Randomness study of the concatenation of generalized sequences

SARA D. CARDELL[*], *Centro de Matemática, Computação e Cognição, UFABC, 09210-580 Santo André, Brazil.*

AMALIA B. ORÚE[**], *Facultad de Ciencias y Tecnología, Universidad Isabel I, 09003 Burgos, Spain.*

VERÓNICA REQUENA[†], *Departamento de Matemáticas, Universidad de Alicante, 03690 Alicante, Spain.*

AMPARO FÚSTER-SABATER[††], *Instituto de Tecnologías Físicas y de la Información, CSIC, 28006 Madrid, Spain.*

Abstract

Keystream sequences should look as random as possible, i.e. should present no logical pattern to be exploited in cryptographic attacks. The generalized self-shrinking generator, a sequence generator based on irregular decimation, produces a family of sequences with good cryptographic properties. In this work, we display a detailed analysis on the randomness of the sequences resulting from the concatenation of elements of this family. We apply the most important batteries of statistical and graphical tests providing powerful results and a new method to construct sequences with good cryptographic properties.

Keywords: Generalized self-shrinking generator, pseudo-random number generator, randomness.

1 Introduction

A random number sequence generator (RNG) produces a sequence of numbers that cannot be reasonably predicted better than by a random chance. There exist two types of RNGs: hardware random number generators (HRNGs), which generate genuinely random numbers, or pseudo-random number generators (PRNGs), whose output must be unpredictable in the absence of knowledge of the inputs. We focus our attention on the latter generators. Both types produce streams of bits that may be divided into sub-streams or blocks of random numbers. The cryptographic quality of pseudo-random sequences is determined by different factors: unpredictability, long periods, large key space, etc.

Irregular decimation is a very habitual technique to produce pseudo-random sequences with cryptographic applications [4, 6, 9, 15, 16, 23]. In practice, the underlying idea of these generators is the irregular decimation of a PN-sequence produced by an LFSR [13]. In this paper, we work with

[*] E-mail: s.cardell@ufabc.edu.br

[**] E-mail: amaliabeatriz.orue@ui1.es

[†] E-mail: vrequena@ua.es

[††] E-mail: amparo@iec.csic.es

Advance Access published 18 February 2022 https://doi.org/10.1093/jigpal/jzac008

the most representative generator in this family of irregular decimation-based sequence generators, that is the generalized self-shrinking generator [15].

The generalized self-shrinking generator [15] consists of one LFSR [13] whose output PN-sequence is self-decimated coming up with a family of new sequences (GSS-sequences) with good cryptographic properties. This generator is fast, easy to implement and produces a set of sequences with good cryptographic properties. In [5], the authors presented a statistical and graphical study of the randomness of the GSS-sequences whose results imply the suitability of this family of sequences for cryptographic applications. As a consequence, this generator seems adequate for its use in lightweight cryptography and low-cost applications.

A huge number of attacks to PRNGs achieve success due to their lack of randomness [18, 19], this means that the quality of the randomness of the pseudo-random generators is very important for the security of many cryptographic schemes. Nowadays, there exists a huge number of statistical tests to determine if a sequence can be considered sufficiently random and secure in cryptographic terms [14]. However, it is difficult to choose a certain number of these tests to determine if the randomness analysis of the sequences generated is satisfactory.

In this work, we use different tools, both statistical and graphic, in the study of the randomness of the concatenation of GSS-sequences, with the purpose of providing long pseudo-random sequences with a low computational cost and good cryptographic properties.

The paper is organized as follows: in Section 2, we present some necessary concepts to understand the rest of the paper. In Section 3, we present some graphic and statistical tests in order to analyse the randomness of the concatenation of GSS-sequences. Finally, the paper concludes with Section 4, where we present some conclusions and future work.

2 Preliminaries

Consider $\mathbb{F}_2$ the Galois field of two elements and $\{a_i\}_{i\geq 0} = \{a_0, a_1, a_2 \ldots\}$ a binary sequence with $a_i \in \mathbb{F}_2$, for $i = 0, 1, 2, \ldots$. We say that a sequence $\{a_i\}_{i\geq 0}$ is periodic if there exists an integer T, called period, such that $a_{i+T} = a_i$, for all $i \geq 0$. From now on, all the sequences considered will be binary sequences and the symbol $+$ will denote the Exclusive-OR (XOR) logic operation.

Let r be a positive integer, and let $d_1, d_2, d_3, \ldots, d_r$ be constant coefficients with $d_j \in \mathbb{F}_2$. A binary sequence $\{a_i\}_{i\geq 0}$ satisfying the relation

$$a_{i+r} = d_r a_i + d_{r-1} a_{i+1} + \cdots + d_3 a_{i+r-3} + d_2 a_{i+r-2} + d_1 a_{i+r-1}, \quad i \geq 0,$$

is called a (r-th order) **linear recurring sequence** in $\mathbb{F}_2$. The terms $\{a_0, a_1, \ldots, a_{r-1}\}$ are referred to as the initial terms and define the construction of the sequence uniquely. Furthermore, the monic polynomial: $p(x) = d_r + d_{r-1}x + \cdots + d_3 x^{r-3} + d_2 x^{r-2} + d_1 x^{r-1} + x^r \in \mathbb{F}_2[x]$ is called the **characteristic polynomial** of the linear recurring sequence and $\{a_i\}_{i\geq 0}$ is said to be generated by $p(x)$.

We can generate linear recurring sequences by means of **Linear Feedback Shift Registers** (LFSRs) [13]. An LFSR is defined as an electronic device with r memory cells (stages) with binary content. At every clock pulse, the element of each stage is shifted to the adjacent stage and a new element is computed through the linear feedback to fill the empty stage (see Figure 1). We say that the LFSR has maximal-length if its characteristic polynomial is primitive. Its output sequence is called PN-sequence (Pseudo-Noise sequence) and has period $T = 2^r - 1$ [13].

Let $\{a_i\}_{i\geq 0}$ be a PN-sequence produced by a maximal-length LFSR with L stages. Let $G = [g_0, g_1, g_2, ..., g_{L-1}] \in \mathbb{F}_2^L$ be an L-dimensional binary vector and $\{v_i\}_{i\geq 0}$ a sequence defined as:

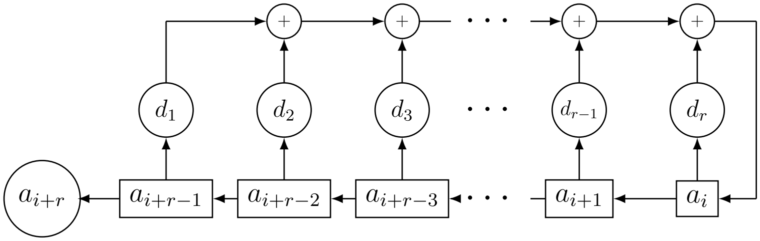


FIGURE 1. LFSR of length r.

$v_i = g_0a_i + g_1a_{i-1} + g_2a_{i-2} + \cdots + g_{L-1}a_{i-L+1}$. For $i \geq 0$, we define the decimation rule as follows:

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

The generator defined previously is the **generalized self-shrinking generator** and the output sequence $\{s_j\}_{j\geq 0}$, denoted by $s(G)$, is called the **generalized self-shrunken sequence** (GSS-sequence) associated with G. When G ranges over $\mathbb{F}_2^L$, then the resulting $\{v_i\}$ sequences correspond to the $2^L - 1$ possible shifts versions of $\{a_i\}$ (see [7, Theorem 2]). Besides, we obtain *the family of generalized self-shrunken sequences* based on the PN-sequence $\{a_i\}_{i\geq 0}$ denoted by $S(a) = \{s(G) \mid G \in \mathbb{F}_2^L\}$.

The **GSS-concatenated sequence** of the family $S(a)$, denoted by $C(S(a))$, is the resulting sequence of concatenating all the sequences in the family except for the trivial ones ({0 0 0 0 . . .}, {1 1 1 1 . . .}, {0 1 0 1 . . .}, and {1 0 1 0 . . .}).

EXAMPLE 2.1
Consider the primitive polynomial $p(x) = x^3 + x + 1$ and the corresponding PN-sequence $\{a_i\}_{i\geq 0} =$ {1 1 1 0 0 1 0}. We can construct the GSS-sequences shown in Table 1. The underlined bits in the different sequences $\{v_i\}_{i\geq 0}$ are the digits of the corresponding $\{s(G)\}$ sequences. The PN-sequence $\{a_i\}_{i\geq 0}$ is written at the bottom. The corresponding GSS-concatenated sequence of this family is $C(S(a)) =$ {0 1 1 0 1 1 0 0 1 0 0 1 0 0 1 1}.

3 Statistical randomness analysis

Statistical randomness tests are designed to analyse the quality of random number generators and considered as an important part of evaluating security of cryptographic algorithms.

In [5], the authors give an exhaustive analysis of randomness of all family of GSS-sequences generated from PN-sequences associated to characteristic polynomials of degree up to 27, obtaining powerful results. Following the previous work, one might wonder if the concatenation of the sequences obtained into a family of GSS-sequences have the same good properties of randomness than individual sequences. With the concatenation of this family of good crytographic sequences, we can obtain pseudo-random sequences longer and which would require less computational cost.

With this aim, we present a statistical randomness analysis from two points of view. On the one hand, we provide different graphical tools based on chaotic cryptographic (see [10, 24]). On the other hand, we use the most powerful batteries of statistical tests as the Diehard battery of tests,

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}	

the Lempel–Ziv Compression Test and the packet FIPS 140-2, provided by the National Institute of Standards and Technology (NIST), among others. All the tests are applicable for a wide range of binary string size and thus exhibit considerable flexibility.

For this study, we generate, from MATLAB R2020b, families of GSS-sequences from PN-sequences coming from maximal-length LFSRs with characteristic polynomials of degree less than or equal to 16. The results presented are based on GSS-concatenated sequences with the characteristic polynomial $p(x) = x^{16} + x^{14} + x^{12} + x + 1$ and whose initial state is the identically 1 vector of length 16. The tests were performed with 2^{29} bit sequences. Most of the tests work associating every eight bits in an octet, obtaining sequences of 2^{26} samples of 8 bits; with the exception of the Linear complexity test that works with just one bit and the Chaos game that works associating the bits two by two.

3.1 Graphic tests

Next, we show some of the main graphic techniques used for visualizing randomness of sequences. These graphic tools are usually used in chaotic dynamic system analysis, and its applicability in the cryptographic study of pseudo-random sequences has been proven in [1, 10, 24]. Specifically, we apply the return map, the chaos game and the linear complexity or the Lyapunov exponent among others.

Return map

The return application, introduced in [10], is a tool that allows to detect the existence of some useful information about the parameters of the system used to generate the sequence.

It consists of drawing a two-dimensional graph of the points of the sequence x_t as a function of x_{t-1} and, can help to reconstruct the value of the parameters of a pseudo-random sequence. The resulting graph must be a cloud of points where we cannot guess no trend, no figure, no line, no symmetry, no pattern.

Figure 2 shows return map of a GSS-concatenated sequence as a disordered cloud which does not provide any useful information for its cryptanalysis.

Linear Complexity

The linear complexity (LC) of a sequence is defined as the length of the shortest LFSR that generates it. If the characteristic polynomial of the linear recurring sequence is primitive [13], then the LFSR is a maximal-length LFSR and its output sequence has period $T = 2^L - 1$, where L is the degree of the characteristic polynomial.

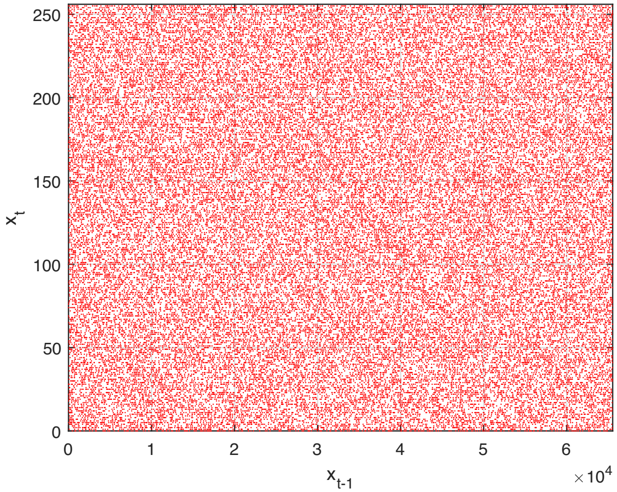


FIGURE 2. Return map of GSS-concatenated sequence of 2^{29} bits.

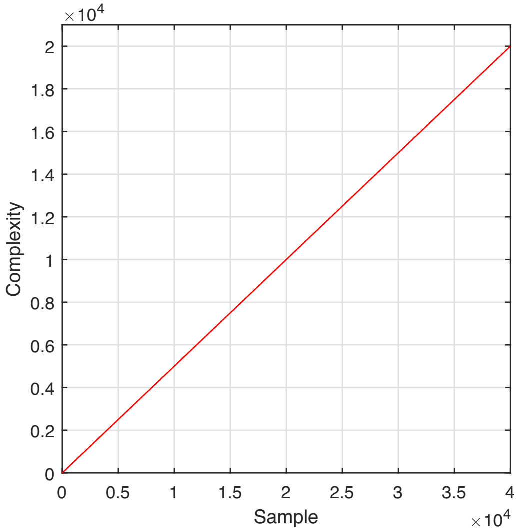


FIGURE 3. Linear complexity of GSS-concatenated sequence of 2^{29} bits.

LC is used as a measure of the unpredictability of a pseudo-random sequence and a much used metric of the security of a keystream sequence [25]; it must be as large as possible, that is, its value has to be very close to half the period [27], $LC \simeq T/2$. We use the Berlekamp–Massey algorithm [22] to compute this parameter. From Figure 3, we have that the value of the linear complexity of the first 40000 bits of the sequence is just half the length, 20000.

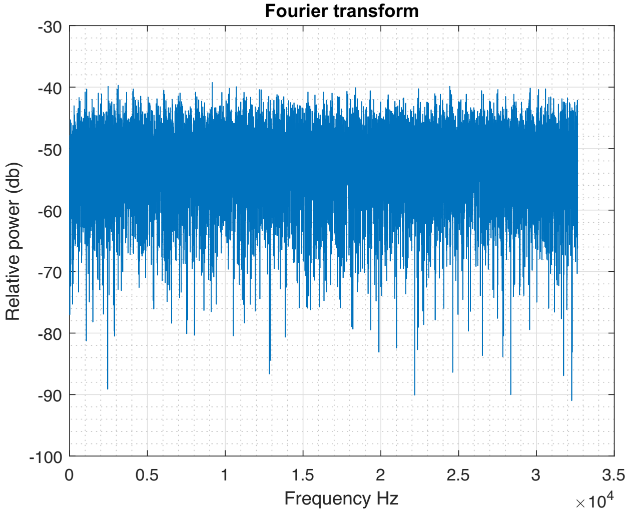


FIGURE 4. Fast Fourier transform of GSS-concatenated sequence.

Fast Fourier transform

The goal of fast Fourier transform test consists of detecting repetitive patterns in the sequence analysed, which would indicate a deviation from the assumption of randomness.

If all the maximum harmonics of fast Fourier transform have approximately the same horizontal level without up or down trend, then the sequence can be considered random.

In Figure 4, we obtain that all values are included in the same range, which means that the test is passed.

Distribution of identical samples

One important property of random sequence is the distance of occurrence between samples of equal value. The most probable distance between two identical samples of a perfect sequence is zero. If this distance increases, then the probability of coincidence between them decreases.

In Figure 5 we observe that the distribution of samples of a GSS-concatenated sequence is close to the ideal.

Lyapunov exponent

Lyapunov exponent provides a quantitative measurement of divergence or convergence of nearby trajectories.

In [20, 24], they present a definition of Lyapunov Hamming exponent in bits as follows:

$$LHE = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} d_{1H}$$

with N the number of iterations and d_{1H} is the Hamming distance.

In cryptology, this value indicates the number of bits that changes in a word. If two numbers are identical, then its LHE value will be 0. Nevertheless, if all the bits of both numbers are different, then its LHE will be $LHE = \log_2\ m = \log_2 2^n = n$, where n is the number of bits with which the numbers are encoded. The best value will be $n/2$.

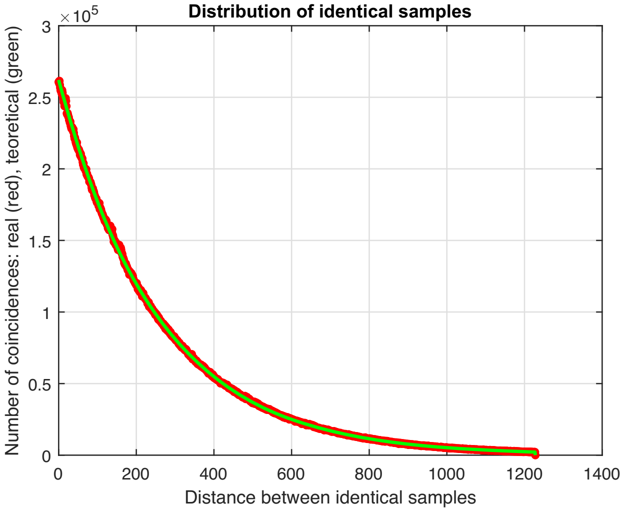


FIGURE 5. Distribution of samples of a GSS-concatenated.

All Lyapunov exponent estimates were close to the maximum value 4. We show the average of the values obtained of the sequences analysed:

$$\begin{aligned}
&\text{Lyapunov Hamming exponent, ideal} = 4\\
&\text{Lyapunov Hamming exponent, real} \quad = 3.999\\
&\text{Absolute desviation from ideal} \qquad = -0.00025106
\end{aligned}$$

hence, the proposed generator passes perfectly this test.

Samples in increasing order

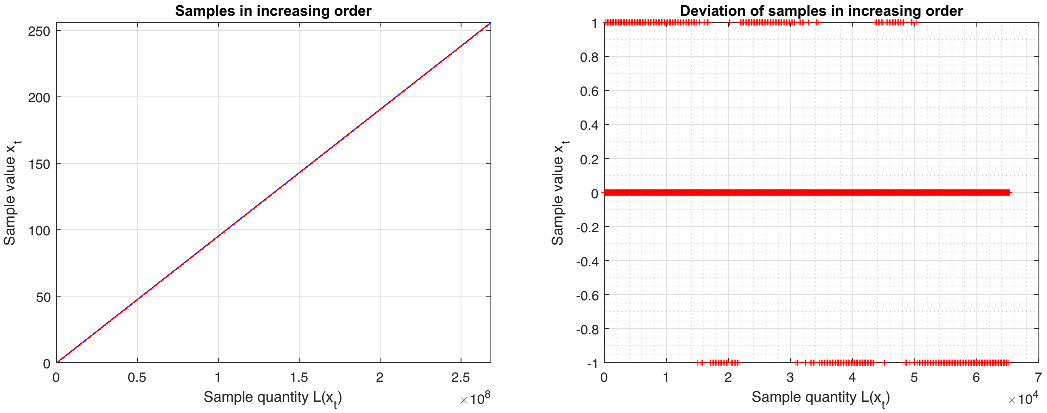
The samples of 8 bits are ordered by increasing value and are represented by a graph.

This representation means that all the numbers are generated (if it is a continuous straight line (red)) and that the density is uniform (if its inclination is 45 degrees). In Figure 6(a), we observe that the samples are perfectly represented by a continuous straight line with the perfect inclination of 45 degrees.

From Figure 6(b), the deviation between the increasing samples is analysed and the values $-1, 0$ or 1 are obtained.

Chaos game

Chaos game can be described mathematically by an iterated functions system (IFS) [2, 10, 26] and through which the transition to chaos associated with fractals can be studied. The result of chaos game is called **attractor**, and not always is a fractal, it may be any compact set. If the output is a graph with fractals or patterns, then it means that the sequence cannot been considered random. In Figure 7, we cannot observe any pattern or fractal, it is an unordered cloud of points which implies good randomness.



(a) Samples in increasing order.

(b) Deviation from increasing order.

FIGURE 6. Samples ordered by increasing value.

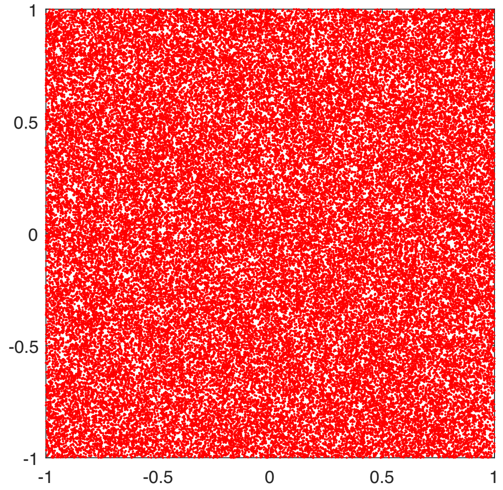


FIGURE 7. GSS-concatenated sequence Chaos game.

3.2 *Statistical batteries of tests*

Diehard Battery of Tests

Diehard battery of tests [21] is a reliable standard for evaluating randomness of sequences of PRNGs. If the sequences analysed do not pass this battery of statistical tests, then we can consider that they are not suitable for cryptographic applications.

Diehard battery consists of 15 different independent statistical tests, some of them repeated but with different parameters. These statistical tests are designed to test the null hypothesis H_0 which

TABLE 2. Diehard battery of tests results for a GSS-concatenated sequence of degree 16.

Test name	p-value or KS p-value[1]	Result
Birthday spacing	0.039711	PASS
Binary ranks (31 × 31)	0.320880	PASS
Binary ranks (32 × 32)	0.979186	PASS
Binary ranks (6 × 8)	0.407298	PASS
Parking lot	0.985256	PASS
Overlapping	0.115366	PASS
permutations	0.982348	
Minimum distance	0.985489	PASS
3D spheres	0.815294	PASS
Squeeze	0.793474	PASS
Overlapping sums	0.989556	PASS
	0.080534	
Runs	0.208853	
	0.589281	PASS
	0.970691	
Craps	0.623129	PASS
	0.392400	
Bit stream (Monkey tests)	[1]	PASS
OPSO	[1]	PASS
OQSO	[1]	PASS
DNA	[1]	PASS
COUNT-THE-1's (specific bytes)	[1]	PASS

All the p-values obtained in the corresponding tests evaluated are in the range $(0, 1)$.

states that the input sequence is randomly generated. If the hypothesis is not rejected in all the tests, then it is implied that the input sequences are random.

Most of the tests in DIEHARD return a p-value or the KS p-value (given by the Kolmogorov-Smirnov test), which should be uniform on [0,1) if the input file contains truly independent random bits. It is considered that a bit stream really fails when it is gotten p-values of 0 or 1 to six or more places.

Hundreds of GSS-concatenated sequences of 2^{29} bits have passed correctly the great majority of tests in the Diehard battery of Marsaglia with good results. In Table 2 we show the results obtained with the Diehard battery from a GSS-concatenated sequence with characteristic polynomial of degree 16.

FIPS test 140-2

The FIPS 140-2 test [11], issued by the American National Institute of Standards and Technology (NIST), has been widely used for the verification the statistical properties of the randomness of the pseudo-random numbers generated by PRNGs. In this package there are 4 statistical random number generator tests: The Monobit Test, The Poker Test, The Runs Test and The Long Runs Test. The proposed GSS-concatenated sequences with characteristic polynomials of degree ≤ 16 pass all these tests with perfect results.

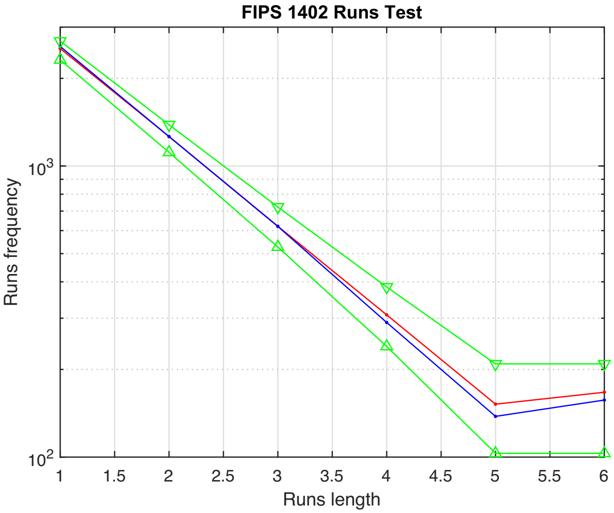


FIGURE 8. Run test for a GSS-concatenated sequence with characteristic polynomial of degree 16.

From Figure 8, we can show a graphic result of the Runs Test, for a GSS-concatenated sequence of degree 16. The test is passed if the runs (for both the runs of zeros, red line, and the runs of ones, blue line) that occur (of lengths 1 through 6) are each within the corresponding interval specified by the green line. Observe that the test is passed since they all the runs fall within the corresponding range specified by the green line.

Lempel–Ziv Compression Test

The focus of this test is the number of cumulatively distinct patterns (words) in the sequence. The purpose of the test is to determine how far the tested sequence can be compressed. The sequence is considered to be non-random if it can be significantly compressed. A random sequence will have a characteristic number of distinct patterns.

The proposed GSS-concatenated sequences with characteristic polynomials of degree ≤ 16, pass this test with perfect results.

Ziv–Lempel Test is passed with a p-value$>= 0.01$. For more than hundreds of GSS-concatenated sequences analysed we have obtain p-values ≥ 0.01.

Maurer's 'Universal Statistical' Test

The focus of this test is the number of bits between matching patterns (a measure that is related to the length of a compressed sequence). The purpose of the test is to detect whether or not the sequence can be significantly compressed without loss of information. A significantly compressible sequence is considered to be non-random.

If the computed p-value is < 0.01, then conclude that the sequence is non-random. Otherwise, conclude that the sequence is random. For more than one thousand of GSS-concatenated sequences analysed, we have obtain p-values ≥ 0.01.

Although statistical and graphical tests analyse in deep the randomness of the sequences, such tests do not provide information on the weaknesses of the generators against cryptanalytic attacks, e.g. linear cryptanalysis. Indeed, the potential linear relationship between the key bits (initial state) and the generated bits has been already studied in the literature on the whole family of decimation-based generators. Most of these cryptanalytic techniques deal with linearization procedures whose main proposals are listed as follows: (i) The decimated bits can be arranged into interleaved sequences whose characteristics and properties are well-known, see [12] and [6, Chapter 3]. (ii) The decimated bits can be generated by means of linear models based on cellular automata (rule 120 and rule 60), see [8] and [6, Chapter 4]. In both cases, the huge amount of intercepted bits needed to launch a linear cryptanalysis successfully defeats the feasibility of this type of attacks.

4 Conclusions

In this article, we perform a deep study of the randomness of the GSS-concatenated sequences, generated from the family of generalized self-shrunken sequences, $S(a)$, based on PN-sequence $\{a_i\}_{i\geq 0}$. As future work, we would like to apply other powerful statistical tests as CRYPT-X [3] or TestU01 [17]. Furthermore, it would be interesting to study if there exist some relations among the GSS-sequences of a same family and if this fact could be advantageous for launching cryptanalytic attacks. Finally, we would like to analyse and study the interleaving of GSS-sequences of the same family and from different families.

Acknowledgements

This research has been partially supported by the Comunidad de Madrid (Spain) under project CYNAMON (P2018/TCS-4566), also co-funded by Fondo Social Europeo (FSE) and European Union Fondo Europeo de Desarrollo Regional (FEDER) funds. The third author was partially supported by Spanish grant VIGROB-287 of the Universitat d'Alacant.

References

[1] G. Alvarez, F. Montoya, M. Romera and G. Pastor. Cryptanalyzing an improved security modulated chaotic encryption scheme using ciphertext absolute value. *Chaos, Solitons & Fractals*, **23**, 1749–1756, 2005.

[2] M. Barnsley. *Fractals Everywhere*, 2nd edn. Academic Press, 1988.

[3] W. Caelli. Crypt x package documentation *Tech. Rep*. Information Security Research, 1992.

[4] S. D. Cardell and A. Fúster-Sabater. The t-Modified Self-shrinking Generator. In *Computational Science—ICCS 2018*, Y. Shi *et al.*, eds. Lecture Notes in Computer Science, vol. 10860, pp. 653–663. Springer, Wuxi, China, 2018.

[5] S. D. Cardell, V. Requena, A. Fúster-Sabater and A. B. Orúe. Randomness analysis for the generalized self-shrinking sequences. *Symmetry*, **11**, 1460–1486, 2019.

[6] S. Díaz Cardell and A. Fúster-Sabater. *Cryptography With Shrinking Generators: Fundamentals and Applications of Keystream Sequence Generators Based on Irregular Decimation*. Springer Briefs in Mathematics. Springer International Publishing, 2019.

[7] S.D Cardell, J.-J. Climent, A. Fúster-Sabater and V. Requena. Representations of generalized self-shrunken sequences.*Mathematics*, **8**, 1006, 2020.

[8] S.D Cardell, D.F Aranha and A. Fúster-Sabater. Recovering decimation-based cryptographic sequences by means of linear CAs. *Logic Journal of the IGPL*, **28**, 430–448, 2020.

[9] D. Coppersmith, H. Krawczyk and Y. Mansour. The shrinking generator. In *Advances in Cryptology—CRYPTO '93*; D. R. Stinson ed. Lecture Notes in Computer Science, **773**: 22–39. Springer, 1994.

[10] A. B. Orúe, A. Fúster-Sabater, V. Fernández, F. Montoya, L. Hernández and A. Martín. Herramientas gráficas de la criptografía caótica para el análisis de la calidad de secuencias

pseudoaleatorias. In *Actas de la XIV Reunión Española Sobre Criptología y Seguridad de la Información, RECSI XIV*, P. L. F. Gomila and M. F. H. Campos, eds, pp. 180–185. Maó, Menorca, Illes Balears, 26–28 Octubre, 2016.

[11] FIPS 186FIPS 186 Digital signature standard. In *Federal Information Processing Standards Publication 186*. U.S. Department of Commerce. N.I.S.T., National Technical Information Service, Springfield, Virginia.

[12] A. Fúster-Sabater and P. Caballero-Gil. Strategic attack on the shrinking generator. *Theoretical Computer Science*, **409**, 530–536, 2008.

[13] S. W. Golomb. *Shift Register-Sequences*. Aegean Park Press, Laguna Hill, California, USA, 1982.

[14] G. Gong, T. Helleseth, P.V. Kumar and W Solomon. Solomon W. Golomb—Mathematician, Engineer, and Pioneer. *IEEE Transactions on Information Theory*, **64**, 2844–2857, 2018.

[15] Y. Hu and G. Xiao. Generalized self-shrinking generator. *IEEE Transactions on Information Theory*, **50**, 714–719, 2004. doi: 10.1016/j.compeleceng.2010.02.004.

[16] A. Kanso. Modified self-shrinking generator. *Computers & Electrical Engineering*, **36**, 993–1001, 2010.

[17] P. L'Ecuyer. and R. Simard. TestU01: a C library for empirical testing of random number generators. *ACM Transactions on Mathematical Software*, **33**, article: 22, 2007.

[18] C. Li, S. Li, G. Chen and W.A. Halang. Cryptanalysis of an image encryption scheme based on a compound chaotic sequence. *Image and Vision Computing* 2009, **27**, 1035–1039.

[19] C. Li, S. Li and K.-T Lo. Breaking a modified substitution-diffusion image cipher based on chaotic standard and logistic maps. *Communications in Nonlinear Science and Numerical Simulation*, **16**, 837–843.

[20] J. Machicao, J.M. Baetens, A.G. Marco, B. de Baets and O.M. Bruno. A dynamical systems approach to the discrimination of the modes of operation of cryptographic systems. *Communications in Nonlinear Science and Numerical Simulation*, **29**: 102–115, 2015.

[21] G. Marsaglia. *The Marsaglia Random Number CDROM including the Diehard battery of tests of randomness*. Florida State University, 1995. http://webhome.phy.duke.edu/rgb/General/dieharder.php.

[22] J. L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, **15**, 122–127, 1969. doi: 10.1109/TIT.1969.1054260.

[23] W. Meier and O. Staffelbach. The self-shrinking generator. In *Advances in Cryptology—EUROCRYPT'94*; Cachin, C. Camenisch, J., eds.; Lecture Notes in Computer Science, vol. 950, pp. 205–214. Springer, 1994.

[24] A. Orúe and A. B. López. *Contribución al Estudio del Criptoanálisis y Diseño de los Criptosistemas Caóticos*. Tesis Doctoral. Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación, 2013.

[25] C. Paar and J. Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer, Berlin Heidelberg, Germany, 2010.

[26] H. O. Peitgen, H. Jurgens and D. Saupe. *Chaos and Fractals*. Springer, 2004.

[27] R. A. Rueppel. Linear Complexity and Random Sequences. In *Advances in Cryptology—EUROCRYPT' 85*, F. Pichler., ed., Lecture Notes in Computer Science, vol. 219, pp. 167–188. Springer, 1986.

Received 20 February 2021