

A Simple Linearisation of the Self-shrinking Generator

Sara D. Cardell¹(✉) and Amparo Fúster-Sabater²

¹ Departamento de Estadística e Investigación Operativa,
Universidad de Alicante, Ap. Correos 99, E-03080 Alicante, Spain
s.diaz@ua.es

² Instituto de Tecnologías Físicas y de la Información (CSIC),
144, Serrano, 28006 Madrid, Spain
amparo@iec.csic.es

Abstract. Nowadays stream ciphers are the fastest among the encryption procedures, thus they are performed in many practical applications. Irregularly decimated generators are very simple sequence generators to be used as keystream generators in stream ciphers. In this paper, a linearisation method for the self-shrinking generator has been developed. The proposal defines linear structures based on cellular automata (rules 102 or 60) able to generate the self-shrunk sequence. The obtained cellular automata are simple, easy to be implemented and can be extended to other sequence generators in a range of cryptographic interest.

Keywords: Self-shrinking generator · Self-shrunk sequence · Cellular automata · Rule 102 · Rule 60 · Stream cipher · Cryptography

1 Introduction

Symmetric key ciphers are usually split into two large classes: stream and block ciphers depending on whether the encryption function is applied either to each individual bit or to a block of bits, respectively.

At the present moment, stream ciphers are the fastest among the encryption procedures so they are implemented in many technological applications e.g. the encryption algorithm RC4 [1] used in Wired Equivalent Privacy (WEP) as a part of the IEEE 802.11 standards, the encryption function E0 in Bluetooth specifications [2] or the recent proposals HC-128 or Rabbit from the eSTREAM Project [3] that are included in the latest release versions of CyaSSL [4] (open source implementation of the SSL/TLS protocol). In fact, from a short secret key (known only by the two interested parties) and a public algorithm (the sequence generator), stream cipher procedures consist in generating a long sequence, the so-called keystream sequence, of seemingly random bits. For encryption, the sender performs the bit-wise XOR operation among the bits of the plaintext and the keystream sequence. The result is the ciphertext to be sent to the receiver. For decryption, the receiver generates the same keystream, performs the same bit-wise XOR operation between the received ciphertext and the keystream sequence and recovers the original plaintext.

Most keystream generators are based on maximal-length Linear Feedback Shift Registers (LFSRs) [5], that is linear structures characterized by their length (the number of memory cells), their characteristic polynomial (the feedback function) and their initial state (the seed or key of the cryptosystem). Their output sequences, the so-called PN-sequences, are combined in a nonlinear way to break their inherent linearity as well as to produce new pseudorandom sequences of cryptographic application. Combinational generators, nonlinear filters, clock-controlled generators, LFSRs with dynamic feedback or irregularly decimated generators are just some of the most popular keystream sequence generators found in the literature [6, 7].

Irregularly decimated generators produce good cryptographic sequences [8] characterized by long periods, good self-correlation, excellent run distribution, balancedness, simplicity of implementation, etc. 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 in the cryptographic procedure. A well known design in the class of irregularly decimated generators is the self-shrinking generator proposed by Meier and Staffelbach [9] that includes only one LFSR. A natural extension of this sequence generator is the generalized self-shrinking generator [10] that generates a whole family of cryptographic sequences.

It is a well known fact that some one-dimensional linear cellular automata [11] generate exactly the same PN-sequences as those generated by LFSRs. Therefore, a cellular automata can be considered as an alternative generator to the maximum length LFSRs [12]. Moreover, some keystream generators can be modeled in terms of linear cellular automata. In [13], the authors modeled the self-shrinking generator by using rules 150 and 90. In this work, we model the same generator by using rules 102 and 60. A comparison between both modelings is also provided.

The main contribution of this work is to define one-dimensional linear CA able to generate the self-shrunked sequence. The generation of such a sequence from a linear model simplifies the cryptanalysis of the self-shrinking generator.

2 Fundamentals and Basic Notation

First of all, different features and properties of the two basic structures (self-shrinking generator and linear binary CA) considered in this paper are introduced.

2.1 The Self-shrinking Generator

The **self-shrinking generator** was designed by Meier and Staffelbach [9] for potential use in stream cipher applications. This generator consists of a maximal-length Linear Feedback Shift Register (LFSR) [5] of L stages whose output sequence the PN-sequence $\{a_i\}$ is self-decimated giving rise to the self-shrunked sequence $\{s_j\}$ or output sequence of the self-shrinking generator. This sequence

generator is attractive by its simplicity and easy implementation as it involves a unique LFSR. The decimation rule is very simple; let (a_{2i}, a_{2i+1}) , with $i = 0, 1, 2, \dots$, be pairs of consecutive bits of the PN-sequence, then the self-shrunk sequence $\{s_j\}$ is given by:

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

The key of this generator is the initial state of the LFSR. Period, linear complexity and statistical properties of the self-shrunk sequence [9] are very adequate for their application in stream cipher. In brief, the self-shrinking generator is a simplified version of the shrinking generator, suggested by Coppersmith et al. [14], which satisfies the same decimation rule but includes two maximal-length LFSRs.

2.2 Cellular Automata

A one-dimensional **Cellular Automata** (CA) is a device composed by memory cells whose content (binary in this work) is updated according to a state transition rule that determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighbourhood [11]. In fact, the value of the cell in position i at time $\tau + 1$, notated $x_i^{\tau+1}$, depends on the value of the k neighbour cells at time τ .

The cellular automata considered in this work are **linear** (only XOR operations are used), **regular** (every cell follows the same rule) and **null** (cells with null content are adjacent to extreme cells). In this work, our attention is focused on one-dimensional linear CA with binary contents whose time evolution is determined by two simple linear transition rules: rule 102 and rule 60.

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

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

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

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

Recall that both rules are linear and that just involve the addition of two bits. The numbers 01100110 and 00111100 are the binary representations of 102 and 60, respectively. That is the reason why they are called rule 102 and rule 60.

In Fig. 1, we can see these rules using the notation introduced by S. Wolfram [15], where 1 is represented by a black square and 0 is represented by a white square.

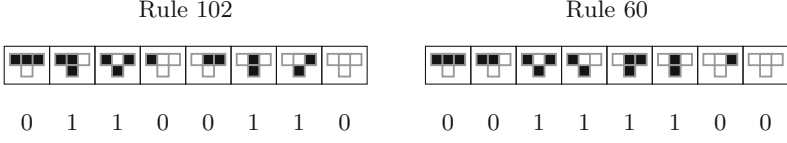


Fig. 1. Rules 102 and 60 depicted in Wolfram's notation

3 The Self-shrinking Generator in Terms of Linear CA

In this section, we propose a family of uniform, null, linear CA that generate the self-shrunked sequence. The following facts characterize the linearisation of the shrunked sequence in terms of CA with rules 102 or 60.

Fact 1. Given a positive integer t , the polynomial $p_t(x)$ is defined $p_t(x) = (1+x)^t$ where $p_t(x) = (1+x)p_{t-1}(x)$.

Fact 2. The characteristic polynomial of the shrunked sequence generated by a maximal-length LFSR of length L is [16]:

$$p_n(x) = (1+x)^n, \quad 2^{L-2} < n \leq 2^{L-1} - (L-2).$$

Fact 3. If the characteristic polynomial of a binary sequence $\{s_i\}$ is $p_n(x)$, then the characteristic polynomial of the sequence $\{u_i\} = \{s_i + s_{i+1}\}$ is $p_{n-1}(x)$.

Fact 4. According to the previous fact, if the first column of a linear CA with rule 102 and length n is the shrunked sequence, then the successive columns of CA will be sequences with characteristic polynomials $p_{n-1}(x), p_{n-2}(x), \dots, p_2(x), p_1(x)$, respectively, where $p_1(x)$ corresponds to the identically 1 sequence.

Fact 5. A uniform, null, linear CA of length n whose first column is the shrunked sequence defined in fact 2 will generate:

- $n - 2^{L-2}$ sequences of period 2^{L-1} ,
- 2^{i-1} sequences of period 2^i , for $1 \leq i \leq L-2$,
- one sequence of period 1 (the identically 1 sequence).

It is worth noticing that the previous facts also hold for uniform, null, linear CA with rule 60. In this case, the CA provides the same sequences but they are obtained in reverse order. The previous results are illustrated in the following example.

Example 1. Given an LFSR with characteristic polynomial $p(x) = 1+x^3+x^4$ and initial state 1001, the self-shrunked sequence obtained is 01001011, with period 2^3 . The characteristic polynomial of this sequence is $p_5(x) = (1+x)^5$. In Table 1, a one-dimensional, uniform, null, linear CA with rule 102 and length 5 is given. Starting at initial state 01011, this CA generates the self-shrunked sequence, in bold, at the first column. It is easy to check that the characteristic polynomials of the remaining sequences are $p_4(x), p_3(x), p_2(x)$ and $p_1(x)$, respectively.

If we consider the same CA of length 5 with rule 60 starting at the symmetric initial state 11010, then the output sequences are the same but they appear in reverse order. See Table 2.

Table 1. 102 CA generating the self-shrunken in Example 1

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

Table 2. 60 CA generating the self-shrunken in Example 1

60	60	60	60	60
1	1	0	1	0
1	0	1	1	1
1	1	1	0	0
1	0	0	1	0
1	1	0	1	1
1	0	1	1	0
1	1	1	0	1
1	0	0	1	1

4 90/150 CA Versus 102/60 CA

Now, we compare the linearisation of the self-shrunken sequence in terms of 102/60 CA with that of [13] carried out in terms of 90/150 CA. In [13], the authors proposed CA based on rules 90/150 that generate the self-shrunken sequence. In fact, the rules 90 and 150 can be defined as follows:

$$\textbf{Rule 90: } x_i^{\tau+1} = x_{i-1}^{\tau} + x_{i+1}^{\tau}$$

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

$$\textbf{Rule 150: } x_i^{\tau+1} = x_{i-1}^{\tau} + x_i^{\tau} + x_{i+1}^{\tau}$$

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

As before, the numbers 01011010 and 10010110 are the binary representations of 90 and 150, respectively.

90/150 CA generating the self-shrunk sequence had a defined structure: rule 90 in extreme cells and rule 150 in the intermediate cells. The length of this CA equals the period of the self-shrunk sequence, 2^{L-1} . In Table 3, the same self-shrunk sequence as that of Example 1, in bold at the first column, is generated by means of these rules. See references [17, 18] for a more detailed description.

Table 3. 90/150 CA generating the self-shrunk in Example 1

90	150	150	150	150	150	150	90
0	1	1	1	0	0	0	1
1	0	1	0	1	0	1	0
0	0	1	0	1	0	1	1
0	1	1	0	1	0	0	1
1	0	0	0	1	1	1	0
0	1	0	1	0	1	0	1
1	1	0	1	0	1	0	0
1	0	0	1	0	1	1	0

In this work, 102/60 CA generating the self-shrunk sequence have a well defined structure too. At the last column the sequence of 1s appears. Besides, there is always a sequence of period 2 (0101... or 1010...). Next, there are 2 sequences of period 4, 4 sequences of period 8 and so on, until we find 2^{L-3} sequences with period 2^{L-2} . The remaining sequences (the length of the CA minus 2^{L-2}) have period 2^{L-1} , including the self-shrunk sequence.

On the other hand, we know that the linear complexity n of the self-shrunk sequence satisfies $2^{L-2} < n \leq 2^{L-1} - (L - 2)$. Hence the length of these CA, n , is less than 2^{L-1} , the length of the CA proposed in [13]. For example, in order to model the self-shrunk sequence in Example 1, we need CA of length 5 (see Tables 1 and 2). If we use the CA proposed in [13], then we need CA of length 8. This difference is more remarkable as far as the length L of the maximal-length LFSR increases.

5 Application of the CA to the Self-shrinking Generators Cryptanalysis

Assume that n is the linear complexity of the self-shrunk sequence. Given $2n$ intercepted bits, it is possible to recover the characteristic polynomial of the maximal-length LFSR that generates the sequence by means of the Berlekamp-Massey algorithm [19].

In our case, we know that there exist CA that generate the self-shrunk sequence as well as that the last sequence for rule 102 is the identically 1 sequence (the first sequence for rule 60 is the identically 1 sequence). Therefore, it is enough to know $n - 1$ bits of the self-shrunk sequence to recover the initial state of the CA and thus, to recover the whole sequence. Notice that this amount is half the needed bits to apply the Berlekamp-Massey algorithm so that the required amount of intercepted bits is reduced by a factor 2.

In Example 1, the self-shrunk sequence had period 8 and linear complexity 5. In Table 4, we can see that intercepting 4 bits of the self-shrunk sequence, we can determine the initial state of the CA and, consequently, the whole self-shrunk sequence.

Table 4. Necessary bits to recover the initial state

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

6 Conclusions

In this work, it is shown that the sequences generated by self-shrinking generators are output sequences of one-dimensional, linear, regular and null cellular automata based on rules 102 and 60. In fact, the linearisation procedure to convert a given self-shrinking generator into the linear cellular model here proposed is quite immediate. It must be noticed that, although the self-shrunk sequences come from PN-sequences irregularly decimated, in practice they can be modeled by means of linear structures. This fact establishes a subtle link between irregular decimation and linearity that as it is shown can be conveniently exploited in cryptanalysis.

A natural extension of this work is the generalization of this procedure to many other cryptographic sequences: (a) The sequences generated by the shrinking generator and the generalized self-shrinking generator as more simple examples of decimation-based keystream generators. (b) The so-called interleaved sequences, as they present very similar structural properties to those of the sequences obtained from irregular decimation generators.

Acknowledgments. The work of the first author was partially supported by Generalitat Valenciana (Spain) with reference APOSTD/2013/081 and by FAPESP with number of process 2015/07246-0. The work of the second author was supported by Ministerio de Ciencia e Innovación (Spain) under Project TIN2014-55325C2-1-R and by Comunidad de Madrid (Spain) under Project CIBERDINE, S2013/ICE3095-CM.

References

1. Paul, G., Maitra, S.: RC4 Stream Cipher and Its Variants. Discrete Mathematics and Its Applications. CRC Press, Taylor & Francis Group, Boca Raton (2012)
2. Bluetooth, Specifications of the Bluetooth system, Version 1.1. <http://www.bluetooth.com/>
3. eSTREAM, the ECRYPT Stream Cipher Project, Call for Primitives. <http://www.ecrypt.eu.org/stream/>
4. Yet Another SSL (YASSL). <http://www.yassl.com>
5. Golomb, S.W.: Shift Register-Sequences. Aegean Park Press, Laguna Hill (1982)
6. Menezes, A.J., et al.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1997)
7. Peinado, A., Fúster-Sabater, A.: Generation of pseudorandom binary sequences by means of LFSRs with dynamic feedback. Math. Comput. Model. **57**(11–12), 2596–2604 (2013)
8. Fúster-Sabater, A.: Linear solutions for irregularly decimated generators of cryptographic sequences. Int. J. Nonlinear Sci. Numer. Simul. **15**(6), 377–385 (2014)
9. Meier, W., Staffelbach, O.: The self-shrinking generator. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 205–214. Springer, Heidelberg (1995)
10. Hu, Y., Xiao, G.: Generalized self-shrinking generator. IEEE Trans. Inf. Theory **50**(4), 714–719 (2004)
11. Das, A.K., Ganguly, A., Dasgupta, A., Bhawmik, S., Chaudhuri, P.P.: Efficient characterisation of cellular automata. IEE Proc. E: Comput. Digit. Tech. **137**(1), 81–87 (1990)
12. Fúster-Sabater, A., Caballero-Gil, P.: Linear solutions for cryptographic nonlinear sequence generators. Phys. Lett. A **369**, 432–437 (2007)
13. Fúster-Sabater, A., Pazo-Robles, M.E., Caballero-Gil, P.: A simple linearization of the self-shrinking generator by means of cellular automata. Neural Netw. **23**(3), 461–464 (2010)
14. Coppersmith, D., Krawczyk, H., Mansour, Y.: The shrinking generator. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 22–39. Springer, Heidelberg (1994)
15. Wolfram, S.: Cellular automata as simple self-organizing system. Caltech preprint CALT 68–938 (1982)
16. Blackburn, S.R.: The linear complexity of the self-shrinking generator. IEEE Trans. Inf. Theory **45**(6), 2073–2077 (1999)
17. Fúster-Sabater, A., Caballero-Gil, P.: Strategic attack on the shrinking generator. Theoret. Comput. Sci. **409**(3), 530–536 (2008)
18. Caballero-Gil, P., Fúster-Sabater, A., Pazo-Robles, M.E.: Using linear equations to model nonlinear cryptographic sequences. Int. J. nonlinear Sci. Numer. Simul. **11**(3), 165–172 (2010)
19. Massey, J.L.: Shift-register synthesis and BCH decoding. IEEE Trans. Inf. Theory **15**(1), 122–127 (1969)

Computer Aided Systems Theory – EUROCAST 2015
15th International Conference, Las Palmas de Gran
Canaria, Spain, February 8-13, 2015, Revised Selected
Papers

Moreno-Díaz, R.; Pichler, F.; Quesada-Arencibia, A.
(Eds.)

2015, XVIII, 887 p. 351 illus. in color., Softcover
ISBN: 978-3-319-27339-6