This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012

Hip to Be (Latin) Square: Maximal Period Sequences from Orthogonal Cellular Automata

Luca Mariot[1]

[1]Cyber Security Research Group, Delft University of Technology ,
Mekelweg 2, 2628 CD Delft, The Netherlands ,
l.mariot@tudelft.nl

May 9, 2022

Abstract

Orthogonal Cellular Automata (OCA) have been recently investigated in the literature as a new approach to construct orthogonal Latin squares for cryptographic applications such as secret sharing schemes. In this paper, we consider OCA for a different cryptographic task, namely the generation of pseudorandom sequences. The idea is to iterate a dynamical system where the output of an OCA pair is fed back as a new set of coordinates on the superposed squares. The main advantage is that OCA ensure a certain amount of diffusion in the generated sequences, a property which is usually missing from traditional CA-based pseudorandom number generators. We study the problem of finding OCA pairs with maximal period by first performing an exhaustive search over local rules of diameter up to d = 5, and then focusing on the subclass of linear bipermutive rules. In this case, we characterize an upper bound on the periods of the sequences in terms of the order of the subgroup generated by an invertible Sylvester matrix. We finally devise an algorithm based on Lagrange's theorem to efficiently enumerate all linear OCA pairs that induce Sylvester matrices of maximal order up to diameter $d = 11$.

Keywords Cellular Automata, Orthogonal Latin Squares, Pseudorandom Sequences, Multipermutations, Sylvester Matrix

1 Introduction

Cellular Automata (CA) have been extensively used in the past to define cryptographic primitives, especially *Pseudorandom Number Generators* (PRNGs). Indeed, CA are an interesting computational model for generating pseudorandom sequences, since they can exhibit very chaotic dynamical behaviors. Moreover, the massive parallelism inherent to CA lends itself to efficient hardware implementations. Nevertheless, CA-based PRNGs such as those based on rule 30 pioneered by Wolfram [26] have later been found insecure, since an adversary can efficiently recover the initial state of the CA by observing only the trace of the cell sampled as a pseudorandom sequence [18, 8]. Later research [17, 3, 9, 10] focused on improving the security of Wolfram-like PRNGs by investigating local rules of higher radii with better cryptographic primitives, especially

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012

related to the *confusion* principle set forth by Shannon [20]. Still, another problem that has received little attention in this research thread is that CA in general also has poor *diffusion*, meaning that the differences between distinct initial states propagate too slowly in the dynamic evolution of the CA [1]. This flaw is mostly due to the local nature of the CA update rule, and it can represent a problem with respect to *differential cryptanalysis* attacks.

In this work, we investigate a new method for generating pseudorandom sequences by cellular automata, based on the iteration of *Orthogonal* CA (OCA). Orthogonal CA are pairs of CA whose superposed global rules generate *orthogonal Latin squares*, and up to now they have been mostly analyzed in connection with *secret sharing schemes* [14, 13]. The main idea underlying our method is to define a dynamical system whose state is the input configuration of an OCA pair. Then, the system is iterated by concatenating the output of the two OCA as a new input configuration. Intuitively, this process starts from a random cell on the orthogonal Latin squares, and uses the superposed entries as the coordinates of the new cell where to "jump" in the next iteration.

The motivation of our work is twofold. First, dynamical systems arising from OCA are reversible, which is useful in cryptographic applications such as block ciphers. Second, the orthogonality of the corresponding Latin squares allows to implement a $(2,2)$-multipermutation [24], which guarantees a certain amount of diffusion between blocks of $d-1$ cells, where d is the diameter of the local rules.

A desirable property for a PRNG is to feature a large period, starting from any seed. For this reason, after giving the necessary background definitions in Section 2 and defining the dynamical system in Section 3.1, we perform an exhaustive search of OCA pairs up to diameter $d = 5$ to compute their cycle decompositions, remarking that the maximal period $2^{2n}-1$ is attained only by pairs formed by linear rules. Subsequently, in Section 4 we relate the periods of the system induced by a linear OCA pair through the order of the subgroup generated by the associated Sylvester matrix. Next, leveraging on Lagrange's theorem, we prove that the maximum order of such a matrix is indeed upper bounded by $2^{2n}-1$. We finally design an algorithm to efficiently enumerate all linear OCA pairs of maximal order based on our theoretical results, and apply it up to diameter $d = 11$. Such findings cue us to several open problems and further directions of research on this subject, which we discuss in the conclusions of the paper in Section 5.

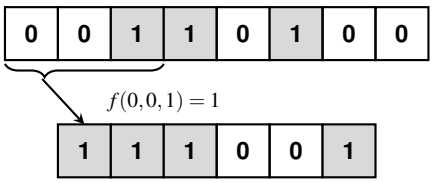
2 Background

We start by giving some basic definitions related to cellular automata and orthogonal Latin squares used in the remainder of the paper. In what follows, by $[N] = \{1, \cdots, N\}$ we denote the set of the first N positive integers, while $\mathbb{F}_q$ stands for the *finite field* with q elements, where q is a power of a prime number. Further, the n-dimensional vector space over $\mathbb{F}_q$ is denoted by $\mathbb{F}_q^n$.

Cellular automata are a parallel computational model whose global state is described by an array of *cells*, usually arranged on a line or a grid. Each cell synchronously updates its state by evaluating a local update rule on itself and a certain amount of neighboring cells. In this work, we are mainly interested in the model of one-dimensional *No-Boundary CA* (NBCA), studied in [16, 13] respectively in the context of S-boxes and orthogonal Latin squares:

Definition 1. *A* No-Boundary CA *is a map* $F : \mathbb{F}_q^n \to \mathbb{F}_q^{n-d+1}$ *defined by a* local rule

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012



(a) Local rule evaluation.

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

(b) Truth table of rule 90.

Figure 1: Example of computation in a CA of length $n = 8$ equipped with rule 90 of diameter $d = 3$, defined as $f(x_i, x_{i+1}, x_{i+2}) = x_i \oplus x_{i+2}$.

$f : \mathbb{F}_q^d \to \mathbb{F}_q$ *of* diameter $d \leq n$, *where*

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

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

In other words, each output coordinate 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 output vector is shorter than the input, since the local rule is applied only until the coordinate $n-d+1$ to avoid boundary conditions (which is the reason why this model is called *No-Boundary*). One of the most studied settings are CA over the binary alphabet (i.e. $q = 2$), where the local rule is a *Boolean function* $f : \mathbb{F}_2^d \to \mathbb{F}_2$ of d variables. In this case, it is common to use *Wolfram's convention* to encode a CA local rule, which is basically the decimal encoding of its 2^d-bit truth table [25]. Figure 1b depicts an example of CA with $n = 8$ input cells, induced by the local rule that computes the XOR of the leftmost and rightmost cells in a neighborhood of diameter $d = 3$. It can be seen from the truth table on the right that the Wolfram code of the rule is 90 by reading the output column from top to bottom and encoding the resulting binary string in decimal form.

A *Latin square* of order $N \in \mathbb{N}$ is a $N \times N$ matrix L with entries from $[N]$ such that every row and every column are permutations of $[N]$. Two Latin squares L_1 and L_2 of order N are called *orthogonal* if

$$(L_1(i_1, j_1), L_2(i_1, j_1)) \neq (L_1(i_2, j_2), L_2(i_2, j_2)) \tag{2}$$

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012

for all distinct pairs of coordinates $(i_1, j_1), (i_2, j_2) \in [N] \times [N]$. Stated otherwise, two Latin squares are orthogonal if their *superposition* yields all the ordered pairs of the Cartesian product $[N] \times [N]$. Orthogonal Latin squares have several applications in cryptography, most notably related to *secret sharing schemes* [22].

In [13], the authors showed how to generate orthogonal Latin squares with cellular automata, which have later been named *orthogonal CA* (OCA) in [14]. The basic idea is to consider CA with bipermutive local rules. A function $f : \mathbb{F}_q^d \to \mathbb{F}_q$ is called *bipermutive* if, by fixing either the leftmost or the righmost $d-1$ input variables to any value, the resulting restriction on the remaining coordinate is a permutation of $\mathbb{F}_q$. Eloranta [2] proved that a NBCA $F : \mathbb{F}_q^{2(d-1)} \to \mathbb{F}_q^{(d-1)}$ with bipermutive local rule $f : \mathbb{F}_q^d \to \mathbb{F}_q$ gives rise to a Latin square of order $N = q^{d-1}$, a result which was later independently rediscovered in [11]. The idea is to use the left and right halves of the input configuration to index respectively the row and the column of the square, and the output configuration as the entry at those coordinates. More precisely, one can associate a $N \times N$ square matrix S_F to F as follows: for each $(i, j) \in [N] \times [N]$, the entry of S_F at row i and column j equals

$$S_F(i, j) = \phi(F(\psi(i)||\psi(j))) \ , \tag{3}$$

where $||$ denotes the concatenation operator. Thus, the entry $S_F(i, j)$ is determined by computing the CA on the input vector where the first $d-1$ bits corresponds to the binary representation of row i, while the last $d-1$ are the binary representation of column j. As an example, Figure 2 depicts 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, defined as $f_{150}(x_i, x_{i+1}, x_{i+2}) = x_i \oplus x_{i+1} \oplus x_{i+2}$. For simplicity, we mapped the entries of $\mathbb{F}_2^2$ to integer numbers using the encoding $00 \mapsto 1$, $10 \mapsto 2$, $01 \mapsto 3$, $11 \mapsto 4$.

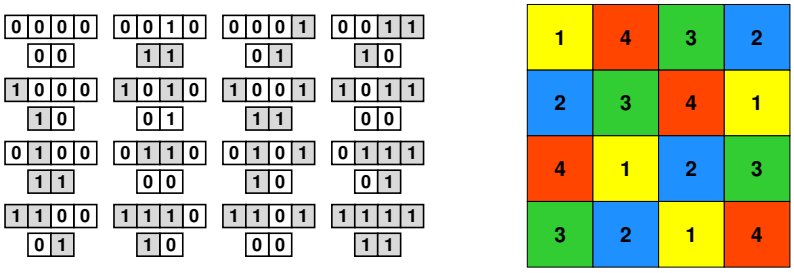
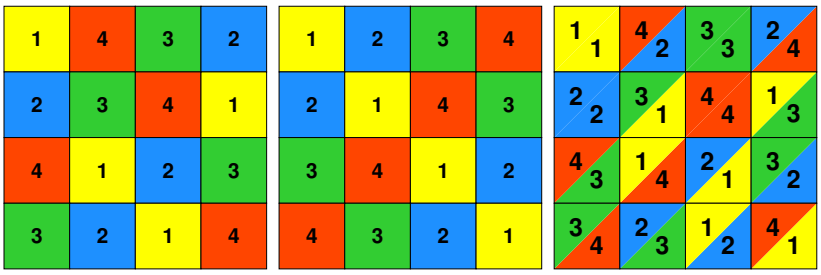


Figure 2: Example of Latin square to the CA $F : \mathbb{F}_2^4 \to \mathbb{F}_2^2$ with local rule 150.

The characterization of OCA given in [13] considers bipermutive local rules that are also *linear*, i.e. $f : \mathbb{F}_q^d \to \mathbb{F}_q$ is defined for all $x \in \mathbb{F}_q^d$ as a linear combination $f(x_1, \cdots, x_d) = a_1 x_1 + \cdots + a_d x_d$, with $a_i \in \mathbb{F}_q$ for $i \in [d]$ and the constraint that $a_1, a_d \neq 0$ to ensure bipermutivity. A polynomial of degree $n = d-1$ with coefficients in $\mathbb{F}_q$ can be naturally associated to a linear rule, by using the mapping $f \mapsto P_f(X) = a_1 + a_2 X + \cdots + a_d X^n$. Then, the characterization of linear OCA proved in [13] can be stated as follows:

Theorem 1. *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 relatively prime.*

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012



(a) Rule 150 (b) Rule 90 (c) Superposition

Figure 3: Orthogonal Latin squares generated by NBCA with rules 150 and 90.

Figure 3 depicts an example of two Latin squares of order 4 arising from the NBCA respectively equipped with the local rules 90 and 150. It can be seen that the polynomials over $\mathbb{F}_2$ associated to rules 90 and 150 are respectively $1+X^2$ and $1+X+X^2$; since they are relatively prime, the corresponding Latin squares are orthogonal.

3 Generating Sequences with Orthogonal CA

In this section we first define the dynamical system used to generate sequences with OCA, and formally state the problem of identifying the local rules that induce maximum length cycles. We then exhaustively enumerate such rules up to diameter $d = 5$.

3.1 Description of the Generator and Problem Statement

As mentioned in Section 2, any bipermutive CA with local rule f of diameter d defines a Latin square of order q^n, where $n = d-1$ and q is the size of the alphabet. However, one cannot use such a CA to directly generate a pseudorandom sequence, as done in Wolfram-like PRNGs. Indeed, since the cellular automaton is in the no-boundary setting and the initial configuration is composed of $2n$ cells, only a configuration of length n results from a single evaluation of the global rule, leaving not enough cells for a second iteration.

Instead of using a single local rule, the main idea behind our generator is to take a *pair* of local rules $f, g : \mathbb{F}_q^d \to \mathbb{F}_q$, both applied to the same initial configuration s of length $2n = 2(d-1)$, as in the case of orthogonal Latin squares. In this way, one gets two output configurations $z = F(s)$, $w = G(s)$, each of length n, generated by the global rules F, G respectively induced by f and g. At this point, 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. Seen on the superposed Latin squares generated by F and G, this process can be visualized as starting from the pair of entries indexed by the initial configuration s, and then using such entries as the destination coordinates where to "jump" in the next step.

We can now give a formal definition of the dynamical system $\mathcal{S}$ intuitively described above. Given $d \in \mathbb{N}$ and q a power of a prime, the phase space of $\mathcal{S}$ is the vector space $\mathbb{F}_q^{2n}$ where $n = d-1$, i.e. the set of all vectors of length $2n$ over the finite field $\mathbb{F}_q$. In particular $\mathbb{F}_q^{2n}$ is isomorphic to the Cartesian product $\mathbb{F}_q^n \times \mathbb{F}_q^n$, the set of all ordered pairs of n-dimensional vectors over $\mathbb{F}_q$. Slightly abusing notation, in what follows we will

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012

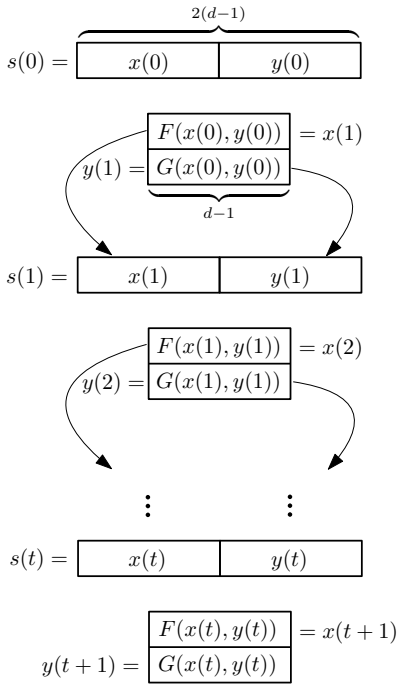


Figure 4: Block diagram for the dynamical evolution of the system starting from the initial state $s(0) = (x(0), y(0)) \in \mathbb{F}_2^{2n}$.

also consider a pair of vectors of n components as an element of $\mathbb{F}_q^{2n}$. In fact, going from one representation to the other simply entails adding and dropping parentheses accordingly.

Let $s(0) = (x(0), y(0)) \in \mathbb{F}_q^{2n}$ be the initial state of the system $\mathcal{S}$. Further, let $f, g : \mathbb{F}_q^d \to \mathbb{F}_q$ be two bipermutive local rules of diameter d, with $F, G : \mathbb{F}_q^{2n} \to \mathbb{F}_q^n$ being the corresponding NBCA global rules. Finally, denote by $H : \mathbb{F}_q^{2n} \to \mathbb{F}_q^{2n}$ the function that *concatenates* the results of the two global rules F and G to the same CA input $s \in \mathbb{F}_q^{2n}$. Then, the system $\mathcal{S}$ updates its current state $s(t)$ at time $t \in \mathbb{N}$ through the following equation:

$$s(t+1) = H(x(t), y(t)) = (F(s(t)), G(s(t))) \ . \tag{4}$$

In other words, the state of the system is always separated in two equal-size parts, where the left part comes from the application of the first global rule on the whole state in the previous step, whereas the right part is defined analogously as the result of the second global rule evaluated on the previous state. Figure 4 depicts the block diagram for the dynamical evolution of the system. In what follows, we will compactly denote such a system $\mathcal{S}$ by the pair $\langle \mathbb{F}_q^{2n}, H \rangle$. In principle, one could sample the orbit arising from the iteration of Equation (4) as a pseudorandom sequence, starting from a random initial configuration $s(0)$. However, pseudorandom sequences adopted in domains such as cryptography need to satisfy several stringent properties, meaning that one cannot simply select a pair of local rules at random. For this reason, beside choosing f and g to be bipermutive local rules, we also require that the Latin squares generated by the global rules F and G are *orthogonal*. The motivation is twofold:

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012

1. As recalled in Section 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 Equation (4) 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 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 [14].
2. 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 [24], the function H defined in Equation (4) 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. Stated differently, such tuples must be at Hamming distance at least 3.

In this work we investigate the dynamics of the system $\mathcal{S} = \langle \mathbb{F}_q^{2n}, H \rangle$ when the underlying local rules f and g generate a pair of OCA. More precisely, we are interested in studying the periods of the cycles in $\mathcal{S}$. 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. In cryptography, pseudorandom sequences with very large periods are usually sought. This is due to the fact that cryptographic primitives such as *stream ciphers* encrypt the plaintext by computing the bitwise XOR between the plaintext and a *keystream*, which is actually a pseudorandom sequence generated by stretching a short secret key [23]. In particular, an attacker can mount certain attacks based on frequency analysis when the pseudorandom sequences used as keystreams have a period that is shorter than the plaintext. Ideally, the dynamics of a pseudorandom generator used in cryptography should be composed of a single large cycle that visits all states in the phase space.

We now state the problem addressed in the rest of the 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 maximal period attainable by the system $\mathcal{S} = \langle \mathbb{F}_q^{2n}, H \rangle$, with H defined as in Equation* (4)*, when the bipermutive local rules $f, g : \mathbb{F}_q^d \to \mathbb{F}_q$ induce a pair of orthogonal CA?*

We will mainly consider the binary alphabet $\mathbb{F}_2$ (i.e., $q = 2$), since this is the simplest case to analyze and also the most useful one for cryptographic applications. However, most of the theoretical results presented in the next sections can be straightforwardly lifted to OCA over any finite field $\mathbb{F}_q$.

3.2 Empirical Search up to $d = 5$

We started our investigation of Problem 1 by conducting an exhaustive search over all pairs of bipermutive rules up to diameter $d = 5$, filtering only those that yield

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012

Table 1: Size of the search space $\mathcal{B}_d^2$ of bipermutive rules pairs with diameter d.

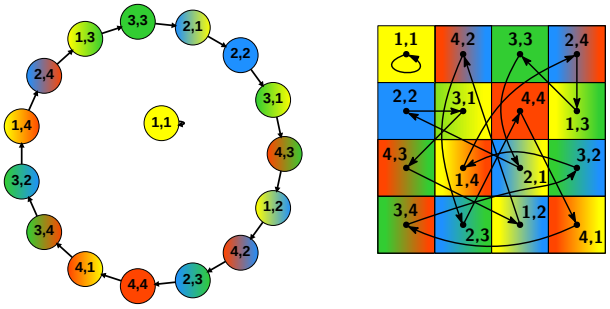
d	n	2^n	$\mathcal{B}_d$	$\mathcal{B}_d^2$	OCA_d
2	1	2	2	4	0
3	2	4	4	16	8
4	3	8	16	256	72
5	4	16	256	65536	1704
6	5	32	65536	$4.3 \cdot 10^9$	533480

OCA and analyzing the cycle decompositions of the corresponding dynamical systems. For each diameter $d \leq 6$, Table 1 reports the size of the CA output configuration $n = d - 1$, 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 that can be formed with them $\mathcal{B}_d^2 = 2^{2^{d-2}} \cdot 2^{2^{d-2}} = 2^{2^n}$, and finally the number of pairs which generate OCA. The last column has been taken from [12], where an exhaustive search of all OCA up to diameter $d = 6$ is performed by means of a combinatorial algorithm that enumerates only pair of bipermutive local rules which are *pairwise balanced*. As a matter of fact, up to now there are no known methods to enumerate generic OCA pairs, unless one narrows the attention to the case of linear rules addressed in [13]. Further, remark that the numbers in the last column of Table 1 multiply by 8 the numbers of OCA pairs reported in [12], since we did not consider any symmetry relation preserving orthogonality as done in that work. In particular, the three relevant relations are *swapping* (changing the order of the local rules in a pair), *complement* (taking the 1-complement of both rules' truth tables) and *reversal* (evaluating the local rules on the input in reversed order). Each of them is an equivalence relation, and the corresponding quotient space is half of the size the set of local rules pairs. Hence, dividing the numbers in the last column of Table 1 by a factor of $2 \times 2 \times 2 = 8$ gives the number of equivalence classes induced by the union of these three symmetry relations.

The size of the search space of interest for our empirical investigation is thus specified by the fourth column of Table 1, $\mathcal{B}_d^2$. In particular, our exhaustive search enumerated all ordered pairs of bipermutive local rules of diameter d, selected only those that generated OCA, and for each of them determined the cycle decomposition of the corresponding dynamical system $\mathcal{S}$ defined in Section 3.1. In principle, it is also possible to extend such search to diameter $d = 6$, since the size of the resulting space ($\mathcal{B}_2^d \approx 4.3 \cdot 10^9$) is still amenable to exhaustive enumeration in a reasonable time. However, in our experiments we limited our search up to $d = 5$ since this was enough to inform our theoretical investigation.

The case of diameter $d = 2$ can be immediately discarded, since no OCA pairs exist with this parameter. In fact, one can easily see that there are only two Latin squares of order $2^{2-1} = 2$, and they are not orthogonal. 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 resulted in the same cycle decomposition structure*, i.e. one fixed point and a single cycle of length 15. As an example, Figure 5 reports the cycle decomposition of the OCA pair formed by the rules with Wolfram codes 90 and 150 respectively, along with the associated paths on the superposed squares. Consequently, all 8 OCA pairs of diameter $d = 3$ feature a maximum cycle length which is equal to the area of the square ($2^{2 \cdot 2} = 16$) minus 1, or equivalently, there is a single walk on the superposed squares that visits all cells except one (i.e., the fixed point).

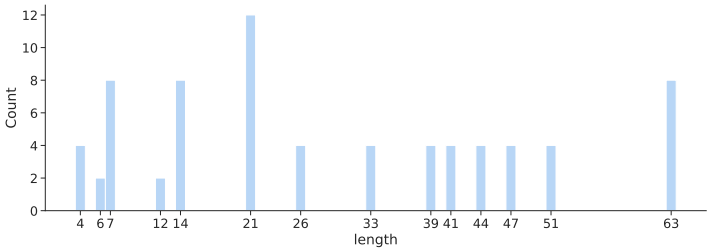
This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012



(a) Cycle decomposition

(b) Paths on superposed squares

Figure 5: Example of cycle decomposition and paths on the squares generated by the OCA with local rules 90 and 150. The color coding is as follows: yellow $\rightarrow$ 1, blue $\rightarrow$ 2, green $\rightarrow$ 3, red $\rightarrow$ 4. Two blended colors represent the superposition of the respective numbers.



(a) $d = 4$

Figure 6: Distribution of maximum cycle lengths for OCA of diameter 4 and 5.

Similar conclusions can be drawn also from the results for $d = 4$ and $d = 5$, with Figure 6 reporting the distributions of the maximum cycle lengths for $d = 4$ as an example. In particular, *no OCA pair is able to attain the* 2^{2n} *upper bound on the maximum cycle length*. In other words, there is no OCA featuring a single "pure cycle" that visits all cells in the superposed squares. Rather, the best decomposition possible is a single fixed point and a cycle of length $2^{2n} - 1$. This almost optimal situation is achieved by 8 OCA pairs for $d = 4$, whose largest cycle has length 63, and 36 pairs for $d = 5$, with a maximum length cycle of 255. A closer inspection of the types of local rules forming such OCA leads us to the second interesting finding: *all OCA pairs reaching a maximum cycle length of* $2^{2n} - 1$ *are defined by linear local rules*. For this reason, in the remainder of this paper we consider only OCA pairs defined by linear rules.

4 The Case of Linear OCA

We now delve into the case of linear OCA pairs, providing an upper bound on their periods. As it often happens when studying the behavior of dynamical systems governed

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012

by a linear transformation, such a characterization is made possible by the use of linear algebra methods.

Let $f, g : \mathbb{F}_2^d \to \mathbb{F}_2$ be two linear bipermutive local rules of diameter d. Following the notation recalled in Section 2, we assume that the linear combinations defining f and g are respectively given by the two vectors $a = (a_1, \cdots, a_d) \in \mathbb{F}_2^d$ $b = (b_1, \cdots, b_d) \in \mathbb{F}_2^d$, where $a_1 = b_1 = a_d = b_d = 1$ to ensure bipermutivity. Let $P_f(X), P_g(X) \in \mathbb{F}_2[X]$ be the monic polynomials of degree $n = d - 1$ and nonzero constant term associated to f and g. Then, by Theorem 1 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 [13], this characterization stands on the fact that the transformation which associates the CA initial configuration $(x, y) \in \mathbb{F}_2^{2n}$ to the pair of outputs $F(x,y), G(x,y)$ is defined by the following $2n \times 2n$ matrix:

$$M_{f,g} = \begin{pmatrix} a_1 & \cdots & a_d & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & a_1 & \cdots & a_d & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_1 & \cdots & a_d \\ b_1 & \cdots & b_d & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & b_1 & \cdots & b_d & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & b_1 & \cdots & b_d \end{pmatrix} . \tag{5}$$

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. The matrix defined in Equation 5 has been extensively investigated in the literature: indeed, $M_{f,g}$ is the *Sylvester matrix* associated to the two polynomials $P_f(X)$ and $P_g(X)$. 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 [5]. Therefore, the research in [13] focused on counting the number of invertible Sylvester matrices defined by linear bipermutive rules, or equivalently on counting the number of linear OCA pairs.

The next lemma shows that computing the t-th iteration of the dynamical system $\mathcal{S}$ defined in Section 3.1 corresponds to multiplying the t-th power of the Sylvester matrix $M_{f,g}$ by the initial state vector, when the local rules are linear.

Lemma 1. *Given $d \in \mathbb{N}$ and $n = d - 1$, let $\mathcal{S} = \langle \mathbb{F}_2^{2n}, H \rangle$ be the dynamical system defined by the update function in Equation* (4)*, where the CA $F, G : \mathbb{F}_2^{2n} \to \mathbb{F}_2^n$ are defined by two bipermutive local rules $f, g : \mathbb{F}_2^d \to \mathbb{F}_2$ of diameter d whose associated polynomials $P_f(X), P_g(X) \in \mathbb{F}_2[X]$ are coprime. Then, for any initial state $s(0) = (x(0), y(0)) \in \mathbb{F}_2^{2n}$, the state of $\mathcal{S}$ at time $t \in \mathbb{N}$ is given by:*

$$s(t) = (x(t), y(t)) = M_{f,g}^t \cdot s(0) = M_{f,g}^t \cdot (x(0), y(0))^\top \ . \tag{6}$$

Proof. We proceed by induction on $t \in \mathbb{N}$. The base case $t = 1$ corresponds to the observation above on Theorem 1: a single application of the transformation $H : \mathbb{F}_2^{2n} \to \mathbb{F}_2^{2n}$ defined in Equation (4) 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:

$$s(t+1) = H^{t+1}(s(0)) = H \circ H^t(s(0)) \ . \tag{7}$$

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012

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 Equation (7) as follows:

$$H^{t+1}(s(0)) = H \circ H^t(s(0)) = M_{f,g} \cdot (M_{f,g}^t \cdot s(0)^\top)^\top \quad , \tag{8}$$

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 when the underlying OCA are defined by a pair of linear local rules, Lemma 1 implies that the periods 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* $GL(2n, \mathbb{F}_2)$. The general linear group $GL(2n, \mathbb{F}_2)$ 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 constrains $M_{f,g}$ to be invertible, and Lemma 1 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 length of the cycle where a state $s \in \mathbb{F}_2^{2n}$ belongs to 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 $M_{f,g}$ is 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 $GL(2n, \mathbb{F}_2)$.

It is a well-known fact (see e.g. [6, 19]) that the order of the general linear group $GL(2n, \mathbb{F}_2)$, or equivalently its cardinality, is equal to:

$$|GL(2n, \mathbb{F}_2)| = (2^{2n} - 1)(2^{2n} - 2)(2^{2n} - 2^2) \cdots (2^{2n} - 2^{2n-1}) \quad . \tag{9}$$

Let us now recall *Lagrange's theorem* [4]: *the order of any subgroup $H \leq G$ of a finite group G must divide the order of G.* Consequently, when determining the order of an invertible Sylvester matrix $M_{f,g} \in GL(2n, \mathbb{F}_2)$, there is no need to consider all powers $t \in \{1, \cdots, |GL(2n, \mathbb{F}_2)|\}$ and check what is the minimum value such that $M_{f,g}^t = I_{2n}$. Rather, one has to check this condition only among the *divisors* of $|GL(2n, \mathbb{F}_2)|$. Moreover, it follows that the maximum order attainable by such a Sylvester matrix is $2^{2n} - 1$. Indeed, we know that the maximum period reachable by a pair of OCA can be at most 2^{2n}, due to the fact that the phase space $\mathbb{F}_2^{2n}$ of $\mathcal{S}$ is composed of 2^{2n} elements, and moreover the null vector is always a fixed point (because the underlying system is linear). To summarize, we have proved the following:

Theorem 2. *Let $d \in \mathbb{N}$, $n = d - 1$ and $\mathcal{S} = \langle \mathbb{F}_2^{2n}, H \rangle$ be the dynamical system where H is defined as in Equation* (4)*, with OCA $F, G : \mathbb{F}_2^{2n} \to \mathbb{F}_2^n$ generated by a pair of linear bipermutive rules $f, g : \mathbb{F}_2^d \to \mathbb{F}_2$. Then, the period p of any state $s \in \mathbb{F}_2^{2n}$ is at most $p \leq 2^{2n} - 1$.*

From a practical point of view, Theorem 2 and Lagrange's theorem gives a further constraint to check if a pair of linear OCA can generate a maximum period sequence: it suffices to check if the minimum exponent t such that $M_{f,g}^t$ equals the identity matrix is $2^{2n} - 1$.

As an application of the results above, we present a combinatorial algorithm to enumerate all linear OCA pairs of diameter d which induce a Sylvester matrix of maximal order:

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012

Table 2: Number of maximal period linear OCA pairs of diameter $d \leq 11$.

d	n	$2^{2n}-1$	$\#\mathcal{LOCA}_d$	$\#m\mathcal{LOCA}_d$	Time
2	1	3	0	–	–
3	2	15	1	1	$<$ 1s
4	3	63	5	1	$<$ 1s
5	4	255	21	3	$<$ 1s
6	5	1023	85	15	$<$ 1s
7	6	4095	341	42	3.967s
8	7	16383	1365	181	59.162s
9	8	65535	5461	572	18m59.302s
10	9	262143	21845	1872	5h56m10.208s
11	10	1048575	87381	5899	4d16h27m22.126s

ENUMERATE-MAXIMAL-LIN-OCA(d)

Init: Set $n = d-1$, $t = 2^{2n}-1$, $C = |GL(2n, \mathbb{F}_2)|$

Loop: For each pair of polynomials $P_f(X), P_g(X) \in \mathbb{F}_2[X]$ with degree n and nonzero constant term do:

If $\gcd(P_f(X), P_g(X)) = 1$ **then**
If $M^t_{f,g} = I_{2n}$ AND $M^e_{f,g} \neq I_{2n}$ for all $e|t$, $e < t$ **then**
Print the pair $P_f(X), P_g(X)$
End If
End If

End Loop

The algorithm visits all pairs of binary polynomials of degree $n = d-1$ and nonzero constant term, and first checks if the associated polynomials are relatively prime (or equivalently if the two rules form an OCA pair) by computing their greatest common divisor. If this is the case, the algorithm verifies whether the Sylvester matrix has maximal order $t = 2^{2n-1}$. By Lagrange's theorem, this operation is accomplished by checking that $M^t_{f,g} = I_{2n}$ and $M^e_{f,g}$ is *not* the identity matrix for any divisor e of the order of $GL(2n, \mathbb{F}_2)$ less than t. If this condition is satisfied, the pair of polynomials $P_f(X), P_g(X)$ is printed.

We applied this algorithm to enumerate all linear OCA pairs with a Sylvester matrix of maximal order up to diameter $d = 11$. For each value of d, Table 2 reports the number of linear OCA pairs ($\#\mathcal{LOCA}_d$, taken from [13]), the numbers of pairs with maximal order $2^{2n}-1$ ($\#m\mathcal{LOCA}_d$) and the time required to enumerate them. In particular, we implemented the algorithm in Java and performed the experiments on a 64-bit Linux machine with a 16-core AMD Ryzen processor running at 3.5 GHz and 48 GB of RAM. As a general remark, it can be noticed that the time required to run ENUMERATE-MAXIMAL-LIN-OCA grows quite rapidly, with more than 4 days required to sift through all pairs of linear bipermutive rules of diameter 11. Indeed, the most time-consuming step is the computation of the period of the Sylvester matrix. Although we used the SQUARE-AND-MULTIPLY algorithm [7] to efficiently exponentiate the matrix, this operation still needs to be performed for all divisors of $2^{2n}-1$ (which is still a

This is a post-print version of an article published in CANDAR 2021, available at:
https://doi.org/10.1109/CANDAR53791.2021.00012

significant reduction rather than checking all exponents smaller than $2^{2n} - 1$). Also, a second observation is that the number of pairs reaching maximal period $2^{2n} - 1$ seems to represent a small subset of linear OCA. In particular, remark that the fourth and fifth columns of Table 2 are normalized up to the symmetries considered in [12].

5 Open Problems and Future Directions

The theoretical results and the empirical findings of the previous section prompt us with several open problems and directions for further research on maximal period sequences generated with OCA. To begin with, it would be interesting to find a recurrence equation to count all linear OCA pairs of maximal order. Equivalently, this problem amounts to count the number of invertible Sylvester binary matrices of size $2n \times 2n$ with maximum order $2^{2n} - 1$. Apparently, this problem has not been studied before in the literature, since the sequence corresponding to the fifth column of Table 2 is not reported in the OEIS [21]. A second interesting direction for further research would be to find an efficient characterization of linear OCA of maximal order. Indeed, the main limitation of our approach is that it relies on computing the order of a matrix, which is computationally expensive. Yet, we are interested in Sylvester matrices, which have a very specific structure. It may thus be possible that the maximal order can be characterized as a property of the polynomials that define the matrix. Finally, more in general, one could broaden the scope of the investigation to characterize linear OCA pairs with smaller periods, and analyze more closely also the periods of nonlinear OCA pairs. The authors of [15] already addressed the construction of nonlinear OCA pairs using *Genetic Algorithms* (GA) and *Genetic Programming* (GP). In this regard, one interesting direction would be to consider the maximization of the largest period as a further optimization goal for GA and GP, either in a single-objective or multi-objective setting.

Appendix: Source Code and Experimental Data

The source code of the algorithm and the experimental data discussed in this paper are available at `https://github.com/rymoah/hip-to-be-latin-square`.

Acknowledgments

The author wishes to thank Luca Manzoni and Antonio E. Porreca for a helpful preliminary discussion on the definition of the dynamical system based on orthogonal CA.

References

[1] J. Daemen, R. Govaerts, and J. Vandewalle. An efficient nonlinear shift-invariant transformation. In *15th Symp. on Information Theory in the Benelux, Louvain-la-Neuve (B)*, pages 30–31, 1994.

[2] K. Eloranta. Partially permutive cellular automata. *Nonlinearity*, 6(6):1009, 1993.

[3] E. Formenti, K. Imai, B. Martin, and J. Yunès. Advances on random sequence generation by uniform cellular automata. In C. S. Calude, R. Freivalds, and

K. Iwama, editors, *Computing with New Resources - Essays Dedicated to Jozef Gruska on the Occasion of His 80th Birthday*, volume 8808 of *Lecture Notes in Computer Science*, pages 56–70. Springer, 2014.

[4] J. Gallian. *Contemporary abstract algebra*. Nelson Education, 2012.

[5] I. M. Gelfand, M. Kapranov, and A. Zelevinsky. *Discriminants, resultants, and multidimensional determinants*. Springer Science & Business Media, 2008.

[6] N. Jacobson. *Basic Algebra, I*. W.H. Freeman and Company, 1985.

[7] D. E. Knuth. *Art of computer programming, volume 2: Seminumerical algorithms*. Addison-Wesley Professional, 2014.

[8] C. K. Koc and A. Apohan. Inversion of cellular automata iterations. *IEE Proceedings-Computers and Digital Techniques*, 144(5):279–284, 1997.

[9] A. Leporati and L. Mariot. 1-resiliency of bipermutive cellular automata rules. In J. Kari, M. Kutrib, and A. Malcher, editors, *Cellular Automata and Discrete Complex Systems - 19th International Workshop, AUTOMATA 2013, Gießen, Germany, September 17-19, 2013. Proceedings*, volume 8155 of *Lecture Notes in Computer Science*, pages 110–123. Springer, 2013.

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

[11] L. Mariot, E. Formenti, and A. Leporati. Constructing orthogonal latin squares from linear cellular automata. *CoRR*, abs/1610.00139, 2016.

[12] L. Mariot, E. Formenti, and A. Leporati. Enumerating orthogonal latin squares generated by bipermutive cellular automata. In A. Dennunzio, E. Formenti, L. Manzoni, and A. E. Porreca, editors, *Cellular Automata and Discrete Complex Systems - 23rd IFIP WG 1.5 International Workshop, AUTOMATA 2017, Milan, Italy, June 7-9, 2017, Proceedings*, volume 10248 of *Lecture Notes in Computer Science*, pages 151–164. Springer, 2017.

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

[14] L. Mariot and A. Leporati. Inversion of mutually orthogonal cellular automata. In G. Mauri, S. E. Yacoubi, A. Dennunzio, K. Nishinari, and L. Manzoni, editors, *Cellular Automata - 13th International Conference on Cellular Automata for Research and Industry, ACRI 2018, Como, Italy, September 17-21, 2018, Proceedings*, volume 11115 of *Lecture Notes in Computer Science*, pages 364–376. Springer, 2018.

[15] L. Mariot, S. Picek, D. Jakobovic, and A. Leporati. Evolutionary algorithms for the design of orthogonal latin squares based on cellular automata. In P. A. N. Bosman, editor, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017, Berlin, Germany, July 15-19, 2017*, pages 306–313. ACM, 2017.

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

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

[18] W. Meier and O. Staffelbach. Analysis of pseudo random sequence generated by cellular automata. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 186–199. Springer, 1991.

[19] G. L. Mullen and D. Panario. *Handbook of finite fields*. CRC Press, 2013.

[20] C. E. Shannon. Communication theory of secrecy systems. *Bell Syst. Tech. J.*, 28(4):656–715, 1949.

[21] N. J. Sloane. The On-line Encyclopedia of Integer Sequences (OEIS). accessed on 14 June 2021.

[22] D. R. Stinson. *Combinatorial designs - constructions and analysis*. Springer, 2004.

[23] D. R. Stinson and M. Paterson. *Cryptography: theory and practice*. CRC press, 2018.

[24] S. Vaudenay. On the need for multipermutations: Cryptanalysis of MD4 and SAFER. In B. Preneel, editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, pages 286–297. Springer, 1994.

[25] S. Wolfram. Statistical mechanics of cellular automata. *Reviews of modern physics*, 55(3):601, 1983.

[26] S. Wolfram. Cryptography with cellular automata. In H. C. Williams, editor, *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*, volume 218 of *Lecture Notes in Computer Science*, pages 429–432. Springer, 1985.