# Building Correlation Immune Functions from Sets of Mutually Orthogonal Cellular Automata

Luca Mariot[1] and Luca Manzoni[2]

[1]Digital Security Group, Radboud University PO Box 9010, Nijmegen, The Netherlands
luca.mariot@ru.nl

[2]Dipartimento di Matematica e Geoscienze, Università degli Studi di Trieste, Via Valerio 12/1, Trieste, 34127, Italy
lmanzoni@units.it

July 19, 2022

## Abstract

Correlation immune Boolean functions play an important role in the implementation of efficient masking countermeasures for side-channel attacks in cryptography. In this paper, we investigate a method to construct correlation immune functions through families of mutually orthogonal cellular automata (MOCA). First, we show that the orthogonal array (OA) associated to a family of MOCA can be expanded to a binary OA of strength at least 2. To prove this result, we exploit the characterization of MOCA in terms of orthogonal labelings on de Bruijn graphs. Then, we use the resulting binary OA to define the support of a second-order correlation immune function. Next, we perform some computational experiments to construct all such functions up to $n = 12$ variables, and observe that their correlation immunity order is actually greater, always at least 3. We conclude by discussing how these results open up interesting perspectives for future research, with respect to the search of new correlation-immune functions and binary orthogonal arrays.

***Keywords*** Boolean Functions, Cellular Automata, Correlation Immunity, Side-channel countermeasures, Orthogonal Latin Squares

## 1 Introduction

Boolean functions are basic combinatorial objects that map a set of fixed size bitstrings to a single output bit. Notwithstanding their simplicity, such

1

functions find countless applications in many diverse fields of computer science and mathematics [5]. In cryptography, Boolean functions have long been used to design low-level primitives in symmetric ciphers, such as combiners and filters for linear feedback shift registers [2]. The rationale is that the resilience of these ciphers against different cryptanalytic attacks can be reduced to the cryptographic properties of the underlying Boolean functions. For example, a Boolean function used in the combiner model for Vernam stream ciphers should be at a high Hamming distance from the set of affine functions to resist fast-correlation attacks, or equivalently it should possess a high nonlinearity [21]. At the same time, the output of the function should be statistically independent from any subset of $t$ or fewer variables, to resist correlation attacks of order $t$: in other words, the Boolean functions should be correlation immune of high order $t$ [24].

The literature related to the cryptographic applications of cellular automata (CA) features a solid body of works investigating the properties of the involved Boolean functions. For example, it has been shown that Wolfram's pseudorandom number generator (PRNG) based on a simple one-dimensional CA is unsuitable for cryptographic purposes, as the underlying local rule 30 is not first-order correlation immune and does not have a high enough nonlinearity [20]. This allows to apply respectively Meier and Staffelbach's correlation attack [22] and Koc and Apohan's inversion attack [12] to efficiently recover the initial configuration of Wolfram's PRNG. Following this research thread, a few subsequent works [6, 14] focused on the search of local rules with a larger diameter and a better trade-off of nonlinearity and correlation immunity. Some of these rules with better properties have then been adopted in the design of CA-based stream ciphers such as CARPENTER and PENTAVIUM [13, 8].

The correlation immunity criterion also gained relevance in recent years concerning a different type of attacks, namely *side-channel analysis* (SCA). Instead of focusing on the mathematical design of a cipher as classical cryptanalysis does, SCA targets the *implementation* of a cipher on a device. In particular, the aim of SCA is to exploit leakages on side-channel sources such as electromagnetic emanations, timings, and variations of voltages to infer the secret key used to encrypt a message on the device. One of the options to counteract these attacks is *Boolean masking*, where noise is added to the intermediate values computed by the cipher during the encryption process, changing them from one execution to the other. In this respect, correlation immune functions of high order $t$ and low Hamming weight (i.e. with as few 1s as possible in the output of their truth tables) can be used to implement masking countermeasures, such as leakage squeezing and rotation S-box masking [3, 23], which have minimal implementation overhead and optimal resistance towards SCA attacks of order $t$.

Contrasting with the large number of works related to the study of the cryptographic properties of CA—be it at the local rule level as mentioned

2

above, or by considering them as S-boxes as in [17, 19]—there seems to be comparatively a smaller literature dedicated to the exploration of CA as a means to construct side-channel countermeasures. To the best of our knowledge, the works by Karmarkar and Roy Chowdury [9, 10, 11] are the only ones addressing the design of leakage squeezing countermeasures through hybrid CA. There, the authors remark that there exist several methods to design linear codes with hybrid CA by leveraging the techniques in [4], which can be readily used to implement a leakage squeezing countermeasure. However, they also argue that the linearity of such codes introduces other weaknesses, and thereby set out to study the cryptographic properties of nonlinear hybrid CA for leakage squeezing.

The goal of this paper is to investigate how cellular automata may be used to design correlation immune functions of low weight. To this end, we use a construction of mutually orthogonal CA (MOCA), i.e. a family of CA giving rise to a set of mutually orthogonal Latin squares (MOLS), recently introduced in [16]. In particular, we exploit the well-known fact that any set of MOLS is equivalent to an orthogonal array (OA) of strength 2. Then, taking any set of MOCA, we prove that the corresponding binary expansion is a binary OA of strength at least 2, leveraging on the MOCA characterization as orthogonal labelings on de Bruijn graphs. This result allows us to use the expanded binary OA of a MOCA family as the support of a Boolean function with correlation immunity order at least 2. We then perform a computational search experiment to generate all families of $k = 3$ MOCA defined by local rules of diameter $d = 4, 5$, and construct the corresponding Boolean functions. Interestingly, all generated functions turn out to have correlation immunity order at least 3, indicating that our theoretical result is not a tight lower bound. Although the Hamming weight reached by our correlation immune functions is far from optimal, we discuss how it could be improved by solving an associated optimization problem. The objective in this case is to remove a subset of rows from the binary expansion of a MOCA family while retaining the correlation immunity order of the resulting function.

The rest of this paper is organized as follows. Section 2 collects all necessary background definitions and results related to Boolean functions, cellular automata, Latin squares and orthogonal arrays used in this work. Section 3 proves the main theoretical result of the paper, i.e. that the binary expansion of a set of MOCA is the support of a Boolean function with correlation immunity order at least 2. Section 4 presents the results of the computational search experiment for families of $k = 3$ MOCA, giving rise to Boolean functions of up to $n = 12$ variables. Finally, Section 5 recaps the main contributions of this paper, and discusses some directions for future research.

3

## 2 Preliminary Definitions

In this section, we recall all relevant definitions to describe our results in the remainder of the paper. We start with basic concepts and results of Boolean functions, and how correlation immune functions can be characterized by orthogonal arrays. Then, we give a formal definition of the CA model used in our work, and describe the CA-based construction of mutually orthogonal Latin squares of [16].

### 2.1 Boolean Functions and Orthogonal Arrays

As a general reference, we follow Carlet's recent book on Boolean functions [2].

Let $\mathbb{F}_2 = \{0, 1\}$ be the finite field with two elements, with sum and multiplication defined respectively as the XOR (denoted by $\oplus$) and logical AND (denoted by concatenation). For any $n \in \mathbb{N}$, the set $\mathbb{F}_2^n$ of all $n$-bit bitstrings is endowed with the structure of a vector space, with vector sum defined as bitwise XOR, and multiplication by a scalar $a \in \mathbb{F}_2$ being the field multiplication of $a$ with each coordinate of a vector $x \in \mathbb{F}_2^n$. Given two vectors $x, y \in \mathbb{F}_2^n$, their Hamming distance $d_H(x, y)$ is the number of coordinates where $x$ and $y$ disagree, while their scalar product is defined as $x \cdot y = \bigoplus_{i=1}^{n} x_i y_i$. The support of a vector $x \in \mathbb{F}_2^n$ is the set $supp(x) = \{i : x_i \neq 0\}$, and its Hamming weight $w_H(x)$ is the cardinality of $supp(x)$. Equivalently, $w_H(x)$ corresponds to the Hamming distance $d_H(x, \underline{0})$ between $x$ and the null vector $\underline{0} \in \mathbb{F}_2^n$. In practice, the Hamming weight is the number of nonzero coordinates in $x$.

For all $n \in \mathbb{N}$, a *Boolean function* of $n$ variables is a mapping $f : \mathbb{F}_2^n \to \mathbb{F}_2$. The most straightforward way to represent $f$ is by means of its truth table. Suppose that a total order is fixed on the vectors of $\mathbb{F}_2^n$ (e.g., the lexicographic order). Then, the truth table of $f$ is the vector $\Omega_f \in \mathbb{F}_2^{2^n}$ defined as:

$$\Omega_f = (f(0, \cdots, 0), f(0, \cdots, 1), \cdots, f(1, \cdots, 1)) \ . \tag{1}$$

In other words, the truth table is the $2^n$-bit vector that specifies for each input vector $x \in \mathbb{F}_2^n$ in lexicographic order the corresponding output value $f(x)$. Similarly to what we defined above for binary vectors, the support of $f$ is the set $supp(f) = \{x \in \mathbb{F}_2^n : f(x) \neq 0\}$, while its Hamming weight is $w_H(f) = |supp(f)|$. Equivalently, support and weight of $f$ are defined respectively as the set of nonzero coordinates and the size of such set in the truth table $\Omega_f$.

A second common representation of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is the *Algebraic Normal Form* (ANF). Recalling that $x^2 = x$ for all $x \in \mathbb{F}_2$, the ANF of $f$ is the multivariate polynomial over the quotient ring

$\mathbb{F}_2[x]/(x_1^2 \oplus x_1, \cdots, x_n^2 \oplus x_n)$:

$$P_f(x) = \bigoplus_{u \in F_2^n} a_u x^u = \bigoplus_{u \in F_2^n} a_u(x_1^{u_1} \cdots x_n^{u_n}) , \qquad (2)$$

where $a_u \in \mathbb{F}_2$ for all $u \in \mathbb{F}_2^n$. The *algebraic degree* of $f$ is defined as $deg(f) = \max_{u \in \mathbb{F}_2^n}\{w_H(u) : u \neq 0\}$, or equivalently $deg(f)$ is the dimension of the largest nonzero monomial in the ANF of $f$. Boolean functions of degree 1 are also called *affine*, and in particular an affine function is *linear* if $a_0 = 0$.

A third method to represent a Boolean function usually adopted in cryptography is the *Walsh transform*. Given $f : \mathbb{F}_2^n \to \mathbb{F}_2$, the Walsh transform of $f$ is the function $W_f : \mathbb{F}_2^n \to \mathbb{Z}$ defined for all $a \in \mathbb{F}_2^n$ as:

$$W_f(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus a \cdot x}, \quad . \qquad (3)$$

Intuitively, $W_f(a)$ measures the correlation between $f$ and the linear function defined by the scalar product $a \cdot x$. The Walsh transform is useful to assess several interesting cryptographic properties of $f$. For example, the nonlinearity property is defined as the Hamming distance of $f$ from the set of all affine functions. Using the Walsh transform, the nonlinearity of $f$ may be computed as:

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{a \in \mathbb{F}_2^n} \{|W_f(a)|\} \quad . \qquad (4)$$

Another cryptographic property which will be of special interest for this paper is correlation immunity. A Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is correlation immune of order $1 \leq t \leq n$ if any subset of at $1 \leq k \leq t$ input variables is statistically independent from the output of $f$. This property has an equivalent characterization through the Walsh transform (originally due to Xiao and Massey [27]): $f$ is $t$-th order correlation immune if and only if $W_f(a) = 0$ for all coefficients $a \in \mathbb{F}_2^n$ of Hamming weight $1 \leq k \leq t$.

Correlation immunity plays an important role in the context of correlation attacks on stream ciphers based on the combiner model [24]. More recently, this criterion also gained relevance for designing masking countermeasures to withstand side-channel analysis. In this case, the goal is to find a $t$-th order correlation immune function to resist SCA attacks of order $t$. At the same time, it is desirable that this function has the lowest Hamming weight possible, to have an efficient implementation of the masking countermeasure.

Beside the Walsh transform, correlation immune functions have also a nice combinatorial characterization in terms of orthogonal arrays. Formally, an orthogonal array of $N$ runs, $k$ factors, $s$ levels and strength $t$ (denoted

as an $OA(N, k, s, t))$ is a $N \times n$ array with entries from a set $S$ with $s$ elements such that, for any $N \times t$ subarray, each $t$-uple of $S^t$ occurs exactly $\lambda = N/s^t$ times [7]. The value $\lambda$ is also called the index of the OA, and is completely determined by the other parameters. The link between binary OA (i.e. with $s = 2$ levels) and correlation immune functions is given by the following result proved in [1]:

**Lemma 1.** *A Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is correlation immune of order $t$ if and only if its support $supp(f) = \{x \in \mathbb{F}_2^n : f(x) \neq 0\}$ is an $OA(N, n, 2, t)$.*

In other words, one can reduce the design of $n$-variable, $t$-th order correlation immune Boolean functions for SCA masking countermeasures to the search of binary OA of $n$ factors and strength $t$. The requirement of minimizing the Hamming weight of the function corresponds to the minimization of the number of runs $N$ in the OA. Once such an OA has been found, one can define the corresponding correlation immune function $f$ by taking the runs of the OA as the vectors in the support of $f$.

## 2.2 Cellular Automata and Latin Squares

In this work, we use cellular automata (CA) as a particular kind of vectorial Boolean functions, which we formally define below:

**Definition 1.** *Let $n, d \in \mathbb{N}$ with $d \leq n$, and let $f : \mathbb{F}_2^d \to \mathbb{F}_2$ be a $d$-variable Boolean function. A cellular automaton of length $n$, diameter $d$ and local rule $f$ is a mapping $F : \mathbb{F}_2^n \to \mathbb{F}_2^{n-d+1}$ defined for all $x \in \mathbb{F}_2^n$ as:*

$$F(x_1, \cdots, x_n) = (f(x_1, \cdots, x_d), f(x_2, \cdots, x_{d+1}), \cdots, f(x_{n-d+1}, \cdots, x_n)) \ . \tag{5}$$

Intuitively, each output coordinate $i \in \{1, \cdots, n - d + 1\}$ of the CA $F$ is determined by evaluating $f$ on the local neighborhood $(x_i, \cdots, x_{i+d-1})$. Remark that the output is smaller than the input: indeed, we can apply $f$ as long as there are enough neighboring coordinates to the right of the current output cell $i$. Therefore, we do not enforce any boundary conditions on the CA state, as it is commonly done in the CA literature. This means that the CA cannot be iterated for multiple time steps, but this issue does not concern us since we are interested only in the single-step application of $F$. This model is also called no-boundary CA in related works [19, 16].

We now introduce the basic concepts related to Latin squares to recall the main results of [16]. For all $n \in \mathbb{N}$, let us denote by $[n] = \{1, \cdots, n\}$ the set of the first $n$ natural numbers. A Latin square of order $n$ is a $n \times n$ matrix $L$ such that each row and each column of $L$ is a permutation of $[n]$. Equivalently, this means that each number from 1 to $n$ occurs exactly once in each row and in each column of $L$. Then, two Latin squares $L_1, L_2$ of order $n$ are called orthogonal if their superposition yields all pairs in the

Cartesian product $[n] \times [n]$. Formally, for any distinct pairs of coordinates $(i_1, j_1), (i_2, j_2) \in [n] \times [n]$, one has:

$$(L_1(i_1, j_1), L_2(i_1, j_1)) \neq (L_1(i_2, j_2), L_2(i_2, j_2)) \ . \tag{6}$$

A set of $k$ Latin squares $L_1, \cdots, L_k$ of order $n$ that are pairwise orthogonal is also called a set of $k$-MOLS (mutually orthogonal Latin squares). The construction of MOLS is a rich research line in the combinatorial designs literature, and finds several applications in cryptography, coding theory and statistics [25]. Interestingly, $k$-MOLS of order $n$ are also equivalent to orthogonal arrays with $N = n^2$ runs, $k$ factors, $n$ levels and strength 2. Indeed, one can construct an $OA(N, k, n, 2)$ from a set of $k$-MOLS by "linearizing" each Latin square as a column of the OA: for each $i \in [k]$, the $i$-th column of the OA corresponds to the Latin square $L_i$ in the MOLS set, with its entries listed in lexicographic order.

The authors of [16] proposed a method to construct sets of $k$-MOLS from cellular automata, by focusing on the subclass of bipermutive rules. Formally, a local rule $f : \mathbb{F}_2^d \to \mathbb{F}_2$ is called bipermutive if it is defined as $f(x_1, \cdots, x_d) = x_1 \oplus \varphi(x_2, \cdots, x_{d-1}) \oplus x_d$ for all $x \in \mathbb{F}_2^d$, where $\varphi : \mathbb{F}_2^{d-2} \to \mathbb{F}_2$ is any function of the $d-2$ central variables. Then, one can construct a Latin square $L_F$ of order $n = 2^{d-1}$ from a CA $F : \mathbb{F}_2^{2(d-1)} \to \mathbb{F}_2^{d-1}$ as follows:

1. The left half of the CA input $(x_1, \cdots, x_{d-1})$ is used to index the rows of $L_F$.
2. The right half $(x_d, \cdots, x_{2(d-1)})$ is used to index the columns of $L_F$.
3. The output $F(x_1, \cdots, x_{2(d-1)})$ of the CA is used as the entry indexed respectively by the coordinates $(x_1, \cdots, x_{d-1})$ and $(x_d, \cdots, x_{2(d-1)})$.

Clearly, the procedure above assumes that a bijective mapping $\phi : \mathbb{F}_2^{d-1} \to [n]$ is used to convert the binary vectors of $\mathbb{F}_2^{d-1}$ in numbers from 1 to $2^{d-1}$, and vice versa. The authors of [16] focused on the construction of MOLS from bipermutive CA by further focusing on the class of linear rules. Here, however, we will consider the general setting of MOLS defined by generic bipermutive CA. In particular, we define two bipermutive CA $F_1, F_2 : \mathbb{F}_2^{2(d-1)} \to \mathbb{F}_2^{d-1}$ to be orthogonal (or equivalently they are OCA) if the corresponding Latin squares are orthogonal. Accordingly, a family of $k$ pairwise orthogonal bipermutive CA $F_1, \cdots, F_k$ will be called a set of $k$-MOCA (mutually orthogonal cellular automata).

## 3 Construction of Correlation Immune Functions

We now prove our main result, namely that a set of $k$-MOCA can be used to define a binary OA of strength at least 2. This will allow us, in turn, to construct correlation immune functions of order at least 2. To this end, let us first review the concept of coupled de Bruijn graph introduced in [18], which will be useful in our proof.

7

Recall that a de Bruijn graph of order $b$ over a set $S$ of $m$ symbols is defined as $G_{m,b} = (V, E)$, where $V = S^b$, while two vertices $u, v \in V$ are connected by a directed edge if and only if they overlap respectively on the rightmost and leftmost $b - 1$ coordinates. Assume now that $S = \mathbb{F}_2$, and let $b = d - 1$. A local rule $f : \mathbb{F}_2^d \to \mathbb{F}_2$ of diameter $d$ can be represented as a labeling function $l_f : E \to \mathbb{F}_2$ on the edges of $G_{2,b}$. In particular, for each pair $u, v \in E$, we set $l(u, v) = f(u \odot v)$, where $u \odot v \in \mathbb{F}_2^d$ represents the fusion of $u$ and $v$ as defined in [26]. In other words, $u \odot v$ is the $d$-variable vector formed by adding the last coordinate of $v$ to $u$. Then, it can be seen that the output of a CA equipped with rule $f$ corresponds to a path on the edges of the associated de Bruijn graph, following the overlapping vertices that form a particular input. Remark that if $f$ is bipermutive, then the labels of the outgoing (respectively, ingoing) edges of any vertex $v \in V$ form a permutation of $\mathbb{F}_2$. This implies that one can traverse the edges in both directions, and that the CA is surjective.

Suppose now that we have two labelings $l_f, l_g : E \to \mathbb{F}_2$ defined by bipermutive rules $f, g : \mathbb{F}_2^d \to \mathbb{F}_2$. We call the corresponding graph as the coupled de Bruijn graph associated to $f$ and $g$, since each label is now a pair of bits $(f(u \odot v), g(u \odot v))$ for each edge $(u, v) \in E$. We call two bipermutive labelings $l_f, l_g$ orthogonal if for each pair of vectors $(x, y) \in \mathbb{F}_2^b$ there exists exactly one path on the edges of the coupled de Bruijn graph that is labelled by $(x, y)$. It is not difficult to see that this is an equivalent characterization of orthogonal CA. We formally state this fact below, since it will become useful later in our proof:

**Lemma 2.** *Let $d \in \mathbb{N}$ with $b = d - 1$, and $f, g : \mathbb{F}_2^d \to \mathbb{F}_2$ be two bipermutive rules of diameter $d$. Then, the CA $F, G : \mathbb{F}_2^{2b} \to \mathbb{F}_2^b$ respectively equipped with rule $f$ and $g$ are orthogonal if and only if the labelings on the coupled de Bruijn graph of $f$ and $g$ are orthogonal.*

To fix ideas, let $d = 3$, and assume that the two bipermutive local rules are $f(x_1, x_2, x_3) = x_1 \oplus x_3$ and $g(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$ (respectively rules 90 and 150 in Wolfram's notation). The two rules induce orthogonal CA with Latin squares of order $2^2 = 4$, and the corresponding coupled de Bruijn graph is depicted in Figure 1. One can see that the two labelings $l_f, l_g$ defined in the table are indeed bipermutive and orthogonal. In particular, for each of the 16 pairs $(x, y) \in (\mathbb{F}_2^2)^2$ there is exactly one path of length 2 in $G_{2,2}$ labeled by $(x, y)$.

Let $F_1, F_2, \cdots, F_k : \mathbb{F}_2^{2b} \to \mathbb{F}_2^b$ be a set of $k$-MOCA respectively defined by local rules $f_1, \cdots, f_k : \mathbb{F}_2^d \to \mathbb{F}_2$ of diameter $d$, with $b = d - 1$. We define the $N \times n$ array $A$ where $N = 2^{2b}$ and $n = kb$ as follows: for each $(x, y) \in \mathbb{F}_2^{2b}$, the row of $A$ indexed by $(x, y)$ is equal to:

$$A(x, y) = (F_1(x, y), F_2(x, y), \cdots, F_k(x, y)) \ . \tag{7}$$
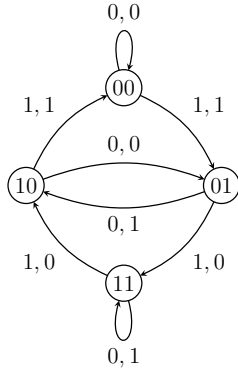
| $(v_1, v_2) \to (u_1, u_2)$ | $l_f$ | $l_g$ |
|:---:|:---:|:---:|
| $00 \to 00$ | 0 | 0 |
| $10 \to 00$ | 1 | 1 |
| $01 \to 10$ | 0 | 1 |
| $11 \to 10$ | 1 | 0 |
| $00 \to 01$ | 1 | 1 |
| $10 \to 01$ | 0 | 0 |
| $01 \to 11$ | 1 | 0 |
| $11 \to 11$ | 0 | 1 |

Figure 1: Example of orthogonal labelings for the de Bruijn graph $G_{2,2}$ induced by the CA local rules 90 and 150 of diameter $d = 3$.

In other words, $A$ is formed by simply juxtaposing the output of the $k$ MOCA for each possible combination of input $(x, y) \in \mathbb{F}_2^{2b}$. We now prove that this array is an OA of strength 2.

**Lemma 3.** *The array $A$ defined in Eq. (7) is an $OA(N, n, 2, 2)$, where the number of runs is $N = 2^{2b}$ and the number of factors is $n = kb$.*

*Proof.* We need to show that in any subset of $t = 2$ columns $i, j$ of $A$ each pair of bits $(x_i, x_j) \in \mathbb{F}_2^2$ occurs exactly $\lambda = N/2^t = 2^{2b-2}$ times. Without loss of generality, we can assume that $k = 2$, since $F_1, \cdots F_k$ is a family of $k$-MOCA. Hence, we have two main cases to check for two columns $i \neq j$:

1. $i, j$ belong to the output of the same CA $F_l$.
2. $i, j$ belong to the output of two different CA $F_l, F_m$ (which are orthogonal).

Let us start from the first case, i.e. $i$ and $j$ are chosen among the columns of the same CA $F_l$. Let $(\tilde{x}_i, \tilde{x}_j) \in \mathbb{F}_2^2$ be the value of the two bits of which we want to compute the multiplicity of occurrence in columns $i$ and $j$. Since the two CA $F_l$ and $F_m$ are orthogonal, it means that by fixing the output of $F_m$ to a specific vector $(y_1, \cdots, y_b) \in \mathbb{F}_2^b$, each possible vector $(x_1, \cdots, x_b) \in \mathbb{F}_2^b$ occurs exactly once as an output of $F_l$. Suppose now that we fix the $i$-th and $j$-th coordinates respectively to $\tilde{x}_i$ and $\tilde{x}_j$ in the output of $F_l$. Since we have $2^{b-2}$ free coordinates, it follows that the pair $(\tilde{x}_i, \tilde{x}_j)$ occurs $2^{b-2}$ times if we keep the output of $F_m$ fixed to $(y_1, \cdots, y_b)$. If we consider the occurrences of $(\tilde{x}_i, \tilde{x}_j)$ in $F_l$ across all possible outputs of $F_m$ we need to multiply $2^{b-2}$ by $2^b$, i.e. the number of possible ways to fix the output of $F_m$. Therefore the total number of occurrences is $2^{b-2} \cdot 2^b = 2^{2b-2}$.

Suppose now that $i$ and $j$ are columns respectively of $F_l$ and $F_m$. If all output coordinates of $F_l$ and $F_m$ are fixed respectively to $x$ and $y \in \mathbb{F}_2^b$, then there exists a single row of $A$ labeled by $x$ and $y$, since $F_l$ and $F_m$ are orthogonal, and by Lemma 2 there is a unique path on the coupled de

9

Bruijn graph labelled by $(x, y)$. We proceed by induction on the number of free coordinates in $(x, y)$ to show that if we only have two of them fixed, i.e. $\tilde{x}_i$ and $\tilde{y}_j$, then there are exactly $2^{2b-2}$ paths on the de Bruijn graph that feature $\tilde{x}_i$ and $\tilde{y}_j$ in those coordinates. As a base case, suppose that we have only one free coordinate in the pair of paths, i.e. all other $2b - 1$ are fixed. Then, since each of the two labelings is bipermutive, it follows that there are exactly 2 paths labelled by the $2b - 1$ fixed coordinates. For the induction step, suppose that there are $1 \leq p < 2b$ free coordinates, and thus $2^p$ paths labelled by the remaining $2b - p$ fixed coordinates by induction hypothesis. If we free an additional coordinate, we need to multiply the number of paths with $p$ free coordinates by 2, since each of them can be completed in 2 different ways in the additional free coordinate, due to the bipermutivity of the two rules. Hence, the number of partially labelled paths with $p + 1$ free coordinates is $2^{p+1}$. If we take the particular case where only 2 coordinates $i$ and $j$ are fixed respectively to $\tilde{x}_i$ and $\tilde{y}_j$ (or equivalently, $2b - 2$ are free), it follows that there are $2^{2b-2}$ paths partially labeled by $\tilde{x}_i$ and $\tilde{y}_j$. □                                                        □

Putting together Lemma (1) and 3, we have thus obtained the following result to construct a second-order correlation immune function from a set of $k$-MOCA:

**Theorem 1.** *Let $F_1, \cdots, F_k : \mathbb{F}_2^{2b} \to \mathbb{F}_2^b$ be a set of $k$-MOCA of diameter $d = b + 1$, and let $n = kb$. Then, the $n$-variable function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ whose support is defined by the array $A$ in Eq. (7) is correlation immune of order at least 2. In particular, the Hamming weight of $f$ is $N = 2^{2b}$.*

Remark that the case where $k = 2$ trivially gives the constant function $f(x) = 1$ for all $x \in \mathbb{F}_2^n$. As a matter of fact, the number of input variables is $n = kb = 2b$, which means that the truth table of $f$ is composed of $2^{2b}$ values. At the same time, the number of runs of the OA corresponding to $k = 2$ MOCA is also $N = 2^{2b}$. Therefore, the support of $f$ coincides with its whole truth table. For this reason, in what follows we will address mainly the case where $k = 3$.

## 4  Computational Search Results

To investigate the construction described in the previous section, we performed an exhaustive search of all $k$-MOCA for $k = 3$ and $d = 4, 5$. In particular, we discarded $d = 3$ since there are no families of 3-MOCA in that case. On the other hand, going for higher values of $k$ and $d$ makes the search space too huge to be exhaustively visited in a limited time. Following Theorem 1, this means that we addressed the construction of correlation immune functions of $n = 9, 12$ variables.

Table 1: Classification of correlation immune functions generated by 3-MOCA of diameter $d \in \{4, 5, 6\}$.

| $d$ | #3-MOCA | $n$ | $w_H$ | $CI$ | $\#CI$ | Min$w_H$ |
|---|---|---|---|---|---|---|
| 4 | 2 | 9 | 64 | 3 | 2 | 20 |
| 5 | 36 | 12 | 256 | 3 | 27 | 24 |
| 5 | 36 | 12 | 256 | 4 | 6 | 24 |

We first generated all pairs of OCA (i.e., 2-MOCA) of diameters $d = 4, 5$ by using the combinatorial algorithm described in [15]. Then, we incrementally constructed the families of 3-MOCA by exhaustively visiting each of the $2^{2^{d-2}}$ bipermutive rules of diameter $d$, and checking that the corresponding Latin squares were orthogonal with each of those in the previously generated lists of OCA. Next, we defined the corresponding Boolean functions of $n = 3b$ variables using Theorem 1, and computed their Walsh transform to verify their order of correlation immunity. Table 1 reports the main results of this computational search experiment. In particular, for each considered diameter ($d$) the table gives the number of 3-MOCA generated (#3-MOCA), the number of variables ($n$), the Hamming weight ($w_H$), the correlation immunity order ($CI$) and the number of functions achieving that order ($\#CI$), and the best known lower bound on the Hamming weight for the corresponding order of correlation immunity (Min($w_H$)), taken from [2].

From the table, one can see that all generated functions actually have a correlation immunity order higher than 2. Indeed, both functions of $n = 9$ variables generated from the 3-MOCA of diameter $d = 4$ have correlation immunity order 3, as the majority of functions of $n = 12$ functions obtained from 3-MOCA of diameter $d = 5$. Further, a few functions of $n = 12$ variables obtained from this latter case are even 4-th order correlation immune. This empirical observation suggests that the correlation immunity order proved in Theorem 1 may not be a tight lower bound. Further experiments on larger diameters should be performed to see if this hypothesis holds, or if there exist functions constructed through our methods that are effectively second-order correlation immune. Additionally, one can remark that the Hamming weight of the functions generated through our method are far from the known best lower bounds, reported in the last column of Table 1. Indeed, for $n = 9$ variables we obtain functions of weight 64, while the best lower bound is 20. For $n = 12$ variables the gap is even greater, since the weight of our functions is 256, while the lower bound is 24.

# 5  Conclusions

In this work, we proposed a method to construct correlation immune Boolean functions from sets of mutually orthogonal cellular automata. Our main theoretical result shows that the binary array formed by juxtaposing the output tables of a set of $k$ MOCA is an orthogonal array of strength 2. Hence, on account of Lemma 1 such an array can be used as the support of a Boolean function with correlation immunity order at least 2. Our exhaustive search experiments on the sets of 3-MOCA defined by rules of diameters $d = 4, 5$ show an interesting fact, namely that the resulting Boolean functions always have a higher order of correlation immunity, namely at least 3.

There are several directions along which this work can be extended in future research. The most natural open question remains whether our Theorem 1 is really a tight lower bound on the correlation immunity of the Boolean functions constructed through $k$-MOCA. If further experiments show that the lowest correlation immunity order is always at least 3, then it would be interesting to try refining Lemma 3, and prove that the binary OA obtained from a set of $k$-MOCA has always at least strength 3.

A second direction concerns the non-optimal weights of the correlation immune functions generated by our method. Indeed, one possible approach to improve on this aspect would be to adopt the so-called expurgation procedure used in the field of error-correcting codes. In the context of orthogonal arrays, this basically amounts to the removal of a subset of rows, such that the resulting array is still an OA of the same strength (but clearly with a smaller index $\lambda$). The choice of the rows to be removed can be conceived as a combinatorial optimization problem, which could be addressed with different optimization algorithms.

# References

[1] P. Camion, C. Carlet, P. Charpin, and N. Sendrier. On correlation-immune functions. In J. Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 86–100. Springer, 1991.

[2] C. Carlet. *Boolean Functions for Cryptography and Coding Theory.* Cambridge University Press, 2021.

[3] C. Carlet, J. Danger, S. Guilley, and H. Maghrebi. Leakage squeezing of order two. In S. D. Galbraith and M. Nandi, editors, *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings*, volume

7668 of *Lecture Notes in Computer Science*, pages 120–139. Springer, 2012.

[4] P. P. Chaudhuri, D. R. Chowdhury, S. Nandi, and S. Chattopadhyay. *Additive Cellular Automata: Theory and Applications, Volume 1*, volume 43. John Wiley & Sons, 1997.

[5] Y. Crama and P. L. Hammer. *Boolean Functions - Theory, Algorithms, and Applications*, volume 142 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2011.

[6] E. Formenti, K. Imai, B. Martin, and J. Yunès. Advances on random sequence generation by uniform cellular automata. In *Computing with New Resources*, volume 8808 of *Lecture Notes in Computer Science*, pages 56–70. Springer, 2014.

[7] A. S. Hedayat, N. J. A. Sloane, and J. Stufken. *Orthogonal arrays: theory and applications*. Springer Science & Business Media, 1999.

[8] A. John, B. C. Nandu, A. Ajesh, and J. Jose. PENTAVIUM: potent trivium-like stream cipher using higher radii cellular automata. In T. M. Gwizdalla, L. Manzoni, G. C. Sirakoulis, S. Bandini, and K. Podlaski, editors, *Cellular Automata - 14th International Conference on Cellular Automata for Research and Industry, ACRI 2020, Lodz, Poland, December 2-4, 2020, Proceedings*, volume 12599 of *Lecture Notes in Computer Science*, pages 90–100. Springer, 2020.

[9] S. Karmakar and D. R. Chowdhury. Countermeasures of side channel attacks on symmetric key ciphers using cellular automata. In G. C. Sirakoulis and S. Bandini, editors, *Cellular Automata - 10th International Conference on Cellular Automata for Research and Industry, ACRI 2012, Santorini Island, Greece, September 24-27, 2012. Proceedings*, volume 7495 of *Lecture Notes in Computer Science*, pages 623–632. Springer, 2012.

[10] S. Karmakar and D. R. Chowdhury. Leakage squeezing using cellular automata and its application to scan attack. *J. Cell. Autom.*, 9(5-6):417–436, 2014.

[11] S. Karmakar and D. R. Chowdhury. Scan-based side channel attack on stream ciphers and its prevention. *J. Cryptogr. Eng.*, 8(4):327–340, 2018.

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

[13] R. Lakra, A. John, and J. Jose. Carpenter: A cellular automata based resilient pentavalent stream cipher. 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 352–363. Springer, 2018.

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

[15] 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.

[16] 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.

[17] L. Mariot and A. Leporati. A cryptographic and coding-theoretic perspective on the global rules of cellular automata. *Nat. Comput.*, 17(3):487–498, 2018.

[18] L. Mariot and A. Leporati. Inversion of mutually orthogonal cellular automata. In *Proceedings of ACRI 2018*, volume 11115 of *Lecture Notes in Computer Science*, pages 364–376. Springer, 2018.

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

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

[21] W. Meier and O. Staffelbach. Fast correlation attacks on stream ciphers (extended abstract). In C. G. Günther, editor, *Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, volume 330 of *Lecture Notes in Computer Science*, pages 301–314. Springer, 1988.

[22] W. Meier and O. Staffelbach. Analysis of pseudo random sequence generated by cellular automata. In *Proceedings of EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 186–199. Springer, 1991.

[23] M. Nassar, Y. Souissi, S. Guilley, and J. Danger. RSM: A small and fast countermeasure for aes, secure against 1st and 2nd-order zero-offset scas. In W. Rosenstiel and L. Thiele, editors, *2012 Design, Automation & Test in Europe Conference & Exhibition, DATE 2012, Dresden, Germany, March 12-16, 2012*, pages 1173–1178. IEEE, 2012.

[24] T. Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. *IEEE Trans. Computers*, 34(1):81–85, 1985.

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

[26] K. Sutner. De bruijn graphs and linear cellular automata. *Complex Syst.*, 5(1), 1991.

[27] G. Xiao and J. L. Massey. A spectral characterization of correlation-immune combining functions. *IEEE Trans. Inf. Theory*, 34(3):569–571, 1988.