Efficient Application of Hybrid 150/90 Cellular Automata to Symmetric Cryptography*

A. Fúster-Sabater¹, P. Caballero-Gil², and M.E. Pazo-Robles¹

¹ Instituto de Física Aplicada, C.S.I.C.
Serrano 144, 28006 Madrid, Spain
amparo@iec.csic.es

² DEIOC, University of La Laguna
38271 La Laguna, Tenerife, Spain
pcaballe@ull.es

Abstract. In this work, it is shown that a wide class of nonlinear sequence generators, the so called interleaved sequence generators, can be modelled in terms of linear cellular automata. A simple modelling procedure based on the concatenation of automata has been derived. The cryptographic characteristics of the generated sequences (period, linear complexity, number of different sequences) have been also analyzed. The technique is very simple and can be applied to generators in a range of practical applications.

Keywords: Linear automata, sequence generator, symmetric cryptography.

1 Introduction

Sensitive information to be transmitted or stored requires certain characteristics of confidentiality. Such a quality makes use of an encryption function currently called *cipher* that converts the *plaintext* into the *ciphertext*. Symmetric cryptography are usually divided into two large classes: stream ciphers and block-ciphers. Stream ciphers are the fastest among the encryption procedures so they are implemented in many technological applications e.g. algorithms A5 in GSM communications [8] or the encryption system E0 used in the Bluetooth specifications [1]. Stream ciphers try to imitate the ultimate one-time pad cipher and are designed to stretch a short secret seed (the secret key) into a long sequence (keystream) of seemingly random bits. This keystream is then XORed with the plaintext in order to obtain the ciphertext. Most keystream generators are based on Linear Feedback Shift Registers (LFSRs) [6] combined in a nonlinear way. Such generators produce keystreams with high linear complexity, long period and good statistical properties.

On the other hand, bit sequences generated by one-dimensional binary Cellular Automata (CA) with three site neighborhood and linear transition rules

* Work supported by Ministerio de Educación y Ciencia (Spain) Projects SEG2004-02418 and SEG2004-04352-C04-03.

[2] have been found to be isomorphic to conventional LFSRs. Thus, the latter structures can be simply substituted by the former ones in order to accomplish the same goal: generation of keystreams with application in symmetric cryptography. Nevertheless, the main advantage of these CA is that multiple generators designed in terms of LFSRs as *nonlinear* structures preserve the *linearity* when they are expressed under the form of CA.

The question that arises in a natural way is: are there one-dimensional linear CA able to produce the sequence obtained from any LFSR-based nonlinear generator? The answer is yes and, in fact, this paper stresses the problem of given a particular LFSR-based generator how to find one-dimensional CA that reproduce exactly their output sequences. More precisely, in this work it is shown that a wide class of LFSR-based nonlinear generators, the so called interleaved sequence generators, can be modelled in terms of linear CA. Among this class of generators we can enumerate: Clock-Controlled generators, Cascade-Clock-Controlled generators, Shrinking generator or the generators producing Kasami sequences, GMW sequences, No sequences, Klapper *et al.* sequences ... (see [7]).

Interleaved sequences [7] satisfy a common property: they can be decomposed into a collection of shifts of an unique *PN*-sequence and zero sequences. This kind of sequences can be obtained from linear multiplicative polynomial CA. That is to say, CA made out of a basic structure concatenated a number of times. Therefore, the goal of this work is double:

- To analyze linear multiplicative polynomial CA with emphasis on the cryptographic parameters of their generated sequences (period, linear complexity, characteristic polynomial, number of different output sequences etc.).
- To model the above nonlinear LFSR-based generators in terms of linear multiplicative polynomial CA.

Once the generators have been linearized, all the theoretical background on linear CA can be applied to their analysis and/or cryptanalysis.

2 Fundamentals and Basic Notation

Our attention is concentrated on one-dimensional binary hybrid linear CA with three neighborhood sites and rules 90 and 150. Both rules can be defined as follows:

Rule 90 $x_{n+1}^k = x_n^{k-1} + x_n^{k+1}$	Rule 150 $x_{n+1}^k = x_n^{k-1} + x_n^k + x_n^{k+1}$
--	---

Indeed, x_{n+1}^k the content of the k -th cell at time $n+1$ depends on the content of either two different cells (rule 90) or three different cells (rule 150) at time n , for $(k = 1, \dots, L)$ where L is the length of the automaton. A natural form of automaton representation is an L -tuple $\Delta = (d_1, d_2, \dots, d_L)$ where $d_k = 0$ if the k -th cell verifies rule 90 while $d_k = 1$ if the k -th cell k verifies rule 150.

For an one-dimensional null hybrid cellular automaton of length $L = 10$ cells, configuration rules (90, 150, 150, 150, 90, 90, 150, 150, 150, 90) and initial state

Table 1. An one-dimensional null hybrid linear cellular automaton of 10 cells with rule 90 and rule 150 starting at a given initial state

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
1	0	0	1	0	0	1	1	0	0
0	1	1	1	1	1	0	0	1	0
1	0	1	1	0	1	1	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(0, 0, 0, 1, 1, 1, 0, 1, 1, 0), Table 1 illustrates the formation of its output sequences (binary sequences read vertically) and the succession of states (binary configurations of 10 bits read horizontally). For the previous rules, the different states of the automaton are grouped in closed cycles [5] as well as any output sequence can be generated at any cell provided that the right state cycle is chosen.

Next, some important definitions are introduced.

Definition 1. A *Multiplicative Polynomial Cellular Automaton* is defined as a cellular automaton whose characteristic polynomial is of the form $P_M(X) = (P(X))^p$ where p is a positive integer. If $P(X)$ is a primitive polynomial, then the automaton is called a *Primitive Multiplicative Polynomial Cellular Automaton*.

A widely accepted measure of the unpredictability of a sequence is its *linear complexity* [9] that can be defined as follows:

Definition 2. The *linear complexity* of a sequence is the shorter linear recursion satisfied by such a sequence.

For a binary sequence $\{x_n\}$, the linear recurrence relationship that specifies the n -th element as a combination of the r previous ones can be written:

$$x_n + \sum_{i=1}^r c_i x_{n-i} = 0, \quad n \geq r \quad (1)$$

where the sequence elements (x_n) as well as the coefficients (c_i) belong to a finite field, $GF(2)$. The linear recursion can be expressed as a linear difference equation:

$$(E^r + \sum_{i=1}^r c_i E^{r-i}) x_n = 0, \quad n \geq 0 \quad (2)$$

where E is the shifting operator that operates on x_n , i.e. $Ex_n = x_{n+1}$. The characteristic polynomial of the difference equation (2) is:

$$P(X) = X^r + \sum_{i=1}^r c_i X^{r-i}. \quad (3)$$

Let $P(X)$ be a primitive polynomial of degree r and $\alpha \in GF(2^r)$ one of its roots. In this case [9],

$$\alpha, \alpha^2, \alpha^{2^2}, \dots, \alpha^{2^{(r-1)}} \quad (4)$$

are the r roots of such a polynomial. If $P_M(X) = (P(X))^p$, then the roots of $P_M(X)$ will be the same as those of $P(X)$ but with multiplicity p . Consequently, all the binary sequences generated by a primitive multiplicative polynomial cellular automaton will satisfy the linear difference equation:

$$(E^r + \sum_{i=1}^r c_i E^{r-i})^p x_n = 0, \quad n \geq 0. \quad (5)$$

Our analysis focuses on all the possible solutions $\{x_n\}$ of this equation. For a particular sequence generated at the k -th cell, the notation will be $\{x_n^k\}$.

3 Cryptographic Properties of Multiplicative Polynomial CA

In this section, the characteristics of the sequences obtained from this kind of CA are considered in terms of their cryptographic parameters.

3.1 Period of the Generated Sequences

The solutions of the equation (5) are linear combinations of $p \cdot r$ solutions of the form:

$$x_n = \sum_{i=0}^{p-1} \binom{n}{i} \sum_{j=0}^{r-1} A_i^{2^j} \alpha^{2^j n}, \quad (6)$$

where $S_n^i = \sum_{j=0}^{r-1} A_i^{2^j} \alpha^{2^j n}$ represents the n -th element of a PN -sequence [9] of period $2^r - 1$ where $A_i \in GF(2^r)$. Thus, the binary sequence $\{x_n\}$ can be written as the sum of p times the same PN -sequence weighted by a binomial coefficient

$$\{x_n\} = \sum_{i=0}^{p-1} \binom{n}{i} \{S_n^i\}. \quad (7)$$

In addition, each binomial coefficient defines a succession of binary values with a constant period p_i . Therefore, the sequence $\{x_n\}$ is the sum of p sequences of distinct periods $T_i = p_i \cdot (2^r - 1)$ and the period of the total sequence will be:

$$T = \max\{T_i / A_i \neq 0 \ (i = 0, \dots, p-1)\}. \quad (8)$$

It can be noticed that the period of the different sequences $\{x_n\}$ generated by a multiplicative polynomial cellular automaton depends on the choice of the coefficients A_i in the equation (6). All sequences generated at the same state cycle have the same period. Nevertheless, the same automaton can produce sequences with distinct periods depending on the state cycle considered.

3.2 Linear Complexity of the Generated Sequences

According to [9], the linear complexity of a sequence equals the number and multiplicity of characteristic polynomial roots that appears in the linear recurrence relationship. Therefore, coming back to the equation (6), the linear complexity of $\{x_n\}$ can be computed. In fact, we have r roots each of them with multiplicity p . Thus, if i_{max} is the greatest value of i ($i = 0, \dots, p-1$) for which $A_i \neq 0$, then the linear complexity LC of the sequence $\{x_n\}$ will be:

$$LC = (i_{max} + 1) \cdot r. \quad (9)$$

It can be noticed that the linear complexity of the different sequences $\{x_n\}$ generated by a multiplicative polynomial cellular automaton depends on the choice of the coefficients A_i in (6). All sequences generated at the same state cycle have the same linear complexity. Nevertheless, the same automaton can produce sequences with distinct linear complexities depending on the state cycle considered.

3.3 The Number of Different Generated Sequences

In order to count the number of different sequences $\{x_n\}$ generated, the choice of the coefficients A_i in the equation (6) must be considered. Different situations can take place:

- If $A_i = 0 \ \forall i$, then all the cells of the cellular automaton will generate the identically null sequence.
- If $A_0 \neq 0$ and $A_i = 0 \ \forall i > 0$, then all the cells of the cellular automaton will generate a unique PN -sequence $\{S_n^0\}$ of period $T_0 = 2^r - 1$ and characteristic polynomial $P(X)$.
- If $A_0 \in GF(2^r)$, $A_1 \neq 0$ and $A_i = 0 \ \forall i > 1$, then there will be $2^r \cdot (2^r - 1)$ possible choices of (A_0, A_1) . According to subsection 3.1, the period of these sequences will be $T_1 = p_1 \cdot (2^r - 1)$. Thus, the number of different sequences for these values of A_i is:

$$N_1 = \frac{2^r \cdot (2^r - 1)}{p_1 \cdot (2^r - 1)} = 2^{r-1}. \quad (10)$$

- In general, if $A_0, A_1, \dots, A_{i-1} \in GF(2^r)$, $A_i \neq 0$ and $A_j = 0 \ \forall j > i$, then there will be $2^{(i \cdot r)} \cdot (2^r - 1)$ possible choices of $(A_0, A_1, \dots, A_i)$. According to subsection (3.1), the period of these sequences will be $T_i = p_i \cdot (2^r - 1)$. Thus, the number of different sequences for these values of A_i is:

$$N_i = \frac{2^{(i \cdot r)} \cdot (2^r - 1)}{T_i} = \frac{2^{(i \cdot r)}}{p_i}. \quad (11)$$

The total number of different sequences obtained from a multiplicative polynomial cellular automaton will be:

$$N_{total} = \sum_{i=0}^{p-1} N_i. \quad (12)$$

The null sequence is excluded.

4 Constructing Keystream Generators by Concatenation of Primitive CA

In the previous section, structural properties of the sequences obtained from multiplicative polynomial cellular automata have been considered. Now the particular form of these automata is analyzed. Since the characteristic polynomial of such automata is $P_M(X) = (P(X))^p$, it seems quite natural to construct a multiplicative polynomial cellular automaton by concatenating p times the automaton whose characteristic polynomial is $P(X)$. The procedure of concatenation is based on the following result.

Lemma 1. *Let $\Delta = (d_1, d_2, \dots, d_k)$ be the representation of an one-dimensional binary hybrid linear cellular automaton with k cells and characteristic polynomial $P_k(X) = (X + d_1)(X + d_2)\dots(X + d_k)$. The cellular automaton whose characteristic polynomial is $P_{2k}(X) = (P_k(X))^2$ is represented by:*

$$\Delta = (d_1, d_2, \dots, \overline{d_k}, \overline{d_k}, \dots, d_2, d_1). \quad (13)$$

Sketch of proof. The result follows from the fact that [2]:

$$P_{\overline{k}}(X) = P_k(X) + P_{k-1}(X)$$

where $P_{\overline{k}}(X)$ is the polynomial corresponding to $\Delta = (d_1, d_2, \dots, \overline{d_k})$. In the same way

$$\begin{aligned} P_{k+1}(X) &= (X + d_k)P_{\overline{k}}(X) + P_k(X) \\ P_{k+2}(X) &= (X + d_{k-1})P_{k+1}(X) + P_{\overline{k}}(X) \\ &\vdots \\ P_{2k}(X) &= (X + d_1)P_{2k-1}(X) + P_{2k-2}(X). \end{aligned}$$

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

$$P_{2k}(X) = (X + d_1)P_{2k-1}(X) + P_{2k-2}(X) = (P_k(X))^2. \quad \square$$

In this way, the construction of a linear model based on CA is carried out by the following algorithm:

Input: The parameters of a nonlinear interleaved sequence generator.

- *Step 1:* Determine the irreducible factor $P(X)$ of the characteristic polynomial of the interleaved sequence.
- *Step 2:* Compute the pair of basic CA [2] whose characteristic polynomial is the irreducible factor $P(X)$.
- *Step 3:* For each one of these basic CA, construct by successive concatenations the longer cellular automaton able to generate the original interleaved sequence.

Output: Two linear CA producing the corresponding interleaved sequence.

Let us see a simple example of application. The shrinking generator [4] is a typical example of cryptographic generator with characteristic polynomial $P_M(X)$. The characteristics of this generator can be summarized as follows: The generator is made of two LFSRs, SR1 and SR2, with lengths $L_j (j = 1, 2)$ and feedback polynomials $P_j(X) (j = 1, 2)$, respectively. The decimation rule is: the bit produced by SR2 is discarded if the corresponding bit of SR1 equals 0. The decimated sequence is just the output sequence of the generator. The generator characteristic polynomial is of the form $P_M(X) = (P(X))^p$, $P(X)$ being a primitive polynomial of degree $r = L_2$ and $2^{(L_1-2)} < p \leq 2^{(L_1-1)}$. Moreover, $P(X)$ is the characteristic polynomial of the cyclotomic coset E in $GF(2^{L_2})$ with $E = 2^0 + 2^1 + \dots + 2^{L_1-1}$, see ([6]).

The output sequence will be a solution of a linear difference equation corresponding to primitive multiplicative polynomial CA. Consequently, the shrinking generator can be expressed in terms of a linear model based on CA. A numerical example illustrates the modelling procedure.

Input: A shrinking generator characterized by two LFSRs of lengths $L_1 = 3$, $L_2 = 5$ respectively and characteristic polynomial $P_2(X) = X^5 + X^4 + X^3 + X + 1$. Now, $p = 2^{L_1-1} = 4$ and $r = L_2 = 5$.

Step 1: $P(X)$ the irreducible factor of the generator characteristic polynomial is:

$$P(X) = X^5 + X^3 + 1.$$

Step 2: The pair of basic CA whose characteristic polynomial is $P(X)$ are:

$$\begin{array}{c} 0 \ 1 \ 1 \ 0 \ 0 \\ 0 \ 0 \ 1 \ 1 \ 0 \end{array}$$

Step 3: Computation of the required pair of CA by successive concatenations.
For the first automaton:

$$\begin{array}{c} 0 \ 1 \ 1 \ 0 \ 0 \\ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \\ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \text{ (final automaton)} \end{array}$$

For the second automaton:

$$\begin{array}{c} 0 \ 0 \ 1 \ 1 \ 0 \\ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \\ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \text{ (final automaton)} \end{array}$$

For each automaton, the procedure in *Step 3* has been carried out $L_1 - 1$ times. In fact, each basic complemented automaton has been concatenated $p = 2^{L_1-1}$ times.

Output: Two binary strings codifying the required CA.

In this way, we have obtained a pair of linear CA able to produce, among other sequences, the interleaved sequence corresponding to the given shrinking generator. Analogous procedure applies for any of the keystream generators mentioned in section 1.

5 Conclusions

In this work, cryptographic properties of the primitive multiplicative polynomial CA have been analyzed. It is shown that a wide class of LFSR-based sequence generators (interleaved sequence generators) can be described in terms of CA-based structures. In this way, sequence generators conceived and designed as complex nonlinear models can be written in terms of simple linear models. The linearity of these cellular models can be advantageously used in the analysis and/or cryptanalysis of such cryptographic generators.

References

1. Bluetooth, *Specifications of the Bluetooth system*, Version 1.1, February 2001, available at <http://www.bluetooth.com/>
2. K. Cattell *et al.* Synthesis of One-Dimensional Linear Hybrid Cellular Automata, *IEEE Trans. Computers-Aided Design*, Vol. 15, No. 3, pp. 325-335, 1996.
3. S. J. Cho *et al.* Computing Phase Shifts of Maximum-Length 90/150 Cellular Automata Sequences. *Proc. of ACRI 2004. Lecture Notes on Computer Science*, Springer-Verlag, Vol. 3305, pp. 31-39, 2004.
4. D. Coppersmith, H. Krawczyk and Y. Mansour, The Shrinking Generator. *Proc. of CRYPTO'93. Lecture Notes in Computer Science*, Springer Verlag, Vol. 773, pp. 22-39, 1994.
5. A. K. Das, A. Ganguly, A. Dasgupta, S. Bhawmik and P. P. Chaudhuri, Efficient Characterization of Cellular Automata. *IEE Proc.*, Part E. **1**, pp. 81-87, 1990.
6. S.W. Golomb, *Shift Register-Sequences*, Aegean Park Press, Laguna Hill, 1982.
7. G. Gong, Theory and Applications of q-ary Interleaved Sequences, *IEEE Trans. on Information Theory*, Vol. 41, No. 2, pp. 400-411, 1995.
8. GSM, *Global Systems for Mobile Communications*, available at <http://cryptome.org/gsm-a512.htm>
9. E.L. Key, An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators, *IEEE Trans. Informat. Theory*, Vol. 22, No. 6, pp. 732-736, 1976.
10. M. Serra *et al.* The Analysis of One-dimensional Linear Cellular Automata and Their Aliasing Properties, *IEEE Trans. on Computer-Aided Design*, Vol. 9, No. 7, pp. 767-778, 1990.
11. X. Sun *et al.* The Concatenation and Partitioning of Linear Finite State Machines, *Int. J. Electronics*. Vol. 78, pp. 809-839, 1995