Enumerating Orthogonal Latin Squares Generated by Bipermutive Cellular Automata

Luca Mariot[1,2], Enrico Formenti[2] and Alberto Leporati[1]

[1] Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Viale Sarca 336, 20126 Milano, Italy
{luca.mariot,leporati}@disco.unimib.it
[2] Université Côte d'Azur, CNRS, I3S, France
mariot@i3s.unice.fr, enrico.formenti@unice.fr

Abstract. We consider the problem of enumerating pairs of bipermutive cellular automata (CA) which generate orthogonal Latin squares. Since the problem has already been settled for bipermutive CA with linear local rules, we address the general case of nonlinear rules, which could be interesting for cryptographic applications such as the design of cheater-immune secret sharing schemes. We first prove that two bipermutive CA generating orthogonal Latin squares must have pairwise balanced local rules. Then, we count the number of pairwise balanced bipermutive Boolean functions and enumerate those which generate orthogonal Latin squares up to $n = 6$ variables, classifying them with respect to their nonlinearity values.

Keywords: cellular automata · Latin squares · bipermutivity · pairwise balancedness

1 Introduction

The construction of *orthogonal Latin squares* is a challenging combinatorial problem. Indeed, besides being one of the most researched topics in combinatorial design theory, orthogonal Latin squares also have numerous applications in cryptography, coding theory and the design of experiments [3,6,14].

Recently, a new construction of orthogonal Latin squares based on bipermutive cellular automata (CA) with linear local rules has been proposed in [10]. In particular, the authors proved that two linear bipermutive local rules generate a pair of orthogonal Latin squares if and only if their associated polynomials are relatively prime.

In this paper, we address the generalized problem of enumerating orthogonal Latin squares induced by *nonlinear* bipermutive CA, which could have interesting cryptographic applications. As a matter of fact, orthogonal Latin squares generated through nonlinear constructions can be employed in the design of *cheater-immune secret sharing schemes* [15].

After covering in Section 2 the necessary preliminary notions about Latin squares and cellular automata, in Section 3 we first prove that the basic reversal

and complementation operations on local rules preserve the orthogonality relation of the resulting Latin squares. Then, we show that two bipermutive local rules that give rise to orthogonal Latin squares must be *pairwise balanced*, which basically means that the four pairs $(0,0)$, $(1,0)$, $(0,1)$ and $(1,1)$ must occur an equal number of times in the superposition of their truth tables. Additionally, we prove that pairwise balancedness is a property preserved from the generating functions to the corresponding bipermutive rules, but not vice versa. In Section 4 we derive a formula for the number of pairwise balanced bipermutive rules, and apply a combinatorial algorithm to enumerate all those pairs which generate orthogonal Latin squares up to $n = 6$ variables. Finally, we classify these pairs with respect to their nonlinearity values. In Section 5 we sum up the contributions of this paper.

2 Preliminaries

In this section, we first recall the basic definitions about orthogonal Latin squares and cellular automata used throughout the paper. We then review the construction of orthogonal Latin squares based on linear bipermutive cellular automata described in [10].

2.1 Basic Definitions

In what follows, we denote by $[N]$ the set of the first N positive integer numbers, i.e. $[N] = \{1, \cdots, N\}$. We begin by defining the basic combinatorial objects of our interest, namely Latin squares:

Definition 1. *Let $N \in \mathbb{N}$. A* Latin square *of order N is a $N \times N$ matrix L such that each element of $[N]$ occurs exactly once in every row and in every column. Two Latin squares L_1 and L_2 of order N are called* orthogonal *if*

$$(L_1(i_1, j_1), L_2(i_1, j_1)) \neq (L_1(i_2, j_2), L_2(i_2, j_2)) \tag{1}$$

for all distinct pairs of coordinates $(i_1, j_1), (i_2, j_2) \in [N] \times [N]$.

Hence, two Latin squares are orthogonal if their *superposition* yields all the ordered pairs of the Cartesian product $[N] \times [N]$.

In this work, we consider a basic one-dimensional model of cellular automaton which can be considered as a special kind of vectorial Boolean function. For this reason, we first cover the necessary notions from the theory of cryptographic boolean functions, referring the reader to [1,2] for a more thorough presentation of the topic.

Let $\mathbb{F}_2$ and $\mathbb{F}_2^n$ respectively denote the finite field with two elements and the n-dimensional vector space over $\mathbb{F}_2$ (that is, the set of all binary n-tuples). In what follows, we assume that the 2^n vectors of $\mathbb{F}_2^n$ are lexicographically ordered, using least significant bit notation. A *Boolean function* of n variables is a mapping $f : \mathbb{F}_2^n \to \mathbb{F}_2$. The *truth table* of f is the vector $\Omega(f) \in \mathbb{F}_2^{2^n}$ defined as

$$\Omega(f) = (f(0, 0, \cdots, 0), f(1, 0, \cdots, 0), \cdots, f(1, 1, \cdots, 1)) \ , \tag{2}$$

that is, $\Omega(f)$ specifies the output values of f for each of the possible 2^n values of the input vectors. Consequently, the set of all 2^n binary vectors coincides with the space of Boolean functions of n variables $\mathcal{F}_n$, which thus has size 2^{2^n}.

Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$. The *reversal* $f_R : \mathbb{F}_2^n \to \mathbb{F}_2$ and the *complementation* $f_C : \mathbb{F}_2^n \to \mathbb{F}_2$ of f are defined as

$$f_R(x_1, \cdots, x_n) = f(x_R) = f(x_n, \cdots, x_1) \quad , \tag{3}$$

$$f_C(x_1, \cdots, x_n) = f(x_1, \cdots, x_n) \oplus 1 \ , \tag{4}$$

for all $x = (x_1, \cdots, x_n) \in \mathbb{F}_2^n$. Clearly, both reversal and complementation are idempotent operations, i.e. $(f_R)_R = f$ and $(f_C)_C = f$.

A Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is called *affine* if it is defined as:

$$f(x_1, \cdots, x_n) = a \oplus a_1 \cdot x_1 \oplus \cdots \oplus a_n \cdot x_n \tag{5}$$

for all $x = (x_1, \cdots, x_n) \in \mathbb{F}_2^n$, where $a, a_1, \cdots, a_n \in \mathbb{F}_2$ and $\oplus$ and $\cdot$ respectively denote the XOR and AND operations. If $a = 0$, then the function is called *linear*.

The *nonlinearity* of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined as the minimum Hamming distance of f from the set of affine functions of n variables, a property which can be expressed using the Walsh transform of f. Given $f : \mathbb{F}_2^n \to \mathbb{F}_2$, the *Walsh transform* of f is the function $W_f : \mathbb{F}_2^n \to \mathbb{R}$ defined as

$$W_f(\omega) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus \omega \cdot x} \tag{6}$$

for all $\omega \in \mathbb{F}_2^n$, where $\omega \cdot x = \omega_1 x_1 \oplus \cdots \oplus \omega_n x_n$ is the *scalar product* between ω and x. The *spectral radius* of f, denoted as $W_{max}(f)$, is the maximum absolute value of its Walsh transform W_f over all vectors $\omega \in \mathbb{F}_2^n$. Then, the nonlinearity of f is formally defined as:

$$Nl(f) = 2^{n-1} - \frac{1}{2} W_{max}(f) \ . \tag{7}$$

In this work, we focus mainly on CA based on bipermutive local rules. Formally, a *bipermutive* Boolean function is defined as follows:

Definition 2. *A boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is called* bipermutive *if, by fixing either the leftmost or the rightmost $n-1$ input coordinates to any value $\tilde{x} \in \mathbb{F}_2^{n-1}$, the resulting restriction on the remaining coordinate is a permutation over $\mathbb{F}_2$. Equivalently, function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is bipermutive if there exists $\varphi : \mathbb{F}_2^{n-2} \to \mathbb{F}_2$ such that*

$$f(x_1, x_2, \cdots, x_{n-1}, x_n) = x_1 \oplus \varphi(x_2, \cdots, x_{n-1}) \oplus x_n \tag{8}$$

for all $x = (x_1, x_2, \cdots, x_{n-1}, x_n) \in \mathbb{F}_2^n$.

The function φ appearing in Equation (8) is also called the *generating function* of f. Hence, the output of f is computed by XORing the leftmost and rightmost variables with the value of φ evaluated on the central variables. In [9], it has

been shown that the nonlinearity of a bipermutive Boolean function is four times the nonlinearity of its generating function, i.e. $Nl(f) = 4 \cdot Nl(\varphi)$. Notice that a linear Boolean function is bipermutive if and only if its leftmost and rightmost coefficients a_1 and a_n are nonzero.

Vectorial Boolean functions generalize the concept of Boolean functions to multiple outputs. Given $n, m \in \mathbb{N}$, a *vectorial Boolean function* (or (n, m)-*function*) is a mapping $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$. For all $i \in [m]$, the i-th *coordinate function* of F is the Boolean function $f_i : \mathbb{F}_2^n \to \mathbb{F}_2$ that specifies the i-th output bit of F, i.e. $f_i(x) = F(x)_i$ for all $x \in \mathbb{F}_2^n$.

Using the above notions on Boolean functions, we can now give a formal definition of cellular automaton.

Definition 3. *Let $m, n \in \mathbb{N}$ such that $m \geq n$, and let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a Boolean function. A* one-dimensional cellular automaton *(CA) of length m with* local rule *f is a vectorial Boolean function $F : \mathbb{F}_2^m \to \mathbb{F}_2^{m-n+1}$ defined as*

$$F(x_1, \cdots, x_m) = (f(x_1, \cdots, x_n), \cdots, f(x_{m-n+1}, \cdots, x_m)) \tag{9}$$

for all $x = (x_1, \cdots, x_m) \in \mathbb{F}_2^m$.

The local rule of a CA is usually identified by its *Wolfram code*, which is the decimal encoding of its truth table. On account of Definition 2, we call a CA *bipermutive* if its local rule is a bipermutive Boolean function.

A CA can be viewed as a vectorial Boolean function where each coordinate function f_i is the local rule f evaluated on the n input variables $x_i, \cdots, x_{i+n-1}$. From a different perspective, one can consider the input variables of the CA as *cells* whose state can be either 0 or 1, and where each of the first $m - n + 1$ cells updates in parallel its state by evaluating the local rule on the *neighborhood* formed by itself and the $n - 1$ cells to its right. Notice that the rightmost $n - 1$ input cells are not updated, hence there is no need to enforce any boundary condition. Remark also that, for the purposes of our work, we do not consider the *iterated behavior* of a CA produced by the repeated application of the local rule in successive time steps.

2.2 Latin Squares Generated by Cellular Automata

We now review the method for constructing Latin squares through bipermutive cellular automata, following the notation of [10]. Let us consider a CA $F : \mathbb{F}_2^{2(n-1)} \to \mathbb{F}_2^{n-1}$ based on a local rule $f : \mathbb{F}_2^n \to \mathbb{F}_2$ of n variables. Thus, F associates configurations of length $2(n-1)$ to configurations of length $n-1$. We can define a square matrix S_F by using the leftmost and rightmost $n-1$ input variables of F to index respectively the rows and the columns of S_F, while the $n-1$ output variables of F are employed to represent the entries of S_F at the respective input coordinates. More formally, let $N = 2^{n-1}$ and assume that $\phi : \mathbb{F}_2^{n-1} \to [N]$ is a one-to-one mapping between $\mathbb{F}_2^{n-1}$ and $[N]$, and let ψ be the inverse mapping of ϕ. Then, the square associated to a CA of length $2(n-1)$ is defined as follows:

Definition 4. *Let $F : \mathbb{F}_2^{2(n-1)} \to \mathbb{F}_2^{n-1}$ be a CA with local rule $f : \mathbb{F}_2^n \to \mathbb{F}_2$. The* square *associated to F is the square matrix S_F of size $N \times N$ defined for all $1 \leq i, j \leq N$ as:*

$$S_F(i, j) = \phi(F(\psi(i)||\psi(j))) \ , \tag{10}$$

where $\psi(i)||\psi(j) \in \mathbb{F}_2^{2(n-1)}$ is the concatenation of vectors $\psi(i), \psi(j) \in \mathbb{F}_2^{n-1}$.

We remark that this particular representation has been adopted in several works in the CA literature, even though under a different guise. Indeed, one can consider the square associated to a CA as the *Cayley table* of an algebraic structure $(A, \circ)$, where A is a set of size 2^{n-1} isomorphic to $\mathbb{F}_2^{n-1}$, and $\circ$ is a binary operation over A. The two operands $x, y \in A$ are represented by the vectors respectively composed of the leftmost and rightmost $n-1$ input cells of the CA, while the $n-1$ output cells represent the result $z = x \circ y$. To the best of our knowledge, the first who employed this algebraic characterization of cellular automata were Pedersen [13] and Eloranta [4], respectively for investigating periodicity and partial reversibility of CA. Other works in this line of research include Moore and Drisko [12], which studied the algebraic properties of the square representation of CA, and Moore [11], which considered the computational complexity of predicting CA whose local rules define solvable and nilpotent groups.

Depending on the underlying local rule, different algebraic structures can arise from the Cayley table of a CA. The case of *quasigroups* is especially interesting for the purposes of our work, since they are related to Latin squares. An algebraic structure $(Q, \circ)$ is a quasigroup if for all $x, y \in Q$ the two equations $x \circ z = y$ and $z \circ x = y$ have a unique solution for every $z \in Q$. When the support set Q is finite, the structure $(Q, \circ)$ is a quasigroup if and only if its Cayley table is a Latin square of order $|Q|$ [14].

A natural question to investigate is what classes of CA generate Latin squares (or equivalently, quasigroups). The following result shows that this is the case for bipermutive CA:

Lemma 1. *Let $F : \mathbb{F}_2^{2(n-1)} \to \mathbb{F}_2^{n-1}$ be a bipermutive CA with rule $f : \mathbb{F}_2^n \to \mathbb{F}_2$. Then, the square S_F induced by F is a Latin square of order $N = 2^{n-1}$.*

A proof of this fact which uses the characterization of quasigroups can be found in [4], while [10] reports a similar proof directly based on Latin squares.

Since bipermutive CA induce Latin squares, one could additionally investigate which pairs of them are orthogonal. This problem has been settled in [10] for the case of *linear* bipermutive CA. Considering Equation (8), this means that the generating functions of the local rules are linear. More precisely, let $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$ be bipermutive Boolean functions with linear generating functions $\varphi, \gamma : \mathbb{F}_2^{n-2} \to \mathbb{F}_2$ respectively defined as:

$$\varphi(x_2, \cdots, x_{n-1}) = a_2 x_2 \oplus \cdots \oplus a_{n-1} x_{n-1} \ , \tag{11}$$

$$\gamma(x_2, \cdots, x_{n-1}) = b_2 x_2 \oplus \cdots \oplus b_{n-1} x_{n-1} \ , \tag{12}$$

where $a_i, b_i \in \mathbb{F}_2$ for $i \in \{2, \cdots, n-1\}$. In this case, we can associate to f and g two polynomials $p_f(X), p_g(X) \in \mathbb{F}_2[X]$ of degree $n-1$ using the coefficients of

their generating functions as follows:

$$p_f(X) = 1 + a_2X \oplus \cdots \oplus a_{n-1}X^{n-2} + X^{n-1} \ , \tag{13}$$

$$p_g(X) = 1 + b_2X \oplus \cdots \oplus b_{n-1}X^{n-1} + X^{n-1} \ . \tag{14}$$

The following result proved in [10] gives a necessary and sufficient condition on the polynomials p_f and p_g in order for F and G to generate orthogonal Latin squares:

Theorem 1. *Let $F, G : \mathbb{F}_2^{2(n-1)} \to \mathbb{F}_2^{n-1}$ be two bipermutive CA with linear local rules $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$, and let p_f and p_g be their associated polynomials. Then, the Latin squares S_F and S_G respectively associated to F and G are orthogonal if and only if p_f and p_g are coprime.*

3 Main Results

Since the problem of characterizing pairs of bipermutive CA which generate orthogonal Latin squares has already been solved in [10] when the underlying local rules are linear, we now consider the more general case of nonlinear bipermutive CA. In order to tackle this problem, in this section we prove some results that allow us to reduce the search space of all bipermutive functions pairs. Then, we will use these results to enumerate all pairs of bipermutive CA that give rise to orthogonal Latin squares, with local rules of up to $n = 6$ variables.

Let $\mathcal{B}_n$ be the set of all pairs of bipermutive Boolean functions of n variables. As bipermutive functions are defined by their generating functions of $n-2$ variables, for all $n \geq 2$ it follows that $|\mathcal{B}_n| = |\mathcal{G}_n|$, where $\mathcal{G}_n = \{(\varphi, \gamma) \in \mathcal{F}_{n-2} \times \mathcal{F}_{n-2}\}$. Since $|\mathcal{F}_{n-2}| = 2^{2^{n-2}}$, the size of $\mathcal{G}_n$ is $2^{2^{n-2}} \cdot 2^{2^{n-2}} = 2^{2^{n-1}}$, meaning that $\mathcal{G}_n$ is isomorphic to $\mathcal{F}_{n-1}$, i.e. the set of Boolean functions of $n-1$ variables.

Clearly, if two bipermutive CA induced by a pair of local rules (f, g) give rise to orthogonal Latin squares, then the CA defined by the swapped pair (g, f) will generate the same orthogonal Latin squares in reverse order. We now show that the basic transformations of reversal and complementation introduced in Section 2.1 preserve the orthogonality relation as well:

Lemma 2. *Let $F, G : \mathbb{F}_2^{2(n-1)} \to \mathbb{F}_2^{n-1}$ be two bipermutive CA respectively defined by local rules $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$ of n variables, and let S_F, S_G be the associated Latin squares of order 2^{n-1}. Additionally, let F_R, G_R and F_C, G_C be the CA respectively defined by the reverses f_R, g_R and the complements f_C, g_C of f, g, and let S_{F_R}, S_{G_R} and S_{F_C}, S_{G_C} be the corresponding Latin squares. Then, the following hold:*

- *S_F and S_G are orthogonal if and only if S_{F_R}, S_{G_R} are orthogonal.*
- *S_F and S_G are orthogonal if and only if S_{F_C}, S_{G_C} are orthogonal.*

Proof. Since both reversal and complementation are idempotent transformations, it suffices to show only one direction of the implications, i.e. assuming that S_F and S_G are orthogonal. This means that

$$(F(x||y), G(x||y)) \neq (F(x'||y'), G(x'||y'))$$

for all distinct pairs $(x,y),(x',y') \in \mathbb{F}_2^{n-1} \times \mathbb{F}_2^{n-1}$*, since the mapping* ϕ *which associates binary vectors of length* $n-1$ *to positive integers in the range* $\{1,\cdots,2^{n-1}\}$ *is bijective.*

Let us now consider the CA F_R *induced by the reversed local rule* f_R*. Then, for all* $(x,y) \in \mathbb{F}_2^{n-1} \times \mathbb{F}_2^{n-1}$ *with* $x = (x_1,\cdots,x_{n-1})$ *and* $y = (y_1,\cdots,y_{n-1})$*, it follows that*

$$\begin{aligned} F_R(x||y) &= (f_R(x_1,\cdots,x_{n-1},y_1),\cdots,f_R(x_{n-1},y_1,\cdots,y_{n-1})) = \\ &= (f(y_1,x_{n-1},\cdots,x_1),\cdots,f(y_{n-1}\cdots,y_1,x_{n-1})) = F(y_R||x_R)_R \ , \end{aligned}$$

i.e., the output value of the reversed CA F_R *is obtained by computing the reversed output of* F *evaluated on the reversed input* $y_R||x_R$*. Analogously, the same fact holds for* G_R *with respect to* G*. Since for all* $(x,y),(x',y') \in \mathbb{F}_2^{n-1} \times \mathbb{F}_2^{n-1}$ *such that* $(x,y) \neq (x',y')$ *one has that* $(y_R,x_R) \neq (y'_R,x'_R)$*, it follows that*

$$(F(y_R||x_R)_R, G(y_R||x_R)_R) \neq (F(y'_R||x'_R)_R, G(y'_R||x'_R)_R) \ ,$$

which means that S_{F_R} *and* S_{G_R} *are orthogonal Latin squares.*

Next, let us consider the CA F_C *induced by the complemented local rule* f_C*. The output value of* F_C *over* $x||y$ *is*

$$\begin{aligned} F_C(x||y) &= (f_c(x_1,\cdots,x_{n-1},y_1),\cdots,f_c(x_{n-1},y_1,\cdots,y_{n-1})) = \\ &= (1 \oplus f(x_1,\cdots,x_{n-1},y_1),\cdots,1 \oplus f(x_{n-1},y_1,\cdots,y_{n-1})) = \\ &= \underline{1} \oplus F(x||y) \ , \end{aligned}$$

where $\underline{1} = (1,\cdots,1) \in \mathbb{F}_2^{n-1}$*. Similarly for* G_C*, one has* $G_C(x||y) = \underline{1} \oplus G(x||y)$*. Given two pairs* $(x,y),(x',y') \in \mathbb{F}_2^{n-1} \times \mathbb{F}_2^{n-1}$ *such that* $(x,y) \neq (x',y')$*, it clearly holds that* $(\underline{1} \oplus x, \underline{1} \oplus y) \neq (\underline{1} \oplus x', \underline{1} \oplus y')$*, from which it follows*

$$(\underline{1} \oplus F(x||y), \underline{1} \oplus G(x||y)) \neq (\underline{1} \oplus F(x'||y'), \underline{1} \oplus G(x'||y')) \ .$$

As a consequence, the Latin squares S_{F_C} *and* S_{G_C} *are orthogonal.* □

We now turn to analyze the truth tables of bipermutive rules whose CA generate orthogonal Latin squares. As an example, consider the pair of functions $f,g : \mathbb{F}_2^3 \to \mathbb{F}_2$ defined as $f(x_1,x_2,x_3) = x_1 \oplus x_3$ and $g(x_1,x_2,x_3) = x_1 \oplus x_2 \oplus x_3$, namely rules 90 and 150 using Wolfram's numbering convention. The Latin squares of order $N = 4$ induced by the corresponding bipermutive CA $F,G : \mathbb{F}_2^4 \to \mathbb{F}_2^2$ are orthogonal, since by Theorem 1 f and g are linear and their associated polynomials $p_f(X) = 1 + X^2$ and $p_g(X) = 1 + X + X^2$ are coprime. The truth tables $\Omega(f), \Omega(g) \in \mathbb{F}_2^8$ are the following:

$$\Omega(f) = (0,1,0,1,1,0,1,0) \ , \tag{15}$$

$$\Omega(g) = (0,1,1,0,1,0,0,1) \ . \tag{16}$$

Placing side by side these truth tables, one can see that there are $2^{3-2} = 2$ occurrences of each of the four pairs $(0,0)$, $(1,0)$, $(0,1)$ and $(1,1)$. We call this property *pairwise balancedness*, formally defined below:

Definition 5. *Two Boolean functions $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$ of n variables are* pairwise balanced *if the $(n, 2)$-function $(f, g) : \mathbb{F}_2^n \to \mathbb{F}_2^2$ defined as $(f, g)(x) = (f(x), g(x))$ is balanced, that is $|(f, g)^{-1}(y_1, y_2)| = 2^{n-2}$ for all $(y_1, y_2) \in \mathbb{F}_2^2$.*

We now prove that pairwise balancedness is a necessary condition for a pair of bipermutive local rules whose CA generate orthogonal Latin squares:

Lemma 3. *Let $F, G : \mathbb{F}_2^{2(n-1)} \to \mathbb{F}_2^{n-1}$ be bipermutive CA respectively induced by local rules $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$, and suppose that the associated Latin squares S_F, S_G are orthogonal. Then, f and g are pairwise balanced.*

Proof. Let $H : \mathbb{F}_2^{n-1} \times \mathbb{F}_2^{n-1} \to \mathbb{F}_2^{n-1} \times \mathbb{F}_2^{n-1}$ be the function defined as

$$H(x, y) = (F(x||y), G(x||y)) \tag{17}$$

for all $(x, y) \in \mathbb{F}_2^{n-1} \times \mathbb{F}_2^{n-1}$. Since S_F and S_G are orthogonal, it follows that H is bijective.

Consider two vectors $c, d \in \mathbb{F}_2^{n-1}$ and, without loss of generality, suppose that the first components of c and d, namely $c_1, d_1 \in \mathbb{F}_2$, are fixed. We want to compute the number of preimages $(x_1, \cdots, x_{n-1}, y_1) \in \mathbb{F}_2^n$ which map to (c_1, d_1) under (f, g). In order to do so, we evaluate the ratio N/M, where:

- *N is the number of input pairs $(x, y) \in \mathbb{F}_2^{n-1} \times \mathbb{F}_2^{n-1}$ such that the first components of the respective output pairs $H(x, y)$ equal (c_1, d_1).*
- *M is the number of input pairs $(x, y) \in \mathbb{F}_2^{n-1} \times \mathbb{F}_2^{n-1}$ where x and the first component of y are fixed.*

In this way, we count the total number of preimages of H which map to (c_1, d_1) and normalize it by the number of preimages where the first n components of H are fixed, thus determining the number of preimages of (c_1, d_1) under (f, g).

As H is bijective, N corresponds to the number of pairs of binary vectors of length $n-1$ where the first components are fixed, which are $2^{n-2} \cdot 2^{n-2} = 2^{2(n-2)}$. On the other hand $M = 2^{n-2}$, since we only have $n-2$ free variables in the input configuration of the CA. Hence, it follows that $|(f, g)^{-1}(y_1, y_2)| = N/M = 2^{2(n-2)}/2^{n-2} = 2^{n-2}$. □

In the next Lemma, we show that pairwise balanced generating functions induce pairwise balanced bipermutive CA:

Lemma 4. *Let $\varphi, \gamma : \mathbb{F}_2^{n-2} \to \mathbb{F}_2$ be pairwise balanced functions of $n-2$ variables, with $n > 2$. Then, the bipermutive rules $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$ induced by φ and γ are pairwise balanced.*

Proof. Let $(y_1, y_2) \in \mathbb{F}_2^2$. One has that $|(\varphi, \gamma)^{-1}(y_1, y_2)| = 2^{n-4}$, since φ and γ are balanced. Additionally, for all $\tilde{x} = (x_2, \cdots, x_{n-1}) \in (\varphi, \gamma)^{-1}(y_1, y_2)$, let $(x_1, \tilde{x}, x_n) = (x_1, x_2, \cdots, x_{n-1}, x_n)$. Then, by Equation (8) it follows that $(0, \tilde{x}, 0) \in (f, g)^{-1}(y_1, y_2)$ and $(1, \tilde{x}, 1) \in (f, g)^{-1}(y_1, y_2)$. Similarly, for all vectors $\tilde{x} \in (\varphi, \gamma)^{-1}(\bar{y}_1, \bar{y}_2)$ where $\bar{y}_1 = 1 \oplus y_1$ and $\bar{y}_2 = 1 \oplus y_2$, it holds that

$(1, \tilde{x}, 0) \in (f, g)^{-1}(y_1, y_2)$ *and* $(0, \tilde{x}, 1) \in (f, g)^{-1}(y_1, y_2)$*. Since the fiber of* (y_1, y_2) *under* (f, g) *is given by*

$$\begin{aligned}(f, g)^{-1}(y_1, y_2) = &\{(0, \tilde{x}, 0) : \tilde{x} \in (\varphi, \gamma)^{-1}(y_1, y_2)\} \cup \\ &\cup \{(1, \tilde{x}, 0) : \tilde{x} \in (\varphi, \gamma)^{-1}(\bar{y}_1, \bar{y}_2)\} \cup \\ &\cup \{(0, \tilde{x}, 1) : \tilde{x} \in (\varphi, \gamma)^{-1}(\bar{y}_1, \bar{y}_2)\} \cup \\ &\cup \{(1, \tilde{x}, 1) : \tilde{x} \in (\varphi, \gamma)^{-1}(y_1, y_2)\} \end{aligned} \tag{18}$$

and since the four sets in Equation (18) *are disjoint and have the same cardinality of* $(\varphi, \gamma)^{-1}(y_1, y_2)$*, we can finally conclude that*

$$|(f, g)^{-1}(y_1, y_2)| = 4 \cdot |(\varphi, \gamma)^{-1}(y_1, y_2)| = 4 \cdot 2^{n-4} = 2^{n-2} \ . \tag{19}$$

□

Remark that the converse of Lemma 4 does not hold. As a matter of fact, already for $n = 4$ variables there exist several instances of bipermutive functions pairs which produce orthogonal Latin squares (and hence are pairwise balanced) but whose generating functions are not pairwise balanced. An example is given by the two following linear rules:

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= 1 \oplus x_1 \oplus x_3 \oplus x_4 \ , \\ g(x_1, x_2, x_3, x_4) &= x_1 \oplus x_4 \ . \end{aligned}$$

The generating function of g in this case is the constant function defined as $\gamma(x) = 0$ for all $x \in \mathbb{F}_2^2$. Hence, the pairs $(0, 1)$ and $(1, 1)$ never occur when superimposing the truth tables of the two generating functions of f and g.

4 Enumeration of Pairwise Balanced Bipermutive Rules

In this section, we enumerate all bipermutive rules pairs generating orthogonal Latin squares up to $n = 6$ variables and we classify them according to their nonlinearity.

The space of pairs of pairwise balanced generating functions is easily characterizable from the combinatorial point of view. In fact, for $n > 2$, each pairwise balanced pair $\varphi, \gamma : \mathbb{F}_2^{n-2} \to \mathbb{F}_2$ can be represented by a string s of length 2^{n-2} over the alphabet $A = \{1, 2, 3, 4\}$, where each symbol in s corresponds to the decimal encoding of one of the possible four pairs $(0, 0)$, $(1, 0)$, $(0, 1)$ and $(1, 1)$ occurring in the superposition of the truth tables. Since φ and γ are pairwise balanced, the string s must be balanced as well, meaning that the number of occurrences of each of the four symbols of A must be 2^{n-4}. Hence, the number of pairwise balanced pairs of generating functions of $n - 2$ variables equals

$$\#Bal\mathcal{G}_n = \binom{2^{n-2}}{2^{n-4}} \cdot \binom{3 \cdot 2^{n-4}}{2^{n-4}} \cdot \binom{2^{n-3}}{2^{n-4}} \ . \tag{20}$$

As a matter of fact, to construct a balanced quaternary string of length 2^{n-2} one has first to select the positions of the 2^{n-4} occurrences of the first symbol, which can be chosen in $\binom{2^{n-2}}{2^{n-4}}$ different ways. Next, the 2^{n-4} occurrences of the second symbol must be chosen among the $2^{n-2} - 2^{n-4} = 3 \cdot 2^{n-4}$ remaining positions, which can be done in $\binom{3 \cdot 2^{n-4}}{2^{n-4}}$ different ways. Finally, for the 2^{n-4} occurrences of the third symbol one has to choose among $2^{n-2} - 2 \cdot 2^{n-4} = 2^{n-3}$ remaining positions, corresponding to $\binom{2^{n-3}}{2^{n-4}}$ possible choices. At this point, the occurrences of the fourth symbols are fixed.

However, we saw at the end of Section 3 that pairwise balancedness is not a necessary condition on the generating functions to obtain pairwise balanced bipermutive rules. Consequently, by enumerating all balanced quaternary strings of length 2^{n-2} one only explores a subset of the space of pairwise balanced bipermutive rules of n variables, and thus in turn a subset of the space of bipermutive CA pairs generating orthogonal Latin squares of order 2^{n-1}.

We thus have to resort to a combinatorial characterization of pairwise balanced bipermutive functions. To this end, we adopt the *graph representation* of bipermutive rules, originally introduced in [8]. Given $n \in \mathbb{N}$, consider an undirected graph $G = (V, E)$ where $V = \mathbb{F}_2^n$. Two nodes $v_1, v_2 \in V$ are connected by an edge if and only if they differ either in their leftmost or rightmost coordinates, while they agree on the remaining ones. Thus, G is composed of 2^{n-2} connected components, and each connected component is composed of 4 nodes all having degree 2. A Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be represented as a labeling function $l_f : V \to \{0, 1\}$ on the nodes of G. If f is bipermutive, then the labels of adjacent nodes must differ, while the labels of two nodes separated by a path of length 2 must be equal.

Clearly, given a pair of bipermutive functions $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$, we can still represent them on the graph as a labeling function $l_{f,g} : V \to \{0, 1\}^2$ on the nodes, where the labels are pairs specifying the outputs of the two functions. Assume that f and g are pairwise balanced: then, each pair $(y_1, y_2) \in \mathbb{F}_2^2$ occurs 2^{n-2} times as a label on G. As an example, Figure 1 depicts the graph representation of rule 90 and 150, which are pairwise balanced. Additionally, due to the property of different labels on adjacent nodes, it follows that exactly half of the connected components contain all $(0, 0)$ and $(1, 1)$ labels, while the remaining half contain all $(1, 0)$ and $(0, 1)$ labels. Since there are only two types of connected components with respect to the labels ($(0, 0)/(1, 1)$ and $(1, 0)/(0, 1)$), it means that we can choose them in $\binom{2^{n-2}}{2^{n-3}}$ different ways. Moreover, let $C = \{v_1, v_2, v_3, v_4\}$ be a connected component where $(v_1, v_2), (v_1, v_3), (v_4, v_2), (v_4, v_3) \in E$, and assume that the labels on the nodes are either $(0, 0)$ or $(1, 1)$. Then, the two labels can be arranged in two different ways, namely $(l_{f,g}(v_1), l_{f,g}(v_4)) = (0, 0)$ and $(l_{f,g}(v_2, v_3)) = (1, 1)$ or $(l_{f,g}(v_1), l_{f,g}(v_4)) = (1, 1)$ and $(l_{f,g}(v_2), l_{f,g}(v_3)) = (0, 0)$. In the same way, the labels on the nodes of a connected component of the type $(1, 0)/(0, 1)$ can be placed in two different ways. As a consequence, each of the $\binom{2^{n-2}}{2^{n-3}}$ ways for choosing the connected components with labels $(0, 0)/(1, 1)$ and

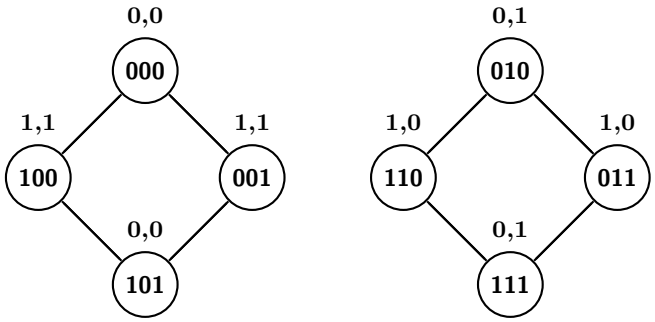

Fig. 1: Graph representation of the pairwise balanced bipermutive rules 90 and 150.

$(1,0)/(0,1)$ gives rise to $2^{2^{n-3}} \cdot 2^{2^{n-3}} = 2^{2^{n-2}}$ pairwise balanced bipermutive functions. We have thus proved the following result:

Lemma 5. *The number of pairwise balanced pairs of bipermutive Boolean functions $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$ of n variables is:*

$$\#Bal\mathcal{B}_n = \binom{2^{n-2}}{2^{n-3}} \cdot 2^{2^{n-2}} \quad . \tag{21}$$

Table 1 reports the sizes of the search spaces for the sets of all pairs of bipermutive functions, the set of pairwise balanced generating functions and the set of pairwise balanced bipermutive functions of up to $n = 7$ variables.

One can notice that for $n \geq 7$ the resulting search space is too large to be exhaustively searched, even by focusing on the subsets of pairwise balanced generating functions. For this reason, we enumerated the set of pairwise balanced bipermutive functions $Bal\mathcal{B}_n$ only up to $n = 6$ variables. To this end, we implemented an algorithm by Knuth [7] to generate all balanced binary strings of length 2^{n-2}, where the positions set to 0 and 1 respectively correspond to the $(0,0)/(1,1)$ and $(1,0)/(0,1)$ connected components. Then, for each balanced combination of connected components we generated all possible $2^{2^{n-2}}$ arrangements of the labels, constructed the resulting pairs of bipermutive functions, and computed

Table 1: Sizes of the search spaces for the different types of sets of bipermutive functions pairs of up to $n = 7$ variables.

n	$\#\mathcal{B}_n$	$\#Bal\mathcal{G}_n$	$\#Bal\mathcal{B}_n$
3	16	0	8
4	256	24	96
5	65536	2520	17920
6	4294967296	63006300	843448320
7	$\approx 1.84 \cdot 10^{19}$	$\approx 9.96 \cdot 10^{15}$	$\approx 2.58 \cdot 10^{18}$

Table 2: Distribution of CA-based orthogonal Latin squares up to $n = 6$.

n	LS_size	#total	#linear	#nonlinear	nl_distribution
3	4×4	1	1	0	–
4	8×8	9	5	4	$(4, 4, 4)$
5	16×16	213	21	192	$(4, 4, 96)$, $(8, 8, 96)$
6	32×32	66685	85	66600	$(4, 4, 512)$, $(8, 8, 4020)$, $(12, 12, 17992)$, $(16, 16, 28388)$, $(20, 20, 14384)$, $(4, 12, 8)$, $(8, 16, 160)$, $(12, 20, 128)$, $(16, 24, 88)$

their respective nonlinearity values through the Walsh transform. Finally, we generated the associated Latin squares of order $N = 2^{n-1}$, and checked for their orthogonality.

We remark that the enumeration of $Bal\mathcal{B}_6$ is a computationally intensive task, since it took approximately 22 hours to complete under our Java implementation on a 64-bit Linux machine with 40 Intel Xeon cores running at 2.4 GHz.

Table 2 reports the distribution of linear and nonlinear pairs of orthogonal Latin squares. For each value of n, the corresponding size of the Latin squares is reported, along with the number of linear and nonlinear pairs of bipermutive functions generating orthogonal Latin squares. Additionally, in the last column we report the distribution of nonlinearity values in triplets $(nl(f), nl(g), \#num)$ where $nl(f)$ and $nl(g)$ respectively denote the nonlinearity values of f and g, while $\#num$ is the number of pairs generating orthogonal Latin squares that achieve those values. Notice that all reported numbers are divided by 8, since we have to take into account the pairs with swapped order, which halve the resulting sets, and the reversal and complementation transformations, which by Lemma 2 additionally reduce them to a quarter.

As a qualitative remark on the distributions reported in Table 2, one may observe that linear pairs become more sparse as the number of variables n increases, while the majority of the pairs are nonlinear. Moreover, one can see that for $n = 6$ there are pairs with functions of different nonlinearities. This finding falsified our initial belief that two bipermutive functions inducing orthogonal Latin squares must have the same value of nonlinearity, an empirical observation which held up to $n = 5$ variables.

5 Conclusions

In this work, we considered the problem of exhaustively enumerating pairs of orthogonal Latin squares generated through bipermutive CA. We first proved that all pairs of bipermutive rules inducing orthogonal Latin squares must be pairwise balanced, meaning that the superposition of their truth tables must yield an equal number of occurrences of the four pairs $(0, 0)$, $(1, 0)$, $(0, 1)$ and $(1, 1)$. We then used a combinatorial algorithm to enumerate all pairwise balanced Boolean functions of up to $n = 6$ variables, finding those which generate orthogonal Latin squares and classifying them with respect to their nonlinearity values. The

results of our computer search showed that, as the number of variables of the local rules increases, most of the orthogonal pairs are nonlinear. This could have interesting applications from the cryptographic point of view, since as mentioned in the Introduction orthogonal Latin squares arising from nonlinear constructions have relevance in the design of cheater-immune secret sharing schemes. We plan to study this issue in future research, in particular by investigating sufficient conditions that two nonlinear bipermutive CA must satisfy in order to generate orthogonal Latin squares. Another direction worth investigating is to analyze the pairs of nonlinear rules found in this paper from the perspective of *pseudorandom number generation*, and compare them with others stemming from different classifications, like those presented in [5,9].

References

1. Carlet, C.: Boolean Functions for Cryptography and Error Correcting Codes. In: Crama, Y., Hammer, P.L. (eds.) Boolean Models and Methods in Mathematics, Computer Science, and Engineering, pp. 257–397. Cambridge University Press, New York, NY, USA, 1st edn. (2010)
2. Carlet, C.: Vectorial Boolean Functions for Cryptography. In: Crama, Y., Hammer, P.L. (eds.) Boolean Models and Methods in Mathematics, Computer Science, and Engineering, pp. 398–469. Cambridge University Press, New York, NY, USA, 1st edn. (2010)
3. Colbourn, C.J., Dinitz, J.H.: Making the mols table. In: Computational and Constructive Design Theory, pp. 67–134. Springer (1996)
4. Eloranta, K.: Partially permutive cellular automata. Nonlinearity 6(6), 1009–1023 (1993)
5. Formenti, E., Imai, K., Martin, B., Yunès, J.: Advances on random sequence generation by uniform cellular automata. In: Computing with New Resources - Essays Dedicated to Jozef Gruska on the Occasion of His 80th Birthday. pp. 56–70 (2014)
6. Keedwell, A.D., Dénes, J.: Latin squares and their applications. Elsevier (2015)
7. Knuth, D.: The art of computer programming, vol. 4, pre-fascicle 3a (2011)
8. Leporati, A., Mariot, L.: 1-resiliency of bipermutive cellular automata rules. In: Cellular Automata and Discrete Complex Systems - 19th International Workshop, AUTOMATA 2013, Gießen, Germany, September 17-19, 2013. Proceedings. pp. 110–123 (2013)
9. Leporati, A., Mariot, L.: Cryptographic properties of bipermutive cellular automata rules. J. Cellular Automata 9(5-6), 437–475 (2014)
10. Mariot, L., Formenti, E., Leporati, A.: Constructing orthogonal latin squares from linear cellular automata. CoRR abs/1610.00139 (2016)
11. Moore, C.: Predicting nonlinear cellular automata quickly by decomposing them into linear ones. Physica D: Nonlinear Phenomena 111(1-4), 27–41 (1998)
12. Moore, C., Drisko, A.A., et al.: Algebraic properties of the block transformation on cellular automata. Complex Systems 10(3), 185–194 (1996)
13. Pedersen, J.: Cellular automata as algebraic systems. Complex Systems 6(3), 237–250 (1992)
14. Stinson, D.R.: Combinatorial designs - constructions and analysis. Springer (2004)
15. Tompa, M., Woll, H.: How to share a secret with cheaters. J. Cryptology 1(2), 133–138 (1988)