Recovering decimation-based cryptographic sequences by means of linear CAs[*]

Sara D. Cardell[1] and Amparo Fúster-Sabater[2]

[1]Imecc, Unicamp, Campinas, Brazil
[2]ITEFI, CSIC, Madrid, Spain

Abstract

The sequences produced by the cryptographic sequence generator known as the shrinking generator can be modelled as the output sequences of linear elementary cellular automata. These sequences are composed of interleaved m-sequences produced by linear structures based on feedback shifts. This profitable characteristic can be used in the cryptanalysis of this generator. In this work we propose an algorithm that takes advantage of the inherent linearity of these cellular automata and the interleaved m-sequences. Although irregularly decimated generators have been conceived and designed as non-linear sequence generators, in practice they can be easily analysed in terms of simple linear structures.

keywords: *decimation, shrinking generator, cellular automata, Zech logarithm, cryptanalysis*
MSC2010: 94A55

1 Introduction

Stream ciphers are cryptographic primitives used to ensure privacy in digital communication [16]. Their procedure consists in generating a long sequence as random as possible, called the keystream sequence, from a short secret key and a public algorithm (the sequence generator). For encryption, the sender performs a bit-wise XOR operation among the bits of the keystream sequence and the message (or plaintext). The resultant message (or ciphertext) is sent to the receiver. For decryption, the receiver generates the same keystream and performs the same bit-wise XOR operation between the ciphertext and the keystream to recover the original message.

Most keystream generators are based on maximal-length Linear Feedback Shift Registers (LFSRs) [8] whose output sequences, the so-called m-sequences, are combined by means of nonlinear functions to produce pseudorandom sequences for cryptographic applications. Desirable properties for such sequences can be enumerated as follows: a) Long Period, b) Good statistical properties, c) Large Linear Complexity [15].

Among the current keystream generators used in this type of cryptography, the class of irregularly decimated generators is one of the most popular [1, 14]. The underlying idea of this kind of generators is the irregular decimation of an m-sequence according to the bits of another m-sequence. The result of this decimation process is a new sequence that will be used as keystream sequence in stream ciphers. One of the most representative elements of this family is the shrinking generator introduced by Coppersmith *et al.* in [6]. The sequence produced by this generator is called the shrunken sequence and has good cryptographic properties.

In [3], the authors showed that the shrunken sequence is composed of interleaving m-sequences generated by the same primitive polynomial. Furthermore, in [3] it was proven that the shrunken sequence can be modelled as one of the (vertical) sequences produced by a linear cellular automata (CAs). This CA produces other sequences, which we call companion sequences of the shrunken sequence. Such sequences have the same characteristic polynomial as the shrunken sequence and are composed of interleaving m-sequences as well. In this work, we use all these properties to propose an algorithm that recovers the

shrunken sequence given a small quantity of intercepted bits of such a sequence. In fact, the number of bits needed for the cryptanalysis is dramatically reduced when compared with the shrunken sequence period.

The paper is organized as follows: In Section 2, some basic concepts and definitions are provided for the understanding of the work. In Section 3, we introduce some properties of the shrunken sequence as well as the characterization of the 102-CA that generates the shrunken sequence as one of its (vertical) sequences. Next, in Section 4, we review the main properties of the Zech logarithm, which will play an important role in the recovering algorithm. In Section 5, an algorithm to recover the initial state of both registers in the shrinking generator is proposed. Finally, the paper concludes with Section 6.

2 Preliminaries

In this section we introduce the basics to understand the next sections. In Subsection 2.1, we remind some basic concepts about sequences and cryptography. In Subsections 2.2 and 2.3, we provide the definition of the main concepts used throughout this paper: shrinking generator and cellular automaton.

2.1 Basic concepts

Let $\mathbb{F}_2$ be the Galois field of two elements. A sequence $\{a_i\} = \{a_0, a_1, \ldots\}$ is a binary sequence or a sequence over $\mathbb{F}_2$ if its terms $a_i \in \mathbb{F}_2$, for $i = 0, 1, \ldots$. Besides, the sequence $\{a_i\}$ is said to be **periodic** if and only if there exists an integer T such that $a_{i+T} = a_i$, for all $i \geq 0$.

Let l be a positive integer, and let $c_0, c_1, \ldots, c_{l-1} \in \mathbb{F}_2$. A binary sequence $\{a_i\}$ satisfying the recurrence relation

$$a_{i+l} = c_0 a_i + c_1 a_{i+1} + \cdots + c_{l-2} a_{i+l-2} + c_{l-1} a_{i+l-1}, \quad i \geq 0, \tag{1}$$

is called an (l-th order) **linear recurring sequence** in $\mathbb{F}_2$. The first l terms $\{a_0, a_1, \ldots, a_{l-1}\}$ determine the rest of the sequence uniquely and are known as the **initial state**. A relation of the form given in expression (1) is called an (l-th order) linear recurrence relation.

The monic polynomial of degree l given by

$$p(x) = c_0 + c_1 x + \cdots + c_{l-2} x^{l-2} + c_{l-1} x^{l-1} + x^l \in \mathbb{F}_2[x],$$

is called the **characteristic polynomial** of the sequence $\{a_i\}$ and this is said to be generated by $p(x)$.

The generation of linear recurring sequences can be implemented by **Linear Feedback Shift Registers** (LFSRs) [8]. These are electronic devices with l memory cells or stages which handle information in the form of elements of $\mathbb{F}_2$ and that are based on shifts and linear feedback. If the characteristic polynomial $p(x)$ is primitive, then the LFSR is said to be maximal-length and the output sequence has period $2^l - 1$. This output sequence is called **m-sequence** (maximal sequence) or **PN-sequence** (pseudonoise sequence).

Example 1: Given the primitive polynomial $p(x) = 1 + x^2 + x^5$, the linear recurrence relation is given by:

$$a_{i+5} = a_{i+2} + a_i, \text{ for } i = 0, 1, 2, \ldots$$

Given a initial state $\{1, 0, 0, 0, 0\}$, the m-sequence generated is the following

$$\{1\,0\,0\,0\,0\,1\,0\,0\,1\,0\,1\,1\,0\,0\,1\,1\,1\,1\,1\,0\,0\,0\,1\,1\,0\,1\,1\,1\,0\,1\,0\}$$

with maximum period $2^5 - 1 = 31$. ■

The **linear complexity**, LC, of a sequence $\{a_i\}$ is defined as the length of the shortest LFSR that generates such a sequence. The linear complexity is related with the amount of sequence needed to reconstruct the whole sequence. In cryptographic terms, the linear complexity must be as large as possible in order to prevent the application of the Berlekamp-Massey algorithm [12]. This algorithm efficiently computes the length and characteristic polynomial of the shortest LFSR given at least $2LC$ sequence bits. A recommended value for LC is about half the sequence period.

Due to their low linear complexity, LFSRs are never used alone as keystream generators. In fact, m-sequences generated by LFSRs have good statistical properties, desirable for keystream design, but their linearity has to be destroyed, i.e., their linear complexity has to be increased before such sequences are used for cryptographic purposes.

2.2 The shrinking generator

The **shrinking generator** was first introduced by Coppersmith *et al.* in [6]. It consists of two m-sequences where one of them decimates the other one. Given two m-sequences $\{a_i\}$ and $\{b_i\}$ ($i \geq 0$) generated by two LFSRs of length L_1 and L_2, respectively, the decimation rule is very simple:

$$\begin{cases} \text{If } a_i = 1 \text{ then } s_j = b_i, \\ \text{If } a_i = 0 \text{ then } b_i \text{ is discarded}, \end{cases}$$

where the generated sequence, $\{s_j\}$ ($j \geq 0$), is said to be the **shrunken sequence**. If $\gcd(L_1, L_2) = 1$, then the period of such sequence is $T = (2^{L_2} - 1)2^{L_1 - 1}$. Furthermore, the characteristic polynomial of this sequence is given by $p(x)^L$, where $p(x)$ is a primitive polynomial of degree L_2 and $2^{L_1 - 2} < L \leq 2^{L_1 - 1}$. Therefore, the linear complexity of the shrunken sequence is given by $LC = L \cdot L_2$. As usual, the key of this generator is the initial state of both registers.

Example 2: Consider the register R_1 with characteristic polynomial $p_1(x) = 1 + x + x^2$ and initial state $\{1, 1\}$. Next, consider the register R_2 with characteristic polynomial $p_2(x) = 1 + x + x^3$ and initial state $\{1, 1, 1\}$. Then the shrunken-sequence can be computed as follows:

$$\begin{array}{llllllllllllllllllllll}
\{a_i\}: & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
\{b_i\}: & 1 & 1 & \cancel{0} & 0 & 0 & \cancel{1} & 0 & 1 & \cancel{1} & 1 & 0 & \cancel{0} & 1 & 0 & \cancel{1} & 1 & 1 & \cancel{0} & 0 & 1 & \cancel{0} \\
\{s_j\}: & \mathbf{1} & \mathbf{1} & & \mathbf{0} & \mathbf{0} & & \mathbf{0} & \mathbf{1} & & \mathbf{1} & \mathbf{0} & & \mathbf{1} & \mathbf{0} & & \mathbf{1} & \mathbf{1} & & \mathbf{0} & \mathbf{1} &
\end{array}$$

The shrunken sequence $\{s_j\}$ (in bold) has period 14 and it is not difficult to check that its characteristic polynomial is $p(x)^2 = (1 + x^2 + x^3)^2$. Therefore, its linear complexity is $LC = 6$. ∎

2.3 Cellular Automaton

Cellular automata (CAs) were first introduced by von Neumann as simple models to study biological processes such as self-reproduction [13]. An elementary one-dimensional CA consists of an arrangement of cells (with binary contents in this work) where the value of each cell evolves deterministically according to a set of rules involving its k nearest neighbours. Thus, the state of the cell in position i at time $t + 1$, denoted by x_i^{t+1}, depends on the state of the k closest cells at time t. If these rules are composed exclusively of XOR operations, then the CA is **linear**. There are several types of CA: **null** (null cells are supposed to be adjacent to extreme cells) or **periodic** (extreme cells are adjacent), **regular** (every cell uses the same updating rule) or **hybrid** (different rules are applied to distinct cells).

The rules considered in this work are:

Rule 102: $x_i^{t+1} = x_i^t + x_{i+1}^t$

111	110	101	100	011	010	001	000
0	1	1	0	0	1	1	0

Rule 60: $x_i^{t+1} = x_{i-1}^t + x_i^t$

111	110	101	100	011	010	001	000
0	0	1	1	1	1	0	0

Recall that the numbers 01100110 and 00111100 are the binary representations of 102 and 60, respectively. This is the reason why they are called rule 102 and rule 60.

Given a CA of length l and initial state of the same length, it generates l (vertical) sequences. In Table 1, we have a regular, periodic, 102-CA, with initial state $\{1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1\}$ that generates 14 (vertical) sequences.

3 Modelling the shrunken sequence

In this section, we underline some properties of the shrunken sequence, which will be used in the algorithm proposed in Section 5. From now on, we consider two registers R_1 and R_2, with characteristic polynomials $p_1(x)$, $p_2(x) \in \mathbb{F}_2[x]$, lengths L_1 and L_2 ($\gcd(L_1, L_2) = 1$) and the periods of their corresponding m-sequences are $T_1 = 2^{L_1} - 1$ and $T_2 = 2^{L_2} - 1$, respectively. Besides, the m-sequences generated by both registers are denoted by $\{a_i\}$ and $\{b_i\}$ ($i \geq 0$), respectively. Since only the 1s of $\{a_i\}$ generate shrunken sequence bits, we assume without loss of generality that $a_0 = 1$. As stated before, the shrunken sequence $\{s_j\}$ ($j \geq 0$) generated by both registers has period $T = 2^{L_1 - 1}(2^{L_2} - 1)$ and characteristic polynomial $p(x)^L$, with $2^{L_1 - 2} < L \leq 2^{L_1 - 1}$.

Table 1: CA that generates the shrunken sequence in Example 2

102	102	102	102	102	102	102	102	102	102	102	102	102	102
1	0	1	0	1	1	0	0	0	1	1	1	0	1
1	1	1	1	0	1	0	0	1	0	0	1	1	0
0	0	0	1	1	1	0	1	1	0	1	0	1	1
0	0	1	0	0	1	1	0	1	1	1	1	0	1
0	1	1	0	1	0	1	1	0	0	0	1	1	1
1	0	1	1	1	1	0	1	0	0	1	0	0	1
1	1	0	0	0	1	1	1	0	1	1	0	1	0
0	1	0	0	1	0	0	1	1	0	1	1	1	1
1	1	0	1	1	0	1	0	1	1	0	0	0	1
0	1	1	0	1	1	1	1	0	1	0	0	1	0
1	0	1	1	0	0	0	1	1	1	0	1	1	0
1	1	0	1	0	0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	1	0	1	0	1	1	0	0
1	0	0	1	1	0	1	1	1	1	0	1	0	0

3.1 Properties of the shrunken sequence

If the shrunken sequence $\{s_j\}$ is decimated by distance $d = 2^{L_1-1}$ starting at position s_i, $i = 0, 1, \ldots, d-1$, then we obtain d m-sequences denoted by $\{s_{d \cdot j+i}\}$, for $i = 0, 1, \ldots, d-1$ and $j \geq 0$. Such sequences are called the **interleaved m-sequences** of the shrunken sequence and its characteristic polynomial is again $p(x)$. The polynomial $p(x)$ can be computed as follows,

$$p(x) = (x + \alpha^{T_1})(x + \alpha^{2T_1})(x + \alpha^{4T_1}) \cdots (x + \alpha^{2^{L_2-1}T_1}),$$

where α is root of $p_2(x)$ (see [3, Theorem. 3.3]).

Let us see an illustrative example.

Example 3: Consider two registers with characteristic polynomials $p_1(x) = 1 + x + x^3$ and $p_2(x) = 1 + x + x^4$ and initial states $\{1, 0, 0\}$ and $\{1, 0, 0, 0\}$, respectively. The shrunken sequence given by

$$\{1000\ 1111\ 1010\ 0001\ 1001\ 0110\ 1100\ 1101\ 0100\ 0010\ 1110\ 0011\ 0111\ 0101\ 1011 \ldots\}$$

has period $T = 60$ and characteristic polynomial $p(x)^4 = (1 + x^3 + x^4)^4$. If we decimate the shrunken sequence by 4, then we find that the shrunken sequence is composed of 4 m-sequences:

$$
\begin{array}{cccccccccccccccc}
 & & b_0 & b_7 & b_{14} & b_6 & b_{13} & b_5 & b_{12} & b_4 & b_{11} & b_3 & b_{10} & b_2 & b_9 & b_1 & b_8 \\
 & & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
\{s_{4j}\} & \to & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
\{s_{4j+1}\} & \to & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\
\{s_{4j+2}\} & \to & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
\{s_{4j+3}\} & \to & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{array}
\tag{2}
$$

These four m-sequences $\{s_{4j+i}\}$ ($0 \leq i < 4$) have the same characteristic polynomial $p(x) = 1 + x^3 + x^4$, thus all of them are shifted versions of one single m-sequence. This shift depends on the positions of the 1s in the m-sequence $\{a_i\}$ (see Section 3.1.2). The bits $⓪$, $\boxed{0}$ and $\Diamond$ in the sequence $\{s_{4j}\}$ represent the starting point of the sequences $\{s_{4j+i}\}$ ($1 \leq i \leq 3$), respectively. ∎

3.1.1 Relation between register R_2 and the shrunken sequence

Given the shrunken sequence $\{s_j\}$, the following results help us to find the m-sequence $\{b_i\}$ generated by register R_2.

Proposition 1: Let $\delta \in \{1, 2, 3, \ldots, T_2 - 1\}$, such that $T_1 \delta = 1 \bmod T_2$. If the first interleaved m-sequence is decimated by distance δ, then the resultant sequence is $\{b_i\}$.

Proof 1: According to the properties of the m-sequences, $\{a_i\}$ has 2^{L_1-1} ones in the first T_1 bits [8]. Besides, we know that the m-sequence $\{a_i\}$ decimates $\{b_i\}$ to obtain the shrunken sequence. Therefore,

the first interleaved m-sequence of the shrunken sequence is $\{b_0, b_{T_1}, b_{2T_1}, \ldots, b_{(T_2-1)T_1}\}$. If this sequence is decimated by distance δ, then the following sequence is obtained: $\{b_0, b_{\delta T_1}, b_{2\delta T_1}, \ldots, b_{(T_2-1)\delta T_1}\}$. We know that $T_1\delta = 1 \bmod T_2$ and, therefore, the sequence can be seen as $\{b_0, b_1, b_2, \ldots, b_{(T_2-1)}\}$, which is the m-sequence generated by the register R_2. $\qquad\square$

The previous proposition leads us to the following results.

Corollary 1: If the shrunken sequence is decimated by distance $2^{L_1-1}\delta$, then the m-sequence $\{b_i\}$ is obtained.

Corollary 2: If the primitive polynomials $p_1(x)$, $p_2(x) \in \mathbb{F}_2[x]$ have degrees L_1 and L_1+1, respectively, then $\delta = T_2 - 2$.

Proof 2: We proceed with the following computations:

$$(T_2-2)T_1 = (2^{L_1+1}-3)(2^{L_1}-1) = 2^{2L_1+1} - 3\cdot 2^{L_1} - 2^{L_1+1} + 3 = 2^{2L_1+1} - (2^{L_1+1}-1)$$
$$-(2^{L_1+1}-1) - 2^{L_1} + 1 = 2^{L_1}(2^{L_1+1}-1) - 2(2^{L_1+1}-1) + 1.$$

Since $T_2 = 2^{L_1+1} - 1$, then $(T_2-2)T_1 = 1 \bmod T_2$. Consequently, we have that $\delta = T_2 - 2$. $\qquad\square$

In Example 3, we had that $L_1 = 3$ and $L_2 = 4$. In this case, we can apply Corollary 2 and $\delta = T_2 - 2 = 13$. If we decimate the first interleaved m-sequence by 13 (see expression (2)), then we obtain $\{b_i\}$, the m-sequence generated by $p_2(x) = 1 + x + x^4$. In this case $\{b_i\} = \{100010011010111\}$.

3.1.2 Relation between register R_1 and the shrunken sequence

In this section we analyse how to recover the m-sequence $\{a_i\}$ from the shrunken sequence $\{s_j\}$. Assume the first interleaved m-sequence is denoted by $\{v_i\}$. Since the other interleaved m-sequences are the same but shifted, we assume they have the form $\{v_{d_1+i}\}, \{v_{d_2+i}\}, \ldots, \{v_{d_{2^{L_1-1}-1}+i}\}$ for some $d_i \in \{0, 1, 2, \ldots, 2^{L_2-2}\}$:

$$
\begin{array}{cccccc}
\{v_i\}: & v_0 & v_1 & v_2 & \cdots & v_{T_2-1} \\
\{v_{d_1+i}\}: & v_{d_1} & v_{d_1+1} & v_{d_1+2} & \cdots & v_{d_1+T_2-1} \\
\{v_{d_2+i}\}: & v_{d_2} & v_{d_2+1} & v_{d_2+2} & \cdots & v_{d_2+T_2-1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\{v_{d_{2^{L_1-1}-1}+i}\}: & v_{d_{2^{L_1-1}-1}} & v_{d_{2^{L_1-1}-1}+1} & v_{d_{2^{L_1-1}-1}+2} & \cdots & v_{d_{2^{L_1-1}-1}+T_2-1}
\end{array}
$$

In order to illustrate this idea, consider again Example 3. In this case, we had four interleaved m-sequences

$$\{s_{4j}\} = \{v_i\}, \quad \{s_{4j+1}\} = \{v_{d_1+i}\}, \quad \{s_{4j+2}\} = \{v_{d_2+i}\} \quad \text{and} \quad \{s_{4j+3}\} = \{v_{d_3+i}\}.$$

In is well known that a maximum-length LFSR of length L, produces an m-sequence with 2^{L-1} ones in the first period. Now, we are ready to introduce the following result.

Proposition 2: Let $\{0, i_1, i_2, \ldots, i_{2^{L_1-1}-1}\}$ be the set of positions of the 1s in the m-sequence $\{a_i\}$ in its first period. Therefore, $d_k = \delta \cdot i_k \bmod (2^{L_1-1}-1)$, for $k = 1, 2, \ldots, 2^{L_1-1}-1$, where δ has the form given in Proposition 1.

In Example 3, we had four interleaved m-sequences $\{v_i\}$, $\{v_{i+d_1}\}$, $\{v_{i+d_2}\}$ and $\{v_{i+d_3}\}$. It is easy to check, from expression (2), that $d_1 = 9$, $d_2 = 5$ and $d_3 = 3$. In this case, $T_1 = 7$ and $T_2 = 15$, then, according to Corollary 2, $\delta = 13$. With this information, we can compute the positions of the 1s in $\{a_i\}$ ($i_0 = 0$, without loss of generality):

$$13 \cdot i_1 = 9 \bmod 15 \rightarrow i_1 = 3$$
$$13 \cdot i_2 = 5 \bmod 15 \rightarrow i_2 = 5$$
$$13 \cdot i_3 = 3 \bmod 15 \rightarrow i_3 = 6$$

Therefore, the set of positions is given by $\{0, 3, 5, 6\}$ and then the m-sequence is $\{a_i\} = \{1, 0, 0, 1, 0, 1, 1\}$. ∎

Table 2: General 102-CA

102	102	102	102	102	...	102	...
s_0	$s_0 + s_1$	$s_0 + s_2$	$s_0 + s_1 + s_2 + s_3$	$s_0 + s_4$	...	$s_0 + s_8$	...
s_1	$s_1 + s_2$	$s_1 + s_3$	$s_1 + s_2 + s_3 + s_4$	$s_1 + s_5$	...	$s_1 + s_9$	...
s_2	$s_2 + s_3$	$s_2 + s_4$	$s_2 + s_3 + s_4 + s_5$	$s_2 + s_6$	...	$s_2 + s_{10}$	...
s_3	$s_3 + s_4$	$s_3 + s_5$	$s_3 + s_4 + s_5 + s_6$	$s_3 + s_7$	...	$s_3 + s_{11}$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	

3.2 Characterization of the CA

In [3, Theorem 3.10], the authors showed that there exists a 102-CA of length $L = \frac{T}{\gcd(2^{L_2}-1, D)}$ that generates the shrunken sequence as one of the (vertical) sequences. We remind that T is the period of the shrunken sequence and $D = \mathcal{Z}_\alpha(1)$, where $\mathcal{Z}_\alpha(1)$ is the Zech logarithm of 1 on basis α [9] (see also Definition 1 in Section 4).

Furthermore, it was shown that there are 2^{L_1-1} different sequences that appear repeated several times along the CA. Such sequences have the same characteristic polynomial $p(x)^L$ as that of the shrunken sequence [3, Theorem 3.9]. We call these sequences the **companion sequences** of the shrunken sequence.

Example 4: In Example 2, we obtained a shrunken sequence of period $T = 14$ and characteristic polynomial $p(x)^2 = (1 + x^2 + x^3)^2$. In Table 1, it is possible to see how to obtain this sequence as one of the output (vertical) sequences of a 102-CA of length 14. It is possible to check that there is another sequence apart from the shrunken sequence that appears shifted along the CA: $\{0\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\}$. Both sequences appear seven times, but shifted each time $D = 2 \cdot \mathcal{Z}_\alpha(1) = 10$ positions. ∎

3.3 Companion sequences

We have seen that if we locate the shrunken sequence in the zero column of the CA, 2^{L_1-1} different sequences are generated, including the shrunken sequence. All these sequences have the same characteristic polynomial $p(x)^L$ and are composed by interleaving 2^{L_1-1} m-sequences with characteristic polynomial $p(x)$ [3].

In Example 4, two sequences were generated by the CA, the shrunken sequence and another sequence with the same period and the same characteristic polynomial $p(x)^2 = (1 + x^2 + x^3)^2$. If we decimate both sequences by 2, we can see that both sequences are composed of two m-sequences whose characteristic polynomial is $p(x) = 1 + x^2 + x^3$. This means, that all the m-sequences are the same but shifted (see Table 3)

Consequently, we can deduce that the fact of knowing some bits of the companion sequences of the CA can help us to recover parts of the shrunken sequence. More precisely, we can recover the first interleaved m-sequence of the shrunken sequence using the interleaved m-sequences of the same shrunken sequence and the interleaved m-sequences of the companion sequences.

Consider a linear 102-CA and consider that the sequence in the zero column is the shrunken sequence $\{s_i\}$. The form of the corresponding companion sequences is computed in Table 2. It is not difficult to check that the companion sequences in columns whose indices are 2^j, for $j = 0, 1, 2, \ldots$, have the form $\{s_i + s_{i+2^j}\}$. In fact, the general form of these columns can be found in [2].

Let us denote the interleaved m-sequences of the shrunken sequence by $\{v_{d_0^0+i}\}$, $\{v_{d_1^0+i}\}$, $\{v_{d_2^0+i}\}$, $\ldots, \{v_{d_{2^{L_1-1}-1}^0+i}\}$, $i \geq 0$, where $d_0^0 = 0$. Remember that the positions d_k^0 depend on the location of the 1s in the m-sequence $\{a_i\}$ generated by the first register R_1 (see Section 3.1.2).

Let us denote the interleaved m-sequences of the first companion sequence in the CA by $\{v_{d_0^1+i}\}$, $\{v_{d_1^1+i}\}$, $\{v_{d_2^1+i}\}$, $\ldots, \{v_{d_{2^{L_1-1}-1}^1+i}\}$, $i \geq 0$. We can compute these new positions using the definitions of rule 102 and Zech logarithm as follows

$$d_k^1 = \mathcal{Z}_\alpha(d_k^0 - d_{k+1}^0) + d_{k+1}^0, k = 0, 1, \ldots, 2^{L_1-1} - 2,$$
$$d_{2^{L_1-1}-1}^1 = \mathcal{Z}_\alpha(d_{2^{L_1-1}-1}^0 - 1) + 1.$$

Table 3: Interleaved m-sequences of the shrunken sequence and the companion sequences of Example 3

(a)		(b)		(c)		(d)	
1	1			0	0	1	0
0	0	0	◇1◇	1	0	1	1
0	①1①	0	0	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	0	1	0	0
1	1	1	1	1	0	0	1
0	1	0	1	0	1	1	0
		□1□	0				

Similarly, we can compute the shifts positions for the j-th companion sequence, $j = 1, 2, \ldots, L - 1$ as:

$$d_k^j = \mathcal{Z}_\alpha(d_k^{j-1} - d_{k+1}^{j-1}) + d_{k+1}^{j-1}, k = 0, 1, \ldots, 2^{L_1-1} - 2, \tag{3}$$

$$d_{2^{L_1-1}-1}^j = \mathcal{Z}_\alpha(d_{2^{L_1-1}-1}^{j-1} - 1) + 1.$$

Notice that the companion sequences in columns $t \cdot 2^{L_1-1}$, with $t = 1, 2, \ldots, L/(2^{L_1-1}) - 1$ in the CA, are again the shrunken sequence, but starting in positions $t \cdot D \cdot 2^{L_1-1}$, respectively [3, Theorem 3.8]. Moreover, the companion sequence in the column $t \cdot 2^{L_1-1} + m$ for $m = 1, 2, \ldots, 2^{L_1-1} - 1$ and $t = 0, 1, \ldots, L/(2^{L_1-1}) - 1$ is the same as the companion sequence in the m-th column starting in position $t \cdot D \cdot 2^{L_1-1}$. Therefore, we have that:

$$d_k^{t \cdot 2^{L_1-1}+m} = d_k^m + t \cdot D \mod (2^{L_2} - 1)$$

for $k = 0, 1, \ldots, 2^{L_1-1} - 1$, $m = 0, 1, \ldots, 2^{L_1-1} - 1$ and $t = 0, 1, \ldots, L/(2^{L_1-1}) - 1$.

This means that, the positions d_k^j for the companion sequence in the j-th column with $j \geq 2^{L_1-1}$ can be computed easily using the positions d_i^s, with $0 \leq s < 2^{L_1-1}$ (without logarithms).

Example 5: Consider the shrunken sequence in Example 2. The CA in Table 1 generates two sequences, the shrunken sequence and one companion sequence. Both sequences are composed of interleaved m-sequences (see Table 3).

Let us consider the sequence in the 7-th column of the CA. In this particular case, we have that $t = 3$, $D = 5$ and $2^{L_1-1} = 2$. Therefore, this sequence is the same sequence as the companion sequence but starting in position $t \cdot D \cdot 2^{2^{L_1-1}} \mod 14 = 2$ (see red bits in Table 1). Furthermore, the two interleaved m-sequences of this sequence are the same as the first interleaved m-sequence of the shrunken sequence (Table 3a) starting in positions:

$$d_0^7 = d_0^1 + D \cdot 3 \mod 7 = 2 \quad \text{and} \quad d_1^7 = d_1^1 + D \cdot 3 \mod 7 = 1, \text{respectively.}$$

Let us consider the column in the 10-th position, that is, $t = 5$. Then, this sequence is the shrunken sequence, but starting in position $t \cdot D \cdot 2^{L_1-1} \mod 14 = 8$ (see the green bits in Table 1). Furthermore, the two interleaved m-sequences this sequence is composed of are the same as the first interleaved m-sequence of the shrunken sequence (Table 3a) starting in positions:

$$d_0^{10} = d_0^0 + D \cdot 5 \mod 7 = 4 \quad \text{and} \quad d_1^{10} = d_1^0 + D \cdot 5 \mod 7 = 2, \text{respectively.}$$

∎

4 Zech's logarithm

Zech logarithms are named after Julius Zech which published a table of these type logarithms (which he called *addition logarithms*) for doing arithmetic in $\mathbb{Z}_p$. These logarithms are also called as Jacobi logarithms after C. G. J. Jacobi who used them for number theoretic investigations [10].

Assume we are working over the finite field $\mathbb{F}_q$, where $q = p^m$, with p prime and m a positive integer. We now introduce the definition of Zech logarithm.

Definition 1: Let $\alpha \in \mathbb{F}_q$ be a primitive element. The **Zech logarithm** with basis α is the application $\mathcal{Z}_\alpha : \mathbb{Z}_q \to \mathbb{Z}_q^* \cup \{\infty\}$, such that each element $t \in \mathbb{Z}_q$ corresponds to $\mathcal{Z}_\alpha(t)$, attaining $1 + \alpha^t = \alpha^{\mathcal{Z}_\alpha(t)}$.

Example 6: Let $\alpha \in \mathbb{F}_{2^3}$ a root of the primitive polynomial $p(x) = 1 + x + x^3$. Then:

$$\alpha^3 = 1 + \alpha \to \mathcal{Z}_\alpha(1) = 3$$
$$\alpha^4 = \alpha + \alpha^2$$
$$\alpha^5 = \alpha^2 + \alpha^3 = 1 + \alpha + \alpha^2 = 1 + \alpha^4 \to \mathcal{Z}_\alpha(4) = 5$$
$$\alpha^6 = \alpha + \alpha^2 + \alpha^3 = 1 + \alpha^2 \to \mathcal{Z}_\alpha(2) = 6$$

Next we find the complete Zech logarithm table for $\mathbb{F}_{2^3}$:

x	1	2	3	4	5	6
$\mathcal{Z}_\alpha(x)$	3	6	1	5	4	2

■

Zech logarithms are discrete logarithms and they are, thus, hard to compute. Now, we are going to show some of their properties that can be used to easily compute them with trivial operations. These results can be found in [9].

Proposition 3: Given the Zech logarithm defined as in Definition 1. The following properties follow easily:

a) $\mathcal{Z}_\alpha(q - 1 - x) = \mathcal{Z}_\alpha(x) - x \bmod (q - 1)$

b) $\mathcal{Z}_\alpha(p\,x) = p\,\mathcal{Z}_\alpha(x) \bmod (q - 1)$

c) $\mathcal{Z}_\alpha(0) = -\infty$, for $p = 2$

d) $\mathcal{Z}_\alpha(\frac{q-1}{2}) = -\infty$, for $p \neq 2$

Remark 1: Notice that for fields of characteristic two we have that $\mathcal{Z}_\alpha(n) = m$ implies that $\mathcal{Z}_\alpha(m) = n$.

Consider now the notion of cyclotomic coset $\bmod(q - 1)$ given in [8].

Definition 2: Let $\mathbb{F}_{2^L}$ denote the Galois field of 2^L elements. An equivalence relation R is defined on its elements $\alpha, \beta \in \mathbb{F}_q$ such as follows: $\alpha\,R\,\beta$ if there exists an integer j, $0 \leq j \leq L - 1$, such that

$$2^j \cdot \alpha = \beta \bmod (2^L - 1).$$

The resultant equivalence classes into which $\mathbb{F}_{2^L}^*$ is partitioned are called the *cyclotomic cosets* modulo $2^L - 1$.

The leader element of every coset is the smallest integer in such an equivalence class.

According to [9], Zech logarithms map cosets onto cosets of the same size, that is, $\mathcal{Z}_\alpha : C_{s_1} \to C_{s_2}$, where $|C_{s_1}| = |C_{s_2}|$.

On the other hand, we consider the mapping $I : N_q \to N_q$, with $I(x) = q - 1 - x$, where $N_q = \{0, 1, 2, \ldots, q - 2\} \cup \{-\infty\}$. According to [9], $\mathcal{Z}_\alpha(I(x)) = \mathcal{Z}_\alpha(x) - x \bmod (q - 1)$ and

$$\mathcal{Z}_\alpha^{-1}(\mathcal{Z}_\alpha(x) - x) = I(x). \tag{4}$$

Like $\mathcal{Z}_\alpha(x)$, the application $I(x)$ also maps a cyclotomic coset onto another of the same size. If $\mathcal{Z}_\alpha(x)$ is known, then the value of $\mathcal{Z}_\alpha(I(x))$ can be computed. For fields with characteristic two, $\mathcal{Z}_\alpha^{-1}(x) = \mathcal{Z}_\alpha(x)$ and the computations are particularly simple.

Example 7: Consider the field $\mathbb{F}_{2^5}$, constructed with the primitive polynomial $p(x) = 1 + x^2 + x^5$. There are six cyclotomic cosets given by:

$$C_1 = \{1, 2, 4, 8, 16\} \qquad\qquad C_7 = \{7, 1, 14, 25, 19\}$$
$$C_3 = \{3, 6, 12, 24, 17\} \qquad\qquad C_{11} = \{11, 22, 13, 26\}$$
$$C_5 = \{5, 10, 20, 9, 18\} \qquad\qquad C_{15} = \{15, 30, 29, 27\}$$

Table 4: Zech logarithms for $\mathbb{F}_{2^5}$

x	$\mathcal{Z}_\alpha(x)$	x	$\mathcal{Z}_\alpha(x)$	x	$\mathcal{Z}_\alpha(x)$	x	$\mathcal{Z}_\alpha(x)$
0	$-\infty$	8	20	16	9	24	15
1	18	9	16	17	30	25	21
2	5	10	4	18	1	26	28
3	29	11	19	19	11	27	6
4	10	12	23	20	8	28	26
5	2	13	14	21	25	29	3
6	27	14	13	22	7	30	17
7	22	15	24	23	12		

From $p(x)$ we know that $\mathcal{Z}_\alpha(2) = 5$. Since we are working over a field with characteristic two and according to b) in Proposition 3, we can compute the logarithms of the other elements of C_2 and C_5:

$$\mathcal{Z}_\alpha(2) = 5 \to \mathcal{Z}_\alpha(5) = 2$$
$$\mathcal{Z}_\alpha(4) = 2\mathcal{Z}_\alpha(2) = 10 \to \mathcal{Z}_\alpha(10) = 4$$
$$\mathcal{Z}_\alpha(8) = 2\mathcal{Z}_\alpha(4) = 20 \to \mathcal{Z}_\alpha(20) = 8$$
$$\mathcal{Z}_\alpha(16) = 2\mathcal{Z}_\alpha(8) = 9 \to \mathcal{Z}_\alpha(9) = 16$$
$$\mathcal{Z}_\alpha(1) = 2\mathcal{Z}_\alpha(16) = 18 \to \mathcal{Z}_\alpha(18) = 1$$

Let us now use the map I to obtain the logarithms for elements in C_3 and C_{15}. According to the definition of this map, it is easy to compute $I(2) = 29$. Then, according to (4), we know that $\mathcal{Z}_\alpha(29) = 3$. Using the same method as above, we can compute every logarithm in C_3 and C_{15}.

Equally, we have $I(3) = 28$ and according to equation (4), $\mathcal{Z}_\alpha(28) = 26$. Therefore, we can compute every logarithm in cosets C_{11} and C_7.

Using these properties, we can compute every logarithm in $\mathbb{F}_{2^5}$, with very simple operations mod 31. The complete Zech logarithm table for $\mathbb{F}_{2^5}$ can be found in Table 4. ∎

Next, we find some minor results about Zech logarithms.

Proposition 4: If $\beta_1 = \mathcal{Z}_\alpha(\alpha_1 - \alpha_2) + \alpha_2$ and $\beta_2 = \mathcal{Z}_\alpha(\alpha_2 - \alpha_3) + \alpha_3$, then:

1. $\beta_1 = \mathcal{Z}_\alpha(\alpha_2 - \alpha_1) + \alpha_1$

2. $\mathcal{Z}_\alpha(\alpha_1 - \alpha_3) + \alpha_3 = \mathcal{Z}_\alpha(\beta_1 - \beta_2) + \beta_2$ (for fields of characteristic equal to two)

Proof 3:   1. According to the definition of Zech logarithm:

$\alpha^{\beta_1} = \alpha^{\mathcal{Z}_\alpha(\alpha_1-\alpha_2)+\alpha_2} = (1 + \alpha^{\alpha_1-\alpha_2})\alpha^{\alpha_2} = \alpha^{\alpha_1} + \alpha^{\alpha_2} = (1 + \alpha^{\alpha_2-\alpha_1})\alpha^{\alpha_1} = \alpha^{\mathcal{Z}_\alpha(\alpha_2-\alpha_1)+\alpha_1}$

2. Again, according to the definition of Zech logarithm:

$$\beta_1 = \mathcal{Z}_\alpha(\alpha_1 - \alpha_2) + \alpha_2 \to \alpha^{\beta_1} = \alpha^{\alpha_1} + \alpha^{\alpha_2} \tag{5}$$
$$\beta_2 = \mathcal{Z}_\alpha(\alpha_2 - \alpha_3) + \alpha_3 \to \alpha^{\beta_2} = \alpha^{\alpha_2} + \alpha^{\alpha_3} \tag{6}$$

Summing both equations (5) and (6), we get

$$\alpha^{\beta_1} + \alpha^{\beta_2} = \alpha^{\alpha_1} + \alpha^{\alpha_3} \to \mathcal{Z}_\alpha(\beta_1 - \beta_2) + \beta_2 = \mathcal{Z}_\alpha(\alpha_1 - \alpha_3) + \alpha_3$$

□

Remark 2:   1. The first part of Proposition 4 implies that

$$d_i^j = \mathcal{Z}_\alpha(d_i^{j-1} - d_{i+1}^{j-1}) + d_{i+1}^{j-1} = \mathcal{Z}_\alpha(d_{i+1}^{j-1} - d_i^{j-1}) + d_i^{j-1},$$

and, thus, we can change the order of the computations if needed.

Table 5: CA that generates the shrunken sequence in Example 2

102	102	102	102	102	102	102	102	102	102	102	102	102	102
1	1	0	1	0	0	1	0	0	1			1	1
0	1	1	1	0		1	0	1				0	
1	0	0	1			1	1	1	1	0	1	0	0
1	0	1				0		0	1	1	1	0	
1	1	1	1	0	1	0	0	1	0	0	1		
0		0	1	1	1	0		1	0	1			
0	0	1	0	0	1			1	1	1	1	0	1
0		1	0	1				0		0	1	1	1
		1	1	1	1	0	1	0	0	1	0	0	1
		0		0	1	1	1	0		1	0	1	
0	1	0	0	1	0	0	1			1	1	1	1
1	1	0		1	0	1				0		0	1
0	1			1	1	1	1	0	1	0	0	1	0
1				0		0	1	1	1	0		1	0

2. The second part of Proposition 4 implies that

$$d_i^j = \mathcal{Z}_\alpha(d_i^{j-1} - d_{i+1}^{j-1}) + d_{i+1}^{j-1} = \mathcal{Z}_\alpha(d_i^{j-2} - d_{i+2}^{j-2}) + d_{i+2}^{j-2},$$

and, thus, we can choose the best positions for our computations.

The algorithm we will propose in Section 5 makes use of the Zech logarithm to compute the positions given in (3). Using the properties we have seen in this section, we can reduce the complexity of the algorithm reducing the number of calculations.

5 Recovering the shrunken sequence

In this section, we use the CAs and their properties to recover the complete shrunken sequence. We know that the shrunken sequence and the companion sequences appear several times along the same CA, with shift $D = 2^{L_1-1}\mathcal{Z}_\alpha(1)$ (see Section 3.2).

In Example 2, we had a shrunken sequence of period 14. In Table 1, we saw that there exists a CA of length 14 that produces this sequence in its leftmost column. If we intercept the first 6 bits of the shrunken sequence, we can recover 21 elements in the CA (see the purple triangle in Table 5). According to the properties of this CA, these sequences are repeated along the CA and, thus, we can recover the same number of bits in other positions (see the other triangles in Table 5). In this case, the triangles of recovered bits overlap. Therefore, we can recover completely all the elements in the CA and, thus, recover the complete shrunken sequence.

In general, we need to intercept

$$N = 2^{L_1-1} + T - D = 2^{L_1-1}(2^{L_2} - \mathcal{Z}_\alpha(1))$$

bits of the shrunken sequence for the recovered triangles to overlap. This number depends completely on the value of $\mathcal{Z}_\alpha(1)$, which depends on the primitive polynomial $p(x)$ (characteristic polynomial of the interleaved m-sequences). This means that sometimes the number of needed intercepted bits will be greater than suitable for practical applications. For example, in Table 6 we can see the different values of $\mathcal{Z}_\alpha(1)$, for different primitive polynomials of degree 5. In order to make more difficult the recovery, we would use $p(x) = 1 + x + x^2 + x^3 + x^5$ since it produces the minimum value of $\mathcal{Z}_\alpha(1)$. In order to recover the sequence, we expect the cryptographer to use $q(x) = 1 + x^2 + x^3 + x^4 + x^5$, which produces the maximum value of $\mathcal{Z}_\alpha(1)$, and we have to intercept a smaller number bits to recover the complete sequence.

5.1 Cryptanalysis

In this section we introduce a cryptanalysis of the shrinking generator based on the results given in Section 3. This attack is based on an exhaustive search over the initial states of the first register R_1. Thus, the complexity of the brute-force attack trying all the possible keys is reduced by a factor 2^{L_2}.

Table 6: Values of $\mathcal{Z}_\alpha(1)$ for different primitive polynomials of degree 5

$p_2(x)$	$\mathcal{Z}_\alpha(1)$
$x^5 + x^2 + 1$	18
$x^5 + x^4 + x^2 + x + 1$	19
$x^5 + x^3 + x^2 + x + 1$	12
$x^5 + x^3 + 1$	14
$x^5 + x^4 + x^3 + x + 1$	13
$x^5 + x^4 + x^3 + x^2 + 1$	20

5.1.1 General idea

Given n bits, $\boldsymbol{s} = \{s_0, s_1, \ldots, s_{n-1}\}$, of the shrunken sequence, Algorithm 1 tests if a given initial state $\boldsymbol{a} = \{a_0, a_1, \ldots, a_{L_1-1}\}$ for the register R_1 is considered correct or not. The idea is to recover bits of the first interleaved m-sequence of the shrunken sequence. If two different bits are stored in the same position, the initial state $\boldsymbol{a}$ is incorrect. If $\boldsymbol{a}$ is considered correct, Algorithm 1 also returns the matrix A with the value and the positions of the recovered bits in the first interleaved m-sequence. Once we have recover a part of the first interleaved m-sequence, we can recover the complete shrunken sequence.

Algorithm 1. Crypto: Test an initial state for R_1

Input: $p_1(x)$, $p(x)$, δ, $\boldsymbol{s}$ and $\boldsymbol{a}$
function $[M, Stop]$ =**SubCrypto**$(p_1(x), p(x), \delta, \boldsymbol{s}, \boldsymbol{a})$

01:   Compute $\{a_i\}$ using $p_1(x)$ and $\boldsymbol{a}$ until finding $n = length(\boldsymbol{s})$ ones;
02:   Store in P the positions of the 1s in the generated bits of $\{a_i\}$;
03:   Store in P the new positions computed as $P_i \cdot d \bmod (2^{L_2} - 1)$;
04:   Store $[P_i, s_i]$ in a matrix M;
05:   $Stop = 1$;
06:   **while** $Stop = 1$ and $n > 1$
07:      Update P with the new positions;
08:      Update $\boldsymbol{s}$ with $\{s_0 + s_1, s_1 + s_2, \ldots, s_{n-2} + s_{n-1}\}$;
09:      Store [m,n]=size(M);
10:      **for** $j = 0$ to $m - 1$
11:         **for** $k = 0$ to $length(P) - 1$
12:            **if** $M_{j1} = P_k$
13:               **if** $M_{j2} \neq s_k$
14:                  Initialise M;
15:                  $Stop = 0$;
16:               **end if**
17:            **end if**
18:            Store $[P_k, s_k]$ in M;
19:         **end for**
20:      **end for**
21:   **end while**
end function
Output:
M: Recovered bits and their position in the first interleaved m-sequence.
$Stop$: 1 if the initial state is considered correct and 0 otherwise.

Remark 3: It is worth pointing out that this work can be equivalently done with rule 60, $x_i^{t+1} = x_{i-1}^t + x_i^t$. In this case, the sequences would appear in reverse order along the CA, but the results would be identical.

5.1.2 Numerical example

We are going to consider the advantageous case when $L_2 = L_1 + 1$. We know that $\delta = T_2 - 2$ (Corollary 2) and, furthermore, $p(x)$ is the reciprocal polynomial of $p_2(x)$ (see [3]). In this case, it is not necessary to compute δ, which prevent us from doing more calculations. We will also see that in this case, the computations of the logarithms on the first round are trivial.

Example 8: Consider two registers R_1 and R_2 with characteristic polynomials $p_1(x) = 1 + x + x^3 + x^4 + x^6$ and $p_2(x) = 1 + x^3 + x^7$, respectively.

Assume we intercept 10 bits of the shrunken sequence: $\boldsymbol{s} = \{1, 0, 0, 0, 0, 1, 0, 0, 0, 0\}$.

Notice that, in this case, the period of the sequence is $2^6(2^7 - 1) = 8128$.

We apply Algorithm 1 in order to check if the initial state $a = \{1, 1, 1, 0, 1, 1\}$ is correct for R_1.

Since $L_2 = L_1 + 1$, we know that $p(x)$ is the reciprocal polynomial of $p_2(x)$, that is, $p(x) = 1 + x^4 + x^7$.

In this case, we have that the distance of decimation is $\delta = T_2 - 2 = 2^7 - 3 = 125$ (Corollary 2).

Input: $p_1(x) = 1 + x + x^3 + x^4 + x^6$, $p(x) = 1 + x^4 + x^7$, $\delta = 125$, $\boldsymbol{a} = \{1\ 1\ 1\ 0\ 1\ 1\}$ and $\boldsymbol{s} = \{1, 0, 0, 0, 0, 1, 0, 0, 0, 0\}$.

We compute bits of the m-sequence generated by R_1 using $\boldsymbol{a}$ and store the positions of the ones until we find 10 ones: $pos = \{0, 1, 2, 4, 5, 6, 8, 10, 11, 13\}$.

The positions of the intercepted 10 bits in the first interleaved m-sequence are:

$$d_i^0 = pos = \{0, 125, 123, 119, 117, 115, 111, 107, 105, 101\}$$

.

Each bit of $\boldsymbol{s}$ is stored in the respective position of pos. We store this information in the matrix M and order by position:

$$M^T = \begin{bmatrix} 0 & 101 & 105 & 107 & 111 & 115 & 117 & 119 & 123 & 125 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We compute new positions and new bits to update the matrix M. The new bits are computed applying rule 102 to the elements of $\boldsymbol{s}$, $\{1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\}$ and they are stored, respectively, in the positions (mod 127):

$$d_0^1 = \mathcal{Z}_\alpha(125) = 65$$
$$d_1^1 = \mathcal{Z}_\alpha(125 - 123) + 123 = 2\mathcal{Z}_\alpha(1) + 123 = 63$$
$$d_2^1 = \mathcal{Z}_\alpha(123 - 119) + 119 = 4\mathcal{Z}_\alpha(1) + 119 = 126$$
$$d_3^1 = \mathcal{Z}_\alpha(119 - 117) + 117 = 2\mathcal{Z}_\alpha(1) + 117 = 57$$
$$d_4^1 = \mathcal{Z}_\alpha(117 - 115) + 115 = 2\mathcal{Z}_\alpha(1) + 115 = 55$$
$$d_5^1 = \mathcal{Z}_\alpha(115 - 111) + 111 = 4\mathcal{Z}_\alpha(1) + 111 = 118$$
$$d_6^1 = \mathcal{Z}_\alpha(111 - 107) + 107 = 4\mathcal{Z}_\alpha(1) + 107 = 114$$
$$d_7^1 = \mathcal{Z}_\alpha(107 - 105) + 105 = 2\mathcal{Z}_\alpha(1) + 105 = 45$$
$$d_8^1 = \mathcal{Z}_\alpha(105 - 101) + 101 = 4\mathcal{Z}_\alpha(1) + 101 = 108$$

Up to now, there are no repeated positions, then we continue with the next round. The new bits to be stored are given by $\{1, 0, 0, 1, 0, 1, 0, 0\}$ and the next positions are:

$$d_0^2 = \mathcal{Z}_\alpha(123) = 3$$
$$d_1^2 = \mathcal{Z}_\alpha(125 - 119) + 119 = 2\mathcal{Z}_\alpha(3) + 119 = 111$$
$$d_2^2 = \mathcal{Z}_\alpha(123 - 117) + 117 = 2\mathcal{Z}_\alpha(3) + 117 = 109$$
$$d_3^2 = \mathcal{Z}_\alpha(119 - 115) + 115 = 4\mathcal{Z}_\alpha(1) + 115 = 122$$
$$d_4^2 = \mathcal{Z}_\alpha(117 - 111) + 111 = 2\mathcal{Z}_\alpha(3) + 111 = 103$$
$$d_5^2 = \mathcal{Z}_\alpha(115 - 107) + 107 = 8\mathcal{Z}_\alpha(1) + 107 = 121$$
$$d_6^2 = \mathcal{Z}_\alpha(111 - 105) + 105 = 2\mathcal{Z}_\alpha(3) + 105 = 97$$
$$d_7^2 = \mathcal{Z}_\alpha(107 - 101) + 101 = 2\mathcal{Z}_\alpha(3) + 101 = 93$$

Position 111 appears again and we have to store 0 again. There is no contradiction.

In next round we have to store the bits $\{1\ 0\ 1\ 1\ 1\ 1\ 0\}$ in the positions d_i^3, $i = 1, \ldots, 6$. However, we know that it is easier to compute the positions d_i^4, $i = 1, \ldots, 5$ (see Remark 2) for the new bits $\{1\ 1\ 0\ 0\ 0\ 1\}$.

$$d_0^4 = \mathcal{Z}_\alpha(117) = 90$$
$$d_1^4 = \mathcal{Z}_\alpha(125 - 115) + 115 = 2\mathcal{Z}_\alpha(5) + 115 = 88$$
$$d_2^4 = \mathcal{Z}_\alpha(123 - 111) + 111 = 4\mathcal{Z}_\alpha(3) + 111 = 95$$
$$d_3^4 = \mathcal{Z}_\alpha(119 - 107) + 107 = 4\mathcal{Z}_\alpha(3) + 107 = 91$$
$$d_4^4 = \mathcal{Z}_\alpha(117 - 105) + 105 = 4\mathcal{Z}_\alpha(3) + 115 = 89$$
$$d_5^4 = \mathcal{Z}_\alpha(115 - 101) + 101 = 2\mathcal{Z}_\alpha(7) + 101 = 109$$

Position 109 appears again and we have to store a 1, but there is another bit in this position with value 0. We have a contradiction, so the guessed initial state is not correct.

Output: $STOP = 0$ The initial state $\boldsymbol{a} = \{1, 1, 0, 0, 0, 0, 0, 0\}$ is not correct since we found a contradiction. ■

In this case, we have just needed to compute the logarithms $\mathcal{Z}_\alpha(1)$, $\mathcal{Z}_\alpha(3)$, $\mathcal{Z}_\alpha(5)$, $\mathcal{Z}_\alpha(7)$, $\mathcal{Z}_\alpha(117)$, $\mathcal{Z}_\alpha(123)$, $\mathcal{Z}_\alpha(125)$. However, according to Section 4 we will see that we just need to compute two of these logarithms.

We start computing $\mathcal{Z}_\alpha(1) = 97$ and according to Remark 1, we have $\mathcal{Z}_\alpha(97) = 1$. It is easy to see that 7 and 97 are in the same cyclotomic coset, since $4 \cdot 97 = 7 \bmod 127$. Therefore, we have:

$$\mathcal{Z}_\alpha(7) = \mathcal{Z}_\alpha(4 \cdot 7) = 4\mathcal{Z}_\alpha(7) = 4$$

Now, we know that

$$\mathcal{Z}_\alpha(123) - 123 = \mathcal{Z}_\alpha(-123) = \mathcal{Z}_\alpha(4) = 4\mathcal{Z}_\alpha(1) = 7 \rightarrow \mathcal{Z}_\alpha(123) = 3$$

and then $\mathcal{Z}_\alpha(3) = 123$. It is aso possible to check that 123 and 125 are in the same cyclic coset, since $125 = 2^6 \cdot 123 \bmod 127$. Therefore, $\mathcal{Z}_\alpha(125) = 2^6 \mathcal{Z}_\alpha(123) = 65$.

Finally, we need to compute $\mathcal{Z}_\alpha(5) = 50$.

$$\mathcal{Z}_\alpha(117) - 117 = \mathcal{Z}_\alpha(-117) = \mathcal{Z}_\alpha(10) = 2\mathcal{Z}_\alpha(5) = 100 \rightarrow \mathcal{Z}_\alpha(117) = 90.$$

So computing $\mathcal{Z}_\alpha(1)$ and $\mathcal{Z}_\alpha(5)$ and due to the properties of Zech logarithm, we can deduce all the logarithms needed in this example. In general, the number of logarithms we need to compute in order to obtain the contradiction we need is smaller than it seems, due to the properties showed in Section 4.

5.2 Recovering the initial state of R_2

The idea is to use the recovered bits of the first interleaved m-sequence to recover the m-sequence produced by R_2. According to Proposition 1, if we know this m-sequence, we can recover the initial state of the m-sequence $\{b_i\}$ generated by R_2.

Example 9: We consider Example 8. Assume we apply Algorithm 1 with the correct initial state $\boldsymbol{a} = \{1, 0, 0, 0, 0, 0\}$. In this case, the algorithm returns STOP=1 and the matrix M with the recovered bits. In Appendix A, we find the returned matrix M and it is possible to see that we have recovered 46 bits and their positions in the first interleaved m-sequence of the shrunken sequence, which we denote now by $\{v_i\}$. We know that these sequence has as characteristic polynomial $p(x) = 1 + x^4 + x^7$ and the period is 127. Therefore, we know that $v_i + v_{i+4} + v_{i+7}$, for $i \geq 0$ and then:

$$v_{46} + v_{50} + v_{53} = 0 \longrightarrow v_{46} = 0$$
$$v_{47} + v_{51} + v_{54} = 0 \longrightarrow v_{51} = 1$$

We known 12 consecutive bits $\{v_{45}, v_{46}, \ldots, v_{56}\}$, of the first interleaved m-sequence $\{v_i\}$ and we know that the characteristic polynomial of the sequence has degree 7. Thus, we compute the whole m-sequence $\{v_i\}$. Finally according to Proposition 1, the initial state of R_2 will correspond to $\{b_0, b_{125}, b_{123}, b_{121}, , b_{119}, b_{117}, b_{115}\}$. ∎

5.3 Discussion

In this algorithm, we have performed an exhaustive search over 2^{L_1-1} initial states of R_1, which reduces the complexity of a brute-force attack by a factor of 2^{L_2}. In Table 7 some numerical results are depicted. We denote by $p_1(x)$ and $p_2(x)$ the characteristic polynomials R_1 and R_2, respectively, n is the number of intercepted bits, T represents the period of the corresponding shrunken sequence and N_{IS} denotes the number of R_1 initial states with no contradiction. From these results we can deduce that our algorithm presents two main advantages against other proposals.

First, compared with other probabilistic approaches (see, for example, [7]) the algorithm here presented is deterministic. This means that depending on the number of intercepted bits, the set of the possible correct states can have different sizes, but the correct one is certainly contained in such a set.

Second, although the higher the degrees of the characteristic polynomials are, the more intercepted bits we need, the required keystream length grows linearly in the length of R_2 while the period of the shrunken sequence grows exponentially. This means that the number of intercepted bits n needed for the attack is very low compared with the period of the shrunken sequence. This fact did not happen in other proposals like [11]. Low requirement of intercepted bits is a quite realistic condition for practical cryptanalysis.

The number of initial states for R_1 with no contradiction is very low compared with the number of initial states analysed, so it makes easier the checking of the true pair of initial states in both registers R_1 and R_2.

Finally, this algorithm is particularly adequate for parallelization. In fact, it is possible to divide the 2^{L_1-1} possible initial states into several groups and process each group of states separately.

The computation of Zech logarithms is the most time-consuming part of the algorithm. However, we have seen in Section 4 that many of the properties of this discrete logarithm can be used to reduce the calculations. For instance, in Example 8, the algorithm had to compute ten logarithms, but due to the properties of the Zech logarithm, computing only two logarithms the problem was solved.

Furthermore, at the end of Section 3.3, we saw that the computations of the positions d_k^m can be computed performing a simple addition $\mod (2_2^L - 1)$.

6 Conclusions

The shrinking generator was conceived and designed as a nonlinear keystream generator based on maximum-length LFSRs. However, this generator can be modelled in terms of 102-CAs. The effort to introduce decimation in order to break the inherent linearity of the LFSRs has been useless, since the shrunken sequence can be modelled as the output sequence of a model based on linear CAs. In this work,

Table 7: Some numerical results for the algorithm

$p_1(x)$	$p_2(x)$	n	T	N_{IS}
$1 + x^2 + x^3$	$1 + x^3 + x^4$	8	60	1
$1 + x^2 + x^3$	$1 + x^3 + x^5$	9	124	1
$1 + x^2 + x^5$	$1 + x + x^6$	11	1008	1
$1 + x^3 + x^5$	$1 + x + x^7$	13	2032	1
$1 + x^2 + x^5$	$1 + x^3 + x^7$	14	2032	1
$1 + x + x^6$	$1 + x^3 + x^7$	16	4046	1
$1 + x + x^7$	$1 + x^2 + x^3 + x^4 + x^8$	16	16320	1
$1 + x + x^7$	$1 + x^4 + x^9$	16	32704	1
$1 + x^2 + x^3 + x^4 + x^8$	$1 + x^4 + x^9$	17	65408	1
$1 + x^4 + x^9$	$1 + x^3 + x^{10}$	18	261888	1
$1 + x^4 + x^9$	$1 + x^2 + x^5 + x^9 + x^{10}$	19	261888	1
$1 + x^2 + x^{11}$	$1 + x + x^5 + x^8 + x^{12}$	27	4193280	3
$1 + x^9 + x^{10} + x^{12} + x^{13}$	$1 + x + x^2 + x^5 + x^6 + x^{13} + x^{14}$	30	67104768	3
$1 + x^9 + x^{10} + x^{12}x^{13}$	$1 + x + x^4 + x^{15} + x^{16}$	52	268431360	1
$1 + x + x^2 + x^5 + x^6 + x^{13} + x^{14}$	$1 + x^2 + x^5 + x^{14} + x^{15}$	40	268427264	126
$1 + x^2 + x^5 + x^{14} + x^{15}$	$1 + x + x^4 + x^6 + x^{16}$	50	1073725440	29
$1 + x + x^4 + x^{15} + x^{16}$	$1 + x + x^2 + x^6 + x^{10} + x^{11} + x^{17}$	58	4294934528	206

we analyse a family of one-dimensional, linear, regular and cyclic 102-CA that describe the behaviour of the shrinking generator. These CAs generate a family of sequences with the same characteristic polynomial and the same period as the shrunken sequence. Taking advantage of the linearity and the similarity between the sequences generated by these CAs, we propose a cryptanalysis based on the exhaustive search among the initial states of the first register.

A natural extension of this work is the generalization of this procedure to other cryptographic sequence generators: (a) All the family of the decimation-based keystream generators (the self-shrinking generator, the generalized self-shrinking generator or the modified self-shrinking generator[2, 4, 5]). (b) The so-called interleaved m-sequences, as they present very similar structural properties to those of the sequences obtained from irregular decimation generators.

References

[1] Lejla Batina, Joseph Lano, Nele Mentens, Sıddıka Berna Örs, Bart Preneel, and Ingrid Verbauwhede. Energy, performance, area versus security trade-offs for stream ciphers. In *In The State of the Art of Stream Ciphers, Workshop Record (2004), ECRYPT*, pages 302–310, 2004.

[2] Sara D. Cardell and Amparo Fúster-Sabater. Linear models for the self-shrinking generator based on CA. *Journal of Cellular Automata*, 11(2-3):195–211, 2016.

[3] Sara D. Cardell and Amparo Fúster-Sabater. Modelling the shrinking generator in terms of linear CA. *Advances in Mathematics of Communications*, 10(4):797–809, 2016.

[4] Sara D. Cardell and Amparo Fúster-Sabater. Recovering the MSS-sequence via CA. *Procedia Computer Science*, 80:599–606, 2016.

[5] Sara D. Cardell and Amparo Fúster-Sabater. Discrete linear models for the generalized self-shrunken sequences. *Finite Fields and Their Applications*, 47:222–241, 2017.

[6] Don Coppersmith, Hugo Krawczyk, and Yishay Mansour. The shrinking generator. In *Advances in Cryptology – CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 23–39. Springer-Verlag, 1993.

[7] Jovan Dj. Golić. Correlation analysis of the shrinking generator. In *Advances in Cryptology-Crypto'2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 440–457. Springer-Verlag, Berlin, 2001.

[8] Solomon W. Golomb. *Shift Register-Sequences*. Aegean Park Press, Laguna Hill, California, 1982.

[9] Klaus Huber. Some comments on Zech's logarithms. *IEEE Transactions on Information Theory*, 36(4):946–950, 1990.

[10] C. G. J. Jacobi. Über die kreisteilung und ihre anwendung auf die zahlentheorie. *Journal für die Reine und Angewandte Mathematik*, 30:166–182, 1846.

[11] Thomas Johansson. Reduced complexity correlation attacks on two clock-controlled generators. In *Advances in Cryptology – ASIACRYPT'98*, volume 1514 of *Lecture Notes in Computer Science*, pages 342–357. Springer-Verlag, Berlin, 1998.

[12] James L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 15(1):122–127, 1969.

[13] John Von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA, 1966.

[14] Christof Paar and Jan Pelzl. *Understanding Cryptography*. Springer, Berlin, 2010.

[15] Alberto Peinado and Amparo Fúster-Sabater. Generation of pseudorandom binary sequences by means of LFSRs with dynamic feedback. *Mathematical and Computer Modelling*, 57(11–12):2596–2604, 2013.

[16] Matthew Robshaw and Olivier Billiet, editors. *New Stream Cipher Designs: The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2008.

A Tables

$$
M = \begin{bmatrix}
0 & 1 \\
12 & 1 \\
23 & 0 \\
24 & 1 \\
25 & 1 \\
32 & 0 \\
38 & 1 \\
40 & 0 \\
41 & 0 \\
43 & 1 \\
45 & 1 \\
47 & 0 \\
48 & 1 \\
49 & 0 \\
50 & 1 \\
52 & 1 \\
53 & 1 \\
54 & 1 \\
55 & 0 \\
56 & 1 \\
59 & 0 \\
60 & 1 \\
64 & 0 \\
66 & 0 \\
68 & 1 \\
78 & 1 \\
79 & 0 \\
87 & 0 \\
91 & 0 \\
95 & 0 \\
98 & 0 \\
99 & 1 \\
101 & 0 \\
103 & 0 \\
105 & 1 \\
107 & 0 \\
108 & 1 \\
109 & 0 \\
110 & 0 \\
111 & 0 \\
112 & 1 \\
114 & 0 \\
115 & 0 \\
117 & 0 \\
118 & 0 \\
119 & 0
\end{bmatrix}
$$