On the Sequences generated by 90-150 Programmable Cellular Automata

Dolores de la Guía Martínez
Dept. de Comunicaciones
Centro Técnico de Informática (CSIC)
C/ Pinar, 19, 28006 Madrid (Spain)

and

Alberto Peinado Domínguez
Dept. Ingeniería de Comunicaciones
E.T.S. Ingeniería de Telecomunicación, Universidad de Málaga
Campus de Teatinos, 29071 Málaga (Spain)

ABSTRACT

The analysis of the sequences produced by a pseudo-random number generator based on the iterated quadratic function defined over $GF(2^n)$ is considered. Interesting results, such as the upper bound for their lengths, are stated, and different configurations of the same generator are presented as non-suitable for cryptographic uses. The sequences generated by 3-neighbourhoud Cellular Automata with combinations of rules 90 and 150 are also analyzed, presenting a similar behavior.

Keywords: Cryptography, Pseudorandom number generation, Cellular Automata.

1. INTRODUCTION

As it is well known [5,6], the Cellular Automata are discrete dynamic systems characterized by a simple structure but a complex behavior. This configuration makes them very attractive to be used in the generation of pseudorandom sequences. In this sense, the cellular automata are studied in order to obtain a characterization of the rules producing maximal length sequences, with a good 0-1 distribution. These sequences can be obtained for a given subset of combinations of rules (for example, combinations of rules 90 and 150).

From a cryptographic point of view, it is very important to study some additional characteristics, such as the linear complexity, to determined the unpredictability of the sequences produced. The results of this study point at to the equivalence between the sequences generated by Cellular Automata and those generated by Linear Feedback Shift Registers (LFSR) [2]. Hence, an additional step is needed to increment the complexity of the sequences. This step corresponds to the concept of Programmable Cellular Automata (PCA).

PCA can be defined as CA whose rules are modified dynamically. Therefore, PCA include the necessary mechanisms to generate different CA configurations. The most simple way to design PCA is to maintain tables with several CA configurations. As one can suppose, not every CA configuration is valid to be included in these tables.

Different schemes have been proposed [3,7] to design PCA, but all of them share a prerequisite: the CA must be a Group CA, that is, the CA configuration must be chosen in such a way that every CA state is reachable. For example, the scheme in [7] proposes the use of a ROM to save the valid configurations. On the other hand, in [3] the term VCCA is introduced consisting of an initial Group CA whose configuration rule generates all the valid configurations by successive operation of the initial matrix.

In this paper, the CA configurations constructed from combinations of rules 90 and 150 are studied, but not only those producing Group CA. From the results in [8], certain class of Nongroup CA can be used to design a pseudorandom sequence generator. The sequences generated present high length and good 0-1 distribution. Here, we study the approximation to this case by means of CA with combinations of rules 90 and 150, instead of those proposed in [8].

The study of 90-150 PCA is performed in terms of their algebraic properties, using the notation introduced in [1]. Hence, the characteristic polynomial of the CA characteristic matrix is employed to classify the combinations of the rules. This fact allows us to compare and approximate to the CA in [8] characterized by a characteristic polynomial of a very special form. Different 90-150 ruled PCA schemes have been considered and the results obtained include sequence

lengths, autocorrelation tests, 0-1 distribution tests, and linear complexity tests.

In Section 2, the pseudorandom number generator derived from [8] is described. Next, the cryptographic properties of the sequences generated are analyzed in section 3. Finally, section 4 presents the study of such a generator implemented by means of 90-150 Cellular Automata.

2. THE GENERATOR

Let us consider the mapping f defined as $f:GF(2^n) \rightarrow GF(2^n)$; $f(x) = x^2 + bx + c$, $b,c \in GF(2^n)$ with $b \neq 0,1$. Then, as it is proved in [8], the cycles length upper bound of the orbits of this function f is 2^{n-1}- 2. The cycle length reaches this bound for certain values of b.

Taking into account that the sequences of quadratic orbits in $GF(2^n)$ can be generated by means of a CA, we can apply the usual techniques to improve the randomness of the sequences produced by a CA. In [3] and [7], different versions of programmable CA are proposed, but they are only useful for Group CA. Hence, they are not applicable to our functions.

The method we consider in this paper is called ARCOS, an early version of which was presented in [4] as a generic algorithm with many variable parameters, and thus, with many possible configurations. The algorithm is summarized in the following steps:

Step 1. Choose a pair (b,c_0) such that $f_0(x) = x^2 + bx + c_0$ presents cycles of maximal length

Step 2. Choose a seed $x_0 \in GF(2^n)$ yielding a maximal cycle. (Note that there only exist four values not yielding these cycles). Let i =0.

Step 3. Iterate v times $f_i(x_i)$, *i.e.*, compute $f_i(x_i)$, $f_i^2(x_i)$, ..., $f_i^v(x_i)$.

Step 4. Update the value of c_i to c_{i+1}.

Step 5. Let $x_{i+1} = f_i^v(x_i)$ be the seed for iterations of that $f_{i+1}(x) = x^2 + bx + c_{i+1}$.

Step 6. Let $i = i + 1$. Go to step 3.

The final sequence is composed by the least significant bit of every element $x \in GF(2^n)$ produced in the algorithm.

Thus, the sequences produced by this algorithm are composed by little pieces of cycles of maximal length (that is the reason of the algorithm name: ARCOS (*arcs*)). Every time the value of c is updated, we are determining a new cycle, always belonging to an orbit with maximal cycles.

The different configurations are determined by the updating method of the coefficient c and the range of its values, the number of iterations v for every value of c, and the way we detect the existence of minimal cycles. It is important to note that when a maximal cycle exist, a very short cycle of length 1 or 2 also exists.

3. ANALYSIS OF THE SEQUENCES

In this section we analyze the cryptographic properties of the sequences generated, in order to determined the set of suitable configurations of the algorithm ARCOS.

Since the sequence is composed by little sequences (arcs) with low linear complexity, the length of these arcs must be about n, in order to avoid great jumps in the linear complexity graph. Note that the linear complexity graph of a truly random sequence is very close to 1/2 slope line. Hence, the parameter v is a fixed value around n. Every configuration considered in this paper makes use of this assumption.

The updating method applied to the coefficient c is simply a n-bit counter. Some differences are considered regarding the range of valid values, determining several configurations.

Prior to define any configuration, it is important to state some generic results.

Proposition 1. Let C_1 and C_2 be the sets defined as $C_1 = \{c \in GF(2^n); Tr(c/b^2) = 0\}$, and $C_2 = \{c \in GF(2^n); Tr(c/b^2) = 1\}$, with b producing maximal cycles, and Tr being the trace function. Then we have

- $f_i[GF(2^n)] = f_j[GF(2^n)]$ if and only if either $c_i, c_j, \in C_1$, or $c_i, c_j \in C_2$.
- $f_i[GF(2^n)] \cap f_j[GF(2^n)] = \varnothing$ if and only if either $c_i \in C_1$, and $c_j, \in C_2$, or $c_i \in C_2$ and $c_j \in C_1$.

Proof. Let α be an element in $f_i[GF(2^n)]$. Then, the following equation is satisfied

$$x^2 + bx + c_i = \alpha,$$

for some x in $GF(2^n)$. In other words, $Tr((\alpha+c_i)/b^2) = 0$, that is to say,

$$Tr(\alpha/b^2) = Tr(c_i/b^2)$$

thus concluding. ð

Remark 2. It can be checked that if $x \in f_i[GF(2^n)]$, then $\sigma(x) \in f_j[GF(2^n)]$, where σ is a permutation over $f_i[GF(2^n)]$, if and only if if either $c_i, c_j, \in C_1$, or $c_i, c_j \in C_2$.

Proposition 3. Let $arc(b,c_i)$ be the arc generated by the coefficients b,c_i. Then, there only exists one element in $arc(b,c_{i-1})$ determining $arc(b,c_i)$.

Proof. It is a direct consequence of previous proposition. ð

Theorem 4 With the above assumptions, the cycle length of the sequences produced by the algorithm ARCOS is

$$l(x) \leq v2^{2n-2}$$

Proof. From the results in propositions 1 and 3, one can consider only 2^{n-1} different arcs for every pair (b,c_i). On the other hand, as it is presented in [4], although the maximal cycle length of an f-orbit is 2^{n-1}-2, it is recommended to use only the first 2^{n-2}-1 elements of the maximal sequence because $f^k(x) = f^{k/2}(x) + b + 1$, for $k = 2^{n-2}$-1. Taking into account that we are considering only binary sequences, for every $x \in f[GF(2^n)]$, we have

- $f^k(x) \equiv f^{k/2}(x) \bmod 2$, if $b \equiv 1 \bmod 2$
- $f^k(x) \equiv f^{k/2}(x) + 1 \bmod 2$, if $b \equiv 0 \bmod 2$

This fact restricts the number of different arcs for the same pair (b,c_i) to 2^{n-2}.

Let $x_1, x_2, ..., x_v$ be the elements in $arc(b,c_0)$, and let $y_1, y_2, ..., y_v$ be the elements in $arc(b,c_1)$. Suppose that $arc(b,c_j)$ with $c_j = c_0$, is composed by the elements

$$f_{c0}{}^{k/2}(x_1), f_{c0}{}^{k/2}(x_2),..., f_{c0}{}^{k/2}(x_v),$$

or equivalently, x_1+b+1, x_2+b+1, ..., x_v+b+1. Hence, applying the algorithm, we have

$$arc(b,c_{j+1}) = \{ f_{c1}(x_v+b+1), f_{c1}{}^2(x_v+b+1), ..., f_{c1}{}^v(x_v+b+1)\}.$$

Since $f_{c1}(x_v+b+1) = f_{c1}(x_v)+b+1 = y_1+b+1$, then

$$arc(b,c_{j+1}) = \{ y_1+b+1, y_2+b+1, ..., y_v+b+1\}.$$

Thus, there only exists 2^{n-2} different elements to be considered as part of $arc(b,c_j)$. Since 2^n values of c are considered, the maximal length is restricted to $v2^n2^{n-2} = v2^{2n-2}$. ð

Next, we consider four basic configurations based on two parameters, the valid range for the coefficient c, and the usage of mechanisms to control short cycles ocurrence.

Definition 5. The configuration satisfying the following items is called *Configuration I*:

- Only one fixed value of b is used.
- The value of b determines maximal cycles.
- The valid range of c is $GF(2^n)$.
- $c_{i+1} = c_i + 1 \bmod 2^n$, with $c_0 = 0$.
- v is configurable.
- No control is applied on the occurrence of short cycles.

Definition. The configuration satisfying the following items is called *Configuration II*:

- Only one fixed value of b is used.
- The value of b determines maximal cycles.
- The valid range of c is $GF(2^n)$.
- $c_{i+1} = c_i + 1 \bmod 2^n$, with $c_0 = 0$.
- v is configurable.
- Short cycles are detected for every pair (c_i,x_i). If a short cycle is determined, then the corresponding arc is not used, and the pair (c_{i+1}, x_i) is now considered.

Definition. The configuration satisfying the following items is called *Configuration III*:

- Only one fixed value of b is used.
- The value of b determines maximal cycles.
- The valid range of c is $GF(2^n)-\{\alpha^{n-1} + \alpha^{n-2} + ...+\alpha+1\}$.
- $c_{i+1} = c_i + 1 \bmod 2^n$, with $c_0 = 0$.
- v is configurable.
- No control is applied on the occurrence of short cycles

Definition. The configuration satisfying the following items is called *Configuration IV*:

- Only one fixed value of b is used.
- The value of b determines maximal cycles.
- The valid range of c is $GF(2^n)-\{\alpha^{n-1} + \alpha^{n-2} + ...+\alpha+1\}$.
- $c_{i+1} = c_i + 1 \bmod 2^n$, with $c_0 = 0$.
- v is configurable.
- Short cycles are detected for every pair (c_i,x_i). If a short cycle is determined, then the corresponding arc is not used, and the pair (c_{i+1}, x_i) is now considered.

In general, the cryptographic properties of the sequences produced by ARCOS depend on many parameters. This fact makes the study hard. In any case, we can state the following results.

The four configurations produce sequences of period far from the theoretical upper bound, although it is important to note that configurations III and IV produce longer sequences than I and II.

The linear complexity (LC) of the sequences produced by means of this configurations I and III is extremely low. However, the LC is very good in configurations II and IV. More precisely, the LC graph evolves around the 1/2 slope line, and the final value corresponds to the effective length of the sequences.
The effective length is determined by the first significant

peak in the autocorrelation graph. This length is, in most cases, the half of the sequence period.

From these results, one can conclude that configuration IV is the best of those considered in this paper. However, the sequences produced by the this configuration are strongly dependent on many others parameters, such as the seed and the arc length *v*. Hence, further study is necessary to establish the optimum parameters.

4. 90-150 CELLULAR AUTOMATA

In the field of the CA, the 3-neighbour rules are the most used because of theirs advantages in software and hardware implantations. Mainly, the hybrid group CA with rules 90 and 150 are the most extended [3], [8].

That is the reason we have applied the previous algorithm to hybrid nongroup CA with rules 90 and 150 and characteristic polynomial of the form $x(x+1)Q(x)$, where $Q(x)$ is primitive. The use of these CA configurations simplifies the hardware implementations and avoid several precomputations to obtain the matrix associated to quadratic functions. Thus, we are studied the next cases for different CA lengths, *n*,:

- For $n = 4$, with CA polinomial $x(x+1)(x^2+x+1)$ and configuration 90-90-150-150
- For $n = 6$, with CA polinomial $x(x+1)(x^4+x+1)$ and configuration 90-150-150-90-90-150
- For $n=7$, with CA polynomial $x(x+1)(x^5+x^4+x^3+x+1)$ and configuration 90-150-90-150-150-90-150
- For $n = 8$, with CA polinomial $x(x+1)(x^6+x+1)$ and configuration 90-90-90-90-150-150-150-90

As in the previous section, the results for the case $n = 4$ are not cryptographically interesting. For $n = 5$; there is no hybrid CA with rules 90 and 150 of characteristic polynomial of the mentioned form. For $n > 5$, the results are very similar to those obtained previously as we can see in the next table.

v	***n* = 6**		***n* = 7**		***n* = 8**	
	Per.	**LC**	**Per.**	**LC**	**Per.**	**LC**
4	1184	596	15250	7621	21084	10542
5	605	302	19055	9527	26355	13177
6	1778	889	22870	11435	10550	5275
7	2058	1029	26698	13349	5243	2621
8					42208	21104
9					15831	7915
10					52720	26360

Therefore, we can conclude that it seems to be possible to simplify the generation of sequences by means of quadratic functions. Specifically, we can generate this kind of sequences using Hybrid Nongroup CA with rules 90 and 150 instead of the CA equivalent to the quadratic functions.

5. CONCLUSIONS

In this paper, the Nongroup CA configurations constructed from combinations of rules 90 and 150 are studied. The sequences generated based on the orbits of quadratic functions in $GF(2^n)$ presents cycles of high length and good 0-1 distribution. A pseudorandom sequence generator based on these functions is developed by means of a Nongroup Hybrid Additive Cellular Automata, after checking that the original CA derived from the quadratic function can be replaced by a simpler Nongroup CA defined by the usual rules 90 and 150.

ACKNOWLEDGEMENTS

This work is supported by CICYT (Spain) under grant TEL98-1020, "*Infraestructuras de Seguridad en Internet e Intranets. Aplicación a Redes Públicas y Corporativas*".

REFERENCES

[1] A.K, Das, A. Ganguly, A. Dasgupta, S. Bhawmik, P.P. Chaudhuri, "Efficient characterisation of cellular automata", *IEE Proc.*, **137** *Part E* (1), 1990, pp. 81-87.

[2] Golomb, S., "Shift register sequences", Aegean Press, revised edition, (1982)

[3] D. de la Guía, A. Fúster, "Cryptographic design based on cellular automata", *IEEE International Symposium on Information Theory*, 1997, pp.180.

[4] D. de la Guía, A. Peinado, "Pseudorandom number generation based on Nongroup Cellular Automata", *Proceedings of 33rd Annual 1999 IEEE International Carnahan Conference on Security Technology, Madrid, October 5-7* , 1999, pp 370-376.

[5] E. Jen, "Global Properties of Cellular Automata", *Journal of Statistical Physics*, **43**, 1986, pp. 219-242.

[6] O. Martin, A.M. Odlyzko, S. Wolfram, "Algebraic properties of Cellular Automata", *Commun. Math. Phys*., **93**, 1984, pp. 219-258.

[7] S. Nandi, B.K. Kar, P.P. Chaudhuri, "Theory and applications of cellular automata in Cryptography", *IEEE Trans, Comput*, **43**, 1994, pp 1346-1357.

[8] J.Muñoz, F.Montoya, A.Peinado, "Iterated Quadratic functions in E_2^n", *International Journal of Applied Mathematics*, **5**, (1) 2001, pp 65-83.