Linear Cellular Automata as Discrete Models for Generating Cryptographic Sequences

A. Fúster-Sabater[1] **P. Caballero-Gil**[2]

[1] Institute of Applied Physics, C.S.I.C.
Serrano 144, 28006 Madrid, Spain
Email: amparo@iec.csic.es

[2] DEIOC,
University of La Laguna,
38271 La Laguna, Tenerife, Spain,
Email: pcaballe@ull.es

Abstract

This work shows that a wide class of cryptographic sequences, the so-called interleaved sequences, can be generated by means of linear multiplicative polynomial cellular automata. In fact, this type of one-dimensional linear 90/150 cellular automata can be devised as generators of pseudo-random sequences. Moreover, these linear automata generate all the solutions of a type of difference equations with constant coefficients. Interleaved sequences are just particular solutions of such equations. In this way, linear discrete models based on cellular automata realize many popular nonlinear sequence generators of current application in stream ciphers. Thus, cryptographic sequence generators conceived and designed originally as complex nonlinear models can be easily written in terms of simple linear equivalents.

Keywords: interleaved sequence, cellular automata, linearization, stream ciphers, cryptography

1 Introduction

Secret-key cryptography is commonly divided into block and stream ciphers. As opposed to block ciphers, stream ciphers encrypt each data symbol (as small as a bit) into a ciphertext symbol under a time-varying transformation. Stream ciphers are the fastest among the encryption procedures so they are implemented in many practical applications e.g. the algorithm RC4 designed by Rivest (1992) for Microsoft Word and Excel, the algorithms A5 in GSM communications GSM (2000) or the encryption system E0 in Bluetooth specifications Bluetooth (2002). From a short secret key (known only by the two interested parties) and a public algorithm (the sequence generator), stream cipher procedure consists in generating a long sequence of seemingly random bits, that is a pseudo-random sequence. In cryptographic terms, such a sequence is called the keystream sequence. For encryption, the sender realizes the bit-wise XOR operation among the bits of the original message or plaintext and the keystream sequence. The result is the ciphertext to be sent. For decryption, the receiver generates the same keystream, realizes the same bit-wise XOR operation between the received ciphertext and the keystream sequence and recovers the original message.

Most keystream generators are based on register machines called Linear Feedback Shift Registers (LFSRs), a good introduction to LFSRs can be found in the reference Golomb (1982). The output sequences of such linear registers are combined by means of nonlinear functions in order to produce keystream sequences of cryptographic application. High linear complexity, long period and good statistical properties are necessary although never sufficient conditions that every keystream sequence must satisfy, see (Caballero-Gil et al. 2004, Fúster-Sabater 2004).

In the literature, there are different families of pseudo-random sequences, the so-called interleaved sequences defined by Gong (1995), with the common characteristic that each sequence can be written in terms of a unique shifted PN-sequence. In fact, a PN-sequence is the output sequence of a Linear Feedback Shift Register (LFSR) with primitive characteristic polynomial. For a survey of PN-sequences, shift equivalence and primitive LFSRs, the interested reader is referred to Golomb (1982).

Interleaved sequences are currently used as keystream sequences and they are generated such as follows:

1. By a LFSR controlled by another LFSR (which may be the same one) e.g. multiplexed sequences Jennings (1983), clock-controlled sequences Beth et al. (1985), cascaded sequences Gollmann et al. (1989), shrinking generator sequences Coppersmith et al. (1994) etc.

2. By one or more than one LFSR and a feedforward nonlinear function e.g. Gold-sequence family, Kasami (small and large set) sequence families, GMW sequences, Klapper sequences, No sequences etc. See the work of Gong (1995) and the references cited therein.

In brief, a large number of popular sequences are included in the class of interleaved sequences. In this work, an easy method of generating interleaved sequences of cryptographic application is presented. Indeed, these sequences are realized by means of Cellular Automata (CA) as solutions of linear difference equations with constant coefficients. The class of linear multiplicative polynomial CA is used in the generation process. In this way, complex nonlinear sequence generators are expressed in terms of simple linear cellular structures.

This work has been done in the frame of the project HESPERIA (http://www.proyecto-hesperia.org) supported by Centro para el Desarrollo Tecnológico Industrial (CDTI) under programme CENIT and also supported by the companies: Soluziona Consultoría y Tecnología, Unión Fenosa, Tecnobit, Visual-Tools, BrainStorm, SAC and TechnoSafe.

More precisely, in the present work, it is showed that one-dimensional linear CA based on rules 90/150 generate all the solutions of linear difference equations with binary constant coefficients. Some of these solutions correspond to the sequences produced by the previous keystream generators. In this way, we have simple CA that not only generate all the solutions of a kind of equation but also they are linear models of nonlinear cryptographic sequence generators. Due to the linearity of the CA transition rules, modelling these CA-based designs is simple and efficient.

2 Fundamentals and Basic Notation

Troughout this work, the following kind of linear difference equations with binary coefficients will be considered:

$$(E^r \oplus \sum_{j=1}^{r} c_j \; E^{r-j}) \; a_n = 0, \qquad n \geq 0 \qquad (1)$$

where E is the shifting operator that operates on a_n (i.e. $Ea_n = a_{n+1}$), $\{a_n\}$ with $a_n \in GF(2)$ (Galois Field) is a binary sequence satisfying the previous equation, $c_j \in GF(2)$ are binary coefficients and the symbol $\oplus$ represents the XOR logic operation. The r-degree characteristic polynomial of the equation (1) is:

$$P(x) = x^r + \sum_{j=1}^{r} c_j \; x^{r-j} \qquad (2)$$

and specifies the linear recurrence relationship of the sequence $\{a_n\}$. This means that its *n-th* term, a_n, can be written as a linear combination of the previous terms:

$$a_n \oplus \sum_{j=1}^{r} c_j \; a_{n-j} = 0, \qquad n \geq r. \qquad (3)$$

If $P(x)$ is an irreducible polynomial Golomb (1982) and α one of its roots, then

$$\alpha, \; \alpha^2, \; \alpha^{2^2}, \ldots, \; \alpha^{2^{(r-1)}} \in GF(2^r) \qquad (4)$$

are the r different roots of such a polynomial (see Lidl et al. (1986)). In this case, the solutions of (1) are of the form (see Key (1976)):

$$a_n = \sum_{j=0}^{r-1} A^{2^j} \; \alpha^{2^j n}, \qquad n \geq 0 \qquad (5)$$

where A is an arbitrary element in $GF(2^r)$. According to equation (5), $\{a_n\}$ is the PN-sequence Golomb (1982) of characteristic polynomial $P(x)$ starting at a particular term given by the value of A.

Next, we can generalize the difference equation given in (1) to a more complex kind such as follows:

$$(E^r \oplus \sum_{j=1}^{r} c_j \; E^{r-j})^p \; a_n = 0, \qquad n \geq 0 \qquad (6)$$

where p is a positive integer. In this case, the characteristic polynomial of (6) is of the form $P_G(x) = P(x)^p$ and its roots will be the same as those of $P(x)$ but with multiplicity p. The solutions of (6) are of the form (see Key (1976)):

$$a_n = \sum_{m=0}^{p-1} \binom{n}{m} (\sum_{j=0}^{r-1} A_m^{2^j} \; \alpha^{2^j n}), \qquad n \geq 0 \qquad (7)$$

where the A_m's are arbitrary elements in $GF(2^r)$. According to equation (7), $\{a_n\}$ is now the bit-wise XOR of p times the same PN-sequence as before $\{\sum_{j=0}^{r-1} A_m^{2^j} \; \alpha^{2^j n}\}$ starting at particular terms determined by the value of A_m and weighted by a binomial coefficient $\binom{n}{m}$. The different choices of the A_m's will give rise to the different sequences $\{a_n\}$ that are all the possible solutions of the equation (6).

3 Interleaved Sequences and Linear Cellular Automata

The two basic structures we are dealing with (interleaved sequences and linear multiplicative polynomial CA) are introduced in the following subsections.

3.1 Fundamentals of the Interleaved Sequences

Let $\boldsymbol{s} = \{s(k)\} = s(0), s(1), \ldots$ be a q-ary linear recurring sequence over $GF(q)$ with $s(k) \in GF(q), k = 0, 1, \ldots$ According to the previous section, the characteristic polynomial of such a sequence $\boldsymbol{s}$ is denoted by

$$f(x) = x^r + \sum_{j=1}^{r} c_j \; x^{r-j} \in GF(q)[x] \qquad (8)$$

and represents its linear recurrence relationship Golomb (1982). Indeed, each element of $\boldsymbol{s}$ can be written as a linear combination of the r previous elements such as follows,

$$s(k+r) = \sum_{j=1}^{r} c_j \; s(k+r-j) \qquad k \geq 0. \qquad (9)$$

In this case $\boldsymbol{s}$ is said to be generated by $f(x)$. The polynomial of the lowest degree in the set of characteristic polynomials of $\boldsymbol{s}$ over $GF(q)$ is called the minimal polynomial of such a sequence.

Definition 1 *Let $f(x)$ be a polynomial over $GF(q)$ of degree r with $f(0) \neq 0$ and let m be a positive integer. For any sequence $\boldsymbol{u} = \{u(k)\}$ over $GF(q)$, write $k = im + j$ $(i = 0, 1, \ldots \; j = 0, \ldots, m-1)$. If $u_j = \{u(im+j)\}_{i \geq 0}$ is generated by $f(x)$ for all j, then $\boldsymbol{u}$ is called an interleaved sequence over $GF(q)$ of size m associated with $f(x)$.*

We can write $\boldsymbol{u} = (\boldsymbol{u}_o, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{m-1})$ where $\boldsymbol{u}_j$'s are called component sequences of $\boldsymbol{u}$. By definition, every $\boldsymbol{u}_j$ is an m-decimation of the sequence $\boldsymbol{u}$. As the LFSRs of cryptographic application have primitive characteristic polynomials, in the sequel $f(x)$ will be a primitive polynomial of degree r in $GF(q)[x]$. In this case, the sequence $\boldsymbol{u}$ is called a *primitive interleaved sequence* and the $\boldsymbol{u}_j$'s are PN-sequences over $GF(q)$ (in fact the same PN-sequence) generated by $f(x)$.

In particular if PN-sequences over $GF(2)$ are considered, then primitive interleaved sequences are characterized by (see Gong (1995)):

1. The period of each $\boldsymbol{u}_j$ is $T = 2^r - 1$, thus the period of the interleaved sequence will be $T_u = m(2^r - 1)$.
2. The minimal polynomial $h(x)$ of $\boldsymbol{u}$ satisfies $h(x) | f(x)^m$ so that the linear complexity of the interleaved sequence (the degree of its minimal polynomial) is upper bound by $LC(\boldsymbol{u}) \leq rm$.

u_0	u_1	u_2	u_3
1	1	1	1
1	0	1	0
0	0	1	1
0	1	0	1
1	0	0	1
0	1	1	0
1	1	0	0

Table 1: Interleaved sequence $\boldsymbol{u}$ with 4 shifted versions of the same PN-sequence

Table 1 shows the interleaved sequence $\boldsymbol{u}$ over $GF(2)$ associated with the 3-degree primitive polynomial $f(x) = x^3 + x + 1$ over $GF(2)$ and size $m = 4$. The period $T_u = m(2^r - 1) = 4(2^3 - 1) = 28$, $h(x) = (x^3 + x + 1)^4$ and $LC(\boldsymbol{u}) = 12$.

Reading by rows, the interleaved sequence is

$$\boldsymbol{u} = \{1,1,1,1,1,0,1,0,0,0,1,1,0,1,0,1,1,0,0,1,0,1,$$
$$1,0,1,1,0,0\}$$

Reading by columns the sequence is made out of four shifted versions of the PN-sequence generated by $f(x)$.

3.2 Linear Multiplicative Polynomial Cellular Automata

CA are particular forms of finite state machines defined as uniform arrays of identical cells in an n-dimensional space Kari (2005). The cells X_i change their states (contents) synchronously at discrete time instants. The next state of each cell depends on the current states of the neighboring cells according to an update *state transition rule*. In particular, linear multiplicative polynomial CA are discrete dynamical systems characterized by Fúster-Sabater et al. (2006):

1. Their underlying topology is one-dimensional, that is they can be represented by a succession of L cells X_i $(i = 1, \ldots, L)$ where L is a positive integer that denotes the length of the automaton. The state of the *i-th* cell at instant n, notated x_i^n, takes values in a finite field $x_i^n \in GF(q)$.

2. They are linear cellular automata as the local transition rule for each cell is a linear mapping $\Phi_i : GF(q)^c \rightarrow GF(q)$ such as follows

   $$x_i^{n+1} = \Phi_i(x_{i-k}^n, \ldots, x_i^n, \ldots, x_{i+k}^n) \;\; (i = 1, ..., L)$$

   where k is a positive integer that denotes the size of the neighborhood and $c = 2k + 1$.

3. Each one of these cellular automata is uniquely represented by an L x L transition matrix M over $GF(q)$. The characteristic polynomial of such matrices is of the form

   $$P_M(x) = P(x)^p \tag{10}$$

   where $P(x)$ is a primitive polynomial of degree r over $GF(q)$ and p is a positive integer. $P_M(x)$ is called the characteristic polynomial of the automaton.

Finally, since the pseudo-random sequences used in communications and cryptography are binary sequences our study is focussed on the finite field $GF(2)$.

4 Realization of Linear Multiplicative Polynomial CA

In the previous sub-section, general characteristics of the multiplicative polynomial CA have been described. Now the particular form of such binary automata is analyzed.

In order to realize this class of CA, transition rules with $k = 3$ will be considered. Thus, there are 2^{2^3} of such rules among which just two (rule 90 and rule 150, see Wolfram (1986) for notation) lead to non-trivial machines. Such rules are defined as:

Rule 90	Rule 150
$x_i^{n+1} = x_{i-1}^n \oplus x_{i+1}^n$	$x_i^{n+1} = x_{i-1}^n \oplus x_i^n \oplus x_{i+1}^n$.

The symbol $\oplus$ denotes addition modulo 2 among cell contents. Remark that such rules are linear, very simple and involve just the addition of either two bits (rule 90) or three bits (rule 150). CA are called *uniform* whether all cells evolve under the same rule while CA are called *hybrid* whether different cells evolve under different rules. At the ends of the array, two different boundary conditions are possible: *null automata* when cells with permanent null contents are supposed adjacent to the extreme cells or *periodic automata* when extreme cells are supposed adjacent. Thus, in this work one-dimensional linear hybrid null automata with rules 90 and 150 are considered.

For a cellular automaton of length $L = 10$ cells, configuration rules:

$$(\,90, 150, 150, 150, 90, 90, 150, 150, 150, 90\,)$$

and initial state:

$$(0,0,0,1,1,1,0,1,1,0),$$

Table 2 illustrates the behavior of this structure: the formation of its output sequences $\{x_i^n\}$ $(i = 1, ..., 10)$ (binary sequences read vertically) as well as the succession of states (binary configurations of 10 bits read horizontally). After a succession of 62 states, this automaton goes back into the initial state. It can be proved that all sequences $\{x_i^n\}$ in a state cycle have the same period, linear complexity and characteristic (minimal) polynomial.

A natural way of specifying 90/150 CA is an L-tuple $\Delta_L = (d_1, d_2, ..., d_L)$, called *rule vector*, where $d_i = 0$ if the *i-th* cell satisfies rule 90 while $d_i = 1$ if the *i-th* cell satisfies rule 150. In the same way, a sub-automaton of the previous automata class consisting of cells 1 through i will be denoted by $\Delta_i = (d_1, d_2, ..., d_i)$. The characteristic matrices M of these CA are tri-diagonal matrices with the rule vector on the main diagonal, 1's on the diagonals below and above the main one and all other entries being zero.

Since the characteristic polynomial of M are of the form $P_M(x) = P(x)^p$, it seems quite natural to construct a multiplicative polynomial cellular automaton by concatenation of the automaton whose characteristic polynomial is $P(x)$. In fact, given $P(x)$ the Cattell and Muzio synthesis algorithm Cattell et al. (1996) presents a method of computing two reversal 90/150 CA corresponding to the given polynomial. That is two different CA whose output sequences have $P(x)$ as characteristic polynomial. The total number of operations required by this algorithm is linear in the degree r of the polynomial Cattell et al. (1996), thus the method is efficient for practical applications. Next, the procedure of concatenation of basic CA is based on the following result.

90	150	150	150	90	90	150	150	150	90
0	0	0	1	1	1	0	1	1	0
0	0	1	0	0	1	0	0	0	1
0	1	1	1	1	0	1	0	1	0
1	0	1	1	1	0	1	0	1	1
0	0	0	1	1	0	1	0	0	1
0	0	1	0	1	0	1	1	1	0
0	1	1	0	0	0	0	1	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	0	0	1	1	0	0	0
0	0	0	0	1	1	0	1	0	0
$\{x_1^n\}$	$\{x_2^n\}$	$\{x_3^n\}$	$\{x_4^n\}$	$\{x_5^n\}$	$\{x_6^n\}$	$\{x_7^n\}$	$\{x_8^n\}$	$\{x_9^n\}$	$\{x_{10}^n\}$

Table 2: A linear 90/150 cellular automaton of 10 cells

Theorem 1 *Let $\Delta_L = (d_1, d_2, ..., d_L)$ be the representation of a 90/150 binary linear cellular automaton with L cells and characteristic polynomial $P_L(x) = (x + d_1)(x + d_2)...(x + d_L)$. The cellular automaton with characteristic polynomial $P_{2L}(x) = P_L(x)^2$ is represented by:*

$$\Delta_{2L} = (d_1, d_2, ..., \overline{d_L}, \overline{d_L}, ..., d_2, d_1) \qquad (11)$$

where the overline symbol represents bit complementation.

Proof. Indeed, it is easy to check that:

$$P_{\overline{L}}(x) = P_L(x) + P_{L-1}(x)$$

where $P_{\overline{L}}(x)$ is the polynomial corresponding to $\Delta_{\overline{L}} = (d_1, d_2, ..., \overline{d_L})$ and $P_{L-1}(x)$ the polynomial corresponding to the sub-automaton Δ_{L-1}. In the same way,

$$P_{L+1}(x) = (x + d_L)P_{\overline{L}}(x) + P_L(x)$$
$$P_{L+2}(x) = (x + d_{L-1})P_{L+1}(x) + P_{\overline{L}}(x)$$
$$\vdots \qquad \vdots$$
$$P_{2L}(x) = (x + d_1)P_{2L-1}(x) + P_{2L-2}(x).$$

Thus, by successive substitutions of the previous polynomial into the next one we get:

$$P_{2L}(x) =$$
$$(x + d_1)P_{2L-1}(x) + P_{2L-2}(x) = P_L(x)^2. \qquad (12)$$

□

The result can be iterated a number of times for successive polynomials and rule vectors:

$$\begin{array}{lcll} P_L(x) & \longleftrightarrow & \Delta_L = (d_1, d_2, ..., d_L) \\ P_L(x)^2 & \longleftrightarrow & \Delta_{2L} = (d_1, d_2, ..., \overline{d_L}, \overline{d_L}, ..., d_2, d_1) \\ P_L(x)^{2^2} & \longleftrightarrow & \Delta_{2^2L} = (d_1, d_2, ..., \overline{d_L}, \overline{d_L}, ..., d_2, \overline{d_1}, \\ & & \qquad \overline{d_1}, d_2, ..., \overline{d_L}, \overline{d_L}, ..., d_2, d_1) \\ \vdots & \longleftrightarrow & \vdots \qquad \vdots \qquad \vdots \end{array}$$

In this way, the concatenation of an automaton (with the least significant bit complemented) and its mirror image allows us to realize linear multiplicative polynomial CA. Remark that the automaton Δ_{2^nL} includes all the previous sub-automata Δ_{2^sL} with $0 \leq s < n$, that is the automaton Δ_{2^nL} generates all the sequences $\{x_i^n\}$ $(i = 1, ..., 2^nL)$ whose characteristic polynomials are $P_L(x)^p$ with $1 \leq p \leq 2^n$. The choice of a particular state cycle determines the corresponding characteristic polynomial $P_L(x)^p$ of its sequences.

5 Generation of Interleaved Sequences from Linear Multiplicative Polynomial CA

In this section, we show how the interleaved sequences can be generated by multiplicative polynomial CA. Depending on the characteristic (minimal) polynomial of the interleaved sequence, different cases can be presented:

Case 1: The characteristic polynomial of the interleaved sequence is

$$h(x) = x^r + \sum_{j=1}^{r} c_j \, x^{r-j}. \qquad (13)$$

Taking $P(x) = h(x)$, $p = 1$ and $P_M(x) = P(x)$, we have that $P_M(x)$ is the characteristic polynomial of a pair of linear multiplicative polynomial CA of $L = r$ cells, that is

$$\Delta_r = (d_1, d_2, ..., d_r)$$

and

$$\Delta_r^* = (d_r, ..., d_2, d_1)$$

given by the Cattell and Muzio algorithm. Furthermore, $P(x) = x^r + \sum_{j=1}^{r} c_j \, x^{r-j}$ specifies the linear recurrence relationship of the sequence $\{x_i^n\}$ obtained at the *i-th* cell $(i = 1, ..., r)$. Hence, such a linear recursion can be expressed as a linear difference equation with constant coefficients in the shifting operator E given by:

$$(E^r + \sum_{j=1}^{r} c_j \, E^{r-j}) \, x_i^n = 0 \qquad n \geq 0 \qquad (14)$$

whose solutions, according to section 2, are of the form $x_i^n = \sum_{j=0}^{r-1} A^{2^j} \, \alpha^{2^j n}$, where $\alpha \in GF(2^r)$ is a root of $P(x)$ and A is an arbitrary element in $GF(2^r)$.

Thus, the sequential solutions $\{x_i^n\}$ of (14) are the *PN*-sequence $\{\sum_{j=0}^{r-1} A^{2^j} \, \alpha^{2^j n}\}$ generated by $P(x)$ starting at a particular point determined by the value of A. Such solutions are realized by the automata Δ_r and Δ_r^*. Each sequence $\{x_i^n\}$ generated by the previous CA is an interleaved sequence of size $m = 1$, period $T = (2^r - 1)$ and linear complexity $LC = r$.

For the 3-degree primitive polynomial $P(x) = x^3 + x + 1$, the Cattell and Muzio algorithm provides us with two reversal 90/150 CA whose rule vectors are $\Delta_3 = (0, 1, 1)$ and $\Delta_3^* = (1, 1, 0)$. Table 3 shows the *PN*-sequence generated by the two reversal CA

starting at the initial state $(1, 1, 0)$. The relative shifts among sequences can be computed efficiently by the algorithm given in Cho et al. (2004).

90	150	150	150	150	90
1	1	0	1	1	0
1	0	1	0	0	1
0	0	1	0	1	0
0	1	1	1	1	1
1	0	0	0	1	1
0	1	0	1	0	1
1	1	1	1	0	0
$\{x_1^n\}$	$\{x_2^n\}$	$\{x_3^n\}$	$\{x_1^n\}$	$\{x_2^n\}$	$\{x_3^n\}$

Table 3: Two reversal 90/150 cellular automata generating the same PN-sequence

Case 2: The characteristic polynomial of the interleaved sequence is

$$h(x) = (x^r + \sum_{j=1}^{r} c_j \; x^{r-j})^{2^s} \qquad (15)$$

where s is a positive integer. Taking $P(x) = x^r + \sum_{j=1}^{r} c_j$, $p = 2^s$ and $P_M(x) = P(x)^p$, we have that $P_M(x)$ is the characteristic polynomial of a pair of linear multiplicative polynomial CA of $L = 2^s \cdot r$ cells, notated $\Delta_{2^s \cdot r}$ and $\Delta^*_{2^s \cdot r}$, obtained by concatenating s times each one of the basic automata Δ_r and Δ^*_r, respectively. Furthermore, $P_M(x)$ specifies the linear recurrence relationship of the sequence $\{x_i^n\}$ obtained at the *i-th* cell ($i = 1, ..., 2^s \cdot r$). Hence, the corresponding linear difference equation is:

$$(E^r + \sum_{j=1}^{r} c_j \; E^{r-j})^p \; x_i^n = 0 \qquad n \geq 0 \qquad (16)$$

whose solutions, according to section 2, are of the form $x_i^n = \sum_{i=0}^{p-1} \binom{n}{i} \; (\sum_{j=0}^{r-1} A_i^{2^j} \; \alpha^{2^j n})$, where $A_i \in GF(2^r)$ with $A_{p-1} \neq 0$.

Thus, the sequential solutions $\{x_i^n\}$ of (16) are the bit-wise addition modulo 2 of p different sequences made out of the PN-sequence $\{\sum_{j=0}^{r-1} A^{2^j} \; \alpha^{2^j n}\}$ generated by $P(x)$ starting at a particular point determined by the value of A_i and weighted by a binomial number $\binom{n}{i}$. Such solutions are generated by the automata $\Delta_{2^s \cdot r}$ and $\Delta^*_{2^s \cdot r}$. Each sequence $\{x_i^n\}$ generated by the previous CA is an interleaved sequence of size $m = 2^s$, period $T = m(2^r - 1)$ and linear complexity $LC = rm$.

For the polynomial $P_M(x) = (x^3 + x + 1)^4$, the two 90/150 CA obtained are:

$$\Delta_{12} = (0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0)$$

and

$$\Delta^*_{12} = (1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1)$$

Table (4: left) shows the sequences generated by Δ_{12} with $T = 28$ and $LC = 12$ starting at the initial state $IS_2 = (1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0)$. Table (4: right) shows the extreme sequence $\{x_1^n\} = \{1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0\}$ of the automaton in interleaved format of size $m = 4$.

Case 3: The characteristic polynomial of the interleaved sequence is

$$h(x) = (x^r + \sum_{j=1}^{r} c_j \; x^{r-j})^p \qquad 2^{s-1} < p \leq 2^s. \quad (17)$$

Taking $P(x) = x^r + \sum_{j=1}^{r} c_j$, $2^{s-1} < p \leq 2^s$ and $P_M(x) = P(x)^p$, now we have that the pair of linear multiplicative polynomial CA are the same as before, that is $\Delta_{2^s \cdot r}$ and $\Delta^*_{2^s \cdot r}$. The sequences $\{x_i^n\}$ are generated in a state cycle whose polynomial is $P(x)^p$. In this case, each sequence $\{x_i^n\}$ is an interleaved sequence of size $m = 2^s$, period $T = m(2^r - 1)$ and linear complexity $LC = pm < rm$. For the polynomial $P_M(x) = (x^3 + x + 1)^3$, Table (5: left) shows the sequences generated by Δ_{12} with $T = 28$ and $LC = 9$ starting at the initial state $IS_3 = (0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1)$. Table (5: right) shows the extreme sequence $\{x_1^n\} = \{0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1\}$ of the automaton in interleaved format of size $m = 4$.

All the solutions of the linear difference equations with constant coefficients given in (16) can be realized by linear multiplicative polynomials CA. The interleaved sequences are just particular solutions of such equations. Remark that the CA here considered are linear models while the cryptographic generators of interleaved sequences are based on nonlinear functions. Thus, a simple linearization process has been carried out.

6 Conclusions

All the sequential solutions of linear constant coefficient difference equations can be realized by means of linear models based on cellular automata. Some of these solutions have a straight cryptographic application in stream ciphers. In fact, interleaved sequences are just particular solutions of these linear difference equations and can be generated by linear multiplicative polynomial CA. In this way, very popular cryptographic sequence generators conceived and designed as nonlinear generators can be linearized in terms of cellular automata. The linearization process is simple and can be applied to cryptographic examples in a range of practical application.

References

Beth, T. & Piper, F. (1984), The Stop-and-Go Generator, *in* 'Advances in Cryptology: EUROCRYPT'84', Vol. 228, Lecture Notes in Computer Science, Springer Verlag, pp. 197–207.

Bluetooth (2002), 'Specifications of the Bluetooth system' available at http://www.bluetooth.com/

Caballero-Gil, P. & Fúster-Sabater, A. (2004), 'A Wide Family of Nonlinear Filter Functions with a Large Linear Span', *Information Sciences* **164**(4), 197–207.

Cattell, K. & Muzio, J.C. (1996), 'Synthesis of One-Dimensional Linear Hybrid Cellular Automata', *IEEE Trans. Computers-Aided Design* **15**(3), 325–335.

Cho, S. J., Piatetsky-Shapiro, G. & Smyth, P. (2004), Computing Phase Shifts of Maximum-Length 90/150 Cellular Automata Sequence, *in*

90	150	90	90	150	150	150	150	90	90	150	90	$\boldsymbol{u}_0$	$\boldsymbol{u}_1$	$\boldsymbol{u}_2$	$\boldsymbol{u}_3$
1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	1
1	1	1	0	0	0	0	0	1	1	0	0	1	0	1	0
1	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1
1	1	0	1	1	0	1	0	0	0	0	1	0	1	0	1
1	0	0	1	0	0	1	1	0	0	1	0	1	0	0	1
0	1	1	0	1	1	0	0	1	1	1	1	0	1	1	0
1	0	1	0	0	0	1	1	1	0	1	1	1	1	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮				
0	0	1	0	1	1	1	0	0	0	0	0				
0	1	0	0	0	1	0	1	0	0	0	0				
$\{x_1^n\}$		...			...	...			...		$\{x_{12}^n\}$				

Table 4: (left) Automaton Δ_{12} starting at IS_2; (right) Sequence $\{x_1^n\}$ in interleaved format

90	150	90	90	150	150	150	150	90	90	150	90	$\boldsymbol{u}_0$	$\boldsymbol{u}_1$	$\boldsymbol{u}_2$	$\boldsymbol{u}_3$
0	1	0	0	0	1	0	1	0	0	0	1	0	1	1	1
1	1	1	0	1	1	0	1	1	0	1	0	1	1	0	1
1	1	1	0	0	0	0	0	1	0	1	1	0	0	0	0
1	1	1	1	0	0	0	1	0	0	0	1	1	0	1	0
1	1	0	1	1	0	1	1	1	0	1	0	1	1	0	1
1	0	0	1	0	0	0	1	1	0	1	1	1	0	1	0
0	1	1	0	1	0	1	0	1	0	0	1	0	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮				
1	1	0	1	0	1	1	1	1	0	0	1				
1	0	0	0	0	0	1	1	1	1	1	0				
$\{x_1^n\}$		...			...	...			...		$\{x_{12}^n\}$				

Table 5: (left) Automaton Δ_{12} starting at IS_3; (right) Sequence $\{x_1^n\}$ in interleaved format

'Advances in Cryptology ACRI', Vol. 3305, Lecture Notes in Computer Science, Springer Verlag, Amsterdam, The Netherlands, pp. 31–39.

Coppersmith, D., Krawczyk, H. & Mansour, Y. (1994), The Shrinking Generator, *in* 'Advances in Cryptology:CRYPTO'93', Vol. 773, Lecture Notes in Computer Science, Springer Verlag, Santa Barbara California, USA, pp. 22–39.

Fúster-Sabater, A. (2004), 'Run Distribution in Nonlinear Binary Generators', *Applied Mathematics Letters* **17**(12), 1427–1432.

Fúster-Sabater, A. & Caballero-Gil, P. (2006), Linear Automata in Cryptanalysis of Stream Ciphers, *in* 'Advances in Cryptology ACRI', Vol. 4173, Lecture Notes in Computer Science, Springer Verlag, Perpignan, France, pp. 611–616.

Gollmann, D. & Chambers, W.G. (1989), 'Clock-Controlled Shift Registers: A Review', *IEEE Journal on Selected Areas in Communications* **7**(4), 525–533.

Golomb, S. (1986), *Shift-Register Sequences*, Aegean Park Press, Laguna Hill California.

Gong, G. (1995), 'Theory and Applications of q-ary Interleaved Sequences', *IEEE Trans. Information Theory* **41**(2), 400–411.

GSM (2000), 'Global Systems for Mobile Communications' available at http://cryptome.org/gsm-a512.htm

Jennings, S. M. (1983), Multiplexed Sequences: Some Properties, *in* 'Advances in Cryptology EUROCRYPT'83', Vol. 149, Lecture Notes in Computer Science, Springer Verlag, pp. 61–76.

Kari, J. (2005), 'Theory of cellular automata: A survey', *Theoretical Computer Science* **334**(1), 3–33.

Key, E.L. (1976), 'An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators', *IEEE Trans. Info. Theory* **22**(11), 247–298.

Lidl, R. & Niederreiter, H. (1986), *Introduction to Finite Fields and Their Applications*, Cambridge, England: Cambridge University Press.

Rivest, R.L. (1992), 'The RC4 Encryption Algorithm', Internal Report, RSA Data Security, Inc., March 12.

Wolfram, S. (1986), 'Random Sequence generation by Cellular Automata', *Advances in Applied Mathematics* **123**(7), 67–93.