An Improved Attack on the Shrinking Generator

P. CABALLERO-GIL[1] AND A. FÚSTER-SABATER[2]
[1]Department of Statistics, Operations Research and Computation
University of La Laguna
38271 Tenerife
SPAIN

[2] Institute of Applied Physics
Spanish Higher Council for Scientific Research
Serrano144, 28006 Madrid
SPAIN

Abstract: - This work proposes a known-plaintext attack on the Shrinking Generator through its characterization by means of Cellular Automata. It is based on the computation of the characteristic polynomials of sub-automata and on the generation of the Galois field associated to one of the Linear Feedback Shift Registers components of the generator. The proposed algorithm allows predicting a large number of unseen bits of the keystream sequence, thanks to the knowledge of both registers lengths, the characteristic polynomial of one of the registers, and some keystream bits.

Key-Words: - Cellular automata, cryptanalysis, stream cipher, shrinking generator

1 Introduction

Stream ciphers may be generally defined as simple bitwise additions between the plaintext stream and the running keystream, [1]. Nonlinear combinations of Linear Feedback Shift Registers (LFSRs) are the most frequently used running key generators because if they are properly designed, keystream sequences easily exhibit some ideal characteristics such as long period and balanced statistics. However, in general it is much more difficult to guarantee their unpredictability. From a cryptanalysis point of view, stream ciphers must be resistant against known-plaintext attacks. In these attacks, it is assumed that cryptanalysts may intercept some bits of the keystream, and their goal is to get some information about the seed of the keystream generator or about the unseen keystream bits, faster than exhaustive search of all possible keys.

The Shrinking Generator (SG) is a nonlinear combinator based on two LFSRs so that the bits of one output are used to determine whether the corresponding bits of the second output are used as part of the overall keystream, [2]. SGs are simple and scalable generators that produce pseudorandom sequences with good security properties. There have been several approaches for attacking the SG. A basic divide-and-conquer attack requiring an exhaustive search through all the possible initial states and feedback polynomials of the selector LFSR was proposed in [3]. The authors of [4] described a correlation attack targeting the second LFSR. A correlation attack based on searching specific subsequences of the output sequence was introduced in [5]. More recently, a distinguishing attack applicable when the second LFSR has a low-weight feedback polynomial was investigated in [6]. However, despite all these attacks, the SG continues being considered resistant against efficient cryptanalysis.

Cellular Automata (CA) are discrete mathematical models in which a lattice of finite state machines, called cells, updates itself synchronously according to local rules, [7]. Because of their simplicity, regularity, modularity and cascadable structure with local neighbourhood, CA are ideally suited for VLSI implementation. CA have been proposed both for secret and public key cryptography [8], [9], [10]. Also cryptanalysis of certain CA based keystream generators have been published [11], [12]. Two different works have explored the use of CA as models for predicting pseudorandom binary sequences. In [13], a Cellular Automata-Based model for the Shrinking Generator was proposed. Such a work may be considered the starting point of this research.

The structure of this work is as follows. Next section gives relevant background about the basic structures we are dealing with: Linear Hybrid Cellular Automata and Shrinking Generators. The model for the Shrinking Generator that is used in this work is described next. Sections 3 and 4 introduce the

Research supported by the Spanish Ministry of Education and Science and the European FEDER Fund under Projects SEG2004-04352-C04-03 and SEG2004-02418

theoretical basis of the proposed CA-based cryptanalysis of the SG and the full description of the algorithm, respectively. Finally, in Section 5 several conclusions and open questions are drawn.

2 Preliminaries

Cellular automata are finite state machines that consist of arrays of n cells. According to local interaction rules, the cells are updated synchronously in discrete time steps. The state of a particular cell at the next time step is determined by the current states of a surrounding neighbourhood of cells. The transitions are usually specified in the form of a rule table that defines the cell's next state for each possible neighbourhood configuration.

The simplest nontrivial CA are binary and one-dimensional, with two possible states per cell and a cell's neighbours defined as the cell on either side of it. These automata were called elementary cellular automata] by Wolfram, who studied extensively their properties [14]. A cell and its two neighbours make out a neighbourhood of 3 cells, so there are 8 possible patterns, and 256 possible rules. These 256 CAs are generally referred to using a standard naming convention invented by Wolfram. The name of a CA is a decimal number which, in binary, gives the rule table. For example, according to rule 90, the value of a particular cell i is the sum modulo 2 of the values of its two neighbours cells on the previous time step t. Rule 150 also includes the value of cell i at time step t.

Null CA are those where cells with permanent null content are supposed adjacent to the extreme cells of the CA. Binary CA where the neighbourhood dependence is just on XOR operations are called linear CA. In [15] it was shown that a three-neighborhood linear CA can be represented by a tridiagonal characteristic matrix - a matrix which has the elements of its diagonal and two off-diagonals as non-zero. If in a CA different rules are applied over different cells, then it is called a hybrid CA. Linear Hybrid Cellular Automata are usually denoted with the acronyms LHCA.

The matrix algebraic tool employing minimal and characteristic polynomials of the characteristic matrix showed various interesting features of CA behaviour. The first important finding was the categorization of linear CA into group and non-group CA. In a group CA each of the states has a single predecessor which is not true for non-group CA. The most effective application of null group CA has been proposed in the field of pseudorandom pattern generation, since the authors of [18] showed that maximum length CA - group CA with all non-zero states lying in a single cycle - produce high quality pseudorandom patterns.

It has been established that the maximum length cycle can be produced only if the characteristic polynomial is primitive as well as only if rule 90 and/or rule 150 is used to construct the CA, [16]. On the other hand, the authors of [17] examined the concatenating maximum length CA to obtain longer or smaller ones maintaining that property. In this research only one-dimensional 90/150 null LHCA are considered. Binary string R_1 R_2 ... R_n are here used to represent n-cell LHCA, where R_i is either 0, if cell i uses rule 90, or 1, if cell i uses rule 150.

Given an irreducible polynomial, several algorithms have been developed to find its corresponding LHCA. The most recent one, proposed in [16], applies the Euclidean algorithm to compute the LHCA in a polynomial running time, so it is sufficiently fast to generate LHCA for polynomials of very large degree.

On the other hand, in [18], a synthesis algorithm based also on the Euclidean algorithm which allows computing in linear time the characteristic polynomial for any given LHCA was introduced. In this work such an algorithm will be called Polynomial-Synthesis Algorithm. When the characteristic polynomial of a CA is primitive, the sequence produced by any cell of the CA can be obtained from phase shift of any other sequence produced from another cell of the same CA, [19]. The converse may not be true.

The shrinking generator was introduced by Coppersmith, Krawczyk, and Mansour [2]. The SG is a well-known keystream generator composed of two LFSRs: a selector register that produces a sequence used to decimate the sequence generated by the other register. The selector register is here denoted by S, its length is L_S, its characteristic polynomial is $P_S(x)$ and the sequence it produces is $\{s_i\}$. The decimated sequence is denoted $\{a_i\}$, the second register that produces it is A, its length is L_A and its characteristic polynomial is $P_A(x)$.So, the shrunken sequence $\{z_j\}$ may be defined according to the following rule:

if s_i=1 then z_j=a_i
if s_i=0 then a_i is discarded

Despite its simplicity, the SG has remained remarkably resistant to cryptanalysis because there are no known attacks that are feasible if both LFSRs are too long for exhaustive search.

3 The CA-Model for the Shrinking Generator

In this work we consider the linear model of the SG described in [13] in terms of LHCA. The equivalent LHCA obtained for any SG through the algorithm described there and here denoted CA-Synthesis Algorithm, are formed by concatenations of basic primitive LHCA and their mirror images, with one or two modifications (complementations of rules) in each LHCA component. In particular, we have found that the numbers of modifications in the described model are two in all but two concatenated LHCA, and only one modification in the two extreme LHCA.

The outputs of the CA-Synthesis Algorithm are two equivalent LHCA for any SG with selector LFSR of length L_S and decimated LFSR sequence produced by A. The characteristic polynomial of the equivalent LHCA is the same as the one of the original SG, that is to say, $P(x)^N$. Since the equivalent LHCA are only related to the LFSR A and the length L_S of the LFSR S, and the modifications consist exclusively in reconfiguring a rule 90 cell to a rule 150 cell or vice versa, the described CA-model of the SG provides a great economy in hardware by allowing the use of the same basic machine for many different SGs.

Since the number of concatenations is between 2^{L_S-2} and 2^{L_S-1}, and the length of the basic primitive LHCA is L_A, we have that the length of the equivalent LHCA is given by an integer L such that 2^{L_S-2} and L_A $2^{L_S-2} < L \leq L_A\, 2^{L_S-2}$. Consequently, in order to generate the whole shrunken sequence in one of the extreme cells of the equivalent LHCA it would be necessary to determinate uniquely the initial state of the equivalent LHCA which is able to produce it, and to get this, it would be necessary to intercept L shrunken bits. So, although we have a linear model of the SG, in order to break the SG with it, we need as many intercepted bits as the linear complexity of the SG.

This work provides an efficient way to use the CA-model of the SG in order to guess unseen bits of the shrunken sequence correctly from the interception of a number of bits lower than the linear complexity of the SG.

4 Theoretical Basis

Let $Z = Z_0 = z_0, z_1, z_2, \ldots$ be the output sequence of the SG whose characteristic polynomial $P(x)^N \in GF(2)[x]$ has degree L. Moreover, $Z_t = z_t, z_{t+1}, z_{t+2}, \ldots$ denotes the t−th phase shift of Z. Finally, let $\alpha \in GF(2^{LA})$ be a root of P(x).

4.1 Chained Sub-triangles

The equivalent LHCA may generate the shrunken sequence in any of its cells. Consequently, given a shrunken sequence $z_0, z_1, z_2, \ldots z_r$ it is always possible to assume, without loss of generality, that its generation is at the left extreme cell. According to this, assuming the knowledge of r bits of the shrunken sequence, it is always possible to reconstruct r sub-sequences of length $r - i + 1$ corresponding to the rules R_i with $1 < i \leq r$. Since rules 90 and 150 are additive and the equivalent LHCA is null boundary, for any rule Ri, the previous reconstruction is made thanks to sums of some elements of the shrunken sequence, whose sub-indexes correspond to the exponents of the unknown in the characteristic polynomial of the LHCA $R_1\, R_2 \ldots R_{i-1}$.

Consequently, if this sub-sequence of length r − i+1 is used recursively as left extreme sequence of the equivalent LHCA in order to reconstruct in the same way as before, r−i+1 sub-sequences of length r−2i+2 corresponding to the rules Ri with $1 < i \leq r - i + 1$, we obtain in the same cell i: $z_{t+2k1} + z_{t+2k2} + \cdots + z_{t+2kri}$.
In this way, if the hypothesis

$$Z^d = Z^{2k1} + Z^{2k2} + \cdots + Z^{2kri} \qquad (1)$$

is fulfilled, then a d−th phase shift of the shrunken sequence reappears at cell i of the equivalent LHCA each second chained sub-triangle generated as explained in the previous paragraph. Furthermore, it is easy to see that if hypothesis (1) is satisfied, then each 2j-th chained triangle provides r−2ji+2j bits of a jd−th phase shift of the shrunken sequence. On the other hand, note that hypothesis (1) may be easily generalized to guarantee the reappearance of a d-th phase shift of the shrunken sequence at cell i in each 2^l-th chained sub-triangle.

4.2 Finite Field

It is well-known that if $\{s_n\}$ is a sequence produced by a LFSR whose characteristic polynomial is irreducible, and alpha is a root of such a polynomial, then each element s_n of the sequence may be written as the trace of the n-power of alpha [20].

Since P(x) is a L_A-degree primitive polynomial, the successive powers α^i, $0 \leq i < 2^{LA} - 1$ generate the finite field $GF(2^{LA})$ and their respective traces equal the corresponding elements s_i of the PN-sequence associated to the polynomial P(x).

On the other hand, since the trace function is linear and all the powers of alpha may be expressed in terms of the first L_A -1 powers, the association between powers of alpha and elements of the PN-sequence may be transferred to linear relations between different phase shifts of the PN-sequence and the first L_A -1 phase shifts.

From [13] we know that the shrunken sequence is composed of interpolations of different phase shifts of the PN-sequence associated to the polynomial P(x), so that the element s_i of the basic PN-sequence corresponds to the shrunken bit Z_{iN}. Consequently, any linear relation between different phase shifts of the PN-sequence deduced as explained in the previous paragraph corresponds to a linear relation between different phase shifts of the shrunken sequence, which are the same phase shifts obtained for the PN-sequence, but multiplied by N.

5 Algorithm

Starting from the theoretical basis of the previous section, in the following we describe an efficient algorithm based on the generation of the finite field GF(2^{LA}), which allows testing the generalization of hypothesis (1) obtained in linear time with the Polynomial-Synthesis Algorithm. The proposed cryptanalysis algorithm requires as input an intercepted shrunken sequence produced by a SG whose structure must be known, and provides as output a variable number of unseen shrunken bits.

Algorithm:

Input: The lengths L_S and L_A, and the characteristic polynomial of A, $P_A(x)$ corresponding to the LFSRs S and A components of the SG.

Off-line Phase:

Step 1: Using the CA-Synthesis Algorithm, compute the two equivalent LHCA that are valid for any SG with selector LFSR of length L_S and decimated LFSR sequence produced by A.

Step 2: Using the primitive L_A,-degree polynomial P(x) associated to the basic LHCA, generate the finite field GF(2^{LA}) formed with the exponentiation of one root of such a polynomial, alpha, and express each element of GF(2^{LA}), α^e as a L_A-length array E=[e_0,e_1, ... ,e_{LA-2},,e_{LA-1}] where e_i=i iff α^i is present in the expression of α^e.

Step 3: Using the Polynomial-Synthesis Algorithm, calculate the 2*L characteristic polynomials for all possible sub-LHCA considering from left rule to each rule of both LHCA obtained in the previous step, and express them as 2*L different (L+1)-length arrays D=[d_0,d_1, ... ,d_{L-1},,d_L] where d_i=i iff i is an exponent of the unknown in the corresponding polynomial. Using the finite field generated in the previous step, decompose the array D as an equivalent linear expression a*B+C with a being a power of 2, and B and C two arrays such that B=[0, b_1, ... ,b_{LA-2},,b_{LA-1}] and C=[c,c,... ,c]. Consider B as the output of this step.

Step 4: For each array B=[0, b_1, ... ,b_{LA-2},,b_{LA-1}] obtained in Step 3, search it within all the L_A-length arrays obtained in Step 2, and if found it, associate to the corresponding rule the (N/a)*(a*e+c)-th phase shift for the N/a-th sub-triangles.

On-line Phase:

Step 5: Once intercepted r shrunken bits, proceed with them by generating the chained sub-triangles indicated in Step 4, to obtain for all successful rules u, bits of different phase shifts of the shrinking sequence.

Output: A variable number of bits of different phase shifts of the shrunken sequence.

Note that all the computations made for any LHCA are useful for any other LHCA with the same basic CA and more concatenations, that is to say, the outputs of steps 2, 3 and 4 obtained for an equivalent LHCA with characteristic polynomial $(P(x))^{N1}$ continue being correct for any other equivalent LHCA with characteristic polynomial $(P(x))^{N2}$, with $N_1|N_2$. Consequently, the cryptanalysis of a SG with LFSRs S_1 and A_1 are useful for the cryptanalysis of any other SG with LFSRs S_2 and A_2 such that its corresponding characteristic polynomial is $(P_A(x))$ N_2 with $N_1|N_2$.

6 Conclusions and Open Problems

This paper has introduced a known-plaintext attack on the shrinking generator that does not require too many intercepted bits in order to predict with absolute certainty approximately the same number of unseen shrunken bits. Any shrinking generator leading to a successful off-line phase of the algorithm, which produces the deduction of many unseen shrunken bits, should be rejected for its cryptographic use. Therefore, the proposed algorithm is useful both for cryptanalysts and for cryptographers who use the shrinking generator. With respect to the efficiency of the proposal, since both synthesis algorithms used within the cryptanalysis are linear, there may be deduced that communication complexity and computational costs of the attack are affordable. One of the subjects that are being object of work in progress is the modelling of other keystream generators through concatenations of maximum length CA.

References:

[1]R.A.Rueppel, Stream Ciphers. Contemporary Cryptology, The Science of Information. IEEE Press, 1992, 65-134.

[2]D.Coppersmith, H.Krawczyk, Y. Mansour, The Shrinking Generator. LNCS773, Springer Verlag, 1994, 22-39

[3]L.Simpson, J.D.Golic, EDawson, A Probabilistic Correlation Attack on the Shrinking Generator. LNCS1438, Springer Verlag, 1998, 147-158.

[4]J.D.Golic, L.O'Connor, A Cryptanalysis of Clock-Controlled Shift Registers with Multiple Steps. Cryptography: Policy and Algorithms, 1995, 174-185.

[5]T.Johansson, Reduced Complexity Correlation Attacks on Two Clock-Controlled Generators. LNCS1514, Springer Verlag, 1998, 342-356.

[6]P.Ekdahl, W.Meier, T.Johansson, Predicting the Shrinking Generator with Fixed Connections. LNCS2656, Springer Verlag, 2004, 345-359.

[7]S.Wolfram, Statistical Mechanics of Cellular Automata, Reviews of Modern Physics 55, 1983, 601-644.

[8]S.Wolfram, Cryptography with Cellular Automata. LNCS218, Springer Verlag, 1986, 429-432.

[9]S.Nandi, B.K.Kar, P.P.Chaudhuri, Theory and Applications of Cellular Automata in

Cryptography. IEEE Transactions on Computers 43,12, 1994, 1346-1357.

[10]F.Seredynski, P.Bouvry, AY.Zomaya, Cellular Automata Computations and Secret Key Cryptography. Parallel Computing archive Volume 30, Issue 5-6, 2004, .

[11]W.Meier, O.Staffelbach, Analysis of Pseudo Random Sequence Generated by Cellular Automata. LNCS547, Springer Verlag, 1992, 186-199.

[12]M.Mihaljevic, Security Examination of a Cellular Automata Based Pseudorandom Bit Generator Using an Algebraic Replica Approach. LNCS1255, Springer Verlag, 1997, 250-262.

[13]A.Fúster-Sabater, D.de la Guía, Cellular Automata Application to the Linearization of Stream Cipher Generators. LNCS3305, 2004, 612-622.

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

[15]A.K.Das, A.Ganguly, A.Dasgupta, S.Bhawmik, and P.P. Chaudhuri. Efficient characterisation of cellular automata. IEE Proc. in Computers and Digital Techniques, 137, 1, : 81-87, January 1990.

[16]K.Cattell and J.C.Muzio. Synthesis of one-dimensional linear hybrid cellular automata. IEEE Transactions on Computer-Aided Design, 1996.

[17]X.Sun, E.Kontopidi, M.Serra, and J. C.Muzio. The concatenation and partitioning of linear finite state machines. International Journal of Electronics, 78, 5, 809-839, 1995.

[18]K.Cattell, S.Zhang, X.Sun, M.Serra, J.C.Muzio, and D. M. Miller, One-Dimensional Linear Hybrid Cellular Automata: Their Synthesis, properties, and Applications in VLSI Testing.

[19]S.Cho, U.Choi, Y.Hwang, H.Kim, Y.Pyo, K. Kim, S.Heo, Computing Phase Shifts of Maximum-Length 90/150 Cellular Automata Sequences. LNCS3305, 2004, 31-39.

[20]E.L.Key, An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators, IEEE Transactions on Information Theory, Vol. IT-22, 1976, 732-736.