RACSAM (2020) 114:79
https://doi.org/10.1007/s13398-020-00807-5

ORIGINAL PAPER

Linear complexity of generalized sequences by comparison of PN-sequences

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

Received: 4 October 2018 / Accepted: 13 January 2020

Abstract
Linear complexity is a much used metric of the security of any binary sequence with application in communication systems and cryptography. In this work, we propose a method of computing the linear complexity of a popular family of cryptographic sequences, the so-called generalized sequences. Such a family is generated by means of the irregular decimation of a single Pseudo Noise sequence (PN-sequence). The computation method is based on the comparison of the PN-sequence with shifted versions of itself. The concept of linear recurrence relationship and the rows of the Sierpinski triangle play a leading part in this computation.

Keywords Decimated sequence · Linear complexity · Generalized generator · Sierpinski triangle · Recurrence relationship

Mathematics Subject Classification 94A55

1 Introduction

The idea of randomness in finite sequences reflects the difficulty of predicting next digits of a sequence from the previous ones. A measure of the unpredictability of a sequence is its linear complexity (LC). Roughly speaking, LC is related with the amount of sequence

An earlier version of this paper was presented at the Conference "Linear Algebra, Matrix Analysis and Applications. ALAMA2018", held in Sant Joan d'Alacant on May/June 2018.

This research has been partially supported by Ministerio de Economía, Industria y Competitividad (MINECO), Agencia Estatal de Investigación (AEI), and Fondo Europeo de Desarrollo Regional (FEDER, UE) under project COPCIS, reference TIN2017-84844-C2-1-R, and by Comunidad de Madrid (Spain) under project CYNAMON, reference P2018/TCS-4566, also co-funded by European Union FEDER funds. Sara D. Cardell was supported by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), Brazil, and Sao Paulo Research Foundation (FAPESP), grant 2013/25977-7.

✉ Sara D. Cardell
sdcardell@ime.unicamp.br

Amparo Fúster-Sabater
amparo@iec.csic.es

[1] Instituto de Tecnologías Físicas y de la Información, CSIC, Madrid, Spain

[2] Instituto de Matemática, Estatística e Computação Científica, University of Campinas, Campinas, Brazil

needed to recover the whole sequence. In terms of security, this amount must be as large as possible: the recommended value is approximately half the period T of such a sequence, that is $LC \simeq T/2$. Traditionally, the LC of a sequence is computed by the Berlekamp–Massey algorithm [17] after processing at least $2 \cdot LC$ of its digits. Nevertheless, for sequences with periods in a cryptographic range ($T \simeq 2^{128}$) the application of this algorithm could be an extremely hard task.

In the literature, there exist different families of pseudorandom binary sequences, e.g. Gold-sequence family, Kasami (small and large set) sequence families, GMW sequences and generalized GMW sequences, Klapper sequences, No sequences, cascaded sequences, multiplexed sequences or some classes of irregular decimated sequences (see [13,19,20] and the references cited therein) with the following property: every one of these sequences is obtained interleaving shifted versions of a single PN-sequence or output sequence of a maximum-length Linear Feedback Shift Register (LFSR), see [13, Definition 1] and Sect. 2.1 for details on interleaved sequences. In brief, a large number of well-known sequences satisfies such a property.

In this work, we analyse the LC of the sequences obtained from a class of irregular decimation-based generators. The underlying idea of this type of generators is the irregular decimation of a PN-sequence according to the bits of another one. Inside the class of generators based on self-decimation, we can enumerate: (a) the self-shrinking generator [18] where a single PN-sequence decimates itself, (b) the modified self-shrinking generator [15] a new variant of the self-shrinking generator that uses a selection rule based on the XORed value of a pair of bits, (c) the t-modified self-shrinking generator [5] a generalization of the previous generator where the PN-sequence is divided into groups of t bits before applying the selection rule and (d) the generalized self-shrinking generator [14] a family of sequence generators that includes among its members the generators listed in (a), (b) and (c), see subsection (2.3). Thus, the generalized self-shrinking generator is the most general and representative element in the class of self-decimated generators. Nevertheless, in reference [14] where such a generator is introduced, no mention to the parameter LC of the generalized sequences or sequences produced by this generator can be found. Keeping in mind this fact, we introduce a method of computing the LC of the generalized sequences. Such a method is based exclusively on the comparison of the bits of shifted versions of one single PN-sequence that constitutes the fundamental structure of the previous sequences.

This method of computing the linear complexity of generalized sequences was first introduced in [9]. Nevertheless, a more complete and detailed version of such a method is here presented. In the last section of this work, we introduce: (a) a discussion of the proposed method, (b) a comparison between the requirements of the Berlekamp–Massey algorithm and those of the algorithm here developed and (c) an extension of such a method to other types of self-decimated generators. The previously enumerated items constitute the novelty of this paper.

The work is organised as follows: In Sect. 2, we introduce the basic concepts and results necessary to understand the rest of the paper. This section includes an explicit description of the concepts: LFSR, linear complexity, generalized sequences and t-modified sequences. In Sect. 3, we describe a method of computing the linear complexity of the generalized sequences. Discussion of such a method as well as its application to self-decimated generators are introduced in Sect. 4. Finally, conclusions in Sect. 5 end the paper.

2 Preliminaires

In this section, we recall basic concepts and results that will be useful throughout this paper.

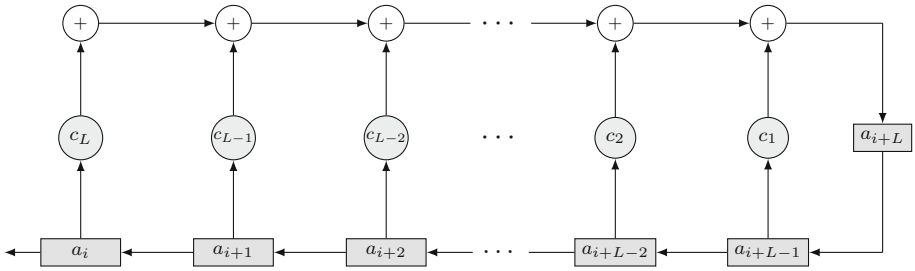


Fig. 1 LFSR of length L

2.1 Binary sequences

Since the pseudorandom sequences used in secure communications and cryptography are binary sequences, this work focusses on the binary field of two elements, denoted by $\mathbb{F}_2$. In fact, let $\{a_i\}$ $(i = 0, 1, 2, \dots)$ be a sequence defined over $\mathbb{F}_2$, that is $a_i \in \mathbb{F}_2$ for all $i \geq 0$. The decimation of the sequence $\{a_i\}$ by distance d is a new sequence $\{b_i\}$ $(i = 0, 1, 2, \dots)$ obtained by taking every dth term of $\{a_i\}$, that is $\{b_i\} = \{a_{d \cdot i}\}$, see [7].

Let L be a positive integer, and let $c_0, c_1, \dots, c_{L-1}$ be given elements of the binary field $\mathbb{F}_2$. A sequence $\{a_i\}$ satisfying the relation

$$a_{i+L} = c_1 a_{i+L-1} + c_2 a_{i+L-2} + \cdots + c_{L-1} a_{i+1} + c_L a_i \qquad i \geq 0 \tag{1}$$

is called an Lth order linear recurring sequence in $\mathbb{F}_2$. The equality given in (1) is an Lth order linear recurrence relationship (l.r.r.). The polynomial of degree L

$$p(x) = x^L + c_1 x^{L-1} + c_2 x^{L-2} + \cdots + c_{L-1} x + c_L \in \mathbb{F}_2[x] \tag{2}$$

is the characteristic polynomial of the linear recurrence relationship and the sequence $\{a_i\}$ is said to be generated by $p(x)$.

The generation of linear recurring sequences can be implemented on Linear Feedback Shift Registers (LFSRs) [12, Chapter II]. These devices handle information in form of bits and they are based on shifts and linear feedback (see Fig. 1). An LFSR consists of L interconnected stages (LFSR length) of binary content, the characteristic polynomial $p(x)$ of its linear recurrence relationship and the non-zero initial state (stage contents at the initial instant). If $p(x)$ is a primitive polynomial [16, Chapter 3], then the register is said to be a maximum-length LFSR and its output sequence $\{a_i\}$ is called a PN-sequence of period $T = 2^L - 1$ with 2^{L-1} ones and $(2^{L-1} - 1)$ zeros. Therefore, a PN-sequence is a sequence generated by a linear recurrence relationship of the form given in (1) whose characteristic polynomial given in (2) is primitive. The structural properties of the PN-sequences are extensively analysed in [12, Chapter III].

The linear complexity of a sequence is the length of the shortest LFSR that generates such a sequence or, equivalently, the order of the lowest linear recurrence relationship able to generate it. Clearly, the linear complexity of a PN-sequence is L, the degree of its corresponding characteristic polynomial, as given just L of its terms $(a_i, a_{i+1}, \dots, a_{i+L-1})$ the remaining terms are uniquely determined.

If α is a root of $p(x)$, then α is a primitive element in $\mathbb{F}_{2^L}$ the extension field of $\mathbb{F}_2$ [16]. Via the characteristic polynomial, there is a natural one-to-one correspondence

$$a_i \leftrightarrow \alpha^i \qquad (i = 0, 1, 2, \dots, 2^L - 2), \tag{3}$$

Table 1 Interleaved sequence $\mathbf{w} = \{w_i\}$ generated by interleaving 4 shifted versions of a single PN-sequence

$\mathbf{w}_0$	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$
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

between the *ith* term of the PN-sequence, notated a_i, and the *ith* power of α, notated α^i.

The trace map is the function $Tr : \mathbb{F}_{2^L} \to \mathbb{F}_2$ defined by

$$Tr(x) = \sum_{j=0}^{L-1} x^{2^j} \qquad \forall x \in \mathbb{F}_{2^L}.$$

The trace map provides an adequate way of relating terms of the PN-sequence with powers of α [16, Theorem 7.47]. In fact, if $\{a_i\}$ $(i = 0, 1, 2, \dots)$ is the PN-sequence generated by a maximum-length LFSR, then there exists a non-zero element $c \in \mathbb{F}_{2^L}$ such that

$$a_i = Tr(c\,\alpha^i) \qquad (i = 0, 1, 2, \dots).$$

Finally, the concept of interleaved sequence is defined as follows [13, Definition 1]:

Definition 1 Let $p(x) \in \mathbb{F}_2[x]$ be a polynomial of degree r defined over $\mathbb{F}_2[x]$ and let m be a positive integer. For any sequence $\mathbf{w} = \{w_i\}$ over $\mathbb{F}_2$, write $k = i \cdot m + j$ with $(i = 0, 1, 2, \dots)$ and $(j = 0, 1, \dots, m-1)$. If the subsequence $\mathbf{w}_j = \{w_{i \cdot m + j}\}$ $(i = 0, 1, 2, \dots)$ is generated by $p(x)$ for all j, then the sequence $\mathbf{w}$ is called an interleaved sequence over $\mathbb{F}_2$ of size m associated with $p(x)$. □

We can write $\mathbf{w} = \{\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{m-1}\}$ where each $\mathbf{w}_j$ $(j = 0, 1, \dots, m-1)$ is an m-decimation of the sequence $\mathbf{w}$ by taking one out of m terms. If the polynomial $p(x)$ is primitive, then the sequence $\mathbf{w}$ is a *primitive interleaved sequence* and all subsequence $\mathbf{w}_j$ is the single PN-sequence generated by $p(x)$. Table 1 shows the interleaved sequence $\mathbf{w}$ over $\mathbb{F}_2$ with size $m = 4$ and associated with the primitive polynomial $p(x) = x^3 + x + 1$ of degree $r = 3$. Its period is $T_w = 28$. By rows, the interleaved sequence is $\mathbf{w} = \{1, 1, 1, 1, 1, 0, 1, 0, 0, 0, \dots, 1, 1, 0, 0\}$. By columns, every subsequence $\mathbf{w}_j$ is a shifted version of the PN-sequence generated by $p(x)$.

2.2 The family of generalized sequences

The generalized self-shrinking generator (or simply generalized generator) was introduced in [14] as a more simple version of the shrinking generator [6], since the generalized generator involves a single LFSR. This generator produces a family of sequences, the so-called generalized sequences, that can be described as follows:

Definition 2 Let $\{a_i\}$ $(i = 0, 1, 2, \dots)$ be a PN-sequence generated by a maximum-length LFSR with an L-degree primitive characteristic polynomial. Let p be an integer and $\{v_i\}$ $(i =$

0, 1, 2, . . .) be a p-position left shifted version of $\{a_i\}$ with ($p = 0, 1, 2, \ldots, 2^L - 2$). The decimation rule is very simple:

1. If $a_i = 1$, then v_i is output.
2. If $a_i = 0$, then v_i is discarded and there is no output bit.

Thus, for each p an output sequence $\{s_0\, s_1\, s_2 \ldots\}_p$ denoted by $\{s_k\}_p$ ($k \geq 0$) is generated. Such a sequence is called the generalized sequence associated with the shift p. □

Recall that $\{a_i\}$ remains fixed while $\{v_i\}$ is the sliding sequence or left-shifted version of $\{a_i\}$. When p ranges in the interval $p \in [0, 1, 2, \ldots, 2^L - 2]$, then the family of $2^L - 1$ generalized sequences is obtained. Such a family, plus the identically null sequence, has structure of Abelian group whose group operation is the bit-wise XOR (addition mod 2), the neutral element is the identically null sequence and the inverse element of each sequence is the sequence itself [14, Theorem 2].

Since 2^{L-1} is the number of ones in the PN-sequence $\{a_i\}$, the period of every generalized sequence will be a divisor of 2^{L-1}. This family always includes [11] the sequence $\{111111\ldots\}$ for $p = 0$ and the sequences $\{101010\ldots\}$ and $\{010101\ldots\}$ for $p = q, q + 1$, where q is an integer corresponding to the power $\alpha^q \in \mathbb{F}_{2^L}$ satisfying $\alpha^{q+1} = \alpha^q + 1$. All the sequences in this family are balanced (equal number of ones and zeros) except for the identically one and null sequences [14, Theorem 1].

Finally, let $\{u_i\}$ ($i = 0, 1, \ldots, 2^{L-1} - 1$) be a sequence of period $T = 2^{L-1}$ whose terms u_i are elements of $\mathbb{F}_{2^L}$. Keeping in mind the one-to-one correspondence defined in (3), the terms of $\{u_i\}$ are the powers of α associated with the ones of $\{a_i\}$. In fact,

$$u_i = \alpha^{\tau(i)}, \tag{4}$$

where $\tau(i)$ with $0 \leq \tau(i) \leq 2^L - 2$ denotes the position of the $(i+1)$th one in the PN-sequence $\{a_i\}$. Next, an upper bound on the LC of $\{u_i\}$ is given.

Theorem 1 ([1], Theorem 6) *The linear complexity of the sequence $\{u_i\}$ defined in (4) is upper bounded by*

$$LC(\{u_i\}) \leq 2^{L-1} - (L - 2).$$

The previous theorem will allow us to obtain an upper bound on the LC of the generalized sequences.

Now two different sequence generators related with the generalized generator are introduced. The self-shrinking generator [18] is a more simplified version of the shrinking generator [6], where the PN-sequence $\{a_i\} = \{a_0, a_1, \ldots\}$ generated by a maximum-length LFSR is self-decimated. In this case, consecutive pairs of bits are considered. If a pair happens to take the value 10 or 11, then it produces the bit 0 or 1, respectively. On the other hand, if a pair happens to be 01 or 00, then this pair is discarded. More formally speaking, given two consecutive bits $\{a_{2i}, a_{2i+1}\}$, $i = 0, 1, 2, \ldots$, the output sequence $\{s_j\} = \{s_0, s_1, \ldots\}$ is computed as:

$$\begin{cases} \text{If } a_{2i} = 1 \text{ then } s_j = a_{2i+1}, \\ \text{If } a_{2i} = 0 \text{ then } a_{2i+1} \text{ is discarded.} \end{cases}$$

The generated sequence is called the self-shrunken sequence.

On the other hand, the modified self-shrinking generator [15] considers groups of three bits of a PN-sequence: given three consecutive bits $\{a_{3i}, a_{3i+1}, a_{3i+2}\}_{i \geq 0}$, the output sequence $\{s_j\} = \{s_0, s_1, \ldots\}$ is computed as:

$$\begin{cases} \text{If } a_{3i} + a_{3i+1} = 1 \text{ then } s_j = a_{3i+2}, \\ \text{If } a_{3i} + a_{3i+1} = 0 \text{ then } a_{3i+2} \text{ is discarded.} \end{cases}$$

The output sequence $\{s_j\}$ is known as the modified self-shrunken sequence.

Both the self-shrunken sequence and the modified-self shrunken sequence are also generalized sequences [3,21]

2.3 The family of *t*-modified sequences

In this subsection, a family of sequences closely related with the generalized sequences is defined. The t-modified self-shrinking generator was introduced in [5] as a generalization of the self-shrinking generator [18] and the modified self-shrinking generator [15]. This generator produces a family of sequences, the so-called t-modified sequences, that can be described as follows.

Definition 3 Let $\{a_i\}$ $(i = 0, 1, 2, \dots)$ be a PN-sequence generated by a maximum-length LFSR with an L-degree primitive characteristic polynomial. From $\{a_i\}$ a t-modified self-shrinking generator $(t = 2, 3, \dots, 2^L - 2)$ with a very simple decimation rule is constructed. In fact, if we divide the PN-sequence into groups of t bits $\{a_{t\cdot i}, a_{t\cdot i+1}, a_{t\cdot i+2}, \dots, a_{t\cdot i+(t-1)}\}$, then the t-modified sequence $\{s_k\}$ $(k = 0, 1, 2, \dots)$ is computed as follows:

$$\begin{cases} \text{If } \sum_{j=0}^{t-2} a_{t\cdot i+j} = 1, \text{ then } s_k = a_{t\cdot i+(t-1)}. \\ \text{If } \sum_{j=0}^{t-2} a_{t\cdot i+j} = 0, \text{ then } a_{t\cdot i+(t-1)} \text{ is discarded.} \end{cases}$$

□

According to this decimation rule, the t-modified sequence is obtained from t decimated sequences $\{Seq(j)\} = \{a_{t\cdot i+j}\}$ $(j = 0, 1, \dots, t-1)$ of the original PN-sequence $\{a_i\}$ with decimation distance t. That is:

$$\begin{array}{rccccccc} \{Seq(0)\}: & a_0 & a_t & a_{2t} & a_{3t} & a_{4t} & a_{5t} & \dots \\ \{Seq(1)\}: & a_1 & a_{t+1} & a_{2t+1} & a_{3t+1} & a_{4t+1} & a_{5t+1} & \dots \\ \{Seq(2)\}: & a_2 & a_{t+2} & a_{2t+2} & a_{3t+2} & a_{4t+2} & a_{5t+2} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \{Seq(t-2)\}: & a_{t-2} & a_{t+t-2} & a_{2t+t-2} & a_{3t+t-2} & a_{4t+t-2} & a_{5t+t-2} & \dots \\ \{Seq(t-1)\}: & a_{t-1} & a_{t+t-1} & a_{2t+t-1} & a_{3t+t-1} & a_{4t+t-1} & a_{5t+t-1} & \dots \end{array}$$

If the decimation distance t satisfies the condition

$$\text{g.c.d.}(t, 2^L - 1) = 1, \tag{5}$$

then all the decimated sequences $\{Seq(j)\}$ $(j = 0, 1, \dots, t-1)$ are the same PN-sequence whose characteristic polynomial is given by

$$p(x) = \left(x + \alpha^t\right)\left(x + \alpha^{2t}\right)\left(x + \alpha^{4t}\right)\cdots(x + \alpha^{t\cdot 2^{L-1}}),$$

where $\alpha \in \mathbb{F}_{2^L}$ is a root of the characteristic polynomial $p(x)$ of the PN-sequence $\{a_i\}$. See [12, Chapter III] for more details about decimation of PN-sequences.

Recall that in order to construct the t-modified sequence, we perform the bit-wise XOR of the first $(t-1)$ sequences $\{Seq(j)\}$ giving rise to the same decimated sequence but starting at a different point, that is:

$$\begin{array}{rccccccc}
\{Seq(0)\}: & a_0 & a_t & a_{2t} & a_{3t} & a_{4t} & a_{5t} & \cdots \\
\{Seq(1)\}: & a_1 & a_{t+1} & a_{2t+1} & a_{3t+1} & a_{4t+1} & a_{5t+1} & \cdots \\
\{Seq(2)\}: & a_2 & a_{t+2} & a_{2t+2} & a_{3t+2} & a_{4t+2} & a_{5t+2} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
\oplus\{Seq(t-2)\}: & a_{t-2} & a_{t+t-2} & a_{2t+t-2} & a_{3t+t-2} & a_{4t+t-2} & a_{5t+t-2} & \cdots \\
\hline
\{a_{d+t\cdot i}\}: & a_d & a_{d+t} & a_{d+2t} & a_{d+3t} & a_{d+4t} & a_{d+5t} & \cdots
\end{array}$$

where d is an integer $0 \leq d \leq 2^L - 2$. Next, to get the t-modified sequence we simply apply the decimation rule given in Definition 2 to the sequences $\{a_{d+t\cdot i}\}$ and $\{Seq(t-1)\}$.

Recall that, as both sequences are the same PN-sequence but starting at different terms, what we are generating via the decimation rule is just a generalized sequence. Consequently, the t-modified sequence is a generalized sequence. Moreover, if $t = 2$, then the 2-modified sequence is the self-shrunken sequence defined in [18] as well as if $t = 3$, then the 3-modified sequence is the modified self-shrunken sequence defined in [15]. In general, for any integer t satisfying the equation (5), the corresponding t-modified sequence is a generalized sequence [5].

In brief, if the Eq. (5) holds, then the self-shrinking generator, the modified self-shrinking generator and the t-modified self-shrinking generator produce generalized sequences [3,5, 21]. Thus, the LC of the sequences produced by the previous generators can be analysed as that of the generalized sequences.

3 Linear complexity of the generalized sequences

In this section, an upper bound on the linear complexity of the generalized sequences is computed. Next and based on this result, we introduce a method of computing such a linear complexity as well as an illustrative example.

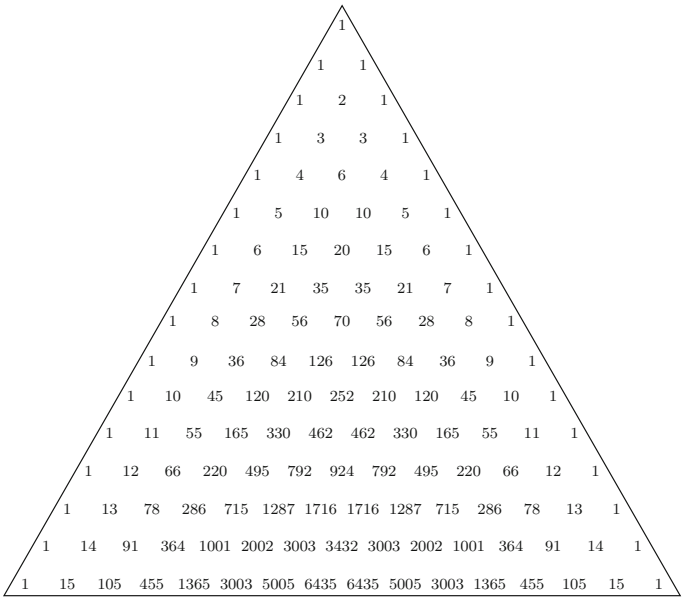
3.1 An upper bound on the linear complexity

Since the period of every generalized sequence is a power of 2, then its minimal polynomial (lowest degree characteristic polynomial) is of the form $(x+1)^{LC}$ [8, Section 5], [10, Theorem 1]. Thus, $(x+1)^N$ with $LC < N$ are characteristic polynomials of higher degree defining linear recurrence relationships that the generalized sequence has to satisfy [9]. Contrarily, $(x+1)^N$ with $N < LC$ are not characteristic polynomials meaning that the generalized sequence does not satisfy their corresponding linear recurrence relationships. This is the key idea to compute the LC in the class of generalized sequences.

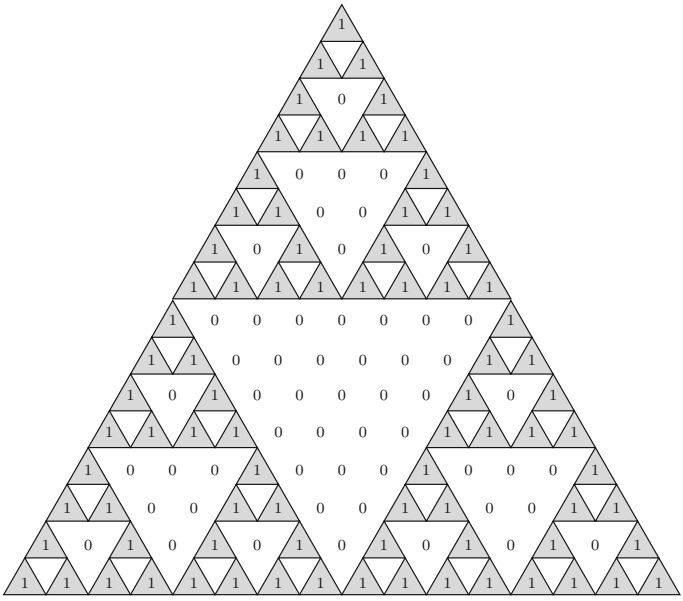
The coefficients of the polynomial $(x+1)^N$ are the binomial terms $\binom{N}{i}$ $(i = 0, 1, \ldots, N)$ of the *Nth* row of the Pascal's triangle (see Fig. 2a) [2,4]. When such a triangle is reduced mod 2, then we get the Sierpinski's triangle (see Fig. 2b). Now an upper bound on the LC of the generalized sequences is given by the following theorem.

Theorem 2 *The linear complexity of every generalized sequence is upper bounded by*

$$LC(\{s_k\}_p) \leq 2^{L-1} - (L-2) \qquad (p = 0, 1, 2, \ldots, 2^L - 2).$$



(a) Pascal's triangle



(b) Sierpinski's triangle

Fig. 2 Pascal and Sierpinski triangles

Proof According to Theorem 1, the sequence $\{u_i\}$ satisfies the l.r.r.

$$\sum_{i=0}^{M_0} c_i\ u_{i+j} = 0 \qquad (j = 0, 1, \ldots, 2^{L-1} - 1), \tag{6}$$

where the c_i are the binary coefficients of the M_0th row in the the Sierpinski's triangle (numbered 0, 1, 2, … from top to bottom), $M_0 = 2^{L-1} - (L-2)$ and the sum $(i + j)$ is taken mod 2^{L-1}.

Multiplying Eq. (6) by $c\ \alpha^p$ where $c \in \mathbb{F}_{2^L}$ and $c \neq 0$, we get

$$c\ \alpha^p \sum_{i=0}^{M_0} c_i\ u_{i+j} = \sum_{i=0}^{M_0} (c_i\ c\ \alpha^p\ u_{i+j}) = 0 \qquad (p = 0, 1, 2, \ldots, 2^L - 2).$$

Thus,

$$Tr\left(\sum_{i=0}^{M_0} c_i\ c\ \alpha^p\ u_{i+j}\right) = \sum_{i=0}^{M_0} c_i\ Tr\left(c\ \alpha^{\tau(i+j)+p}\right) = \sum_{i=0}^{M_0} c_i\ a_{\tau(i+j)+p} = Tr(0) = 0, \tag{7}$$

where $\{a_{\tau(i+j)+p}\}$ denotes the generalized sequence associated with the shift p. Therefore, according to Eq. (7), all the generalized sequences satisfy the l.r.r. given in (6) and their linear complexities are upper bounded by $M_0 = 2^{L-1} - (L-2)$. □

3.2 A method of computing the linear complexity of the generalized sequences

Based on the upper bound provided by Theorem 2, the practical computation of the LC for the class of generalized sequences is performed. The main idea is to test decreasing values, M_i, of LC starting at the upper bound $M_0 = 2^{L-1} - (L-2)$, that is $M_0 > M_1 > M_2 > M_3 > M_4 > \cdots$ and to check for each M_i the l.r.r. of the different generalized sequences.

For $M_1 < M_0$ and taking the coefficients c_i of the M_1th row in the Sierpinski's triangle, we compute:

$$\sum_{i=0}^{M_1} c_i\ u_{i+j} = \alpha^n \neq 0 \qquad (j = 0, 1, \ldots, 2^{L-1} - 1),$$

n being a non-negative integer. Therefore, proceeding in the same way as before we get:

$$\sum_{i=0}^{M_1} c_i\ \alpha^{\tau(i+j)+p} = \alpha^{n+p} \quad \Rightarrow \quad \sum_{i=0}^{M_1} c_i\ a_{\tau(i+j)+p} = a_{n+p}. \tag{8}$$

Thus, $\{a_{n+p}\}\,(p = 0, 1, 2, \ldots, 2^L - 2)$ is the PN-sequence $\{a_i\}$ starting at the term a_n. Therefore,

1. If $a_{n+p} = 0$ for some p, then the l.r.r. in (8) holds for the corresponding sequence $\{s_k\}_p$ and its $LC(\{s_k\}_p) \leq M_1$.
2. If $a_{n+p} = 1$ for some p, then the l.r.r. in (8) does not hold for the corresponding sequence $\{s_k\}_p$ and its LC takes the previous value $LC(\{s_k\}_p) = M_0$ (as for M_0 the l.r.r. was satisfied).

In fact, there are $(2^{L-1}-1)$ terms $a_{n+p}=0$, which are the zeros of the PN-sequence. Thus, we have $(2^{L-1}-1)$ generalized sequences with $LC \leq M_1$. In the same way, there are 2^{L-1} terms $a_{n+p}=1$, which are the ones of the PN-sequence. Thus, we have 2^{L-1} generalized sequences with $LC = M_0$.

For $M_2 < M_1$ and taking the coefficients c_i of the M_2th row in the Sierpinski's triangle, we compute two alternative values of the l.r.r when j ranges in the interval $j \in [0, 1, 2, \ldots, 2^{L-1}-1]$.

$$\sum_{i=0}^{M_2} c_i\ u_{i+j} = \alpha^{n_1} \neq 0 \quad \text{or} \quad \sum_{i=0}^{M_2} c_i\ u_{i+j} = \alpha^{n_2} \neq 0.$$

This yields to:

$$\sum_{i=0}^{M_2} c_i\ a_{\tau(i+j)+p} = a_{n_1+p} \quad \text{or} \quad \sum_{i=0}^{M_2} c_i\ a_{\tau(i+j)+p} = a_{n_2+p}. \tag{9}$$

Therefore, we get two shifted versions of the same PN-sequence $\{a_i\}$, one of them starting at a_{n_1} and the other at a_{n_2}.

1. If $a_{n_1+p} = a_{n_2+p} = 0$ for some p, then the l.r.r. in (9) holds for the corresponding sequence $\{s_k\}_p$ and its $LC(\{s_k\}_p) \leq M_2$.
2. If $a_{n_1+p} = a_{n_2+p} = 1$ for some p, then the l.r.r. in (9) does not hold for the corresponding sequence $\{s_k\}_p$ and its LC takes the previous value $LC(\{s_k\}_p) = M_1$.

Thus, we have $(2^{L-2}-1)$ terms $a_{n_1+p} = a_{n_2+p} = a_{n+p} = 0$, that is there are $(2^{L-2}-1)$ generalized sequences with $LC(\{s_k\}_p) \leq M_2$. In the same way, we have 2^{L-2} terms $a_{n_1+p} = a_{n_2+p} = 1$ and $a_{n+p} = 0$, that is there are 2^{L-2} generalized sequences with $LC(\{s_k\}_p) = M_1$.

For successive and decreasing values of M_i, we get 1, 2, 4, 8, 16, $\ldots$, 2^{L-2} shifted versions of the PN-sequence $\{a_i\}$, respectively. Each one of these $L-1$ groups of shifted sequences provides with $2^{L-1}, 2^{L-2}, 2^{L-3}, \ldots, 4, 2, 1$, generalized sequences with $LC = M_0, M_1, M_2, \ldots, M_{L-3}, M_{L-2} = 2, M_{L-1} = 1$, respectively. The values M_i $(i = 1, 2, \ldots, L-3)$ are not necessarily consecutive values. In fact, if any M_i does not provide new values in the Eq. (6), then M_i is decreased in one unit and the process is repeated for the new M_i.

Recall that he number of generalized sequences that satisfy (do not satisfy) the linear recurrence relationship in one step is half the number of sequences obtained in the previous step. Now a numerical example illustrates the computational method.

3.3 An illustrative example

Let us consider a maximum-length LFSR of length $L = 5$, with characteristic polynomial $p(x) = x^5 + x^3 + 1$, initial state (1, 1, 1, 1, 1) and period $T = 31$. The sequence $\{u_i\} = \{1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^7, \alpha^8, \alpha^{10}, \alpha^{13}, \alpha^{18}, \alpha^{20}, \alpha^{22}, \alpha^{23}, \alpha^{24}, \alpha^{26}, \alpha^{27}\}$ and $\alpha^{18} = \alpha^{17} + 1$. We compute the linear recurrence relationship for different values M_i.

1. For $M_0 = 2^{L-1} - (L-2) = 13$:
   According to Theorem 1, the sequence $\{u_i\}$ satisfies the l.r.r. given by the Eq. (6), where the coefficients c_i are the bits of the 13th row in Fig. 2b, that is (1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1). Consequently, the linear complexity of all the generalized sequences is $LC(\{s_k\}_p) \leq 13$.

2. For $M_1 = 2^{L-1} - (L-2) - 1 = 12$:
   We check the Eq. (6) with $M_1 = 12$ where the coefficients c_i are now the bits of the 12th row in Fig. 2b, that is $(1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1)$. In fact, $\sum_{i=0}^{12} c_i\ \alpha^{\tau(i+j)+p} = \alpha^{29} \neq 0$, then $\{a_{29+p}\}$ $(p = 0, 1, 2, \ldots, 30)$ is the PN-sequence $\{a_i\}$ starting at the term a_{29}.
   Thus, according to Table 2, the 15 generalized sequences corresponding to $p = 0, 1, 7, 8, 11, 13, 14, 16, 17, 18, 19, 21, 23, 27, 30$ (the zeros of $\{a_{29+p}\}$ in bold) will satisfy $LC(\{s_k\}_p) \leq 12$; while the 16 generalized sequences corresponding to the remainder values of p (the ones of $\{a_{29+p}\}$) will have $LC(\{s_k\}_p) = 13$. Therefore, we have computed the LC of 16 generalized sequences just by analysing the binary digits of the PN-sequence $\{a_{29+p}\}$.
3. For $M_2 = 2^{L-1} - (L-2) - 2 = 11$:
   We check the Eq. (6) for $M_2 = 11$ where the coefficients c_i are now the bits of the 11th row in Fig. 2b, that is $(1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1)$. In fact, $\sum_{i=0}^{11} c_i\ \alpha^{\tau(i+j)+p} = \alpha^{19}, \alpha^{25} \neq 0$, alternatively for the successive values of j.
   Thus, according to Table 3, the 7 generalized sequences corresponding to $p = 0, 11, 17, 18, 21, 23, 27$ will satisfy $LC(\{s_k\}_p) \leq 11$. Recall that the previous values of p correspond to the zeros coinciding in the three shifted PN-sequences $\{a_{29+p}\} = \{a_{19+p}\} = \{a_{25+p}\} = 0$, see the columns in bold in Table 3.
   On the other hand, the 8 generalized sequences corresponding to $p = 1, 7, 8, 13, 14, 16, 19, 30$ will have $LC(\{s_k\}_p) = 12$. Recall that the previous values of p correspond to the ones coinciding in the two shifted PN-sequences $\{a_{19+p}\} = \{a_{25+p}\} = 1$ (on grey rectangles) but with $\{a_{29+p}\} = 0$, see the columns in Table 3. Therefore, the comparison of binary digits in three shifted version of a single PN-sequence allows us to compute the LC of 8 generalized sequences.
4. For $M_3 = 2^{L-1} - (L-2) - 3 = 10$:
   We check the Eq. (6) for $M_3 = 10$ where the coefficients c_i are now the bits of the 10th row in Fig. 2b, that is $(1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1)$. In fact, $\sum_{i=0}^{10} c_i\ \alpha^{\tau(i+j)+p} = \alpha^{11}, \alpha^{30}, \alpha^{28}, \alpha^{12} \neq 0$, alternatively for the successive values of j.
   Thus, according to Table 4, the three generalized sequences corresponding to $p = 0, 17, 18$ will satisfy $LC(\{s_k\}_p) \leq 10$. Recall that the previous values of p correspond to the zeros coinciding in the seven shifted PN-sequences $\{a_{29+p}\} = \{a_{19+p}\} = \{a_{25+p}\} = \{a_{11+p}\} = \{a_{30+p}\} = \{a_{28+p}\} = \{a_{12+p}\} = 0$, see the columns in bold in Table 4.
   On the other hand, the 4 generalized sequences corresponding to $p = 11, 21, 23, 27$ will have $LC(\{s_k\}_p) = 11$. Recall that the previous values of p correspond to the ones coinciding in the 4 shifted PN-sequences $\{a_{11+p}\} = \{a_{30+p}\} = \{a_{28+p}\} = \{a_{12+p}\} = 1$ (on grey rectangles) but with $\{a_{29+p}\} = \{a_{19+p}\} = \{a_{25+p}\} = 0$, see the columns in Table 4. Now, the comparison of binary digits in seven shifted version of a single PN-sequence allows us to compute the LC of 4 generalized sequences.
5. When the LC of the last 3 generalized sequences is considered, then the successive M_i $(M_i = 9, 8, \ldots, 2)$ do not provide new values in the Eq. (6).
6. Until for $M_i = 1$:
   We check the Eq. (6) for $M_i = 1$ where the coefficients c_i are now the bits of the 1st row in Fig. 2b, that is $(1, 1)$. In fact, $\sum_{i=0}^{1} c_i\ \alpha^{\tau(i+j)+p} = \alpha^{14}, \alpha^{15}, \alpha^{16}, \alpha^{17}, \alpha^{9}, \alpha^{21}, \alpha^{5}, \alpha^{6} \neq 0$ for the successive values of j. Thus, according to Table 5, the single generalized sequence corresponding to $p = 0$ will satisfy $LC(\{s_k\}_p) = 1$. It is the identically 1 sequence. Recall that the value $p = 0$ corresponds to the zeros coinciding in the 15 shifted PN-sequences $\{a_{29+p}\} = \{a_{19+p}\} = \{a_{25+p}\} = \{a_{11+p}\} = \{a_{30+p}\} = \{a_{28+p}\} =$

Table 2 Linear Complexity of generalized sequences

p =	0		4					8				12				16					20				24				28		30
$\{a_{29+p}\}$	**0**	**0**	1	1	1	1	1	**0**	**0**	1	1	**0**	1	**0**	**0**	1	**0**	**0**	**0**	**0**	1	**0**	1	**0**	1	1	1	**0**	1	1	**0**

Table 3 Linear Complexity of generalized sequences

p =	**0**			**4**				**8**				**12**				**16**					**20**				**24**				**28**		**30**
$\{a_{29+p}\}$	**0**	0	1	1	1	1	1	0	0	1	1	**0**	1	0	0	1	0	**0**	**0**	0	1	**0**	1	**0**	1	1	1	**0**	1	1	0
$\{a_{19+p}\}$	**0**	1	0	1	1	1	0	1	1	0	0	**0**	1	1	1	1	1	**0**	**0**	1	1	**0**	1	**0**	0	1	0	**0**	0	0	1
$\{a_{25+p}\}$	**0**	1	1	0	0	0	1	1	1	1	1	**0**	0	1	1	0	1	**0**	**0**	1	0	**0**	0	**0**	1	0	1	**0**	1	1	1

Table 4 Linear Complexity of generalized sequences

p =	**0**			**4**				**8**				**12**				**16**					**20**				**24**				**28**		**30**
$\{a_{29+p}\}$	**0**	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	**0**	**0**	0	1	0	1	0	1	1	1	0	1	1	0
$\{a_{19+p}\}$	**0**	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	**0**	**0**	1	1	0	1	0	0	1	0	0	0	0	1
$\{a_{25+p}\}$	**0**	1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	1	**0**	**0**	1	0	0	0	0	1	0	1	0	1	1	1
$\{a_{11+p}\}$	**0**	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	**0**	**0**	0	1	1	1	1	1	0	0	1	1	0	1
$\{a_{30+p}\}$	**0**	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	**0**	**0**	1	0	1	0	1	1	1	0	1	1	0	0
$\{a_{28+p}\}$	**0**	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	**0**	**0**	0	0	1	0	1	0	1	1	1	0	1	1
$\{a_{12+p}\}$	**0**	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	**0**	**0**	1	1	1	1	1	0	0	1	1	0	1	0

$\{a_{12+p}\} = \{a_{14+p}\} = \{a_{15+p}\} = \{a_{16+p}\} = \{a_{17+p}\} = \{a_{9+p}\} = \{a_{21+p}\} = \{a_{5+p}\} = \{a_{6+p}\} = 0$, see the column in bold in Table 5.

On the other hand, the 2 generalized sequences corresponding to $p = 17, 18$ will satisfy $LC(\{s_k\}_p) = 2$. They correspond to the generalized sequences $\{1010\ldots\}$ and $\{0101\ldots\}$. Recall that the previous values of p correspond to the ones coinciding in the eight shifted PN-sequences $\{a_{14+p}\} = \{a_{15+p}\} = \{a_{16+p}\} = \{a_{17+p}\} = \{a_{9+p}\} = \{a_{21+p}\} = \{a_{5+p}\} = \{a_{6+p}\} = 1$ (on grey rectangles) but with $\{a_{29+p}\} = \{a_{19+p}\} = \{a_{25+p}\} = \{a_{11+p}\} = \{a_{30+p}\} = \{a_{28+p}\} = \{a_{12+p}\} = 0$, see the columns in Table 5.

In this way, we have computed the LC of the whole family of generalized sequences for this example. The numerical results are depicted in Table 6.

In the general case, there will be 2^{L-1} generalized sequences with $LC = M_0 = 2^{L-1} - (L-2)$, 2^{L-2} sequences with $LC = M_1 < M_0$, 2^{L-3} sequences with $LC = M_2 < M_1, \ldots$, until we get $2^{L-(L-2)} = 4$ sequences with $LC = M_{L-3} < M_{L-4}$. Finally, we obtain $2^{L-(L-1)} = 2$ sequences, the generalized sequences $\{101010\ldots\}$ and $\{010101\ldots\}$, with $LC = M_{L-2} = 2$ and $2^{(L-L)} = 1$ sequence, the identically 1 sequence $\{111111\ldots\}$, with $LC = M_{L-1} = 1$. Table 7 depicts the different linear complexities for an arbitrary family of generalized sequences.

4 Discussion of the method and application to self-decimated generators

To our knowledge, no numerical method of computing the LC of the generalized sequences can be found in the literature, apart from the Berlekamp–Massey algorithm [17] that can

Table 5 Linear complexity of generalized sequences

p =	0				4			8				12				16				20				24				28		30
$\{a_{29+p}\}$	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0
$\{a_{19+p}\}$	0	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1
$\{a_{25+p}\}$	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1
$\{a_{11+p}\}$	0	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	1
$\{a_{30+p}\}$	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0
$\{a_{28+p}\}$	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1
$\{a_{12+p}\}$	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	1	1	0	1
$\{a_{14+p}\}$	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1
$\{a_{15+p}\}$	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0
$\{a_{16+p}\}$	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0
$\{a_{17+p}\}$	0	1	0	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	0
$\{a_{9+p}\}$	0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1
$\{a_{21+p}\}$	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1
$\{a_{5+p}\}$	0	0	1	1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1
$\{a_{6+p}\}$	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0

Table 6 Linear complexity of this family of generalized sequences

Generalized sequences	LC
$\{s_k\}_p$ with $p = 2, 3, 4, 5, 6, 9, 10, 12, 15, 20, 22, 24, 25, 26, 28, 29$	13
$\{s_k\}_p$ with $p = 1, 7, 8, 13, 14, 16, 19, 30$	12
$\{s_k\}_p$ with $p = 11, 21, 23, 27$	11
$\{s_k\}_p$ with $p = 17, 18$	2
$\{s_k\}_p$ with $p = 0$	1

Table 7 Linear complexity of an arbitrary family of generalized sequences

Number of generalized sequences	LC
2^{L-1} sequences	$M_0 = 2^{L-1} - (L-2)$
2^{L-2} sequences	$M_1 < M_0$
2^{L-3} sequences	$M_2 < M_1$
…	…
$2^{L-(L-2)}$ sequences	$M_{L-3} < M_{L-4}$
$2^{L-(L-1)} = 2$ sequences	$M_{L-2} = 2$
$2^{L-(L)} = 1$ sequence	$M_{L-1} = 1$

be applied to any binary sequence. In this section, the numerical method here proposed is discussed. Next, the application of such a method to other self-decimated sequence generators is presented.

4.1 The method of computing the *LC* of the generalized sequences: a discussion

The method here developed involves very simple operations: computation and checking of particular bits of a single PN-sequence $\{a_i\}$ whose characteristic polynomial $p(x)$ of degree L is known.

Useful facts for the computation of the LC can be enumerated:

1. According to the properties of the PN-sequences [12] and the l.r.r. in Eq. (1), any arbitrary term a_i of a PN-sequence can be expressed as a linear combination of its L first terms $(a_0, a_1, \ldots, a_{L-1})$, that is $a_i = \sum_{j=0}^{L-1} d_j^i \, a_j$ with $d_j^i \in \mathbb{F}_2$, where d_j^i $(j = 0, \ldots, L-1)$ are the coefficients of the polynomial $d^i(x) = d_{L-1}^i x^{L-1} + \cdots + d_1^i x + d_0^i$ computed as follows:

$$d^i(x) \equiv x^i \bmod p(x), \tag{10}$$

that is the polynomial x^i reduced modulo $p(x)$.
2. According to the one-to-one correspondence given in (3), any element α^i in $\mathbb{F}_{2^L}$ can be written as a linear combination of the L first powers $(1, \alpha, \alpha^2, \ldots, \alpha^{L-1})$, that is $\alpha^i = \sum_{j=0}^{L-1} d_j^i \, \alpha^j$ where $d_j^i \in \mathbb{F}_2$ are the coefficients of the polynomial $d^i(x)$ defined in (10). Thus, any term of the sequence $\{u_i\}$, that is a power of α, can be written as a linear combination of $(1, \alpha, \alpha^2, \ldots, \alpha^{L-1})$ weighted by binary coefficients.
3. All the shifted versions of $\{a_i\}$ starting with 0 are arranged into groups of $1, 2, 4, \ldots, 2^{L-2}$ sequences, in total $L-1$ groups and $2^{L-1}-1$ sequences, see Table 5. In practice, only one sequence of each group is needed. More precisely, just one sequence of each one of the $L-2$ first groups, notated $\{a_{m_j}\}$ $(0 \le j \le L-3)$ where $\{a_{m_j}\}$ is the PN-sequence starting at a_{m_j}, will be needed for the computation. In fact, the last group only provides the linear complexities of the sequences $\{101010\ldots\}$, $\{010101\ldots\}$ and $\{111111\ldots\}$, which are already known.
4. Only L bits of each sequence $\{a_{m_j}\}$ $(0 \le j \le L-3)$ are necessary to compute the specific term a_{m_j+p} via the representation given in item 1. Thus, the memory requirements of this method, $(L-2) \cdot L$ bits, are minimum.
5. The fractal structure of the Sierpinski's triangle simplifies the computation of the l.r.r. given in (6) when the Nth row of this triangle is considered.

   (a) For $2^{L-2} \le N < 2^{L-1}$, that is $N = 2^{L-2} + k$ with $(0 \le k < 2^{L-2})$, the Eq. (6) with $j = 0$ can be decomposed as follows:

   $$\sum_{i=0}^{N} c_i \, u_i = \sum_{i=0}^{k} c_i \, u_i + \sum_{i=2^{L-2}}^{2^{L-2}+k} c_i \, u_i, \tag{11}$$

   where the first summation corresponds to the Eq. (6) for the kth row of the Sierpinski's triangle. At its turn, such a summation can be decomposed and computed recursively. The second summation is the new expression to be computed.

   (b) Due to the symmetry of the Sierpinski's triangle, the coefficients c_i satisfy:

   $$c_i = c_{i+2^{L-2}} \quad (i = 0, \ldots, k).$$

   (c) The Nth row of the Sierpinski's triangle has, at most, 2^{L-2} coefficients $c_i = 1$, half in the firs summation and half in the second summation of (11). The remaining coefficients equal zero.

In order to compute the LC of an arbitrary generalized sequence $\{s_k\}_p$, the successive bits a_{m_j+p} $(0 \le j \le L-3)$ are computed until a bit $a_{m_j+p} = 1$ is obtained, then the linear complexity of such a sequence is $LC(\{s_k\}_p) = M_{i-1}$.

A comparison between the Berlekamp–Massey algorithm and the numerical method here proposed can be stated. In the worst case, for a generalized sequence $\{s_k\}_p$ with maximum $LC = 2^{L-1} - (L-2)$ the requirements are:

(a) The Berlekamp–Massey algorithm needs $2 \cdot LC$ digits to compute the linear complexity of the generalized sequence. Thus, it has to generate and process $2^L - 2 \cdot (L-2) \simeq 2^L$ bits (nearly twice the period of the sequence) or it has to generate and store one period of the sequence to be processed twice. Time complexity of this algorithm is $\mathbf{O}(2^L)$.

(b) The method here proposed needs to compute:

$$\sum_{i=0}^{M_1} c_i \; u_i = \alpha^{m_0}, \tag{12}$$

where every $u_i \in \mathbb{F}_{2^L}$ is represented as mentioned in item 2, $M_1 = 2^{L-1} - (L-2) - 1$ and the summation can be decomposed as mentioned in item 5. Consequently, the Eq. (12) just performs bit-wise additions modulo 2 with time complexity $\mathbf{O}(2^{L-2})$. Next, the L first bits of the sequence $\{a_{m_0}\}$, that is the sequence $\{a_i\}$ starting at the term a_{m_0}, are enough to determine the bit a_{m_0+p} by means of the modular expression

$$d^{m_0+p}(x) \equiv x^{m_0+p} \bmod p(x).$$

Except for the L bits of $\{a_{m_0}\}$, no more digits of the PN-sequence have to be stored. Thus, the memory requirements of this method are minimum compared with those of the Berlekamp–Massey algorithm.

In brief, the computational complexities in what memory and speed are concerned are in the method here proposed less than in the Berlekamp–Massey algorithm.

In cryptographic terms, LC must be as large as possible so the computing method allows us to determine generalized sequences with guaranteed maximum LC, that is $LC = 2^{L-1} - (L-2)$. In fact, any term $a_{m_0+p} = 1$, see Table 2, provides us with the shifting p of a generalized sequence $\{s_k\}_p$ whose linear complexity meets the maximum value.

4.2 Application to other self-decimated sequence generators

This method of computing the linear complexity can be extended to the sequences produced by self-decimated generators.

(a) Computation of LC for the self-shrinking generator:

In this case, the two shifted versions of the PN-sequence involved in the decimation rule are [18]:

$$\{Seq(0)\} : a_0 \; a_2 \; a_4 \; \ldots \; a_{2^L-2} \; a_1 \; a_3 \; \ldots \; a_{2^L-3}$$
$$\{Seq(1)\} : a_1 \; a_3 \; a_5 \; \ldots \; a_{2^L-3} \; a_0 \; a_2 \; \ldots \; a_{2^L-2}$$

Thus, the relative shift between both sequences is the value p solution of the equation:

$$0 + p \cdot 2 \equiv 1 \bmod 2^L - 1.$$

That is, $p = 2^{L-1}$. Therefore, the self-shrunken sequence is the generalized sequence $\{s_k\}_{2^{L-1}}$ generated from the PN-sequence $\{a_0, a_2, a_4, a_6, \ldots\}$ and whose LC can be computed by means of the method developed in Sect. 3.

In [18], the authors only give a lower bound on the LC of the self-shrunken sequence corresponding to the expression

$$2^{\lfloor L/2 \rfloor - 1} < LC,$$

here we present a method of computing the exact value of LC.

(b) Computation of LC for the modified self-shrinking generator:

In this case, the three shifted versions of the PN-sequence involved in the decimation rule are [15]:

$$\begin{array}{l}\{Seq(0)\}: a_0\ a_3\ a_6\ a_9\ a_{12}\ a_{15}\ \ldots\\ \{Seq(1)\}: a_1\ a_4\ a_7\ a_{10}\ a_{13}\ a_{16}\ \ldots\\ \{Seq(2)\}: a_2\ a_5\ a_8\ a_{11}\ a_{14}\ a_{17}\ \ldots\end{array}$$

The bit-wise XOR of $Seq(0)$ and $Seq(1)$ gives:

$$\begin{array}{rl}\{Seq(0)\}: & a_0\ \ a_3\ \ a_6\ \ a_9\ \ a_{12}\ \ a_{15}\ \ \ldots\\ \oplus\{Seq(1)\}: & a_1\ \ a_4\ \ a_7\ \ a_{10}\ \ a_{13}\ \ a_{16}\ \ \ldots\\ \hline \{a_{d+3\cdot i}\}: & a_d\ a_{d+3}\ a_{d+6}\ a_{d+9}\ a_{d+12}\ a_{d+15}\ \ldots\end{array}$$

where $a_d = a_0 + a_1$, d being an integer $2 \leq d \leq 2^L - 2$. Therefore the two shifted versions of the same PN-sequence involved in the decimation rule are:

$$\begin{array}{l}a_d\ a_{d+3}\ a_{d+6}\ a_{d+9}\ a_{d+12}\ a_{d+15}\ a_{d+18}\ \ldots\\ a_2\ \ a_5\ \ \ a_8\ \ \ a_{11}\ \ a_{14}\ \ a_{17}\ \ \ a_{20}\ \ \ldots\end{array}$$

Thus, the relative shift between both sequences is the value p solution of the equation:

$$d + p \cdot 3 \equiv 2 \bmod 2^L - 1. \tag{13}$$

Therefore, the modified self-shrunken sequence is the generalized sequence $\{s_k\}_p$ with p defined in (13) and generated from the PN-sequence $\{a_d, a_{d+3}, a_{d+6}, a_{d+9}, a_{d+12}, \ldots\}$. Thus, its LC can be computed by means of the method developed in Sect. 3.

In [15], the author only gives a lower bound on the LC of the modified self-shrunken sequence corresponding to the expression

$$2^{\lfloor L/3 \rfloor - 1} < LC,$$

here we present a method of computing the exact value of LC.

(c) Computation of LC for the t-modified self-shrinking generator [5]:

Assume that g.c.d.$(t, 2^L - 1) = 1$. In this case, the t shifted versions of the PN-sequence involved in the decimation rule are:

$$\begin{array}{rl}\{Seq(0)\}: & a_0\ \ a_t\ \ a_{2t}\ \ a_{3t}\ \ a_{4t}\ \ a_{5t}\ \ \ldots\\ \{Seq(1)\}: & a_1\ \ a_{t+1}\ \ a_{2t+1}\ \ a_{3t+1}\ \ a_{4t+1}\ \ a_{5t+1}\ \ \ldots\\ \vdots & \vdots\ \ \ \vdots\ \ \ \vdots\\ \{Seq(t-2)\}: & a_{t-2}\ a_{t+t-2}\ a_{2t+t-2}\ a_{3t+t-2}\ a_{4t+t-2}\ a_{5t+t-2}\ \ldots\\ \{Seq(t-1)\}: & a_{t-1}\ a_{t+t-1}\ a_{2t+t-1}\ a_{3t+t-1}\ a_{4t+t-1}\ a_{5t+t-1}\ \ldots\end{array}$$

The bit-wise XOR of $Seq(0), \ldots, Seq(t-2)$ gives:

$$\begin{array}{rl}\{Seq(0)\}: & a_0\ \ a_t\ \ a_{2t}\ \ a_{3t}\ \ a_{4t}\ \ a_{5t}\ \ \ldots\\ \{Seq(1)\}: & a_1\ \ a_{t+1}\ \ a_{2t+1}\ \ a_{3t+1}\ \ a_{4t+1}\ \ a_{5t+1}\ \ \ldots\\ \vdots & \vdots\ \ \ \vdots\ \ \ \vdots\\ \oplus\{Seq(t-2)\}: & a_{t-2}\ a_{t+t-2}\ a_{2t+t-2}\ a_{3t+t-2}\ a_{4t+t-2}\ a_{5t+t-2}\ \ldots\\ \hline \{a_{d'+t\cdot i}\}: & a_{d'}\ \ a_{d'+t}\ \ a_{d'+2t}\ \ a_{d'+3t}\ \ a_{d'+4t}\ \ a_{d'+5t}\ \ \ldots\end{array}$$

where $a_{d'} = a_0 + a_1 + \cdots + a_{t-2}$, d' being an integer $0 \leq d' \leq 2^L - 2$. Therefore the two shifted versions of the PN-sequence involved in the decimation rule are:

$$\begin{array}{l}a_{d'}\ \ a_{d'+t}\ \ a_{d'+2t}\ \ a_{d'+3t}\ \ a_{d'+4t}\ \ a_{d'+5t}\ \ a_{d'+6t}\ \ \ldots\\ a_{t-1}\ a_{t+t-1}\ a_{2t+t-1}\ a_{3t+t-1}\ a_{4t+t-1}\ a_{5t+t-1}\ a_{6t+t-1}\ \ldots\end{array}$$

Thus, the relative shift between both sequences is the value p solution of the equation:

$$d' + p \cdot t \equiv t - 1 \bmod 2^L - 1. \tag{14}$$

Therefore, the t-modified self-shrunken sequence is the generalized sequence $\{s_k\}_p$ with p defined in (14) and generated from the PN-sequence $\{a_{d'}, a_{d'+t}, a_{d'+2t}, a_{d'+3t}, a_{d'+4t}, a_{d'+5t}, \ldots\}$. Thus, its LC can be computed by means of the method developed in Sect. 3.

5 Conclusions

Although the class of generalized sequences exhibit good cryptographic properties, the parameter Linear Complexity has never been specifically analysed. At any rate, we present the first known method of computing the linear complexity of such sequences. In fact, a practical method based exclusively on the comparison of shifted versions of a single PN-sequence is introduced and developed. The numerical method is efficient and can be applied to sequence generators in a cryptographic range. The procedure allows us to classify, group and compute the LC of the family of generalized sequences. In addition, the procedure can be extended to the class of self-decimated generators due to the close relationship between the sequences produced by such generators and the generalized sequences.

As the fundamental structure of all these computations is the PN-sequence, the systematic extension of these ideas to the whole class of interleaved sequences is intended as the main direction of the future work.

References

1. Blackburn, S.R.: The linear complexity of the self-shrinking generator. IEEE Trans. Inf. Theory **45**(6), 2073–2077 (1999). https://doi.org/10.1109/18.782139
2. Cardell, S.D., Fúster-Sabater, A.: Binomial representation of cryptographic binary sequences and its relation to cellular automata. Complexity **2019**, 1–13 (2019). https://doi.org/10.1155/2019/2108014
3. Cardell, S.D., Fúster-Sabater, A.: Discrete linear models for the generalized self-shrunken sequences. Finite Fields Appl. **47**, 222–241 (2017). https://doi.org/10.1016/j.ffa.2017.06.010
4. Cardell, S.D., Fúster-Sabater, A.: Linear models for high-complexity sequences. In: Gervasi, O., et al. (eds.) Computational Science and Its Applications. Lecture Notes in Computer Science, vol. 10404, pp. 314–324. Springer, New York (2017)
5. Cardell, S.D., Fúster-Sabater, A.: The t-modified self-shrinking generator. In: Y. Shi, H. Fu, Y. Tian, V.V. Krzhizhanovskaya, M.H. Lees, J. Dongarra, P.M.A. Sloot (eds.) Computational Science—ICCS 2018. Lecture Notes in Computer Science, vol. 10860, pp. 653–663. Springer, New York (2018)
6. Coppersmith, D., Krawczyk, H., Mansour, Y.: The shrinking generator. In: D. Stinson (ed.) Advances in Cryptology—CRYPTO '93. Lecture Notes in Computer Science, vol. 773, pp. 22–39. Springer, New York (1994). https://doi.org/10.1007/3-540-48329-2_3
7. Duvall, P.F., Mortick, J.C.: Decimation of periodic sequences. SIAM J. Appl. Math. **21**(3), 367–372 (1971). https://doi.org/10.1137/0121039
8. Fúster-Sabater, A., Caballero-Gil, P.: Synthesis of cryptographic interleaved sequences by means of linear cellular automata. Appl. Math. Lett. **22**, 1518–1524 (2009)
9. Fúster-Sabater, A., Cardell, S.D.: Computing the linear complexity in a class of cryptographic sequences. In: O. Gervasi, B. Murgante, S. Misra, E. Stankova, C. Torre, A. Rocha, D. Taniar, B. Apduhan, E. Tarantino, Y. Ryu (eds.) Computational Science and Its Applications—ICCSA 2018. Lecture Notes in Computer Science, vol. 10960, pp. 110–122. Springer, New York (2018)
10. Fúster-Sabater, A.: Generation of cryptographic sequences by means of difference equations. Appl. Math. Inf. Sci. **8**(2), 475–484 (2014)
11. Fúster-Sabater, A., Caballero-Gil, P.: Chaotic modelling of the generalized self-shrinking generator. Appl. Soft Comput. **11**(2), 1876–1880 (2011). https://doi.org/10.1016/j.asoc.2010.06.002
12. Golomb, S.W.: Shift Register-Sequences. Aegean Park Press, Laguna Hill (1982)

13. Gong, G.: Theory and applications of q-ary interleaved sequences. IEEE Trans. Inf. Theory **41**(2), 400–411 (1995)
14. Hu, Y., Xiao, G.: Generalized self-shrinking generator. IEEE Trans. Inf. Theory **50**(4), 714–719 (2004). https://doi.org/10.1109/TIT.2004.825256
15. Kanso, A.: Modified self-shrinking generator. Comput. Electr. Eng. **36**(5), 993–1001 (2010). https://doi.org/10.1016/j.compeleceng.2010.02.004
16. Lidl, R., Niederreiter, H.: Finite Fields. Cambridge University Press, Cambridge (1997)
17. Massey, J.L.: Shift-register synthesis and BCH decoding. IEEE Trans. Inf. Theory **15**(1), 122–127 (1969). https://doi.org/10.1109/TIT.1969.1054260
18. Meier, W., Staffelbach, O.: The self-shrinking generator. In: A. De Santis (ed.) Advances in Cryptology—EUROCRYPT 1994. Lecture Notes in Computer Science, vol. 950, pp. 205–214. Springer, Berlin (1995). https://doi.org/10.1007/BFb0053436
19. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)
20. Paar, C., Pelzl, J.: Understanding Cryptography. Springer, Berlin (2010)
21. Zhang, B., Feng, D.: New guess-and-determine attack on the self-shrinking generator. In: X. Lai, K. Chen (eds.) Advances in Cryptology—ASIACRYPT 2006.Lecture Notes in Computer Science, vol. 4284, pp. 54–68. Springer, Berlin (2006). https://doi.org/10.1007/11935230_4