The use of Linear Hybrid Cellular Automata as Pseudorandom Bit Generators in Cryptography

C. Fraile Rubio[1], L. Hernández Encinas[2], S. Hoya White[1],
A. Martín del Rey[3] and G. Rodríguez Sánchez[4]
[1]Dpto. Matemática Aplicada, E.T.S.I.I., Universidad de Salamanca
Avda. Fernández Ballesteros s/n, 37700-Béjar, Salamanca, Spain.
E-mails: cjfrailer@terra.es, sarahw@usal.es
[2]Instituto de Física Aplicada, CSIC
C/Serrano 144, 28006-Madrid, Spain.
E-mail: luis@iec.csic.es
[3]Dpto. Matemática Aplicada, E.P.S., Universidad de Salamanca
C/Santo Tomás s/n, 05003-Ávila, Spain.
E-mail: delrey@usal.es
[4]Dpto. Matemática Aplicada, E.P.S., Universidad de Salamanca
Avda. Requejo 33, 49022-Zamora, Spain.
E-mail: gerardo@usal.es

Abstract

The main goal of this paper is to study the behaviour of a particular type of hybrid cellular automata, as cryptographically secure pseudorandom bit generators. The hybrid cellular automata considered have been passed the statistical tests defined in the cryptographic literature to study the security of the sequences generated for cryptographic purposes: frequency test, serial test, poker test, run test and autocorrelation test. Moreover, a study of their dynamical behaviour have been done.

Keywords- Cryptography. Linear Hybrid Cellular Automata. Pseudorandom Number Generators.

1. INTRODUCTION

Random number and random bit generators play an important role in different computer simulation methods such as Monte Carlo techniques, Browmian dynamics, stochastic optimization, computer-based gaming, design and testing of VSLI chips, cryptographic systems, etc. (Niederreiter, 1992).

For example, the security of several cryptosystems depends on the generation of random numbers and random bit sequences. That is the case for the pair of primes in RSA cryptosystem, the secret key of DES and Triple-DES cryptosystems, the private key of DSA digital signature scheme, the keystream of stream cyphers, etc. (Menezes *et al.*, 1997; Stinson, 2002). There exist two methods to produce such quantities: non-deterministic and deterministic algorithms.

The first type uses natural sources of randomness and usually are based on hardware —by using the randomness occuring in some natural physiscal process: elapsed time between emission of particles during a radioactive decay, the frequency instability of a free running oscillator, etc.— and on software —elapsed time between mouse movements, the content of input/output buffers, etc.— Nevertheless, these algorithms are not suitable for cryptographic purposes since the generator must always produce the same output sequence starting from the same initial seed and these non-deterministic procedures do not satisfy this property. That is why all methods used in cryptography are based on deterministic algorithms. Due to this way of generating, these numbers are called pseudorandom numbers.

Specifically, given a short truly random binary sequence of fixed length n (seed), pseudorandom bit generators produce a binary sequence of length $k >> n$ which seems to be random. Such sequences are used to encrypt a plaintext by using stream cyphers in such a way that the binary sequence defined by the plaintext is added, bit by bit, with the bit sequence generated by the pseudorandom bit generator. The secret key is the seed used in the generator. Hence, good random properties of the generator are convenient to prevent statistical attacks but, moreover, it is necessary that the generator must be sure. The security, in this sense, means that the probability that an algorithm can produce in a polynomial time the next bit of a given sequence, is negligible.

In this work, we are interested in the use of linear hybrid cellular automata as pseudorandom bits generators in relation to their cryptographic properties. One-dimensional cellular automata (CA for short) are finite state machines consisting of a finite number of interconnected cells arranged linearly in one dimension, each of which can be in one of a finite number of possible states. Here, we only consider boolean cellular automata, that is, CA whose state set is $\mathbb{Z}_2 = \{0, 1\}$. Every cell essentially

comprises of a memory element built with a D flip-flop and a combinatorial logic that generates the next-state of the cell from the present states of its neighbouring cells. When all cells evolve according to the same logic function, the CA is called uniform, otherwise it is called hybrid. Moreover, linear CA are those whose logic function employs only the XOR gate.

The use of CA to design cryptosystems goes back to middle eighties when S. Wolfram proposed the cellular automaton with rule number 30 as a pseudorandom bit generator (Wolfram, 1986) for cryptographic purposes. Since then, many CA-based cryptosystems have been proposed not only for text (Bardell, 1990; Cattell & Muzio, 1998; Díaz Len *et al.*, 2003; Guan, 1987; Gutowitz, 1993; Nandi, Kar & Chaudhuri, 1994; Tomassini & Perrenoud, 2001; Tomassini *et al.*, 1999) but also for images (Álvarez Marañon *et al.*, 2003; Hernández Encinas *et al.*, 2002).

Most of these works are devoted to the study of cellular automata as cryptographic secure pseudorandom bit generators. Traditionally, only uniform cellular automata have been considered. In this paper, we focus our attention on hybrid CA, and consequently the main goal of this work is to study the pseudorandom properties of linear hybrid cellular automata as cryptographic secure pseudorandom bit generators.

The organization of this paper is as follows. In Section 2, an overview of pseudorandom number generators is presented; in Section 3, the basic concepts of the theory of cellular automata are introduced, focusing our attention on linear hybrid cellular automata; in Section 4, the study of such cellular automata as pseudorandom bit generators is made. Finally, in Section 5 the conclusions are presented.

2. OVERVIEW OF PSEUDORANDOM NUMBER GENERATORS

There exist several ways for generating pseudorandom numbers and pseudorandom sequences of bits —for a general review of them, view (Menezes *et al.*, 1997)—. The most popular are linear congruential generators, lagged-Fibonacci generators and linear feedback shift registers (LFSR).

To assure good pseudorandom properties of a bit sequence, it has to pass several statistical tests —see (Knuth, 1998; Niederreiter, 1992)—. The five basic statistical tests that are usually used for determining whether a sequence of bits possesses some specific features that a truly random sequence would be likely to exhibit are the *frequency test*, the *serial test*, the *poker test*, the *run test* and the *autocorrelation test* (Menezes *et al.*, 1997). They have been developed *ad hoc* for cryptographic use and they are based on Golomb's randomness postulates (Golomb, 1967). Before to introduced these postulates, some basic notations and definitions are shown.

Let $B = \{b_0, b_1, b_2, ..\}$ be an infinite bit sequence. An n-subsequence of B is $B^n = \{b_0, ..., b_{n-1}\}$. A sequence B is said to be *N-periodic* if $b_i = b_{i+N}$, for every $i \geq 0$. Moreover, B is *periodic* if it is N-periodic for some positive integer N. In this case, the *period* of B is the minimun integer number with the last property. If B is periodic of period N, then every subsequence B^N is a *cycle*. Moreover, a *run* of B is a subsequence consisting of consecutive zeros or consecutive ones which is neither preceded nor succeeded by the same symbol. A run of zeros is called *gap*, whereas a run of ones is called *block*. If B is periodic of period N, then the *autocorrelation function* of B is the following integer-valued function:

$$C(t) = \frac{1}{N}\sum_{i=0}^{N-1}(2b_i - 1)(2b_{i+t} - 1), \quad 0 \leq t \leq N-1. \tag{1}$$

This function measures the amount of similarity between the sequence B and a shift of B by t positions. If B is a random periodic sequence of period N, then $|N \cdot C(t)|$ can expected to be quite small for all values of t (Menezes *et al.*, 1997).

Consequently, the three *Golomb's randomness postulates* are the following:

First postulate. In the cycle B^N of B, the number of ones differs from the number of zeros by at most 1.

Second postulate. In the cycle B^N, at least half the runs have length 1, at least one-fourth have length 2, at least one-eighth have length 3, etc., as long as the number of runs so indicated exceeds 1. Furthermore, for each of these lengths, there are almost equally many gaps and blocks.

Third postulate. The autocorrelation function is two-valued. Consequently, there exists an integer k such that:

$$N \cdot C(t) = \sum_{i=0}^{N-1}(2b_i - 1)(2b_{i+t} - 1) = \begin{cases} N & t = 0 \\ k & 1 \leq t \leq N-1 \end{cases} \tag{2}$$

A sequence of bits which satisfy Golomb's postulates is called a *pseudo-noise sequence*. As a consequence the mean features of the tests last mentioned are the following:

1. The *frequency test* has the purpose of determining whether the number of 0's and 1's in the sequence $B = \{b_0, \ldots, b_{n-1}\}$ are approximately the same, as it is expected for a truly random sequence. If n_0, n_1 denotes the number of 0's and 1's is x, respectively, the statistic considered, which follows a χ^2 distribution with 1 degree of freedom if $n \geq 10$, is:

$$X_f = \frac{(n_0 - n_1)^2}{n}. \tag{3}$$

2. The *serial test* tries to determine if the number of pairs 00, 01, 10 and 11 in the sequence x, are approximately the same. If n_{00}, n_{01}, n_{10} and n_{11} are, respectively, the number of such occurrences, the statistic used, which follows a χ^2 distribution with 2 degrees of freedom if $n \geq 21$, is:

$$X_s = \frac{4}{n-1}\left(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2\right) - \frac{2}{n}\left(n_0^2 + n_1^2\right) + 1. \tag{4}$$

3. In the *poker test* the sequence x is divided into k non-overlapping parts of length m, where m is an integer such that $\lfloor n/m \rfloor \geq 5 \cdot 2^m$. Let n_i be the number of occurrences of the ith type of sequences of length m, such that each of them appear the same number of times in x. Then, the statistic considered, which follows a χ^2 distribution with $2^m - 1$ degrees of freedom, is:

$$X_p = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k. \tag{5}$$

4. As the expected number of runs of length i in a random sequence of length n is

$$e_i = \frac{n-i+3}{2^{i+2}}, \tag{6}$$

the purpose of the *run test* is to check if the number of runs of several lengths in x is as expected in a random sequence. Let k be the largest integer i for which $e_i \geq 5$ and let G_i and B_i the number of gaps and blocks of length i such that $1 \leq i \leq k$. The statistic used for this test, which follows a χ^2 distribution with $2k-2$ degress of freedom, is:

$$X_r = \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i}. \tag{7}$$

5. The *autocorrelation test* determines the correlation between the sequence x and non-cyclic shifted versions of x. Let d be an integer such that $1 \leq d \leq \lfloor n/2 \rfloor$. The number of bits in x not equal to their d-shifts is given by

$$A(d) = \sum_{i=0}^{n-d-1} x_i \oplus x_{i+d}. \tag{8}$$

The statistic considered, which follows a $N(0,1)$ distribution if $n - d \geq 10$, is:

$$X_a = \frac{2A(d) - n + d}{\sqrt{n-d}}. \tag{9}$$

3. CELLULAR AUTOMATA

3.1. Basic definitions

One-dimensional cellular automata are discrete dynamical systems consisting of a finite number of identical objects, called *cells*, arranged linearly in one dimension and such a way that every one of them is in any one of a finite number of possible states. These states change in discrete time steps according to a rule, called *local transition function*, such that the state of a cell at the next time step is determined by the current states of a surronding neighborhood of cells.

More precisely, a one-dimensional CA is a 4-uplet $\mathcal{A} = (I, S, V, f)$, where I is the cellular space consisting of a one-dimensional array of n cells. Each cell is denoted by $\langle i \rangle$, $0 \leq i \leq n-1$ (see Figure 1). The finite set S is the set of all possible states of the cells; It is usually given by $\mathbb{Z}_k$. Moreover, $a_i^{(t)}$ stands for the state of the cell $\langle i \rangle$ at time t. The ordered *set of indices* of the CA, $V \subset \mathbb{Z}$, gives the *neighbourhood* of every cell $\langle i \rangle$, consisting of the cells whose states at a time step determine the state of $\langle i \rangle$ at the next time step. In this work we consider symmetric neighbourhoods of radius r, that is, the set of indices is $V = \{-r, \ldots, 0, \ldots, r\}$, and consequently the neighbourhood of the cell $\langle i \rangle$ is given by:

$$V_{\langle i \rangle} = \{\langle i-r \rangle, \ldots, \langle i \rangle, \ldots, \langle i+r \rangle\}, \quad 0 \leq i \leq n-1. \tag{10}$$

Finally, the *local transition function*, $f\colon S^{2r+1} \to S$, determines the evolution of the CA throughout time, *i.e.*, the changes of the states of every cell taking the states of its neighbours into account. As a consequence:

$$a_i^{(t+1)} = f\left(a_{i-r}^{(t)}, \ldots, a_i^{(t)}, \ldots, a_{i+r}^{(t)}\right). \tag{11}$$

As the cellular space is finite, boundary conditions must be considered in order to assure the well-defined evolution of the CA. Here we will stablish two types of boundary conditions:

1. *Periodic boundary conditions.* This type consider $a_i^{(t)} = a_j^{(t)}$ if and only if $i \equiv j \pmod{n}$.

2. *Null boundary conditions.* These conditions make $a_i^{(t)} = 0$ if $i < 0$ or $i > n-1$.

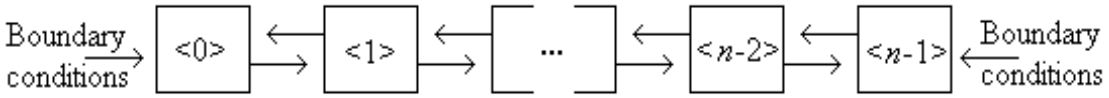

Figure 1. One-dimensional cellular automata with n cells

The set of states of all cells at time t is called the *configuration at time* t of the CA, and it is denoted by $C^{(t)}$. In particular, $C^{(0)}$ is the *initial configuration*. The set of all possible configurations of a CA is denoted by $\mathcal{C}$; if $|S| = k$, then $|\mathcal{C}| = k^n$. Moreover, for every cell $\langle i \rangle$, the vector $\left(a_i^{(0)}, a_i^{(1)}, \ldots, a_i^{(t)}\right)$ is called the *temporal evolution of order* $t+1$ of $\langle i \rangle$.

3.2. Wolfram cellular automata

A particular and very interesting class of CA are *Wolfram cellular automata* —WCA for short— (Wolfram, 1983), for which $S = \mathbb{Z}_2$, the neighbourhoods are symmetric of radius $r = 1$, they have periodic boundary conditions and, consequently, the local transition function is given by the following expression:

$$a_i^{(t+1)} = f\left(a_{i-1}^{(t)}, a_i^{(t)}, a_{i+1}^{(t)}\right), \quad 0 \leq i \leq n-1. \tag{12}$$

As $|S| = 2$ and $|V| = 3$, then there are $2^{2^3} = 256$ WCAs. Each WCA has associated a *Wolfram rule number* w, $0 \leq w \leq 255$, which is defined as follows: There are 8 possible values for the neighbourhoods: $(0,0,0)$, $(0,0,1)$,...,$(1,1,1)$; then for the WCA defined by (12) one has:

$$\begin{aligned} f_0 &= f(0,0,0), & f_1 &= f(0,0,1), \\ f_2 &= f(0,1,0), & f_3 &= f(0,1,1), \\ f_4 &= f(1,0,0), & f_5 &= f(1,0,1), \\ f_6 &= f(1,1,0), & f_7 &= f(1,1,1), \end{aligned} \tag{13}$$

and one can define $w = f_0 \cdot 2^0 + f_1 \cdot 2^1 + \ldots + f_7 \cdot 2^7$. In this way, a WCA with number w will be denoted by $WCA(w)$.

A graphic representation of the evolution of the CA by means of the *evolution diagram* can be obtained. This diagram represents the configurations of the CA in the rows by simple substituting the state 1 by ■, and the state 0 by □.

A very important type of WCA are *linear* WCA in which the next-state generating logic employs only XOR logic operation. As a consequence the algebraic expression of their local transition functions are given by:

$$a_i^{(t+1)} = \alpha a_{i-1}^{(t)} + \beta a_i^{(t)} + \gamma a_{i+1}^{(t)}, \pmod 2, \qquad 0 \leq i \leq n-1, \tag{14}$$

where $\alpha, \beta, \gamma \in \mathbb{Z}_2$. There are eight linear WCA, whose explicit expressions are the

following:

$$
\begin{aligned}
WCA\,(0) &\equiv a_i^{(t+1)} = 0 \pmod 2\,, && (15)\\
WCA\,(60) &\equiv a_i^{(t+1)} = a_{i-1}^{(t)} + a_i^{(t)} \pmod 2\,, && (16)\\
WCA\,(90) &\equiv a_i^{(t+1)} = a_{i-1}^{(t)} + a_{i+1}^{(t)} \pmod 2\,, && (17)\\
WCA\,(102) &\equiv a_i^{(t+1)} = a_i^{(t)} + a_{i+1}^{(t)} \pmod 2\,, && (18)\\
WCA\,(150) &\equiv a_i^{(t+1)} = a_{i-1}^{(t)} + a_i^{(t)} + a_{i+1}^{(t)} \pmod 2\,, && (19)\\
WCA\,(170) &\equiv a_i^{(t+1)} = a_{i+1}^{(t)} \pmod 2\,, && (20)\\
WCA\,(204) &\equiv a_i^{(t+1)} = a_i^{(t)} \pmod 2\,, && (21)\\
WCA\,(240) &\equiv a_i^{(t+1)} = a_{i-1}^{(t)} \pmod 2\,. && (22)
\end{aligned}
$$

The importance of such automata lies in their interpretation in terms of Linear Algebra. Let $WCA\,(w)$ be a linear WCA with periodic boundary conditions and local transition function given by (14). Then, its evolution is given by the following expression:

$$C^{(t+1),T} = M \cdot C^{(t),T} \pmod 2\,, \tag{23}$$

where $C^{(t),T}$ stands for the transpose matrix of $C^{(t)}$, and

$$M = \begin{pmatrix}
\beta & \gamma & 0 & \cdots & 0 & \alpha \\
\alpha & \beta & \gamma & \cdots & 0 & 0 \\
0 & \alpha & \beta & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \beta & \gamma \\
\gamma & 0 & 0 & \cdots & \alpha & \beta
\end{pmatrix} \tag{24}$$

is called the *transition matrix* of $WCA\,(w)$.

Note that in the case of considering null boundary conditions, the $(1,n)$-th coefficient and the $(n,1)$-th coefficient of this matrix are null.

Based on the statistical properties of the dynamics of cellular automata, *i.e.*, the patterns generated during the evolution of the CA from disordered initial configurations, S. Wolfram classified them into four categories (Wolfram, 1983; Wolfram, 1984):

1. Class I: The evolution of such automata leads from almost all initial configurations to a homogeneous final configuration: $\left(1, \overset{(n}{\ldots}, 1\right)$ or $\left(0, \overset{(n}{\ldots}, 0\right)$. Consequently, pattern disappears with time and changes in the initial configuration yield no changes in the final configuration (see Figure 2).

Figure 2. Evolution of Class I CA

2. Class II: The evolution of these CA leads to a set of stable or simple periodic structures (see Figure 3). As a consequence, the reduction of the set of configurations generated by this type of cellular automata is reflected in a decrease in its entropy. Small changes in the initial configuration yield changes only in a region of finite size.

Figure 3. Evolution of Class II CA

3. Class III: This class is formed by all those CA which exhibits a chaotic aperiodic or pseudorandom behaviour (see Figure 4). Consequently, in this type pattern grows indefinitely at a fixed rate (small changes in the initial configuration yield changes over a region of ever-increasing size) and they are specially suitable for pseudorandom number generation.

Figure 4. Evolution of Class III CA

4. Class IV: The evolution of this type of CA yields to complicated localized and propagating structures. As a consequence, this class exhibits more complex behaviour, and is conjectured to be capable of universal computation (see Figure 5).

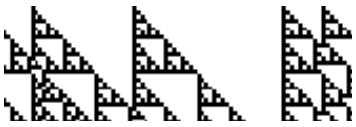

Figure 5. Evolution of Class IV CA

Note that continuous dynamical systems provide similar classes of behaviour seen in cellular automata. The cellular automata of first class may be considered to evolve to limit points; the second class cellular automata may be considered to evolve to limit cycles. Cellular automata of class 3 exhibit chaotic behaviour similar to dynamical systems with strange attractors, and finally, the cellular automata of class 4 have very long transients, and no direct analogue for them has been identified among continuous dynamical systems

3.3. Linear hybrid cellular automata

In the last two subsections we have considered CA in which all cells evolve according to the same local transition rule, for this reason they are called *uniform* CA. Nevertheless, we can consider CA in which the local transition functions are not the same for each cell. In this case, we have *hybrid* CA (or *non-uniform* CA) —HCA for short—.

Here, we study HCA based on the combination of two linear Wolfram local transition functions:

$$WCA\,(u) \equiv a_i^{(t+1)} = \alpha_u a_{i-1}^{(t)} + \beta_u a_i^{(t)} + \gamma_u a_{i+1}^{(t)} \pmod 2\,, \tag{25}$$

$$WCA\,(v) \equiv a_i^{(t+1)} = \alpha_v a_{i-1}^{(t)} + \beta_v a_i^{(t)} + \gamma_v a_{i+1}^{(t)} \pmod 2\,, \tag{26}$$

with $u < v$ and $\alpha_u, \beta_u, \gamma_u, \alpha_v, \beta_v, \gamma_v \in \mathbb{Z}_2$. This linear HCA (LHCA for short) will be denoted by $\{u, v\}$. If the number of cells of the cellular space of this LHCA is n, then it is characterized by the n-upla of bits $(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{n-1})$, in such a way that the cell $\langle i \rangle$ evolves according to $WCA\,(u)$ if $\varepsilon_i = 0$, or according to $WCA\,(v)$ if $\varepsilon_i = 1$. This n-upla is called the *evolution rule* of the LHCA.

As in the uniform case, we can consider every LHCA in terms of Linear Algebra. In particular, the evolution of the n-cell hybrid cellular automata $\{u, v\}$ defined by the local transitions functions (25)-(26) with the evolution vector $(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{n-1})$ and periodic boundary conditions, is defined by:

$$C^{(t+1),T} = M \cdot C^{(t),T} \pmod 2\,, \tag{27}$$

where M is the transition matrix:

$$M = \begin{pmatrix} \beta_{\xi_0} & \gamma_{\xi_0} & 0 & \cdots & 0 & \alpha_{\xi_0} \\ \alpha_{\xi_1} & \beta_{\xi_1} & \gamma_{\xi_1} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_{\xi_{n-1}} & 0 & 0 & \cdots & \alpha_{\xi_{n-1}} & \beta_{\xi_{n-1}} \end{pmatrix}. \tag{28}$$

and

$$\xi_i = (1 - \varepsilon_i)\, u + \varepsilon_i v, \quad 0 \le i \le n-1. \tag{29}$$

For null boundary conditions, the $(1, n)$-th coefficient and the $(n, 1)$-th coefficient of M are null.

4. LHCA AS PSEUDORANDOM NUMBER GENERATORS

In this section, we firstly describe the procedure for generating sequences of bits by means of cellular automata. Moreover, three sieves are presented in order to choose

the CA with better random properties in order to generate pseudorandom sequences of bits.

4.1 The procedure for generating bits

Cellular automata and, particullarly LHCA, can be considered in a very simple way as pseudorandom bit generators. Starting from an initial configuration on length n, $C^{(0)} = \left(a_0^{(0)}, \ldots, a_{n-1}^{(0)}\right)$, it is easy to construct a sequence of bits of length $k \cdot n >> n$ by simple linking together the first k configurations, $C^{(0)}, \ldots, C^{(k-1)}$, of the evolution. Nevertheless, this is not a cryptographic secure procedure: If an adversary obtains a portion (of length greater than l) of such linked sequence, and knows the cellular automata used, it is very easy to generate the rest of configurations given by such automata during its evolution.

More secure bit sequences can be obtained by simply sampling the values that a fixed cell attains in the evolution of the CA; that is, the bit sequence generated by the CA is the temporal evolution of a particular cell $\langle i \rangle$:

$$\left(a_i^{(0)}, a_i^{(1)}, a_i^{(2)}, \ldots\right). \tag{30}$$

Note that this procedure is computationally more expensive than the previous one, but it is also much more hard for an adversary to generate the rest of the sequence knowing only one state of each past configuration, as it is shown in §4.4. Consequently, that is the procedure to generate bits used in this work with initial configurations of 512 bits.

4.2 First sieve: The study of the evolution diagram

As we mentioned above, those CA which exhibit chaotic or pseudorandom behaviour, have good pseudorandom properties. Consequently a first sieve can achieved by using the evolution diagrams of LHCAs of the form $\{u, v\}$ presented in Appendix 1. Taking into account these diagrams, one can classify the LHCAs into the four classes mentioned in §3.2. As a consequence, the LHCA with apparent suitable behaviour as psedorandom bit generators are the same for periodic boundary conditions and null boundary conditions, and they are the following:

$$\begin{aligned} &\{60, 90\}, \{60, 150\}, \{60, 240\}, \{90, 102\}, \{90, 150\}, \{90, 170\}, \\ &\{90, 240\}, \{102, 150\}, \{102, 170\}, \{150, 170\}, \{150, 240\}. \end{aligned} \tag{31}$$

Note that in this sieve, several random initial configurations for every LHCA are considered.

4.3 Second and third sieves: The tests for pseudorandomness

In the second and third sieve, the statistical tests for pseudorandomness (see §2.2) are applied to the LHCA which passed the first sieve. The main features of these sieves are the following:

1. The initial configuration in both cases is formed by 512 cells.

2. The number of analyzed sequences is 100, each one of them of 1000 bits in the second sieve, and 2500 bits in the third sieve.

3. Every test's significance level is 0.05 and the parameters of the poker and autocorrelation tests: m, d, are randomly choosen.

4. Finally, in both sieves we have rejected a LHCA if the number of sequences which not pass any of the tests is bigger than 10.

In the second sieve the results obtained for LHCA with periodic boundary conditions are shown in Table 1, where the values in the columns stand for the number of rejected sequences:

$LHCA$	1th	2th	3th	4th	5th
$\{60, 90\}$	2	6	5	6	4
$\{60, 150\}$	9	3	9	6	6
$\{60, 240\}$	4	5	5	8	6
$\{90, 102\}$	3	5	5	8	3
$\{90, 150\}$	3	4	3	8	3
$\{90, 170\}$	6	6	9	2	3
$\{90, 240\}$	4	5	8	5	5
$\{102, 150\}$	8	9	4	7	8
$\{102, 170\}$	7	2	4	8	6
$\{150, 170\}$	7	5	8	7	5
$\{150, 240\}$	6	6	6	4	2

Table 1. LHCA with periodic boundary conditions (second sieve)

whereas the results for LHCA with null boundary conditions are shown in Table 2:

$LHCA$	1th	2th	3th	4th	5th
$\{60, 90\}$	3	4	3	4	4
$\{60, 150\}$	3	3	5	4	7
$\{60, 240\}$	61	—	—	—	—
$\{90, 102\}$	6	9	8	13	—
$\{90, 150\}$	2	1	3	4	6
$\{90, 170\}$	0	0	0	0	12
$\{90, 240\}$	32	—	—	—	—
$\{102, 150\}$	100	—	—	—	—
$\{102, 170\}$	0	0	0	100	—
$\{150, 170\}$	4	3	5	7	4
$\{150, 240\}$	0	0	0	0	0

Table 2. LHCA with periodic null conditions (second sieve)

Consequently, all LHCA with periodic boundary conditions passed the second sieve, and only five of them with null boundary conditions: $\{60, 90\}$, $\{60, 150\}$, $\{90, 150\}$, $\{150, 170\}$ and $\{150, 240\}$, passed the second sieve.

The third sieve is passed for all LHCA with periodic boundary conditions except of the LHCA $\{150, 170\}$, as it is shown in Table 3:

$LHCA$	1th	2th	3th	4th	5th
$\{60, 90\}$	5	5	5	8	6
$\{60, 150\}$	4	6	3	5	3
$\{60, 240\}$	7	8	6	6	7
$\{90, 102\}$	3	2	3	4	6
$\{90, 150\}$	3	6	5	6	4
$\{90, 170\}$	1	2	3	5	6
$\{90, 240\}$	4	8	7	7	7
$\{102, 150\}$	8	3	2	5	4
$\{102, 170\}$	3	4	6	5	6
$\{150, 170\}$	7	8	7	10	—
$\{150, 240\}$	3	7	5	5	6

Table 3. LHCA with periodic boundary conditions (third sieve)

In the other case, that is, for LHCA with null boundary conditions, three of the five LHCA which passed the second sieve, $\{60, 90\}$, $\{90, 150\}$ and $\{150, 240\}$ also passed

this third sieve:

$LHCA$	1th	2th	3th	4th	5th
$\{60, 90\}$	6	0	6	0	0
$\{60, 150\}$	100	—	—	—	—
$\{60, 240\}$	—	—	—	—	—
$\{90, 102\}$	—	—	—	—	—
$\{90, 150\}$	3	3	6	5	7
$\{90, 170\}$	—	—	—	—	—
$\{90, 240\}$	—	—	—	—	—
$\{102, 150\}$	—	—	—	—	—
$\{102, 170\}$	—	—	—	—	—
$\{150, 170\}$	98	—	—	—	—
$\{150, 240\}$	0	6	0	1	1

Table 4. LHCA with null boundary conditions (third sieve)

4.4 Cryptographic security

Some cryptanalyst attacks based on the algebraic properties of cellular automata (Díaz *et al.*, Meier & Stafelbach, 1992) are efficient if the number of cells of the CA (*i.e.,* the number of states of the initial configuration) is less than 500. As it is mentioned above, in this work the number of cells of the LHCA analysed is equal to 512; consequently, such attacks are avoided.

Moreover, good pseudorandom properties of the bit sequences generated are guaranteed by using the statistical (with cryptographic significance) tests passed in this work.

Furthermore, also "brute force" attacks are avoided as the length of the key is formed by 1024 bits (512 representing the initial configuration and 512 representing the evolution rule of the LHCA) and, consequently there are $2^{1024} \simeq 1.8 \cdot 10^{308}$ possible keys.

5. CONCLUSIONS

We have studied all the 28 linear hybrid cellular automata which are formed by two linear Wolfram cellular automata for their use as pseudorandom bit generators in cryptography. Consequently, we have analyced their pseudorandom properties by means of several statistical tests with cryptographic significance. From the results obtained, we have considered 10 linear hybrid cellular automata with periodic boundary conditions and 3 with null boundary conditions, with good pseudorandom properties.

6. ACKNOWLEDGES

This work has been supported by the Consejería de Educación y Cultura of Junta de Castilla y León (Spain), under the grant SA052/03, and by Ministerio de Ciencia y Tecnología (Spain) under grant TIC2001-0586.

REFERENCES

1. Alvarez Marañón, G., Hernández Encinas A., Hernández Encinas, L., Martín del Rey, A. & Rodríguez Sánchez, G. (2003). Graphics cryptography with pseudorandom bit generators and cellular automata, *Lecture Notes in Artificial Intelligence* v. 2773, pp. 1207–1214.

2. Bardell, P. H. (1990). Analysis of cellular automata used as pseudorandom pattern generators, *Proc. of 1990 International Test Conference*. pp. 762–768.

3. Cattell, K. & Muzio, J.C. (1998). An explicit similarity transform between cellular automata and LFSR matrices, *Finite Fields and Applications* v. 4, pp. 239–251.

4. Díaz, D., Hernández, A., Hernández, L., Hoya, S., Martín. A., Rodríguez, G. & Visus, I. (2003). Wolfram cellular automata and their cryptography use as pseudorandom bit generators, *International Journal of Pure and Applied Mathematics* v.4, pp. 87–103.

5. Golomb, S.W. (1967). *Shift register sequences*. San Francisco, Holden-Day.

6. Guan, P. (1987). Cellular automaton public-key cryptosystem, *Complex Systems* v. 1, pp. 51–57.

7. Gutowitz, H. A. (1993). Cryptography with dynamical systems, *Proc. of the NATO Advanced Study Institute*, pp. 237–274.

8. Hernández Encinas, L., Martín del Rey, M. & Hernández Encinas, A. (2002). Encryption of images with 2-dimensional cellular automata, *Proc. of 6-th Multiconference on Systemics, Cybernetics and Informatics*. pp. 471–476.

9. Knuth, D.E. (1998). *The art of computer programming, Vol. 2. Seminumerical algorithms*. Reading, MA, 3rd ed., Addison-Wesley.

10. Massey, J.L. (1969). Shift-regiser synthesis and BCH decoding, *IEEE Transactions on Information Theory* v. 15, pp. 122–127.

11. Meier, W. & Staffelbach, O. (1992). Analysis of pseudorandom sequences generated by cellular automata, *Lecture Notes in Computer Science* v. 547, pp. 186-189.

12. Menezes, A., van Oorschot, P. & Vanstone, S. (1997). *Handbook of applied cryptography*. Boca Raton, FL., CRC Press.

13. Nandi, S., Kar, B.K. & Chaudhuri, P.P. (1994). Theory and applications of cellular automata in cryptography, *IEEE Transactions on Computers* v. 43, pp. 1346–1357.

14. Niederreiter, N. (1992). *Random number generation and quasi-Monte Carlo methods*. Philadelphia, SIAM.

15. Stinson, D. R. (2002). *Cryptography Theory and Practice, Second Edition*, Boca Raton, FL, CRC Press.

16. Tomassini, M. & Perrenoud, M. (2001). Cryptography with cellular automata, *Applied Software and Computing* v.1, pp. 151–160.

17. Tomassini, M., Sipper, M., Zolla, M. & Perrenoud, M. (1999). Generating high-quality random numbers in parallel by cellular automata, *Future Generation Computer Systems* v. 16, pp. 291–305.

18. Wolfram, S. (1983). Cellular automata, *Los Alamos Science* v. 9, pp. 2–21.

19. Wolfram, S. (1984). Computation theory of cellular automata, *Communications in Mathematical Physics* v. 96, pp. 15–57 .

20. Wolfram, S. (1986). Random sequence generation by cellular automata, *Advances in Applied Mathematics* vol. 7, pp. 123–169.

Appendix 1

Evolution diagrams of the LHCA studied with periodic boundary conditions

Evolution diagrams of the LHCA studied with null boundary conditions