New Attack Strategy for the Shrinking Generator

P. Caballero-Gil[1], A. Fúster-Sabater[2] and M.E. Pazo-Robles[3]

[1] Faculty of Maths, D.E.I.O.C., University of La Laguna, 38271 Tenerife, Spain
pcaballe@ull.es
[2] Institute of Applied Physics, C.S.I.C., Serrano 144, 28006 Madrid, Spain
amparo@iec.csic.es
[3] ITBA Instituto Tecnológico de Buenos Aires, Av. E. Madero 399, (C1106ACD) Buenos Aires, Argentina
eugepazorobles@gmail.com

Abstract. This work shows that the cryptanalysis of the shrinking generator requires fewer intercepted bits than that indicated by the linear complexity. Indeed, whereas the linear complexity of shrunken sequences is between $A{\cdot}2^{(S-2)}$ and $A{\cdot}2^{(S-1)}$, we claim that the initial states of both component registers are easily computed with less than $A \cdot S$ shrunken bits. Such a result is proven thanks to the definition of shrunken sequences as interleaved sequences. Consequently, it is conjectured that this statement can be extended to all interleaved sequences. Furthermore, this paper confirms that certain bits of the interleaved sequences have a greater strategic importance than others, which may be considered as a proof of weakness of interleaved generators.

Keywords: Cryptanalysis, Stream Cipher
ACM Classification: E.3 (Data Encryption), B.6.1 (Design Styles)

1 Introduction

Stream ciphers are considered nowadays the fastest encryption procedures. Consequently, they are implemented in many practical applications e.g. the algorithms A5 in GSM communications [10], the encryption system E0 in Bluetooth specifications [2] or the algorithm RC4 [15] used in SSL, WEP and Microsoft Word and Excel. From a short secret key (known only by the two interested parties) and a public algorithm (the sequence generator), a stream cipher procedure is based on the generation of a long sequence of seemingly random bits. Such a sequence is called the keystream sequence. For the encryption, the sender computes the bitwise exclusive OR (XOR) operation among the bits of the original message or plaintext and the keystream sequence. The result is the ciphertext to be sent. For the decryption, the receiver generates the same keystream, computes the same bitwise XOR operation between the received ciphertext and the keystream sequence and obtains again the original message.

Most keystream generators are based on Linear Feedback Shift Registers (LFSRs) [8], which are linear structures characterized by their length (the number of memory cells), their characteristic polynomial (the feedback function) and their initial state (the seed or key of the cryptosystem). If the characteristic polynomial is a primitive polynomial [14], then LFSRs generate Pseudo-Noise sequences (PN-sequences) with good characteristics of pseudorandomness. For a survey on recurring sequences, primitive LFSRs, and PN-sequences, the interested reader is referred to [8]. In stream cipher procedures, the PN-sequences are combined by means of nonlinear functions in order to produce keystream sequences of cryptographic application. Combinational generators, nonlinear filters, clock-controlled generators, irregularly decimated generators ... are just some of the most popular nonlinear sequence generators. All of them produce keystreams with high linear complexity, long period and good statistical properties (see [6] and [3]).

Most cryptanalysis on stream ciphers are performed under a known plaintext hypothesis, that is to say, it is assumed that the attacker has direct access to a portion of the keystream sequence (*intercepted sequence*). From the intercepted bits, the attacker has to deduce the cryptosystem key. Once the key is known, as the sequence generator is public, the whole keystream sequence can be reconstructed. The complexity of this attack is always compared with that of the key exhaustive search. If the former complexity is lesser, then the cryptosystem is said to be broken.

This work focuses on a particular kind of stream ciphers based on $LFSRs$: the class of shrinking generators. They are made out of two LFSRs and an irregular decimation. Shrinking generators have been thoroughly analyzed in several papers such as [17], [13] and [4]. Nevertheless, we present a new and efficient cryptanalytic attack requiring much lesser amount of intercepted bits than that of the previous attacks. The basic idea of this cryptanalysis consists in defining the output sequence of a shrinking generator as an interleaved sequence (see [9] and [12]). The characteristics of the interleaved sequences reveal weaknesses that lead to practical attacks. In addition, we conjecture that these weaknesses can be extended to all interleaved sequence generators with application in cryptography.

The paper is organized as follows: in section 2, the description and characteristics of the shrinking generator are introduced. Interleaved configuration and related results are developed in section 3. A cryptanalytic attack against the shrinking generator that exploits the condition of interleaved sequence is presented in section 4, while the generalization of this technique to other cryptographic interleaved generators appears in section 5. Finally, conclusions in section 6 end the paper.

2 The Shrinking Generator

The Shrinking Generator is a pseudorandom number generator based on a nonlinear combination of the recurring sequences produced by two $LFSRs$. It was first introduced by Coppersmith, Krawczyk and Mansour at Crypto'93, see [5]. This construction uses two sources of pseudorandom bits to create a third source

of potentially better quality than the original ones. Here quality means difficulty of predicting a pseudorandom sequence. We denote by SRA and SRS the first and second LFSR, respectively. The first register SRA has length A, characteristic polynomial $P_A(x) \in GF(2)[x]$ and its output sequence is denoted by $\{a_i\}$ $(i \geq 0)$ with $a_i \in GF(2)$. The selector register SRS has length S, characteristic polynomial $P_S(x) \in GF(2)[x]$ and its output sequence is denoted by $\{s_i\}$ $(i \geq 0)$ with $s_i \in GF(2)$. In addition, the lengths of both registers A, S are relatively prime $(A, S) = 1$, the characteristic polynomials $P_A(x)$, $P_S(x)$ are primitive polynomials in $GF(2)[x]$ and the output sequences $\{a_i\}$, $\{s_i\}$ are PN-sequences of period $(2^A - 1)$ and $(2^S - 1)$, respectively. The output sequence of the shrinking generator, the so-called *shrunken sequence* denoted by $\{z_j\}$ $(j \geq 0)$ with $z_j \in GF(2)$, is a sub-sequence of $\{a_i\}$ whose terms are chosen according to the positions of '1' bits in the sequence $\{s_i\}$. More precisely, the decimation rule is defined such as follows:

1. If $s_i = 1 \Longrightarrow z_j = a_i$
2. If $s_i = 0 \Longrightarrow a_i$ is discarded.

As different pairs of SRA/SRS initial states can generate the same shrunken sequence, in the sequel we assume that the first term of the sequence $\{s_i\}$ equals 1, that is $s_0 = 1$.

According to [5], the period of the shrunken sequence is:

$$T = (2^A - 1) \cdot 2^{(S-1)}, \tag{1}$$

its linear complexity, notated LC, satisfies the following inequality:

$$A \cdot 2^{(S-2)} < LC \leq A \cdot 2^{(S-1)}, \tag{2}$$

and its characteristic polynomial is of the form:

$$P_{ss}(x) = P(x)^p \tag{3}$$

where $P(x)$ is an A-degree primitive polynomial in $GF(2)[x]$ and p is an integer in the interval $2^{(S-2)} < p \leq 2^{(S-1)}$. Moreover, it can be proved [16] that the shrunken sequence has also good distributional statistics. Therefore, this scheme has been traditionally used as keystream sequence generator with application in secret-key cryptography.

3 Interleaved Configuration

The $(2^A - 1) \cdot 2^{(S-1)}$ bits of a period of any shrunken sequence $\{z_j\}$ can be arranged into a $(2^A - 1) \cdot 2^{(S-1)}$ matrix that we will call *interleaved configuration* and will denote by IC. In fact,

$$IC = \begin{pmatrix} z_0 & z_1 & z_2 & \cdots & z_{2^{(S-1)}-1} \\ z_{2^{(S-1)}} & z_{2^{(S-1)}+1} & z_{2^{(S-1)}+2} & \cdots & z_{2 \cdot 2^{(S-1)}-1} \\ z_{2 \cdot 2^{(S-1)}} & z_{2 \cdot 2^{(S-1)}+1} & z_{2 \cdot 2^{(S-1)}+2} & \cdots & z_{3 \cdot 2^{(S-1)}-1} \\ z_{3 \cdot 2^{(S-1)}} & z_{3 \cdot 2^{(S-1)}+1} & z_{3 \cdot 2^{(S-1)}+2} & \cdots & z_{4 \cdot 2^{(S-1)}-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ z_{(2^A-2) \cdot 2^{(S-1)}} & z_{(2^A-2) \cdot 2^{(S-1)}+1} & z_{(2^A-2) \cdot 2^{(S-1)}+2} & \cdots & z_{(2^A-1) \cdot 2^{(S-1)}-1} \end{pmatrix}$$

Now the following result allows one to identify each element of the matrix IC with the corresponding term of the sequence $\{a_i\}$.

Theorem 1. *The interleaved configuration matrix IC can be written in terms of the elements of the sequence* $\{a_i\}$ *such as follows:* $IC =$

$$\begin{pmatrix} a_{o0} & a_{o1} & a_{o2} & \cdots & a_{o(2^{(S-1)}-1)} \\ a_{(2^S-1)+o0} & a_{(2^S-1)+o1} & a_{(2^S-1)+o2} & \cdots & a_{(2^S-1)+o(2^{(S-1)}-1)} \\ a_{2\cdot(2^S-1)+o0} & a_{2\cdot(2^S-1)+o1} & a_{2\cdot(2^S-1)+o2} & \cdots & a_{2\cdot(2^S-1)+o(2^{(S-1)}-1)} \\ a_{3\cdot(2^S-1)+o0} & a_{3\cdot(2^S-1)+o1} & a_{3\cdot(2^S-1)+o2} & \cdots & a_{3\cdot(2^S-1)+o(2^{(S-1)}-1)} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{(2^A-2)\cdot(2^S-1)+o0} & a_{(2^A-2)\cdot(2^S-1)+o1} & a_{(2^A-2)\cdot(2^S-1)+o2} & \cdots & a_{(2^A-2)\cdot(2^S-1)+o(2^{(S-1)}-1)} \end{pmatrix}$$

where the additive sub-indices oj $(j = 0, 1, \ldots, 2^{(S-1)} - 1)$ *depend on the bits of the sequence* $\{s_i\}$ *in the following way: if* $s_i = 1$*, then the corresponding sub-index* oj *equals the sub-index* i*,* $oj = i$*. All the sub-indices are taken module* $2^A - 1$*, which is the period of the sequence* $\{a_i\}$*.*

Proof. Since the period of the *PN*-sequence $\{s_i\}$ is $(2^S - 1)$, the number of bits with value '1' in a period is exactly $2^{(S-1)}$, and all the elements of any column of IC come from the same term $s_i = 1$ of the *PN*-sequence, the above expression for the matrix IC in terms of the elements of $\{a_i\}$ is obtained. □

Note that according to the assumption $s_0 = 1$, the sub-index $o0 = 0$. Next, the following result analyzes the characteristics of the columns of the matrix IC.

Theorem 2. *The sequences* $\{\boldsymbol{d}_j\} = \{a_{k+oj} : k = 0, (2^S-1), 2\cdot(2^S-1), \ldots, (2^A-2)\cdot(2^S-1)\}$ $(j = 0, 1, \ldots, 2^{(S-1)} - 1)$ *corresponding to the columns of the matrix IC are shifted versions of a unique PN-sequence whose characteristic polynomial is given by:*

$$P_D(x) = (x + \alpha^N)(x + \alpha^{2\cdot N})(x + \alpha^{2^{2}\cdot N}) \ldots (x + \alpha^{2^{(A-1)}\cdot N}),$$

where N *is an integer defined as* $N = 2^0 + 2^1 + \ldots + 2^{(S-1)}$ *and* $\alpha \in GF(2^A)$ *a root of the primitive polynomial* $P_A(x)$*.*

Proof. Every sequence $\{\boldsymbol{d}_j\}$ corresponding to the *j-th* column of IC is a regular decimation of the *PN*-sequence $\{a_i\}$. More precisely, such a sequence is obtained by taking one out of $(2^S - 1)$ terms in $\{a_i\}$. The primality of A and S guarantees the primality of $(2^A - 1)$ and $(2^S - 1)$. Thus, the decimated sequence $\{\boldsymbol{d}_j\}$ is also a *PN*-sequence. In addition, as every $\{\boldsymbol{d}_j\}$ has been obtained from $\{a_i\}$ with a decimation ratio of value $(2^S - 1)$, then its characteristic polynomial $P_D(x)$ is the polynomial of the cyclotomic coset $(2^S - 1)$ in the Galois Field $GF(2^A)$ generated by the roots of the polynomial $P_A(x)$, see [4]. The starting point of each $\{\boldsymbol{d}_j\}$ is given by the corresponding sub-index oj. □

4 Cryptanalytic Attack on the Shrinking Generator

The cryptanalytic attack consists in the computation of the initial states of both registers SRA and SRS. From some known bits of the shrunken sequence we have to determine the first A bits $(a_0, a_1, \ldots, a_{A-1})$ of the sequence $\{a_i\}$ (initial state of SRA) as well as the first S bits $(s_0, s_1, \ldots, s_{S-1})$ of the sequence $\{s_i\}$ (initial state of SRS). The number of bits needed for the cryptanalysis is at most (A x S) bits, what is a minimum amount of shrunken sequence compared with the value of its linear complexity given by the equation (2). Nevertheless, these bits must be located at very particular positions inside the shrunken sequence. In fact, the needed bits are exclusively those ones located at the top-left corner (A x S) sub-matrix of IC. Remark that the bits required for the cryptanalysis are not all consecutive, since between two successive rows of the sub-matrix there are a great number of shrunken sequence bits (as many as $(2^{(S-1)} - S)$) whose knowledge is useless. The generation of the needed bits is straightly related with the register state succession. Indeed, each row of this sub-matrix is a portion of the shrunken sequence starting at the following register states:

- The same initial state of SRS.
- An initial state of SRA shifted $2^S - 1$ states from that one that generated the previous row of the sub-matrix.

The procedure is repeated systematically for every row of the sub-matrix. Clearly, the first row of the sub-matrix is generated from the initial states of SRA and SRS. After these considerations, this cryptanalytic attack can be divided into two different steps. In the first one, the computation of the initial state of SRA is carried out. In the second step and based on the SRA initial state, we determine the corresponding initial state of the register SRS.

4.1 Computation of the SRA Initial State

Previously to the computation of the initial state, the following result is introduced.

Lemma 1. *Given A bits of the shrunken sequence corresponding to A successive elements of any column of IC, the remaining bits of such a column can be determined.*

Proof. Theorem 2 defines $P_D(x)$, that is the characteristic polynomial of the PN-sequence corresponding to every column of IC. Thus, knowing A successive bits of any column and its characteristic polynomial, the linear recurrence relationship allows one to compute any of the remaining bits of such a PN-sequence. □

Now the computation of the SRA initial state is described in the next result.

Theorem 3. *Given A bits of the shrunken sequence corresponding to A successive elements of the first column of IC, the bits of the initial state of the register SRA can be determined.*

Proof. Lemma 1 shows that the knowledge of: i) A successive elements of the first column of IC and ii) its linear recurrence relationship, allows one to generate any other bit of such a column. On the other hand, from Theorem 1 we know that the $(n+1)$-*th* element of the first column of IC corresponds to $a_{n\cdot(2^S-1)}$, that is to say the $(n\cdot(2^S-1)+1)$-*th* term of the PN-sequence generated by the register SRA. Consequently, we first solve the following system of modular equations in the unknowns n_i

$$n_i \cdot (2^S - 1) \equiv i \ \ mod\,(2^A - 1) \qquad (i = 0, 1, \ldots, (A-1)),$$

and then, making use of the linear recurrence relationship, we compute the elements of the first column of IC at the positions (n_i+1)-*th* $(i = 0, 1, \ldots, (A-1))$. Such elements correspond to $a_0, a_1, \ldots, a_{A-1}$, respectively. □

4.2 Computation of the SRS Initial State

The computation of the SRS initial state is described in the next result.

Theorem 4. *Given $A\cdot S$ bits of the shrunken sequence corresponding to the top-left corner $(A$ x $S)$ sub-matrix of IC, the bits of the initial state of the register SRS can be determined.*

Proof. Firstly, from the linear recurrence relationship and theorem 3, we can compute $(A-1)$ blocks of A consecutive bits, B_i $(i = 1, 2, \ldots, (A-1))$, starting each of them at the (n_i+1)-*th* $(i = 1, 2, \ldots, (A-1))$ bit of the first column of IC, respectively.

$$SUB_{IC} = \begin{pmatrix} a_0 & a_{o1} & \cdots & a_{o(S-1)} \\ a_{2^S-1} & a_{(2^S-1)+o1} & \cdots & a_{(2^S-1)+o(S-1)} \\ a_{2\cdot(2^S-1)} & a_{2\cdot(2^S-1)+o1} & \cdots & a_{2\cdot(2^S-1)+o(S-1)} \\ a_{3\cdot(2^S-1)} & a_{3\cdot(2^S-1)+o1} & \cdots & a_{3\cdot(2^S-1)+o(S-1)} \\ \cdots & \cdots & \cdots & \cdots \\ a_{(A-1)\cdot(2^S-1)} & a_{(A-1)\cdot(2^S-1)+o1} & \cdots & a_{(A-1)\cdot(2^S-1)+o(S-1)} \end{pmatrix}$$

Secondly, since the sequence in every column of IC is exactly the same but starting at different points given by a_{oj}, we compare each block B_i with the corresponding column of the sub-matrix of IC. As soon as a coincidence is found the sub-index oj is univocally determined, that is $oj = i$. In addition, each sub-index oj indicates the position of the $(j+1)$-*th* 1 in the initial state of SRS while the intermediate bits are 0's. Thus, the above procedure can be repeated for $j = 1, 2, \cdots$ till we get $oj \geq (S-1)$. In this way, the initial state of the register SRS is thoroughly determined. □

4.3 An illustrative example

Let us consider a shrinking generator characterized by:

1. SRA with length $A = 5$, characteristic polynomial $P_A(x) = x^5 + x^4 + x^3 + x^2 + 1$ and output sequence $\{a_i\}$.
2. SRS with length $S = 4$, characteristic polynomial $P_S(x) = x^4 + x^3 + 1$ and output sequence $\{s_i\}$.
3. The characteristic polynomial of the shrunken sequence is $P_{ss}(x) = P_D(x)^p = (x^5 + x^3 + x^2 + x + 1)^8$.

Given 20 bits of the shrunken sequence corresponding to a (5 x 4) sub-matrix of IC

$$SUB_{IC} = \begin{pmatrix} 1\,0\,1\,1 \\ 1\,0\,0\,1 \\ 0\,1\,0\,1 \\ 0\,1\,1\,1 \\ 0\,0\,0\,1 \end{pmatrix},$$

we can launch a cryptanalytic attack against the shrinking generator in order to obtain the initial states of both LFSRs. Table 1 shows the calculations carried out for cryptanalyzing the above described generator. The most left column represents the indices n_i numbered $(0, 1, \ldots, 2^A - 2 = 30)$. Next column shows from Theorem 1 the position of the terms $(a_0, a_1, \ldots, a_4)$ of the sequence $\{a_i\}$ regarding the first column $\{\boldsymbol{d}_0\}$ of the matrix IC. The following columns of the Table 1 represent the matrix IC: in boldface the left-corner (5 x 4) sub-matrix with the known bits, the remaining bits of $\{\boldsymbol{d}_0\}$ are the bits computed to determine the initial states of SRA and SRS, and the symbol $-$ corresponds to unknown bits of the shrunken sequence that do not need to be computed for the cryptanalysis.

Computation of the SRA initial state: According to Theorem 3, we compute the positions of the $(n_i + 1)$-*th* elements of the first column of IC by solving the equation system

$$n_i \cdot 15 \equiv i \mod 31 \qquad (i = 0, 1, \ldots, 4),$$

That is, $n_0 = 0$, $n_1 = 29$, $n_2 = 27$, $n_3 = 25$, $n_4 = 23$. Then, by means of the characteristic polynomial $P_D(x)$ we determine the values of the $(n_i + 1)$-*th* $(i = 0, 1, \ldots, 4)$ elements of the first column $\{\boldsymbol{d}_0\}$ of IC. This is just a backward application of the linear recurrence relationship to the first column of the sub-matrix of IC, that is $a_{n+5} = a_{n+3} + a_{n+2} + a_{n+1} + a_n$ with $n \geq 30$. In fact, we get, $a_0 = 1$, $a_1 = 0$, $a_2 = 0$, $a_3 = 1$, $a_4 = 1$, see Table 1. Therefore, the initial state of the register SRA $(1, 0, 0, 1, 1)$ has been determined.

Computation of the SRS initial state: According to Theorem 4, we compute the relative shifts between consecutive columns in the matrix IC:

- *Computation of* $o1$*:* We know a_1 at the $(29 + 1)$-*th* position of the first column $\{\boldsymbol{d}_0\}$ and compute its $S - 1 = 4$ successive bits. We compare this block of 5 bits $B_1 = (0, 0, 1, 1, 0)$ with the the second column $\{\boldsymbol{d}_1\}$ of the sub-matrix $(0, 0, 1, 1, 0)'$, see Table 1 . There is coincidence, thus $o1 = 1$.

- *Computation of o2:* We know a_2 at the (27+1)-*th* position of the $\{\boldsymbol{d}_0\}$ and compute its 4 successive bits. We compare this block of 5 bits $B_2 = (0, 1, 0, 0, 1)$ with the third column $\{\boldsymbol{d}_2\}$ of the sub-matrix $(1, 0, 0, 1, 0)'$. There is no coincidence, thus we analyze the following bit a_3. We know a_3 at the $(25 + 1)$-*th* position of $\{\boldsymbol{d}_0\}$ and compute its 4 successive bits. We compare this block of 5 bits $B_3 = (1, 0, 0, 1, 0)$ with the third column $\{\boldsymbol{d}_2\}$ of the sub-matrix $(1, 0, 0, 1, 0)'$, see Table 1. There is coincidence, thus $o2 = 3$.

Table 1. Matrix IC corresponding to the described shrinking generator

n_i	$\{a_i\}$	$\boldsymbol{d}_0$	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$	$\boldsymbol{d}_3$	$\boldsymbol{d}_4$	$\boldsymbol{d}_5$	$\boldsymbol{d}_6$	$\boldsymbol{d}_7$
0	a_0	**1**	**0**	**1**	**1**	–	–	–	–
1		**1**	**0**	**0**	**1**	–	–	–	–
2		**0**	**1**	**0**	**1**	–	–	–	–
3		**0**	**1**	**1**	**1**	–	–	–	–
4		**0**	**0**	**0**	**1**	–	–	–	–
5		–	–	–	–	–	–	–	–
6		–	–	–	–	–	–	–	–
7		–	–	–	–	–	–	–	–
8		–	–	–	–	–	–	–	–
9		–	–	–	–	–	–	–	–
10		–	–	–	–	–	–	–	–
11		–	–	–	–	–	–	–	–
12		–	–	–	–	–	–	–	–
13		–	–	–	–	–	–	–	–
14		–	–	–	–	–	–	–	–
15		–	–	–	–	–	–	–	–
16		–	–	–	–	–	–	–	–
17		–	–	–	–	–	–	–	–
18		–	–	–	–	–	–	–	–
19		–	–	–	–	–	–	–	–
20		–	–	–	–	–	–	–	–
21		–	–	–	–	–	–	–	–
22		–	–	–	–	–	–	–	–
23	a_4	1	–	–	–	–	–	–	–
24		–	–	–	–	–	–	–	–
25	a_3	1	–	–	–	–	–	–	–
26		0	–	–	–	–	–	–	–
27	a_2	0	–	–	–	–	–	–	–
28		1	–	–	–	–	–	–	–
29	a_1	0	–	–	–	–	–	–	–
30		0	–	–	–	–	–	–	–

Since $o2 = 3 \geq S - 1$, we have determined the initial state of SRS. In fact, $s_0 = 1$, $o1 = 1$ implies $s_1 = 1$, $o2 = 3$ implies $s_2 = 0$ and $s_3 = 1$. Therefore, the SRS initial state is $(s_0, s_1, s_2, s_3) = (1, 1, 0, 1)$. Remark that only the knowledge of three columns of the sub-matrix has been necessary to identify the initial state of SRS. Indeed, the number of columns needed equals the number of '1' bits

in the initial state of the selector register. The maximum number of known bits corresponds to SRS initial state with all bits '1'. In the remaining cases, less bits are sufficient.

Once the initial states of both register are determined, the whole shrunken sequence that is the keystream sequence can be computed.

5 Generalization of this Technique to Interleaved Sequences

First of all, we introduce the general definition of interleaved sequence [12].

Definition 1. *Let $f(x)$ be a polynomial over $GF(q)$ of degree r and let m be a positive integer. For any sequence $\{u_k\}$ over $GF(q)$, we write $k = i \cdot m + j$ with $(i = 0, 1, \dots)$ and $(j = 0, \dots, m-1)$. If every sub-sequence $\{\boldsymbol{u}_j\}$ of $\{u_k\}$ defined as $\{u_{i \cdot m+j}\}$ is generated by $f(x)$, then the sequence $\{u_k\}$ is called an interleaved sequence over $GF(q)$ of size m associated with the polynomial $f(x)$.*

Table 2 shows the interleaved sequence $\{u_k\}$ over $GF(2)$ associated with the 3-degree characteristic polynomial $f(x) = x^3 + x + 1$ over $GF(2)$ and size $m = 4$. Reading by rows, the interleaved sequence is $\{u_k\} = \{1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0\}$ while by columns the sequence is made out of $\{\boldsymbol{u}_j\}$ $(j = 0, \dots, 3)$ four shifted versions of the PN-sequence generated by $f(x)$.

Table 2. Interleaved sequence with 4 shifted versions of the same PN-sequence

$\boldsymbol{u}_0$	$\boldsymbol{u}_1$	$\boldsymbol{u}_2$	$\boldsymbol{u}_3$
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

Interleaved sequences are currently used as keystream sequences with application in cryptography. See the introduction of [9] and the types of keystream sequences enumerated there. They can be generated in different ways:

1. By a LFSR controlled by another LFSR (which may be the same one) e.g. multiplexed sequences [11], clock-controlled sequences [1], cascaded sequences [7], shrinking generator sequences [5] etc.

2. By one or more than one LFSR and a feed-forward nonlinear function e.g. Gold-sequence family, Kasami (small and large set) sequence families, GMW sequences, Klapper sequences, No sequences etc. See [9] and the references cited therein.

In brief, a large number of well-known cryptographic sequences are included in the class of interleaved sequences. Next, the link between interleaved sequences and shrunken sequences is expressed in the following result.

Theorem 5. *Shrunken sequences are interleaved sequences of size* $2^{(S-1)}$.

Proof. Let $\{z_k\}$ be a shrunken sequence with characteristic polynomial $P(x)^p$ where $P(x)$ is an A-degree primitive polynomial and p is an integer in the interval $2^{(S-2)} < p \leq 2^{(S-1)}$. According to the interleaved configuration IC, we may express $\{z_k\}$ in terms of m sequences $\{\boldsymbol{z}_j\}$ where $\{\boldsymbol{z}_j\} = \{z_{i \cdot m+j}\}$ with $i \geq 0$, $m = 2^{(S-1)}$ and $(j = 0, \ldots, m-1)$. Since by Theorem 2 the sequences $\{\boldsymbol{z}_j\}$ are generated by the same characteristic polynomial $P_D(x)$, we get that the shrunken sequence $\{z_k\}$ is an interleaved sequence of size $2^{(S-1)}$ associated with the polynomial $P_D(x)$. □

The previous theorem proves that shrunken sequences are interleaved sequences. Moreover, section 4 shows that the knowledge of a number of bits of the shrunken sequence allows one to lauch a cryptanalytic attack against the shrinking generator. As many cryptographic sequence generators produce interleaved sequences, then the previous considerations take us into the following conjecture:

Conjecture 1. Given a number of bits corresponding to an initial sub-matrix of the interleaved configuration IC of an interleaved sequence, it is possible to obtain the whole interleaved sequence.

The confirmation of this conjecture would prove the weakness of interleaved generators for cryptographic purposes.

6 Conclusions

In this work, a new cryptanalytic attack against the class of shrinking generators has been proposed. The amount of intercepted bits necessary to realize such an attack is much lesser than that of other standard cryptanalysis. The basic idea consists in defining the shrunken sequence as an interleaved sequence. Hence the weaknesses inherent to interleaved sequences can be advantageously used in the practical attack. A direct consequence of this technique is its generalization to other interleaved sequence generators of cryptographic purpose. In this way, the security of this kind of generators must be carefully checked.

References

1. T. Beth, F. Piper, The Stop-and-Go Generator, in Proceedings of EUROCRYPT'84, in: Lecture Notes in Computer Science, vol. 228, Springer Verlag, 1985, pp. 228-238.

2. Bluetooth, *Specifications of the Bluetooth system,* Version 1.1, February 2001, http://www.bluetooth.com/
3. P. Caballero-Gil, A. Fúster-Sabater, A Wide Family of Nonlinear Filter Functions with a Large Linear Span, Information Sciences, 164 (2004) 197-207.
4. P. Caballero-Gil, A. Fúster-Sabater, Using Linear Hybrid Cellular Automata to Attack the Shrinking Generator, IEICE Transactions on Fundamentals of Electronics Communications and Computer, E89-A (2006) 1166-1172.
5. D. Coppersmith, H. Krawczyk, H. Mansour, The Shrinking Generator, in Proceedings of CRYPTO'93, in: Lecture Notes in Computer Science, vol. 773, Springer-Verlag, 1994, pp. 22-39.
6. A. Fúster-Sabater, Run Distribution in Nonlinear Binary Generators, Applied Mathematics Letters 17 (2004) 1427-1432.
7. D. Gollmann, W.G. Chambers, Clock-Controlled Shift Register, IEEE J. Selected Areas Commun 7 (1989) 525-533.
8. S. Golomb, Shift-Register Sequences, Aegean Park Press, Laguna Hill California (1982).
9. G. Gong, Theory and Applications of q-ary Interleaved Sequences, IEEE Trans. Information Theory 41 (2) (1995) 400-411.
10. GSM, *Global Systems for Mobile Communications,* available at http://cryptome.org/gsm-a512.htm
11. S.M. Jennings, Multiplexed Sequences: Some Properties, in Proceedings of EUROCRYPT'83, in: Lecture Notes in Computer Science, vol. 149, Springer Verlag, 1983, pp. 210-221.
12. S. Jiang, Z. Dai and G. Gong. On interleaved sequences over finite fields. Discrete Maths, 252 (2002) 161-178.
13. A. Kanso, Clock-Controlled Shrinking Generator of Feedback Shift Registers, in: Lecture Notes in Computer Science, vol. 2727, Springer Verlag, 2003, pp. 443-451.
14. R. Lidl, H. Niederreiter, Introduction to Finite Fields and Their Applications, Cambridge, England: Cambridge University Press, 1986.
15. R.L. Rivest, RSA Data Security, Inc., March 12, 1998.
16. I. Shparlinski, On Some Properties of the Shrinking Generator, Designs, Codes and Cryptography 23 (2001) 147-156.
17. L. Simpsom, J. Golic, E. Dawson, A Probabilistic Correlation Attack on the Shrinking Generator, in Proceedings of EUROCRYPT'98, in: Lecture Notes in Computer Science, vol. 1438, Springer Verlag, 1998, pp. 147-158.