The t-modified self-shrinking generator *

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

[1] Instituto de Matemática, Estatística e Computação Científica
UNICAMP, R. Sérgio Buarque de Holanda, 651,
13083-859, Campinas-SP, Brazil
[2]Instituto de Tecnologías Físicas y de la Información, C.S.I.C.
Serrano 144, 28006 Madrid, Spain.
sdcardell@ime.unicamp.br amparo@iec.csic.es

Abstract. Pseudo-random sequences exhibit interesting properties with applications in many and distinct areas ranging from reliable communications to number generation or cryptography. Inside the family of decimation-based sequence generators, the modified self-shrinking generator (an improved version of the self-shrinking generator) is one of its best-known elements. In fact, such a generator divides the PN-sequence produced by a maximum-length LFSR into groups of three bits. When the sum of the first two bits in a group is one, then the generator returns the third bit, otherwise the bit is discarded. In this work, we introduce a generalization of this generator, where the PN-sequence is divided into groups of t bits, $t \geq 2$. It is possible to check that the properties of the output sequences produced by this family of generators have the same or better properties than those of the classic modified self-shrunken sequences. Moreover, the number of sequences generated by this new family with application in stream cipher cryptography increases dramatically.

Keywords: decimation, modified self-shrinking generator, linear complexity, characteristic polynomial

1 Introduction

Many of the pseudo-random sequence generators are based on maximum-length Linear Feedback Shift Registers (LFSRs) [1, 2] whose output sequences, known as PN-sequences, are combined via a non-linear Boolean function in order to produce pseudo-random sequences. Traditionally, LFSRs have been designed to operate over the binary field of two elements, which is an appropriate approach

* The first author was supported by FAPESP with number of process 2015/07246-0 and CAPES. 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 reference S2013/ICE-3095-CIBERDINE-CM, also co-funded by European Union FEDER funds.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50

for hardware implementations. One of the best-known and more promising families of pseudo-random sequence generators is the family of decimation-based generators. The underlying idea of this kind of generators is the irregular decimation of a PN-sequence according to the bits of another one. The result of this decimation is a binary sequence that will be used as keystream sequence for encryption/decryption in stream cipher cryptography [3].

The first generator based on irregular decimation was introduced in 1993 by Coppersmith *et al.* [4] and deeply studied in [5, 6]. Such a generator, called the shrinking generator, uses two maximum-length LFSRs; one generates output bits while the other controls (accepts/rejects) such bits. Later, Meier and Sttafelbach introduced the self-shrinking generator [7], a more simple version of the shrinking generator, where a single PN-sequence decimates itself. Both generators are attractive since they are fast, simple to be implemented and their output sequences exhibit good cryptographic properties. In [8], Kanso introduced the modified self-shrinking generator, a new variant of the self-shrinking generator that used an extended selection rule based on the XORed value of a pair of bits.

In this work, we introduce a new family of keystream generators called the t-modified self-shrinking generators, which is a generalization of the modified self-shrinking generator introduced in [8]. For a given value of t, the PN-sequence is divided into groups of t bits. When the XOR of the first $t-1$ bits of each group is one, then we keep the last bit of the group, otherwise, it is discarded. If the length of the PN-sequence and the parameter t satisfy certain conditions, then the t-modified sequences have similar properties to those of the modified self-shrunken sequence [8] as well as we dramatically increase the number of generated sequences with application in cryptography.

The work is organized as follows: in Section 2, the family of self-shrinking generators are introduced as well as their formation rules and main characteristics. In Section 3, we introduce the novel definition of t-modified self-shrinking generator and some illustrative examples. The properties of the sequences produced by this generator and its relationship with the generalized self-shrinking generator are described in Section 4. Finally, conclusions in Section 5 end the paper.

2 The self-shrinking generators

The **self-shrinking generator** was introduced by Meier and Sttafelbach in [7]. This generator is a more simple version of the shrinking generator [4], 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, the decimation rule can be described as follows: given two consecutive bits $\{a_{2i}, a_{2i+1}\}$, $i = 0, 1, 2, \ldots$, the output sequence $\{s_j\} =$

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50

$\{s_0, s_1, \ldots\}$ 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 sequence $\{s_j\}$ is called the **self-shrunken sequence**. If L is the number of stages of the maximum-length LFSR, then the linear complexity of $\{s_j\}$, denoted by LC, meets the condition $2^{L-2} < LC \leq 2^{L-1} - (L-2)$ [9]. In addition, the characteristic polynomial of this sequence has the form $p_{LC} = (x+1)^{LC}$ [7].

Example 1. Consider the LFSR of $L = 3$ stages with characteristic polynomial $p_1(x) = x^3 + x^2 + 1$ and initial state $\{100\}$. The PN-sequence generated is $\{1001110\ldots\}$. Now the self-shrunken sequence can be computed in the following way:

$$R : \underbrace{1 \quad 0}_{\mathbf{0}} \quad 0 \quad 1 \quad \underbrace{1 \quad 1}_{\mathbf{1}} \quad 0 \quad 1 \quad 0 \quad 0 \quad \underbrace{1 \quad 1}_{\mathbf{1}} \quad \underbrace{1 \quad 0}_{\mathbf{0}} \quad \ldots$$

The self-shrunken sequence $\{0110\ldots\}$ has period $T = 4$ and it is possible to check that its characteristic polynomial is $p_3(x) = (x+1)^3$, then $LC = 3$. ■

The **modified self-shrinking generator** was introduced by Kanso in [8]. The PN-sequence $\{a_i\}$ generated by a maximum-length LFSR is self-decimated as follows: 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**.

According to [8], if L is the number of stages of the LFSR, then the linear complexity LC of the modified self-shrunken sequence satisfies:

$$2^{\lfloor \frac{L}{3} \rfloor - 1} \leq LC \leq 2^{L-1} - (L-2),$$

and the period T, when L is odd, is given by

$$2^{\lfloor \frac{L}{3} \rfloor} \leq T \leq 2^{L-1}.$$

Furthermore, the characteristic polynomial of the modified self-shrinking sequences is of the form $p_{LC} = (x+1)^{LC}$ [10].

Example 2. Consider the LFSR with $L = 5$ stages with characteristic polynomial $p(x) = x^5 + x^2 + 1$ and initial state $\{11111\}$. The PN-sequence generated by this register is the following: $\{1111100011011101010000100101100\ldots\}$. Then, the corresponding modified self-shrunken sequence is given by $\{1100100101110010\}$. The obtained sequence has period $T = 16$ and it can be checked that its characteristic polynomial is $p_4(x) = (x+1)^4$, then $LC = 4$. ■

The key of both generators is the initial state of the LFSR. Additionally, the characteristic polynomial of the register is also recommended to be part of the key.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50

Algorithm: Generating the t-modified self-shrunken sequence

Input: $p(x)$, $\boldsymbol{a}$, t
Compute $T = 2^L - 1$.
Compute $d = \gcd\{T, t\}$.
Compute $t \cdot T/d$ bits of $\{a_i\}$ using the polynomial $p(x)$ and the initial state $\boldsymbol{a}$.
Initialize $\boldsymbol{s}$.
for i=1: $t \cdot T/d$
  if $\sum_{j=1}^{i+t-2} a_j = 1$
    Store a_{i+t-1} in $\boldsymbol{s}$.
  endif
endfor
Output:
The t-modified self shrunken sequence $\{s_j\}$

3 The *t*-modified self-shrinking generator

Consider an LFSR with L stages and characteristic polynomial $p(x)$ that generates the PN-sequence $\{a_i\}$. We can construct an ***t*-modified self-shrinking generator**, with $(t = 2, 3, \ldots, 2^L - 2)$ whose decimation rule is very simple: given t consecutive bits $\{a_{ti}, a_{ti+1}, a_{ti+2}, \ldots, a_{ti+(t-1)}\}$ of the PN-sequence, the t-modified self-shrunken sequence is computed as follows:

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

Notice that the value $t = 2$ gives rise the self-shrunken sequence while the value $t = 3$ produces the modified self-shrunken sequence.

Algorithm 1 shows how to generate the sequence produced by the t-modified self-shrinking generator, given the characteristic polynomial $p(x)$ of the LFSR, an initial state $\boldsymbol{a}$ and the parameter t.

Next, a simple example of t-modified self-shrinking generator is presented.

Example 3. Consider the PN-sequence sequence generated by the primitive polynomial $p(x) = x^7 + x + 1$ and the initial state $\{1111111\}$:

$$\{1111111000000100000110000101000111100100010110011101010011111010000111000100100110110101101111011000110100101110111001100101010\}.$$

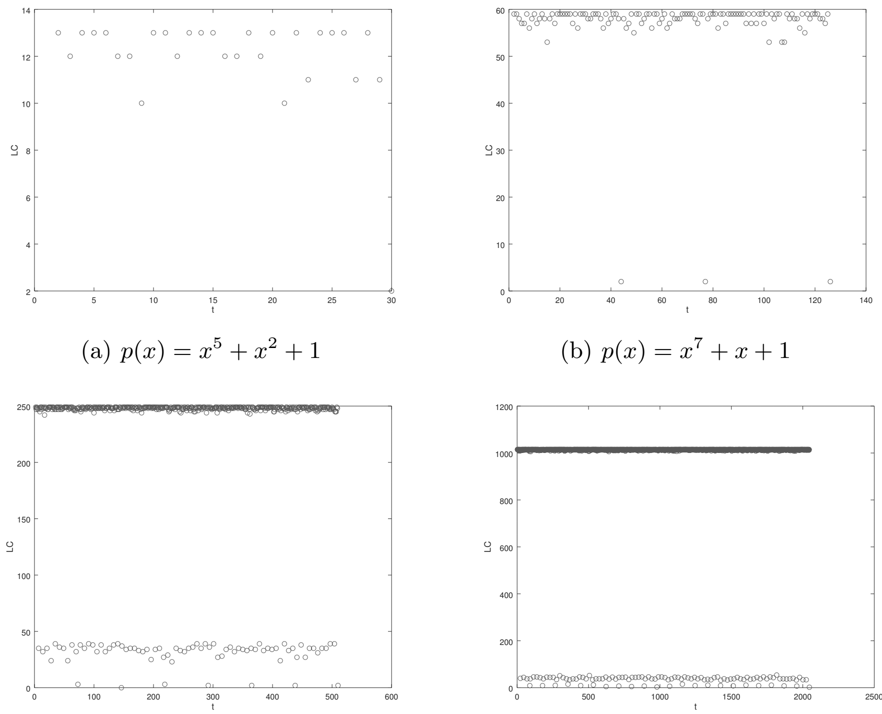
The 5-modified self-shrunken sequence is given by:

$$\{0010101000110110011010010000000111010101101111100101110010111100\}.$$

This sequence has period $T = 64$ and linear complexity $LC = 57$. If we consider the classic modified self-shrunken sequence from the same PN-sequence, then the resultant sequence is given by:

$$\{0010010111100011010100100110010000111111110101000110100011101010\}.$$

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50



(a) $p(x) = x^5 + x^2 + 1$ (b) $p(x) = x^7 + x + 1$

(c) $p(x) = x^9 + x^4 + 1$ (d) $p(x) = x^{11} + x^2 + 1$

Fig. 1. LC for different values of t for different polynomials

This sequence has period $T = 64$ and linear complexity $LC = 59$. This means that the sequence generated by our 5-modified self-shrinking generator is comparable to the classic modified self-shrunken sequence in terms of period and linear complexity. ■

In [8], the author considered exclusively modified self-shrinking generators where the LFSR characteristic polynomial had odd degree. In Figure 1, it is possible to check the values of the linear complexity for several t-modified self-shrunken sequences with different values of t and different polynomial degrees.

In Figure 1(a), we consider the LFSR with primitive polynomial $p(x) = 1 + x^2 + x^5$ and initial state $\{11111\}$. It is possible to check that the linear complexity of the sequences generated by different $t = 2, 3, \ldots, 30$ flutuates between 10 and 13.

In Figure 1(b), we consider the LFSR with primitive polynomial $p(x) = 1 + x + x^7$ and initial state $\{1111111\}$. In this case, we consider $t = 2, 3, \ldots, 126$ and in most cases LC oscillates between the values 53 and 59. Nevertheless, there are also a few cases where the linear complexity is 2.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50

In Figure 1(c), we consider the LFSR with a primitive polynomial $p(x) = 1 + x^4 + x^9$ (with degree different from a prime number) and initial state $\{111111111\}$. In this case, it is possible to check that the range of values of LC is much wider than that of the previous examples. In most cases the LC is between 242 and 249. Nevertheless, there are other cases where the LC ranges in the interval between 20 and 40 as well as there are also a few cases with complexity 2, 3 or 0.

In Figure 1(d), we consider the LFSR with primitive polynomial $p(x) = 1+x^2+x^{11}$ and initial state $\{11111111111\}$. In this case, 11 is prime but $2^{11}-1$ is not prime and it happens the same fact as that of the previous case. In general, the LC is between 1000 and 1015. However, there are several cases where the LC is much smaller.

The previous numerical results for the LC of the t-modified self-shrunken sequences will be justified in next section.

4 Analysis of the sequences

In order to analyse the characteristics of the t-modified self-shrunken sequences, two fundamental concepts, the generalized self-shrinking generator and the cyclotomic cosets, are introduced.

The generalized self-shrinking generator:

Let $\{a_i\}$ be a PN-sequence generated by a maximum-length LFSR of L stages. Let G be an L-dimensional binary vector $G = (g_0, g_1, g_2, ..., g_{L-1}) \in \mathbb{F}_{2^L}$ and $\{v_i\}$ a sequence defined as: $v_i = g_0 a_i \oplus g_1 a_{i-1} \oplus g_2 a_{i-2} \oplus \cdots \oplus g_{L-1} a_{i-L+1}$, where the symbol $\oplus$ represents the XOR logic operation. For $i \geq 0$, let us define the following decimation rule:

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

The sequence generator with the previous decimation rule is known as the **generalized self-shrinking generator** [11]. Its output sequence $\{s_j\}$, denoted by $s(G)$, is called the **generalized self-shrunken sequence** associated with the vector G.

When G ranges over $\mathbb{F}_{2^L}$, $\{v_i\}$ corresponds to the 2^L-1 possible shifts of $\{a_i\}$. Furthermore, the set of sequences denoted by $S(a) = \{s(G) \mid G \in \mathbb{F}_{2^L}\}$ is the **family of generalized self-shrunken sequences** based on the PN-sequence $\{a_i\}$.

It is worth noticing that the sequence $\{v_i\}$ is a shifted version of the sequence $\{a_i\}$. When the sequence $\{v_i\}$ is shifted 2^{L-1} bits regarding the sequence $\{a_i\}$ [12], then the generated sequence $\{s_j\}$ is the self-shrunken sequence introduced in Section 2. The family of generalized self-shrunken sequences includes the identically null sequence $\{0000\ldots\}$, the identically 1 sequence $\{1111\ldots\}$ and the sequences $\{1010\ldots\}$ and $\{0101\ldots\}$ with $T=2$ and $LC=2$. The remaining elements of this family are balanced and have period $T = 2^{L-1}$ and LC satisfies

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

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50

Table 1. Family $S(a)$ of GSS-sequences generated by $p(x) = x^3 + x + 1$

	G	$\{v_i\}$	$s(G)$
0	0 0 0	$\underline{0}\ \underline{0}\ \underline{0}\ 0\ 0\ \underline{0}\ 0$	0 0 0 0
1	0 0 1	$\underline{1}\ \underline{0}\ \underline{1}\ 1\ 1\ \underline{0}\ 0$	1 0 1 0
2	0 1 0	$\underline{0}\ \underline{1}\ \underline{1}\ 1\ 0\ \underline{0}\ 1$	0 1 1 0
3	0 1 1	$\underline{1}\ \underline{1}\ \underline{0}\ 0\ 1\ \underline{0}\ 1$	1 1 0 0
4	1 0 0	$\underline{1}\ \underline{1}\ \underline{1}\ 0\ 0\ \underline{1}\ 0$	1 1 1 1
5	1 0 1	$\underline{0}\ \underline{1}\ \underline{0}\ 1\ 1\ \underline{1}\ 0$	0 1 0 1
6	1 1 0	$\underline{1}\ \underline{0}\ \underline{0}\ 1\ 0\ \underline{1}\ 1$	1 0 0 1
7	1 1 1	$\underline{0}\ \underline{0}\ \underline{1}\ 0\ 1\ \underline{1}\ 1$	0 0 1 1
$\{a_i\}$		1 1 1 0 0 1 0	

Example 4. Consider the LFSR with characteristic polynomial $p(x) = x^3+x+1$ and output PN-sequence $\{1\ 1\ 1\ 0\ 0\ 1\ 0\}$. For this parameters, we can compute the generalized self-shrinking sequences shown in Table 1. The underlined bits in the different sequences $\{v_i\}$ are the digits of the corresponding $s(G)$ sequences. The PN-sequence $\{a_i\}$ is written at the bottom of the table. Note that in this example there are exactly 4 different sequences. The remanining sequences are just shifted versiones of these four sequences. Furthermore, the self-shrunken sequence computed in Example 1 corresponds to the GSS-sequence number 2.■

Now, let us consider the concept of cyclotomic coset $\bmod(2^L - 1)$ given in [1].

Cyclotomic cosets $\bmod(2^L - 1)$: Let $\mathbb{Z}_{2^L}$ denote the set of integers with 2^L elements. An equivalence relation R is defined on its elements $k_1, k_2 \in \mathbb{Z}_{2^L}$ such as follows: $k_1\ R\ k_2$ if there exists an integer j, $0 \le j \le L-1$, such that

$$2^j \cdot k_1 = k_2 \bmod (2^L - 1).$$

The resultant equivalence classes into which $\mathbb{Z}^*_{2^L}$ is partitioned are called the **cyclotomic cosets** mod $(2^L - 1)$. The leader element of every coset is the smallest integer in such an equivalence class. The cardinal of a coset (the number of elements in such a coset) is L or a proper divisor of L. The characteristic polynomial of a cyclotomic coset E is a polynomial $P_E(x) = (x+\alpha^E)(x+\alpha^{2E})\ldots$ $(x + \alpha^{2^{r-1}E})$, where the degree r ($r \le L$) of $P_E(x)$ equals the cardinal of the coset E and α is a root of the LFSR characteristic polynomial.

Example 5. Consider the set $\mathbb{Z}^*_{2^5}$. There are six cyclotomic cosets given by:

$$C_1 = \{1, 2, 4, 8, 16\} \quad C_5 = \{5, 10, 20, 9, 18\} \quad C_{15} = \{15, 30, 29, 27, 23\}$$
$$C_3 = \{3, 6, 12, 24, 17\} \quad C_7 = \{7, 14, 28, 25, 19\} \quad C_{11} = \{11, 22, 13, 26, 21\}$$

In this case, all cosets are *proper* cosets in Golomb's terminology [1, Chapter 4] and have cardinal 5. ■

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50

Notice that when $2^L - 1$ is prime, known as *Mersenne prime*, then the number of primitive polynomials of degree L coincides with the number of cyclotomic cosets of cardinal L in $\mathbb{Z}^*_{2^L}$. Furthermore, each coset has L elements and an associated primitive polynomial of degree L (see [1]).

Notice that when $2^L - 1$ is not prime, then different types of cyclotomic cosets can appear:

1. Cyclotomic cosets with cardinal L whose associated polynomial is primitive.
2. Cyclotomic cosets with cardinal L whose associated polynomial is irreducible but not primitive.
3. Cyclotomic cosets with cardinal r, where r is a proper divisor of L, whose associated polynomial is primitive or irreducible of degree r.

In fact, if $\gcd(2^L - 1, t) = 1$, then the PN-sequence $\{a_i\}$ decimated by distance t gives rise to a new PN-sequence $\{b_i\}$ and the sum $\sum_{j=0}^{t-2} a_{ti+j}$ of $t-1$ bits in equation (1) is just a bit of $\{b_i\}$. Thus, in this case the decimation rule of the t-modified self shrinking generator coincides with that of the generalized self-shrinking generator [13].

Depending on the type of coset in which t takes values, the corresponding t-modified self-shrunken sequences will have different values for the linear complexity. Observing the previous examples, we can draw the following conclusions:

- When $2^L - 1$ is prime, all the t-modified sequences generated with $t = 2, 3, \ldots, 2^L - 2$ are generalized sequences obtained from different primitive polynomials of degree L. Thus, the LC of such sequences satisfies the equation (2). It is the case of Figure 1(a) and Figure 1(b) whose LC satisfies the equation (2) for $L = 5$ and $L = 7$, respectively. In particular, in Figure 1(b) we can find some values of $LC = 2$ when the corresponding t-modified sequence is the sequence $\{1010\ldots\}$ or $\{0101\ldots\}$.
- When $2^L - 1$ is not prime we have observed different cases:
  - For t in cosets of cardinal L whose associated polynomial is primitive (that is when $\gcd(2^L - 1, t) = 1$), all the t-modified sequences generated are generalized sequences obtained from different primitive polynomials of degree L. Indeed, the greatest values of LC in Figure 1(c) and Figure 1(d) correspond to the upper bound of equation (2) for $L = 9$ and $L = 11$, respectively.
  - For t in cosets of cardinal L whose associated polynomial is irreducible (not primitive), the t-modified sequences generated are not generalized sequences nor necessarily balanced. This case corresponds to the intermediated values of Figure 1(c). However, the balanced ones have relatively high LC compared with their periods. These sequences are cryptographically interesting.
  - For t in cosets where the cardinal is a proper divisor of L, the produced sequences are generalized sequences with low LC as long as the associated polynomials are primitive. This case corresponds to the lowest values of Figure 1(c).

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50

Table 2. t-modified sequences for $p(x) = x^5 + x^3 + x^2 + x + 1$

t	t-modified sequence	LC
2	1 1 0 1 0 1 1 1 1 0 0 0 0 0 1 0	10
3	0 1 0 1 1 0 1 0 0 1 1 0 0 1 1 0	13
4	1 1 1 1 1 0 1 0 0 0 0 1 0 1 0 0	12
5	0 1 1 1 1 0 0 1 1 0 0 1 1 0 0 0	13
6	0 0 1 1 1 0 1 1 0 0 1 1 0 1 0 0	13
7	0 0 0 1 1 1 1 1 0 1 0 0 1 0 1 0	10
8	1 1 0 0 1 1 0 1 0 0 1 0 1 1 0 0	13
9	0 1 0 0 1 1 0 0 1 1 0 0 1 0 1 1	13
10	1 0 0 1 0 0 1 0 0 1 0 1 1 1 1 0	11
11	1 0 1 1 1 1 0 1 0 0 1 0 0 1 0 0	11
12	1 1 1 1 0 1 0 0 1 0 0 0 1 1 0 0	13
13	0 0 1 0 1 1 0 0 0 1 1 1 0 1 1 0	13
14	1 1 1 1 0 0 0 1 0 0 0 0 1 1 1 0	9
15	1 0 1 1 0 1 0 0 0 0 0 1 1 1 1 0	10
16	1 0 1 1 1 0 0 1 0 0 0 0 1 1 0 1	13
17	1 1 0 0 0 0 1 0 0 0 1 1 1 1 0 1	9
18	0 1 0 1 1 1 1 0 0 0 1 1 1 0 0 0	11
19	1 0 0 1 0 0 1 1 0 0 1 0 0 1 1 1	13
20	0 0 0 0 1 0 1 1 1 1 0 1 0 1 1 0	12
21	0 1 1 1 1 1 0 1 0 0 0 0 1 0 1 0	12
22	1 1 0 1 1 0 0 0 1 1 1 0 0 1 0 0	13
23	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	2
24	1 0 1 0 0 1 0 1 1 1 1 1 0 0 0 0	10
25	0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0	13
26	0 0 1 0 1 0 0 1 1 1 1 1 0 1 0 0	12
27	0 0 1 1 0 0 0 0 1 0 1 1 0 1 1 1	13
28	1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	2
29	0 0 0 0 0 1 1 1 1 0 1 0 1 1 0 1	10
30	1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	2

Example 6. Let us consider the primitive polynomial $p(x) = x^5+x^3+x^2+x+1$. In Table 2 one can find the different t-modified sequences generated by $p(x)$ for different values of t. Since $2^5 - 1$ is prime, all the sequences are generalized sequences produced by other primitive polynomials of degree 5. Indeed, in Example 5, we checked that all the cosets have length 5 and that the associated polynomial to each one is a primitive polynomial of degree 5. ■

Let us consider a more complex example.

Example 7. For $L = 6$ the distribution of cosets can be found in Table 3. Since $2^6 - 1$ is not prime, we have to analyse different cases :

– When t is such that $\gcd(2^6-1, t) = 1$, the corresponding cosets have primitive associated polynomials. In this case these cosets are: $C_1, C_5, C_{11}, C_{13}, C_{23}$ and C_{31}, each one associated to a primitive polynomial of degree 6. When $t \in$

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50

Table 3. Cosets for $L = 6$

Coset	Associated polynomial
$C_1 = \{1, 2, 4, 8, 16, 32\}$	$x^6 + x^5 + x^2 + x + 1$
$C_3 = \{3, 6, 12, 24, 48, 33\}$	$x^6 + x^5 + x^4 + x^2 + 1$
$C_5 = \{5, 10, 20, 40, 17, 34\}$	$x^6 + x^5 + x^3 + x^2 + 1$
$C_9 = \{9, 18, 36\}$	$x^3 + x + 1$
$C_7 = \{7, 14, 28, 56, 49, 35\}$	$x^6 + x^3 + 1$
$C_{11} = \{11, 22, 44, 25, 50, 37\}$	$x^6 + x^5 + 1$
$C_{13} = \{13, 26, 52, 41, 19, 38\}$	$x^6 + x + 1$
$C_{21} = \{21, 42\}$	$x^2 + x + 1$
$C_{15} = \{15, 30, 60, 57, 51, 39\}$	$x^6 + x^4 + x^2 + x + 1$
$C_{23} = \{23, 46, 29, 58, 53, 43\}$	$x^6 + x^4 + x^3 + x + 1$
$C_{27} = \{27, 54, 45\}$	$x^3 + x^2 + 1$
$C_{31} = \{31, 62, 61, 59, 55, 47\}$	$x^6 + x^5 + x^4 + x + 1$

C_i with $i = 1, 5, 11, 13, 23$, the t-modified sequences generated are generalized sequences. For example, for $p(x) = x^6+x+1$ and $t = 5 \in C_5$, we can generate the sequence $\{00100101111010101101110100100001\}$ which is a generalized sequence obtained with polynomial $1 + x + x^2 + x^5 + x^6$.

– For t such that $\gcd(2^6 - 1, t) \neq 1$, we observe two different cases:
  - C_3, C_7, C_{15} have cardinal equal to six and their associated polynomials are irreducible. In this case, the sequences produced are not generalized nor necessarily balanced. For example, for $t = 14 \in C_7$ and the same $p(x)$ considered before, we can generate $\{01000\}$, which is not a generalized sequence neither balanced.
  - C_9, C_{21}, C_{27} have cardinal less than 6 and their associated polynomials are primitive with degree less than 6. In this case, the elements t contained in these cosets produce generalized sequences with low LC. For instance, using $t = 21 \in C_{21}$ we generate the zero sequence and for $t = 27 \in C_{27}$ we can generate the sequence $\{1100\}$. ■

5 Conclusions

In this work, we have proposed a generalized version of the modified self-shrinking generator by using and extended selection rule based on the XORred value of t bits of a PN-sequence. Via the concept of cyclotomic coset, we have classified the generated sequences and analysed their characteristics. Emphasis is on the linear complexity of such sequences. For some values of t, the t-modified sequences coincide with the sequences produced by the generalized self-shrinking generator. Thus, the t-modified self-shrinking generator here proposed provides a large class of sequences most of them with a clear application to stream cipher cryptography.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50

References

1. Golomb, S.W.: Shift Register-Sequences. Aegean Park Press, Laguna Hill, California (1982)
2. Delgado-Mohatar, O., Fúster-Sabater, A.: Software Implementation of Cryptographic Sequence Generators over Extended Fields. Logic Journal of the IGPL **23**(1) (2015) 73–87
3. Paar, C., Pelzl, J.: Understanding Cryptography. Springer, Berlin (2010)
4. Coppersmith, D., Krawczyk, H., Mansour, Y.: The shrinking generator. In Stinson, D., ed.: Advances in Cryptology – CRYPTO '93. Volume 773 of Lecture Notes in Computer Science. Springer-Verlag (1994) 22–39
5. Cardell, S.D., Fúster-Sabater, A.: Modelling the shrinking generator in terms of linear CA. Advances in Mathematics of Communications **10**(4) (2016) 797–809
6. Cardell, S.D., Fúster-Sabater, A., Ranea, A.: Linearity in decimation-based generators: an improved cryptanalysis on the shrinking generator. Open Mathematics **16**(1)(April 2018)
7. Meier, W., Staffelbach, O.: The self-shrinking generator. In De Santis, A., ed.: Advances in Cryptology – EUROCRYPT 1994. Volume 950 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg (1995) 205–214
8. Kanso, A.: Modified self-shrinking generator. Computers and Electrical Engineering **36**(5) (2010) 993–1001
9. Blackburn, S.R., Galbraith, S.: Cryptanalysis of two cryptosystems based on group actions. In Lam, K.Y., Okamoto, E., Xing, C., eds.: Advances in Cryptology – ASIACRYPT '99. Volume 1716 of Lecture Notes in Computer Science. Springer-Verlag, Berlin (1999) 52–61
10. Cardell, S.D., Fúster-Sabater, A.: Recovering the MSS-sequence via CA. Procedia Computer Science **80** (2016) 599–606
11. Hu, Y., Xiao, G.: Generalized self-shrinking generator. IEEE Transactions on Information Theory **50**(4) (2004) 714–719
12. Zhang, Y., Lei, J.G., Zhang, S.P.: A new family of almost difference sets and some necessary conditions. IEEE Transactions on Information Theory **52**(5) (2006) 2052–2061
13. Cardell, S.D., Fúster-Sabater, A.: Discrete linear models for the generalized self-shrunken sequences. Finite Fields and Their Applications **47** (2017) 222–241

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93698-7_50