1997). Martin (2008) remarked that some of the weaknesses of this PRNG can be traced back to the poor cryptographic properties of rule 30 when interpreted as a Boolean function. For this reason, more recent works Formenti et al. (2014); Leporati and Mariot (2014) focused on searching larger local rules with a better trade-off of cryptographic properties, using various combinatorial search methods. Still, this research thread does not consider another serious issue when using CA to generate pseudorandom sequences as originally meant by Wolfram: as identified already by Daemen et al. (1994), CA always have poor diffusion, due to the local nature of the model that does not allow information to spread very far in a single iteration.

Consequently, studying the dynamics of CA that generate orthogonal Latin squares (also called *orthogonal CA*, or OCA) can be regarded as an alternative approach that starts to address the diffusion issue of classic CA-based PRNGs. In this paper, we perform a preliminary



(a) Left coordinate (b) Right coordinate (c) Superposition

Fig. 2 Decomposition in orthogonal Latin squares

investigation of OCA pairs yielding sequences of maximal period, especially focusing on the case where the underlying local rules are linear.

This work is an extended version of the paper "*Hip to Be (Latin) Square: Maximal Period Sequences from Orthogonal Cellular Automata*" presented by the author at CANDAR 2021 (Mariot 2021). In particular, the new and improved contributions with respect to the conference version are summarized as follows:

1. We extend the exhaustive search experiments on the distribution of maximal periods for OCA pairs defined over $\mathbb{F}_2$ up to diameter 6, leveraging on the combinatorial algorithm described in Mariot et al. (2017a) to efficiently enumerate the search space. We also extend this investigation to the OCA pairs of diameter 7 and 8 constructed in Mariot et al. (2017b) by means of evolutionary algorithms. The new results fix an inaccurate claim in the findings of our conference paper, i.e. that the highest maximal period of $2^{2n} - 1$ is achievable only by linear OCA pairs. Indeed, the correct results show that there are also maximal period OCA defined by nonlinear rules already from diameter $d = 5$.
2. Leveraging on the theory of *Linear Modular Systems* (LMS), we describe a method to compactly represent the cycle structure of linear OCA pairs. Such a method is based on the computation of the *rational canonical form* of a Sylvester matrix, and allows us to find a simple condition to check whether an OCA pair can attain maximal period. This boils down to verify if the minimal polynomial of the Sylvester matrix is primitive, and it is equivalent (but more efficient, as shown below) to the previous theoretical result of Mariot (2021), where a method to determine the upper bound on the maximal period was given in terms of Lagrange's theorem.
3. Based on the above primitivity check, we devise a much more time-efficient algorithm to enumerate all linear OCA pairs of maximal period over $\mathbb{F}_2$, implementing it in Magma. In this way, we are able to enumerate all such pairs up to diameter $d = 16$ in a bit less than one hour, a significant gain over the algorithm used in Mariot (2021), which took almost five days to arrive only up to $d = 11$. The downside of this new algorithm, on the other hand, is its memory usage, with approximately 25 GB required to reach $d = 16$. Incidentally, we also fix the counts of Table II of Mariot (2021), which were wrong due to an implementation bug, and we provide also the numbers of maximal linear OCA pairs over the alphabet $\mathbb{F}_3$, up to diameter $d = 13$.

The rest of this paper is structured as follows. Section 2 covers all preliminary definitions related to CA and orthogonal Latin squares, which are necessary to introduce the main results in the next sections. Section 3 formally defines the dynamical system based on a pair of OCA, and shows the empirical distributions of the maximum periods up to diameter $d = 8$. Section 4 focuses on linear OCA pairs, providing a characterization of their periods in terms of the rational canonical form of the underlying Sylvester matrix. Next, Sect. 5 presents an improved algorithm to enumerate all linear OCA pairs with maximal period of a given diameter, and reports the results up to $d = 16$ and $d = 12$ for OCA respectively over the binary and ternary alphabets. Finally, Sect. 6 sums up the key contributions of the paper, and discusses some directions for future research on the subject.

2 Preliminaries

In this section, we first recall some basic notions about the cellular automata (CA) model used in the rest of this paper. We then summarize the main results from the relevant literature related to the construction of orthogonal Latin squares by means of bipermutive CA. As a general notation, for any $n \in \mathbb{N}$ we denote by $[N] = \{1, \ldots, N\}$ the set of all positive integer numbers smaller than or equal to N. Further, given $q = p^a$ with p a prime number and $a \in \mathbb{N}$, we use $\mathbb{F}_q$ to denote the finite field of order q, with $+$ and $\cdot$ standing respectively for the sum and multiplication operations. In particular, when $q = 2$ the sum coincides with the XOR (denoted as $\oplus$), while the multiplication is the logical AND. For any $n \in \mathbb{N}$ we denote the n-dimensional vector space over $\mathbb{F}_q$ by $\mathbb{F}_q^n$, with vector sum and multiplication by a scalar induced by the ground field operations in the usual way. Finally, given the finite field $\mathbb{F}_q$, the ring of polynomials in the indeterminate X with coefficients in $\mathbb{F}_q$ is denoted as $\mathbb{F}_q[X]$.

2.1 Cellular automata

Cellular automata are one of the oldest natural computing models studied in the literature, and they generally consist of a regular lattice of cells, whose states take values over a finite alphabet. Each cell updates its state in parallel according to the same local rule evaluated over the corresponding neighborhood. Most of the research in this field concerns the *long-term* behavior and properties of CA, which in this case are considered as a particular type of discrete-time dynamical systems. This usually leads to the setting where the cellular lattice is infinite, and a CA can be characterized as a shift-invariant transformation over the

full-shift space which is uniformly continuous with respect to the Cantor distance (Hedlund 1969). In concrete simulations of CA, the lattice must of course be finite, implying that the long-term dynamics is always ultimately periodic. For our paper, we consider a finite model that is even more constrained. In particular, we focus only on the *short-term* behavior of finite CA, often by just considering a single application of the global rule.

Formally, we define the following model of one-dimensional *No-Boundary CA* (NBCA), which is adopted in Mariot et al. (2020) to introduce the CA-based construction of orthogonal Latin squares:

Definition 1 A *No-Boundary CA* is a vectorial function $F : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^{n-d+1}$ defined by a *local rule* $f : \mathbb{F}_q^d \rightarrow \mathbb{F}_q$ of *diameter* $d \leq n$, where

$$F(x_1, \ldots, x_n) = (f(x_1, \ldots, x_d), \ldots, f(x_{n-d+1}, \ldots, x_n)) \quad (1)$$

for all $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$.

From a practical perspective, the output coordinate $i \in [n - d + 1]$ of a CA is determined by evaluating the local rule f on the *neighborhood* formed by the i-th input cell and the $d - 1$ cells to its right. The CA is called no-boundary since the local rule is applied only until the coordinate $n - d + 1$, as the remaining ones do not have enough neighbors to their right. Clearly, this implies that the global rule of a NBCA can be iterated as long as there are at least d cells remaining in the current cellular array. As we mentioned above, this does not pose an issue since we will be mostly interested in the short-term behavior arising from a single application of the global rule. In this way, we can effectively identify a CA with the vectorial function F. For other CA models that also contemplate boundary conditions, we refer the reader to Kari (2005).

CA are usually considered over the binary alphabet, i.e. with $q = 2$. In this case, the local rule can be interpreted as a d-variable *Boolean function* $f : \mathbb{F}_2^d \rightarrow \mathbb{F}_2$, and the most common way to represent it is by means of its *truth table*. In particular, the truth table of f is defined as

$$\Omega_f = (f(0, \ldots, 0), f(0, \ldots, 1), \ldots, f(1, \ldots, 1)).$$

Stated otherwise, Ω_f is the vector that lists the value of f for all 2^d input vectors in $\mathbb{F}_2^d$, assuming they are sorted in lexicographic order. The *Wolfram code* of rule f corresponds to the decimal encoding of the truth table Ω_f.

A second common method to uniquely identify a Boolean function is the *Algebraic Normal Form* (ANF). Considering that any element x is *idempotent* over $\mathbb{F}_2$ (i.e., $x^2 = x$), the ANF is the following multivariate polynomial over the quotient ring $\mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2 \oplus x_1, \ldots, x_n^2 \oplus x_n)$:

$$P_f(x) = \bigoplus_{I \in 2^{[n]}} a_I \left(\prod_{i \in I} x_i \right), \quad (2)$$

where $2^{[n]}$ denotes the power set of $[n] = \{1, \ldots, n\}$. The *algebraic degree* of f is defined in the natural way, i.e. as the number of terms in the largest nonzero monomial of its ANF, or formally as the cardinality of the largest subset $I \in 2^{[n]}$ such that $a_I \neq 0$. Functions of degree at most 1 are called *affine*, and affine functions whose ANF have a null constant term are called *linear*. When a binary CA is defined by a linear local rule, the next state of each cell is basically an XOR of a subset of cells in its neighborhood.

Figure 3a depicts an example of CA with $n = 8$ input cells, induced by the linear local rule of diameter $d = 3$ with ANF $f(x_1, x_2, x_3) = x_1 \oplus x_3$, i.e. only the $i - th$ and $(i + 2) - th$ cell in the neighborhood are XORed together. The Wolfram code of this rule is 90, since it corresponds to the decimal encoding of the truth table (0, 1, 0, 1, 1, 0, 1, 0), which is reported in Fig. 3b.

Further information on the ANF of Boolean functions may be found in Carlet's recent book (Carlet 2021). In what follows, we will develop our theoretical results for CA over a generic finite field $\mathbb{F}_q$, although our empirical results and examples will mostly refer to the binary case.

2.2 Orthogonal Latin squares from cellular automata

Let us turn our attention to Latin squares, starting from the following definition:

Definition 2 A *Latin square* of order $N \in \mathbb{N}$ is a $N \times N$ matrix L with entries in $[N]$, such that the following two conditions hold:

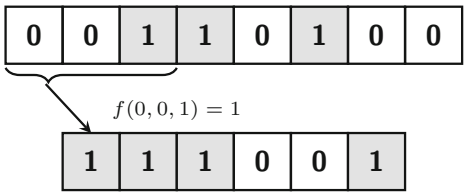
1. $L(i, j_1) \neq L(i, j_2)$ for each row coordinate $i \in [N]$ and column coordinates $j_1, j_2 \in [N]$ with $j_1 \neq j_1$.
2. $L(i_1, j) \neq L(i_2, j)$ for each column coordinate $j \in [N]$ and row coordinates $i_1, i_2 \in [N]$ with $i_1 \neq i_1$.

Intuitively, each number from 1 to N occurs exactly once in each row and in each column of a Latin square of order N; equivalently, each row and each column forms a permutation of $[N]$. The concept of orthogonality is defined in terms of the superposition of two Latin squares:

Definition 3 Two Latin squares of order L_1, L_2 of order $N \in \mathbb{N}$ are called *orthogonal* if for any $(i_1, j_1) \neq (i_2, j_2)$ with $i_1, j_1, i_2, j_2 \in [N]$ it holds that

$$(L_1(i_1, j_1), L_2(i_1, j_1)) \neq (L_1(i_2, j_2), L_2(i_2, j_2)).$$

Equivalently, L_1 and L_2 are orthogonal if the map $H : [N] \times [N] \rightarrow [N] \times [N]$ defined as $H(i, j) = (L_1(i, j), L_2(i, j))$ for all $i, j \in [N]$ is bijective.



(a) Local rule evaluation.

x_i, x_{i+1}, x_{i+2}	$f(x_i, x_{i+1}, x_{i+2})$
000	0
001	1
010	0
011	1
100	1
101	0
110	1
111	0

(b) Truth table of rule 90.

Fig. 3 Example of computation in a CA of length $n = 8$ equipped with the linear local rule 90 of diameter $d = 3$

From an intuitive point of view, two Latin squares are orthogonal if and only if their *superposition* yields every order pair (i, j) in the Cartesian product $[N] \times [N]$ exactly once. Figure 2 in the introductory section of this paper depicts an example of two orthogonal Latin squares of order 4.

Despite their simple definition, Latin squares spawned a very broad research field, also due to their numerous applications in statistics, cryptography and coding theory. There exist a few known constructions for families of Mutually Orthogonal Latin Squares (MOLS) in the literature, a good account of which can be found in Keedwell and Denes's book (Keedwell and Dénes 2015).

The use of CA to construct orthogonal Latin squares was originally suggested in Mariot et al. (2020), with the original goal of designing a threshold *Secret Sharing Scheme* (SSS). A (k, n)-threshold SSS is a protocol that enables a dealer to share a secret value S among a set of n participants, in such a way that at least k participants must combine their respective shares in order to uniquely recover S. All coalitions of less than k participants, on the contrary, gain no information on the value of S (Shamir 1979). It can be shown that families of n MOLS are equivalent to $(2, n)$-threshold SSS (see e.g. Stinson 2004). Most of the SSS based on CA previously published in the literature, on the other hand, feature a *sequential threshold*, meaning that the k shares required to recover the secret must also be adjacent with respect to the order of the participants (del Rey et al. 2005; Mariot and Leporati 2014; Herranz and Sáez 2018).

The authors of Mariot et al. (2020) showed how to generate orthogonal Latin squares with CA, which have later been named *orthogonal CA* (OCA) in Mariot and Leporati (2018). The construction entails two steps: first, one needs to determine how to define a Latin square from a no-boundary CA. This can be done in a rather natural way by considering CA with *bipermutive local rules*, which we define below:

Definition 4 A local rule $f : \mathbb{F}_q^d \to \mathbb{F}_q$ is *bipermutive* if by fixing the leftmost or the rightmost $d - 1$ coordinates to any vector $\tilde{x} \in \mathbb{F}_q^{d-1}$, the resulting left and right restrictions $f_{l,\tilde{x}} : \mathbb{F}_q \to \mathbb{F}_q$ and $f_{r,\tilde{x}} : \mathbb{F}_q \to \mathbb{F}_q$ respectively defined for all $x' \in \mathbb{F}_q$ as:

$$f_{l,\tilde{x}}(x) = f(\tilde{x}_1, \ldots, \tilde{x}_{d-1}, x')$$
$$f_{r,\tilde{x}}(x) = f(x', \tilde{x}_1, \ldots, \tilde{x}_{d-1})$$

are permutations of $\mathbb{F}_q$.

Remark that for $q = 2$ a local rule $f : \mathbb{F}_2^d \to \mathbb{F}_2$ is bipermutive if and only if there exists a $(d - 2)$-variable function $g : \mathbb{F}_2^{d-2} \to \mathbb{F}_2$ such that

$$f(x) = x_1 \oplus g(x_2, \ldots, x_{d-1}) \oplus x_d$$

for all input vectors $x = (x_1, x_2, \ldots, x_{d-1}, x_d) \in \mathbb{F}_2^d$. In other words, rule f depends in a linear way from the leftmost and rightmost variables, since they are independently XORed with a function of the central $d - 2$ coordinates. For this reason, CA equipped with bipermutive local rules are also called *quasilinear* in the related literature (Moore 1997). Rule 90 used in the example of Fig. 3 is bipermutive with $g : \mathbb{F}_2 \to \mathbb{F}_2$ being the zero function.

Consider now a NBCA $F : \mathbb{F}_q^{2(d-1)} \to \mathbb{F}_q^{d-1}$ with a local rule of diameter d. Since the input vector is twice the size of the output, we can use the CA to build a square matrix S_F of size $N \times N$ with $N = q^{d-1}$ as follows. Given $x, y \in \mathbb{F}_q^{d-1}$, their concatenation $x \| y \in \mathbb{F}_q^{2(d-1)}$ is used as an input vector for the CA. Then, the output $F(x \| y)$ computed by the CA is the entry of the square S_F where x and y represent respectively the row and column coordinates. From a formal point of view, assume that $\phi : \mathbb{F}_q^{d-1} \to [N]$ is a one-to-one mapping from the vectors of $\mathbb{F}_q^{d-1}$ to the set of the first N positive natural numbers, with $\psi : [N] \to \mathbb{F}_q^{d-1}$ denoting the inverse mapping. Then, for all $i, j \in [N]$ the entry of S_F at row i and column j is defined as:

$$S_F(i,j) = \phi(F(\psi(i)\|\psi(j))). \quad (3)$$

Eloranta (1993) and Mariot et al. (2016) independently proved the following sufficient condition for the square S_F to be Latin:

Lemma 1 *Let $F : \mathbb{F}_q^{2(d-1)} \to \mathbb{F}_q^{d-1}$ be a NBCA defined by a bipermutive local rule $f : \mathbb{F}_q^d \to \mathbb{F}_q$ of diameter d. Then, the square S_F in Eq. (3) is a Latin square of order $N = q^{d-1}$.*

As an example, Fig. 4 shows the Latin square of order $N = 4$ associated to the CA $F : \mathbb{F}_2^4 \to \mathbb{F}_2^2$ with bipermutive local rule 150, whose ANF is defined as $f_{150}(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$. In this case, the mapping $\phi : \mathbb{F}_2^2 \to [N]$ is given by $\phi(0,0) = 1$, $\phi(1,0) \mapsto 2$, $\phi(0,1) = 3$, and $\phi(1,1) = 4$. The inverse mapping is thus $\psi(1) = (0,0)$, $\psi(2) = (1,0)$, $\psi(3) = (0,1)$ and $\psi(4) = (1,1)$.

After figuring out how Latin squares can be constructed through CA, the next step in the construction is to determine when their superposition yields an orthogonal pair. To this end, the characterization of OCA proved by the authors of Mariot et al. (2020) considers bipermutive local rules that are also linear. We have already introduced above linear rules for the binary alphabet, as a particular case of the ANF of Boolean functions. For a generic finite field $\mathbb{F}_q$ a linear rule $f : \mathbb{F}_q^d \to \mathbb{F}_q$ is defined similarly, i.e. as the following linear combination:

$$f(x_1, \ldots, x_d) = a_1 x_1 + \cdots + a_d x_d \quad (4)$$

for all $x \in \mathbb{F}_q^d$, where $a_i \in \mathbb{F}_q$ for $i \in [d]$. Notice that f is bipermutive if and only if the leftmost and rightmost coefficients are not null, that is $a_1 \neq 0$ and $a_d \neq 0$. It is possible to associate a polynomial of degree $n = d - 1$ with coefficients in $\mathbb{F}_q$ to a linear rule as follows:

$$f \mapsto P_f(X) = a_1 + a_2 X + \cdots + a_d X^n \quad (5)$$

Hence, we simply use the coefficients of the linear rule reported in Eq. (4) as the coefficients of the indeterminate's increasing powers.

The characterization of linear OCA proved in Mariot et al. (2020) can be stated as follows:

Theorem 2 *Let $F, G : \mathbb{F}_q^{2n} \to \mathbb{F}_q^n$ be two NBCA defined by linear bipermutive local rules $f, g : \mathbb{F}_q^d \to \mathbb{F}_q$ of diameter d, with $n = d - 1$. Then, the two Latin squares of order $N = q^n$ generated by F and G are orthogonal if and only if the polynomials $P_f(X), P_g(X) \in \mathbb{F}_q[X]$ of degree n respectively associated to f and g are coprime.*

Therefore, given two linear bipermutive rules f, g of diameter d, it suffices to compute the greatest common divisor of the two associated polynomials P_f and P_g. By Theorem 2, the Latin squares S_f and S_g are orthogonal if and only if the GCD of P_f and P_g is 1.

As an example, the two polynomials over $\mathbb{F}_2$ associated to the local rules 90 and 150 of diameter $d = 3$ are respectively $P_f(X) = X^2 + 1$ and $P_g(X) = X^2 + X + 1$. Clearly one has $\gcd(P_f, P_g) = 1$ since P_g is irreducible, and thus the corresponding Latin squares S_F and S_G of order 4 are orthogonal. Indeed, these squares are depicted in the example of Fig. 2 featured in the Introduction.

3 Dynamical systems induced by OCA

In this section, we start by formally defining the dynamical system to generate pseudorandom sequences by employing a pair of OCA. Next, we present the results of our empirical search experiments on the maximum periods attainable by such a system. These include both an exhaustive search approach up to diameter $d = 6$, and an analysis of OCA constructed with evolutionary algorithms (EA) for diameters $d = 7$ and $d = 8$.

3.1 Formalization and problem statement

We discussed earlier in Sect. 2.1 that a no-boundary CA can be iterated only for a finite number of steps, due to the fact that the size of the cellular array shrinks by $d - 1$ cells after each evaluation of the global rule F. Hence, although a NBCA equipped with a bipermutive local rule generates a

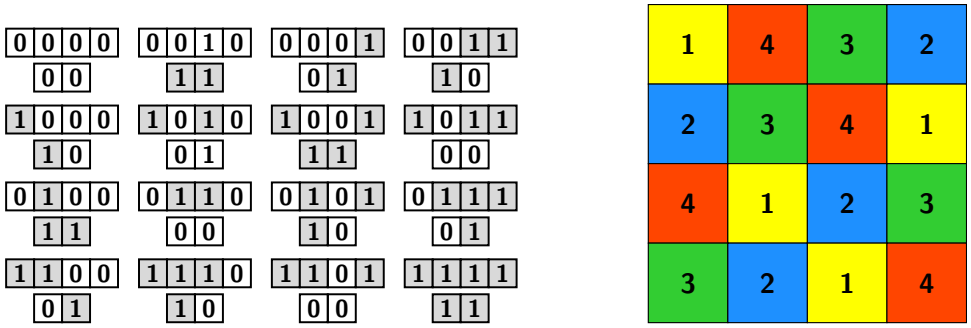


Fig. 4 Example of Latin square generated by the CA with local rule 150

Latin square on account of Lemma 1, it is not possible to use it directly for the generation of pseudorandom sequences. Indeed, a pseudorandom number generator can be viewed as a discrete-time dynamical system $\mathcal{S} = \langle A, f \rangle$ where A is a (finite) set representing the *phase space* of the system, while $f : A \to A$ is an endofunction which maps the current state $s(t) \in A$ at time step $t \in \mathbb{N}$ into the next one $s(t+1) \in A$ at time step $t+1$.[1]

For this reason, the main idea of our pseudorandom generator is to take a *pair* of local rules $f, g : \mathbb{F}_q^d \to \mathbb{F}_q$, instead of a single one. Both rules are applied to the same initial configuration s of length $2n = 2(d-1)$, as in the case of orthogonal Latin squares. Hence, one obtains two output vectors $z = F(s)$, $w = G(s)$ of length n, generated by the NBCA F and G respectively defined by f and g. Next, we construct a new configuration of length $2n$ by concatenating z and w. Therefore, the outputs of the NBCA F, G are used respectively as a new row and a new column coordinate, which will in turn point to a new pair of entries given by F and G. Considering the superposed representation of the Latin squares generated by F and G, this operation can be conceived as starting from the pair of entries occurring at the coordinates indexed by the initial configuration s, and using them as the new coordinates where to "jump" next (see Fig. 1 in the Introduction).

We now formally define the dynamical system $\mathcal{S}$ intuitively described above.

Definition 5 Let $d \in \mathbb{N}$ with $d > 1$ and $n = d - 1$, and let $q = p^a$ be a power of a prime number. Additionally, let $F, G : \mathbb{F}_q^{2n} \to \mathbb{F}_q^n$ be two OCA defined by bipermutive local rules of diameter d. Then, the dynamical system induced by F and G is defined as $\mathcal{S} = \langle A, H \rangle$, where:

- $A = \mathbb{F}_q^{2n}$, i.e. the phase space is the $2n$-dimensional vector space over $\mathbb{F}_q$.
- $H : \mathbb{F}_q^{2n} \to \mathbb{F}_q^{2n}$ is the update function defined for all $x, y \in \mathbb{F}_q^n$ as:

$$H(x\|y) = F(x\|y)\|G(x\|y), \tag{6}$$

with $\|$ denoting the concatenation of two vectors.

In other words, the state of the system is always separated in two equal-sized parts. When updating the state through H, the left part comes from the application of the first NBCA on the whole state in the previous step, whereas the right part is defined analogously as the result of the second NBCA evaluated on the previous state. Figure 5 depicts the block diagram for the dynamical evolution of the system starting from an initial state $s(0) = x(0)\|y(0)$.

In principle, one could sample the orbit arising from the iteration of Eq. (6) as a pseudorandom sequence, starting from a random initial configuration $s(0) \in \mathbb{F}_q^{2n}$. However, pseudorandom sequences adopted in domains such as cryptography need to satisfy several stringent properties, which implies that randomly selecting the local rule is not a good option. The motivation by which we require that the Latin squares generated by the NBCA F and G are also orthogonal in Definition 5 is twofold.

First, as recalled in Sect. 2.2, a pair of orthogonal Latin squares of order N defines a *permutation* over the Cartesian product $[N] \times [N]$. It follows that the update function defined in Eq. (6) is bijective. Thus, the resulting system is *reversible*, or equivalently its trajectories are all disjoint cycles, without transient parts. In practice, reversibility implies that the system can also be run backward in time, by applying the inverse permutation. Such a property is important in certain cryptographic primitives (e.g., SPN[2] block ciphers) where, beside generating pseudorandom sequences, there is also the need of inverting the global state of the cipher to ensure decryption. In the particular setting of OCA, one could invert the system by using the algorithm based on coupled de Bruijn graphs described in Mariot and Leporati (2018).

Second, orthogonal Latin squares coincide with a particular kind of *Maximum Distance Separable (MDS) codes*, which are of great importance in the design of *diffusion layers* for block ciphers. The reason is that layers based on MDS codes spread the statistical structure of the plaintext over the ciphertext in an optimal way, providing resistance against differential cryptanalysis. In particular, as shown by Vaudenay (1994), the function H defined in Eq. (6) corresponds to a (2, 2)-*multipermutation*, i.e. any distinct pair of input/output tuples $(x, y, F(x, y), G(x, y))$ and $(x', y', F(x', y'), G(x', y'))$ *cannot agree* on any 2 coordinates. Thus, such tuples must be at Hamming distance at least 3.

The aim of this work is to investigate the cyclic structure of the dynamical system $\mathcal{S}$, paying particular attention to cycles of maximal period. Given a state $s \in \mathbb{F}_q^{2n}$, the (minimum) *period* of s under $\mathcal{S}$ is the smallest positive integer p such that $H^p(s) = s$. In other words, p is the smallest number of iterations of H after which the state of the system returns to the initial condition s. Pseudorandom sequences with very large periods are usually sought in cryptography especially in the context of stream ciphers (Stinson and Paterson 2018). Indeed, if a pseudorandom

[1] Usually, the general definition of a dynamical system also requires that A is a metric space and that f is continuous with respect to the topology induced by the distance over A (Kůrka 2003). However, since we deal only with the case where the phase space is finite, every update function is trivially continuous with respect to the discrete topology.

[2] Substitution-Permutation Network.

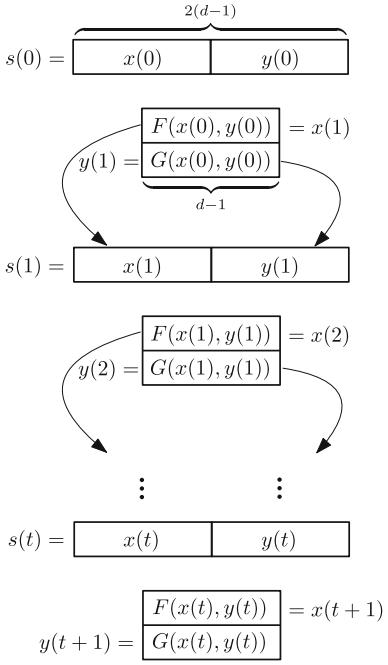


Fig. 5 Block diagram for the dynamical evolution of the system starting from the initial state $s(0) = (x(0)\|y(0)) \in \mathbb{F}_q^{2n}$

sequence used as a keystream has a shorter period than the plaintext length, an adversary can mount certain attacks based on frequency analysis. Ideally, the dynamics of a pseudorandom generator used in cryptography should be composed of a single large cycle that visits (in a non-trivial and unpredictable way) all states in the phase space.

We do not delve into the technical details needed for a rigorous treatment of pseudorandom number generators (PRNGs) for cryptographic purposes, as this would lead us astray from the main topic of this paper. Instead, we only observe that the notion of pseudorandomness implicitly assumed in this work is that of *computational security*, where the strength of a specific PRNG model is assessed with respect to specific attacks. However, the OCA approach proposed in this paper is far from being a full-fledged generator model, which is the reason why, in what follows, we do not conduct statistical tests on the pseudorandom sequences produced by OCA, nor do we prove any particular security property except focusing on their periods. Instead, the theoretical results provided in the next sections show that linear OCA can be an interesting substitute for the linear components in a standard PRNG design, such as the Linear Feedback Shift Registers (LFSRs) in the combiner or the filter model (Carlet 2021).

We conclude this section by formally stating the problem addressed in the rest of this paper:

Problem 1 Let $d \in \mathbb{N}$ and q be a power of a prime number, and let $n = d - 1$. What is the largest period attainable by the system $\mathcal{S} = \langle \mathbb{F}_q^{2n}, H \rangle$, with H defined as in Eq. (6), when F and G are OCA?

3.2 Empirical search results

We begin our study of the periods of OCA by performing an empirical search, focusing on the case of the binary alphabet, i.e. $q = 2$. The number of all Boolean functions of d variables is 2^{2^d}, since one can assign either 0 or 1 to each of the 2^d input vectors in the truth table. This prevents any exhaustive search already for $d > 5$ variables. However, concerning Problem 1 we are only interested in those dynamical systems $\mathcal{S}$ defined by two NBCA of diameter d whose local rules are bipermutive Boolean functions. A bipermutive function of d variables is effectively defined by the generating function of the $d - 2$ central cells, and thus the total number of bipermutive functions to enumerate is $2^{2^{d-2}}$. Since we consider pairs of bipermutive local rules, we have that the search space is composed of $2^{2^{d-2}} \times 2^{2^{d-2}} = 2^{2^{d-1}}$ feasible solutions. This allows to stretch the exhaustive search approach up to $d = 6$, since in that case we have at most $2^{2^{6-1}} \approx 4.3 \cdot 10^9$ pairs to enumerate. Further, Mariot et al. (2017a) gave a necessary condition on the local rules of two OCA, showing that their truth tables must be *pairwise balanced*. Two rules $f, g : \mathbb{F}_2^d \to \mathbb{F}_2$ are pairwise balanced if each of the four pairs of bits $(x, y) \in \mathbb{F}_2^2$ occurs exactly 2^{d-2} times in the juxtaposition of the truth tables of f and g. This property allows us to further reduce the search space to about $6.3 \cdot 10^7$ pairs, using the combinatorial algorithm described in Mariot et al. (2017a).

Beyond diameter $d = 6$ exhaustive search becomes unfeasible. For this reason, to expand the scope of our empirical search (especially with respect to our previous conference paper Mariot 2021), we also considered two samples of OCA pairs of diameters $d = 7$ and $d = 8$. Such samples are taken from the paper Mariot et al. (2017b), where the authors employed *Genetic Algorithms* (GA) and *Genetic Programming* (GP) to construct OCA pairs, and they are composed respectively of 68 pairs for $d = 7$ and 50 pairs for $d = 8$.

For each pair $f, g : \mathbb{F}_2^d \to \mathbb{F}_2$ of bipermutive local rules considered in our empirical search, we need to perform the following steps:

1. Check if the Latin squares generated by the NBCA $F, G : \mathbb{F}_2^{2n} \to \mathbb{F}_2^n$ respectively defined by f and g are orthogonal. Of course, this step is optional for the samples of $d = 7$ and $d = 8$, since there we already know that all pairs induce OCA.

2. If the two NBCA are orthogonal, compute the cycle decomposition of the dynamical system $\mathcal{S} = \langle \mathbb{F}_2^{2n}, H \rangle$ with H defined as in Eq. (6).
3. Find the length of the largest cycle(s) in $\mathcal{S}$.

It makes sense to start our empirical search from diameter 3: in fact, there are no OCA pairs of diameter 2, since there do not exist orthogonal Latin squares of order $2^{2-1} = 2$ in general, be them induced by CA or not. For diameter $d = 3$, a total of 8 OCA pairs result from the search over all 16 pairs of bipermutive rules. All these OCA pairs yielded the same cycle decomposition structure, i.e. one fixed point and a single maximal cycle of length 15. This is expected, since for $d = 3$ only linear OCA pairs exist, and they are all equivalent by three symmetry relations observed in Mariot et al. (2017a), namely *swap*, *complement* and *reflection*. In particular, the swap symmetry changes the order of the local rules in a pair, the complement negates the truth tables of both rules, and reflection evaluates them on the input in reversed order. Each of these symmetries is an equivalence relation which halves the search space of local rules pairs. Hence, the number all OCA pairs can actually be divided by 8, meaning that there exists only a single OCA pairs of diameter $d = 3$ up to swap, complement and reflection. This explains why the 8 pairs mentioned above all exhibit the same dynamics.

We refer to the boxplots in Fig. 6 for a general outlook of the distributions of maximal cycle lengths for diameters $4 \leq d \leq 8$.

Figure 7a, b depict more in detail the distributions of diameters $d = 4$ and $d = 5$ as histograms. We omitted the histograms for the remaining diameters since they could not be displayed properly, due to either too dense (for $d = 6$) or too sparse (for $d = 7$ and $d = 8$) distributions.

As a general remark, one can notice that the distributions up to $d = 6$ all have a very small minimum value. This is reasonable, since in all those cases we were able to perform an exhaustive search, meaning that we are considering the *complete* distributions, instead of a sample. Hence, our exhaustive search finds several OCA pairs characterized by many cycles of small length, or even by many fixed points. In any case, it is interesting to observe that the interquartile range is always compressed towards the maximum value, meaning that the great majority of OCA pairs have a large maximum period. This trend is confirmed also for the distributions of $d = 7$ and $d = 8$. Indeed, here the minimum values are way above those of $d \leq 6$, which suggests that the GA and GP proposed in Mariot et al. (2017b) are able to sample OCA pairs with large maximum cycle lengths. A third interesting remark, moreover, is that *the largest maximum cycle length observed in our experiments is* $2^{2n} - 1$. In other words, we found no OCA pairs giving a "pure cycle" of length 2^{2n} which generates the whole phase space $\mathbb{F}_2^{2d}$. Up to the considered diameters, the best possible setting is always the case where an OCA visits all cells in the superposed squares except one, which corresponds to a fixed point. An analogous property will be investigated later in Sect. 4 when we restrict our attention to linear OCA.

For each diameter $2 \leq d \leq 6$, Table 1 reports the results of our exhaustive search concerning those OCA pairs reaching a maximum cycle length of $2^{2n} - 1$. In particular,

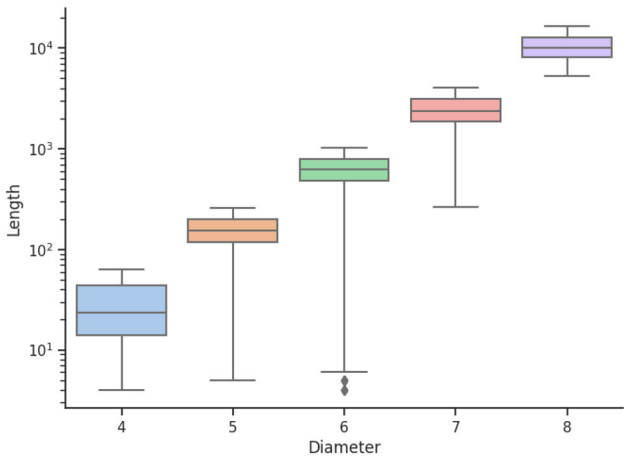


Fig. 6 Distribution of maximum cycle lengths for OCA of diameters $4 \leq d \leq 8$

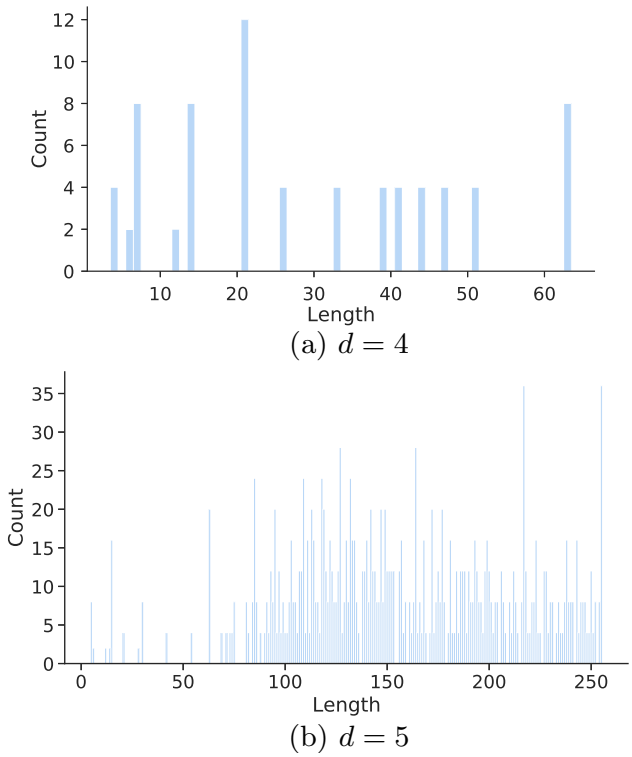


Fig. 7 Distribution of maximum cycle lengths for OCA of diameter $d = 4, 5$

Table 1 Exhaustive search results for OCA pairs of diameter $2 \leq d \leq 6$

d	n	2^n	$\mathcal{B}_d$	$\mathcal{B}_d^2$	$\mathcal{OCA}_d$	$\#m\mathcal{OCA}_d$	$\#m\mathcal{NOCA}_d$	$\#m\mathcal{LOCA}_d$
2	1	2	2	4	0	0	0	0
3	2	4	4	16	8	8	0	8
4	3	8	16	256	72	8	0	8
5	4	16	256	65,536	1704	36	12	24
6	5	32	65,536	$6.3 \cdot 10^7$	533,480	1968	1840	128

the first six columns from left to right report respectively the diameter d and $n = d - 1$, the order of the corresponding Latin squares 2^n, the number of bipermutive local rules $\mathcal{B}_d = 2^{2^{d-2}}$, the number of ordered pairs $\mathcal{B}_d^2$ visited by our exhaustive search algorithm, and the number of pairs which generate OCA $\mathcal{OCA}_d$ (taken from Mariot et al. 2017a). Finally, the last three columns report the total number $\#m\mathcal{OCA}_d$ of OCA pairs having a maximum cycle length of $2^{2n} - 1$, and then their classification in nonlinear and linear pairs, respectively denoted as $\#m\mathcal{NOCA}_d$ and $\#m\mathcal{LOCA}_d$. Remark that the numbers in the last four columns of Table 1 are not normalized up to the three symmetry relations mentioned above. The values given in the seventh column of the table show that the number of OCA pairs with a maximum cycle length of $2^{2n} - 1$ represent a very small fraction of all OCA pairs, which moreover becomes even smaller as the diameter increases. Further, contrary to what we reported in our previous conference work Mariot (2021), *there do exist nonlinear OCA pairs of maximum cycle length* $2^{2n} - 1$. Indeed, by extending the search to $d = 6$ (while in Mariot 2021 we arrived at $d = 5$), one can even see that the proportion of linear OCA is quite small compared to that of nonlinear OCA. Nevertheless, in what follows we focus on the linear OCA since in that case it is possible to use results from linear algebra over finite fields to give a precise characterization of their cycle structures.

4 Periods of linear OCA

We now focus on the linear case, describing a method to completely determine the cycle structure of a pair of linear OCA. This improves on the previous results of our conference paper Mariot (2021), where only an upper bound on the maximal period of linear OCA was given. As it often happens when studying the behavior of dynamical systems governed by a linear transformation, our method leverages on linear algebra methods, and in particular on the theory of *Linear Modular Systems* (LMS). A LMS is a finite dynamical system whose phase space is a vector space V over a finite field, and whose update function is a linear mapping over V. A good overview of the results about LMS that we use in this section can be found in Lidl and Niederreiter's book on finite fields (Lidl and Niederreiter 1997).

Let $f, g : \mathbb{F}_q^d \to \mathbb{F}_q$ be two linear bipermutive local rules of diameter d. Following the notation recalled in Sect. 2, we assume that the linear combinations defining f and g are respectively given by the two vectors $a = (a_1, \ldots, a_d) \in \mathbb{F}_q^d$ and $b = (b_1, \ldots, b_d) \in \mathbb{F}_q^d$, where a_1, b_1, a_d, b_d are all nonzero to ensure bipermutivity. In particular, we assume that $a_d = b_d = 1$ to obtain monic polynomials, which simplifies our calculations. Therefore, suppose that $P_f(X), P_g(X) \in \mathbb{F}_q[X]$ are the monic polynomials of degree $n = d - 1$ and nonzero constant term associated to f and g. By Theorem 2 f and g induce a pair of OCA if and only if their polynomials $P_f(X)$ and $P_g(X)$ are relatively prime. As proved in Mariot et al. (2020), this characterization stands on the fact that the transformation which associates the CA input configuration $x \| y \in \mathbb{F}_q^{2n}$ to the output $F(x\|y)\|G(x\|y)$ is defined by the following $2n \times 2n$ *Sylvester matrix*:

$$M_{f,g} = \begin{pmatrix} a_1 & \cdots & a_d & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & a_1 & \cdots & a_d & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_1 & \cdots & a_d \\ b_1 & \cdots & b_d & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & b_1 & \cdots & b_d & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & b_1 & \cdots & b_d \end{pmatrix}. \tag{7}$$

In particular, the two rules generate a pair of OCA if and only if the transformation $M_{f,g} \cdot (x, y)^\top$ is bijective, or equivalently if and only if $M_{f,g}$ is invertible. It is a well known fact that the determinant of a Sylvester matrix—also called the *resultant*—is not null if and only if $P_f(X)$ and $P_g(X)$ do not have any factor in common (Gelfand et al. 2008). Hence, the authors' focus in Mariot et al. (2020) was to count the number of linear OCA pairs by counting the number of invertible Sylvester matrices

defined by linear bipermutive rules, or equivalently the number of pairs of coprime polynomials with degree n and nonzero constant term over $\mathbb{F}_q$.

As it usually happens when dealing with a dynamical system whose updating function is described by a matrix, the t-th iterate of the system $\mathcal{S}$ defined in Sect. 3.1 consists of multiplying the t-th power of the Sylvester matrix $M_{f,g}$ by the initial state vector, as shown in the next lemma:

Lemma 3 *Given $d \in \mathbb{N}$ and $n = d - 1$, let $\mathcal{S} = \langle \mathbb{F}_q^{2n}, H \rangle$ be the dynamical system defined by the update function in Eq. (6), where the CA $F, G : \mathbb{F}_q^{2n} \to \mathbb{F}_q^n$ are defined by two bipermutive linear rules $f, g : \mathbb{F}_q^d \to \mathbb{F}_q$ of diameter d, with coprime associated polynomials $P_f(X), P_g(X) \in \mathbb{F}_q[X]$. Then, for any initial state $s(0) = x(0) \| y(0) \in \mathbb{F}_q^{2n}$, the state of $\mathcal{S}$ at time $t \in \mathbb{N}$ is given by:*

$$s(t) = x(t) \| y(t) = M_{f,g}^t \cdot s(0) = M_{f,g}^t \cdot (x(0) \| y(0))^\top. \quad (8)$$

Proof We proceed by induction on $t \in \mathbb{N}$. The base case $t = 1$ corresponds to the above observation about Theorem 2: a single application of the map $H : \mathbb{F}_2^{2n} \to \mathbb{F}_2^{2n}$ defined in Eq. (6) corresponds to the matrix-vector multiplication $M_{f,g} \cdot (x(0) \| y(0))^\top$. Let us assume now that the claim is valid for any $t \in \mathbb{N}$, and consider the case $t + 1$: this is equivalent to iterating H for $t + 1$ steps starting from $s(0)$, which can be written equivalently as the composition of H with its t-th iterate H^t:

$$s(t + 1) = H^{t+1}(s(0)) = H \circ H^t(s(0)). \quad (9)$$

By induction hypothesis, we know that $H^t(s(0)) = M_{f,g}^t \cdot s(0)^\top$, and that a single application of H amounts to multiplying $M_{f,g}$ with the current state vector. Hence, we can rewrite Eq. (9) as follows:

$$H^{t+1}(s(0)) = H \circ H^t(s(0)) = M_{f,g} \cdot (M_{f,g}^t \cdot s(0)^\top)^\top, \quad (10)$$

from which we conclude that

$$s(t + 1) = M_{f,g}^{t+1} \cdot s(0)^\top = M_{f,g}^{t+1} \cdot (x(0), y(0))^\top.$$

□

Concerning Problem 1, Lemma 3 implies that the maximum length of the cycles in system $\mathcal{S}$ are bounded above by the *order* of the associated Sylvester matrix $M_{f,g}$, considered as an element of the *general linear group* $GL(2n, \mathbb{F}_q)$. The general linear group $GL(2n, \mathbb{F}_q)$ is defined as the set of all invertible matrices of size $2n \times 2n$ with entries in $\mathbb{F}_2$, equipped with matrix multiplication as a group operation. Indeed, the orthogonality requirement forces $M_{f,g}$ to be invertible, since the resulting linear map, which is the superposition of the two Latin squares, is a permutation; moreover, Lemma 3 establishes that the t-th iterate of the transformation H corresponds to the t-th power of such matrix. Thus, determining the upper bound for the maximum cycle length is equivalent to finding the minimum $t \in \mathbb{N}$ such that $M_{f,g}^t = I_{2n}$, i.e. the t-th power of $M_{f,g}$ which transforms it into the identity matrix of order $2n$. This is, in turn, equivalent to determining the order of the cyclic subgroup generated by $M_{f,g}$ in $GL(2n, \mathbb{F}_q)$. It is a well-known fact (see e.g. Jacobson 1985; Mullen and Panario 2013) that the order of the general linear group $GL(2n, \mathbb{F}_q)$, or equivalently its cardinality, is equal to:

$$\#GL(2n, \mathbb{F}_q) = (q^{2n} - 1)(q^{2n} - q)(q^{2n} - q^2) \cdots (q^{2n} - q^{2n-1}). \quad (11)$$

Let us now recall *Lagrange's theorem* (Gallian 2012): *the order of any subgroup $H \leq G$ of a finite group G must divide the order of G*. This means that the order of the cyclic subgroup generated by the Sylvester matrix can only be a divisor of $\#GL(2n, \mathbb{F}_q)$ as defined in Eq. (11). Moreover, we know that the maximum period reachable by a pair of OCA can be at most q^{2n}, due to the fact that the phase space $\mathbb{F}_2^{2n}$ of $\mathcal{S}$ is composed of q^{2n} elements, and the null vector is always a fixed point (because the underlying system is linear). Thus, we have concluded that the order of the Sylvester matrix can be at most $q^{2n} - 1$, therefore obtaining an upper bound for the maximum cycle length achievable by a pair of linear OCA. To summarize, we have proved the following result:

Lemma 4 *Let $d \in \mathbb{N}$, $n = d - 1$ and $\mathcal{S} = \langle \mathbb{F}_q^{2n}, H \rangle$ be the dynamical system where H is defined as in Eq. (6), with OCA $F, G : \mathbb{F}_q^{2n} \to \mathbb{F}_q^n$ generated by a pair of linear bipermutive rules $f, g : \mathbb{F}_q^d \to \mathbb{F}_q$. Then, the period p of any state $s \in \mathbb{F}_q^{2n}$ is at most $p \leq q^{2n} - 1$.*

It is important to stress that the above bound is not always reached. Indeed, it might be the case that even though the Sylvester matrix has maximum order $q^{2n} - 1$, the cycle structure of two linear OCA is characterized by shorter periods. In particular, assume that the system $\mathcal{S} = \langle \mathbb{F}_q^{2n}, H \rangle$ has cycles of periods $t_1, \ldots, t_k$. Then, the order of the Sylvester matrix $M_{f,g}$ is actually the least common multiple of $t_1, \ldots, t_k$. As a matter of fact, assume that t is the order of $M_{f,g}$: we have that $M_{f,g}^t \cdot s^\top = s$ for any state $s \in \mathbb{F}_q^{2n}$. Thus, t must be a multiple of $l = \mathrm{lcm}(t_1, \ldots, t_k)$. Moreover, $(A^l - I) \cdot s^\top = 0$ for all $s \in \mathbb{F}_q^{2n}$, where I denotes the identity matrix. Therefore, we obtain that $A^l = I$, which means that $l \geq t$, and thus $t = \mathrm{lcm}(t_1, \ldots, t_k)$.

To give a more precise characterization of the cycle structure of the system $\mathcal{S}$ in the linear case, we introduce the following *cycle sum* notation following (Lidl and Niederreiter 1997):

$$\sum(\mathcal{S}) = (n_1, t_1) + (n_2, t_2) + \cdots + (n_k, t_k). \tag{12}$$

This is a formal sum which indicates that $\mathcal{S}$ has n_i cycles of length t_i, for all $i \in [k]$. A summand in (12) is also called a *cycle term*.

Recall that the *characteristic polynomial* of a square matrix A over $\mathbb{F}_q$ is defined as the determinant of $XI - A$, while the *minimal polynomial* of A is the monic polynomial $m(X) \in \mathbb{F}_q[X]$ of smallest degree such that $m(A)$ is the zero matrix. A monic polynomial $g(X) = X^k + a_{k-1}X^{k-1} + \cdots a_1X + a_0 \in \mathbb{F}_q[X]$ is the characteristic and minimal polynomial of its associated *companion matrix* $M(g(X))$. In particular, the characteristic polynomial of any square matrix A over $\mathbb{F}_q$ is the product of its *elementary divisors* $g_1(X), \cdots, g_r(X)$, and the *rational canonical form* of A is the matrix A^* defined as:

$$A^* = \begin{pmatrix} M(g_1(X)) & 0 & \cdots & 0 \\ 0 & M(g_2(X)) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & M(g_r(X)) \end{pmatrix}, \tag{13}$$

where $M(g_i(X))$ denotes the companion matrix of the elementary divisor $g_i(X)$. The two matrices A and A^* are related by the equation $A = P^{-1}AP$, where P is an invertible matrix over $\mathbb{F}_q$.

The cycle structure of a linear modular system defined by a transition matrix A can be expressed in terms of the orders of the elementary divisors occurring in its rational canonical form A^*. In particular, given a linear OCA pair defined by a nonsingular Sylvester matrix $M_{f,g} \in GL(2n, \mathbb{F}_q)$, the cycle sum of the corresponding dynamical system $\mathcal{S} = \langle \mathbb{F}_q^{2n}, H \rangle$ can be determined using the following procedure described in Lidl and Niederreiter (1997):

1. Determine the elementary divisors $g_1(X), \ldots, g_r(X)$ of the Sylvester matrix $M_{f,g}$, where $g_i(X) = f_i(X)^{m_i}$ with $f_i(X)$ monic and irreducible over $\mathbb{F}_q$ for all $i \in [r]$.
2. Determine the orders $t_1^{(i)} = \mathrm{ord}(f_i(X))$ of the polynomials $f_i(X)$.
3. Compute the orders $t_h^{(i)} = \mathrm{ord}(f_i(X)^h)$ for $i \in [r]$ and $h \in [m_i]$.
4. Find the cycle sum $\sum(\mathcal{S}_i)$ of the system defined by the elementary block $M(g_i(X))$, for $i \in [r]$, using Theorem 9.96 in Lidl and Niederreiter (1997).
5. Determine the cycle sum of the whole system $\mathcal{S}$ as the product of the cycle sums $\sum(\mathcal{S}_i)$, for $i \in [r]$.

The details to compute the product of cycle sums in the last step of the procedure are omitted for the sake of brevity, but can be found in Lidl and Niederreiter (1997).

5 Enumeration algorithms and results

Given a pair of linear OCA, the procedure described at the end of the previous section can be used to completely determine the cycle structure of the associated dynamical system $\mathcal{S}$. In this last section, we are interested in determining when the order of the Sylvester matrix $M_{f,g}$ is *exactly* the maximum allowed by Lemma 4, i.e. $q^{2n} - 1$. Recall that an irreducible polynomial $p(X) \in \mathbb{F}[X]$ of degree d is called *primitive* if it is a generator of the multiplicative group of the extension field $\mathbb{F}_{q^d}$. Then, one has the following result (see e.g. Ghorpade et al. 2011 for a proof):

Theorem 5 *Let $A \in GL(2n, \mathbb{F}_q)$ be a $2n \times 2n$ nonsingular matrix over $\mathbb{F}_q$, and let t be the order of A, i.e. the smallest $t \in \mathbb{N}$ such that $A^t = I$. Then, $t = q^{2n} - 1$ if and only if its minimal polynomial $m_A(X)$ is primitive.*

Hence, to enumerate all linear OCA pairs whose associated Sylvester matrix has maximum order $q^{2n} - 1$, we can determine its minimal polynomial and check whether it is primitive. This strategy is summarized in the following procedure:

(1) Set $n = d - 1$.
(2) For each pair of polynomials $P_f(X), P_g(X) \in \mathbb{F}_q[X]$ with degree n and nonzero constant term do:
   - if $\gcd(P_f(X), P_g(X)) = 1$ then:
     - Determine the minimal polynomial $m(X)$ of $M_{f,g}$
     - If $m(X)$ is primitive print the pair $P_f(X), P_g(X)$

Remark that this enumeration algorithm is different from the one proposed in our previous conference paper Mariot (2021): there, we employed a different method to determine the order of the Sylvester matrix, namely relying on Lagrange's theorem to check only the divisors of the order of $GL(2n, \mathbb{F}_q)$.

We implemented the above procedure in Magma, and applied it to enumerate Sylvester matrices of maximum order $q^{2n} - 1$ for $q = 2$. In particular, this improved enumeration algorithm turned out to be much more efficient than our previous version, since we managed to enumerate

all such matrices for linear OCA pairs up to diameter $d = 16$ in a bit less than an hour, using a 64-bit Linux machine with a 16-core AMD Ryzen processor running at 3.5 GHz and 48 GB of RAM. In contrast, our previous algorithm based on Lagrange's theorem implemented in Java took almost 5 days to enumerate all such pairs only up to $d = 11$, using the same machine. The bottleneck of our new algorithm, on the other hand, becomes the memory: the check of primitivity is the step where MAGMA consumes the most memory, and for $d = 16$ it reached 25 GB. We did not manage to go further since for the next instance of $d = 17$ we ran out of memory. Beside this experiment, we also applied our improved algorithm to enumerate invertible Sylvester matrices over a ternary alphabet, i.e. with $q = 3$. In this case, the time becomes again the bigger bottleneck before the memory does: the enumeration for diameter $d = 14$ did not finish within 10 days of computation, hence we stopped at $d = 13$.

Table 2 reports the numbers obtained from the two experiments described above. The third and fourth column give for each diameter the maximum possible order for Sylvester matrices of size $2n$ respectively over $\mathbb{F}_2$ and $\mathbb{F}_3$. The fifth and the sixth column, likewise, report the number of Sylvester matrices reaching those orders.

Remark that the column $\mathcal{M}_2$ reporting the numbers of maximum order invertible Sylvester matrices over $\mathbb{F}_2$ differs from the column $\#m\mathcal{LOCA}_d$ in Table II of our conference paper Mariot (2021). Our new results in Table 2 have been double-checked by running the same algorithm in MAGMA and computing the order of the matrix instead of checking the primitivity of the minimal polynomial, and we obtained the same results. Hence, we can be confident that the new counts reported in Table 2 are correct.

Table 2 Number of invertible $2n \times 2n$ Sylvester matrices of maximum order over $\mathbb{F}_q$, with $q = 2, 3$

d	n	$2^{2n} - 1$	$3^{2n} - 1$	$\mathcal{M}_2$	$\mathcal{M}_3$
2	1	3	80	0	0
3	2	15	728	1	0
4	3	63	6560	1	3
5	4	255	59,048	3	15
6	5	1023	531,440	17	216
7	6	4095	4,782,968	34	1001
8	7	16,383	43,046,720	191	14,168
9	8	65,535	387,420,488	500	77,890
10	9	262,143	387,420,488	1886	652,603
11	10	1,048,575	3,486,784,400	5981	5,108,147
12	11	4,194,303	31,381,059,608	30,120	55,906,579
13	12	16,777,215	$2.54 \cdot 10^{12}$	68,813	296,956,782
14	13	67,108,863	–	429,937	–
15	14	268,435,455	–	1,185,306	–
16	15	1,073,741,823	–	4,447,563	–

6 Conclusions

In this paper, we investigated a novel approach to generate pseudorandom sequences by means of cellular automata, namely by defining a dynamical systems based on two orthogonal CA. The trajectories of this system can be visualized as "jumps" over the superposed orthogonal Latin squares generated by the two CA, using the entries in each visited cell as the new set of row and column coordinates for the next cell. Remarking that two orthogonal CA induce a bijective superposition, the dynamics of the system is reversible and thus composed only of disjoint cycles. For this reason, we set up our investigation to search for OCA pairs that produce the largest cycles possible, which is a desirable property when considering pseudorandom generators for cryptographic applications. Further, the fact that the system is defined by a pair of orthogonal Latin squares implies that the update function is a multipermutation, which is a useful primitive when designing the diffusion layers of block ciphers.

We first performed an empirical search for the maximum cycle lengths of OCA pairs over the binary alphabet. This entailed first an exhaustive enumeration approach up to diameter $d = 6$, and then an analysis of a sample of OCA pairs produced by the evolutionary algorithms described in Mariot et al. (2017b) for $d = 7$ and $d = 8$. The results showed that there are both linear and nonlinear OCA pairs reaching the maximum possible cycle length of $2^{2n} - 1$. Subsequently, we described a method to completely determine the cycle structure of linear OCA pairs, using the rational canonical form of the Sylvester matrix associated to the two linear rules. Further, observing that a Sylvester matrix has a maximum order of $q^{2n} - 1$ if and only if its minimal polynomial is primitive, we devised an improved enumeration algorithm to generate them all for $q = 2$ and $q = 3$, respectively up to diameter $d = 16$ and $d = 13$. In doing that, we also fixed the numbers of such matrices for the binary case, which were reported incorrectly in the conference version of our paper Mariot (2021).

There are several interesting directions and open problems for future research on this topic. The condition granted by Theorem 5 surely gives a better way to determine if a Sylvester matrix has maximum order than using Lagrange's Theorem. However, it would be nice to give a precise characterization of when the minimal polynomial of a Sylvester matrix is primitive, which would probably yield a more efficient condition to check. The same goes

also for a more precise characterization of the cycle sum of a Sylvester matrix. This would allow not only to determine the order of the matrix, but even to give a complete characterization of the cycles of a linear OCA pair. We are not aware of any result on the minimal polynomial or the rational canonical form of a Sylvester matrix, and we think this might be a good starting point for further research. Possibly, the minimal polynomial and rational canonical form in this case are related to the two polynomials that defines the Sylvester matrix.

From the perspective of our computer experiments, it might also be interesting to extend them by improving the primitivity test in our search algorithm. For instance, fast and efficient algorithms for primitivity check have been proposed by Brent and Zimmermann (2008), although they target only trinomials.

A third interesting direction is to broaden the scope of the investigation to the more general nonlinear case. As we have seen in Sect. 3.2, there exist also nonlinear OCA pairs that achieve a maximum cycle length of $2^{2n} - 1$. One way to approach this problem would be to consider the ANF of the local rules, and define a system of (multivariate) polynomial equations whose associated matrix resembles a Sylvester matrix, or one of its generalizations (Gelfand et al. 2008). The study of nonlinear OCA pairs would also be interesting from a practical point of view, for a twofold reason. First, one can consider nonlinear OCA as a direct method to generate pseudorandom sequences. As a matter of fact, nonlinear OCA might also possess good confusion qualities that could help in withstanding general attacks on PRNG. This contrasts with the linear case, where linear OCA can only be used as components of a PRNG (e.g. replacing the LFSRs in a combiner or filter model). A preliminary step for the analysis of nonlinear OCA would be to filter out bad pairs that do not produce good pseudorandom sequences as measured by standard statistical test suites such as NIST (Bassham et al. 2010) or DIEHARD (Marsaglia 1996). Clearly, following this approach would also imply to consider pairs of a larger diameter than those examined through empirical search in this paper. Indeed, up to diameter $d = 8$ the corresponding maximum period is still too small to produce a decent sample of pseudorandom sequences for statistical purposes. Beside pseudorandom number generators, the second reason concerns the design of *nonlinear diffusion layers* for block ciphers (Liu et al. 2018). Nonlinear OCA pairs could be considered for the design of such layers, thus providing both diffusion and confusion properties. More generally, one could also consider the use of nonlinear OCA pairs to design *S-boxes*, which constitutes the confusion layer of block ciphers. There is quite an extensive body of literature concerning the design of S-boxes with good cryptographic properties based on CA, see for instance (Szaban and Seredynski 2011; Picek et al. 2017; Ghoshal et al. 2018; Mariot et al. 2019). Most of these works focus on the trade-off between reaching a high nonlinearity and a low differential uniformity to withstand certain attacks. In this respect, it would be interesting to determine whether the vectorial function H defined by two nonlinear OCA pairs has also a good nonlinearity, and if the property of being a multipermutation positively affects the differential uniformity.

Author contributions Not applicable.

Funding No funding was received to assist with the preparation of this manuscript.

Availability of data and materials The experimental data are available at https://github.com/rymoah/hip-to-be-latin-square.

Code availability The source code is available at https://github.com/rymoah/hip-to-be-latin-square.

Declarations

Conflict of interest The author has no competing interests to declare that are relevant to the content of this article.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethical approval Not applicable.

References

Bassham III LE, Rukhin AL, Soto J, Nechvatal JR, Smid ME, Barker EB, Leigh SD, Levenson M, Vangel M, Banks DL et al (2010) Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards & Technology

Brent R, Zimmermann P (2008) A multi-level blocking distinct degree factorization algorithm. Contemp Math 461:47–58

Carlet C (2021) Boolean functions for cryptography and coding theory. Cambridge University Press, Cambridge

Daemen J, Govaerts R, Vandewalle J (1994) An efficient nonlinear shift-invariant transformation. In: 15th Symp. on information theory in the Benelux, Louvain-la-Neuve (B), pp 30–31

del Rey ÁM, Mateus JP, Sánchez GR (2005) A secret sharing scheme based on cellular automata. Appl Math Comput 170(2):1356–1364

Eloranta K (1993) Partially permutive cellular automata. Nonlinearity 6(6):1009

Formenti E, Imai K, Martin B, Yunès J (2014) Advances on random sequence generation by uniform cellular automata. In: Calude CS, Freivalds R, Iwama K (eds) Computing with new resources, vol 8808. Lecture notes in computer science. Springer, Berlin, pp 56–70

Gallian J (2012) Contemporary abstract algebra. Nelson Education, Scarborough

Gelfand IM, Kapranov M, Zelevinsky A (2008) Discriminants, resultants, and multidimensional determinants. Springer, Berlin

Ghorpade SR, Hasan SU, Kumari M (2011) Primitive polynomials, singer cycles and word-oriented linear feedback shift registers. Des Codes Cryptogr 58(2):123–134

Ghoshal A, Sadhukhan R, Patranabis S, Datta N, Picek S, Mukhopadhyay D (2018) Lightweight and side-channel secure 4 × 4 S-boxes from cellular automata rules. IACR Trans Symmetric Cryptol 2018(3):311–334

Hedlund GA (1969) Endomorphisms and automorphisms of the shift dynamical systems. Math Syst Theory 3(4):320–375

Herranz J, Sáez G (2018) Secret sharing schemes for (k, n)-consecutive access structures. In: Camenisch J, Papadimitratos P (eds) CANS 2018, Proceedings. Lecture notes in computer science, vol 11124. Springer, Berlin, pp 463–480

Jacobson N (1985) Basic Algebra, I. W.H. Freeman and Company

Kari J (2005) Theory of cellular automata: a survey. Theor Comput Sci 334(1–3):3–33

Keedwell AD, Dénes J (2015) Latin squares and their applications. Elsevier, Amsterdam

Koç CK, Apohan A (1997) Inversion of cellular automata iterations. IEE Proc Comput Digit Tech 144(5):279–284

Kůrka P (2003) Topological and symbolic dynamics. Société mathématique de France

Leporati A, Mariot L (2014) Cryptographic properties of bipermutive cellular automata rules. J Cell Autom 9(5–6):437–475

Lidl R, Niederreiter H (1997) Finite fields. Cambridge University Press, Cambridge

Liu Y, Rijmen V, Leander G (2018) Nonlinear diffusion layers. Des Codes Cryptogr 86(11):2469–2484

Mariot L (2021) Hip to be (Latin) square: maximal period sequences from orthogonal cellular automata. In: CANDAR 2021, Proceedings. IEEE, pp 29–37

Mariot L, Leporati A (2014) Sharing secrets by computing preimages of bipermutive cellular automata. In: Was J, Sirakoulis GC, Bandini S (eds) ACRI 2014, Proceedings. Lecture notes in computer science, vol 8751. Springer, Berlin, pp 417–426

Mariot L, Leporati A (2018) Inversion of mutually orthogonal cellular automata. In: Mauri G, Yacoubi SE, Dennunzio A, Nishinari K, Manzoni L (eds) ACRI 2018, Proceedings. Lecture notes in computer science, vol 11115. Springer, Berlin, pp 364–376

Mariot L, Formenti E, Leporati A (2016) Constructing orthogonal Latin squares from linear cellular automata. CoRR. arXiv:1610.00139

Mariot L, Formenti E, Leporati A (2017a) Enumerating orthogonal Latin squares generated by bipermutive cellular automata. In: Dennunzio A, Formenti E, Manzoni L, Porreca AE (eds) AUTOMATA 2017, Proceedings. Lecture notes in computer science, vol 10248. Springer, Berlin, pp 151–164

Mariot L, Picek S, Jakobovic D, Leporati A (2017b) Evolutionary algorithms for the design of orthogonal Latin squares based on cellular automata. In: Bosman PAN (ed) GECCO 2017, Proceedings. ACM, pp 306–313

Mariot L, Gadouleau M, Formenti E, Leporati A (2020) Mutually orthogonal Latin squares based on cellular automata. Des Codes Cryptogr 88(2):391–411

Mariot L, Picek S, Leporati A, Jakobovic D (2019) Cellular automata based S-boxes. Cryptogr Commun 11(1):41–62

Marsaglia G (1996) Diehard: a battery of tests of randomness. http://stat.fsu.edu/geo

Martin B (2008) A Walsh exploration of elementary CA rules. J Cell Autom 3(2):145–156

Meier W, Staffelbach O (1991) Analysis of pseudo random sequence generated by cellular automata. In: Davies DW (ed) EUROCRYPT'91, Proceedings. Lecture notes in computer science, vol 547. Springer, Berlin, pp 186–199

Moore C (1997) Quasilinear cellular automata. Physica D 103(1–4):100–132

Mullen GL, Panario D (2013) Handbook of finite fields. CRC Press, Boca Raton

Picek S, Mariot L, Yang B, Jakobovic D, Mentens N (2017) Design of S-boxes defined with cellular automata rules. In: CF'17, Proceedings. ACM, pp 409–414

Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613

Stinson DR (2004) Combinatorial designs—constructions and analysis. Springer, Berlin

Stinson DR, Paterson M (2018) Cryptography: theory and practice. CRC Press, Boca Raton

Szaban M, Seredynski F (2011) Designing cryptographically strong S-boxes with use of 1d cellular automata. J Cell Autom 6(1):91–104

Vaudenay S (1994) On the need for multipermutations: cryptanalysis of MD4 and SAFER. In: Preneel B (edi) FSE'94, Proceedings. Lecture notes in computer science, vol 1008. Springer, Berlin, pp 286–297

Wolfram S (1985) Cryptography with cellular automata. In: Williams HC (ed) CRYPTO'85, Proceedings. Lecture notes in computer science, vol 218. Springer, Berlin, pp 429–432