Cryptographic Properties of Bipermutive Cellular Automata Rules

ALBERTO LEPORATI*, LUCA MARIOT

Dipartimento di Informatica, Sistemistica e Comunicazione,
Università degli Studi Milano - Bicocca,
Viale Sarca 336/14, 20124 Milano, Italy

Bipermutive rules are known to induce both expansive and mixing chaotic cellular automata. In this paper, we study some cryptographic properties of bipermutive rules, initially proving that they also satisfy 1-resiliency, which combines balancedness and first order correlation immunity. We thus carry out an exhaustive exploration of the 256 bipermutive rules of radius 2, in order to select those rules satisfying additional cryptographic criteria (2-resiliency and high nonlinearity), and we test them through the ENT and NIST statistical test suites. We then complete the theoretical analysis of bipermutive rules by showing how several other properties (algebraic degree, nonlinearity, *k*-resiliency, number of linear structures) can be deduced by the properties of their *generating functions*. Finally, we explore the set of bipermutive rules having radius 3, always selecting the ones which satisfy the best trade-offs among the considered properties, and we test them as well with the ENT and NIST suites.

Key words: Cellular automata, boolean functions, pseudorandom number generators, stream ciphers, block ciphers, permutivity, resiliency, nonlinearity, strict avalanche criterion, Walsh transform, ENT test suite, NIST test suite.

* email: `alberto.leporati@unimib.it`

1 INTRODUCTION

Cellular automata (CA) have widely been used in the past to define pseudorandom number generators (PRNG) for the design of stream ciphers. Starting with Wolfram [30], particular interest has been devoted to the study of CA rules of radius 1. Wolfram proposed to use a CA equipped with rule 30 and to sample the trace of its central cell as a pseudorandom sequence, to be subsequently used as a keystream for a Vernam-like stream cipher. Unfortunately, even if rule 30 is nonlinear and balanced, and even if it is chaotic with respect to Devaney's definition of topological chaos [6], it does not satisfy the property of *first order correlation immunity*, introduced by Siegenthaler in [22]. More generally, Martin has pointed out in [14] that all nonlinear and balanced rules of radius 1 are not first order correlation immune. As a consequence, a CA-based PRNG using these rules may pass classic statistical randomness tests, but it is susceptible to correlation attacks.

Cattaneo, Finelli and Margara showed in [3] that *bipermutive* rules (that is, rules which are both leftmost and rightmost permutive) are *expansively chaotic*, while in [4] it has been proved that rules which are either leftmost or rightmost permutive are *mixing chaotic*. Thus, bipermutive rules satisfy stronger definitions of topological chaos than the one given by Devaney.

The aim of this paper is to analyse the cryptographic properties of bipermutive rules, in order to investigate their possible application to CA-based PRNG for stream ciphers, such as Wolfram's generator. More precisely, we focus our attention on the properties of nonlinearity, resiliency (a combination of balancedness and correlation immunity) and algebraic degree, which are related to specific cryptanalytic attacks on the boolean functions involved in stream ciphers. We also consider two properties usually studied in the context of block ciphers, namely the strict avalanche criterion (SAC) and the number of linear structures. As a matter of fact, CA-based block ciphers have also been proposed in the literature (see for example Gutowitz [8]), thus it would be useful to find local rules which could be applied to the design of both stream ciphers and block ciphers based on CA.

We begin our analysis by proving that bipermutive rules are 1-resilient, and we derive a graph-based encoding to enumerate all bipermutive rules of a given radius r. We then apply this encoding to generate all 256 bipermutive rules of radius 2, and compute their Walsh transforms to select only those which are nonlinear and 2-resilient. We successively filter out the remaining rules which do not generate sequences of 2^{16} bits (under Wolfram's generation method) that pass the statistical tests from the ENT suite [27], using rule

30 as a benchmark. Next, we apply the more stringent NIST test suite [18] to longer sequences (10^6 bits) produced by the remaining rules, observing that three of them pass all the tests, like rule 30.

Building upon the observations made for the case of radius 2, we generalise our theoretical analysis by showing how the cryptographic properties besides 1-resiliency of bipermutive rules can be deduced by studying their *generating functions*, which are obtained either using the Shannon decomposition of boolean functions or the graph-based encoding. Specifically, we show how the algebraic degree, the nonlinearity and the order of resiliency of a bipermutive rule can be determined using its generating function. We also prove that bipermutive rules never satisfy the SAC, and we derive a formula to compute the number of linear structures in a bipermutive rule. We finally perform a combinatorial exploration on the set of bipermutive rules of radius 3, using the results we proved to characterise three subclasses of rules, each satisfying a particular combination of cryptographic properties. These three subclasses are finally subjected to the ENT and NIST suites using the same procedure adopted for the case of radius 2, and it is found that two of them contain rules passing all the tests.

The rest of this paper is organised as follows. Section 2 recalls basic definitions and theoretical results about cellular automata and topological chaos. Section 3 discusses the cryptographic properties of boolean functions and the mathematical transforms used to compute them. In Section 4, after a brief introduction to the basic definitions pertaining to permutive rules and their chaotic behaviour, it is proved that bipermutive rules are also 1-resilient. Section 5 describes an enumerative encoding for bipermutive rules based on a graph representation, and the application of this encoding to the generation of bipermutive rules of radius 2, in order to recover only those which are nonlinear and 2-resilient. The results of the statistical tests of the ENT and NIST suites on the pseudorandom sequences generated by these rules through Wolfram's PRNG are also reported. Section 6 completes the theoretical analysis started in Section 4 by showing how the considered cryptographic properties of bipermutive rules are related to those of their generating functions. An exhaustive exploration on the set of bipermutive rules of radius 3 is also presented, along with the results of the ENT and NIST suites obtained by the best rules. Finally, Section 7 sums up the results discussed in the paper, and describes some possible future lines of research on the subject.

This paper is an extended version of [12].

2 CELLULAR AUTOMATA

2.1 Finite Cellular Automata

Cellular automata are a particular type of discrete dynamical systems, originally introduced by Ulam [25] and von Neumann [26] as a mathematical abstraction for self-reproduction phenomena. A CA is characterised by a regular lattice of *cells*. At each discrete time step, all the cells synchronously update their states by applying a *local rule*. Formally, we give the following definition of *finite one-dimensional cellular automaton*, which is the typical model of CA used in cryptographic applications.

Definition 2.1. A *finite one-dimensional cellular automaton* is a 4-tuple

$$CA = \langle n, A, r, f \rangle$$

where $n \in \mathbb{N}$ is the number of cells, A is the set of local states, $r \in \mathbb{N}$ is the radius and $f : A^{2r+1} \to A$ is the local rule.

Thus, essentially, a finite one-dimensional CA is composed of an array of n cells. In what follows, we assume $A = \mathbb{F}_2$: the CA, in this case, is called *boolean*. For all $i \in \{1, ..., n\}$ and $t \in \mathbb{N}$, we denote by c_i^t the state of the i-th cell at time t, and the next state is computed as $c_i^{t+1} = f(c_{i-r}^t, ..., c_i^t, ..., c_{i+r}^t)$. The *configuration* of the CA at time t is the binary vector $c^t = (c_1^t, ..., c_n^t)$. To update the cells at the boundaries, two approaches are possible: *null boundary conditions*, where r cells with constant states are added before the first cell and after the last one, and *periodic boundary conditions*, in which the array can be viewed as a ring, so that the last cell precedes the first one. For all radii $r \in \mathbb{N}$, each of the $2^{2^{2r+1}}$ local rules can be indexed by its *Wolfram code*, introduced in [29], which is basically the decimal representation of the binary string that encodes the truth table of the rule.

Wolfram extensively studied the 256 *elementary* rules (that is, rules of radius $r = 1$), and in [30] he proposed to use a CA with rule 30 as a pseudorandom number generator for cryptographic purposes, since it exhibits a chaotic behaviour when observing the sequence of configurations $\{c^t\}_{t \in \mathbb{N}}$. The CA is initialised with a random configuration c^0 (the seed), and at each time step the state of the $\lceil \frac{n}{2} \rceil$-th cell is taken as a new pseudorandom bit. Wolfram analysed this PRNG by applying several statistical tests, which suggested it could generate good pseudorandom sequences to be used in a Vernam-like stream cipher. In this case, a short secret key is used as a seed for the PRNG, and the resulting pseudorandom sequence (called the keystream) is bitwise XORed with the plaintext to obtain the ciphertext.

To a lesser extent, CA have also been used to design block ciphers, where the plaintext is processed in fixed-length chunks of several bits at once. The first attempt dates back to Gutowitz [8], who proposed to use CA for the diffusion and substitution phases in a multiple-round block cipher. In particular, in the diffusion phase he used local rules which give rise to *irreversible* CA, where every configuration has several possible preimages, while for the substitution phase he adopted local rules inducing *reversible* CA, where every configuration has a unique predecessor, to implement a *substitution box*, or *S-box*. Gutowitz performed some experiments to assess the security of his block cipher, which seemed to indicate that it could resist to differential attacks.

2.2 Infinite CA and Topological Chaos

The dynamics of one-dimensional CA is generally studied on the space of *bi-infinite sequences* $A^{\mathbb{Z}} = \{c : \mathbb{Z} \to A\}$, since every finite CA is trivially periodic in time. In this case, a configuration c is a function which assigns to each integer number a symbol from the alphabet A. The set $A^{\mathbb{Z}}$ is usually endowed with the *Tychonoff distance*, which in the boolean case $A = \mathbb{F}_2$ is defined $\forall x, y \in A^{\mathbb{Z}}$ as

$$d(x,y) = \sum_{i=-\infty}^{+\infty} \frac{1}{2^{|i|}} |x(i) - y(i)| \quad . \tag{1}$$

Under this distance, $A^{\mathbb{Z}}$ is a compact and perfect (i.e., without isolated points) metric space. Moreover, any global rule $F : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ induced by a CA local rule is a uniformly continuous function with respect to the Tychonoff distance. Thus a one-dimensional CA, now denoted by a triple $\langle A, r, f \rangle$, can be considered as a discrete time dynamical system (DTDS) $\langle X, F \rangle$, where the phase space is $X = A^{\mathbb{Z}}$ and the update function is the *global rule* $F : A^{\mathbb{Z}} \to A^{\mathbb{Z}}$ which applies at each time step the local rule f to all the cells $i \in \mathbb{Z}$.

The notion of *deterministic chaos* has been formalised in several rigorous definitions in the literature of dynamical systems. The most popular among them is perhaps the definition given by Devaney in [6], which uses a topological approach.

Definition 2.2. A DTDS $\langle X, F \rangle$ is *Devaney-chaotic* (*D-chaotic*) if it satisfies the following conditions:

1. *Topological transitivity:* for all nonempty open subsets $U, V \subset X$, there exists a $t \in \mathbb{N}$ such that $F^t(U) \cap V \neq \emptyset$.

2. *Topological regularity:* The set $Per(F) = \{x \in X : \exists p \in \mathbb{N} : F^p(x) = x\}$ of temporally periodic points is dense in X.
3. *Sensitivity to initial conditions:* there exists an $\varepsilon > 0$ such that $\forall x \in X$, $\forall \delta > 0$, $\exists y \in X$ and $\exists t \in \mathbb{N}$ such that $d(x,y) < \delta$ and $d(F^t(x), F^t(y)) \geq \varepsilon$.

Other definitions of chaos have been introduced by substituting stronger conditions to the three proposed by Devaney. In particular, the definition of *expansive chaos* (E-chaos) in a perfect DTDS $\langle X, F \rangle$ reported in [3] substitutes sensitivity to initial conditions with *positive expansivity*: there exists an $\varepsilon > 0$ such that, $\forall x, y \in X$, $x \neq y$, $\exists t \in \mathbb{N}$ such that $d(F^t(x), F^t(y)) \geq \varepsilon$. In *mixing chaos* (M-chaos) [4] topological transitivity is replaced by *topological mixing*: for all nonempty open subsets $U, V \subset X$, $\exists t \in \mathbb{N}$ such that $\forall s \geq t$, $F^s(U) \cap V \neq \emptyset$.

3 BOOLEAN FUNCTIONS

Boolean functions are fundamental in cryptography, in the design of both stream ciphers and block ciphers. Here we summarise the essential definitions and properties of the theory of cryptographic boolean functions applied in the rest of the paper to the local rules of CA. Where not otherwise specified, the proofs of all the theorems and propositions reported in this section can be found in [1] and [13].

3.1 Basic Definitions and Transforms of Boolean Functions

A *boolean function* in m variables is a mapping from the set $\mathbb{F}_2^m$ of binary m-tuples to $\mathbb{F}_2$. Given $f : \mathbb{F}_2^m \to \mathbb{F}_2$ and a particular ordering of the input vectors $(x_1, \cdots, x_m) \in \mathbb{F}_2^m$, the *truth table* of f is the 2^m-bit string encoding the output values of f. In what follows, we will assume that the input vectors are ordered lexicographically, with the least significant bit on the left. We denote by $\mathcal{F}_m$ the set of 2^{2^m} boolean functions in m variables.

The *algebraic normal form* (ANF) represents a boolean function f as a sum of products over $\mathbb{F}_2$. Specifically, given $f : \mathbb{F}_2^m \to \mathbb{F}_2$, $M = \{1, \cdots, m\}$ and $\mathcal{P}(M)$ the power set of M, the ANF of f is defined by the following polynomial:

$$f(x) = \bigoplus_{I \in \mathcal{P}(M)} a_I \left(\prod_{i \in I} x_i \right) \quad . \tag{2}$$

This representation is unique, since the mapping which associates a boolean function to its algebraic normal form is a bijection from $\mathcal{F}_m$ to the quotient ring $\mathbb{F}_2[x_1,\cdots,x_m]/(x_1^2 \oplus x_1,\cdots,x_m^2 \oplus x_m)$. The *algebraic degree* of a boolean function f is the cardinality of the largest subset $I \in \mathcal{P}(M)$ in the ANF of f such that $a_I \neq 0$. Boolean functions having degree $d = 1$ are called *affine* or *linear* functions.

Given ω and x vectors of $\mathbb{F}_2^m$, by $\omega \cdot x$ we denote the *scalar product* between ω and x, computed as $\omega \cdot x = \bigoplus_{i=1}^m \omega_i \cdot x_i$. The polar value of $f(x)$ is defined as $\hat{f}(x) = (-1)^{f(x)}$. The *Hamming weight* of a vector $x \in \mathbb{F}_2^m$, denoted by $w_H(x)$, is the number of nonzero coordinates in x. A boolean function $f: \mathbb{F}_2^m \to \mathbb{F}_2$ is *balanced* if $|f^{-1}(0)| = |f^{-1}(1)| = 2^{m-1}$.

We now recall the definition of the *Walsh Transform*, an essential tool used to characterise cryptographic properties of boolean functions.

Definition 3.1. The *Walsh Transform* of a boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is a function $\hat{F} : \mathbb{F}_2^m \to \mathbb{R}$ defined as follows: $\forall \omega \in \mathbb{F}_2^m$

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^m} \hat{f}(x) \cdot (-1)^{\omega \cdot x} \quad . \tag{3}$$

The value $\hat{F}(\omega)$ is also called the *Walsh coefficient* of f with respect to the vector ω. Given a boolean function f, the maximum absolute value of its Walsh coefficients, $W_{\max}(f)$, is called the *spectral radius* of f. A naive algorithm to compute the Walsh Transform of a boolean function having a truth table of $n = 2^m$ bits requires $O(n^2)$ operations. There is, however, a *Fast Walsh Transform* (FWT) algorithm, described in [1], which requires only $O(n \log_2 n)$ operations.

We describe some properties of the Walsh Transform which will be used extensively to prove the theoretical results of this paper:

Property 3.2. Denoting by 0 the null vector of $\mathbb{F}_2^m$, it follows that its Walsh coefficient is $\hat{F}(0) = \sum_{x \in \mathbb{F}_2^m} \hat{f}(x)$.

Property 3.3. From Property 3.2, it is obvious that a function f is balanced if and only if $\hat{F}(0) = 0$.

Property 3.4. If $w_H(\omega) = 1$, then $\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^m} \hat{f}(x) \cdot (-1)^{x_i}$, where i is the index of the nonzero coordinate of ω.

Another fundamental property is *Parseval's relation*, which says that the sum of the squared Walsh spectrum is constant for all boolean functions defined on m variables.

Theorem 3.5 (Parseval's Relation)**.** *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a boolean function. Then,*

$$\sum_{\omega \in \mathbb{F}_2^m} \hat{F}^2(\omega) = 2^{2m} \quad . \tag{4}$$

A second transform which is broadly used to study the cryptographic properties of boolean functions is the *autocorrelation function*, defined as follows.

Definition 3.6. Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a boolean function. The *autocorrelation function* of f is the function $\hat{r} : \mathbb{F}_2^m \to \mathbb{R}$ defined as follows:

$$\hat{r}(s) = \sum_{x \in \mathbb{F}_2^m} \hat{f}(x) \cdot \hat{f}(x \oplus s) \quad . \tag{5}$$

The following theorem relates the autocorrelation function with the squared Walsh spectrum of a boolean function.

Theorem 3.7 (Wiener-Khintchine Theorem)**.** *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a boolean function. The following equality holds for all $\omega \in \mathbb{F}_2^m$:*

$$\hat{F}^2(\omega) = \sum_{s \in \mathbb{F}_2^m} \hat{r}(s) \cdot (-1)^{\omega \cdot s} \quad . \tag{6}$$

The practical consequence of the Wiener-Khintchine theorem is that one can efficiently compute the autocorrelation function by using the FWT algorithm.

3.2 Cryptographic Properties of Boolean Functions

Several properties and criteria have been defined in the cryptographic literature which the boolean functions used in symmetric ciphers should satisfy in order to resist to specific attacks (for a detailed survey, see for example [1]). In particular, some of these properties refer to the cryptanalysis of certain types of PRNG and stream ciphers, such as the *combiner model* where the outputs of m Linear Feedback Shift Registers (LFSR) are combined by an m-variable boolean function. On the other hand, other properties pertain more to attacks on the S-Boxes used in block ciphers.

We provide here a brief overview of the most significant properties which will be applied in the rest of the paper to analyse the local rules of cellular automata.

Balancedness

Balanced boolean functions have already been defined in section 3.1 as those functions having the counterimages of 0 and 1 with the same cardinality. This means that the truth table representation of such functions is a string composed of an equal number of 0 and 1. Balancedness is a fundamental criterion, that every boolean function used for cryptographic applications should satisfy. In fact, unbalanced functions present a statistical bias which can be exploited for linear and differential cryptanalysis.

Algebraic Degree
Stream ciphers and cryptographic PRNG based on the combiner model which use boolean functions having low algebraic degree can be attacked using the *Berlekamp-Massey algorithm* [16]. In [20] it is shown that as the degree increases, this algorithm becomes computationally infeasible. As a consequence, the algebraic degree of the boolean functions involved the design of stream ciphers and cryptographic PRNG should be as high as possible.

Nonlinearity
Considering the truth table representation, cryptographic boolean functions should have a high Hamming distance from all affine functions. Formally, the *nonlinearity* of f is defined as follows.

Definition 3.8. Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a boolean function. Denoting by W_{max} the *spectral radius* of f, the *nonlinearity* of f is defined as

$$Nl(f) = 2^{m-1} - \frac{1}{2}W_{max} \quad .$$

When used in stream ciphers based on the combiner model, boolean functions having low nonlinearity may expose to *fast-correlation attacks* (for details, see for example [5]). For this reason, the nonlinearity should be as high as possible, to provide better confusion.

Correlation Immunity and Resiliency
Another essential cryptographic criterion for boolean functions, introduced by Siegenthaler in [22], is *correlation immunity*, defined below.

Definition 3.9. A boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is *k-th order correlation immune* (with $1 \leq k \leq m$) if the restrictions of f obtained by fixing at most k input coordinates all have the same Hamming weight.

A function which is both balanced and k-th order correlation immune is also called *k-resilient*. If a boolean function f is not k-resilient, then there is a correlation between at most k input coordinates of f and its output, which can be exploited (if k is sufficiently small) to recover the initialisations of k LFSR in the combiner model, as shown by Siegenthaler in [23].

A Walsh characterisation of correlation immunity has been proved by Xiao and Massey in [31].

Theorem 3.10. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a boolean function. Then, f is k-th order correlation immune if and only if $\hat{F}(\omega) = 0$ for all $\omega \in \mathbb{F}_2^m$ having weight $1 \leq w_{\mathrm{H}}(\omega) \leq k$.*

Hence, by Property 3.3 and Theorem 3.10, in order to check whether a given boolean function is k-resilient it is sufficient to verify that its Walsh transform vanishes for all vectors ω having Hamming weight at most k.

The three properties of resiliency, algebraic degree and nonlinearity induce two trade-offs; in particular, Siegenthaler [22] proved that the algebraic degree of a k-resilient boolean function in m variables can be at most $m-k-1$, while Tarannikov [24] showed that the maximum nonlinearity obtainable in k-resilient functions (with $k \leq m-2$) is $2^{m-1}-2^{k+1}$.

Strict Avalanche Criterion and Propagation Criterion

The *Strict Avalanche Criterion* (SAC) was defined by Webster and Tavares in [28] as a more stringent property than the *avalanche effect*. If a boolean function f satisfies the SAC, then whenever a single input bit is complemented the probability that the output bit changes is $1/2$. A generalisation of the SAC, described in [19], is the *propagation criterion* of order l, which takes into account the complementation of at most l input bits.

Definition 3.11. A boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is said to satisfy the *propagation criterion of order l* (with $1 \leq l \leq m$) if, for all nonzero vectors $s \in \mathbb{F}_2^m$ having Hamming weight at most l, the function $f(x) \oplus f(x \oplus s)$ is balanced.

Functions satisfying the propagation criterion of order l are also called $PC(l)$ functions. Similarly to resiliency, in [19] a characterisation of the propagation criterion based on the zeros of the autocorrelation function was proved.

Theorem 3.12. *A boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is $PC(l)$ if and only if $\hat{r}(s) = 0$ for all $s \in \mathbb{F}_2^m$ such that $1 \leq w_{\mathrm{H}}(s) \leq l$.*

Boolean functions which satisfy the SAC and propagation criteria $PC(l)$ for $l > 1$ allow to reach a better *diffusion* in symmetric cryptosystems, particularly in the S-Boxes of block ciphers. From a practical point of view, this means that starting from slightly different inputs the outputs produced by the cipher will be completely different.

Nonexistence of Linear Structures

A boolean function f has a *linear structure* if there exists a nonzero vector $s \in \mathbb{F}_2^m$ such that the function $f(x) \oplus f(x \oplus s)$ is constant. Using boolean functions having linear structures introduces weaknesses in block ciphers, as discussed in [7]. Carlet [1] suggests that boolean functions having linear structures should be avoided in stream ciphers as well, despite the fact that so far there are no known attacks exploiting them.

The following proposition allows one to check the presence of a linear structure in a boolean function by using its autocorrelation function.

Proposition 3.13. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a boolean function and $s \in \mathbb{F}_2^m$ a nonzero input vector. Then, s is a linear structure of f if and only if $|\hat{r}(s)| = 2^m$.*

3.3 Correlation Immunity of Elementary CA Rules

The local rule of a CA can be viewed as a boolean function with an odd number of variables since it is always defined on $2r+1$ cells, where r is the radius. Thus, it possible to verify whether a CA-based PRNG is vulnerable to particular attacks by checking the cryptographic properties of the adopted local rules.

Returning to Wolfram's PRNG, it turns out that rule 30 is both balanced and nonlinear (with $Nl(f_{30}) = 2$), but it is not first order correlation immune. More generally, Martin has shown in [14] by an exhaustive search that, among the 256 elementary rules, only 8 linear rules are 1-resilient. This fact can also be interpreted as a corollary of Tarannikov's bound: if $r = 1$ then the local rule is defined over $m = 3$ variables, and the maximum value of nonlinearity for 1-resilient functions is $2^{3-1} - 2^{1+1} = 0$.

As Martin points out, the fact that rule 30 is not 1-resilient is one of the reasons why the attack discovered by Meier and Staffelbach [15], which exploits the *quasi-linearity* of rule 30, is so efficient on Wolfram's PRNG. In a CA with n cells, an attacker which knows at least $n/2$ consecutive bits of the central trace has only to guess the $n/2$ right bits of the initial configuration, since the left ones can be recovered by a *backwards completion* procedure if the local rule is quasi-linear. Thus, the actual keyspace in Wolfram's PRNG is bounded above by $2^{n/2}$. However, since rule 30 is not first order correlation immune, an attacker can take advantage of the existing correlations between the input variables and the output to further reduce the keyspace. As a matter of fact, Meier and Staffelbach estimated that, in the case of a CA equipped with rule 30 and having $n = 300$ cells, this correlation attack has a complexity of just 18.1 bits, with a success probability $\delta = 0.5$.

This attack shows that even if local rules are employed differently in cellular automata than boolean functions in classic PRNG/stream ciphers models, it is still useful to study their cryptographic properties, such as resiliency. Since there are no elementary rules which are both 1-resilient and nonlinear, the consequence is that it is necessary to explore the spaces of rules having higher radii.

4 BIPERMUTIVE RULES

4.1 Permutive Rules

We now turn to the *permutivity* property of a boolean function, successively applying it to CA local rules. Given $f : \mathbb{F}_2^m \to \mathbb{F}_2$, $x = (x_1, ..., x_{m-1}) \in \mathbb{F}_2^{m-1}$ and $\tilde{x} \in \mathbb{F}_2$, let us denote by $(x, \tilde{x}_{\{i\}})$, with $i \in \{1, ..., m\}$, the vector

$$(x, \tilde{x}_{\{i\}}) = (x_1, ..., x_{i-1}, \tilde{x}, x_i, ..., x_{m-1}) \ \in \mathbb{F}_2^m .$$

In other words, $(x, \tilde{x}_{\{i\}})$ is the vector of $\mathbb{F}_2^m$ created by inserting at position i in x the value $\tilde{x}$, and shifting to the right by one place all the components x_j with $j \geq i$.

Definition 4.1. A boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is called *i-permutive* (or *permutive in the i-th variable*) if, $\forall x = (x_1, ..., x_{m-1}) \in \mathbb{F}_2^{m-1}$, it holds that

$$f(x, 0_{\{i\}}) \neq f(x, 1_{\{i\}}) \ . \tag{7}$$

A function f which is 1-permutive is also called *leftmost permutive* (or *L-permutive*), while a function which is m-permutive is called *rightmost permutive* (or *R-permutive*). We call *bipermutive* a function which is both L-permutive and R-permutive.

In [3] and [4] two important relationships between permutive rules and chaotic CA have been proved, which can be summarised as follows:

Theorem 4.2. *The following sufficient conditions hold:*

1. *A CA based on a local rule f which is bipermutive is E-chaotic.*
2. *A CA based on a local rule f which is either L-permutive or R-permutive is M-chaotic.*

Thus, bipermutive rules induce CA which are strongly chaotic, since they satisfy both the definitions of M-chaos and E-chaos. In the case of elementary CA, rule 30 is R-permutive (and so M-chaotic), while the bipermutive rules are 90, 105, 150 and 165, which are all linear.

4.2 Basic Cryptographic Properties of Bipermutive Rules

We can now prove the following property: bipermutive rules, besides the chaotic behaviour they induce in CA, are also 1-resilient. We begin by showing that if a boolean function is permutive in one of its variables, then it is balanced.

Lemma 4.3. *If $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is i-permutive, then it is balanced.*

Proof. Considering Property 3.2, we rewrite the Walsh Transform of the null vector as follows:

$$\hat{F}(0) = \sum_{\{x \in \mathbb{F}_2^m :\, x_i = 0\}} \hat{f}(x) \; + \sum_{\{x \in \mathbb{F}_2^m :\, x_i = 1\}} \hat{f}(x) \; . \tag{8}$$

The function f is i-permutive, so $\forall x \in \mathbb{F}_2^{m-1}$, $\hat{f}(x, 1_{\{i\}}) = -\hat{f}(x, 0_{\{i\}})$. The second sum in (8) is exactly the opposite of the first sum, hence $\hat{F}(0) = 0$. By Property 3.3, this means that f is balanced.

□

Now we show that bipermutive rules are first order correlation immune.

Lemma 4.4. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be bipermutive. Then f is first order correlation immune.*

Proof. Using the characterisation of correlation immunity given in Theorem 3.10, it is sufficient to show that $\hat{F}(\omega) = 0$ for all $\omega \in \mathbb{F}_2^m$ such that $w_H(\omega) = 1$. Let ω be a generic vector having Hamming weight 1. By Property 3.4, the Walsh Transform of ω can be computed as

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^m} \hat{f}(x) \cdot (-1)^{x_i} \; , \tag{9}$$

where i is the index of the nonzero coordinate in ω. We distinguish two cases:

1. ω has the nonzero coordinate in the first $m-1$ positions (there are $m-1$ vectors of such kind, from $(1,0,...,0,0)$ to $(0,0,...,1,0)$). We rewrite (9) as follows:

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 0_{\{m\}}) \cdot (-1)^{x_i} \; + \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 1_{\{m\}}) \cdot (-1)^{x_i} \; , \tag{10}$$

with $i \in \{1,...,m-1\}$. Since f is R-permutive, $\hat{f}(x, 1_{\{m\}}) = -\hat{f}(x, 0_{\{m\}})$. Moreover, since in (10) x varies in $\mathbb{F}_2^{m-1}$, the terms $(-1)^{x_i}$ are the same in both sums. Thus, it follows that

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 0_{\{m\}}) \cdot (-1)^{x_i} \; - \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 0_{\{m\}}) \cdot (-1)^{x_i} \; = \; 0 \; .$$

2. ω has the nonzero coordinate in the last position, that is, $\omega = (0, 0, ..., 1)$. The Walsh Transform of ω is thus given by

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^m} \hat{f}(x) \cdot (-1)^{x_m} \quad . \tag{11}$$

We observe that the substitution $\hat{f}(x, 1_{\{m\}}) = -\hat{f}(x, 0_{\{m\}})$ used in the previous case does not work here, since the second sum in (10) would gather all the vectors with value 1 in the last coordinate, and the signs would all be changed ($(-1)^{x_m} = -1$, $\forall x \in \mathbb{F}_2^{m-1}$). We thus rewrite (11) as follows:

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 0_{\{1\}}) \cdot (-1)^{x_m} \; + \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 1_{\{1\}}) \cdot (-1)^{x_m} \quad . \tag{12}$$

Now, f is also L-permutive, so $\hat{f}(x, 1_{\{1\}}) = -\hat{f}(x, 0_{\{1\}})$. By using an argument analogous to the one used in case 1, it follows that

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 0_{\{1\}}) \cdot (-1)^{x_m} \; - \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 0_{\{1\}}) \cdot (-1)^{x_m} \; = \; 0 \quad .$$

In conclusion, the Walsh Transform vanishes for all vectors having Hamming weight 1, thus the function f is first order correlation immune.

□

By combining Lemmas 4.3 and 4.4, we finally get the following

Theorem 4.5. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a bipermutive boolean function. Then, f is 1-resilient.*

5 GENERATING BIPERMUTIVE RULES OF A GIVEN RADIUS

5.1 An Enumerative Encoding for Bipermutive Functions

Theorem 4.5 motivates the search for bipermutive boolean functions to be used in CA-based PRNG, since they are both strongly chaotic and of cryptographic interest. The idea is to span the space of bipermutive functions of a given odd number of variables (or, equivalently, of a given radius) in order to check additional cryptographic properties. We propose a simple enumerative encoding which allows us to represent bipermutive functions $f : \mathbb{F}_2^m \to \mathbb{F}_2$ as strings of 2^{m-2} bits.

Let us denote by $\mathcal{F}_m = \{f : \mathbb{F}_2^m \to \mathbb{F}_2\}$ the space of boolean functions in $m \geq 2$ variables, and let $G = (V, E)$ be a graph where $V = \mathbb{F}_2^m$ is the set of vertices. The set of edges $E \subseteq V \times V$ is defined by the following relation: for all $x = (x_1, \cdots, x_m)$ and $y = (y_1, \cdots, y_m) \in V$, the edge $\{x,y\}$ is in E if and only if

$$(x_1 = \bar{y}_1 \ \wedge \ (\forall i \in M_L \ \ x_i = y_i)) \bigvee (x_m = \bar{y}_m \ \wedge \ (\forall i \in M_R \ \ x_i = y_i)) \ \ ,$$

where $M_L = \{2, \cdots, m\}$, $M_R = \{1, \cdots, m-1\}$ and $\bar{y}_j$ is the complement of bit y_j. In other words, the edges in E connect those inputs in $\mathbb{F}_2^m$ which must have different output values in order to satisfy either L-permutivity or R-permutivity in a boolean function. The relation which defines E is symmetric, so the graph G is undirected. We now show some simple properties of G.

Property 5.1. The degree of each node $x \in V$ is 2. In fact, for all $x \in \mathbb{F}_2^m$, there exists a unique $x' \in \mathbb{F}_2^m$ such that $x_1 = \bar{x}'_1$ and $x_i = x'_i$ for all $i \in M_L$. Similarly, there exists a unique $x'' \in \mathbb{F}_2^m$ such that $x'' \neq x'$ and $x_m = \bar{x}''_m$ and $x_i = x''_i$ for all $i \in M_R$.

Property 5.2. Let x, y be vectors of $\mathbb{F}_2^m$ such that $x_1 = \bar{y}_1$, $x_m = \bar{y}_m$ and $x_i = y_i$ for all $i \in M_L \cap M_R = \{2, \cdots, m-1\}$. Then, the two adjacent nodes of x are the same as the adjacent nodes of y. In fact, let us suppose that $x', x'' \in \mathbb{F}_2^m$ are the two adjacent nodes of x, in particular that $x_1 = \bar{x}'_1$, $x_i = x'_i$ for all $i \in M_L$ and $x_m = \bar{x}''_m$, $x_i = x''_i$ for all $i \in M_R$. Then, $x'_1 = y_1$ and $x'_i = y_i$ for all $i \in M_L \cap M_R$. Since $x_m = x'_m$, it follows that $y_m = \bar{x}'_m$, so $\{y, x'\} \in E$. A similar argument shows that $\{y, x''\} \in E$, so x', x'' are also the adjacent nodes of y.

Property 5.3. Since the relation which defines E is symmetric, from Property 5.2 we can deduce that the adjacent nodes of x' and x'' are exactly x and y, hence $\{x, x', x'', y\}$ is a connected component of G. There are 2^{m-2} pairs $\{x,y\}$ of vectors which differ in the leftmost and rightmost coordinates and coincide in the $m-2$ central ones. Thus G is composed of 2^{m-2} disjoint connected components of this kind.

A boolean function $f \in \mathcal{F}_m$ is essentially a label function $f : V \to \mathbb{F}_2$ on the vertices of G. If f is bipermutive then the label of each node x is different from the labels of its two adjacent nodes, while the labels of the nodes which are connected via a path of length 2 are the same. Considering Property 5.3, this means that the label of a single node uniquely determines the labels of the remaining nodes in the connected component where x resides. So, in the case of a bipermutive function, we can define the *configuration* of a generic connected component in G as the value of the label of one of its nodes x, called the *representative* of the connected component. The most natural choice is to select in each connected component the node x whose binary vector encodes the smallest integer number as representative, which is the one having value

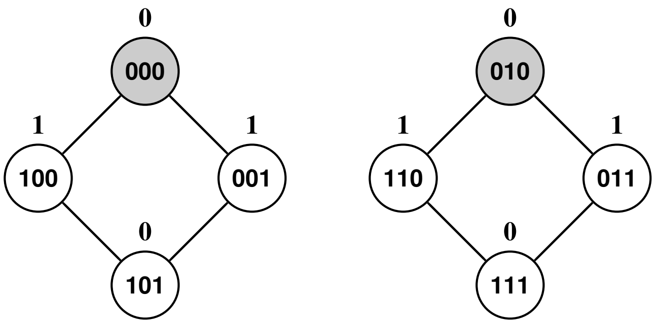

Figure 1
Representation of the bipermutive function 01011010 (rule 90) on the corresponding graph G. The representatives are shaded in gray, so this function corresponds to the string $c = 00$.

zero in the leftmost and rightmost coordinates. From a 2^{m-2}-bit string c we can thus recover the truth table of the corresponding bipermutive function as follows: for all $j \in \{0, \cdots, 2^{m-2}-1\}$, we label the representative r_j of the j-th connected component with the value c_j. The adjacent nodes of r_j are then labelled with $\bar{c}_j$, and the last node in the connected component (the one having nonzero value in the leftmost and rightmost coordinates) is labelled with c_j. Figure 1 reports an example of bipermutive rule represented on the graph G in the case of $m = 3$ variables. Given $m \in \mathbb{N}$, there are exactly $2^{2^{m-2}}$ bipermutive functions of m variables; moreover, by using this choice of representatives in G the truth tables of the functions can be enumerated in lexicographic order.

5.2 Application to the case $r = 2$

It has already been observed in Section 4.1 that in the set of elementary CA ($r = 1$) there are only four affine bipermutive rules. We have thus used the enumerative encoding described in Section 5.1 to explore the set of $2^{2^3} = 256$ bipermutive rules of radius $r = 2$. The algorithm used to generate these functions is straightforward, since it simply loops on the set $\{0, \cdots, 255\}$, converts each integer i to the corresponding binary expansion c_i and instantiates the labels on the vertices of G according to the configurations of the connected components encoded by the value of c_i.

By applying Tarannikov's bound to the case of boolean functions of 5 variables (which is exactly the set of CA rules of radius 2) we see that, with

Res	(NL, AD, LS)		
	(0, 1, 31)	(4, 3, 7)	(8, 2, 3)
1-Res	2	128	56
2-Res	14	0	56

Table 1
Distribution of bipermutive rules of radius 2 with respect to resiliency (1-Res, 2-Res), and the three observed combinations of nonlinearity (NL), algebraic degree (AD) and number of linear structures (LS).

respect to the property of nonlinearity, there can be 1-resilient rules with $Nl = 12$ and 2-resilient rules with $Nl = 8$. For higher orders of resiliency, there are only linear (affine) functions. For each bipermutive rule generated by our algorithm, we checked its cryptographic properties by computing its Walsh Transform, autocorrelation function and ANF.

We now describe the observed results. The maximum value achieved for nonlinearity was $Nl = 8$ also in the rules which satisfied only 1-resiliency, whereas the other two values for nonlinearity were $Nl = 4$ and $Nl = 0$. On the other hand, the set of 2-resilient rules was composed only of linear rules or rules having $Nl = 8$. Concerning the other cryptographic properties, all the 256 bipermutive rules did not satisfy the Strict Avalanche Criterion (that is, $PC(1)$) and featured linear structures, ranging from the extreme case of 31 structures for linear functions to a minimum of 3. Regarding the algebraic degree, the 2-resilient nonlinear rules had value $d = 2$, thus they reached also Siegenthaler's bound. Among the rules which were only 1-resilient, the nonlinear ones had algebraic degree $d = 2$ and $d = 3$ (reaching in the latter case Siegenthaler's bound). Table 1 recaps the cardinalities of subsets of bipermutive rules with respect to resiliency, nonlinearity, algebraic degree and number of linear structures.

We thus selected the set of nonlinear and 2-resilient rules, considering that they achieve both Tarannikov's and Siegenthaler's bounds. Consulting Table 1, this left us in total with 56 rules.

We successively subjected these resulting 2-resilient nonlinear bipermutive rules to a series of statistical tests, in order to find which of them generate pseudorandom sequences which are at least as good as the ones produced by rule 30. We structured our analysis in two phases. First, we removed the rules which generated small pseudorandom sequences (2^{16} bits) that did not pass the tests of the ENT suite [27], using rule 30 as a benchmark. Then, we applied the NIST test suite [18] to longer sequences (10^6 bits) generated by the remaining rules. In both phases, we used Wolfram's method for pseudorandom generation. In particular, we employed a finite CA with periodic boundary conditions composed of $n = 64$ cells (since 64 bits is a common value for the length of the seed in many standard PRNG, like ANSI X9.17) and we sampled the trace of the 32nd cell to generate the pseudorandom sequences.

The ENT Test Suite, assembled by Walker and described in [27], is a battery of 5 statistical tests (Entropy, Chi-Square, Arithmetic Mean, Monte Carlo Value for π and Serial Correlation Coefficient) which can be used to check the quality of pseudorandom sequences. For each bipermutive 2-resilient nonlinear rule of radius 2 we generated a single sequence of length $l = 2^{16} = 65536$ bits, using as initial seed the configuration containing only a 1 in the 32nd cell. This method is similar to the one adopted by Koza in [10], where he evolved a CA-based PRNG by a genetic programming algorithm (even if, in that case, the fitness function was only the entropy of the generated sequence). Interestingly, the best rule found by Koza with his approach was rule 30.

As a first selection step, we discarded the rules which did not generate sequences that passed the Chi-Square test, since this is the most sensitive test in detecting deviations from randomness. As suggested in [27], a sequence passes the Chi-Square test if the corresponding p-value is included in the interval $[0.1, 0.9]$. After this selection, only 42 rules remained, and we subsequently compared their results with those obtained by rule 30, selecting only the ones with an error $err_\pi < 1\%$ in the approximation of π. The resulting 28 rules were similar or even better than rule 30 with respect to the other tests (entropy, arithmetic mean and serial correlation coefficient), so no further selection was performed.

We observed that 24 rules presented the same ENT results in pairs. This fact was expected, since in each pair the rules are related by the *reflexive* transformation, mentioned in [14]. Given a vector $x = (x_1, \cdots, x_m) \in \mathbb{F}_2^m$, the *mirror image* of x is defined as $x_M = (x_m, \cdots, x_1)$. The *reflex* of a boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is the function f_R defined as $f_R(x) = f(x_M)$, $\forall x \in \mathbb{F}_2^m$. The remaining non-coupled four rules are self-reflexive, that is, $f_R(x) = f(x)$.

The reflection preserves all the cryptographic properties of a function. In fact, given a balanced boolean function f the resulting reflected function f_R is also balanced, since using another ordering of the input variables does not influence the output distribution of f. Further, the algebraic normal form of f_R

may be determined by simply substituting each term x_i in the polynomial representing the ANF of f by its mirrored version x_{M_i}, thus the algebraic degree of f_R equals the degree of f. Moreover, the Hamming weight of an input vector $x \in \mathbb{F}_2^m$ is clearly preserved under the mirroring operation. Consequently, the Walsh and autocorrelation spectra of f_R are permutations of the corresponding spectra of f. In particular, the spectral radius remains unaltered, thus $Nl(f_R) = Nl(f)$. Additionally, if the Walsh Transform of f vanishes for all vectors having a fixed Hamming weight, then the same stands for the corresponding vectors of f_R. Hence, f is k-th order correlation immune if and only if also f_R satisfies this property. By an analogous argument on the autocorrelation functions, it is easily seen that f satisfies the propagation criterion $PC(l)$ if and only if f_R satisfies it. Finally, if f has a linear structure $a \in \mathbb{F}_2^m$, $a \neq 0$, then the vector a_M is a linear structure for f_R.

Considering our method of pseudorandom generation described earlier, it follows that two rules equivalent by reflexive transformation produce two sequences of configurations which are symmetric, thus the trace of the 32nd cell is the same.

Table 2 shows the ENT results of the 28 final rules, grouped by reflection pairs. In each pair the rule having the suffix “a” is the one with the highest Wolfram code[⋆], while self-reflexive rules are identified by the label “(sr)”. The results of the elementary rule 30 are also reported for comparison.

To further investigate the randomness quality of the rules selected with the ENT suite, we applied the more stringent statistical tests devised by the NIST in [18] to longer generated sequences. For each pair of rules equivalent by reflexive transformation, we chose to test only the rule with the smallest Wolfram code (since the other is expected to show a similar pseudorandom behaviour), so in total we tested 12 rules plus the 4 self-reflexive ones.

The NIST suite includes 15 tests, some of which are repeated several times with different parameters and patterns: the total number of tests run on each sample of pseudorandom sequences is thus 187. The technical details of the tests can be found in [18]. For the sake of our discussion, it is sufficient to know that each test in the suite produces a p-value for each sequence in the sample, and that the sequence passes the test if its corresponding p-value is included in the confidence interval $[\alpha, 1-\alpha]$, where α is the significance level. Then, the results of a test over the entire sample of sequences generated by a rule are interpreted using two approaches. First, the proportion of passing

[⋆] The Wolfram codes of the local rules of radius 2 and 3 referenced in this paper can be found at `http://openit.disco.unimib.it/~mariot/wolfram_codes.html`.

Rule - Reflex	E_8	χ^2	μ_{dev}	err_π	scc
R01 - R01a	7.979592	0.83	0.004848	0.37%	-0.002338
R02 - R02a	7.977838	0.56	0.008593	0.66%	0.002280
R03 - R03a	7.979487	0.85	0.000567	0.37%	-0.003930
R04 - R04a	7.978750	0.69	0.004215	0.75%	0.003161
R05 - R05a	7.976643	0.30	0.003097	0.01%	-0.012526
R06 (sr)	7.977783	0.57	0.003332	0.10%	0.003791
R07 - R07a	7.976146	0.32	0.001983	0.01%	0.015071
R08 (sr)	7.979135	0.82	0.006708	0.09%	0.001310
R09 - R09a	7.976625	0.34	0.008589	0.18%	0.017063
R10 (sr)	7.976147	0.27	0.004326	0.38%	0.002607
R11 - R11a	7.977823	0.52	0.005322	0.38%	-0.013957
R12 - R12a	7.976643	0.30	0.005385	0.55%	-0.025343
R13 - R13a	7.978825	0.73	0.000548	0.10%	-0.005077
R14 - R14a	7.978674	0.76	0.008456	0.57%	0.013556
R15 (sr)	7.979135	0.82	0.000952	0.75%	-0.010592
R16 - R16a	7.978866	0.83	0.007370	0.66%	0.011000
rule 30	7.979031	0.80	0.004169	0.66%	-0.013926

Table 2
ENT tests results on the pseudorandom sequences generated by the 28 rules after the selection process. E_8 stands for the entropy computed on an 8-bit schema, χ^2 is the p-value of the Chi-Square test, μ_{dev} is the normalised deviation from the mean value $\mu = 127.5$, err_π is the error in the approximation of π and scc is the Serial Correlation Coefficient.

sequences is computed, and this proportion is considered acceptable if it lies above the *minimum pass rate*

$$mpr = \hat{p} - 3\sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \ ,$$

where $\hat{p} = 1 - \alpha$ and N is the sample size. Second, a Chi-Square test is performed to verify whether the p-values are well distributed, by dividing $[0,1]$ in 10 subintervals.

To set up the parameters of the tests, we followed the recommendations suggested in [18]. In particular, for each rule we generated a sample of $N = 1000$ pseudorandom sequences of length $l = 10^6$ bits. The 64-bit seeds for the CA have been created with the *HotBits* service (available at `http://www.fourmilab.ch/hotbits/`), which is a true random number generator (TRNG) based on the radioactive decays of a Caesium-137 source. The significance level adopted was $\alpha = 0.001$.

Table 3 reports the results of the 16 rules tested (along with rule 30, always used as a benchmark). For each rule, the value in the column "Approach 1" refers to the number of tests passed with respect to the proportions of passing sequences, while the value in "Approach 2" represents the number of tests passed with respect to the distribution of p-values. We can observe that, except for rule R05, the worst results are obtained by the self-reflexive rules, with very low pass rates concerning Approach 1. The reason could lie in the intrinsic symmetries of the space-time diagrams produced by this kind of rules, which are evident by using our pseudorandom generation method (with the initial configuration having only a 1 in the central cell).

The remaining rules all have pass rates close to the maximum, and three of them (R01, R07 and R15) pass all the tests with respect to both approaches, like rule 30.

6 FURTHER CRYPTOGRAPHIC PROPERTIES OF BIPERMUTIVE RULES

6.1 Shannon Decomposition and Generating Functions

The exhaustive exploration carried out for the case of radius $r = 2$ gave us some hints about the cryptographic properties of bipermutive rules, apart from 1-resiliency. For instance, we noticed that the values of nonlinearity are all multiples of 4. Moreover, if we add the 14 affine 2-resilient bipermutive functions to the 56 nonlinear ones we get a total of 70 rules, which is exactly the number of balanced 3-variable boolean functions (in fact, $\binom{8}{4} = 70$). As a

Rule	Approach 1	Approach 2
R01	187/187	187/187
R02	186/187	186/187
R03	187/187	186/187
R04	186/187	184/187
R05	187/187	37/187
R06 (sr)	184/187	94/187
R07	187/187	187/187
R08 (sr)	186/187	25/187
R09	187/187	185/187
R10 (sr)	187/187	24/187
R11	187/187	186/187
R12	186/187	129/187
R13	186/187	187/187
R14	186/187	186/187
R15 (sr)	185/187	25/187
R16	187/187	187/187
rule 30	187/187	187/187

Table 3
NIST tests results on the pseudorandom sequences generated by the 16 final rules of radius 2 and the elementary rule 30, used as a benchmark.

matter of fact, the set of all boolean functions defined over 3 variables coincides with the space of binary configurations which encode bipermutive rules of radius 2.

These empirical observations made us conjecture that the cryptographic properties of bipermutive rules are related to those of their graph configurations, the latter considered as boolean functions themselves. From this section on we pursue this idea, proving several other results about the cryptographic characterisation of bipermutive rules.

We start our investigation by studying the ANF of bipermutive rules using the *Shannon decomposition formula* [21]. In particular, let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a boolean function and $i \in \{1, \cdots, m\}$. Then, for all $x \in \mathbb{F}_2^{m-1}$ and $y \in \mathbb{F}_2$ the following identity holds:

$$f(x, y_{\{i\}}) = y \cdot f(x, 1_{\{i\}}) \oplus \bar{y} \cdot f(x, 0_{\{i\}}) \ . \tag{13}$$

In what follows, given a vector $x \in \mathbb{F}_2^{m-2}$ and $x_1, x_m \in \mathbb{F}_2$, by (x_1, x, x_m) we denote the vector of $\mathbb{F}_2^m$ obtained by juxtaposing x_1, x and x_m.

Let us suppose that $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is a bipermutive function. We begin by decomposing f with respect to x_1.

$$f(x_1, x, x_m) = x_1 \cdot f(1, x, x_m) \oplus \bar{x}_1 \cdot f(0, x, x_m) \ . \tag{14}$$

We now apply the same decomposition to $f(1, x, x_m)$ and $f(0, x, x_m)$ with respect to x_m:

$$f(1, x, x_m) = x_m \cdot f(1, x, 1) \oplus \bar{x}_m \cdot f(1, x, 0) \ , \tag{15}$$

$$f(0, x, x_m) = x_m \cdot f(0, x, 1) \oplus \bar{x}_m \cdot f(0, x, 0) \ . \tag{16}$$

By substituting (15) and (16) in equation (14) we get:

$$\begin{aligned} f(x_1, x, x_m) = x_1 \cdot [x_m \cdot f(1, x, 1) \oplus \bar{x}_m \cdot f(1, x, 0)] \oplus \\ \oplus \bar{x}_1 \cdot [x_m \cdot f(0, x, 1) \oplus \bar{x}_m \cdot f(0, x, 0)] \ . \end{aligned} \tag{17}$$

Thus, we have rewritten f by using four functions in $m-2$ variables: $f(1, x, 1)$, $f(0, x, 1)$, $f(1, x, 0)$ and $f(0, x, 0)$. Since f is also bipermutive, for all $x \in \mathbb{F}_2^{m-2}$ the following relations hold:

$$f(1, x, 1) = f(0, x, 0) \ ;$$

$$f(1, x, 0) = f(0, x, 1) = \overline{f(0, x, 0)} \ .$$

We now reformulate equation (17) in terms of $f(0,x,0)$. Since $f(0,x,0)$ depends only on the $m-2$ central variables, we relabel it by $g(x)$.

$$f(x_1,x,x_m) = \; x_1 \cdot [x_m \cdot g(x) \oplus \bar{x}_m \cdot \overline{g(x)}] \; \oplus \; \bar{x}_1 \cdot [x_m \cdot \overline{g(x)} \oplus \bar{x}_m \cdot g(x))] \; . \quad (18)$$

In order to simplify equation (18), we multiply the terms in parentheses by x_1 and x_m, and regroup them with respect to $g(x)$ and $\overline{g(x)}$, obtaining:

$$\begin{aligned} f(x_1,x,x_m) &= x_1 \cdot [x_m \cdot g(x) \oplus \bar{x}_m \cdot \overline{g(x)}] \; \oplus \; \bar{x}_1 \cdot [x_m \cdot \overline{g(x)} \oplus \bar{x}_m \cdot g(x))] = \\ &= g(x) \cdot (\overline{x_1 \oplus x_m}) \oplus \overline{g(x)} \cdot (x_1 \oplus x_m) = \\ &= g(x) \cdot [1 \oplus (x_1 \oplus x_m)] \oplus [1 \oplus g(x)] \cdot (x_1 \oplus x_m) = \\ &= g(x) \oplus \cancel{[g(x) \cdot (x_1 \oplus x_m)]} \oplus (x_1 \oplus x_m) \oplus \cancel{[g(x) \cdot (x_1 \oplus x_m)]} = \\ &= g(x) \oplus x_1 \oplus x_m \;\; . \end{aligned} \quad (19)$$

Hence, we can evaluate a bipermutive boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ on a particular input vector $(x_1,x,x_m) \in \mathbb{F}_2^m$ by simply computing the restriction $g(x) = f(0,x,0)$ on the central $m-2$ variables and by summing to it modulo 2 the values of the leftmost and rightmost variables. It is easy to see that the truth table of g corresponds to the graph encoding of f discussed in Section 5.1. For this reason, we call g the *generating function* of f.

The following result on the algebraic degree of a bipermutive function is now immediate:

Theorem 6.1. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a bipermutive function with generating function $g : \mathbb{F}_2^{m-2} \to \mathbb{F}_2$ having algebraic degree $deg(g) \geq 1$. Then, the algebraic degree of f equals $deg(g)$.*

Proof. Simply observe that the algebraic normal form of f, as shown in equation (19), is obtained by adding two terms of degree 1 to the ANF of g. □

6.2 Walsh Spectrum and Nonlinearity

We now show how the Walsh spectrum of a bipermutive function f can be efficiently computed by using the spectrum of its generating function g.

Lemma 6.2. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a bipermutive function with generating function $g : \mathbb{F}_2^{m-2} \to \mathbb{F}_2$. Then, for all $\omega \in \mathbb{F}_2^{m-2}$,*

$$\hat{F}(1,\omega,1) = 4 \cdot \hat{G}(\omega) \; . \quad (20)$$

Proof. Let us first rewrite the Walsh transform of the vector $(1,\omega,1)$ by grouping the terms in the sum with respect to the values of x_1 and x_m:

$$\begin{aligned}\hat{F}(1,\omega,1) = & \sum_{x\in\mathbb{F}_2^{m-2}} \hat{f}(0,x,0)\cdot(-1)^{(1,\omega,1)\cdot(0,x,0)} + \\ & + \sum_{x\in\mathbb{F}_2^{m-2}} \hat{f}(1,x,1)\cdot(-1)^{(1,\omega,1)\cdot(1,x,1)} + \\ & + \sum_{x\in\mathbb{F}_2^{m-2}} \hat{f}(0,x,1)\cdot(-1)^{(1,\omega,1)\cdot(0,x,1)} + \\ & + \sum_{x\in\mathbb{F}_2^{m-2}} \hat{f}(1,x,0)\cdot(-1)^{(1,\omega,1)\cdot(1,x,0)} \ .\end{aligned} \tag{21}$$

We now have to evaluate all scalar products in (21). For all $x \in \mathbb{F}_2^{m-2}$, the following relations hold:

$$\begin{aligned}(1,\omega,1)\cdot(0,x,0) &= 1\cdot 0 \oplus \omega\cdot x \oplus 1\cdot 0 = \omega\cdot x \ ; \\ (1,\omega,1)\cdot(1,x,1) &= 1\cdot 1 \oplus \omega\cdot x \oplus 1\cdot 1 = \omega\cdot x \ ; \\ (1,\omega,1)\cdot(0,x,1) &= 1\cdot 0 \oplus \omega\cdot x \oplus 1\cdot 1 = \omega\cdot x \oplus 1 \ ; \\ (1,\omega,1)\cdot(1,x,0) &= 1\cdot 1 \oplus \omega\cdot x \oplus 1\cdot 0 = \omega\cdot x \oplus 1 \ .\end{aligned}$$

Moreover, since f is bipermutive, it follows that $\hat{f}(0,x,0) = \hat{f}(1,x,1)$ and $\hat{f}(0,x,1) = \hat{f}(1,x,0)$ for all $x \in \mathbb{F}_2^{m-2}$. By substituting (21), we can simplify the expression obtaining

$$\hat{F}(1,\omega,1) = 2\cdot\left(\sum_{x\in\mathbb{F}_2^{m-2}} \hat{f}(1,x,1)\cdot(-1)^{\omega\cdot x} + \sum_{x\in\mathbb{F}_2^{m-2}} \hat{f}(1,x,0)\cdot(-1)^{\omega\cdot x\oplus 1}\right) \ . \tag{22}$$

Since $(-1)^{\omega\cdot x\oplus 1} = (-1)\cdot(-1)^{\omega\cdot x}$, the second sum in (22) changes sign while the scalar product becomes the same as that in the first sum:

$$\hat{F}(1,\omega,1) = 2\cdot\left(\sum_{x\in\mathbb{F}_2^{m-2}} \hat{f}(1,x,1)\cdot(-1)^{\omega\cdot x} - \sum_{x\in\mathbb{F}_2^{m-2}} \hat{f}(1,x,0)\cdot(-1)^{\omega\cdot x}\right) \ . \tag{23}$$

The function f is R-permutive, so $\hat{f}(1,x,1) = -\hat{f}(1,x,0)$ for all $x \in \mathbb{F}_2^{m-2}$. Then, Equation (23) can be rewritten as:

$$\begin{aligned}\hat{F}(1,\omega,1) = 2\cdot\left(\sum_{x\in\mathbb{F}_2^{m-2}}\hat{f}(1,x,1)\cdot(-1)^{\omega\cdot x}+\sum_{x\in\mathbb{F}_2^{m-2}}\hat{f}(1,x,1)\cdot(-1)^{\omega\cdot x}\right) = \\ = 4\cdot\left(\sum_{x\in\mathbb{F}_2^{m-2}}\hat{f}(1,x,1)\cdot(-1)^{\omega\cdot x}\right) \quad .\end{aligned} \tag{24}$$

Finally, we know that $\hat{f}(1,x,1)=\hat{f}(0,x,0)=\hat{g}(x)$. Since the last sum in (24) varies on $\mathbb{F}_2^{m-2}$, it is exactly the Walsh transform of g on ω:

$$\hat{F}(1,\omega,1) = 4\cdot\left(\sum_{x\in\mathbb{F}_2^{m-2}}\hat{g}(x)\cdot(-1)^{\omega\cdot x}\right) = 4\cdot\hat{G}(\omega) \quad . \tag{25}$$

□

Lemma 6.3. *Let $f:\mathbb{F}_2^m\rightarrow\mathbb{F}_2$ be a bipermutive function and $g:\mathbb{F}_2^{m-2}\rightarrow\mathbb{F}_2$ its generating function. Then, $\hat{F}(\omega)=0$ for all $\omega\in\mathbb{F}_2^m$ such that $\omega_1=0$ or $\omega_m=0$.*

Proof. By Lemma 6.2 we know that $\hat{F}(1,\omega,1)=4\cdot\hat{G}(\omega)$ for all $\omega\in\mathbb{F}_2^{m-2}$. Then, the sum of these squared Walsh coefficients is

$$\sum_{\tilde{\omega}\in\mathbb{F}_2^{m-2}}\hat{F}^2(1,\tilde{\omega},1) = \sum_{\tilde{\omega}\in\mathbb{F}_2^{m-2}}16\cdot\hat{G}^2(\tilde{\omega}) \;=\; 2^4\cdot\sum_{\tilde{\omega}\in\mathbb{F}_2^{m-2}}\hat{G}^2(\tilde{\omega}) \quad . \tag{26}$$

If we apply Parseval's relation to the generating function g, it follows that

$$\sum_{\tilde{\omega}\in\mathbb{F}_2^{m-2}}\hat{G}^2(\tilde{\omega}) = 2^{2(m-2)} \quad . \tag{27}$$

By substituting the result of (27) in equation (26) we get

$$2^4\cdot\sum_{\tilde{\omega}\in\mathbb{F}_2^{m-2}}\hat{G}^2(\tilde{\omega}) = 2^4\cdot 2^{2(m-2)} = 2^{4+2m-4} = 2^{2m} \quad . \tag{28}$$

We have thus concluded that, in a bipermutive rule, the sum of the squared Walsh coefficients $\hat{F}^2(1,\tilde{\omega},1)$ equals 2^{2m}. Parseval's relation also tells us that 2^{2m} is the sum of *all* squared Walsh coefficients of f. This means that for all $\tilde{\omega}\in\mathbb{F}_2^{m-2}$ the remaining coefficients $\hat{F}(0,\tilde{\omega},0)$, $\hat{F}(0,\tilde{\omega},1)$ and $\hat{F}(1,\tilde{\omega},0)$ must necessarily be null, hence the thesis. □

We can now give another proof of the fact that bipermutive functions are 1-resilient.

Theorem 6.4. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a bipermutive function. Then f is 1-resilient.*

Proof. Lemma 6.3 states that the Walsh transform of f vanishes for all vectors ω belonging to the set $S = \{\omega \in \mathbb{F}_2^m : \omega_1 = 0 \vee \omega_m = 0\}$. Clearly, S includes all the vectors having Hamming weight at most 1, hence f is 1-resilient. □

The next theorem shows how the nonlinearity of a bipermutive function is related to the nonlinearity of its generating function.

Theorem 6.5. *Given a bipermutive function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ and its generating function $g : \mathbb{F}_2^{m-2} \to \mathbb{F}_2$, the nonlinearity of f is equal to*

$$Nl(f) = 4 \cdot Nl(g) \ . \tag{29}$$

Proof. In order to determine the nonlinearity of f we have to compute its spectral radius $W_{\max}(f)$. Using Lemma 6.2, it is easy to see that $W_{\max}(f) = 4 \cdot W_{\max}(g)$. Thus, the nonlinearity of f is

$$Nl(f) = 2^{m-1} - \frac{W_{\max}(f)}{2} = 2^{m-1} - \frac{4 \cdot W_{\max}(g)}{2} = 2^{m-1} - 2 \cdot W_{\max}(g) \ . \tag{30}$$

Furthermore, we can express $W_{\max}(g)$ in terms of $Nl(g)$ as follows:

$$Nl(g) = 2^{m-3} - \frac{W_{\max}(g)}{2} \ . \tag{31}$$

Hence,

$$W_{\max}(g) = 2 \cdot (2^{m-3} - Nl(g)) \ . \tag{32}$$

By substituting (31) in equation (30) we finally get

$$Nl(f) = 2^{m-1} - 2^2 \cdot (2^{m-3} - Nl(g)) = 2^{m-1} - 2^{m-1} + 2^2 \cdot Nl(g) = 4 \cdot Nl(g) \ . \tag{33}$$

□

Consequently, Theorem 6.5 explains why in our exhaustive exploration of bipermutive rules of radius 2 we only found functions having nonlinearity values which were multiples of 4.

6.3 *k*-Resiliency

The next theorem generalises Theorems 4.5 and 6.4 to *k*-resilient bipermutive functions.

Theorem 6.6. *Let* $f : \mathbb{F}_2^m \to \mathbb{F}_2$ *be a bipermutive function having generating function* $g : \mathbb{F}_2^{m-2} \to \mathbb{F}_2$*. Then,* f *is k-resilient if and only if* g *is* $(k-2)$*-resilient.*

Proof. Let us suppose that f is k-resilient. We have to verify that $\hat{G}(\omega) = 0$ for all $\omega \in \mathbb{F}_2^{m-2}$ such that $w_H(\omega) \leq k-2$. Since f is also bipermutive, using Lemma 6.2 we can compute the Walsh transform of g as follows:

$$\hat{G}(\omega) = \frac{\hat{F}(1,\omega,1)}{4} \quad . \tag{34}$$

It is clear that if $w_H(1,\omega,1) \leq k$ then $w_H(\omega) \leq k-2$. Since $\hat{F}(1,\omega,1) = 0$ for all $\omega \in \mathbb{F}_2^{m-2}$ having Hamming weight at most $k-2$ (because f is k-resilient), by equation (34) it also follows that $\hat{G}(\omega) = 0$ for such ω. Thus, we deduced that g is $(k-2)$-resilient.

Next, let us suppose that g is $(k-2)$-resilient. By Lemma 6.3 we already know that the Walsh transform of f vanishes for all vectors $\omega \in \mathbb{F}_2^m$ whose leftmost or rightmost component is zero, thus we have to check the condition of k-resiliency only for the remaining vectors of the kind $(1,\omega,1)$ having Hamming weight at most k. Since g is $(k-2)$-resilient and considering equation (34), for all $\omega \in \mathbb{F}_2^{m-2}$ with $w_H(\omega) \leq k-2$ the following equalities hold:

$$\hat{G}(\omega) = 0 = \hat{F}(1,\omega,1) \quad . \tag{35}$$

Finally, if ω has Hamming weight at most $k-2$, then the weight of the vector $(1,\omega,1)$ is at most k, hence f is k-resilient. □

We can now understand why the number of 2-resilient bipermutive functions of radius 2 coincides with the cardinality of balanced boolean functions in 3 variables. In fact, balanced functions can be considered as being 0-resilient, thus by Theorem 6.6 the order of resiliency of bipermutive rules generated from balanced generating functions is $k = 0+2 = 2$.

6.4 Autocorrelation Spectrum, SAC and Linear Structures

The remaining cryptographic properties to be investigated are those related to the autocorrelation function, namely the propagation criterion $PC(l)$ and the presence of linear structures. In what follows, we denote by $\underline{0}$ the null vector of $\mathbb{F}_2^{m-2}$.

Lemma 6.7. *Given $i \in \{1, \cdots, m\}$, let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be an i-permutive function, and let $\hat{r} : \mathbb{F}_2^m \to \mathbb{R}$ be the autocorrelation function. Then, $\hat{r}(0, 1_{\{i\}}) = -2^m$. Moreover, if f is bipermutive, then $\hat{r}(1, \underline{0}, 1) = 2^m$.*

Proof. We begin by assuming that f is i-permutive. Since $(0, 1_{\{i\}})$ is the vector composed of zeros except in the i-th component, for all $x \in \mathbb{F}_2^{m-1}$ it follows that

$$\begin{gathered}((x, 0_{\{i\}}) \oplus (0, 1_{\{i\}})) = (x, 1_{\{i\}}) \\ ((x, 1_{\{i\}}) \oplus (0, 1_{\{i\}})) = (x, 0_{\{i\}})\end{gathered}$$

As a consequence, we can rewrite the autocorrelation function $\hat{r}(0, 1_{\{i\}})$ as follows:

$$\hat{r}(0, 1_{\{i\}}) = \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 0_{\{i\}}) \cdot \hat{f}(x, 1_{\{i\}}) + \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 1_{\{i\}}) \cdot \hat{f}(x, 0_{\{i\}}) \ . \quad (36)$$

Since f is i-permutive, all the products $\hat{f}(x, 0_{\{i\}}) \cdot \hat{f}(x, 1_{\{i\}})$ and $\hat{f}(x, 1_{\{i\}}) \cdot \hat{f}(x, 0_{\{i\}})$ give -1 as a result, thus each of the two sums in (36) equals -2^{m-1}:

$$\hat{r}(0, 1_{\{i\}}) = -2^{m-1} - 2^{m-1} = -2^m \ . \quad (37)$$

Next, let us suppose that f is bipermutive. For all $x \in \mathbb{F}_2^{m-2}$ the following identities hold:

$$\begin{gathered}(0, x, 0) \oplus (1, \underline{0}, 1) = (1, x, 1) \\ (1, x, 0) \oplus (1, \underline{0}, 1) = (0, x, 1) \\ (0, x, 1) \oplus (1, \underline{0}, 1) = (1, x, 0) \\ (1, x, 1) \oplus (1, \underline{0}, 1) = (0, x, 0)\end{gathered}$$

Hence, $\hat{r}(1, \underline{0}, 1)$ can be expressed as

$$\begin{aligned}\hat{r}(1, \underline{0}, 1) &= \sum_{x \in \mathbb{F}_2^{m-2}} \hat{f}(0, x, 0) \cdot \hat{f}(1, x, 1) + \sum_{x \in \mathbb{F}_2^{m-2}} \hat{f}(1, x, 0) \cdot \hat{f}(0, x, 1) + \\ &+ \sum_{x \in \mathbb{F}_2^{m-2}} \hat{f}(0, x, 1) \cdot \hat{f}(1, x, 0) + \sum_{x \in \mathbb{F}_2^{m-2}} \hat{f}(1, x, 1) \cdot \hat{f}(0, x, 0) = \\ &= 2 \cdot \left(\sum_{x \in \mathbb{F}_2^{m-2}} \hat{f}(0, x, 0) \cdot \hat{f}(1, x, 1) + \sum_{x \in \mathbb{F}_2^{m-2}} \hat{f}(1, x, 0) \cdot \hat{f}(0, x, 1) \right) \ .\end{aligned} \quad (38)$$

Since f is bipermutive, $\hat{f}(0,x,0) = \hat{f}(1,x,1)$ and $\hat{f}(1,x,0) = \hat{f}(0,x,1)$. As a consequence, all the products in equation (38) give 1 as a result, thus each sum equals 2^{m-2}:

$$\hat{r}(1,\underline{0},1) = 2 \cdot (2^{m-2} + 2^{m-2}) = 2^m \quad . \tag{39}$$

□

We can use Lemma 6.7 to make the following conclusion about the propagation criterion and the linear structures of a bipermutive function.

Theorem 6.8. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a bipermutive function. Then, f does not satisfy $PC(1)$ (i.e., the Strict Avalanche Criterion) and furthermore f has at least three nonzero linear structures.*

Proof. A boolean function satisfies $PC(1)$ if and only if the autocorrelation function is null for all vectors having Hamming weight 1. Since f is bipermutive, by Lemma 6.7 we know that the autocorrelation function of the two vectors $(0,1_{\{1\}})$ and $(0,1_{\{m\}})$ equals $-2^m \neq 0$. Additionally, a boolean function has a nonzero linear structure if and only if $|\hat{r}(s)| = 2^m$ for a certain vector s. Hence, f has at least three linear structures, corresponding to the vectors $(0,1_{\{1\}})$, $(0,1_{\{m\}})$ and $(1,\underline{0},1)$. □

We now refine the previous results by relating the linear structures of a generating function to those of the corresponding bipermutive rule.

Lemma 6.9. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a bipermutive function with generating function $g : \mathbb{F}_2^{m-2} \to \mathbb{F}_2$. If $a \in \mathbb{F}_2^{m-2}$ is a linear structure for g, then $(0,a,0)$, $(1,a,0)$, $(0,a,1)$ and $(1,a,1)$ are linear structures for f.*

Proof. We prove only the case $(0,a,0)$: since f is bipermutive, it follows that $\hat{f}(0,a,0) = \hat{f}(1,a,1)$ and $\hat{f}(1,a,0) = \hat{f}(0,a,1) = -\hat{f}(0,a,0)$, thus the remaining three cases can be proved in a similar way.

Let us assume that $a \in \mathbb{F}_2^{m-2}$ is a linear structure for g. We write the autocorrelation function of f with respect to vector $(0,a,0)$, as follows:

$$\begin{aligned}\hat{r}(0,a,0) = &\sum_{x \in \mathbb{F}_2^{m-2}} \hat{f}(0,x,0) \cdot \hat{f}(0,x \oplus a,0) + \sum_{x \in \mathbb{F}_2^{m-2}} \hat{f}(1,x,0) \cdot \hat{f}(1,x \oplus a,0) + \\ &+ \sum_{x \in \mathbb{F}_2^{m-2}} \hat{f}(0,x,1) \cdot \hat{f}(0,x \oplus a,1) + \sum_{x \in \mathbb{F}_2^{m-2}} \hat{f}(1,x,1) \cdot \hat{f}(1,x \oplus a,1) \quad .\end{aligned} \tag{40}$$

Since f is bipermutive, we have the following identities:

$$\hat{f}(0,x,0)=\hat{f}(1,x,1)=\hat{g}(x) \ ; \ \hat{f}(0,x\oplus a,0)=\hat{f}(1,x\oplus a,1)=\hat{g}(x\oplus a) \ ;$$
$$\hat{f}(1,x,0)=\hat{f}(0,x,1)=-\hat{g}(x) \ ; \ \hat{f}(1,x\oplus a,0)=\hat{f}(0,x\oplus a,1)=-\hat{g}(x\oplus a) \ .$$

Thus, equation (40) becomes

$$\begin{aligned}\hat{r}(0,a,0) = & \sum_{x\in\mathbb{F}_2^{m-2}} \hat{g}(x)\cdot\hat{g}(x\oplus a) + \sum_{x\in\mathbb{F}_2^{m-2}} (-\hat{g}(x))\cdot(-\hat{g}(x\oplus a)) + \\ & + \sum_{x\in\mathbb{F}_2^{m-2}} (-\hat{g}(x))\cdot(-\hat{g}(x\oplus a)) + \sum_{x\in\mathbb{F}_2^{m-2}} \hat{g}(x)\cdot\hat{g}(x\oplus a) = \\ = & \ \ 4\cdot\left(\sum_{x\in\mathbb{F}_2^{m-2}} \hat{g}(x)\cdot\hat{g}(x\oplus a)\right) \ .\end{aligned} \tag{41}$$

Since vector $a\in\mathbb{F}_2^{m-2}$ is a linear structure for g, by Proposition 3.13 it follows that

$$\left|\sum_{x\in\mathbb{F}_2^{m-2}} \hat{g}(x)\cdot\hat{g}(x\oplus a)\right| = 2^{m-2} \ . \tag{42}$$

By substituting (42) in (41) we obtain $|\hat{r}(0,a,0)|=2^m$, hence $(0,a,0)$ is a linear structure for f. □

Using Lemma 6.9, we can finally derive the following

Theorem 6.10. *Let $f:\mathbb{F}_2^m\to\mathbb{F}_2$ be a bipermutive function with generating function $g:\mathbb{F}_2^{m-2}\to\mathbb{F}_2$. If g has k linear structures, then f has $3+4k$ linear structures.*

Proof. By Theorem 6.8 we know that all bipermutive functions always have at least three linear structures. Moreover, by Lemma 6.9 if $a\in\mathbb{F}_2^{m-2}$ is a linear structure for g, then the four vectors obtained by surrounding a with the possible combinations of values 0 and 1 are linear structures for f. Globally, these facts mean that if g has k linear structures, then f has *at least* $3+4k$ linear structures.

However, the argument used in the proof of Lemma 6.9 can be easily adapted to prove also that if $s\in\mathbb{F}_2^{m-2}$ is *not* a linear structure for g and s is not null, then the four vectors $(0,s,0)$, $(1,s,0)$, $(0,s,1)$ and $(1,s,1)$ are not linear structures for f as well. In fact, in Lemma 6.9 we proved that for all $s_1, s_m \in \mathbb{F}_2$ the absolute value of the autocorrelation function of (s_1, s, s_m) is 4 times the autocorrelation function of g computed on s. Since $|\hat{r}_g(s)| \neq 2^{m-2}$, it also follows that $|\hat{r}_f(s_1, s, s_m)| \neq 2^m$, thus (s_1, s, s_m) cannot be a linear structure for f. The consequence is that the linear structures of f are exactly the three corresponding to the vectors $(0, 1_{\{1\}})$, $(0, 1_{\{m\}})$ and $(1, \underline{0}, 1)$ plus the $4k$ which can be obtained by juxtaposing the k linear structures of g with the four possible pairings of values $s_1, s_m \in \mathbb{F}_2$ in the leftmost and rightmost coordinates. □

6.5 Application to the Case $r = 3$

We now summarise the various theoretical results about bipermutive rules that we proved in the previous sections. Given a bipermutive rule $f : \mathbb{F}_2^m \to \mathbb{F}_2$ and its generating function $g : \mathbb{F}_2^{m-2} \to \mathbb{F}_2$, we have the following facts:

- The algebraic degree of f equals the degree of g, except when g is the constant function 0. In this case, the algebraic degree of f is 1 (Theorem 6.1).

- The nonlinearity of f is 4 times the nonlinearity of g (Theorem 6.5) .

- Rule f is k-resilient if and only if g is $(k-2)$-resilient (Theorem 6.6). In particular, every bipermutive rule generated by a balanced boolean function is 2-resilient, and every bipermutive rule is 1-resilient (Theorems 4.5 and 6.4).

- If g has k linear structures, then f always has $3 + 4k$ linear structures (Theorem 6.10). Moreover, f never satisfies the SAC (Theorem 6.8).

Hence, the problem of finding bipermutive rules potentially useful for the design of CA-based cryptographic PRNG can be reduced to the search of good generating functions. Since all bipermutive functions do not satisfy the SAC and always have at least three linear structures, their usefulness for CA-based block ciphers is limited. However, statistical evidence suggests that these two criteria should be taken into account also in the generating functions of bipermutive rules used for pseudorandom generation. In fact, looking back to the exhaustive search performed in the case of radius $r = 2$, it turns out that the three rules R01, R07 and R16 passing all the NIST tests are defined by generating functions which are $PC(1)$ and have no linear structures. Thus, a possible strategy is to search for generating functions which satisfy the SAC and minimise the number of linear structures. The resulting bipermutive functions will have the minimum number of linear structures and will satisfy the *Restricted SAC* (or *RSAC*), that is, the Strict Avalanche Criterion computed only on the input vectors $(0,x,0)$, for all $x \in \mathbb{F}_2^{m-2}$.

There is a total of $2^{2^{7-2}} = 4294967296$ bipermutive rules of radius $r = 3$, a number sufficiently limited to allow an exhaustive search in a reasonable time. We can additionally reduce the search space to the set of $\binom{32}{16} = 601080390$ balanced boolean functions in 5 variables, since in this way we can generate only bipermutive rules which are at least 2-resilient.

To completely span this space of functions we used a basic combinatorial algorithm described by Knuth in [9]. The algorithm simply generates all the 32-bit balanced strings (which represent the truth tables of the functions) by starting from the string having all the 1s in the least significant positions, and gradually modifying it by shifting the most significant 1s to the right. The strings are generated in lexicographic order, so the corresponding decimal representations of the functions are listed in their natural order. This algorithm is more efficient in the situations where, considering the binomial coefficient $\binom{n}{k}$, $k \leq n/2$, which is exactly our case. For each 32-bit string f generated, we computed its cryptographic properties by using the mathematical transforms mentioned in Section 3.1. Considering Tarannikov's bound, in the case of radius $r = 3$ the maximum order of resiliency for nonlinear functions is 4, thus we kept only those generating functions which were at least 1-resilient.

We remark the fact that there are more sophisticated and efficient techniques to enumerate all the resilient boolean functions of a certain number of variables, like the one described in [2]. However, the implementation in Java of our combined algorithm (lexicographic generation of balanced strings and computation of their cryptographic properties) took only one hour to explore the entire set of 5-variable balanced boolean functions, using a single core machine with a 1.6 GHz processor. The algorithm returned in total 807980 1-resilient functions in 5 variables. This quantity is coherent with the results reported in [11], where the number of 1-resilient boolean functions is derived by an algebraic method.

Among the resulting functions, we isolated three subsets which satisfied the best available trade-offs among the cryptographic properties considered. In particular, we used the concept of *deviation from* $PC(l)$, originally introduced in [17], to find the functions which featured a minimal deviation from the SAC.

Definition 6.11. Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be a boolean function. For all $l \in \{1, \cdots, m\}$ the *deviation from the propagation criterion* $PC(l)$ of f is defined as

$$pcdev_f(l) = \max\{|\hat{r}(s)| : 1 \leq w_H(s) \leq l\} \ .$$

Clearly, a function f satisfies $PC(l)$ if and only if $pcdev_f(l) = 0$, and $pcdev_f(1)$ is the deviation from the SAC. The details of the selected subsets are reported in Table 4.

Set ID	RES	$PC1$	NL	AD	LS	$\#CARD$
$SET1_G$	1	8	12	3	0	96768
$SET2_G$	1	0	8	3	0	3840
$SET3_G$	2	32	8	2	3	520

Table 4
Cryptographic properties of the selected classes of generating functions in 5 variables. *RES* stands for resiliency order, *PC*1 is the deviation from the SAC, *NL* is the nonlinearity, *AD* the algebraic degree, *LS* the number of linear structures and *#CARD* the cardinality of the class.

From $SET1_G$, $SET2_G$ and $SET3_G$ we built the corresponding classes of bipermutive rules of radius 3, respectively named $SET1_B$, $SET2_B$ and $SET3_B$. Table 5 shows the cryptographic properties of these three classes.

Set ID	RES	NL	AD	LS
$SET1_B$	3	48	3	3
$SET2_B$	3	32	3	3
$SET3_B$	4	32	2	7

Table 5
Cryptographic properties of the generated bipermutive rules of radius 3.

We successively scrutinised the rules of $SET1_B$, $SET2_B$ and $SET3_B$ by means of the ENT and NIST suites, using the same methodology described in Section 5.2. Dealing in this case with thousands of rules, we adopted stricter

Set ID	err_π	$\lvert scc \rvert$	μ_{dev}	$\#CARD$
$SET1_B$	$< 0.1\%$	< 0.001	< 0.001	40
$SET2_B$	$< 1\%$	< 0.001	< 0.001	10
$SET3_B$	$< 1\%$	< 0.01	< 0.001	14

Table 6
Thresholds adopted for the selection of the rules to be subsequently investigated by means of the NIST tests.

criteria than the ones employed for the case of radius $r = 2$ to select the final rules to be tested with the NIST battery. In particular, after having removed from all the sets the rules which failed the Chi-Square test, we opted for the thresholds reported in Table 6. Tables 7, 8 and 9 report the ENT results of the selected bipermutive rules.

We finally subjected the remaining rules in the three sets to the NIST suite. Table 10 reports the results of the tests.

In the group of rules selected from $SET1_B$, four rules passed all the NIST tests (R17, R20, R28, and R30), while among the rules filtered from $SET2_B$ only one passed all of them (rule R40). Class $SET3_B$ is the one which scored the worst results, with rules R45 and R47 having very low pass rates with respect to the proportion of passing sequences (respectively, 26 and 147 out of 187) and no rules which passed all the tests, though the remaining ones all have pass rates close to the maximum.

7 CONCLUSIONS

In this paper we showed that bipermutive rules, besides generating CA which are expansive and mixing chaotic, are also potentially useful for the design of strong cryptographic PRNG. In particular, we proved that all bipermutive rules are also 1-resilient, and we derived an enumerative encoding for bipermutive rules based on a graph representation which groups the 2^m inputs of a boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ in 2^{m-2} connected components. Since by Tarannikov's bound there are no nonlinear and resilient elementary CA rules, we applied this encoding to generate the 256 bipermutive rules of radius 2, and used the mathematical transform discussed in Section 3.1 to check their cryptographic properties, in particular 2-resiliency and high nonlinearity.

Rule - Reflection	E_8	χ^2	μ_{dev}	err_{π}	scc
R17 - R17a	7.976113	0.23	0.000223	0.09%	-0.000582
R18 - R18a	7.975715	0.24	0.000160	0.01%	0.000432
R19 - R19a	7.976280	0.28	0.000589	0.09%	0.000970
R20 - R20a	7.977819	0.54	0.000431	0.09%	0.000485
R21 - R21a	7.975340	0.13	0.000506	0.01%	0.000594
R22 - R22a	7.976285	0.24	0.000274	0.01%	0.000312
R23 - R23a	7.979331	0.82	0.000457	0.01%	0.000488
R24 - R24a	7.977144	0.39	0.000434	0.09%	0.000162
R25 - R25a	7.975499	0.23	0.000472	0.01%	-0.000494
R26 - R26a	7.978890	0.72	0.000241	0.01%	-0.000078
R27 - R27a	7.977617	0.50	0.000831	0.09%	0.000166
R28 - R28a	7.976794	0.37	0.000749	0.09%	-0.000579
R29 - R29a	7.976941	0.34	0.000096	0.09%	0.000419
R30 - R30a	7.978999	0.76	0.000707	0.01%	0.000741
R31 - R31a	7.979656	0.88	0.000336	0.09%	0.000038
R32 - R32a	7.979713	0.88	0.000385	0.01%	0.000423
R33 - R33a	7.976338	0.31	0.000183	0.09%	0.000279
R34 - R34a	7.975078	0.13	0.000138	0.01%	0.000483
R35 - R35a	7.979510	0.85	0.000424	0.01%	0.000776
R36 - R36a	7.975770	0.22	0.000030	0.09%	0.000042

Table 7
ENT tests results on the pseudorandom sequences generated by the final 40 rules of radius 3 selected from $SET1_B$.

Rule - Reflection	E_8	χ^2	μ_{dev}	err_π	scc
R37 - R37a	7.975941	0.19	0.000720	0.57%	0.000781
R38 - R38a	7.976647	0.31	0.000930	0.46%	0.000075
R39 - R39a	7.976391	0.31	0.000397	0.10%	0.000887
R40 - R40a	7.978808	0.73	0.000918	0.94%	0.000012
R41 - R41a	7.978865	0.30	0.000364	0.29%	-0.000043

Table 8
ENT tests results on the pseudorandom sequences generated by the final 10 rules of radius 3 selected from $SET2_B$.

Rule - Reflection	E_8	χ^2	μ_{dev}	err_π	scc
R42 - R42a	7.976535	0.33	0.000434	0.46%	0.004211
R43 - R43a	7.978922	0.72	0.000542	0.83%	0.005450
R44 - R44a	7.979220	0.80	0.000677	0.29%	0.001004
R45 - R45a	7.976311	0.27	0.000107	0.74%	-0.002056
R46 - R46a	7.977311	0.43	0.000344	0.65%	-0.005901
R47 - R47a	7.978972	0.78	0.000543	0.65%	-0.002483
R48 - R48a	7.979771	0.89	0.000711	0.09%	-0.006638

Table 9
ENT tests results on the pseudorandom sequences generated by the final 14 rules of radius 3 selected from $SET3_B$.

Rule	P-value	Proportion
R17	187/187	187/187
R18	186/187	185/187
R19	187/187	186/187
R20	187/187	187/187
R21	186/187	186/187
R22	186/187	186/187
R23	186/187	186/187
R24	186/187	186/187
R25	187/187	186/187
R26	187/187	186/187
R27	186/187	187/187
R28	187/187	187/187
R29	187/187	186/187
R30	187/187	187/187
R31	187/187	186/187
R32	187/187	186/187
R33	186/187	186/187
R34	187/187	186/187
R35	186/187	186/187
R36	186/187	187/187
R37	187/187	186/187
R38	186/187	185/187
R39	187/187	186/187
R40	187/187	187/187
R41	187/187	185/187
R42	187/187	186/187
R43	187/187	185/187
R44	186/187	187/187
R45	187/187	26/187
R46	187/187	186/187
R47	187/187	147/187
R48	186/187	184/187

Table 10
NIST tests results on the pseudorandom sequences generated by the 32 final bipermutive rules of radius 3 selected from $SET1_B$, $SET2_B$ and $SET3_B$.

We successively tested the resulting 56 nonlinear and 2-resilient rules with two batteries of statistical randomness tests, the ENT suite and the NIST suite. We used the former to discard the rules which did not generate good pseudorandom sequences of 2^{16} bits, and the latter to investigate more thoroughly the remaining 16 rules by sequences of 10^6 bits, taking in both phases the results obtained by rule 30 as a benchmark. The final results showed that rules R01, R07 and R16 passed all the 187 NIST tests.

We continued our theoretical investigation on bipermutive rules by showing how their algebraic degree, nonlinearity and order of resiliency can be determined by using the corresponding *generating functions*, which are derived by applying the Shannon decomposition on the leftmost and rightmost variables. Moreover, we observed that the truth tables of the generating functions can be easily characterised by the labelling of the graph-based encoding mentioned above.

We successively used these theoretical results to reduce the problem of finding good bipermutive rules for cryptographic CA-based PRNG to the search of their generating functions, and in particular to the optimisation of their cryptographic properties. Even if bipermutive rules never satisfy the SAC and always have at least three linear structures (and thus they are not good candidates for a CA-based block cipher), these two properties should be considered also in the context of pseudorandom generation, since the best rules of radius $r = 2$ which passed the NIST tests had generating functions which satisfied the SAC and had no linear structures.

In the case of bipermutive rules of radius $r = 3$, the associated set of generating functions is sufficiently limited to allow an exhaustive search, which we performed by running a combinatorial algorithm devised by Knuth [9] on the space of balanced boolean functions in 5 variables. We then selected three subsets of generating functions satisfying the best trade-offs among the considered cryptographic properties, and we investigated them by means of the ENT and NIST tests as well. We found that five rules (R17, R20, R28, R30 and R40) passed all the tests.

Considering that the elementary rule 30 passed all the NIST tests as well, we can thus reasonably conclude that these eight rules of radius 2 and 3 are at least as good as rule 30 for pseudorandom number generation, and moreover they satisfy an additional stronger definition of chaos (E-chaos and M-chaos) and good trade-offs among the cryptographic properties we considered in this paper. In particular, all these rules are at least 2-resilient, and they reach both Siegenthaler's and Tarannikov's bounds, with the exception of rule R40.

In any case, it is important to remark that the cryptographic properties considered in this paper are not sufficient to verify the robustness of a CA-based PRNG using local rules satisfying them. As a matter of fact, there are other properties of cryptographic boolean functions which we did not analyse in this paper, such as *algebraic immunity*, described in [1]. Further research is also needed to assess whether the fact that bipermutive rules are not good with respect to the autocorrelation-related properties (i.e. the propagation criterion and the number of linear structures) can be used to attack a CA-based PRNG.

The enumerative encoding described in Section 4.1 gives an effective mean to explore the spaces of rules having higher radii. The interest in doing such kind of search is twofold. The first motivation is practical: it is intuitive to think that, as the radius of the rules increases, the diffusion of a CA-based PRNG gets better. The second reason which motivates the exploration of rules with higher radii is to test conjectures about the aforementioned cryptographic properties, by finding counterexamples.

For all radii $r \geq 4$, however, the set of possible bipermutive rules is so large that heuristic methods would be necessary to efficiently visit the search space. For example, we observe that it would be straightforward to apply our enumerative encoding to evolve bipermutive rules by means of genetic algorithms, such as the one described by Millan, Clark and Dawson in [17].

REFERENCES

[1] Carlet, C.: Boolean Functions for Cryptography and Error-Correcting Codes. In: Crama, Y., Hammer, P.L. (eds.) Boolean Models and Methods in Mathematics, Computer Science, and Engineering. Cambridge University Press, New York (2010)

[2] Carrasco, N., Le Bars, J.-M., Viola, A.: Enumerative Encoding of Correlation-Immune Boolean Functions. Theor. Comput. Sci. 487, 23-36 (2013)

[3] Cattaneo, G., Finelli, M., Margara, L.: Investigating Topological Chaos by Elementary Cellular Automata Dynamics. Theor. Comput. Sci. 244(1-2), 219-244 (2000)

[4] Cattaneo, G., Dennunzio, A., Margara, L.: Chaotic Subshifts and Related Languages Applications to One-Dimensional Cellular Automata. Fundam. Inform. 52(1-3), 39-80 (2002)

[5] Chepyzhov, V.V., Smeets, B.J.M.: On a Fast Correlation Attack on Certain Stream Ciphers. In: Davis, D.W. (ed.) EUROCRYPT '91. LNCS, vol. 547, pp. 176-185. Springer, Heidelberg (1992)

[6] Devaney, R.L.: An Introduction to Chaotic Dynamical Systems. Addison-Wesley, Reading (1989)

[7] Evertse, J.H.: Linear Structures in Block Ciphers. In: Chaum, D., Price, W.L. (eds.) EUROCRYPT '87. LNCS vol. 304, pp. 249-266. Springer, Heidelberg (1988)

[8] Gutowitz, H.: Cryptography with Dynamical Systems. In: Goles, E., Boccara, N. (eds.) Cellular Automata and Cooperative Phenomena. pp. 237–274. Kluwer Academic Press (1993)

[9] Knuth, D.E.: The Art of Computer Programming, vol. 4a: Combinatorial Algorithms, Part 1. Addison-Wesley, Reading (2011)

[10] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)

[11] Le Bars, J.-M., Viola, A.: Equivalence Classes of Boolean Functions for First-Order Correlation. IEEE Trans. Inf. Theory 56(3), 1247-1261 (2010)

[12] Leporati, A., Mariot, L.: 1-Resiliency of Bipermutive Cellular Automata Rules. In: Kari, J., Kutrib, M., Malcher, A. (eds.) AUTOMATA 2013. LNCS, vol. 8155, pp. 110-123. Springer, Heidelberg (2013)

[13] Logachev, O.A., Salnikov, A.A., Yashchenko, V.V.: Boolean Functions in Coding Theory and Cryptography. American Mathematical Society, Providence (2012)

[14] Martin, B.: A Walsh Exploration of Elementary CA Rules. J. Cell. Aut. 3(2), 145-156 (2008)

[15] Meier, W., Staffelbach, O.: Analysis of Pseudo Random Sequences Generated by Cellular Automata. In: Davies, D.W. (ed.) EUROCRYPT '91. LNCS, vol. 547, pp. 186-200. Springer, Heidelberg (1991)

[16] Massey, J.L.: Shift-Register Synthesis and BCH Decoding. IEEE Trans. Inf. Theory 15(1), pp. 122-127 (1969)

[17] Millan, W., Clark, A., Dawson, E.: Heuristic Design of Cryptographically Strong Balanced Boolean Functions. In: Nyberg, K. (ed.) EUROCRYPT '98. LNCS, vol. 1403, pp. 489-499. Springer, Heidelberg (1998)

[18] National Institute of Standards and Technology: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. Special Publication 800-22, Revision 1a (2010)

[19] Preneel, B., Van Leekwijck, W., Van Linden, L., Govaerts, R., Vandewalle, J.: Propagation Characteristics of Boolean Functions. In: Damgård, I. (ed.) EUROCRYPT '90. LNCS, vol. 473, pp. 161-173. Springer, Heidelberg (1991)

[20] Rueppel, R.A., Staffelbach, O.: Products of Linear Recurring Sequences with Maximum Complexity. IEEE Trans. Inf. Theory 33(1), pp. 124-131 (1987)

[21] Shannon, C.: The Synthesis of Two-Terminal Switching Circuits. Bell Syst. Tech. J. 28(1), 59-98 (1949)

[22] Siegenthaler, T.: Correlation Immunity of Nonlinear Combining Functions for Cryptographic Applications. IEEE Trans. Inf. Theory 30(5), 776–780 (1984)

[23] Siegenthaler, T.: Decrypting a Class of Stream Ciphers Using Ciphertext Only. IEEE Trans. Comput. C-34(1), 81-85 (1985)

[24] Tarannikov. Y.V.: On Resilient Boolean Functions with Maximum Possible Nonlinearity. In: Roy, B.K., Okamoto, E. (eds.) INDOCRYPT 2000. LNCS, vol. 1977, pp. 19-30.

[25] Ulam, S.: Random Processes and Transformations. Proc. Int. Congress of Math. 2, 264-275 (1952)

[26] Von Neumann, J.: Theory of Self-Reproducing Automata. University of Illinois Press, Champaign (1966)

[27] Walker, J.: ENT Test Suite. `http://www.fourmilab.ch/random`

[28] Webster, A.F., Tavares, S.E.: On the Design of S-Boxes. In: Williams, H.C. (ed.) CRYPTO '85. LNCS, vol. 218, pp. 523-534. Springer, Heidelberg (1985)

[29] Wolfram, S.: Statistical Mechanics of Cellular Automata. Rev. Mod. Phys. 55(3), 601-644 (1983)

[30] Wolfram, S.: Random Sequence Generation by Cellular Automata. Adv. Appl. Math. 7(2), 123-169 (1986)

[31] Xiao, G.-Z., Massey, J.L.: A Spectral Characterization of Correlation immune Combining Functions. IEEE Trans. Inf. Theory 34(3), 569-571 (1988)