EPiC Series in Computing

Volume 98, 2024, Pages 140–149

Proceedings of 39th International Conference on Computers and Their Applications

Pseudorandom Binary Generator Based on the Combination of Linear Feedback Shift Registers with Nonlinear Filtering Function

Narayan Debnath[1] Andrés Francisco Farías[2] Andrés Alejandro Farías[2]
Ana Gabriela Garis[3] Daniel Riesco[3] Germán Antonio Montejano[3]
[1]School of Computing and Information Technology Easten International University Vietnam
[2]Academic Department of Physical, Mathematical and Natural Sciences
National University of La Rioja, Argentina
[3]Department of Computer Science, Faculty of Physical-Mathematical and Natural Sciences,
National University of San Luis, Argentina
narayan debnath@eiu.edu.vn, afarias665@yahoo.com.ar,
andres_af86@hotmail.com, agaris@unsl.edu.ar, driesco@unsl.edu.ar,
gmonte@unsl.edu.ar

Abstract

This work develops the procedure for the construction of a pseudorandom binary generator based on a structure made up of two blocks of four Linear Feedback Shift Registers (LFSR) each, to which nonlinear filtering functions are attached, which deliver sequences binaries that are combined using a block containing the combination devices.

The process includes: characteristics of the LFSR, definition of the model, choice of the different LFSR, selection of Boolean functions based on their optimal cryptographic properties, key, composition of the generator with the selected elements, choice of the statistical randomness tests to be used and the criteria to analyze the results, implementation with one hundred different keys and verification of the randomness of the set of sequences obtained.

Keywords: LFSR, generator, key, boolean function, non-linearity.

1 Introduction

Pseudorandom Pseudorandom binary generators must provide quality sequences and must also be unpredictable and easy to implement, but fundamentally they must have a period of significant length.

It is in these terms that a model that responds to such demands is proposed. The chosen modality is based on the non-linear combination of generators formed by two blocks of four LFSRs, each one, which has a non-linear Boolean filtering function coupled [1], [2].

The construction procedure of a pseudorandom generator with these characteristics requires several stages:

- Pseudorandom binary generator
  - Characteristics of the LFSR
  - Definition of the model.
  - Election of the different LFSR, which make up each block.
  - Selection of four-variable Boolean functions based on their optimal cryptographic properties.
  - Key and procedure to generate the initial states of the LFSR.
  - Composition of the generator with the elements already selected.
- Tests of randomness
  - Choice of the statistical tests to use and the criteria for analyzing the results.
  - Put the generator into operation with one hundred different keys and perform the necessary randomness tests on the sequences obtained.

2 Pseudorandom binary generator

2.1 Characteristics of the LFSR

The LFSRs used have the following structure indicated in Figure 1, the LFSR itself has a coupled connection polynomial that generates the linear feedback. The polynomial must be primitive, to achieve the maximum period of the sequence. Additionally, the four-variable Boolean function is attached, which produces non-linear filtering.

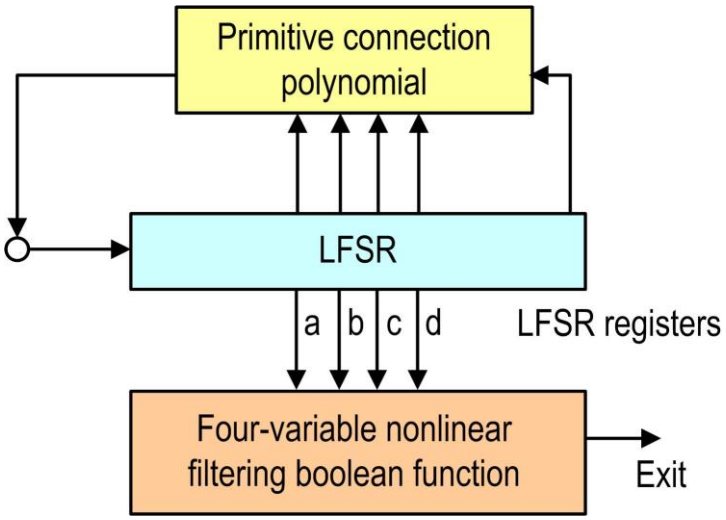

Figure 1: LFSR Scheme

2.2 Schematic definition of the model

The generator is composed of two blocks of four LFSRs that produce eight binary sequences and a block for their combination, which contains two majority functions of three variables and a Boolean function of four variables. Each of the LFSRs has a nonlinear filtering four-variable Boolean function coupled to it. The left block provides four sequences, three are combined by the majority function and their result together with the fourth sequence feed the final join Boolean function. The same goes for the right block. Everything described is visualized in Figure 2:

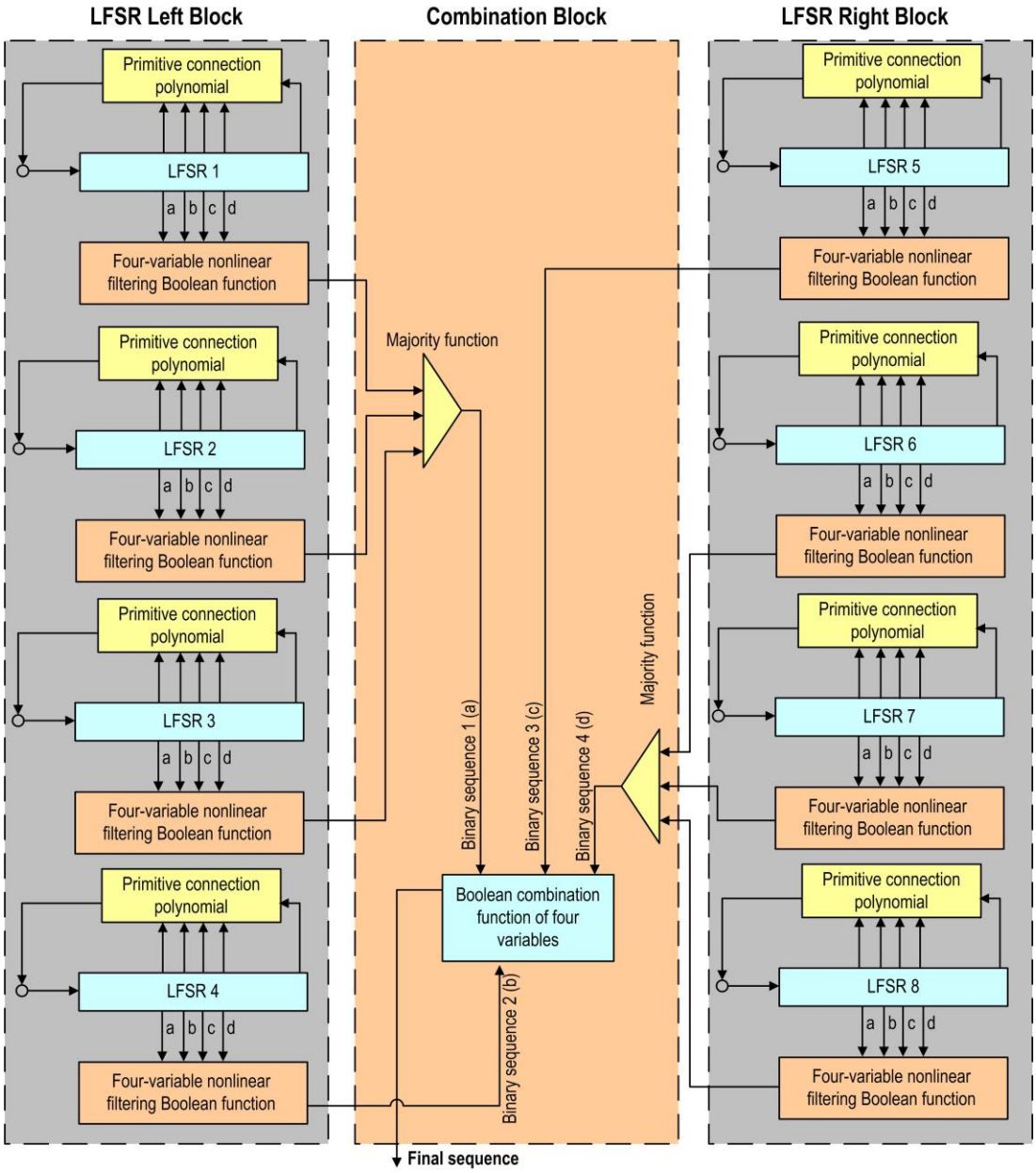

Figure 2: Pseudorandom binary generator scheme

2.3 Choice of the different LFSRs

The lengths and primitive polynomials [3], [4], [5] of the LFSRs that make up each block are those indicated in Table 1:

Block	LFSR	Lengths	Primitive polynomials
Left	1	31	$P(x)_1 = x^{31} + x^{25} + x^{23} + x^8 + 1$
	2	37	$P(x)_2 = x^{37} + x^{22} + x^{14} + x^{25} + 1$
	3	59	$P(x)_3 = x^{59} + x^{54} + x^{46} + x^{26} + 1$
	4	53	$P(x)_4 = x^{53} + x^{50} + x^{41} + x^{20} + 1$
Right	5	59	$P(x)_5 = x^{59} + x^{54} + x^{46} + x^{26} + 1$
	6	47	$P(x)_6 = x^{47} + x^{32} + x^{24} + x^{11} + 1$
	7	53	$P(x)_7 = x^{53} + x^{50} + x^{41} + x^{20} + 1$
	8	61	$P(x)_8 = x^{61} + x^{44} + x^{19} + x^{15} + 1$

Table 1: LFSR, lengths and primitive polynomials of the Generator

2.4 Boolean function selection

Boolean functions of four variables are adopted, both for those that perform non-linear filtering and the one that fulfills the combination task. For the selection, some of the desirable cryptographic properties for these functions are taken into account.

2.5 Desirable cryptographic properties

Below are some of the most cryptographically significant properties, adopted for this work [6], [7], [8]:

- Balanced Function
- High non-linearity
- Meets strict avalanche criteria (SAC)

Following the criteria indicated above, the accepted Boolean functions are shown in Table 2 and Table3:

					LFSR registers			
	f_{NAF}	Balanced	Non-linearity	SAC compliant	a	b	c	d
Block	Nonlinear filtering functions							
Left	$f_{84} = a \cdot c \oplus b \cdot c \oplus a \cdot d \oplus b \cdot d \oplus c \cdot d$	yes	4	yes	3	17	23	26
	$f_{89} = a \cdot c \oplus b \cdot c \oplus d \oplus a \cdot d \oplus b \cdot d$	yes	4	yes	1	17	23	35
	$f_{2402} = b \oplus a \cdot c \oplus b \cdot c \oplus d \oplus c \cdot d$	yes	4	yes	1	6	12	28
	$f_{3338} = a \oplus a \cdot b \oplus c \oplus b \cdot c \oplus b \cdot d$	yes	4	yes	1	7	13	27
Right	$f_{4722} = a \oplus b \oplus a \cdot c \oplus b \cdot c \oplus c \cdot d$	yes	4	yes	6	14	27	34
	$f_{4393} = a \oplus c \oplus a \cdot d \oplus b \cdot d \oplus c \cdot d$	yes	4	yes	6	16	25	31
	$f_{3981} = a \oplus a \cdot c \oplus b \cdot c \oplus d \oplus c \cdot d$	yes	4	yes	3	10	16	26
	$f_{3672} = a \oplus a \cdot b \oplus c \oplus a \cdot c \oplus a \cdot d$	yes	4	yes	3	9	18	25

Table 2: Boolean functions of four variables adopted for nonlinear filtering

					Binary sequences			
	f_{NAF}	Balanced	Non-linearitv	SAC compliant	bs1	bs2	bs3	bs4
Block	Combination function							
Comb.	$f_{762} = a \cdot b \oplus a \cdot c \oplus b \cdot c \oplus b \cdot d \oplus c \cdot d$	yes	4	yes	a	b	c	d

Table 3: Boolean function of four variables adopted for combination

2.6 Key

To create the initial states of the different LFSRs, a process is carried out that uses a 32-character key, which expressed in ASCII code (American Standard Code for Information Interchange), is 256 bits long. The key is subjected to a cryptographic process, which is indicated in Figure 3.

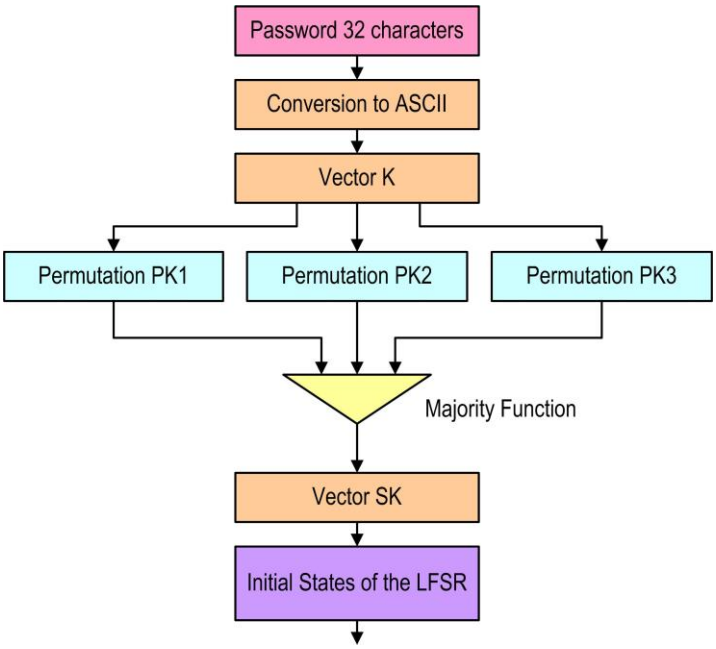

Figure 3: Key to the generator

The permutations are calculated with a multiplicative congruent generator [9]. The generator has the following expression:

$$x_{i+1} = (a_x \cdot x_i) mod\ m_x \ \rightarrow \ (a_x = multiplier;\ \ m_x = module;\ \ x_0 = seed)$$

Table 4 shows the values of the vectors, modules, multipliers and seeds:

Vector	module	multiplier	seed
PK1	1048573	1759	3249
PK2	1048573	1759	3271
PK3	1048573	1759	3301

Table 4: Vectors, modules, multipliers and seeds

The operation results in a 256-bit vector SK[j], which will provide the initial states of the LFSR, sequentially.

2.7 Composition of the generator

With the previously selected components, the structure of the pseudorandom binary generator is completed, Figure 4.

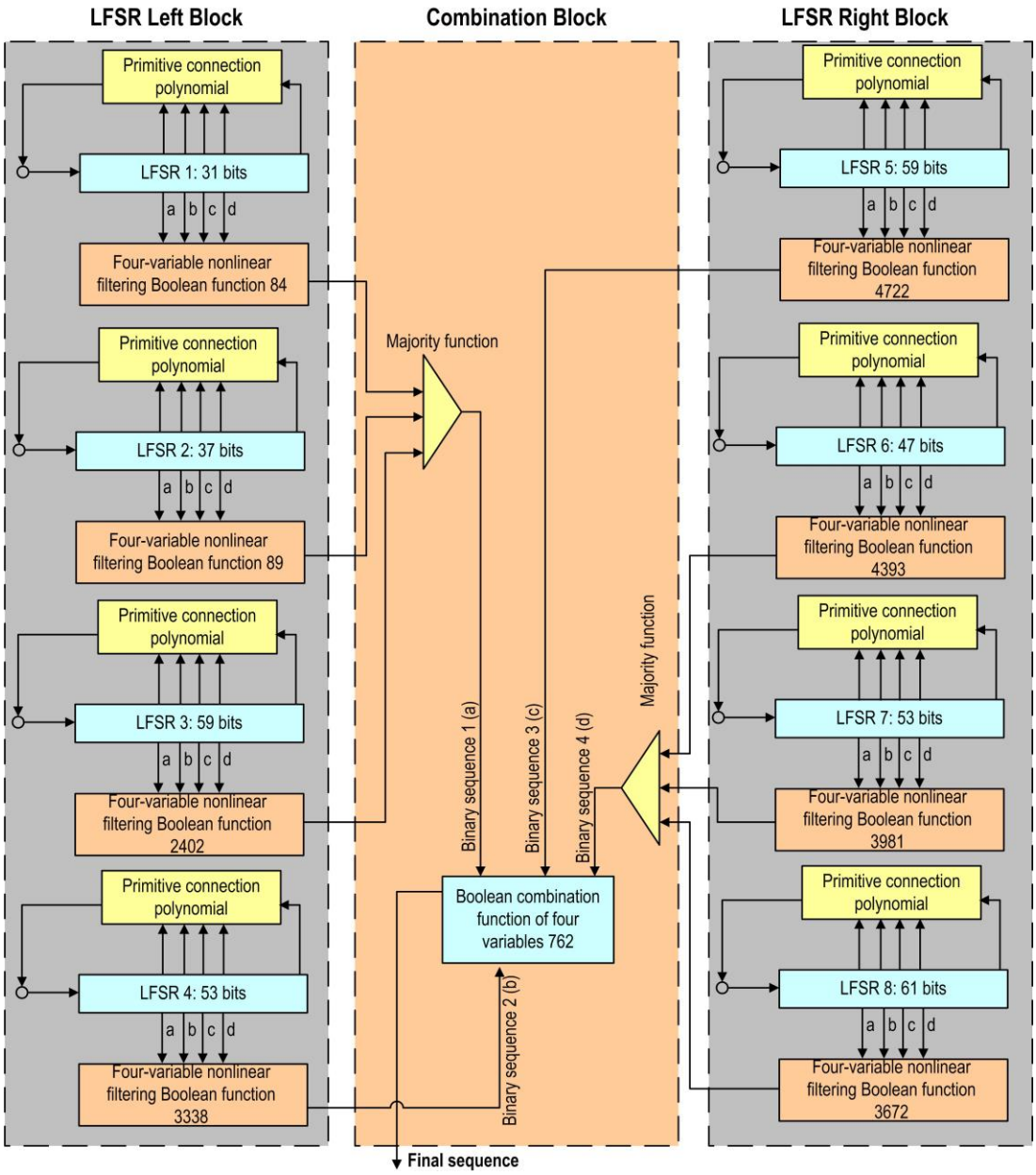

Figure 4: Pseudorandom binary generator scheme with selected LFSR bits and Boolean functions

3 Tests of randomness

3.1 Choice of statistical tests

The statistical test suite for random and pseudorandom number generators for cryptographic applications was selected from the National Institute of Standards and Technology (NIST) Special Publication 800-22 revision 1a, from the work of Rukhin (et al.) [10]. Table 5 shows the statistical tests for random and pseudo-random numbers that make up the package.

	Statistical Tests for Random and Pseudorandom Number
1	Frequency (Monobit)
2	Frequency Test within a Block
3	Runs Test
4	Test for the Longest Run of Ones in a Block
5	Binary Matrix Rank Test
6	Discrete Fourier Transform (Spectral) Test
7	Non-overlapping Template Matching Test
8	Overlapping Template Matching Test
9	Maurer's "Universal Statistical" Test
10	Linear Complexity Test
11	Serial Test:
12	Approximate Entropy Test
13	Cumulative Sums Test (Forward)
14	Cumulative Sums Test (Backward)
15	Random Excursions Test
16	Random Excursions Variant Test

Table 5: Statistical Tests for Random and Pseudorandom Number NIST 800-22

3.2 Tests on the generator

One hundred binary sequences of 1,000,000 bits were analyzed, obtained from the generator from one hundred different keys.

The significance level adopted for the statistical tests is:

$$\alpha = 0.01$$

The null hypothesis is:

$$H_0 \rightarrow p\_value > 0.01$$

3.3 Results analysis

Following the directives of NIST 800-22, to analyze the results, the proportion of samples that pass the tests is determined, and with this data a point graph is constructed, where it must be met that all points are within the upper and lower limits, to accept that the tests were successful.

$$LS, LI = (1 - \alpha) \pm 3 \cdot \sqrt{\frac{\alpha(1-\alpha)}{k}}$$

In our case:

$$k = 100$$

And the chosen significance level is:

$$\alpha = 0.01$$

The upper limit is:

$$LS = (1-\alpha) + 3 \cdot \sqrt{\frac{\alpha(1-\alpha)}{k}} = 1.02$$

$$LS = (1-0.01) + 3 \cdot \sqrt{\frac{0.01(1-0.01)}{100}} = 1.02$$

The lower limit is:

$$LI = (1-\alpha) - 3 \cdot \sqrt{\frac{\alpha(1-\alpha)}{k}} = 0.96$$

$$LI = (1-0.01) - 3 \cdot \sqrt{\frac{0.01(1-0.01)}{100}} = 0.96$$

3.4 Proportion of samples that pass the tests

All tests are considered and the results are indicated in Table 6:

	Statistical Tests for Random and Pseudorandom Number	Total	Pass	Ratio	Upper	Lower
1	Frequency (Monobit)	100	100	1.00	1.02	0.96
2	Frequency Test within a Block	100	100	1.00	1.02	0.96
3	Runs Test	100	97	0.97	1.02	0.96
4	Test for the Longest Run of Ones in a Block	100	100	1.00	1.02	0.96
5	Binary Matrix Rank Test	100	99	0.99	1.02	0.96
6	Discrete Fourier Transform (Spectral) Test	100	99	0.99	1.02	0.96
7	Non-overlapping Template Matching Test	100	100	1.00	1.02	0.96
8	Overlapping Template Matching Test	100	100	1.00	1.02	0.96
9	Maurer's "Universal Statistical" Test	100	97	0.97	1.02	0.96
10	Linear Complexity Test	100	99	0.99	1.02	0.96
11	Serial Test:	100	97	0.97	1.02	0.96
12	Approximate Entropy Test	100	98	0.98	1.02	0.96
13	Cumulative Sums Test (Forward)	100	100	1.00	1.02	0.96
14	Cumulative Sums Test (Backward)	100	100	1.00	1.02	0.96
15	Random Excursions Test	100	97	0.97	1.02	0.96
16	Random Excursions Variant Test	100	97	0.97	1.02	0.96

Table 6: Proportion of samples that pass the tests

The result can be seen in the graph, the points are within the acceptance limits, finally the sequences delivered by the generator pass the randomness tests, in Figure 5:

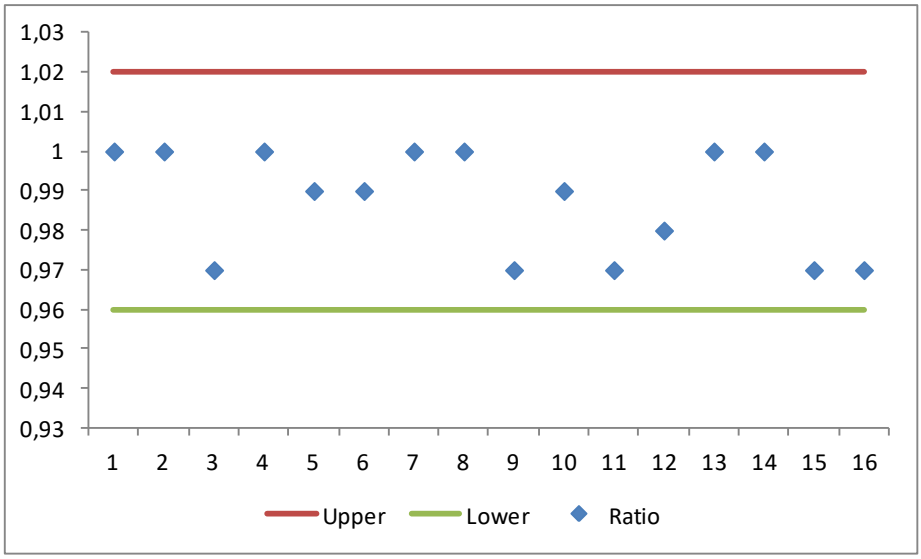

Figure 5: Dot plot

4 Conclusions

A device was designed that non-linearly combines the sequences produced by two groups composed of four LFSRs each, with coupled non-linear filtering functions. Each group of LFSRs produces four binary sequences each, which are mixed using a merge block to obtain the final pseudorandom binary sequence.

The LFSRs that make up each generator have primitive connection polynomials, which ensure the maximum period in the resulting sequence.

Boolean functions are responsible for the non-linear process and the final combination, they guarantee the best cryptographic performance. Once the selection process was carried out, the functions were incorporated into the generator to then put it to work with different key values and generate the respective binary sequences.

Statistical tests of randomness and a subsequent interpretation of the results were carried out on them.

The results obtained were satisfactory, so the model presented is considered valid for the generation of pseudorandom binary sequences of good cryptographic quality.

References

[1] F. Massodi, S. Alam S. and M. Bokhari, "A Analysis of Linear Feedback Shift Registers in Stream Ciphers", 2012, *International Journal of Computer Application*, 16 (17), pp. 0975-887.

[2] A. Menezes, P. Van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", 1996, Massachusetts Institute of Technology.

[3] C. Parr and L. Pelzl, "Understanding Cryptography", 2010, Springer.

[4] W. Stahnke, "Primitive Binary Polynomials", 1973, Mathematics of Computation, 27 (124), pp. 977-980.

[5] G. Seroussi, “Table of Low-Weight Binary Irreducible Polynomials”, 1998, Computer Systems Laboratory.

[6] J. Clark, J. Jacob, S. Maitra, P. Stanica, “Almost Boolean Functions: The Design of Boolean Functions by Spectral Inversion”, 2004, Computational intelligence, 20 (3), pp.450-462.

[7] A. Braeken, “Cryptographic Properties of Boolean Functions and S-Boxes”, 2003, Faculteit Ingenieurswetenschappen. Katholieke Universiteit Leuven.

[8] A. Elhosary, N. Hamdy, I. Farag, I. Rohiem, “State of the Art in Boolean Functions Cryptographic Assessment”, 2013, International Journal of Computer Networks and Communications Security, 1 (3), pp. 88-94.

[9] G. Fishman, “Multiplicative Congruential Random Number Generators with Modulus 2ß : An Exhaustive Analysis for ß = 32 and a Partial Analysis for ß = 48”, 1990, Mathematics of Computation, 54 (189), pp.33-344.

[10] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, “A Statistical Suite for Random and Pseudorandom Number Generators for Cryptographic Applications”, 2010, National Institute of Standards and Technology