EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
Vol. 3, No. 4, 2010, 765-778
ISSN 1307-5543 – www.ejpam.com

The Cryptographic Properties of Von Neumann Cellular Automata

J. Escuadra Burrieza[1], A. Martín del Rey [2,*], J.L. Pérez Iglesias[1], G. Rodríguez Sánchez [3], A. Queiruga Dios[4], A. de la Villa Cuenca[5]

[1] *Department of Informatics and Automation, High Polytechnic School of Zamora, Universidad de Salamanca, Zamora, Spain*
[2] *Department of Applied Mathematics, High Polytechnic School of Ávila, Universidad de Salamanca, Ávila, Spain*
[3] *Department of Applied Mathematics, High Polytechnic School of Zamora, Universidad de Salamanca, Zamora, Spain*
[4] *Department of Applied Mathematics, High Technic School of Béjar, Universidad de Salamanca, Béjar-Salamanca, Spain*
[5] *Department of Applied Mathematics and Computation, ICAI, Universidad Pontifia Comillas, Madrid, Spain*

Abstract. In this paper it is shown that two-dimensional cellular automata with Von Neumann neighborhoods are not suitable for cryptographic purposes. This result is obtained after analyzing the most important cryptographic properties of boolean functions defining their local transition rules

2000 Mathematics Subject Classifications: 68Q80, 94A60, 94C10

Key Words and Phrases: Cellular automata, Boolean functions, Cryptographic properties, Non-linearity, Walsh transform

1. Introduction and Preliminaries

Cellular automata are simple models of computation in which time and space are discrete. They are formed by a finite set of memory units called cells which are endowed with a state (0 or 1) at each time step. These states change according to a local transition rule whose variables are the states of the neighbor cells at the previous time step [see 14].

Despite its simplicity, cellular automata are able to simulate several types of complex phenomena. Thus, a lot of works have been appeared in scientific literature proposing models

*Corresponding author.

Email addresses: jeb@usal.es (J. Escuadra), delrey@usal.es (A. Martín del Rey), jpi@usal.es (J. Iglesias), gerardo@usal.es (G. Sánchez), queirugadios@usal.es (A. Dios), avilla@dmc.icai.upcomillas.es (A. de la Villa Cuenca)

based on cellular automata to simulate the spread of forest fires or epidemics, the behavior of traffic flow, growth phenomena, reaction-diffusion processes, image processing algorithms, etc. Specially interesting cellular automata applications are those related to cryptography: secret key and public key cryptosystems have been proposed [see, for example, 1, 3, 10], and also hash functions and secret sharing schemes have been designed [see, for example, 7, 8].

Most of cryptographic protocols use one-dimensional cellular automata, *i.e.* the cells are arranged in a line as a string of binary values. This is due to the data managed through computer devices (text files, voice files, image files, etc.) is represented by sequences of bits. However, it is sometimes necessary to handle data which is given in a two-dimensional format (for example two-dimensional codes: QR code, PDF 417, Datamatrix codes, MaxiCode, etc.). In this regard, two-dimensional cellular automata (the cells are arranged in the form of an homogeneous two-dimensional lattice) must be used. Unfortunately, although some two-dimensional cellular automata-based cryptographic protocols for digital images have been proposed, to date any study of their cryptographic properties has been done. In the case of elementary cellular automata (whose local transition functions are defined by means of 3-variable boolean functions), the analysis has been done and, unfortunately, there is not any elementary cellular automata with good cryptographic properties [see 6].

As a consequence, this work aims to study and classify the two-dimensional cellular automata with Von Neumann neighborhoods according to its cryptographic use. Specifically, as local transition functions of two-dimensional cellular automata are boolean functions, the most important cryptographic properties of such boolean functions will be explored.

The rest of the paper is organized as follows: In section 2 the basic theory of boolean functions is introduced; the mathematical background on two-dimensional cellular automata is shown in section 3; the main cryptographic properties that must satisfy a cryptographic boolean function are shown in section 4; in section 5 we test these properties for the local transition functions of two-dimensional cellular automata; finally, the discussion is presented in section 6.

2. Boolean functions

Let $\mathbb{F}_2^n$ be the n-dimensional vector space over the Galois field $\mathbb{F}_2 = \{0,1\}$. The functions of the form $f : \mathbb{F}_2^n \to \mathbb{F}_2$ are called boolean functions, and the set of all n-variable boolean functions is denoted by $\mathscr{BF}_n$ being its cardinal $\left|\mathscr{BF}_n\right| = 2^{2^n}$. The Hamming weight of a vector $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$ is the number of its non-zero coordinates whereas the Hamming weight of an n-variable boolean function f is defined as

$$w_H(f) = \left|\left\{x \in \mathbb{F}_2^n \text{ such that } f(x) \neq 0\right\}\right|, \tag{1}$$

that is, it is the cardinal of its support. Furthermore, the Hamming distance between two boolean functions $f, g \in \mathscr{BF}_n$ is $d_H(f,g) = w_H(f \oplus g)$, where $(f \oplus g)(x) = f(x) \oplus g(x)$.

The usual representation of a boolean function f is by means of its Algebraic Normal Form (ANF for short), which is the n-variable polynomial representation over $\mathbb{F}_2$, that is:

$$f\left(x_1,\ldots,x_n\right)=\bigoplus_{\substack{1\leq k\leq n\\ 1\leq i_1,i_2,\ldots,i_k\leq n}} a_{i_1i_2\ldots i_k}\cdot x_{i_1}\cdot x_{i_2}\cdot\ldots\cdot x_{i_k}, \tag{2}$$

where $a_{i_1\ldots i_k}\in\mathbb{F}_2$. The degree of the ANF is the algebraic degree of the function. The simplest boolean functions considering their ANF are the affine functions: $f\left(x_1,\ldots,x_n\right)=a_1x_1\oplus a_2x_2\oplus\ldots\oplus a_nx_n\oplus a_0$, where $a_0,a_1,\ldots,a_n\in\mathbb{F}_2$. When $a_0=0$, we have the linear functions and they are denoted by $l_a\left(x\right)$ with $a=\left(a_1,\ldots,a_n\right)\in\mathbb{F}_2^n$.

The Walsh transform of an n-boolean function f is the pseudo-boolean function $\hat{f}$ such that

$$\begin{aligned}\hat{f}:\mathbb{F}_2^n &\to \mathbb{R}\\ u &\mapsto \hat{f}\left(u\right)=\sum_{x\in\mathbb{F}_2^n}(-1)^{x\bullet u}f\left(x\right)\end{aligned} \tag{3}$$

where $x\bullet u$ stands for the usual inner product. Note that

$$\hat{f}\left(0\right)=\sum_{x\in\mathbb{F}_2^n}(-1)^{x\bullet 0}f\left(x\right)=\sum_{x\in\mathbb{F}_2^n}f\left(x\right)=w_H\left(f\right), \tag{4}$$

and, as a consequence, $d_H\left(f,g\right)=w_H\left(f\oplus g\right)=\widehat{\left(f+g\right)}\left(0\right)$.

The derivative of the n-variable boolean function f with respect to $b\in\mathbb{F}_2^n$ is the following boolean function:

$$\begin{aligned}D_bf:\mathbb{F}_2^n &\to \mathbb{F}_2\\ x &\mapsto D_bf\left(x\right)=f\left(x\right)\oplus f\left(x\oplus b\right)\end{aligned} \tag{5}$$

The linear kernel of f is denoted by $\mathscr{LK}\left(f\right)$ and it is defined as the following subspace of $\mathbb{F}_2^n$:

$$\mathscr{LK}\left(f\right)=\left\{b\in\mathbb{F}_2^n \text{ such that } D_bf \text{ is a constant function}\right\}. \tag{6}$$

Every element $b\in\mathscr{LK}\left(f\right)$ is called linear structure of f.

As is well known, boolean functions play an important role in symmetric cryptography [see, for example 9, and references therein]. The cryptographic usefulness of boolean functions is measured by some cryptographic characteristics. The most important of these properties are the following: high algebraic degree, balancedness, high non-linearity, resiliency, higher-order propagation criteria and the non-existence of nonzero linear structures. In order to resist modern cryptanalytic attacks (based on linear approximation and differential characteristics), highly nonlinear boolean functions with good propagation criteria and less linear structure are basically needed.

3. Two-dimensional Cellular Automata

Two-dimensional cellular automata (CA for short) are finite state machines consisting of $r\times c$ cells which are uniformly arranged on a finite two-dimensional lattice. Each cell can

assume two states (0 or 1) and it changes synchronously in discrete steps of time according to a local transition function f. In this regard, the state of each cell at time $t+1$ depends on the states of the its neighbor cells at time t. Then, if s_{ij}^t is the state of the (i,j)-th cell at time $t+1$, then:

$$s_{ij}^{t+1} = f\left(s_{i-1,j}^t, s_{i,j-1}^t, s_{ij}^t, s_{i,j+1}^t, s_{i+1,j}^t\right) \in \mathbb{F}_2. \tag{7}$$

Note that in this work Von Neumann neighborhoods are considered, that is, the neighborhood of each cell is formed by its four nearest cells placed at north, south, east and west.

As the number of cells is finite, it is necessary to establish some type of boundary conditions. Usually one considers periodic boundary conditions (in topological terms, the lattice can be thought of as being mapped onto a three-dimensional torus) or null boundary conditions (the cells beyond each end are modified to maintain state 0 at every step of time).

Note that the local transition function

$$\begin{array}{rcl} f: \mathbb{F}_2^4 & \to & \mathbb{F}_2 \\ \left(s_{i-1,j}^t, s_{i,j-1}^t, s_{i,j+1}^t, s_{i+1,j}^t\right) & \longmapsto & s_{ij}^{t+1} = f\left(s_{i-1,j}^t, s_{i,j-1}^t, s_{i,j+1}^t, s_{i+1,j}^t\right) \end{array} \tag{8}$$

is a 4-variable boolean function. Consequently, there exists $2^{2^4} = 2^{16} = 65,536$ bidimensional cellular automata with Von Neumann neighborhoods. As the ANF of such CA is:

$$\begin{array}{rcl} f\left(s_{i-1,j}^t, s_{i,j-1}^t, s_{i,j+1}^t, s_{i+1,j}^t\right) & = & \lambda_0 \oplus \lambda_1 s_{i-1,j}^t \oplus \lambda_2 s_{i,j-1}^t \oplus \lambda_3 s_{i,j+1}^t \oplus \lambda_4 s_{i+1,j}^t \\ & & \oplus \lambda_5 s_{i-1,j}^t s_{i,j-1}^t \oplus \lambda_6 s_{i-1,j}^t s_{i,j+1}^t \oplus \lambda_7 s_{i-1,j}^t s_{i+1,j}^t \\ & & \oplus \lambda_8 s_{i,j-1}^t s_{i,j+1}^t \oplus \lambda_9 s_{i,j-1}^t s_{i+1,j}^t \oplus \lambda_{10} s_{i,j+1}^t s_{i+1,j}^t \\ & & \oplus \lambda_{11} s_{i-1,j}^t s_{i,j-1}^t s_{i,j+1}^t \oplus \lambda_{12} s_{i-1,j}^t s_{i,j-1}^t s_{i+1,j}^t \\ & & \oplus \lambda_{13} s_{i-1,j}^t s_{i,j+1}^t s_{i+1,j}^t \oplus \lambda_{14} s_{i,j-1}^t s_{i,j+1}^t s_{i+1,j}^t \\ & & \oplus \lambda_{15} s_{i-1,j}^t s_{i,j-1}^t s_{i,j+1}^t s_{i+1,j}^t, \end{array} \tag{9}$$

then the CA can be indexed by the following rule number:

$$w = \sum_{k=0}^{15} \lambda_k 2^k. \tag{10}$$

4. The Cryptographic Properties of Boolean Functions

As is mentioned in the Introduction, the suitable boolean functions for cryptographic uses must satisfy some properties. In this sense the most important cryptographic criteria for boolean functions are the algebraic degree, the balancedness, the resiliency, the nonlinearity, the propagation criterion and the non-existence of nonzero linear structures. Obviously, all of these characteristics cannot be optimum at the same time and consequently trade-offs must be taken into account. In what follows, we describe the mentioned properties:

The Algebraic Degree Cryptographic n-variable boolean functions must have high algebraic degrees since otherwise the cryptographic protocols can be successfully cryptanalyzed [see, for example, 5, 4, 12]. Nevertheless, we have to take into account that the n-variable boolean functions with algebraic degrees n or $n-1$ do not optimally achieve other cryptographic properties such as nonlinearity or resiliency.

Balancedness Cryptographic boolean functions must be balanced, that is, their outputs must be uniformly distributed over $\mathbb{F}_2$. This properties allows one to avoid statistical dependence between the input and the output, that can be used in some types of cryptanalytic attacks.

Resiliency There is an additional condition to balancedness with special importance in the case of the design of stream ciphers: The m-resiliency. An n -variable boolean function f is said to be m-resilient if it is a balanced function when we keep constant $0 < m \leq n$ variables. If a boolean function is not m-resilient then there exists a correlation between the output of the function and (at most) m coordinates of its input (correlation attack). G.Z. Xiao and J.L. Massey characterized the resiliency by means of the discrete Fourier transformation [see 15]:

Theorem 1. *An n-variable boolean function f is m-resilient if and only if it is balanced and $\hat{f}(u) = 0$ for all $u \in \mathbb{F}_2^n$ such that $0 < w_H(u) \leq m$.*

The Nonlinearity Cryptographic boolean functions must lie at large Hamming distance to all affine boolean functions. The nonlinearity of an n-variable boolean function f is defined as

$$\mathcal{NL}(f) = \min_{a \in \mathbb{F}_2^n} \{d_H(f, l_a)\}; \tag{11}$$

in this sense, a boolean function will be considered as highly nonlinear if its nonlinearity is near 2^n. It is shown that $\mathcal{NL}(f) \leq 2^{n-1} - 2^{n/2-1}$, and if the equality holds, f is called bent function. Note that if n is odd the last inequality for $\mathcal{NL}(f)$ cannot be an equality and in this case $\mathcal{NL}(f) \leq 2^{n-1} - 2^{\frac{n-1}{2}}$.

The Bent functions that are not balanced are not suitable for cryptographic purposes. As a consequence it is necessary to study the n-variable boolean functions which have large but not optimal nonlinearities, say between $2^{n-1} - 2^{\frac{n-1}{2}}$ and $2^{n-1} - 2^{n/2-1}$, among which some balanced functions exist.

The Propagation Criterion (PC) In order to assure well diffusion properties, boolean functions must satisfy the Propagation Criterion. This criterion was introduced by B. Preneel *et al.* [see 11] and it is based on the properties of the derivatives of boolean functions that gives the behaviour of such functions when some variables of the input are complemented. The n-variable boolean function f satisfies the propagation criterion PC with respect to $B \subset \mathbb{F}_2^n$ if for every $b \in B$ the derivative function, $D_b f$, is balanced. Moreover, the boolean function f

satisfies $PC(k)$ if it satisfies PC with respect to the set

$$W(k) = \left\{ b \in \mathbb{F}_2^n - \{0\} \text{ such that } w_H(b) \leq k \right\}. \tag{12}$$

The Strict Avalanche Criterion (SAC) introduced by Webster and Tavares in [13] is the particular case of PC for $k = 1$.

Non-existence of Nonzero Linear Structure Nonlinear n-variable boolean functions with applications in Cryptography (specially in block ciphers) should have not nonzero linear structures [see 2].

5. Analysis of the cryptographic properties of CA

In what follows we will test the cryptographic properties of local transition functions of two-dimensional cellular automata with Von Neumann neighborhoods, and the results obtained will be analyzed.

As the number of two-dimensional cellular automata satisfying some cryptographic properties is too high (for example, as is mentioned below, the number of balanced rules is $12,870$), it is impossible to be listed all their rule numbers in the paper. As a consequence, a simple code in Mathematica to compute explicitly all this CA is included in the Appendix.

The Algebraic Degree In Table 1, the number of CA rules with the different algebraic degrees are shown:

Table 1: Classification of CAs according to their algebraic degree.

Algebraic degree	Number of CAs
0	2
1	30
2	2,016
3	30,720
4	32,768

As is mentioned above, affine CA are especially important; In our case, there are 32 affine local transition functions: those given by the following rule numbers $0, 1, \ldots, 31$.

Balancedness A 4-variable boolean function is balanced when the number of 1's and 0's of its truth table is the same and equals to 8. A simple computation shows that there are $12,870$ CAs whose local transition function is a balanced 4-variable boolean function. Morevoer, in Table 2 the classification of balanced CAs according to its algebraic degree is shown:

Table 2: Classification of CAs according to their algebraic degree and balancedness.

Algebraic degree	Number of balanced CAs
0	0
1	30
2	840
3	12,000
4	0

The Nonlinearity For 4-variable boolean functions it is $\mathcal{NL}(f) \leq 2^3 - 2 = 6$. In Table 3 the results obtained for balanced CAs whose algebraic degree is two or three are shown (there are not balanced functions of degree 4 and the functions of algebraic degree 1 are affine).

Table 3: Classification of CAs according to nonlinearity property.

$\mathcal{NL}$	Balanced CAs of algebraic degree 3	Balanced CAs of algebraic degree 2
6	0	0
5	0	0
4	10,080	840
3	0	0
2	1,920	0
1	0	0
0	0	0

Non-existence of Nonzero Linear Structure The non-linear CA without nonzero linear structures are all those balanced CAs of algebraic degree 3 and non-linearity equals to 4; that is, there are 10,080 CA rules with such properties. The remaining balanced CAs with nonzero non-linearity have nonzero linear structures.

Resiliency Finally, it is easy to check that the last obtained CA are not m-resilient for $0 \leq m \leq 4$.

Propagation Criterion It is not necessary to check this criterion since there is not CA rule passing the last one stated.

6. Discussion

Taking into account the results shown in the last section, it is easy to check that there not exist two-dimensional cellular automata with Von Neumann neighborhoods satisfying the following properties at the same time:

- Algebraic degree of the local transition functions of the CA: 2.
- Balancedness.
- Non-existence of nonzero linear structures.
- Non-linearity of the local transition functions of ECA: between 0 and 6.
- Resiliency: local transition functions m-resilients, $0 \leq m \leq 4$.

Consequently, it is shown that two-dimensional cellular automata cannot be used directly in the design of S-boxes or as combining functions for LFSRs outputs. Nevertheless, it is also true, that CA can be used in the design of other cryptographic protocols as is mentioned in the Introduction, since other mathematical properties, such as statistical properties or reversibility properties, are required for its used in such protocols. Note that this properties often depends not on the local transition function but on the global transition function of the CA.

ACKNOWLEDGEMENTS This work has been supported by Ministerio de Ciencia e Innovación (Spain) under grant MTM2008-02773.

References

[1] I. Damgard, A design principle for hash functions, in *Advances in Cryptology: Proc. of Crypto'89* (G. Brassard, Ed.), *Lect. Notes Comput. Sci.* 435: 416-427 1990.

[2] J.H. Evertse, Linear structures in block ciphers, in *Advances in Cryptology, Proc. of Eurocrypt'87* (D. Chaum, W. L. Price, Eds.), *Lect. Notes Comput. Sci.* 304: 249–266 1988.

[3] A. Fúster, P. Caballero, On the Use of Cellular Automata in Symmetric Cryptography. *Acta Appl. Math.* 93, 2: 215–236, 2006.

[4] X. Lai, Higher order derivatives and differential cryptanalysis, in *Communications and Cryptology*, Kluwer Academic Publishers, p. 227-233. 1994.

[5] L.R. Knudsen, Truncated and higher order differentials, in *Proc. of 2nd FSE* (B. Preneel, Ed.), Lect. Notes Comput. Sci. 1008: 196–211. 1995.

[6] B. Martin, A Walsh exploration of elementary CA rules. *J. Cellular Automata* 3, 2: 145–156. 2008.

[7] A. Martín del Rey, J. Pereira Mateus, G. Rodríguez Sánchez, A secret sharing scheme based on cellular automata. *Appl. Math. Comput.* 170: 1356-1364. 2005.

[8] A. Martín del Rey, A. Queiruga Dios, G. Rodríguez Sánchez, A (2,n)-secret sharing scheme based on linear cellular automata. *Int. J. Mod. Phys. C* 19, 10: 1529-1535. 2008.

[9] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1997.

[10] M. Mukherjee, N. Ganguly, P.P. Chaudhuri, Cellular automata based authentication. in *Proc. of ACRI 2002* (S. Bandini, B. Chopard, M. Tomassini, Eds.), *Lect. Notes Comput. Sci.* 2493: 259-269. 2002.

[11] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, J. Vandevalle, Propagation characteristic of boolean functions. in *Advances in Cryptology, Proc. of Eurocrypt'90* (I.B. Damgard, Ed.), *Lect. Notes Comput. Sci.* 473: 161–173. 1991.

[12] R.A. Rueppel, and O.J. Staffelbach, Products of linear recurring sequences with maximum complexity. *IEEE Trans. Inform. Theory* 33, 1: 124–131. 1987.

[13] A.F. Webster, S.E. Tavares, On the design of S-boxes. in *Advances in Cryptology, Proc. of Crypto'85* (H. C. Williams, Ed.), *Lect. Notes Comput. Sci.* 219: 523–534. 1985.

[14] S. Wolfram, *A New Kind of Science*. Wolfram Media Inc., Champaing, IL, 2002.

[15] G.Z. Xiao, J.L. Massey, A spectral characterization of correlation-immune combining functions, *IEEE Trans. Inform. Theory* 34, 3: 569–571. 1988.

Appendix: Mathematica code

In this Appendix the Mathematica code that allows one to obtain the results stated in the paper is shown.

- Local transition function of a CA whose rule number is encoded in the variable `r` as a string of bits:

```
f[r_,var_]:=Module[{x1,x2,x3,x4},
    x1=var[[1]];x2=var[[2]];x3=var[[3]];x4=var[[4]];
    BitXor[
        r[[1]],
        BitAnd[r[[2]],x1],
        BitAnd[r[[3]],x2],
        BitAnd[r[[4]],x3],
        BitAnd[r[[5]],x4],
        BitAnd[r[[6]],x1,x2],
        BitAnd[r[[7]],x1,x3],
        BitAnd[r[[8]],x1,x4],
        BitAnd[r[[9]],x2,x3],
        BitAnd[r[[10]],x2,x4],
        BitAnd[r[[11]],x3,x4],
        BitAnd[r[[12]],x1,x2,x3],
        BitAnd[r[[13]],x1,x2,x4],
        BitAnd[r[[14]],x1,x3,x4],
```

```
        BitAnd[r[[15]],x2,x3,x4],
        BitAnd[r[[16]],x1,x2,x3,x4]
        ]
]
```

- Computation of balanced CA. Their rule numbers are the elements of the variable `balanced`.

```
balanced ={};
Do[
    r=IntegerDigits[k,2,16];
    If[Count[Table[f[r,IntegerDigits[i,2,4]],{i,0,2^4-1}],1]==8,
    AppendTo[balanced,k]]
,{k,0,2^16-1}]
```

- Clasifying CA according to the algebraic degree. The rule CA with algebraic degree k are saved in the variable `ag[`k`]`.

```
(* algebraic degree 4 *)
ag[4]={};
Do[
    aux=Append[IntegerDigits[k,2,15],1];
    AppendTo[ag[4],FromDigits[aux,2]]
,{k,0,2^15-1}];

(* algebraic degree 3 *)
g={};
Do[
    aux1=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{1,0,0,0,0}]],2];
    aux2=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{0,1,0,0,0}]],2];
    aux3=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{0,0,1,0,0}]],2];
    aux4=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{0,0,0,1,0}]],2];
    aux5=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{1,1,0,0,0}]],2];
    aux6=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{1,0,1,0,0}]],2];
    aux7=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{1,0,0,1,0}]],2];
    aux8=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{0,1,1,0,0}]],2];
```

```
    aux9=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{0,1,0,1,0}]],2];
    aux10=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{0,0,1,1,0}]],2];
    aux11=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{1,1,1,0,0}]],2];
    aux12=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{1,1,0,1,0}]],2];
    aux13=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{1,0,1,1,0}]],2];
    aux14=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{0,1,1,1,0}]],2];
    aux15=FromDigits[Flatten[Append[IntegerDigits[k,2,11]
        ,{1,1,1,1,0}]],2];
    Flatten[AppendTo[g,{aux1,aux2,aux3,aux4,aux5,aux6,aux7,aux8
        ,aux9,aux10,aux11,aux12,aux13, aux14,aux15}]]
,{k,0,2^11-1}];
ag[3]=Flatten[g];

(* algebraic degree 1 *)
ag[1]={};
Do[
    aux16=FromDigits[Flatten[Reverse[Append[IntegerDigits[k,2,5]
        ,{0,0,0,0,0,0,0,0,0,0,0}]]],2];
    AppendTo[ag[1],aux16]
,{k,0,2^5-1}]

(* algebraic degree 2 *)
total=Table[k,{k,0,2^16-1}];
aux17=Union[ag[4],ag[3],ag[1],{0}];
ag[2]=Complement[total,aux17];
```

- Computation of the non-linearity. The balanced CA of algebraic degree 3 (*resp*. 2) with non-linearity equals to k is saved in the variable `NlBalGrad3[k]` (*resp*. `NlBalGrad2[k]`).

```
(* Definition of the Hamming weight of a vector *)
wh[w_]:=Count[w,1];

(* Definition of Hamming weight of a boolean function *)
wH[x_]:=Count[
Table[f[IntegerDigits[x,2,16],IntegerDigits[i,2,4]],{i,0,2^4-1}]
,1];

(* Definition of the Hamming distance *)
```

```
dH[x_, y_]:=Module[{z},z=BitXor[x,y];wH[z]];

affin=ag[1];
class=Intersection[balanced,ag[3]];
Do[
    x=class[[j]];
    dis[x]=Table[dH[x,affin[[i]]],{i,1,Length[affin]}];
    nl[x]=Min[dis[x]]
,{j,1,Length[class]}];
Do[NlBalGrad3[k]={},{k,0,6}];
Do[
    x=class[[j]];
    Which[
        nl[x]==6,AppendTo[NlBalGrad3[6],x],
        nl[x]==5,AppendTo[NlBalGrad3[5],x],
        nl[x]==4,AppendTo[NlBalGrad3[4],x],
        nl[x]==3,AppendTo[NlBalGrad3[3],x],
        nl[x]==2,AppendTo[NlBalGrad3[2],x],
        nl[x]==1,AppendTo[NlBalGrad3[1],x],
        nl[x]==0,AppendTo[NlBalGrad3[0],x]
    ]
,{j,1,Length[clase]}];

class=Intersection[balanced,ag[2]];
Do[
    x=class[[j]];
    dis[x]= Table[dH[x,afines[[i]]],{i,1,Length[affin]}];
    nl[x]=Min[dis[x]]
,{j,1,Length[class]}];
Do[NlBalGrad2[k]={},{k,0,6}];
Do[
    x=class[[j]];
    Which[
        nl[x]==6,AppendTo[NlBalGrad2[6],x],
        nl[x]==5,AppendTo[NlBalGrad2[5],x],
        nl[x]==4,AppendTo[NlBalGrad2[4],x],
        nl[x]==3,AppendTo[NlBalGrad2[3],x],
        nl[x]==2,AppendTo[NlBalGrad2[2],x],
        nl[x]==1,AppendTo[NlBalGrad2[1],x],
        nl[x]==0,AppendTo[NlBalGrad2[0],x]
    ]
,{j,1,Length[class]}];
```

- Study of the non-existence of nonzero linear structures. The results are saved in the variables `Nl4BalGrad3LK`, `Nl2BalGrad2LK` and `Nl4BalGrad2LK`.

```
(* Definition of the derivative of a boolean function *)
der[w,b,x]:=
    BitXor[f[IntegerDigits[w,2,16],x],
    f[IntegerDigits[w,2,16],BitXor[b,x]]];


class=NlBalGrad3[4];
Nl4BalGrad3LK={};
Do[
    w=class[[l]];
    LK[w]={};
    Do[
        If[
            Or[Table[der[w,IntegerDigits[i,2,4],IntegerDigits[j,2,4]]
                ,{j,0,2^4-1}]=={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
                ,Table[der[w,IntegerDigits[i,2,4],IntegerDigits[j,2,4]]
                ,{j,0,2^4-1}]=={1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}
            ]
            ,AppendTo[LK[w],i]]
    ,{i,0,2^4-1}];
    If[LK[w]=={0},AppendTo[Nl4BalGrad3LK,w]]
,{l,1,Length[class]}];


class=NlBalGrad3[2];
Nl2BalGrad3LK ={};
Do[
    w =class[[l]];
    LK[w]={};
    Do[
        If[
        Or[Table[der[w,IntegerDigits[i,2,4],IntegerDigits[j,2,4]]
            ,{j,0,2^4-1}]=={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            Table[der[w,IntegerDigits[i,2,4],IntegerDigits[j,2,4]]
            ,{j,0,2^4-1}]=={1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}
        ]
        ,AppendTo[LK[w],i]]
    ,{i,0,2^4 -1}];
    If[LK[w]=={0},AppendTo[Nl2BalGrad3LK,w]]
,{l,1,Length[class]}];
```

```
class=Nl1BalGrad2[4];
Nl4BalGrad2LK={};
Do[
    w=class[[l]];
    LK[w]= {};
    Do[
        If[
            Or[Table[der[w,IntegerDigits[i,2,4],IntegerDigits[j,2,4]]
                ,{j,0,2^4-1}]=={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                Table[der[w,IntegerDigits[i,2,4],IntegerDigits[j,2,4]]
                ,{j,0,2^4-1}]=={1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}
        ]
        ,AppendTo[LK[w],i]]
    ,{i,0,2^4-1}];
    If[LK[w]=={0},AppendTo[Nl4BalGrad2LK,w]]
,{l,1,Length[class]}];
```

- Study of the resiliency. The m-resilient CA rules are saved in the variable `Resilients[m]`.

```
(* Definition of Walsh Transform *)
wt[w_,u_]:=Sum[(-1)^(IntegerDigits[i,2,4].u)*
f[IntegerDigits[w,2,16],IntegerDigits[i,2,4]]
,{i, 0, 2^3 - 1}];
Do[
    vectorsHW[m]={};
    Do[
        If[wh[IntegerDigits[i,2,4]]<=m,
        AppendTo[vectorsHW[m],IntegerDigits[i,2,4]]]
    ,{i, 0,2^4-1}]
,{m,0,2^4}];
Do[
    aux=vectorsHW[m];
    class=Nl4BalGrad3LK;
    Resilients[m]={};
    Do[
        w=class[[j]];
        If[Table[wt[w,aux[[i]]],{i,1,Length[aux]}]==
            Table[0,{i,1,Length[aux]}]
            ,AppendTo[Resilients[m],w]
        ]
    ,{j,1,Length[class]}]
,{m,0,2^4}];
```