electronics

MDPI

Article

A New Approach to the Development of Additive Fibonacci Generators Based on Prime Numbers

Volodymyr Maksymovych [1], Oleh Harasymchuk [1], Mikolaj Karpinski [2,*], Mariia Shabatura [1], Daniel Jancarczyk [2] and Krzysztof Kajstura [2]

[1] Department of Information Technology Security, Lviv Polytechnic National University, 79013 Lviv, Ukraine; volodymyr.m.maksymovych@lpnu.ua (V.M.); oleh.i.harasymchuk@lpnu.ua (O.H.); mariia.m.mandrona@lpnu.ua (M.S.)

[2] Department of Computer Science and Automatics, University of Bielsko-Biala, 43-309 Bielsko-Biala, Poland; djancarczyk@ath.bielsko.pl (D.J.); kkajstura@ath.bielsko.pl (K.K.)

* Correspondence: mkarpinski@ath.bielsko.pl

Abstract: Pseudorandom number and bit sequence generators are widely used in cybersecurity, measurement, and other technology fields. A special place among such generators is occupied by additive Fibonacci generators (AFG). By itself, such a generator is not cryptographically strong. Nevertheless, when used as a primary it can be quite resistant to cryptanalysis generators. This paper proposes a modification to AGF, the essence of which is to use prime numbers as modules of recurrent equations describing the operation of generators. This modification made it possible to ensure the constancy of the repetition period of the output pseudorandom pulse sequence in the entire range of possible values of the initial settings–keys (seed) at specific values of the module. In addition, it has proposed a new generator scheme, which consists of two generators: the first of which is based on a modified AFG and the second is based on a linear feedback shift register (LFSR). The output pulses of both generators are combined through a logic element XOR. The results of the experiment show that the specific values of modules provide a constant repetition period of the output pseudorandom pulse sequence in a whole range of possible values of the initial settings–keys (seed) and provide all the requirements of the NIST test to statistical characteristics of the sequence. Modified AFGs are designed primarily for hardware implementation, which allows them to provide high performance.

Keywords: cybersecurity; pseudorandom sequences generators; prime numbers; additive Fibonacci generator; statistical characteristics

Citation: Maksymovych, V.; Harasymchuk, O.; Karpinski, M.; Shabatura, M.; Jancarczyk, D.; Kajstura, K. A New Approach to the Development of Additive Fibonacci Generators Based on Prime Numbers. *Electronics* **2021**, *10*, 2912. https://doi.org/10.3390/electronics10232912

Academic Editor: Myung-Sup Kim

Received: 16 October 2021
Accepted: 22 November 2021
Published: 24 November 2021

1. Introduction

At the present stage of scientific development and technological progress, pseudorandom bit sequence generators have found more and more application areas. The scientific and practical importance of generating qualitative pseudorandom sequences are significant and many researchers are devoted to this area to find the best algorithms for generating pseudorandom sequences with properties that are closest to random sequences.

In particular, Professor Amalia Beatriz Orue Lopez from Isabel I University in Burgos, Spain [1–3], Professor Miguel Angel Murillo-Escoba from the Centre for Research and Higher Education in Ensenada, Baja California, Mexico [4,5] and Rafik Hamza from LAMIE Laboratory, University of Batna, Algeria [6] in their articles research the various methods of constructing pseudorandom sequence generators, in order to determine issues regarding information protection and estimation quality.

Pseudorandom sequence generators are used in various fields of science and technology. A special place among such generators is occupied by additive Fibonacci generators (AFG) [7–16].

Almost all Fibonacci generators are designed for hardware implementation are used as recurrent equation modules equal to the power of two. This greatly simplifies the hardware implementation of Fibonacci generators but narrows their functionality and does not allow for the improvement of their statistical characteristics without a significant increase in the number of bits of structural elements. The main motivation for this work was the awareness of the need to solve the hardware implementation of Fibonacci generators with an arbitrary module value. This was also facilitated by the authors' experience gained in the hardware implementation of controlled digital frequency synthesizers [17].

Creating new circuit engineering solutions for the hardware implementation of Fibonacci generators allows for the implementation of a new approach to its creation, which, unlike previous approaches, enables the design of generators with an arbitrary module of the recurrent equation, in particular with modules whose values are prime numbers. As a result, there is an opportunity to significantly improve the generator's statistical characteristics.

This work was aimed at researching the hardware implementation of an additive Fibonacci generator, based on an algorithm that uses a module of prime numbers, and to analyse their characteristics.

2. Related Works

Additive Fibonacci generators (AFG) are widely used in cybersecurity devices to generate pseudorandom sequences of bits or numbers.

By itself, such a generator is not cryptographically strong. Nevertheless, using it is fundamental to create a completely secure and resistant cryptanalysis algorithm. For example, based on these generators, the algorithms Fish, Pike and Mush are implemented [7,8].

The use of Additive Fibonacci Generators is not limited to cybersecurity systems (cryptography), they are also used for other applications.

In particular, Ref. [18] describes a method for constructing a pseudorandom number generator based on a recurrent linear sequence of Fibonacci p-numbers to generate a variable carrier frequency of pulse-width modulation (PWM) of the power converter control system to reduce acoustic noise and electromagnetic obstacle level.

The classical algorithm of AGF was formed based on the equation:

$$x_i = (x_{i-1} + x_{i-k})\ mod(m),\ l > k > 0 \quad (1)$$

General view:

$$x_i = (x_{i-a} + x_{i-b} + \ldots + x_{i-p})\ mod(m),\ a > b > \ldots > p > 0 \quad (2)$$

An effective hardware implementation of Equations (1) and (2) are chosen according to Equation $2 - m = 2^n$. This simplifies the hardware implementation of the generators. Compliance with the requirements for the selection of parameters l, k and a, b, $\ldots$, p in Equations (1) and (2) ensures that the repetition period of the sequence at the output of the sequence of generators will be no less than $2^n - 1$ [8].

In articles [10–19], modified additive Fibonacci generators (MAFG) were proposed, operating according to the equation:

$$x_i = (x_{i-a} + x_{i-b} + \ldots + x_{i-p} + a)\ mod(2^n), \quad (3)$$

where, $a = a_0 \oplus a_1 \oplus \ldots \oplus a_z$; a_i $((i = 0, 1, \ldots, z), (z \leq n - 1))$—values of the number x_i binary bits.

In the studies of [10,11,19], it is shown that the process of adding the number "a" causes certain "confusion"—the dependence of each bit of the number, including the youngest bit, from all its other bits, allows to significantly improve the statistical characteristics of the output signals of the MAFG. An array of initial values of numbers x_i, x_{i-a}, x_{i-b}, $\ldots$, x_{i-p}, is called the cryptographic generator key and is under the condition of hardware

implementation. These numbers are used as the initial values of the registers that are part of its block diagram.

Our research on AGF and MAGF [10,11,13,18,19] show a significant dependence of the statistical characteristics of the pseudorandom sequence at the output of the generator on the output parameters. In particular, they strongly depend on the value of the repetition period of the output sequence [10,13]. This means the presence of so-called "weak keys", which could be relatively easily disclosed.

This paper presents the results of research aimed at eliminating this shortcoming of AFG and MAFG. We focus on the hardware implementation of generators.

3. Case Study

3.1. The Structure Schema and the Work Principle of the New AFG

As emphasized above, the construction AGF and MAGF uses algorithms in which the modulus of recurrent Equations (1)–(3) is the power of number 2. This significantly simplifies hardware implementation.

Papers [17,18] proposed a new approach to constructing two-level frequency synthesizers using the change of the average value of the output frequency with an arbitrarily given step. These approaches can be effectively applied in the hardware implementation of our proposed generators.

Figure 1 shows a variant of one such additive Fibonacci generator [10].

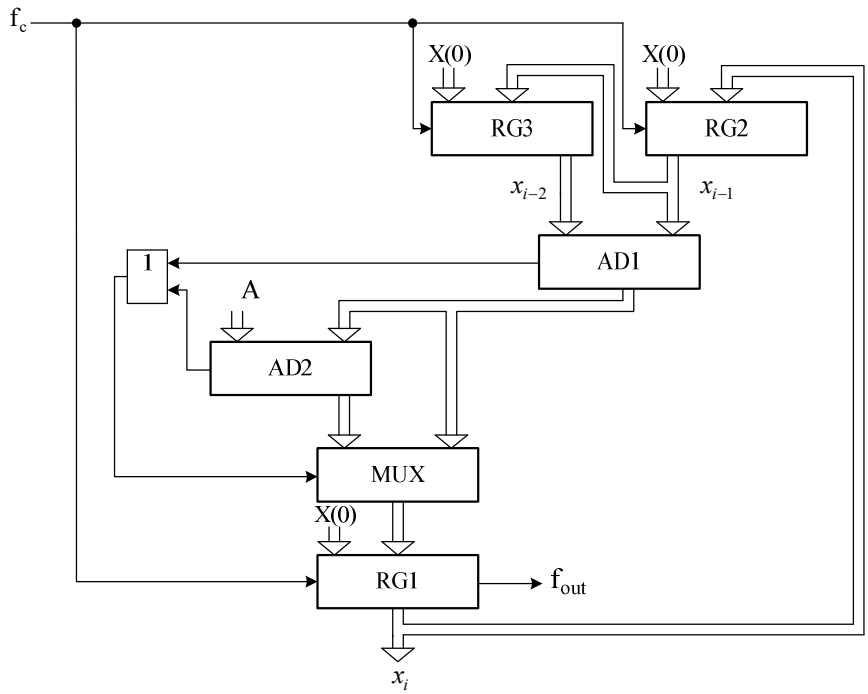

Figure 1. Structure schema of AFG.

AFG consists of registers RG1-RG3, adders AD1-AD2, multiplexer MUX and logical element OR. The generator functions according to the equation:

$$x_i = (x_{i-2} + x_{i-1})\ mod(m), \tag{4}$$

where, m—prime number; x_i, x_{i-1}, x_{i-2}—numbers in registers RG1, RG2 i RG3.

The number of binary bits n of the structural elements of the scheme (RG1-RG3, AD1, AD2) is selected based on the need to ensure the condition $2^n > m$.

Herewith, the smallest value n is selected, at which this condition is fulfilled. The number A, which is applied to one of the AD2 input groups, is determined by the equation $A = 2^n - m$. In the absence of carrying signals on the outputs of AD1 and AD2 to the RG1

information inputs through the multiplex, MUX passes a number from the output AD1, and in the presence of one of these signals the number from the output AD2. The initial number—the key (seed) $X(0)$—is written to registers RG1-RG3.

Clock pulses receive at the clock inputs of the registers RG1-RG3. The output pseudo-random bits sequence formed on one of the register's RG1 bits. The described operating mode of the generator provides a change of the numbers in registers RG1-RG3 in the range of values $0 \div m - 1$.

3.2. Research of the New AFG Characteristics

Figure 2 shows the dependences of the repetition periods of the studied pseudorandom numbers sequence generators on the value of the key $X(0)$.

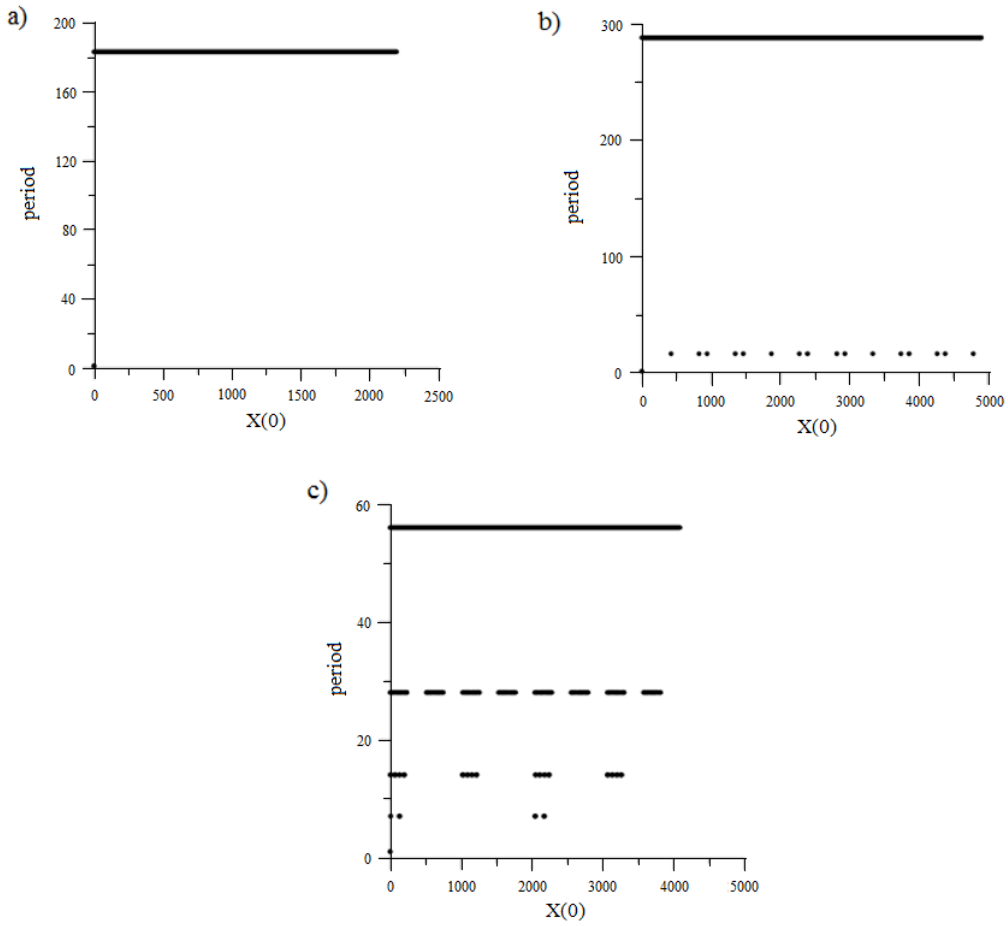

Figure 2. Dependences of AFG repetition periods on the key.

Figure 2a,b shows the dependencies for the new AFG, that function following Equation (4): $m = 13$ (Figure 2a) and $m = 17$ (Figure 2b). In Figure 2c, the corresponding dependence for the classical AFG, which operates following Equation (4) at $m = 2^4 = 16$, is given for comparison. In order to go through all possible values, the initial number is determined by the formula:

$$X(0) = (x_{i-2}(0) + m\, x_{i-1}(0) + m^2\, x_i(0), \quad (5)$$

where, $x_i(0)$, $x_{i-1}(0)$, $x_{i-2}(0)$ are the initial values of the numbers in the registers RG1-RG3, accordingly.

This article presents only some results of different AFG versions of repetition periods research. During the work, a large amount of AFG at different modulus values was analysed. This allows us to draw the following conclusions:

- A new type of AFG, in which the modulus of recurrent equations is a prime number (Figure 1), differs favourably from classical AFG, in which the modulus of recurrent equations is a power of 2, in the absence or relatively small number of "weak keys" (in which the repetition period of the pseudo-random sequence is small);
- In AFG of a new type (Figure 1), there are values of the module for which there are no "weak keys".

Table 1 presents the values of the repetition periods of the output sequence of new AFG for some m values fixed on the whole set of possible $X(0)$ values.

Table 1. The dependence of the repetition periods of the new AFG output sequence for some m values on the whole set of possible $X(0)$ values.

	Prime Numbers, Max and Min Repetition Period Values							
m	2	3	5	7	11	13	17	19
period	7	13	24 4	48 6	120 10	183	288 16	180 9
m	23	29	31	37	41	43	47	53
period	506 22	871	993	1368 36	1723	231 21	2257	1404
m	59	61	67	71	73	79	83	89
period	58	930	4488 66	5113	5403	3120	2296 82	3960 44
m	97	101	103	107	109	113	127	131
period	3116	100 50	3536	2862	1485	4256	16,257	

In Table 1, for values m = 2, 3, 13, 29, 31, 41, 47, 53, 59, 61, 71, 73, 79, 97, 103, 107, 109, 113, 127 no "weak keys" were found in the whole $X(0)$ range values. The only fixed value of the period is indicated in the table. For other m values, a small number of "weak keys" were fixed, for which, along with the principal (predominant) reduced values of the period were indicated.

At sufficiently large m values, the procedure for finding the repetition periods of the output sequence for all possible $X(0)$ values requires a lot of machine time and, under certain conditions, is such that it is practically not implemented. Table 2 shows the values of the repetition periods for relatively large values of m prime numbers when $x_i(0) = 1$, $x_{i-1}(0) = 1$, $x_{i-2}(0) = 1$.

Table 2. Dependence of repetition periods of the new AFG output sequence for some m values, at $x_i(0) = 1$, $x_{i-1}(0) = 1$, $x_{i-2}(0) = 1$.

	Prime Numbers, Repetition Period Values			
m	8191	9973	65,537 (Fermat number)	2,147,483,647 (Marsenn number)
period	22,366,291	99,46,728	1,431,699,455	$>10^{10}$

The tendencies revealed at small values of the module m (Table 1) allow us to state with a high probability that, at relatively large values, the number of "weak keys" will be small or absent.

According to this property, the proposed Fibonacci generator, in which the modules of the recurrent equation are prime numbers, differs favourably from the known Fibonacci generators, in which the value of the modulus is equal to the power of two. For comparison, Table 3 shows some research results of the repetition periods of the output pseudo-random sequence of the classical additive Fibonacci generator, which functions according to Equation (4), at $n = 2^m$. The results are obtained by imitation modelling.

Table 3. Dependence of repetition periods of the output sequence of classical AGF for some values of m on the whole set of values $X(0)$.

***m* Values, Max and Min Repetition Period Values**								
m	2	4	8	16	32	64	128	256
period	7	14 7	28 7	56 7	112 7	224 7	448 7	896 7

Thus, in contrast to the proposed device, in the known device, at $2 > m$, there are different values of the repetition periods, including those that have critically small values. This indicates the presence of "weak keys". In addition, the maximum values of the repetition periods are usually smaller than the relative values of module m in the proposed device. These trends are also observed for arbitrary and much larger values of the modulus m.

Research of the statistical characteristics of the output pseudorandom bit sequences of new AFGs were carried out with NIST tests package [20]. If the proportion fell outside of this interval (0.98–1.0), then this was evidence that the data were non-random. Testing was carried out at different values. As a result, the sequence was entirely non-random, so requirements of statistical security were not accepted. For example, Figure 3 presents a statistical portrait of the output sequence at m = 2,147,483,647.

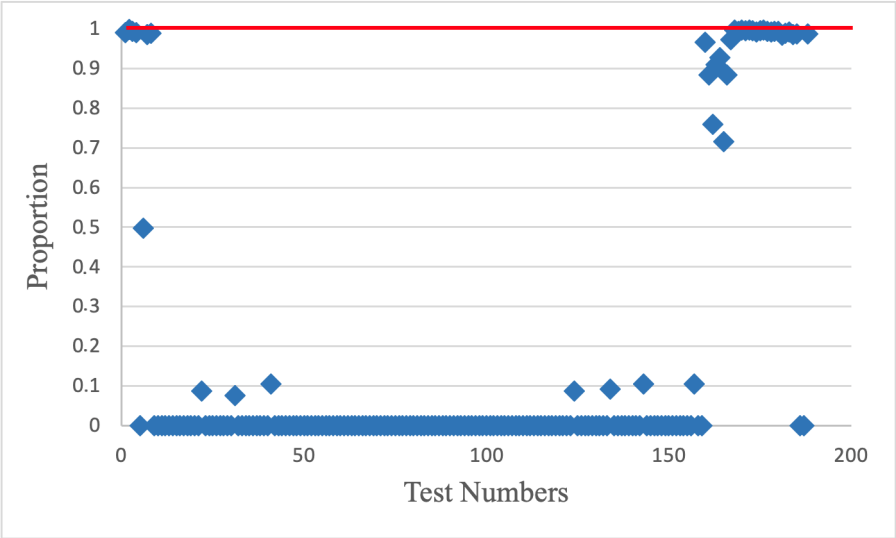

Figure 3. Statistical portrait of AFG at m = 2,147,483,647.

As can be seen from Figure 3, the sequence of the investigated generator does not meet the requirements of randomness as most of the tests were valued at 0 and did not fall within the specified interval.

Thus, new AFGs, built using prime numbers as modules of recurrent equations, provide the absence or the small number of "weak keys". At the same time, they do not accord to the criteria of statistical security; however, they can be used in conjunction with other pseudorandom bit sequence generators (PRBSGs). In this case, their useful property can be used to ensure the constancy of the repetition period of the output sequence for all possible values of the initial settings, and for many values of the module m.

3.3. Structure Scheme and Operation Principle of the Combined PRBSG

The structure scheme of the combined PRBSG is given in Figure 4.

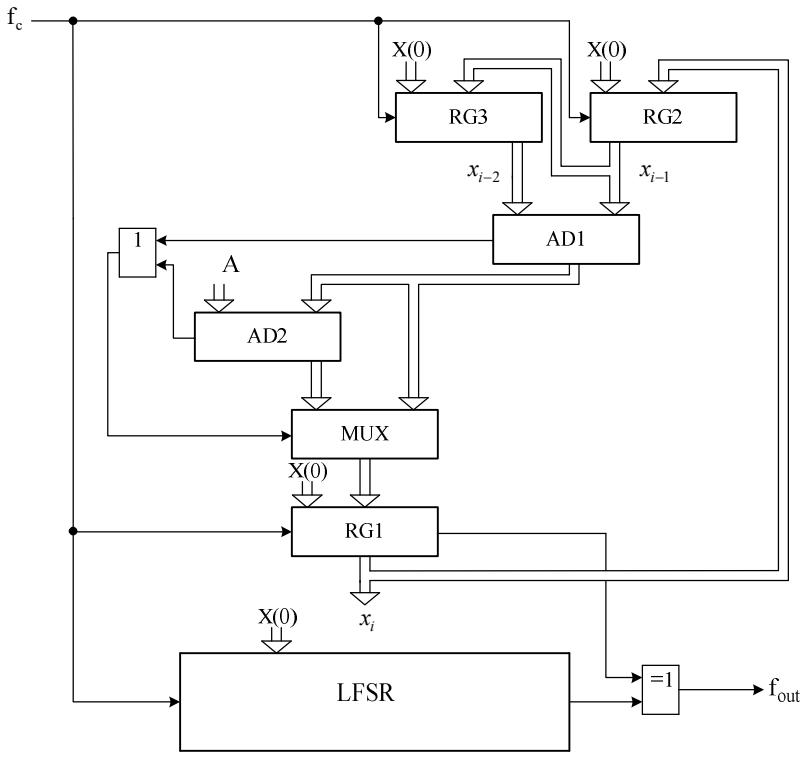

Figure 4. Structure scheme of the combined PRBSG.

The combined PRBSG consists of two generators: a generator based on a new AFG (Figure 1) and a generator based on the shift register with linear feedbacks LFSR. The output pulses of both generators are combined through a logic element XOR. The choice of type and LFSR bit number depends on the need to provide the specified characteristics of the output bit sequence. Instead of LFSR, other types of PRBSGs be used, which requires additional research.

In this work, the combined PRBSG used LFSR work according to the forming equation $F(x) = 1 + 18x + 31x$. The matrix $T1$ and the power of the matrix $r = 10$ [8] are used.

Figure 5 shows a statistical portrait of the LFSR, from which it follows that the output pseudo-random sequence does not pass only two tests from the NIST set.

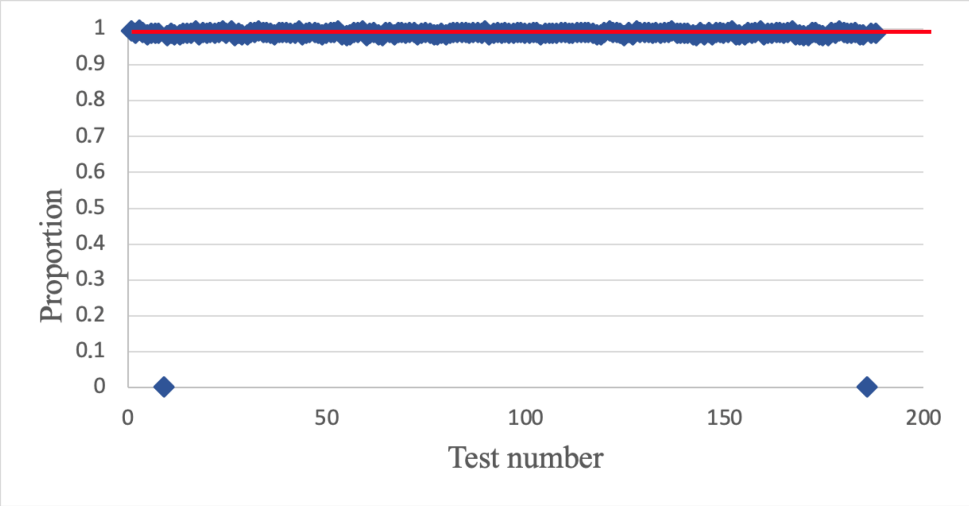

Figure 5. Statistical portrait of the LFSR ($F(x) = 1 + 18x + 31x$, matrix $T1$, power of the matrix $r = 10$).

Figure 6 shows the result of testing combined PRBSG (Figure 4) with such parameters: new AFG m = 2,147,483,647, LFSR $F(x) = 1 + 18x + 31x$, matrix $T1$ and the power of the matrix $r = 10$. The output sequence passes all tests from the NIST set.

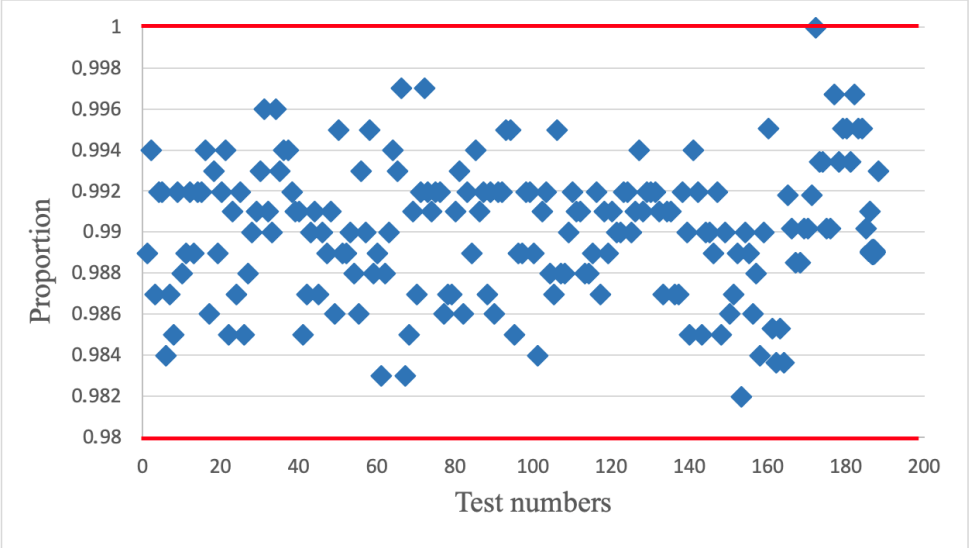

Figure 6. Statistical portrait of the combined PRBSG.

Thus, as can be seen from Figure 6, the results of all tests are within the allowable range. This suggests that the combined PRBSG generator provides the formation of the output pseudo-random sequence with high statistical characteristics.

4. Conclusions

New AFG (Figure 1), built using prime numbers as modules of recurrent equations at specific values of modules, provide a constant repetition period of the output pseudo-random pulse sequence in the whole range of possible values of the initial settings keys (seed).

According to this property, the proposed Fibonacci generator differs favourably from the known Fibonacci generators, in which the value of the modulus is equal to the power of two. In contrast to the proposed device, in the known device, at $m > 2$, there are different values of the repetition periods, including those with critically small values. This indicates

the presence of "weak keys". In addition, the maximum values of the repetition periods in the known device are usually smaller than the relative values of the module m of the proposed device.

When two pseudorandom pulse sequences combine through a logical element XOR, the period of the combined sequence is not less than the repetition period of each of them.

Combined PRBSG (Figure 4), under specific requirements for their construction, can provide the specified statistical characteristics and the absence of "weak keys" in the whole range of possible values of the initial settings–keys (seed).

The results obtained and presented in the article show that the proposed generators can be effectively used in cyber security, particularly as components of cryptographic information protection or noise generators for information security, or as noise-like code sequences of modern communication systems.

Perspective for further research is the development and analysis of other types of combined PRBSG with the possibility of their hardware implementation, as well as expanding the scope of such generators.

Author Contributions: Conceptualization, V.M., O.H., M.S.; Methodology, D.J., K.K., M.K.; Validation, D.J., K.K., V.M.; Formal Analysis, M.K., O.H., M.S.; Investigation, V.M., O.H., M.S., D.J., K.K., M.K.; Data Curation, V.M., O.H., M.S., M.K.; Writing—Original Draft Preparation, D.J., K.K., O.H., M.S.; Writing—Review and Editing, V.M., O.H., M.S, D.J., K.K., M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cardell, S.D.; Requena, V.; Fuster-Sabater, A.; Orue, A.B. Randomness Analysis for the Generalized Self-Shrinking Sequences. *Symmetry* **2019**, *11*, 1460. [CrossRef]
2. Blanco, A.; Orúe, A.B.; López, A.; Martín, A. On-the-Fly Testing an Implementation of Arrow Lightweight PRNG Using a LabVIEW Framework. In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2019; pp. 175–184.
3. Orúe, A.B.; Encinas, L.H.; Fernández, V.; Montoya, F. A Review of Cryptographically Secure PRNGs in Constrained Devices for the IoT. In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2017; pp. 672–682.
4. Murillo-Escobar, M.A.; Cruz-Hernández, C.; Cardoza-Avendaño, L.; Méndez-Ramírez, R. A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. *Nonlinear Dyn.* **2017**, *87*, 407–425. [CrossRef]
5. Meranza-Castillón, M.O.; Murillo-Escobar, M.A.; López-Gutiérrez, R.M.; Cruz-Hernández, C. Pseudorandom number generator based on enhanced Hénon map and its implementation. *J. AEU-Int. J. Electron. Commun.* **2019**, *107*, 239–251. [CrossRef]
6. Hamza, R. A novel pseudo random sequence generator for image-cryptographic applications. *J. Info. Secur. Appl.* **2017**, *35*, 119–127. [CrossRef]
7. Ivanov, M.A.; Chugunkov, I.V. *Theory, Application and Evaluation of the Quality of Pseudorandom Consequences Generators*; KUDITS-OBRAZ: Moskow, Russia, 2003; p. 240.
8. Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*; John Wiley & Sons: Hoboken, NJ, USA, 2007; p. 675.
9. Orue, A.B.; Montoya, F.; Encinas, L.H. Trifork, a New Pseudorandom Number Generator Based on Lagged Fibonacci Maps. *J. Comput. Sci. Eng.* **2010**, *2*, 46–51.
10. Maksymovych, V.; Harasymchuk, O.; Mandrona, M. Additive Fibonacci Generators Using Prime Numbers. In Proceedings of the VIIth International Scientific and Technical Conference "Information protection and Information Systems Security", Lviv, Ukraine, 30–31 May 2019; pp. 66–68.
11. Mandrona, M.; Maksymovych, V.; Harasymchuk, O.; Kostiv, Y. Generator of pseudorandom bit sequence with increased cryptographic immunity. *Metall. Min. Ind.* **2014**, *6*, 24–28.
12. Aluru, S. Lagged Fibonacci Random Number Generators for Distributed Memory Parallel Computers. *J. Parallel Distrib. Computing* **1997**, *45*, 1–12. [CrossRef]
13. Mandrona, M.; Maksymovych, V. Investigation of the statistical characteristics of the modified Fibonacci generators. *J. Autom. Inf. Sci.* **2014**, *46*, 48–53. [CrossRef]
14. Baldoni, S.; Battisti, F.; Carli, M.; Pascucci, F. On the Use of Fibonacci Sequences for Detecting Injection Attacks in Cyber Physical Systems. *IEEE Access* **2021**, *9*, 41787–41798. [CrossRef]

15. Agarwal, A.; Agarwal, S.; Singh, B.K. Algorithm for data encryption & decryption using Fibonacci primes. *J. Math. Control. Sci. Appl.* **2020**, *6*, 63–71.
16. Yacoab, M.; Sha, M.; Mustaq Ahmed, M. Secured Data Aggregation Using Fibonacci Numbers and Unicode Symbols for Wsn. *Int. J. Comput. Eng. Technol.* **2019**, *10*, 218–225. [CrossRef]
17. Wang, J.; Przystupa, K.; Maksymovych, V.; Stakhiv, R.; Kochan, O. Computer Modelling of Two-level Digital Frequency Synthesizer with Poisson Probability Distribution of Output Pulses. *Meas. Sci. Rev.* **2020**, *20*, 65–72. [CrossRef]
18. Maksymovych, V.; Mandrona, M.; Garasimchuk, O.; Kostiv, Y. A study of the characteristics of the Fibonacci modified additive generator with a delay. *J. Autom. Inf. Sci.* **2016**, *48*, 76–82. [CrossRef]
19. Maksymovych, V.; Mandrona, M.; Harasymchuk, O. Dosimetric Detector Hardware Simulation Model Based on Modified Additive Fibonacci Generator. *Adv. Intell. Syst. Comput.* **2020**, *938*, 162–171.
20. NIST SP 800-22 version 1a. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; NIST: Gaithersburg, MD, USA, 2010; p. 131.