The final publication is available at Springer via http://dx.doi.org/10.1007/s11047-017-9635-0

A Cryptographic and Coding-Theoretic Perspective on the Global Rules of Cellular Automata

Luca Mariot, Alberto Leporati

Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Viale Sarca 336, 20126 Milano, Italy
{luca.mariot,leporati}@disco.unimib.it

Abstract. Cellular Automata (CA) have widely been studied to design cryptographic primitives such as stream ciphers and pseudorandom number generators, focusing in particular on the properties of the underlying local rules. On the other hand, there have been comparatively fewer works concerning the applications of CA to the design of S-boxes and block ciphers, a task that calls for a study of CA global rules in terms of vectorial boolean functions. The aim of this paper is to analyze some of the most basic cryptographic criteria of the global rules of CA. We start by observing that the algebraic degree of a CA global rule equals the degree of its local rule. Then, we characterize the Walsh spectrum of CA induced by permutive local rules, from which we derive a formula for the nonlinearity of such CA. Additionally, we prove that the 1-resiliency property of bipermutive local rules transfers to the corresponding global rules. This result leads us to consider CA global rules from a coding-theoretic point of view: in particular, we show that linear CA are equivalent to linear cyclic codes, observing that the syndrome computation process corresponds to the application of the CA global rule, while the error-correction capability of the code is related to the resiliency order of the global rule.

Keywords: cellular automata, boolean functions, S-boxes, nonlinearity, resiliency, cyclic codes

1 Introduction

Block ciphers constitute one of the most fundamental building blocks in the design of several cryptographic protocols. The security of block ciphers frequently depends on the involved *Substitution boxes* (S-boxes), which can be considered as *vectorial boolean functions*. As a matter of fact, S-boxes are usually the only nonlinear component in a block cipher. Thus, particular care must be taken in choosing S-boxes with good cryptographic properties, so that the overall block cipher design can withstand particular attacks like linear and differential cryptanalysis.

Cellular Automata (CA) are a nature-inspired parallel computational model initially introduced by [21] and [22] to study self-reproduction phenomena. CA represent an interesting computational model for developing S-boxes, for a twofold reason: first, depending on the local rule, CA can exhibit chaotic and unpredictable dynamic behaviors, a characteristic which is useful to achieve the *confusion principle* set forth by [17] that

every secure symmetric cryptosystem should satisfy. Second, being a massively parallel model, CA can be efficiently realized in hardware, and thus they are interesting for implementing S-boxes on devices with limited computational resources.

However, one can observe that most of the literature pertaining cryptographic applications of CA is centered on the design of *stream ciphers* and *pseudorandom number generators*. In fact, it was [24] who pioneered the use of CA for keystream generation, using the elementary rule 30. However, the design was discovered to be insecure first by [14], and then by [8]. In particular, Meier and Staffelbach showed a correlation attack on the sequences produced by Wolfram's generator which exploited the fact that rule 30 is not 1-resilient, while Koc and Apohan described an inversion attack based on the low nonlinearity of such rule. Since then, some researchers (see [12,9,6]) focused on the search of CA local rules having good cryptographic profiles in order to thwart these kinds of attacks, but retaining Wolfram's overall design of CA pseudorandom generator.

On the other hand, the design of S-boxes based on CA is a research topic which has received relatively little attention in the literature. This could be the reason why, at least as far as our knowledge goes, there is almost no work concerning the cryptographic properties of CA *global rules*. One remarkable exception in this regard is [4], where the authors analyzed the propagation and correlation characteristics of a CA equipped with rule χ, which corresponds to the elementary rule 210 in Wolfram's numbering convention (see [23]). Interestingly, χ is an example of a CA-based S-box employed in real-world applications, since it is the only nonlinear component of the KECCAK sponge construction, selected by the NIST as the SHA-3 standard for cryptographic hash functions (see [1]).

The aim of this paper, which is an extended version of [11], is to undertake an investigation of the cryptographic properties of CA global rules by considering them as a particular kind of vectorial boolean functions, and to relate them to the properties of the underlying local rules. To this end, we consider criteria that are relevant both for the design of S-boxes in block ciphers, like *nonlinearity*, and for stream ciphers, like *resiliency*. In addition, we also exploit the connection between resiliency and minimum distance of *linear codes* to analyze CA from the standpoint of coding theory. Nevertheless, the motivation for this coding theoretic aspect of our work is again related to cryptography, since certain classes of linear codes (especially MDS codes) can be used to implement the diffusion layer of block ciphers.

To begin with, we first observe that the algebraic degree of the global rule of a CA equals the algebraic degree of its local rule, leveraging on the fact that the coordinate functions of a CA correspond to its local rule applied to different neighborhoods. Next, we narrow our attention to the class of CA equipped with *permutive* rules, a property which allows us to characterize the *Walsh spectrum* of the CA global rule. In particular, we show how the Walsh spectrum in a left or right permutive CA changes by adding a new cell. From this result, we then prove that the nonlinearity of a left or right permutive CA with m output cells is 2^{m-1} times the nonlinearity of the local rule. Subsequently, we show that the global rules of *bipermutive CA* are always at least 1-resilient, thus generalizing the result in [9] about bipermutive local rules. We then prove an equivalence between *linear CA* and linear *cyclic codes*. In particular, we show how the systematic encoding of cyclic codes actually corresponds to the preimage computation process of

the all-zeros configuration in linear CA, while syndrome computation is equivalent to the application of the CA global rule. Leveraging on the theory of resilient vectorial functions, we remark that the resiliency order of a linear CA can be used to determine the minimum distance of its associated cyclic code, and we show as an example how encoding and decoding of the $(7,4,3)$ cyclic Hamming code can be realized using the dynamics of a CA with radius $r = 2$ and length $n = 7$.

The rest of the paper is organized as follows. Section 2 collects some basic facts about vectorial boolean functions and their cryptographic criteria, and introduces the model of cellular automaton we adopt throughout the paper. Section 3 is devoted to the analysis of the global rules of CA, focusing on their algebraic degree, nonlinearity and resiliency order. Section 4 recalls some key concepts about the theory of error-correcting codes, presents the connection between linear cyclic codes and linear CA and shows how to simulate the $(7,4,3)$ cyclic Hamming codes using linear CA. Finally, Section 5 summarizes the main contributions of the paper and discusses some directions for future research on the topic.

2 Preliminary Definitions

In this section, we outline the basic concepts concerning vectorial boolean functions and cellular automata which we use in the remainder of the paper.

2.1 Vectorial Boolean Functions

We cover only the fundamental definitions and results related to the theory of cryptographic boolean functions, referring the reader to [2,3] for a more thorough treatment of the subject.

A *boolean function* is a mapping $f : \mathbb{F}_2^n \to \mathbb{F}_2$ where $\mathbb{F}_2$ denotes the finite field with two elements. The basic way to represent a boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is by means of its *truth table*, which specifies for each of the possible 2^n input vectors of $\mathbb{F}_2^n$ the corresponding output value of f. Hence, for any $n \in \mathbb{N}$ the set of boolean functions of n variables is composed of 2^{2^n} functions. Once an ordering of the input variables $x_1, \cdots, x_n$ has been established, a truth table can be compactly described just by the 2^n-bit string representing the output values of the corresponding function.

Another common representation of boolean functions is the *Algebraic Normal Form* (ANF). In particular, the ANF of a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined by the following multivariate polynomial:

$$P_f(x) = \bigoplus_{I \in \mathscr{P}(N)} a_I \left(\prod_{i \in I} x_i \right) \ , \tag{1}$$

where $N = \{1, \cdots, n\}$ and $\mathscr{P}(N)$ denotes the power set of N, and $a_I \in \mathbb{F}_2$ for all $I \in \mathscr{P}(N)$. Hence, the ANF represents a boolean function as a sum of products over $\mathbb{F}_2$. The relationship between the ANF coefficients and the truth table of f is given by the *Möbius transform*, defined for all $x \in \mathbb{F}_2^n$ as:

$$f(x) = \bigoplus_{I \subseteq supp(x)} a_I \ , \tag{2}$$

where $supp(x) = \{i : x_i \neq 0\}$ is the *support* of x.

A third representation which is useful to characterize several cryptographic properties of boolean functions is the *Walsh transform*. Given $f : \mathbb{F}_2^n \to \mathbb{F}_2$, the Walsh transform of f is the function $W_f : \mathbb{F}_2^n \to \mathbb{R}$ defined for all $\omega \in \mathbb{F}_2^n$ as

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

where $\omega \cdot x = \omega_1 x_1 \oplus \cdots \omega_n x_n$ is the *scalar product* of ω and x. The value $W_f(\omega)$ is also called the *Walsh coefficient of* f with respect to $\omega \in \mathbb{F}_2^n$. The set of all Walsh coefficients of f is the *Walsh spectrum* of f, while the maximum coefficient in absolute value is called the *spectral radius* of f.

The boolean functions adopted in cryptography must satisfy several criteria in order to resist various types of attacks. In this paper we consider four cryptographic properties, namely *balancedness*, *algebraic degree*, *nonlinearity* and *resiliency*, which we briefly define below along with a description of the corresponding design criterion.

A boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is *balanced* if its truth table is composed of an equal number of 0s and 1s, or equivalently if its Walsh transform vanishes on the null vector, i.e. $W_f(\underline{0}) = 0$. As a general criterion, all boolean functions used in the design of both stream and block ciphers should be balanced.

The *algebraic degree* of a boolean function f is the degree of its ANF. Considering Equation (1), the degree of f can formally defined as:

$$deg(f) = max_{I \in \mathscr{P}(N)} \{|I| : a_I \neq 0\} \quad . \tag{4}$$

Functions having degree 1 are also called *affine* functions. As a cryptographic criterion, the algebraic degree of boolean functions used in both stream and block ciphers should be as high as possible.

The *nonlinearity* of $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is the minimum Hamming distance of f from the set of affine functions, and it is defined through the Walsh transform by the following formula:

$$Nl(f) = 2^{n-1} - \frac{1}{2} \max_{\omega \in \mathbb{F}_2^n} \{|W_f(\omega)|\} \quad . \tag{5}$$

Similarly to the algebraic degree criterion, the nonlinearity of boolean functions involved in stream and block ciphers should be as high as possible.

Finally, a boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is said to be *t-resilient* if, by fixing at most t input coordinates, the resulting restriction of f is balanced. This is equivalent to say that the Walsh transform of f vanishes for all those input vectors ω having Hamming weight at most t. As a cryptographic criterion, the resiliency of boolean functions of stream ciphers should be as high as possible, to avoid correlation attacks. Notice that the case $t = 0$ corresponds to balancedness.

We now turn our attention to vectorial boolean functions. Let $n \geq m$. A *vectorial boolean function* (also called a *S-box* in the cryptographic context) is a mapping $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ with n input variables and m outputs. By $f_1, \cdots, f_m : \mathbb{F}_2^n \to \mathbb{F}_2$ we denote the *coordinate functions* of F, that is, the m boolean functions which specify the value of each output bit of F:

$$F(x_1, \cdots, x_n) = (f_1(x_1, \cdots, x_n), \cdots, f_m(x_1, \cdots, x_n)) \quad .$$

The *component functions* of F are defined as $v \cdot F$ for all $v \in (\mathbb{F}_2^m)^* = \mathbb{F}_2^m \setminus \{\underline{0}\}$. Since

$$v \cdot F = v_1 f_1(x_1, \cdots, x_n) \oplus \cdots \oplus v_m f_m(x_1, \cdots, x_n) \ ,$$

it follows that the component functions are the (non-trivial) linear combinations of the coordinate functions of F.

In the remainder of this section, we show how the vectorial counterparts of the cryptographic properties of boolean functions are characterized in terms of either the coordinates or the component functions of S-boxes.

A vectorial boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is balanced if for all output vectors $y \in \mathbb{F}_2^m$ the cardinality of the fiber $F^{-1}(y)$ is 2^{n-m}. Equivalently, F is balanced if and only if all its component functions are balanced.

The algebraic degree of a vectorial function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is defined as the maximal degree of its coordinate functions. On the other hand, the nonlinearity of F is the *minimal* nonlinearity of all its component functions, i.e.

$$Nl(F) = min_{v \in (\mathbb{F}_2^n)^*} \left\{ 2^{n-1} - \frac{1}{2} \max_{\omega \in \mathbb{F}_2^n} \{ |W_{v \cdot F}(\omega)| \} \right\} \ . \tag{6}$$

Resiliency for vectorial functions is defined analogously to the single-output case. In particular, $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is *t-resilient* if, by fixing any t input variables $x_{i_1}, \cdots, x_{i_t}$, the resulting restriction $\tilde{F} : \mathbb{F}_2^{n-t} \to \mathbb{F}_2^m$ is balanced, i.e. for all $y \in \mathbb{F}_2^m$ it follows that $|\tilde{F}^{-1}(y)| = 2^{n-t-m}$. Note that the definition of vectorial t-resiliency is actually equivalent to t-resiliency for boolean functions. Similarly to nonlinearity and balancedness, the resiliency of a vectorial function can also be characterized by the resiliency of its component functions, as the next result reported in [3] shows:

Proposition 1. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a vectorial boolean function in n variables and m outputs. Then, F is t-resilient if and only if for all $v \in (\mathbb{F}_2^m)^*$ the component function $v \cdot F$ is t-resilient.*

2.2 Cellular Automata

In what follows, we consider exclusively one-dimensional boolean cellular automata, formally defined below.

Definition 1. *A* one-dimensional boolean cellular automaton *(CA) is a triple $\langle C, \delta, f \rangle$, where C is a finite one-dimensional array of binary* cells*, $\delta \in \mathbb{N}$ is the* diameter *and $f : \mathbb{F}_2^\delta \to \mathbb{F}_2$ is the* local rule*.*

Given an array C of length $n \geq \delta$, the update of a CA is done as follows. If the diameter δ is odd with $\delta = 2r+1$ for $r \in \mathbb{N}$, then each cell i in the range $\{r+1, \cdots, n-r\}$ synchronously updates its state by applying rule f to the neighborhood $\{i-r, \cdots, i, \cdots, i+r\}$. Otherwise, if δ is even and $r = \delta/2$, then each cell i in the range $\{r, \cdots, n-r\}$ synchronously updates its state by applying rule f to the neighborhood $\{i-r+1, \cdots, i+r\}$. In both cases, the parameter r is called the *radius* of the CA.

From the discussion above, one can observe that we do not consider any *boundary condition* in our definition of CA, since only the central cells having sufficiently

enough left and right neighbors are allowed to update their states. This contrasts with the approach usually adopted in the CA literature, in which *null* or *periodic* boundary conditions are considered (see for example [7]), which makes the CA having the same number of input and output cells. In particular, periodic boundary conditions are commonly used in the design of CA-based S-boxes, as in the case of the CA χ employed in KECCAK. This is because several block ciphers are based on the *Substitution-Permutation Network* paradigm, where decryption depends on the fact that the involved S-boxes are invertible (thus implying an equal number of input and output bits). However, our CA model without boundary conditions does not limit the cryptographic applicability of the results presented in this paper, since there are also block ciphers models where decryption does not rely on the invertibility of the underlying S-boxes (such as for example in *Feistel ciphers*, see [19]).

We now define the *global rule* of a CA:

Definition 2. *The* global rule *of a CA* $\langle C, \delta, f\rangle$ *of length* $n = m+\delta-1$ *is the vectorial function* $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ *defined for all possible states* $x = (x_1, \cdots, x_n) \in \mathbb{F}_2^n$ *of array C as follows:*

$$F(x) = (f(x_1, \cdots, x_\delta), \cdots, f(x_{n-\delta+1}, \cdots, x_n)) \ . \tag{7}$$

In what follows, we identify a CA $\langle C, \delta, f\rangle$ with its global rule $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$.

Since the local rule of a CA is a boolean function of δ variables, the most common way to represent it is by means of its truth table. Another convenient way of representing a CA rule f is through its *Wolfram code* (see [23]), which is the decimal encoding of the truth table of f.

3 Cryptographic Properties of CA Global Rules

In this section we investigate the cryptographic properties of CA global rules, starting from their algebraic degree. We then introduce the class of *permutive* CA, and use this additional property to characterize the Walsh spectra of the component functions of such CA. As a consequence, this result allows us to determine a formula for the nonlinearity of the global rules of permutive CA. Finally, we employ the quasi-linearity of *permutive* local rules to prove that *bipermutive* CA are always at least 1-resilient. This last result generalizes the work of [9] that was carried out on bipermutive local rules to the case of global rules.

3.1 Algebraic Degree

We begin with the following remark:

Remark 1. Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a one-dimensional boolean cellular automaton of length $n = m+\delta-1$ defined by a local rule $f : \mathbb{F}_2^\delta \to \mathbb{F}_2$ of diameter δ. Since each output cell y_i depends only on the input cells $x_i, \cdots, x_{i+\delta-1}$ under application of the local rule, the coordinate functions of F are $f_i(x_1, \cdots, x_n) = f(x_i, \cdots, x_{i+\delta-1})$ for $i \in \{1, \cdots, m\}$.

Since the algebraic degree of a vectorial boolean function equals the maximal degree of its coordinate functions, we obtain the following result:

Proposition 2. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a CA with $n = m + \delta - 1$ defined by a local rule $f : \mathbb{F}_2^\delta \to \mathbb{F}_2$. Then, the algebraic degree of F equals the degree of f.*

Proof. For $k \in \{1, \cdots, m\}$, define $N_k = \{k, \cdots, k+\delta-1\}$ and let us denote by $\mathscr{P}(N_k)$ the power set of N_k. Notice that $N_1 = N$, where N is the index set for the ANF of the local rule f. For all $I = \{I_1, \cdots, I_j\} \in \mathscr{P}(N)$, let us define the shifted *subset of I as $\sigma_k(I) = \{I_1 + k - 1, \cdots, I_j + k - 1\}$, which ranges in the power set $\mathscr{P}(N_k)$. On the other hand, given $L \in \mathscr{P}(N_k)$ one can recover the original subset $I \in \mathscr{P}(N)$ by computing $I = \sigma_{-k}(L) = \{L_1 - k + 1, \cdots, L_j - k + 1\}$. Then, by Equation* (1) *we have that*

$$P_{f_k}(x) = \bigoplus_{L \in \mathscr{P}(N_k)} a_L \left(\prod_{l \in L} x_l \right) \quad . \tag{8}$$

Since for every $L \in \mathscr{P}(N_k)$ there exists $I \in \mathscr{P}(N)$ such that $I = \sigma_{-k}(L)$, by Remark 1 it also follows that $a_L = a_I$, so we can rewrite (8) *as:*

$$P_{f_k}(x) = \bigoplus_{L \in \mathscr{P}(N_k)} a_I \left(\prod_{l \in L} x_l \right) \quad , \text{ where } I = \sigma_{-k}(L). \tag{9}$$

Since the shifting operation does not change the cardinality of subsets, we have

$$max_{I \in \mathscr{P}(N)}\{|I| : a_I \neq 0\} = max_{L \in \mathscr{P}(N_k)}\{|L| : a_I \neq 0\} \quad , \tag{10}$$

from which one obtains that $deg(f_k) = deg(f_1) = deg(f)$. □

3.2 Walsh Spectra and Nonlinearity of Permutive CA

The result about the algebraic degree laid out in the previous section holds for CA with generic local rules. In what follows, we narrow our analysis to CA equipped with *permutive* local rules, showing that in this case further information can be obtained on the Walsh spectra of the associated global rules. This allows us to express the nonlinearity of permutive global rules in terms of the nonlinearity of their local rules.

We first recall the notion of permutive boolean function. To this end, let us denote by $(x, \tilde{x}), (\tilde{x}, x) \in \mathbb{F}_2^n$ the two vectors of length n obtained by appending $x \in \mathbb{F}_2$ respectively to the left and to the right of $\tilde{x} \in \mathbb{F}_2^{n-1}$. Then, permutive functions are formally defined as follows:

Definition 3. *A boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is called* left permutive *(respectively,* right permutive*) if, for all $\tilde{x} \in \mathbb{F}_2^{n-1}$ and $x, x' \in \mathbb{F}_2$ such that $x \neq x'$, it results that $f(x, \tilde{x}) \neq f(x', \tilde{x})$ (respectively, $f(\tilde{x}, x) \neq f(\tilde{x}, x')$). A function which is both left and right permutive is called* bipermutive*.*

As shown in [9], permutive functions have a simple characterization in terms of generating functions. In particular, $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is left permutive if there exists a function $g : \mathbb{F}_2^{n-1} \to \mathbb{F}_2$ of $n-1$ variables (called the *generating function* of f) such that

$$f(x_1, x_2, \cdots, x_n) = x_1 \oplus g(x_2, \cdots, x_n) \quad , \tag{11}$$

for all $x = (x_1, x_2, \cdots, x_n)$. Right permutive functions are characterized symmetrically by XORing x_n with the value of the generating function computed on the leftmost $n-1$ variables. Hence, a bipermutive function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be equivalently defined by a generating function $g : \mathbb{F}_2^{n-2} \to \mathbb{F}_2$ of $n-2$ variables such that

$$f(x_1, x_2, \cdots, x_{n-1}, x_n) = x_1 \oplus g(x_2, \cdots, x_{n-1}) \oplus x_n \ . \tag{12}$$

The next result shows how the Walsh coefficients of the component functions in a permutive CA are affected by adding a new cell. We state and prove the theorem just for the right permutive case, since the left permutive one can be obtained by a simple symmetrical argument.

Theorem 1. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a CA of length $n = m + \delta - 1$ defined by a right permutive local rule $f : \mathbb{F}_2^\delta \to \mathbb{F}_2$ with diameter δ. Additionally, let $F' : \mathbb{F}_2^{n+1} \to \mathbb{F}_2^{m+1}$ be the corresponding CA of length $n+1$ obtained by appending an additional cell to the right, and let $v \cdot F'$ be the component function of F' determined by $v = (\tilde{v}, v_{n+1}) \in (\mathbb{F}_2^{m+1})^*$, with $\tilde{v} \in \mathbb{F}_2^n$ and $v_{n+1} \in \mathbb{F}_2$. Then, for all $\tilde{\omega} \in \mathbb{F}_2^n$ and $\omega_{n+1} \in \mathbb{F}_2$, the Walsh coefficient of F' over $\omega = (\tilde{\omega}, \omega_{n+1})$ can assume only the following values:*

- $W_{v \cdot F'}(\omega) = 0$
- $W_{v \cdot F'}(\omega) = 2 \cdot W_{\tilde{\omega} \cdot F}(\omega)$.

Proof. We proceed by induction on $m \in \mathbb{N}$.

Let $m = 1$. We have that $F = f$, i.e. the global rule of the CA corresponds to its local rule, and by appending a cell to the right we obtain a CA F' with $\delta + 1$ input cells and 2 output cells. We will show that in this case $W_{v \cdot F'}(\omega) = 0$ or $W_{v \cdot F'}(\omega) = 2 \cdot W_f(\omega)$ for all $\omega \in \mathbb{F}_2^{\delta+1}$ and for all component functions $v \cdot F'$. Since $m + 1 = 2$, there is a total of $2^2 - 1 = 3$ component functions to consider, namely those determined by the vectors $(1,0)$, $(0,1)$ and $(1,1)$. Assume that $v = (1,0)$. Then, the component function in this case coincides with local rule f computed on the input variables $x_1, \cdots, x_\delta$ of F'. By Equation (3)*, this means that for $\tilde{\omega} \in \mathbb{F}_2^\delta$ and $\omega_{\delta+1} \in \mathbb{F}_2$ the Walsh coefficient of $v \cdot F'$ over $\omega = (\tilde{\omega}, \omega_{\delta+1})$ is:*

$$\begin{aligned} W_{v \cdot F'}(\omega) &= \sum_{x \in \mathbb{F}_2^{\delta+1}} (-1)^{v \cdot F'(x) \oplus \omega \cdot x} = \\ &= \sum_{x \in \mathbb{F}_2^{\delta+1}} (-1)^{f(x_1, \cdots, x_\delta) \oplus \omega \cdot x} \ . \end{aligned} \tag{13}$$

Since $\omega \cdot x = \omega_1 x_1 \oplus \cdots \oplus \omega_\delta x_\delta \oplus \omega_{\delta+1} x_{\delta+1}$, we can split Equation (13) *by grouping the terms with $x_{\delta+1} = 0$ in one sum and the terms with $x_{\delta+1} = 1$ in another sum.*

$$\begin{aligned} W_{v \cdot F'}(\omega) = &\sum_{\substack{x \in \mathbb{F}_2^{\delta+1}: \\ x_{\delta+1} = 0}} (-1)^{f(x_1, \cdots, x_\delta) \oplus \omega_1 x_1 \oplus \cdots \oplus \omega_\delta x_\delta} \\ &+ \sum_{\substack{x \in \mathbb{F}_2^{\delta+1}: \\ x_{\delta+1} = 1}} (-1)^{f(x_1, \cdots, x_\delta) \oplus \omega_1 x_1 \oplus \cdots \oplus \omega_\delta x_\delta \oplus \omega_{\delta+1}} \ . \end{aligned} \tag{14}$$

Notice that the term $\omega_{\delta+1}$ *in the exponent of the second sum of* (14) *corresponds to a multiplicative constant* $(-1)^{\omega_{\delta+1}}$*, which can thus be extracted from the sum:*

$$W_{v\cdot F'}(\omega) = \sum_{\substack{x\in\mathbb{F}_2^{\delta+1}:\\ x_{\delta+1}=0}} (-1)^{f(x_1,\cdots,x_\delta)\oplus\omega_1 x_1\oplus\cdots\oplus\omega_\delta x_\delta} + (-1)^{\omega_{\delta+1}}\cdot\sum_{\substack{x\in\mathbb{F}_2^{\delta+1}:\\ x_{\delta+1}=1}} (-1)^{f(x_1,\cdots,x_\delta)\oplus\omega_1 x_1\oplus\cdots\oplus\omega_\delta x_\delta} \quad . \tag{15}$$

Remark now that the two sums in (15) *are the same and correspond to the Walsh coefficient* $W_f(\tilde{\omega})$ *of rule* f*:*

$$W_{v\cdot F'}(\omega) = W_f(\tilde{\omega}) + (-1)^{\omega_{\delta+1}}\cdot W_f(\tilde{\omega}) \quad . \tag{16}$$

Therefore, it results that $W_{v\cdot F'}(\omega) = 2\cdot W_{v\cdot F'}(\omega)$ *if* $\omega_{\delta+1}=0$*, and* $W_{v\cdot F'}(\omega)=0$ *when* $\omega_{\delta+1}=1$*, which proves the statement for* $v=(1,0)$*. An analogous argument holds also for* $v=(0,1)$*. Hence, to conclude the base of the induction, it remains to be analyzed the case* $v=(1,1)$*, where the Walsh coefficient of* $v\cdot F'$ *over* $\omega=(\tilde{\omega},\omega_{\delta+1})\in\mathbb{F}_2^{\delta+1}$ *equals:*

$$W_{v\cdot F'}(\omega) = \sum_{x\in\mathbb{F}_2^{\delta+1}} (-1)^{v\cdot F'(x)\oplus\omega\cdot x} = = \sum_{x\in\mathbb{F}_2^{\delta+1}} (-1)^{f(x_1,\cdots,x_\delta)\oplus f(x_2,\cdots,x_{\delta+1})\oplus\omega\cdot x} \quad . \tag{17}$$

Like in the previous case, we split the sum of Equation (17) *with respect to the value of* $x_{\delta+1}$ *and extract the multiplicative constant* $(-1)^{\omega_{\delta+1}}$ *from the second sum. Denoting by* $\tilde{x}=(x_1,\cdots,x_\delta)$*, this yields:*

$$W_{v\cdot F'}(\omega) = \sum_{\substack{x\in\mathbb{F}_2^{\delta+1}:\\ x_{\delta+1}=0}} (-1)^{f(x_1,\cdots,x_\delta)\oplus f(x_2,\cdots,0)\oplus\tilde{\omega}\cdot\tilde{x}} + (-1)^{\omega_{\delta+1}}\cdot\sum_{\substack{x\in\mathbb{F}_2^{\delta+1}:\\ x_{\delta+1}=1}} (-1)^{f(x_1,\cdots,x_\delta)\oplus f(x_2,\cdots,1)\oplus\tilde{\omega}\cdot\tilde{x}} \quad . \tag{18}$$

By separating the terms $f(x_2,\cdots,0)$ *and* $f(x_2,\cdots,1)$ *in the exponents of Equation* (18)*, we obtain:*

$$W_{v\cdot F'}(\omega) = \sum_{\substack{x\in\mathbb{F}_2^{\delta+1}:\\ x_{\delta+1}=0}} (-1)^{f(x_1,\cdots,x_\delta)\oplus\tilde{\omega}\cdot\tilde{x}}\cdot(-1)^{f(x_2,\cdots,0)} + (-1)^{\omega_{\delta+1}}\cdot\sum_{\substack{x\in\mathbb{F}_2^{\delta+1}:\\ x_{\delta+1}=1}} (-1)^{f(x_1,\cdots,x_\delta)\oplus\tilde{\omega}\cdot\tilde{x}}\cdot(-1)^{f(x_2,\cdots,1)} \tag{19}$$

Notice that the two sums in Equation (19) *correspond to the Walsh coefficient* $W_f(\omega)$*, with the exception that in the first sum each term is multiplied by* $(-1)^{f(x_2,\cdots,0)}$ *and in*

the second by $(-1)^{f(x_2,\cdots,1)}$. *Since* f *is right permutive, we have that* $(-1)^{f(x_2,\cdots,0)} \neq (-1)^{f(x_2,\cdots,1)}$ *for all* $(x_2,\cdots,x_\delta) \in \mathbb{F}_2^{\delta-1}$. *It follows that for each vector* $\tilde{x} \in \mathbb{F}_2^{\delta}$ *the corresponding terms in the two sums of* (19) *always have different signs. Hence, one has* $W_{v\cdot F'}(\omega) = 0$ *for* $\omega_{\delta+1} = 0$, *and* $W_{v\cdot F} = 2 \cdot W_f(\omega)$ *for* $\omega_{\delta+1} = 1$.

Next, assume that $m > 1$ *and let* $F' : \mathbb{F}_2^{n+1} \to \mathbb{F}_2^{m+1}$ *be the global rule of the CA obtained by appending a cell to the right of* $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$. *Given a component function of* F *selected by* $v \in (\mathbb{F}_2^m)^*$, *one can construct two component functions of* F' *respectively as* $(v,0)\cdot F'$ *and* $(v,1)\cdot F'$.

In order to shorten the notation, let $x = (\tilde{x}, x_{n+1}) \in \mathbb{F}_2^{n+1}$ *and* $\omega = (\tilde{\omega}, \omega_{n+1}) \in \mathbb{F}_2^{n+1}$, *where* $\tilde{x} = (x_1,\cdots,x_n) \in \mathbb{F}_2^n$ *and* $\tilde{\omega} = (\omega_1,\cdots,\omega_n) \in \mathbb{F}_2^n$, *and* $x_{n+1}, \omega_{n+1} \in \mathbb{F}_2$. *Let us now consider all those component functions* $(v,0)\cdot F'$, *i.e. those that do not select the last coordinate function* $f(x_{m+1},\cdots,x_{n+1})$ *in the linear combination. Then, the Walsh coefficient over* ω *in this case equals:*

$$W_{(v,0)\cdot F'}(\omega) = \sum_{x\in\mathbb{F}_2^{n+1}} (-1)^{v\cdot F(\tilde{x})\oplus\omega\cdot x} = \\ = \sum_{x\in\mathbb{F}_2^{n+1}} (-1)^{v\cdot F(\tilde{x})\oplus\tilde{\omega}\cdot\tilde{x}\oplus\omega_{n+1}x_{n+1}} \quad . \tag{20}$$

By splitting the sum with respect to the value of x_{n+1} *Equation* (20) *can be rewritten as:*

$$W_{(v,0)\cdot F'}(\omega) = \sum_{\substack{x\in\mathbb{F}_2^{n+1}:\\ x_{n+1}=0}} (-1)^{v\cdot F(\tilde{x})\oplus\omega\cdot x} \\ + (-1)^{\omega_{n+1}} \cdot \sum_{\substack{x\in\mathbb{F}_2^{n+1}:\\ x_{n+1}=1}} (-1)^{v\cdot F(\tilde{x})\oplus\omega\cdot x} \quad . \tag{21}$$

Similarly to the base case $v = (1,0)$, *one can see that for all* $\omega = (\tilde{\omega}, 0)$ *the two sums in* (21) *have the same sign, and these sums both correspond to* $W_{v\cdot F}(\tilde{\omega})$. *Hence, one obtains that* $W_{(v,0)\cdot F'}(\tilde{\omega}, 0) = 2 \cdot W_{v\cdot F}(\tilde{\omega})$. *On the other hand, for all* $\omega = (\tilde{\omega}, 1)$ *the two sums have different signs, thus in this case it holds that* $W_{(v,0)\cdot F'}(\tilde{\omega}, 1) = 0$.

The last case we need to consider includes all those component functions of the form $(v,1)\cdot F'$, *where the last coordinate function appears in the linear combination. The Walsh coefficient over* ω *is:*

$$W_{(v,1)\cdot F'}(\omega) = \sum_{x\in\mathbb{F}_2^{n+1}} (-1)^{v\cdot F(\tilde{x})\oplus f(x_{m+1},\cdots,x_{n+1})\oplus\omega\cdot x} = \\ = \sum_{x\in\mathbb{F}_2^{n+1}} (-1)^{v\cdot F(\tilde{x})\oplus\tilde{\omega}\cdot\tilde{x}} \cdot (-1)^{f(x_{m+1},\cdots,x_{n+1})\oplus\omega_{n+1}\cdot x_{n+1}} \quad . \tag{22}$$

Again, let us split the sum of (22) *with respect to the value of* x_{n+1} *as follows:*

$$W_{(v,1)\cdot F'}(\omega) = \sum_{\substack{x\in\mathbb{F}_2^{n+1}:\\ x_{n+1}=0}} (-1)^{v\cdot F(\tilde{x})\oplus\omega\cdot x}\cdot(-1)^{f(x_{m+1},\cdots,0)} + (-1)^{\omega_{n+1}}\cdot\sum_{\substack{x\in\mathbb{F}_2^{n+1}:\\ x_{n+1}=1}} (-1)^{v\cdot F(\tilde{x})\oplus\omega\cdot x}\cdot(-1)^{f(x_{m+1},\cdots,1)} \quad . \tag{23}$$

Analogously to the case of $v=(1,1)$ *for* $m=2$ *discussed above, for each* $x\in\mathbb{F}_2^{n+1}$ *the terms in the two sums of Equation* (23) *always have different signs. Since the first part of the two sums coincides with the Walsh coefficient of* $\tilde{v}\cdot F$ *over* $\tilde{\omega}$*, it results that* $W_{(v,1)\cdot F'}(\omega)=0$ *for* $\omega_{n+1}=0$ *and* $W_{(v,1)\cdot F'}(\omega)=2\cdot W_{v\cdot F}(\tilde{\omega})$ *for* $\omega_{n+1}=1$. □

From Theorem 1, we can now determine the nonlinearity of the global rule of a permutive CA in terms of the nonlinearity of its local rule:

Corollary 1. *Let* $F:\mathbb{F}_2^n\to\mathbb{F}_2^m$ *a CA of length* $n=m+\delta-1$ *with left or right permutive local rule* $f:\mathbb{F}_2^\delta\to\mathbb{F}_2$*. Then, the nonlinearity of* F *equals*

$$Nl(F)=2^{m-1}\cdot Nl(f) \quad . \tag{24}$$

Proof. We proceed by induction on m. For $m=1$*, the global rule coincides with the local rule and Equation* (24) *is trivially true. Let us now consider the case* $m>1$ *and assume that the statement is true up to* $m-1$*. Then, by Theorem 1 we know that the Walsh coefficients of the component functions of* $F':\mathbb{F}_2^n\to\mathbb{F}_2^m$ *can only be zero or twice the coefficients of the corresponding components of* $F:\mathbb{F}_2^{n-1}\to\mathbb{F}_2^{m-1}$ *obtained by removing the last coordinate from the linear combination. This means that for each* $v=(\tilde{v},v_m)\in(\mathbb{F}_2^m)^*$ *the spectral radius of the component* $v\cdot F'$ *is twice the spectral radius of* $\tilde{v}\cdot F$*. Hence, the nonlinearity of* $v\cdot F'$ *is given by*

$$\begin{aligned} Nl(v\cdot F') &= 2^{n-1}-\frac{1}{2}max_{\omega\in\mathbb{F}_2^n}\{|W_{v\cdot F'}(\omega)|\}= \\ &= 2^{n-1}-\frac{1}{2}\cdot\left(2\cdot max_{\tilde{\omega}\in\mathbb{F}_2^{n-1}}\{|W_{\tilde{v}\cdot F}(\tilde{\omega})|\}\right)= \\ &= 2\cdot 2^{n-2}-max_{\tilde{\omega}\in\mathbb{F}_2^{n-1}}\{|W_{\tilde{v}\cdot F}(\tilde{\omega}|)\}=2\cdot Nl(\tilde{v}\cdot F) \end{aligned} \tag{25}$$

By induction hypothesis, we know that $Nl(F)=2^{m-2}\cdot Nl(f)$*, and this is the minimal nonlinearity among the component functions of* F*. Thus, by Equation* (25) *it means that the minimal nonlinearity among the components of* F' *is*

$$2\cdot Nl(F)=2\cdot 2^{m-2}\cdot Nl(f)=2^{m-1}\cdot Nl(f) \quad , \tag{26}$$

which is by definition the nonlinearity of F'. □

3.3 Resiliency of Bipermutive CA

We now show that bipermutive cellular automata are always at least 1-resilient when considered as vectorial boolean functions. To this end, we first recall a secondary

construction to obtain a $(t+1)$-resilient boolean function of $n+1$ variables from a t-resilient function of n variables, originally proved in [18]. This method is formalized in the following result:

Proposition 3. *Let $I = \{i_1, \cdots, i_{t+1}\} \subseteq \{1, \cdots, n\}$ and $J = \{j_1, \cdots, j_{n-t-1}\} = \{1, \cdots, n\} \setminus I$ be complementary sets of indices. Additionally, let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a boolean function of n variables defined as*

$$f(x_1, \cdots, x_n) = g(x_{j_1}, \cdots, x_{j_{n-t-1}}) \oplus x_{i_1} \oplus \cdots \oplus x_{i_{t+1}} \quad ,$$

where $g : \mathbb{F}_2^{n-t-1} \to \mathbb{F}_2$ is a boolean function of $n-t-1$ variables. Then, f is t-resilient.

Hence, XORing one variable with g makes the resulting function 0-resilient (or, equivalently, balanced), and then any new XORed variable increases the resiliency order by 1.

Clearly, by Proposition 3 any bipermutive local rule is also a 1-resilient boolean function. A different proof of this fact based on the zeros of the Walsh transform can be found in [9].

The following result characterizes the component functions of a bipermutive CA based on its associated generating function:

Lemma 1. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a cellular automaton of length $n = m + \delta - 1$ defined by a bipermutive rule $f : \mathbb{F}_2^\delta \to \mathbb{F}_2$. Then, for all $v \in (\mathbb{F}_2^m)^*$ the component function $v \cdot F$ is bipermutive as well.*

Proof. *Let $f(x_1, x_2, \cdots, x_{\delta-1}, x_\delta) = x_1 \oplus g(x_2, \cdots, x_{\delta-1}) \oplus x_\delta$ with $g : \mathbb{F}_2^{\delta-2} \to \mathbb{F}_2$. Given $v \in (\mathbb{F}_2^m)^*$, the component function $v \cdot F$ can be expressed as:*

$$\begin{aligned} v \cdot F = x_{i_1} &\oplus g(x_{i_1+1}, \cdots, x_{i_1+\delta-2}) \oplus x_{i_1+\delta-1} \oplus \\ &\oplus \cdots \oplus x_{i_k} \oplus g(x_{i_k+1}, \cdots, x_{i_k+\delta-2}) \oplus x_{i_k+\delta-1} \quad . \end{aligned} \tag{27}$$

Notice that the leftmost and rightmost variables x_{i_1} and $x_{i_k+\delta-1}$ appear exactly once in Equation (27), thus they are never canceled. Let G be the boolean function defined as:

$$\begin{aligned} G(x_{i_1+1}, \cdots, x_{i_k+\delta-2}) = g(x_{i_1+1}, \cdots, x_{i_1+\delta-2}) &\oplus x_{i_1+\delta-1} \oplus \\ \cdots \oplus x_{i_k} \oplus g(x_{i_k+1}, \cdots, x_{i_k+\delta-2}) \quad . \end{aligned} \tag{28}$$

Hence, the component function $v \cdot F$ has the form:

$$v \cdot F = x_{i_1} \oplus G(x_{i_1+1}, \cdots, x_{i_k+\delta-2}) \oplus x_{i_k+\delta-1} \quad , \tag{29}$$

and thus it is bipermutive. □

Combining Lemma 1 and Proposition 3, we get the following result:

Theorem 2. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a CA of length $n = m + \delta - 1$ defined by a bipermutive rule $f : \mathbb{F}_2^\delta \to \mathbb{F}_2$. Then, F is at least 1-resilient.*

4 Linear CA and Linear Codes

Besides the applications to the design of stream ciphers, the resiliency criterion has also relevance in coding theory, since it is related to the minimum distance of linear codes. Motivated by the result on the 1-resiliency of the global rules of bipermutive CA, in this section we investigate linear CA from the perspective of coding theory. We first recall some basic concepts about binary linear codes. We then show that linear CA are equivalent to cyclic linear codes, and observe that the minimum distance of the latter is related to the resiliency order of the former. To wrap up the discussion, we finally show how the encoding and decoding process in the cyclic Hamming code $(7,4,3)$ correspond respectively to preimage computation and forward iteration of a bipermutive linear CA of radius 2 with a 2-resilient global rule.

4.1 Basics on Linear Codes

We now briefly discuss the basic definitions and results related to linear and cyclic error-correcting codes. For a thorough treatment of the subject, the reader can refer to [13].

Definition 4. *Let $n,m,d \in \mathbb{N}$ such that $n \geq m$, and let $q = \rho^\alpha$ be the power of a prime number ρ. A (n,m,d)* linear code *C is a m-dimensional subspace of the vector space $\mathbb{F}_q^n$, such that the Hamming distance between any two vectors $c_1, c_2 \in C$ (called* codewords*) is at least d. The parameters n, m and d are respectively called the* length*, the* dimension *and the* minimum distance *of C.*

In what follows, we focus on the case of *binary linear codes*, where $q = 2$.

Since a (n,m,d) linear code C is a subspace of dimension m of $\mathbb{F}_2^n$, it is possible to specify it using a $m \times n$ matrix G whose rows form a set of m linearly independent codewords of C. Such a matrix G is called a *generator matrix* for code C. The encoding process simply amounts to multiplying a *message vector* $\mu \in \mathbb{F}_2^m$ by matrix G, thus obtaining the codeword $c = \mu G$. Another matrix associated to a linear code is its *parity check matrix*, which is useful for error correction. The parity check matrix for C is a matrix H of dimensions $(n-m) \times n$ such that $Hx^\top = \underline{0}$ if and only if $x \in C$. In general, the vector $s = Hx^\top$ is called the *syndrome* of $x \in \mathbb{F}_2^n$.

The *dual code* of a (n,m,d) linear code C is the set $C^\perp = \{x \in \mathbb{F}_2^n : \ x \cdot y = 0, \forall y \in C\}$, that is, the set of all vectors in $\mathbb{F}_2^n$ which are orthogonal to the codewords in C. The parity check matrix H of C is a generator matrix for $C^\perp$, and vice versa the generator matrix G of C is a parity check matrix for $C^\perp$. Thus, A (n,m,d) linear code $C \subseteq \mathbb{F}_2^n$ is called *cyclic* if it is closed under *cyclic shifts*, that is, $c' = (c_2, \cdots, c_n, c_1) \in C$ for all $c = (c_1, c_2 \cdots, c_n) \in C$. A cyclic code is described by its *generator polynomial*:

$$g(x) = g_0 + g_1 x + \cdots + g_{n-m} x^{n-m} \ , \tag{30}$$

where $g_i \in \mathbb{F}_2$ for all $i \in \{0, \cdots, n-m\}$. If one represents the m-bit message $\mu = (\mu_0, \cdots, \mu_{m-1})$ by the polynomial $\mu(x) = \mu_0 + \mu_1 x + \cdots + \mu_{m-1} x^{m-1}$, then the polynomial corresponding to the codeword c is $c(x) = \mu(x) g(x)$. There exists a one-to-one

correspondence between cyclic codes and divisors of x^n-1. In particular, a (n,m,d) code C is cyclic if and only if its generator polynomial $g(x)$ divides x^n-1.

Given a (n,m,d) cyclic code C with generator polynomial $g(x)$ of degree $n-m$, the polynomial $h(x)=(x^n-1)/g(x)$ of degree m is the *parity check polynomial* of C. Analogously to the parity check matrix, $h(x)$ satisfies the property that the codeword associated to a polynomial $d(x)$ belongs to C if and only if $d(x)h(x)=0$. The following result relates the generator/parity check polynomials of a cyclic code C to its generator/parity check matrices:

Theorem 3. *Let $C\subseteq\mathbb{F}_2^n$ be a (n,m,d) cyclic linear code with generator polynomial $g(x)=g_0+g_1x+\cdots+g_{n-m}x^{n-m}$ and parity check polynomial $h(x)=h_0+h_1x+\cdots+h_mx^m$. Then the following are respectively a generator and a parity check matrix for C:*

$$G=\begin{pmatrix} g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-m} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & g_0 & \cdots & g_{n-m} \end{pmatrix} \tag{31}$$

$$H=\begin{pmatrix} h_m & \cdots & h_0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & h_m & \cdots & h_0 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & h_m & \cdots & h_0 \end{pmatrix}. \tag{32}$$

As a consequence of Theorem 3, the dual code $C^\top$ of a cyclic code is again a cyclic code of length n and dimension $n-m$.

One of the main advantages of cyclic codes is that they can be easily implemented using *Linear Feedback Shift Registers* (LFSR). A LFSR of order k is a discrete device composed of k registers D_0, D_1, $\cdots$, D_{k-1}. At each step $n\in\mathbb{N}$, the elements s_n, s_{n+1}, $\cdots$, $s_{n+k-1}\in\mathbb{F}_2$ in the registers are shifted one place to the left, and D_{k-1} is updated with the linear combination $a_0\cdot s_n+\cdots+a_{k-1}\cdot s_{n+k-1}$ (See Figure 1). The *tap polynomial* of the LFSR is the polynomial over $\mathbb{F}_2$ of degree k defined by the coefficients $a_0,\cdots,a_{k-1}$ of the LFSR. As shown in [13, pp. 193–195], if the parity check polynomial $h(x)$ of a (n,m,d) cyclic code is such that $h_0\neq 0$, the codeword of a message $\mu\in\mathbb{F}_2^m$ can be generated by a LFSR of length m whose tap polynomial is the reciprocal $\tilde{h}(x)=h(1/x)=h_m+h_{m-1}x+\cdots+x^m$ of $h(x)$, i.e. the multiplicative inverse of $h(x)$ over the ring $\mathbb{F}_2[x]$. The registers are initialized to the values $\mu_0,\cdots,\mu_{m-1}$ of μ, and

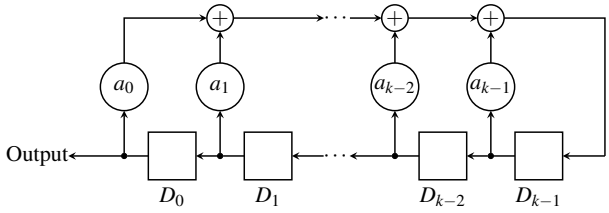


Fig. 1: Example of linear feedback shift register.

the LFSR is evolved for n steps. The output of length n produced by the LFSR is the codeword corresponding to μ. Notice that the first m output bits are exactly the original message μ, while the remaining $n-m$ are the parity check bits. This encoding procedure is called *systematic*, since the bits of the message appear unaltered in the corresponding codeword. If no errors are introduced by the channel, the decoding process is immediate since it just consists of truncating the codeword to its first m bits.

4.2 Linear CA and Cyclic Codes

A cellular automaton $F : \mathbb{F}_2^{m+\delta-1} \to \mathbb{F}_2^m$ is called *linear* if its local rule is defined as $f(x_1, \cdots x_\delta) = a_1 x_1 \oplus \cdots \oplus a_\delta x_\delta$, with $a_i \in \mathbb{F}_2$ for all $i \in \{1, \cdots, \delta\}$. The global rule of F is described by a $m \times (m+\delta-1)$ *transition matrix* M_F of the following form:

$$M_F = \begin{pmatrix} a_1 & \cdots & a_\delta & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & a_1 & \cdots & a_\delta & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_1 & \cdots & a_\delta \end{pmatrix} . \tag{33}$$

In particular, when the CA is bipermutive and linear we have $a_1 = a_\delta = 1$. The application of the CA global rule F to a configuration $x \in \mathbb{F}_2^{m+\delta-1}$ corresponds to the multiplication $y = M_F x^\top$.

One can notice that the generator and parity check matrices of Equation (31) and (32) in Theorem 3 have the same form of the linear CA matrix in Equation (33). In particular, the systematic encoding for cyclic codes described above can be simulated through cellular automata. As observed in [10], computing a preimage of a spatially periodic configuration in a linear bipermutive CA is equivalent to a *concatenation* of LFSR, where the LFSR associated to the local rule is disturbed by the LFSR which generates the spatially periodic configuration. In our case, we are only interested in a preimage of a finite configuration. Thus the general scheme consists of the LFSR associated to the rule where the feedback is additively disturbed by the bits of the configuration. If one takes the all-zeros configuration $\underline{0}$, it can be observed that the resulting concatenated LFSR of Figure 2 is equivalent to the LFSR used for the systematic encoding of a cyclic code. As a matter of fact, adding a sequence of zeros to the feedback of a LFSR does not change its dynamics. In the context of cellular automata, the system represented in Figure 2 is equivalent to the computation of a preimage of $\underline{0} \in \mathbb{F}_2^{n-m}$, in particular the preimage determined by the m-bit block μ.

To summarize the discussion above, we have thus proved the following result:

Theorem 4. *Let $F : \mathbb{F}_2^{m+\rho} \to \mathbb{F}_2^m$ be a linear cellular automaton defined by a local rule $f(x) = a_1 x_1 \oplus \cdots \oplus a_\delta x_\delta$ of diameter $\delta = \rho + 1$ with $\rho \in \mathbb{N}$, and let $g(x) = a_1 + a_2 x + \cdots + a_\delta x^\rho$ be the polynomial associated with f. If $g(x)$ divides $x^n - 1$ where $n = m + \rho$, then F is equivalent to a cyclic code C of length n and dimension m. The generator matrix of C is the CA matrix M_F associated to F, while $g(x)$ is the generator polynomial of C. Additionally, let $h(x)$ be the reciprocal of the parity check polynomial $h(x) = (x^n - 1)/g(x)$, defined as $\tilde{h}(x) = h_m + h_{m-1}x + \cdots + h_0 x^m$ and let $\tilde{f}(x) = h_m x_1 \oplus \cdots \oplus h_0 x_{m+1}$ be the corresponding local rule. Then, the matrix $M_{\tilde{F}}$ associated to the linear CA $\tilde{F} : \mathbb{F}_2^{m+\rho} \to \mathbb{F}_2^\rho$ induced by rule $\tilde{f}$ is a parity check matrix for C, and $C = \tilde{F}^{-1}(\underline{0})$.*

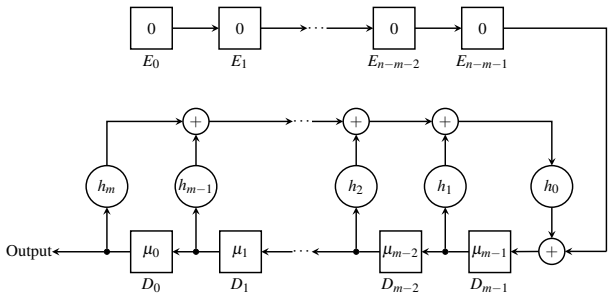


Fig. 2: Concatenation of a LFSR with a sequence of $n-m$ zeros, which computes a preimage $x \in F^{-1}(\underline{0})$. Each element μ_i in the registers correspond to a symbol of the message μ.

In other words, by Theorem 4 we can employ a linear CA in the encoding and decoding process of a linear cyclic code of length n and dimension m as follows:

1. Given m and $n = m+\rho$ with $\rho \in \mathbb{N}$, determine a local rule f of diameter $\delta = \rho + 1$ such that the associated polynomial $g(x)$ divides $x^n - 1$.
2. Compute the reciprocal $\tilde{h}(x)$ of the parity check polynomial $h(x) = (x^n - 1)/g(x)$, and determine the corresponding local rule $\tilde{f}$ of diameter $m+1$.
3. *Systematic encoding*: Let $\tilde{F} : \mathbb{F}_2^{m+\rho} \to \mathbb{F}_2^{\rho}$ be the linear CA of length n induced by $\tilde{f}$. A message $\mu \in \mathbb{F}_2^m$ is encoded by computing the preimage $x \in \tilde{F}^{-1}(\underline{0})$ whose leftmost m-bit block equals μ. This preimage can be computed by the LFSR in Figure 2.
4. *Syndrome computation*: given $x \in \mathbb{F}_2^{m+\rho}$, the syndrome of x is $s = \tilde{F}(x)$. If the syndrome s equals $\underline{0} \in \mathbb{F}_2^{\rho}$ then x is a codeword of C. Otherwise, one can apply the syndrome decoding procedure to retrieve the original codeword.

Notice that up to now we did not consider the minimum distance of the cyclic codes generated through linear CA, which is necessary in order to assess their error-correction capability. This is where the resiliency order of the CA comes into play. In particular, the connection between general linear resilient functions and linear codes is given by the following theorem reported in [20]:

Theorem 5. *A $(d-1)$-resilient linear function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ is equivalent to a (n,m,d) linear code C.*

We already know from the previous section that all bipermutive CA are always at least 1-resilient, thus a linear and bipermutive CA which satisfies the hypotheses of Theorem 4 is equivalent to a linear cyclic code with minimum distance at least 2. More in general, we can refine Theorem 4 by using Theorem 5 as follows:

Theorem 6. *Let $F : \mathbb{F}_2^{m+\rho} \to \mathbb{F}_2^m$ be a linear CA satisfying the hypotheses of Theorem 4. If F is $(d-1)$–resilient, then the cyclic code associated to F has minimum distance d.*

4.3 Cyclic Hamming Codes through Linear CA

To sum up the results presented in the previous section, we show an example of cyclic code generated by a linear CA. In particular we focus on *cyclic Hamming codes*, which are codes with minimum distance $d = 3$ and thus they can correct up to 1 error. The main reason for this choice is the simplicity of syndrome decoding in Hamming codes. As a matter of fact, the position of the column of the parity check matrix H containing the value of the syndrome is the position where the error occurred.

Example 1. Let $F : \mathbb{F}_2^7 \to \mathbb{F}_2^4$ be the linear CA induced by the local rule $f : \mathbb{F}_2^4 \to \mathbb{F}_2$ defined as $f(x) = x_1 \oplus x_2 \oplus x_4$. The associated polynomial is $g(x) = 1 + x + x^3$, while the CA matrix is:

$$M_F = \begin{pmatrix} 1\,1\,0\,1\,0\,0\,0 \\ 0\,1\,1\,0\,1\,0\,0 \\ 0\,0\,1\,1\,0\,1\,0 \\ 0\,0\,0\,1\,1\,0\,1 \end{pmatrix} \quad . \tag{34}$$

The polynomial $g(x)$ divides $x^7 - 1$, and it results that $h(x) = (x^7 - 1)/g(x) = 1 + x + x^2 + x^4$. Further, we can deduce from matrix M_F that F is 2-resilient. As a matter of fact, it is not difficult to see by exhaustive enumeration that each nonzero vector v results in a sum of rows which always have at least 3 ones. Hence, by Theorem 6 the code C associated to F is the $(7,4,3)$ cyclic Hamming code. Remark that the reciprocal of the parity check polynomial $h(x)$ is $\tilde{h}(x) = 1 + x^2 + x^3 + x^4$. The local rule $\tilde{f}$ associated to the polynomial $\tilde{h}(x)$ is $\tilde{f}(x) = x_1 \oplus x_3 \oplus x_4 \oplus x_5$, and thus it has radius $r = 2$. In particular, the Wolfram code representing the truth table of $\tilde{f}$ is 1768527510. The transition matrix of the linear CA $\tilde{F} : \mathbb{F}_2^7 \to \mathbb{F}_2^3$ induced by rule $\tilde{f}$ is the following:

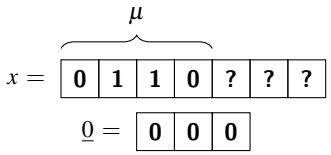
$$M_{\tilde{F}} = \begin{pmatrix} 1\,0\,1\,1\,1\,0\,0 \\ 0\,1\,0\,1\,1\,1\,0 \\ 0\,0\,1\,0\,1\,1\,1 \end{pmatrix} \quad . \tag{35}$$

Let $\mu = (0,1,1,0) \in \mathbb{F}_2^4$ be a 4-bit message. The systematic encoding of μ under the Hamming code $(7,4,3)$ can be accomplished by computing the preimage x of $(0,0,0)$ under the action of $\tilde{F}$, with the leftmost 4 bits of x initialized to μ. This process is depicted in Figure 3. Hence, the codeword corresponding to μ is $x = (0,1,1,0,1,0,0)$.

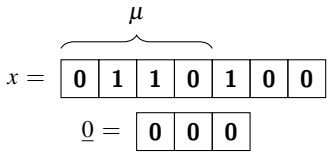
Let us now assume that x is transmitted through a noisy channel and the fourth bit of x is flipped, thus yielding the word $\tilde{x} = (0,1,1,1,1,0,0)$. The receiver applies to $\tilde{x}$ the CA $\tilde{F}$ defined by rule 1768527510, thus obtaining the syndrome $s = F(x) = (1,1,0)$, as shown in Figure 4(a). To correct the error, the receiver looks at the CA matrix $M_{\tilde{F}}$ and finds that the syndrome appears in the fourth column. Thus, the receiver knows that a transmission error has occurred in the fourth position of $\tilde{x}$, and the original codeword can be recovered as $\tilde{x} \oplus (0,0,0,1,0,0,0) = x$.

5 Conclusions and Future Directions

In this work, we began investigating the cryptographic properties of the global rules of CA with no boundary conditions, focusing on their algebraic degree, nonlinearity

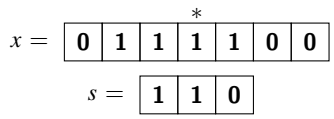


(a) Initialization



(b) Complete codeword

Fig. 3: Systematic encoding of $\mu = (0,1,1,0) \in \mathbb{F}_2^4$ using rule 1768527510, defined as $\tilde{f}(x) = x_1 \oplus x_3 \oplus x_4 \oplus x_5$.



(a) Syndrome computation

$$\begin{pmatrix} 1\,0\,1\,1\,1\,0\,0 \\ 0\,1\,0\,1\,1\,1\,0 \\ 0\,0\,1\,0\,1\,1\,1 \end{pmatrix}$$
$$\uparrow$$

(b) Error correction

Fig. 4: Example of error correction using rule 1768527510. The cell marked by $*$ indicates where the error occurred.

and resiliency. As a first result, we proved that the algebraic degree of a CA global rule coincides with the degree of its local rule. Subsequently, by restricting our analysis to the class of CA with permutive local rules, we investigated how the addition of a new cell to the CA affects the Walsh spectrum of its component functions. This allowed us to determine the nonlinearity of permutive CA in terms of the nonlinearity of their local rules. Then, we proved that the global rule of a bipermutive CA F is always at least 1-resilient, since each component of F is still a bipermutive boolean function. Since the resiliency criterion is also related to the error correction capability of linear codes, we analyzed CA from the point of view of coding theory, proving an equivalence between linear cyclic codes and linear CA. In particular, we observed that the syndrome computation process in the former is equivalent to applying the global rule to the received word in the latter. Finally, the resiliency order of a linear and bipermutive CA can be used to determine the minimum distance of the corresponding cyclic code, and we applied these results by showing how the encoding and decoding process of the $(7,4,3)$ cyclic Hamming code can be realized using a 2-resilient linear CA of radius $r = 2$.

There are several directions along which the research discussed in this paper can be extended. Concerning the cryptographic properties of the global rules, an interesting direction to develop is the study of the *differential uniformity* in CA, a criterion related

to the resistance of S-boxes to differential cryptanalysis (see [15]). Additionally, another direction to consider is the generalization of the results presented in this paper to the case of CA with periodic boundary conditions. As a matter of fact, periodic CA whose length coincides with the diameter of the local rule are known in the cryptographic literature under the name of *rotation-symmetric S-boxes* (see [16]). An interesting question to investigate in this regard would be to show lower and upper bounds on the nonlinearity of global rules with respect to the length and the diameter of the CA. The trade-off to consider in this case is the minimization of the diameter of the CA while retaining a good nonlinearity on the resulting S-boxes, in order to obtain strong S-boxes which can be efficiently implemented in hardware, like rule χ in the case of KECCAK.

About the coding-theoretic part of our work, cyclic codes form a broad class including for example *BCH* and *Reed-Solomon* codes. Hence, it could be interesting to investigate how to implement these codes through CA by elaborating on the method presented in this paper. As we mentioned in the Introduction, *MDS codes* are also employed to design the *diffusion layers* of block ciphers, such as for example the MIXCOLUMNS operation of Rijndael, the encryption algorithm which constitutes the AES standard (see [5]). Thus, another direction of research worth exploring is to consider the design of MDS codes by means of linear CA for lightweight implementations of diffusion linear layers.

References

1. G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche. Keccak. In *EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 313–314, 2013.
2. C. Carlet. Boolean functions for cryptography and error correcting codes. *Boolean models and methods in mathematics, computer science, and engineering*, 2:257–397, 2010.
3. C. Carlet. Vectorial boolean functions for cryptography. *Boolean models and methods in mathematics, computer science, and engineering*, 134:398–469, 2010.
4. J. Daemen, R. Govaerts, and J. Vandewalle. An efficient nonlinear shift-invariant transformation. In *Proceedings of the 15th Symposium on Information Theory in the Benelux, B. Macq, Ed., Werkgemeenschap voor Informatie-en Communicatietheorie*, pages 108–115. Citeseer, 1994.
5. J. Daemen and V. Rijmen. *The Design of Rijndael*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.
6. E. Formenti, K. Imai, B. Martin, and J. Yunès. 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*, pages 56–70, 2014.
7. J. Kari. Basic concepts of cellular automata. In *Handbook of Natural Computing*, pages 3–24. 2012.
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. Cryptographic properties of bipermutive cellular automata rules. *J. Cellular Automata*, 9(5-6):437–475, 2014.
10. L. Mariot and A. Leporati. On the periods of spatially periodic preimages in linear bipermutive cellular automata. In *Cellular Automata and Discrete Complex Systems - 21st IFIP WG 1.5 International Workshop, AUTOMATA 2015, Turku, Finland, June 8-10, 2015. Proceedings*, pages 181–195, 2015.

11. L. Mariot and A. Leporati. Resilient vectorial functions and cyclic codes arising from cellular automata. In *Cellular Automata - 12th International Conference on Cellular Automata for Research and Industry, ACRI 2016, Fez, Morocco, September 5-8, 2016. Proceedings*, pages 34–44, 2016.
12. B. Martin. A walsh exploration of elementary CA rules. *J. Cellular Automata*, 3(2):145–156, 2008.
13. R. McEliece. *The theory of information and coding*. Cambridge University Press, 2002.
14. W. Meier and O. Staffelbach. Analysis of pseudo random sequence generated by cellular automata. In *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, pages 186–199, 1991.
15. K. Nyberg. S-boxes and round functions with controllable linearity and differential uniformity. In *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, pages 111–130, 1994.
16. V. Rijmen, P. S. L. M. Barreto, and D. L. G. Filho. Rotation symmetry in algebraically generated cryptographic substitution tables. *Inf. Process. Lett.*, 106(6):246–250, 2008.
17. C. E. Shannon. Communication theory of secrecy systems. *Bell Labs Technical Journal*, 28(4):656–715, 1949.
18. T. Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. *IEEE Trans. Computers*, 34(1):81–85, 1985.
19. D. R. Stinson. *Cryptography - theory and practice*. Discrete mathematics and its applications series. CRC Press, 1995.
20. D. R. Stinson. *Combinatorial designs - constructions and analysis*. Springer, 2004.
21. S. Ulam. Random processes and transformations. In *Proceedings of the International Congress on Mathematics*, volume 2, pages 264–275, 1952.
22. J. Von Neumann. *Theory of self-reproducing automata*. Edited by Burks, Arthur W. University of Illinois Press, 1966.
23. S. Wolfram. Statistical mechanics of cellular automata. *Reviews of modern physics*, 55(3):601, 1983.
24. S. Wolfram. Cryptography with cellular automata. In *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*, pages 429–432, 1985.