

## Search space reduction of asynchrony immune cellular automata

Mariot, Luca; Manzoni, Luca; Dennunzio, Alberto

**DOI**

[10.1007/s11047-020-09788-1](https://doi.org/10.1007/s11047-020-09788-1)

**Publication date**

2020

**Document Version**

Final published version

**Published in**

Natural Computing

**Citation (APA)**

Mariot, L., Manzoni, L., & Dennunzio, A. (2020). Search space reduction of asynchrony immune cellular automata. *Natural Computing*, 19(2), 287-293. <https://doi.org/10.1007/s11047-020-09788-1>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.



# Search space reduction of asynchrony immune cellular automata

Luca Mariot<sup>1</sup> · Luca Manzoni<sup>2</sup> · Alberto Dennunzio<sup>3</sup>

Published online: 27 April 2020  
© The Author(s) 2020

## Abstract

We continue the study of asynchrony immunity in cellular automata (CA), which can be considered as a generalization of correlation immunity in the case of vectorial Boolean functions. The property could have applications as a countermeasure for side-channel attacks in CA-based cryptographic primitives, such as S-boxes and pseudorandom number generators. We first give some theoretical results on the properties that a CA rule must satisfy in order to meet asynchrony immunity, like central permutivity. Next, we perform an exhaustive search of all asynchrony immune CA rules of neighborhood size up to 5, leveraging on the discovered theoretical properties to greatly reduce the size of the search space.

**Keywords** Cellular automata · Cryptography · Asynchrony immunity · Correlation immunity · Nonlinearity · Side-channel attacks · Permutivity

## 1 Introduction

In the last years, research about cryptographic applications of cellular automata (CA) focused on the properties of the underlying local rules (Martin 2006; Loporati and Mariot 2014; Formenti et al. 2014). In fact, designing a CA-based cryptographic primitive using local rules that are not highly nonlinear and correlation immune could make certain attacks more efficient.

The aim of this paper is to investigate a new property related to *asynchronous CA* called *asynchrony immunity* (AI), which could be of interest in the context of side-channel attacks. This property can be described by a three-

move game between a user and an adversary. Let  $\ell, r, m \in \mathbb{N}$ ,  $n = m + \ell + r$  and  $t \leq m$ . The game works as follows:

1. The user chooses a local rule  $f: \mathbb{F}_2^{\ell+r+1} \rightarrow \mathbb{F}_2$  of memory  $\ell$  and anticipation  $r$ .
2. The adversary chooses  $j \leq t$  cells of the CA in the range  $\{0, \dots, m-1\}$ .
3. The user evaluates the output distribution  $D$  of the CA  $F: \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  and the distribution  $\tilde{D}$  of the asynchronous CA  $\tilde{F}: \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  where the  $j$  cells selected by the adversary are not updated.
4. *Outcome*: if both  $D$  and  $\tilde{D}$  equals the uniform distribution, the user wins. Otherwise, the adversary wins.

A cellular automaton rule  $f: \mathbb{F}_2^{\ell+r+1} \rightarrow \mathbb{F}_2$  is called  $(t, n)$ -asynchrony immune if, for every subset  $I$  of at most  $t$  cells both the asynchronous CA  $\tilde{F}: \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  resulting from not updating on the subset  $I$  of cells and the corresponding synchronous CA  $F: \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  are balanced, that is, the cardinality of the counterimage of each  $m$ -bit configuration equals  $2^{\ell+r}$ . Thus, asynchrony immune CA rules represent the winning strategies of the user in the game described above.

Notice the difference between the asynchrony immunity game and the  $t$ -resilient functions game (Chor et al. 1985): in the latter, generic vectorial Boolean functions  $F: \mathbb{F}_2^n \rightarrow$

✉ Luca Mariot  
l.mariot@tudelft.nl

Luca Manzoni  
lmanzoni@units.it

Alberto Dennunzio  
dennunzio@disco.unimib.it

<sup>1</sup> Cyber Security Research Group, Delft University of Technology, Mekelweg 2, Delft, The Netherlands

<sup>2</sup> Dipartimento di Matematica e Geoscienze, Università degli Studi di Trieste, Via Valerio 12/1, 34127 Trieste, Italy

<sup>3</sup> Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Viale Sarca 336/14, 20126 Milan, Italy

$\mathbb{F}_2^m$  are considered instead of cellular automata, and the adversary selects both values and positions of the  $t$  input variables.

The side-channel attack model motivating our work is the following. Suppose that a CA of length  $n$  is used as an S-box in a block cipher, and that an attacker is able to inject *clock faults* by making  $t$  cells not updating. If the CA is not  $(t, n)$ -AI, then the attacker could gain some information on the internal state of the cipher by analyzing the differences of the output distributions in the original CA and the asynchronous CA. Similar fault attacks have already been investigated on stream ciphers based on clock-controlled *Linear Feedback Shift Registers* (LFSR), such as LILI-128 (Dawson et al. 2000). For further information on the topic, Hoch and Shamir (2004) provide more references on clock fault attacks on stream ciphers.

This paper is an extended version of (Mariot 2016). In particular, the new contribution is twofold: from the theoretical side, we formally prove the necessity of central permutivity to have asynchrony immunity, which was conjectured in (Mariot 2016) according to the experimental results reported there. From the empirical point of view, we employ this new theoretical result to consistently extend the experimental search of asynchrony immune rules, by considering larger neighborhood sizes.

In the remainder of this paper, we recall in Sect. 2 the necessary basic notions about Boolean functions and (asynchronous) CA, and we formally introduce the definition of asynchrony immunity in Sect. 3, giving some theoretical results regarding this property. In particular, we show that AI is invariant under the operations of reflection and complement and that, for high enough values of  $t$  (the maximum number of blocked cells), *central permutivity* is a necessary condition for asynchrony immunity. We then perform in Sect. 4 an exhaustive search of asynchrony immune CA having 8 output cells and neighborhood size up to 5, computing also their nonlinearity and algebraic normal form. Finally, we provide some possible ways to generalize the notion of asynchrony immunity and how this property can be linked to existing CA models in Sect. 5, as well as pointing out other avenues for future research on the subject.

## 2 Basic notions

In this section, we cover all necessary background definitions about one-dimensional CA, Boolean functions, and vectorial Boolean functions. In particular, we refer the reader to (Carlet 2010a, b) for an in-depth discussion of (vectorial) Boolean functions.

Recall that a *Boolean function* is a mapping  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ , where  $\mathbb{F}_2 = \{0, 1\}$  denotes the finite field of two elements.

Once an ordering of the  $n$ -bit input vectors has been fixed, each Boolean function  $f$  can be uniquely represented by the output column of its *truth table*, which is a vector  $\Omega_f$  of  $2^n$  binary elements. Therefore, the set of all possible Boolean functions of  $n$  variables, denoted by  $\mathcal{B}_n$ , has cardinality  $2^{2^n}$ . The interpretation of the vector  $\Omega_f$  as a decimal number is also called the *Wolfram code* of the function  $f$ . Another common way of representing a Boolean function is through its *Algebraic Normal Form* (ANF), that is, as a sum of products over its input variables. More formally, given  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  and  $x \in \mathbb{F}_2^n$ , the ANF will be of the form

$$P_f(x) = \bigoplus_{I \in 2^{[n]}} a_I \left( \prod_{i \in I} x_i \right), \quad (1)$$

where  $[n]$  is the initial segment of the natural numbers determined by  $n \in \mathbb{N}$ , i.e.,  $[n] = \{0, \dots, n-1\}$ , and the set  $I = \{i_1, \dots, i_t\} \subseteq [n]$  is a subset of  $t$  indices and thus an element of  $2^{[n]}$ , the power set of  $[n]$ . For all  $I \in 2^{[n]}$  the coefficient  $a_I \in \mathbb{F}_2$  is determined through the *Möbius transform* (Carlet 2010a). A function  $f$  is called *affine* if the only non null coefficients  $a_I$  are such that  $|I| \leq 1$ . In other words, the ANF is composed only of monomials of degree at most 1.

Boolean functions used in the design of symmetric ciphers must satisfy a certain number of properties in order to withstand particular cryptanalytic attacks. Two of the most important properties are *balancedness* and *nonlinearity*. A Boolean function  $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$  is balanced if its output vector  $\Omega_f$  is composed of an equal number of zeros and ones. Unbalanced Boolean functions produce a statistical bias in the output of a symmetric cipher, which can be exploited by an attacker.

The nonlinearity of  $f$ , on the other hand, is the minimum Hamming distance of  $\Omega_f$  from the set of all affine functions. The value of nonlinearity of  $f$  can be computed as  $Nl(f) = 2^{-1}(2^n - W_{\max}(f))$ , where  $W_{\max}(f)$  is the maximum absolute value of the *Walsh transform* of  $f$  (Carlet 2010a). The nonlinearity of a Boolean function used in a cipher should be as high as possible, in order to thwart *linear cryptanalysis* attacks. Nonetheless, there exist upper bounds on the nonlinearity achievable by a Boolean function with respect to the number of its input variables. In particular, for  $n$  even it holds that  $Nl(f) \leq 2^{n-1} - 2^{\frac{n}{2}-1}$ . Functions satisfying this bound with equality are called *bent*. On the other hand, for  $n$  odd the upper bound when  $n \leq 7$  is  $Nl(f) \leq 2^{n-1} - 2^{\frac{n-1}{2}}$ , which is achieved by *quadratic functions*. For  $n > 7$ , the exact bound is still not known.

Let  $n, m \in \mathbb{N}$ . A *vectorial Boolean function* of  $n$  input variables and  $m$  output variables (also called an  $(n, m)$ -function) is a mapping  $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ . In particular, a  $(n, m)$ -

function is defined by  $m$  Boolean functions of the form  $f_i : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ , called *coordinate functions*. Each  $0 \leq i < m$ , each  $f_i$  specifies the  $i$ -th output bit of  $F$ . That is, for each  $x \in \mathbb{F}_2^n$ , we have  $F(x)_i = f_i(x)$  for  $0 \leq i < m$ .

A *one-dimensional cellular automaton* (CA) can be seen as a particular case of vectorial Boolean function by limiting the way the coordinate functions can be defined. Let  $\ell, m, r \in \mathbb{N}$  be non-negative integers and let  $n = \ell + m + r$ . Let  $f : \mathbb{F}_2^{\ell+r+1} \rightarrow \mathbb{F}_2$  be a Boolean function of  $\ell + r + 1$  variables. A *cellular automaton* of length  $n$  with local rule  $f$ , *memory*  $\ell$  and *anticipation*  $r$  is the  $(n, m)$ -function  $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  defined for all  $i \in \{0, \dots, m - 1\}$  and for all  $x = (x_{-\ell}, \dots, x_{m+r}) \in \mathbb{F}_2^n$  as:

$$F(x_{-\ell}, \dots, x_{m+r-1})_i = f(x_{i-\ell}, \dots, x_{i+r}). \tag{2}$$

Thus, a CA is the special case of a vector Boolean function where all coordinate functions are defined uniformly.

A *t-asynchronous CA*, or *t-ACA*, induced by  $I$  is denoted by  $\tilde{F}_I$  and it is defined by the following global function  $\tilde{F}_I : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ :

$$\tilde{F}_I(x_{-\ell}, \dots, x_{m+r-1})_i = \begin{cases} f_i(x_{i-\ell}, \dots, x_{i+r}) & \text{if } i \notin I \\ x_i & \text{if } i \in I. \end{cases}$$

We also recall that a local rule  $f : \mathbb{F}_2^{\ell+r+1} \rightarrow \mathbb{F}_2$  is said to be *center permutive* when for each  $u \in \mathbb{F}_2^\ell, v \in \mathbb{F}_2^r$ , and  $y \in \mathbb{F}_2$  there exists a *unique*  $x \in \mathbb{F}_2$  such that  $f(uyv) = x$ . In the field  $\mathbb{F}_2$ , center permutivity can also be expressed in another way. A local rule  $f : \mathbb{F}_2^{\ell+r} \rightarrow \mathbb{F}_2$  is center permutive if there exists a function  $g : \mathbb{F}_2^{\ell+r} \rightarrow \mathbb{F}_2$  such that for all  $x = (x_0, \dots, x_{\ell+r}) \in \mathbb{F}_2^{\ell+r+1}$  we have that:

$$f(x_0, \dots, x_{\ell+r}) = x_\ell \oplus g(x_0, \dots, x_{\ell-1}, x_{\ell+1}, \dots, x_{\ell+r}).$$

### 3 Definition of asynchrony immunity

Recall that a CA  $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  with  $n = \ell + r + m$  is said to be *balanced* if for each  $y \in \mathbb{F}_2^m$ , the preimages of  $y$ , i.e., all  $x \in \mathbb{F}_2^n$  such that  $F(x) = y$ , denoted by  $F^{-1}(y)$  is such that  $|F^{-1}(y)| = 2^{\ell+r}$ . Asynchrony immune CA can then be defined as follows:

**Definition 1** Let  $n, m, r, \ell, t \in \mathbb{N}$  be non-negative integers, with  $n = \ell + m + r$ , and  $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  a balanced CA having local rule  $f : \mathbb{F}_2^{\ell+r+1} \rightarrow \mathbb{F}_2$ .

The CA  $F$  is said to be  $(t, n)$ -*asynchrony immune* (for short,  $(t, n)$ -AI) if for all sets  $I \subseteq [m]$  with  $|I| \leq t$  the resulting  $|I|$ -ACA  $\tilde{F}_I$  is balanced.

Among all possible  $2^{2^{\ell+r+1}}$  rules of memory  $\ell$  and anticipation  $r$ , we are interested in finding local rules that generates asynchrony immune CA satisfying additional useful cryptographic properties, such as high nonlinearity. As a consequence, proving necessary conditions for a rule to generate a  $(t, n)$ -AI is useful in reducing the size of the search space.

We start by proving that, for large enough CA and for high enough values of  $t$ , a necessary condition of  $f$  is central permutivity.

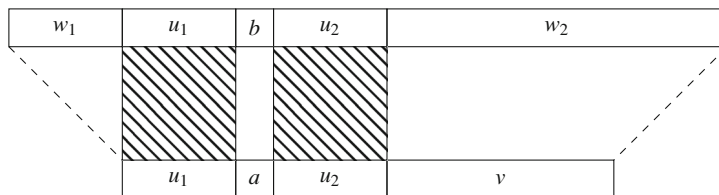
**Theorem 1** Let  $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  be a  $(t, n)$ -AI CA with memory  $\ell$  and anticipation  $r$ . If  $t \geq \ell + r$  and  $n \geq 2\ell + 2r + 1$  then the local rule  $f : \mathbb{F}_2^{\ell+r+1} \rightarrow \mathbb{F}_2$  is center permutive.

**Proof** Suppose  $F$  to be  $(t, n)$ -AI with  $t$  and  $n$  as in the hypothesis. Let  $y = u_1 a u_2 v \in \mathbb{F}_2^m$  be a configuration with  $u_1 \in \mathbb{F}_2^\ell, a \in \mathbb{F}_2, u_2 \in \mathbb{F}_2^r$ , and  $v \in \mathbb{F}_2^{m-\ell-r-1}$ . Let the set  $I \supseteq \{0, \dots, \ell, \ell + 2, \dots, r\}$  be a set of indices to be blocked. It then follows that each preimage of  $y$  can be expressed in the form  $x = w_1 u_1 b u_2 w_2$  with  $w_1 \in \mathbb{F}_2^\ell, b \in \mathbb{F}_2$ , and  $w_2 \in \mathbb{F}_2^{m+r-\ell-1}$ . Notice that both  $u_1$  and  $u_2$  remain unchanged when applying  $\tilde{F}_I$  to  $x$ , since their indices are all contained in  $I$ . This situation is illustrated in Fig. 1.

Since the value of the cells in  $w_1$  cannot influence any cell in  $\tilde{F}_I(x)$  (since all cells that can be influenced are blocked), if  $x = w_1 u_1 b u_2 w_2$  is a preimage of  $y$ , also  $x' = w'_1 u_1 b u_2 w_2$  for every  $w'_1 \in \mathbb{F}_2^\ell$  is a preimage of  $y$ . Hence, the first  $\ell$  cells of the automaton contribute a multiplicative factor of  $2^\ell$  for the number of preimages.

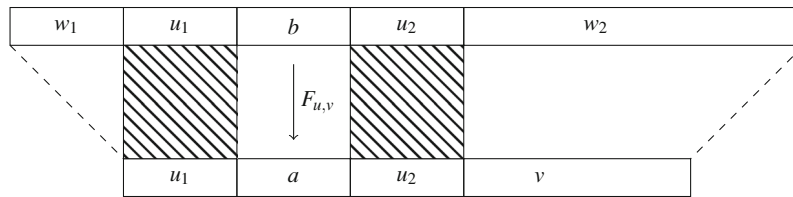
We are now going to prove that the remaining factor of  $2^r$  for the number of preimages is entirely due to the last  $m + r - 1$  cells (i.e., the part denoted by  $w_2$ ).

For the sake of argument, suppose that the multiplicative factor contributed by the last  $m + r - \ell - 1$  cells (i.e., the



**Fig. 1** The construction employed by the proof of Theorem 1. The patterned background denotes the blocked cells. Here it is possible to see that the part labeled with  $w_1$  cannot influence any of the output

cells. The cell labeled  $b$  can influence only the cell labeled  $a$  in the output, thus forcing the local rule to be center permutive



**Fig. 2** The construction employed by the proof of Theorem 2. The patterned background denotes the blocked cells. For each value of  $u$  and  $v$  the function  $F_{u,v}$  is a bijection from  $\mathbb{F}_2^k$  to  $\mathbb{F}_2^k$  where  $k$  is the length of  $b$

part denoted by  $w_2$  in the preimages) is less than  $2^r$ , since only a single other cell in the preimage can change (the one denoted by  $b$ ), it follows that, in that case the following two configurations are preimages of  $y$  for some choice of  $w_2$ :

$$x = w_1 u_1 0 u_2 w_2$$

$$x' = w_1 u_1 1 u_2 w_2.$$

Notice that the value of  $a$  in  $y$  is either 0 or 1 and it is influenced only by its own value and the value of  $u_1$  and  $u_2$ . Without loss of generality, suppose that  $a = 0$ . Consider now the preimages of  $y' = u_1 1 u_2 v$ . To obtain 1 in the unblocked position between  $u_1$  and  $u_2$  then, it must be  $f(u_1 0 u_2) = 1$  or  $f(u_1 1 u_2) = 1$ , but by our previous assumption, both  $f(u_1 0 u_2)$  and  $f(u_1 1 u_2)$  are equal to 0, and  $y'$  has no preimages. Hence, our hypothesis that the part denote by  $w_2$  in the preimages contributes less than a factor of  $2^r$  in the number or preimages is inconsistent with the fact that  $\tilde{F}_I$  must be balanced.

Therefore, the parts  $w_1$  and  $w_2$  contribute, respectively, factors  $2^\ell$  and  $2^r$  in the number of preimages, for a total of  $2^{\ell+r}$  preimages. It follows that, for each  $u_1 \in \mathbb{F}_2^\ell$ ,  $u_2 \in \mathbb{F}_2^r$ , and  $a \in \mathbb{F}_2$  there should be only *one* value  $b \in \mathbb{F}_2$  such that  $f(u_1 b u_2) = a$ . This means that  $f$  is center permutive.  $\square$

The previous theorem can be generalized as follows:

**Theorem 2** Let  $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  be a  $(t, n)$ -AI CA with memory  $\ell$  and anticipation  $r$  and  $k \in \mathbb{N}$  be a non-negative integer. Then, if  $t \geq \ell + r$  and  $n \geq 2\ell + 2r + k$ , the function  $F_{u,v} : \mathbb{F}_2^k \rightarrow \mathbb{F}_2^k$ , which, for each  $u \in \mathbb{F}_2^\ell$  and  $v \in \mathbb{F}_2^r$ , is defined as  $F_{u,v}(x) = F'(uxv)$  where  $F' : \mathbb{F}_2^{k+\ell+r} \rightarrow \mathbb{F}_2^k$  is a CA with the same local rule as  $F$ , is a bijection.

**Proof** The proof of this theorem follows the same reasoning of the proof of Theorem 1, as depicted in Fig. 2. Let  $I$  be a set of indices to be blocked such that  $I \supseteq \{0, \dots, \ell - 1, \ell + k, \ell + k + r\}$ . Each element of  $\mathbb{F}_2^m$  can then be rewritten in the form  $y = u_1 a u_2 v$  with  $u_1 \in \mathbb{F}_2^\ell$ ,  $u_2 \in \mathbb{F}_2^r$ ,  $a \in \mathbb{F}_2^k$ , and  $v \in \mathbb{F}_2^{m-\ell-r-k}$ . Similarly, a preimage of  $y$  can be expressed in the form  $x = w_1 u_1 a u_2 w_2$  with  $w_1 \in \mathbb{F}_2^\ell$ ,  $w_2 \in \mathbb{F}_2^{m+r-\ell-k}$ , and  $a \in \mathbb{F}_2^k$ . Following the same reasoning of the proof of Theorem 1, it can be shown that the  $w_1$  part of the preimage contributes a factor  $2^\ell$  in the

number of preimages and that the  $w_2$  part contributes a factor of  $2^r$ . Hence, the part denoted by  $b$  in  $y$  can have only one preimage. Therefore, when restricted to the  $k$  cells “surrounded” by  $u_1$  and  $u_2$ , the global function of the CA is a bijection, as desired.  $\square$

Recall that the *reverse* of a vector  $x = (x_0, \dots, x_{n-1})$  is the vector  $x^R = (x_{n-1}, \dots, x_0)$  with all components of  $x$  appearing in reverse order. Also, the *complement* of  $x$  is the vector  $x^C = (1 \oplus x_0, \dots, 1 \oplus x_{n-1})$  where all components of  $x$  appear negated. Given a local rule  $f : \mathbb{F}_2^{\ell+r+1} \rightarrow \mathbb{F}_2$  it is possible to define its reverse  $f^R : \mathbb{F}_2^{\ell+r+1} \rightarrow \mathbb{F}_2$  as  $f^R(x) = f(x^R)$  and its complement  $f^C : \mathbb{F}_2^{\ell+r+1} \rightarrow \mathbb{F}_2$  as  $f^C(x) = 1 \oplus f(x)$  for all  $x \in \mathbb{F}_2^{\ell+r+1}$ . The definition of reverse and complement can also be extended to a CA  $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  in the following way:

$$F^R(x)_i = (F(x^R))^R_i = f(x_{i+r}, \dots, x_{i-\ell})$$

$$F^C(x)_i = 1 \oplus F(x)_i = 1 \oplus f(x_{i-\ell}, \dots, x_{i+r})$$

We can now show that, for a given  $(t, n)$ -AI CA it is possible to obtain other (not necessarily distinct)  $(t, n)$ -AI by taking either its reverse or its complement.

**Proposition 1** Let  $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  be a  $(t, n)$ -AI CA for some  $n, m, t \in \mathbb{N}$  with  $n = m + r + \ell$  and  $r = \ell$ . Then its reverse  $F^R$  is also a  $(t, n)$ -AI CA.

**Proof** Starting with the reverse CA, by definition  $F^R(x)$  is  $F(x^R)^R$ . Hence, given a set of indices  $I$  with  $|I| \leq t$ , the reflection of the  $|I|$ -ACA  $\tilde{F}_I^R$  is:

$$\tilde{F}_I^R(x)_i = (\tilde{F}_J(x^R))^R_i = \begin{cases} f(x_{i+r}, \dots, x_{i-\ell}) & \text{if } i \notin J \\ x_i & \text{if } i \in J \end{cases} \tag{3}$$

where  $J \subseteq \{-\ell, \dots, m + r - 1\}$  is defined as a “reverse” of the set  $I$  of indices, that is  $J = \{m + r - \ell - 1 - i : i \in I\}$ . Notice that  $J \subseteq [m]$  in all cases only if  $\ell = r$ . This means that for every set  $I$  of indices for  $F^R$ , the corresponding set  $J$  of indices in  $F$  is still a valid one (i.e., a subset of  $[m]$ ). Notice that since  $f$  generates a  $(t, n)$ -AI CA and  $|J| = |I| \leq t$ , the resulting ACA is still  $(t, n)$ -AI.  $\square$



Notice that, in general, if a  $(t, n)$ -AI CA has memory  $\ell$  and anticipation  $r$  with  $\ell \neq r$ , its reverse might not be a  $(t, n)$ -AI CA. In fact, since center permutivity of the local rule is not preserved, this negates a condition for asynchrony-immunity that, by Theorem 1, is necessary for large enough values of  $t$  and  $n$ .

**Proposition 2** *Let  $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$  be a  $(t, n)$ -AI CA for some  $n, m, t \in \mathbb{N}$ . Then its complement  $F^C$  is also a  $(t, n)$ -AI CA.*

**Proof** Let  $y \in \mathbb{F}_2^m$  be a configuration,  $I \subseteq [m]$  with  $|I| \leq t$ , and let  $(F_I^C)^{-1}(y)$  be the set of preimages of  $y$  under the function  $F_I^C$ . By definition, for each  $x \in \mathbb{F}_2^n$ ,  $F^C(x) = 1 \oplus F(x)$ . Hence, the set  $(F_I^C)^{-1}(y)$  is  $\{x : 1 \oplus F_I(x) = y\}$ , which is  $\{x : F_I(x) = 1 \oplus y\}$  which corresponds to  $F_I^{-1}(1 \oplus y)$ . Since  $F$  is a  $(t, n)$ -AI CA, and all  $y$  ranges across all elements of  $\mathbb{F}_2^m$  (and thus  $1 \oplus y$  does the same),  $F_I^{-1}$  is balanced and  $(F_I^C)^{-1}$  is also balanced. Since this holds for every set  $I$  of cardinality at most  $t$ , it follows that  $F^C$  is also a  $(t, n)$ -AI CA, as required.  $\square$

Upper bounds on the size of the search space could be derived using techniques from Cattaneo et al. (1997) w.r.t. to the set of transformations  $F^R, F^C, F^{RC}, Id$ , where  $Id$  is the identity transformation.

### 4 Search of AI rules up to 5 variables

In order to search for asynchrony immune rules having additional cryptographic properties, by Theorem 1 and Propositions 1 and 2 we only need to explore center-permutive rules under the equivalence classes induced by reflection and complement.

In our experiments, we fixed the number of output bits in the CA to  $m = 8$ . Since we are considering only center-permutive rules, we tested only the smallest value of  $t$  satisfying the hypothesis of Theorem 1. The reason why we limited our analysis to these particular values is twofold. First, checking for asynchrony immunity is a computationally cumbersome task, since it requires to determine the output distribution of the  $t$ -ACA for all possible choices of at most  $t$  blocked cells. Second, the sizes of vectorial Boolean functions employed as nonlinear components in several real-world cryptographic primitives is limited. A concrete example is given by AES (NIST/ITL/CSD 2001), which employs a S-box with 8 output bits.

Table 1 shows all CA parameters considered in our experiments from 3 to 5 input variables of the local rules, while keeping the value of output bits fixed to  $m = 8$ . Recall that, since we need to consider only center permutive local rules, we do not need to explore the entire  $\mathcal{B}_{\ell+r+1}$  space, but only the subset  $\mathcal{C}_{\ell+r+1}$  having cardinality  $2^{2^{\ell+r}}$ .

**Table 1** CA parameters for  $m = 8$  output bits

$n$	$\ell$	$r$	$t$	$ \mathcal{B}_{\ell+r+1} $	$ \mathcal{C}_{\ell+r+1} $
10	1	1	2	256	16
11	1	2	3	65536	256
12	2	2	4	$\approx 4.3 \times 10^9$	65,536

We started our investigation by performing an exhaustive search among all CA rules with  $\ell = r = 1$  (that is, rules of 3 variables), which are also known in the CA literature as *elementary rules*. Up to reflection and complement, and neglecting the identity rule that is trivially AI for every length  $n$  and order  $t$ , out of the  $2^{2^3} = 256$  elementary rules we found that only rule 60 is  $(2, 10)$ -asynchrony immune. However, rule 60 is not interesting from the cryptographic standpoint, since it is linear (its ANF being  $x_2 \oplus x_3$ ).

We thus extended the search by considering all local rules of 4 and 5 input variables, according to the values of  $\ell$  and  $r$  reported in Table 1.

For the case of 4 variables, the search returned a total of 18 rules satisfying  $(3, 11)$ -asynchrony immunity, among which several of them were nonlinear. Table 2 reports the Wolfram codes of the discovered rules, along with their nonlinearity values and algebraic normal form. It can be observed that 12 rules out of 18 are nonlinear, but none of them is a bent function (since the nonlinearity value in this case would be 6).

For 5 variables, Table 3 reports the list of  $(4, 12)$ -AI CA. One can see that in this case most of the asynchrony immune functions are nonlinear, and moreover two of them achieve the maximum nonlinearity allowed by the quadratic bound, which in this case is 12.

### 5 Open problems

There are many possible research directions for exploring asynchrony immune CA, mainly related to generalizations and relations with other models.

From the generalization point of view, we can relax the assumption that an attacker can control the updating of at most  $t$  cells on  $n$  cells CA. We can suppose that additional “anti-tamper” measures are present and, for example, that the attacker can only take control of non-consecutive cells. More in general, we can define  $(\mathcal{F}, n)$ -asynchrony immune CA where  $\mathcal{F} \subseteq 2^{[m]}$  is a family of subsets of  $\{0, \dots, m - 1\}$ . The standard  $(t, n)$ -AI CA can be recovered by taking  $\mathcal{F}$  as the set of all subsets of  $[m]$  with cardinality at most  $t$ . It would be interesting to understand for what families of sets the theorems of this paper still

**Table 2** List of (3, 11)-asynchrony immune CA rules of neighborhood size 4

Rule	$Nl(f)$	$f(x_0, x_1, x_2, x_3)$	Rule	$Nl(f)$	$f(x_0, x_1, x_2, x_3)$
13107	0	$1 \oplus x_1$	14028	2	$x_1 \oplus x_0x_3 \oplus x_2x_3 \oplus x_0x_2x_3$
13116	4	$x_1 \oplus x_2 \oplus x_3 \oplus x_2x_3$	14643	2	$1 \oplus x_1 \oplus x_0x_3 \oplus x_0x_2x_3$
13155	2	$1 \oplus x_1 \oplus x_2 \oplus x_0x_2 \oplus x_2x_3 \oplus x_0x_2x_3$	14796	2	$x_1 \oplus x_3 \oplus x_0x_3 \oplus x_0x_2x_3$
13164	2	$x_1 \oplus x_0x_2 \oplus x_3 \oplus x_0x_2x_3$	15411	4	$1 \oplus x_1 \oplus x_3 \oplus x_2x_3$
13203	2	$1 \oplus x_1 \oplus x_0x_2 \oplus x_0x_2x_3$	15420	0	$x_1 \oplus x_2$
13212	2	$x_1 \oplus x_2 \oplus x_0x_2 \oplus x_3 \oplus x_2x_3 \oplus x_0x_2x_3$	15555	0	$1 \oplus x_1 \oplus x_2 \oplus x_3$
13251	4	$1 \oplus x_1 \oplus x_2 \oplus x_2x_3$	15564	4	$x_1 \oplus x_2x_3$
13260	0	$x_1 \oplus x_3$	26214	0	$x_0 \oplus x_1$
13875	2	$1 \oplus x_1 \oplus x_3 \oplus x_0x_3 \oplus x_2x_3 \oplus x_0x_2x_3$	26265	0	$1 \oplus x_0 \oplus x_1 \oplus x_3$

**Table 3** List of (4, 12)-asynchrony immune CA rules of neighborhood size 5

Rule	$Nl(f)$	$f(x_1, x_2, x_3, x_4, x_5)$	Rule	$Nl(f)$	$f(x_1, x_2, x_3, x_4, x_5)$
252691440	4	$x_3 \oplus x_4 \oplus x_2x_4 \oplus x_5 \oplus x_4x_5 \oplus x_2x_4x_5$	3031741620	8	$x_2 \oplus x_1x_2 \oplus x_3$
252702960	0	$x_3 \oplus x_5$	3035673780	6	$x_2 \oplus x_1x_2 \oplus x_3 \oplus x_2x_5 \oplus x_1x_2x_5 \oplus x_2x_4x_5 \oplus x_1x_2x_4x_5$
253678110	10	$x_1 \oplus x_2 \oplus x_1x_2 \oplus x_3 \oplus x_2x_4 \oplus x_4x_5 \oplus x_1x_4x_5 \oplus x_1x_2x_4x_5$	3537031890	8	$x_1 \oplus x_1x_2 \oplus x_3$
255652080	4	$x_3 \oplus x_2x_5 \oplus x_4x_5 \oplus x_2x_4x_5$	3537035730	8	$x_1 \oplus x_1x_2 \oplus x_3 \oplus x_4 \oplus x_2x_4 \oplus x_4x_5 \oplus x_2x_4x_5$
264499440	4	$x_3 \oplus x_5 \oplus x_2x_5 \oplus x_2x_4x_5$	3539005680	2	$x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_4x_5$
267390960	0	$x_3 \oplus x_4$	4027576500	6	$x_2 \oplus x_1x_2 \oplus x_3 \oplus x_2x_4 \oplus x_1x_2x_4 \oplus x_5 \oplus x_2x_5 \oplus x_1x_2x_5 \oplus x_4x_5 \oplus x_2x_4x_5 \oplus x_1x_2x_4x_5$
267448560	8	$x_3 \oplus x_4x_5$	4030525680	4	$x_3 \oplus x_2x_5 \oplus x_2x_4x_5$
505290270	8	$x_1 \oplus x_2 \oplus x_1x_2 \oplus x_3$	4031508720	6	$x_3 \oplus x_5 \oplus x_2x_5 \oplus x_1x_2x_5 \oplus x_4x_5 \oplus x_2x_4x_5 \oplus x_1x_2x_4x_5$
505336350	8	$x_1 \oplus x_2 \oplus x_1x_2 \oplus x_3 \oplus x_2x_4 \oplus x_2x_4x_5$	4038390000	2	$x_3 \oplus x_2x_5 \oplus x_1x_2x_5 \oplus x_2x_4x_5 \oplus x_1x_2x_4x_5$
509222490	4	$x_1 \oplus x_3 \oplus x_2x_4 \oplus x_1x_2x_4$	4039373040	4	$x_3 \oplus x_5 \oplus x_2x_5 \oplus x_4x_5 \oplus x_2x_4x_5$
517136850	12	$x_1 \oplus x_1x_2 \oplus x_3 \oplus x_4 \oplus x_2x_4 \oplus x_4x_5$	4040348370	6	$x_1 \oplus x_1x_2 \oplus x_3 \oplus x_1x_4x_5 \oplus x_1x_2x_4x_5$
756994590	12	$x_1 \oplus x_2 \oplus x_1x_2 \oplus x_3 \oplus x_2x_4 \oplus x_4x_5$	4042268400	6	$x_3 \oplus x_1x_4 \oplus x_2x_4 \oplus x_1x_2x_4 \oplus x_1x_4x_5 \oplus x_2x_4x_5 \oplus x_1x_2x_4x_5$
2018211960	8	$x_1x_2 \oplus x_3 \oplus x_5 \oplus x_2x_5 \oplus x_4x_5 \oplus x_2x_4x_5$	4042276080	4	$x_3 \oplus x_2x_4 \oplus x_2x_4x_5$
2018212080	10	$x_3 \oplus x_1x_2x_4 \oplus x_5 \oplus x_2x_5 \oplus x_1x_2x_5 \oplus x_4x_5 \oplus x_2x_4x_5 \oplus x_1x_2x_4x_5$	4042310640	4	$x_3 \oplus x_4 \oplus x_2x_4 \oplus x_4x_5 \oplus x_2x_4x_5$
2526451350	0	$x_1 \oplus x_2 \oplus x_3$	4042318320	2	$x_3 \oplus x_4 \oplus x_1x_4 \oplus x_2x_4 \oplus x_1x_2x_4 \oplus x_4x_5 \oplus x_1x_4x_5 \oplus x_2x_4x_5 \oplus x_1x_2x_4x_5$
3023877300	6	$x_2 \oplus x_1x_2 \oplus x_3 \oplus x_1x_2x_5 \oplus x_1x_2x_4x_5$	4042322160	0	$x_3$
3027809460	8	$x_2 \oplus x_1x_2 \oplus x_3 \oplus x_2x_5 \oplus x_2x_4x_5$			

hold. Also, what are some families that are “plausible” from a real-world point of view? This study will also require to explore the different methods that can be employed by an attacker to take control of some cells and what physical limits restrict the patterns of blocked cells that can be generated.

Another research direction is to find relations with already existing CA models that can be used to implement AI CA. Take, for example, the *Multiple Updating Cycles CA* (MUCCA) (Manzoni et al. 2016), where each cell has a speed  $1/k$  for a positive  $k \in \mathbb{N}$  and a cell updates only if the current time step is a multiple of  $k$ . This means that, at different time steps, different cells might be active. If the current time step is not known or if it is under the attacker’s control, then a CA that is  $(t, n)$ -AI can withstand any situation in which the number of “slow” cells (i.e., with speed less than 1) is bounded by  $t$ . More generally, in what other models of ACA being asynchrony immune can protect from an attacker that controls some variables (like the time step in MUCCA)?

Subsequently, we have found that for size  $n = 11$  there are no  $(11, 4)$ -AI CA rules reaching maximum nonlinearity, that is, none of them is a bent function. Hence, an interesting question would be if there exists at least one bent AI CA rule of larger number of variables, and if it is possible to design an infinite family of bent AI CA.

Finally, from the cryptanalysis point of view, it would be interesting to analyze the resistance to clock-fault attacks of cryptographic primitives and ciphers based on cellular automata, such as the stream cipher CAR30 (Das and Chowdhury 2013), the  $\chi$  S-box employed in the Keccak sponge construction (Bertoni et al. 2008), or the CA-based S-boxes optimized through *Genetic Programming* in (Picek et al. 2017; Mariot et al. 2019) and to verify if plugging in their design one of the AI CA rules found here decreases their possible vulnerability.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Bertoni G, Daemen J, Peeters M, Van Assche G, Van Keer R (2008) The KECCAK reference. <http://keccak.team>. Accessed 6 Sept 2018
- Carlet C (2010a) Boolean functions for cryptography and error correcting codes. *Boolean Models Methods Math Comput Sci Eng* 2:257–397
- Carlet C (2010b) Vectorial Boolean functions for cryptography. *Boolean Models Methods Math Comput Sci Eng* 134:398–469
- Cattaneo G, Formenti E, Margara L, Mauri G (1997) Transformations of the one-dimensional cellular automata rule space. *Parallel Comput* 23(11):1593–1611
- Chor B, Goldreich O, Hastad J, Freidmann J, Rudich S, Smolensky R (1985) The bit extraction problem or t-resilient functions. In: 26th annual symposium on foundations of computer science. IEEE, pp 396–407
- Das S, Chowdhury DR (2013) CAR30: a new scalable stream cipher with rule 30. *Cryptogr Commun* 5(2):137–162
- Dawson E, Clark A, Golic J, Millan W, Penna L, Simpson L (2000) The lili-128 keystream generator. In: Proceedings of first NESSIE workshop
- Formenti E, Imai K, Martin B, Yunès JB (2014) Advances on random sequence generation by uniform cellular automata. In: Calude C, Freivalds R, Kazuo I (eds) *Computing with new resources*. Springer, Berlin, pp 56–70
- Hoch JJ, Shamir A (2004) Fault analysis of stream ciphers. In: 6th international workshop on cryptographic hardware and embedded systems—CHES 2004, August 11–13, 2004. Proceedings. Cambridge, MA, pp 240–253
- Leporati A, Mariot L (2014) Cryptographic properties of bipermutive cellular automata rules. *J Cell Autom* 9:437–475
- Manzoni L, Porreca AE, Umeo H (2016) The firing squad synchronization problem on higher-dimensional CA with multiple updating cycles. In: 4th international workshop on applications and fundamentals of cellular automata—AFCA 2016, Hiroshima, Japan
- Mariot L (2016) Asynchrony immune cellular automata. In: *Cellular automata—12th international conference on cellular automata for research and industry, ACRI 2016, Fez, Morocco, September 5–8, 2016*. Proceedings, pp 176–181
- Mariot L, Picek S, Leporati A, Jakobovic D (2019) Cellular automata based S-boxes. *Cryptogr Commun* 11(1):41–62
- Martin B (2006) A Walsh exploration of elementary CA rules. In: *International workshop on cellular automata, Hiroshima University*, pp 25–30
- NIST/ITL/CSD (2001) Advanced encryption standard (AES). FIPS PUB 197. <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>. Accessed 6 Sept 2018
- Picek S, Mariot L, Yang B, Jakobovic D, Mentens N (2017) Design of S-boxes defined with cellular automata rules. In: *Proceedings of the computing frontiers conference, CF’17, Siena, Italy, May 15–17, 2017*, pp 409–414

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.