1-Resiliency of Bipermutive Cellular Automata Rules

Alberto Leporati and Luca Mariot

Dipartimento di Informatica, Sistemistica e Comunicazione,
Università degli Studi Milano - Bicocca,
Viale Sarca 336/14, 20124 Milano, Italy
alberto.leporati@unimib.it, l.mariot@campus.unimib.it

Abstract. It is known that CA rules which are both leftmost and rightmost permutive (bipermutive rules) are expansively and mixing chaotic. In this paper, we prove that bipermutive rules also satisfy the condition of 1-resiliency (that is, balancedness and first order correlation-immunity), which is an important property used in the design of pseudorandom number generators for cryptographic purposes. We thus derive an enumerative encoding for bipermutive rules based on a graph representation, and we use it to generate all the 256 bipermutive rules of radius 2. Among these rules we select the ones which satisfy additional cryptographic properties: high nonlinearity and 2-resiliency. Finally, we assess the quality of the pseudorandom sequences generated by these remaining rules with the ENT and NIST statistical test suites, taking the elementary rule 30 as a benchmark.

Keywords: Cellular automata, boolean functions, pseudorandom number generators, stream ciphers, deterministic chaos, permutivity, resiliency, nonlinearity, Walsh transform, ENT test suite, NIST test suite.

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 [13], 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. Unfortunately, even if rule 30 is nonlinear and balanced, and even if it is chaotic with respect to Devaney's definition of topological chaos [4], it does not satisfy the property of *first order correlation-immunity*, introduced by Siegenthaler in [8]. More generally, Martin has pointed out in [6] 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 [2] that *bipermutive* rules (that is, rules which are both leftmost and rightmost permutive) are expansively chaotic, while in [3] 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.

J. Kari, M. Kutrib, and A. Malcher (Eds.): AUTOMATA 2013, LNCS 8155, pp. 110–123, 2013.

The aim of this paper is to study the class of bipermutive rules with respect to the cryptographic property of *resiliency*, which includes balancedness and correlation-immunity. In particular, we prove 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 (which is, by Tarannikov's bound [10], the best possible trade-off between these two properties in the case of boolean functions of 5 variables). We successively filter out the rules which do not generate sequences of 2^{16} bits that pass the statistical tests from the ENT suite, using rule 30 as a benchmark. Finally, we apply the more stringent NIST test suite to longer sequences (10^6 bits) produced by the remaining rules, observing that three of them pass all the tests, like rule 30.

The rest of this paper is organized as follows. Section 2 recalls basic definitions and theoretical results about cellular automata and the properties of nonlinearity and m-resilience a CA rule should satisfy for cryptographic applications. Section 3, after a brief introduction to topological chaos in CAs and permutive rules, reports the main theoretical contribution of the paper, namely the proof that bipermutive rules are also 1-resilient. Section 4 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. Section 5 reports the results of the statistical tests of the ENT and NIST suites applied to the pseudorandom sequences generated by the rules found in Section 4. Finally, Section 6 sums up the results presented throughout the paper, and points out some possible future developments and improvements on the subject.

2 Cellular Automata and Cryptographic Properties of Boolean Functions

2.1 Cellular Automata

Cellular automata are a particular type of discrete dynamical systems, 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 1. *A* finite one-dimensional cellular automaton *is a 4-tuple* $\langle n, A, r, f \rangle$ *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 by 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 with 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 [12], 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 [13] 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 central 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.

2.2 Cryptographic 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. An excellent reference for cryptographic boolean functions is [1].

A *boolean function* in m variables is a mapping $f : \mathbb{F}_2^m \to \mathbb{F}_2$, which in the following we will identify by the 2^m-bit string representing its truth table. 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 $\hat{f}(x) = (-1)^{f(x)}$. 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 called *balanced* if $|f^{-1}(0)| = |f^{-1}(1)| = 2^{m-1}$. Unbalanced functions are generally not desirable in cryptographic applications, since they present a statistical bias which can be exploited for linear and differential cryptanalysis.

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

Definition 2. *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: $\forall \omega \in \mathbb{F}_2^m$*

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

The value $\hat{F}(\omega)$ is also called the *Walsh coefficient* of f with respect to the vector ω. 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 1. Denoting by 0 the null vector of $\mathbb{F}_2^m$, it follows that $\hat{F}(0) = \sum_{x \in \mathbb{F}_2^m} \hat{f}(x)$.

Property 2. From Property 1, it is obvious that a function f is balanced if and only if $\hat{F}(0) = 0$.

Property 3. 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. Thus in this case the sign of the generic term in (1) is uniquely determined by the value of x_i.

Given a boolean function f, the maximum absolute value of its Walsh coefficients, $W_{\max}(f)$, is called the *spectral radius* of f. The spectral radius is useful to characterise the *nonlinearity* of a boolean function, which is formally defined as the Hamming distance from the set of affine functions: in [1] it is shown that given a boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ its nonlinearity is $Nl(f) = 2^{-1}(2^m - W_{\max}(f))$. In the design of stream or block ciphers, the nonlinearity of the boolean functions selected should be as high as possible, since it provides better confusion.

A second important property for cryptographic boolean functions is correlation-immunity, introduced by Siegenthaler in [8]. A boolean function f is said to be *k-th order correlation-immune* if the restrictions of f obtained by fixing k input coordinates of f all have the same Hamming weight. If a function used in a stream cipher does not satisfy this property, it is possible to apply a divide-and-conquer *correlation attack* described in [9] using k Linear Feedback Shift Registers, in order to recover the plaintext. A function which is both balanced and k-th order correlation-immune is called *k-resilient*. Xiao and Massey proved in [14] a necessary and sufficient condition for a boolean function to be k-th order correlation-immune, using its Walsh Transform.

Theorem 1. *A boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is k-th order correlation-immune if and only if, $\forall \omega \in \mathbb{F}_2^m$ such that $1 \leq w_{\mathrm{H}}(\omega) \leq k$, $\hat{F}(\omega) = 0$.*

Hence, in order to verify whether a given boolean function is k-resilient, by Property 2 and Theorem 1 it suffices to check that its Walsh Transform vanishes for all the input vectors having Hamming weight less than or equal to k, including the null vector.

The three properties of balancedness, nonlinearity and k-th order correlation-immunity induce a trade-off; in particular, Tarannikov [10] showed an upper bound on the maximum nonlinearity obtainable in k-resilient functions (with $k \leq m-2$), which is $2^{m-1} - 2^{k+1}$.

2.3 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), so it is possible to verify if it is suitable to design a CA-based PRNG by checking its balancedness, nonlinearity and correlation-immunity. 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 [6] 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 nonlinearity for 1-resilient functions is $2^{3-1} - 2^{1+1} = 0$. The consequence is that elementary CA rules are not adequate for building a cryptographic PRNG or a stream cipher, so it is necessary to explore the spaces of rules having higher radii.

3 Bipermutive CA Rules

3.1 Symbolic Dynamics and Topological Chaos in Cellular Automata

The dynamics of one-dimensional CAs 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 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

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

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 formalized in several rigorous definitions in the literature of dynamical systems. The most popular among them is perhaps the definition given by Devaney in [4], which uses a topological approach.

Definition 3. *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 $U, V \subset X$, $\exists t \in \mathbb{N}$ 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 $\forall x \in X$, $\forall \delta > 0$, $\exists y \in X$ with $d(x,y) < \delta$ and $\exists t \in \mathbb{N}$ such that $d(F^t(x), F^t(y)) \geq \varepsilon$.*

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 [2] substitutes sensitivity to initial conditions by *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) [3] topological transitivity is replaced by *topological mixing*: for all nonempty open subsets $U, V \subset X$, $\exists t \in \mathbb{N}$ such that $\forall s \geq t$, $F^s(U) \cap V \neq \emptyset$.

3.2 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. *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 results that*

$$f(x,0_{\{i\}}) \neq f(x,1_{\{i\}}) \ . \tag{3}$$

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

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

3.3 Resiliency of Bipermutive Rules

We can now prove the following property: bipermutive rules, besides the chaotic behaviour they induce in CAs, 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 1. *If $f : \mathbb{F}_2^m \to \mathbb{F}_2$ is i-permutive, then f is balanced.*

Proof. Considering Property 1, we rewrite the Walsh Transform of the null vector as follows:

$$\hat{F}(0) = \sum_{\{x \in \mathbb{F}_2^m :\, x_i = 0\}} \hat{f}(x) \ + \sum_{\{x \in \mathbb{F}_2^m :\, x_i = 1\}} \hat{f}(x) \ . \tag{4}$$

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 (4) *is exactly the opposite of the first sum, and $\hat{F}(0) = 0$. By Property 2, this means that f is balanced.* □

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

Lemma 2. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2$ be bipermutive. Then f is first order correlation-immune.*

Proof. Using the characterization of correlation-immunity given in Theorem 1, it is sufficient to show that $\forall \omega \in \mathbb{F}_2^m$ such that $w_{\mathrm{H}}(\omega) = 1$, $\hat{F}(\omega) = 0$. Let ω be a generic

vector having Hamming weight 1. By Property 3, the Walsh Transform of ω *can be computed as*

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^m} \hat{f}(x) \cdot (-1)^{x_i} \ . \tag{5}$$

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* (5) *as follows:*

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 0_{\{m\}}) \cdot (-1)^{x_i} \ + \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 1_{\{m\}}) \cdot (-1)^{x_i} \tag{6}$$

where $i \in \{1,...,m-1\}$*. Since* f *is R-permutive,* $\hat{f}(x, 1_{\{m\}}) = -\hat{f}(x, 0_{\{m\}})$*. Moreover, since in* (6) 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 given by*

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^m} \hat{f}(x) \cdot (-1)^{x_m} \ . \tag{7}$$

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* (6) *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* (7) *in the following way:*

$$\hat{F}(\omega) = \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 0_{\{1\}}) \cdot (-1)^{x_m} \ + \sum_{x \in \mathbb{F}_2^{m-1}} \hat{f}(x, 1_{\{1\}}) \cdot (-1)^{x_m} \ . \tag{8}$$

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*

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

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

By combining Lemmas 1 and 2, we finally get the following

Theorem 3. *Let* $f : \mathbb{F}_2^m \to \mathbb{F}_2$ *be a bipermutive boolean function. Then,* f *is 1-resilient.*

4 Generating Bipermutive Rules of a Given Radius

Theorem 3 motivates the search for bipermutive boolean functions to be used in CA-based PRNGs, 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, in particular, high nonlinearity and higher order of resiliency. We propose a simple enumerative encoding which allows us to represent a bipermutive function $f : \mathbb{F}_2^m \to \mathbb{F}_2$ as a string of 2^{m-2} bits, and then we apply it to exhaustively explore the set of bipermutive boolean functions defined on 5 variables.

4.1 An Enumerative Encoding for Bipermutive Functions

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, and $E \subseteq V \times V$ is the set of edges 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

$$(x_1 = \bar{y}_1 \wedge (\forall i \in \{2, \cdots, m\}\, x_i = y_i)) \bigvee (x_m = \bar{y}_m \wedge (\forall i \in \{1, \cdots, m-1\}\, x_i = y_i)) \ ,$$

where $\bar{y}_j$ 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 4. 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 $i \in \{2, \cdots, m\}$. 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 $i \in \{1, \cdots, m-1\}$.

Property 5. 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 $i \in \{2, \cdots, m-1\}$. 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 $i \in \{2, \cdots, m\}$ and $x_m = \bar{x}''_m$, $x_i = x''_i$ for all $i \in \{1, \cdots, m-1\}$. Then, $x'_1 = y_1$ and $x'_i = y_i$ for all $i \in \{2, \cdots, m-1\}$. 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 6. Since the relation which defines E is symmetric, from Property 5 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 $(x, y) \in \mathbb{F}_2^m$ of vectors which differ in the leftmost and rightmost coordinates and are equal in the $m-2$ central ones. Thus G is composed by 2^{m-2} disjoint connected components of this kind.

A boolean function $f \in \mathcal{F}_m$ can be represented as 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 6, 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 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 $f : \mathbb{F}_2^m \to \mathbb{F}_2$ 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.

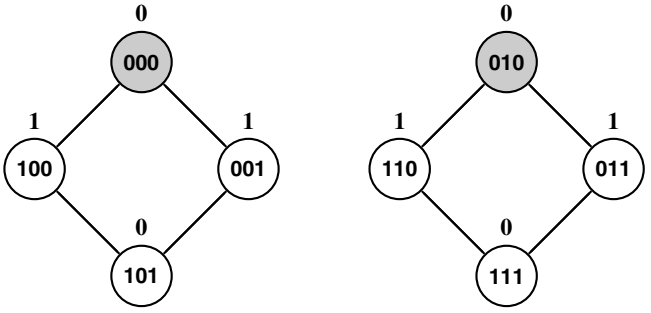

Fig. 1. Representation of the function 01011010 (rule 90) on the graph G. The representatives are shaded in gray, so this function corresponds to the string $c = 00$.

4.2 Application to the Case $r = 2$

It has already been observed that in the case of elementary CAs ($r = 1$) there are only four bipermutive rules which are all linear. We have thus used the enumerative encoding described in Section 4.1 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 in 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 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 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 functions. For each bipermutive rule generated by our algorithm, we computed its nonlinearity and checked if it

was 2-resilient by using the Fast Walsh Transform. It turned out that all the rules were either nonlinear with $Nl = 8$ or linear. We thus selected the rules which were nonlinear and 2-resilient, since they can resist to second order correlation attacks. This left us, in total, with 56 rules.

5 Statistical Randomness Tests

Nonlinearity, resiliency and bipermutivity are not sufficient conditions to make a CA rule suitable for the design of a cryptographic PRNG: for this reason, we subjected the 56 2-resilient nonlinear bipermutive rules discovered by the method discussed in Section 4 to a series of statistical tests, in order to find which of them generate pseudorandom sequences 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 [11], using rule 30 as a benchmark. Then, we applied the NIST test suite [7] 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 by $n = 64$ cells (since 64 bits is a common value for the length of the seed in many standard PRNGs, like ANSI X9.17) and we sampled the trace of the 32nd cell to generate the pseudorandom sequences.

5.1 ENT Tests Results

The ENT Test Suite, assembled by Walker and described in [11], 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 [5], 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 [11], 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 couples. This fact was expected, since in each couple the rules are related by the *reflexive* transformation, mentioned in [6]. Given a binary vector $x \in \mathbb{F}_2^m$, with $x = (x_1, \cdots, x_m)$, the *mirror image* of x is defined as the vector $x_M = (x_m, \cdots, x_1)$. The *reflex* of a boolean function

$f : \mathbb{F}_2^m \to \mathbb{F}_2$ is the function f_R defined as $f_R(x) = f(x_M)$, $\forall x \in \mathbb{F}_2^m$. This transformation preserves the nonlinearity and resiliency of a function, since the spectral radius remains unaltered, and the Hamming weight of an input vector is the same as that of its mirror image. The remaining four rules not coupled are self-reflexive, that is $f_R(x) = f(x)$.

Considering our method of pseudorandom generation described earlier, 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 1 shows the ENT results of the 28 final rules, grouped by reflection couples. The results of rule 30 are also reported for comparison.

Table 1. 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 normalized 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.

Rule - Reflected rule	E_8	χ^2	μ_{dev}	err_π	*scc*
1452976485 - 1717213605	7.979592	0.83	0.004848	0.37%	-0.002338
1453762905 - 1701485205	7.977838	0.56	0.008593	0.66%	0.002280
1453959510 - 1718196630	7.979487	0.85	0.000567	0.37%	-0.003930
1500161685 - 1516676505	7.978750	0.69	0.004215	0.75%	0.003161
1503307365 - 1784059305	7.976643	0.30	0.003097	0.01%	-0.012526
1516873110 (self-reflexive)	7.977783	0.57	0.003332	0.10%	0.003791
1520018790 - 1784255910	7.976146	0.32	0.001983	0.01%	0.015071
1705417305 (self-reflexive)	7.979135	0.82	0.006708	0.09%	0.001310
1705613910 - 1722128730	7.976625	0.34	0.008589	0.18%	0.017063
1772459610 (self-reflexive)	7.976147	0.27	0.004326	0.38%	0.002607
2509924965 - 2790676905	7.977823	0.52	0.005322	0.38%	-0.013957
2510907990 - 2791659930	7.976643	0.30	0.005385	0.55%	-0.025343
2526636390 - 2790873510	7.978825	0.73	0.000548	0.10%	-0.005077
2573821590 - 2590336410	7.978674	0.76	0.008456	0.57%	0.013556
2589549990 (self-reflexive)	7.979135	0.82	0.000952	0.75%	-0.010592
2778290790 - 2794805610	7.978866	0.83	0.007370	0.66%	0.011000
30 (benchmark)	7.979031	0.80	0.004169	0.66%	-0.013926

5.2 NIST Tests Results

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 [7] to longer generated sequences. For each couple 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 [7].

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

$$mpr = \hat{p} - 3\sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \quad , \tag{9}$$

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 [7]. In particular, for each rule we generated a sample of $N = 1000$ pseudorandom sequences of length $l = 10^6$ bits. The 1000 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 2 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 1503307365, the worst results are

Table 2. 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.

Rule	Approach 1	Approach 2
1452976485	187/187	187/187
1453762905	186/187	186/187
1453959510	186/187	187/187
1500161685	184/187	186/187
1503307365	37/187	187/187
1516873110 (self-reflexive)	94/187	184/187
1520018790	187/187	187/187
1705417305 (self-reflexive)	25/187	186/187
1705613910	185/187	187/187
1772459610 (self-reflexive)	24/187	187/187
2509924965	186/187	187/187
2510907990	129/187	186/187
2526636390	187/187	186/187
2573821590	186/187	186/187
2589549990 (self-reflexive)	25/187	185/187
2778290790	187/187	187/187
30 (benchmark)	187/187	187/187

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 the pseudorandom generation method of Section 5.1 (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 (1452976485, 1520018790 and 2778290790) pass all the tests with respect to both approaches, like rule 30. One could thus reasonably conclude that these three rules 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 2-resiliency.

6 Conclusions

In this paper we showed that bipermutive rules, besides generating CAs which are expansive and mixing chaotic, are also 1-resilient, and thus potentially useful for the design of strong cryptographic PRNGs. We also 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} disjoint connected components. Since it is already known by Tarannikov's bound that among the elementary CA rules there are no nonlinear resilient rules, we applied this encoding to generate the 256 bipermutive rules of radius 2, and used the Fast Walsh Transform to compute their nonlinearities and check whether they were 2-resilient.

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 1452976485, 1520018790 and 2778290790 passed all the 187 NIST tests, like rule 30.

It is important to remark, however, that although these three rules are chaotic, 2-resilient, nonlinear, and generate good pseudorandom sequences, they cannot be used *alone* in the design of either a cryptographic PRNG or a stream cipher. In fact, there are many other properties of cryptographic boolean functions, described in [1], which we did not consider in this paper: propagation criterion, algebraic degree and algebraic immunity are some of the most important ones. An interesting direction for future research is thus to study the class of bipermutive rules with respect to these additional properties.We also saw that there are no bipermutive rules of radius 2 reaching the maximum nonlinearity allowed by Tarannikov's bound, even if they are not 2-resilient. Further investigation is needed to verify whether bipermutivity induces a stricter bound on the nonlinearity achievable by a boolean function.

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.

In the case of $r = 3$ and $r = 4$ there are $2^{2^5} = 4294967296$ and $2^{128} \approx 3.4 \cdot 10^{38}$ bipermutive rules, respectively; an exhaustive exploration as we did for $r = 2$ is thus infeasible. However, these search spaces could be reduced by improving our encoding in order to enumerate only those rules which are 2-resilient and highly nonlinear, using the Shannon decomposition. This approach will be pursued in future research. For all radii $r > 4$, instead, the set of possible rules is so large that heuristic methods would be necessary to efficiently visit the search space, even under the new encoding. For example, we observe that it would be straightforward to apply our enumerative encoding to evolve bipermutive rules by means of genetic algorithms.

Acknowledgements. We thank the anonymous referees for their suggestions on how to improve the current paper. This research was partially supported by Università degli Studi di Milano-Bicocca, Fondo di Ateneo (FA) 2011.

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. Cattaneo, G., Finelli, M., Margara, L.: Investigating Topological Chaos by Elementary Cellular Automata Dynamics. Theor. Comput. Sci. 244(1-2), 219–244 (2000)
3. 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)
4. Devaney, R.L.: An Introduction to Chaotic Dynamical Systems. Addison-Wesley, Reading (1989)
5. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
6. Martin, B.: A Walsh Exploration of Elementary CA Rules. J. Cell. Aut. 3(2), 145–156 (2008)
7. 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)
8. Siegenthaler, T.: Correlation-Immunity of Nonlinear Combining Functions for Cryptographic Applications. IEEE Trans. Inf. Theory 30(5), 776–780 (1984)
9. Siegenthaler, T.: Decrypting a Class of Stream Ciphers Using Ciphertext Only. IEEE Trans. Comput. C-34(1), 81–85 (1985)
10. Tarannikov, Y.V.: On Resilient Boolean Functions with Maximal Possible Nonlinearity. In: Roy, B., Okamoto, E. (eds.) INDOCRYPT 2000. LNCS, vol. 1977, pp. 19–30. Springer, Heidelberg (2000)
11. Walker, J.: ENT Randomness Test Suite, `http://www.fourmilab.ch/random/`
12. Wolfram, S.: Statistical Mechanics of Cellular Automata. Rev. Mod. Phys. 55(3), 601–644 (1983)
13. Wolfram, S.: Random Sequence Generation by Cellular Automata. Adv. Appl. Math. 7(2), 123–169 (1986)
14. Xiao, G.-Z., Massey, J.L.: A Spectral Characterization of Correlation-Immune Combining Functions. IEEE Trans. Inf. Theory 34(3), 569–571 (1988)