Electronic version of an article published as International Journal of Modern Physics C, 25 (04), 1350105 (2014)
https://doi.org/10.1142/S0129183113501052

International Journal of Modern Physics C

Pseudo-Random Bit Generator Based on Lag Time Series

M. García-Martínez[a] and E. Campos-Cantón[b]
División de Matemáticas Aplicadas,
Instituto Potosino de Investigación Científica y Técnologica,
Camino a la Presa San José 2055, Col. Lomas 4 sección
San Luis Potosí, S.L.P., CP. 78216,México
moises.garcia@ipicyt.edu.mx [a] *eric.campos@ipicyt.edu.mx* [b]

Received Day Month Year
Revised Day Month Year

In this work we present a pseudo-random bit generator based on two lag time series of the logistic map using positive and negative values in the bifurcation parameter. In order to hidden the map used to build the pseudo-random series we have used a delay in the generation of time series. These new series when they are mapped x_n against x_{n+1} present a cloud of points unrelated to the logistic map. Finally, the pseudo-random sequences have been tested with the suite of NIST giving satisfactory results for use in stream ciphers.

Keywords: Chaotic behavior; Lyapunov exponent; Bifurcation Diagram; Pseudo-Random Generator; Stream cipher.

PACS Nos.: 11.25.Hf, 123.1K

1. Introduction

The development in the field of communication systems has grown tremendously due to modern needs to e-commerce, e-mail, online banking, etcetera; in addition to the increased use of mobile devices. Usually, the information in this type of communication systems is broadcasted on public channels, so there is a need for privacy of information transmitted. Cryptography is focused on solving problems such as confidentiality, integrity and authentication [1]. The confidentiality is accomplished by an algorithm that takes a plaintext and it is converted to a ciphertext (text without sense), the process can be reversed by a key and gets the plaintext from the ciphertext, this reverse process is called decryption. Cryptosystems can be characterized in two types depending on the key which could be private key (symmetric) or public key (asymmetric). The former uses the same key for encrypt and decrypt while the latter has two different keys, one for encrypt and another for decrypt. The symmetric cryptosystems can be divided in block ciphers and stream ciphers, the first one takes groups of characters and encrypt simultaneously and the second one takes individually characters.

One classic way to build a stream cipher [1,2] is by means of a Pseudo-Random Bit Generator (PRBG), the sequence of bits and information are masked through a xor gate and the output corresponds to a ciphertext. In this type of systems the key is a parameter (like the initial condition), so if we use a different key we will have a different sequence.

Many researchers in the field of nonlinear dynamics have become aware of the relationship between chaos and cryptography, for example in [3,4] the authors have made a comparison between the properties of these areas and show that the ergodicity, sensitivity to initial conditions and control parameter, mixing property, deterministic dynamics and complex structure are analogous to confusion, sensitivity to key, diffusion, deterministic pseudo randomness and algorithm complexity resulting in a very active new area of research, where the cryptosystems based on chaotic systems can be classified in two: continuous-time systems and discrete-time systems.

Cryptosystems based on continuous-time used techniques such as modulation [5,6], masking [7,8], synchronization [9,10], and even hyperchaotic systems [11], etc. On the other hand, discrete-time systems have been used for designing stream ciphers [12−15] (in these cryptosystems the pseudo random sequences are obtained from the time series) and block cipher [16−19]. The idea of using chaotic systems to generate pseudo random sequences was given by S. Oishi *et al* [20] in 1982, since then many approaches have proposed various ways to exploit the sensitivity to initial conditions and their parameters as well as their behavior like randomization and unpredictability, without forgetting that their behavior is deterministic and easy to reproduce.

In this paper, we present a pseudo-random bit generator based on two lag time series of the logistic map using positive and negative values in the bifurcation parameter. These pseudo-random sequences passed the NIST test, so they can be used in stream ciphers algorithms.

This paper is organized as follows. In the next Section we briefly introduce the logistic map when the parameter takes negative and positive values. In Section 3, a basic classification of the pseudo-random number generation is given and also a pseudo random bit generator is introduced. In Section 4 we show the results of the statistical suite test of randomness proposed by NIST which is used to prove safety of the sequences. Finally, in Section 5 are the conclusions.

2. Negative and positive values in the parameter of logistic map

The logistic map is a discrete time system whose dynamics is given in one dimension, and has the form

$$x_{k+1} = f(x_k),\ k = 0, 1, 2, \ldots$$

Where $x_k \in \Re$ and x_0 is the initial condition, such dynamical system is usually referred to as **map**, as it is fully determined by its right hand side. To ensure boundedness of trajectories, the study is usually restricted to maps that are mapping some compact interval into itself. The simplest non-monotonous maps are the so-called unimodal maps.

The logistic map was first presented by Verhulst [21] as a model for the growth of species and is one of the classics in the field of discrete nonlinear dynamics. Later Fiegenbaum [22,23] reported some of the universal quantitative features. The logistic map has been extensively studied and some basic properties are given in [24,25] and more properties can be found in [26]. A generalized logistic map for multiple modal has been reported in [27,28].

The logistic map is defined as

$$f_L(x, \alpha) = \alpha x(1 - x), \tag{1}$$

where $\alpha \in I_p \subset \Re$ is the bifurcation parameter. Generally, the parameter α has been studied on the interval $I_+ = [0, 4]$. However, mathematically nothing restricts to take negative values, thus the logistic map has been also studied for negative values in the interval $I_- = [-2, 0)$. In [29] have had a look at the dynamics of the map at these two real intervals and they have obtained useful information for sociospatial stocks. Because when the parameter α belongs to these intervals I_+ and I_- ensures that orbits do not escape to infinity for some initial conditions. Figure 1 shows the logistic map for different values of $\alpha \in I_p = I_+ \cup I_-$ which presents one or two fixed points located at 0 and at $\dfrac{\alpha - 1}{\alpha}$, for $\alpha \neq 0$.

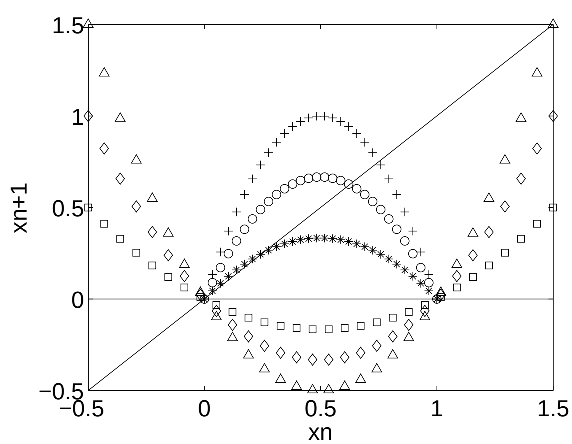

Fig. 1. Logistic map for different values of α: -2 (line formed by triangles), -1.3333 (line formed by diamonds), -0.666 (line formed by squares), 1.3333 (line formed by asterisks), 2.666 (line formed by circles), 4 (line formed by crossings).

The local stability of fixed points can be attractive or repulsive as is shown in

Figure 2, where an asterisk denotes an attractive fixed point and a circle denotes a repulsive fixed point. The fixed point located at zero is repulsive for $\alpha \in [-2, -1] \cup [1, 4]$ and is attractive for $\alpha \in (-1, 1)$. Then local bifurcations occurs at (x, α) equals to $(0, -1)$ and $(0, 1)$. The second fixed point given by $\frac{\alpha-1}{\alpha}$ is attractive for $\alpha \in [1, 3)$ and is repulsive for $\alpha \in [3, 4]$. Also there is another fixed point which is repulsive and is located at 1.5 and only exists for $\alpha = -2$. Thus local bifurcations occurs at (x, α) equals to $(0, 1)$ and $(2/3, 3)$.

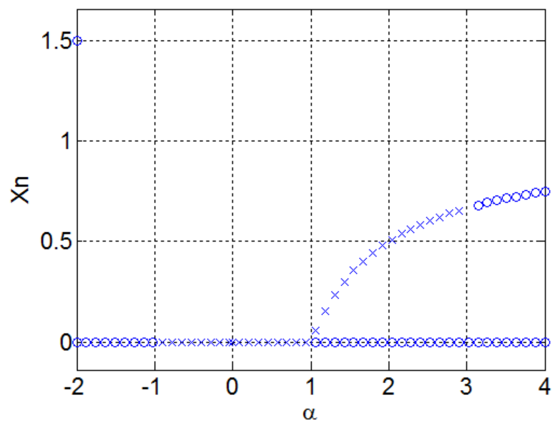

Fig. 2. Stability of the fixed points. The asterisks and circles denote stable and unstable fixed points, respectively.

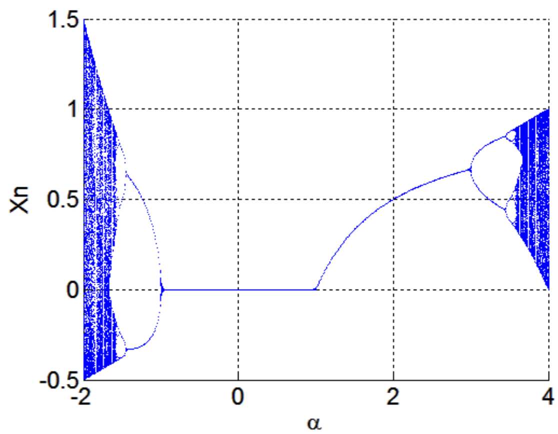

Fig. 3. Bifurcation diagram for the logistic map given by Eq 1.

It is well known that an attractive fixed point does not let oscillations due to all orbits converge to it, meanwhile a repulsive fixed point can yield periodic orbits and even chaotic orbits. Figure 3 shows a bifurcation diagram of the logistic map $f_L(x_0, \alpha)$, with $\alpha \in [-2, 4]$. However, $f_L : [0, 1] \to [0, 1]$ for $\alpha \in [0, 4]$, otherwise $f_L : [-0.5, 1.5] \to [-0.5, 1.5]$. The bifurcation diagram nicely shows the forking of the possible periods of stable orbits from 1 to 2 to 4 to 8 etc. Each of these bifurcation points is a period-doubling bifurcation for the right side of the bifurcation diagram. The ratio of the lengths of successive intervals between values of α for which bifurcation occurs converges to the first Feigenbaum constant. On the other hand, a period halving bifurcation is presented for the left side of the bifurcation diagram. A series of period-halving bifurcations leads the system from chaos to order.

The Lyapunov exponent, which is denoted by λ, gives the global stability of the system Eq.(1) and it is shown in Figure 4. Note that the graph given by the Lyapunov exponent is symmetric with respect to $\alpha = 1$, therein the dynamics of the logistic maps for the parameter $\alpha \in [1, 4]$ is resembled for $\alpha \in [-2, 1]$. This symmetry is given despite of fixed points are different, see Figure 2. When $\alpha \in (-1, 1)$ the system only has one attractive fixed point located at zero and $\lambda < 0$, so every orbit converges to the fixed point. For $\alpha \in [1, 3)$, the system has two fixed points: one attractive and the other repulsive, but $\lambda < 0$, again every orbit converges to the attractive fixed point. For $\alpha = 3$ the system presents a bifurcation and the value of $\lambda = 0$. For $\alpha \in (3, 4]$, the system has two fixed points and both are repulsive and $\lambda < 0$ when the orbit periodically oscillates or $\lambda > 0$ when the orbit oscillates chaotically. Finally, when $\alpha = -2$ the system has two fixed points and both are repulsive and $\lambda > 0$ therefore the orbit oscillates chaotically.

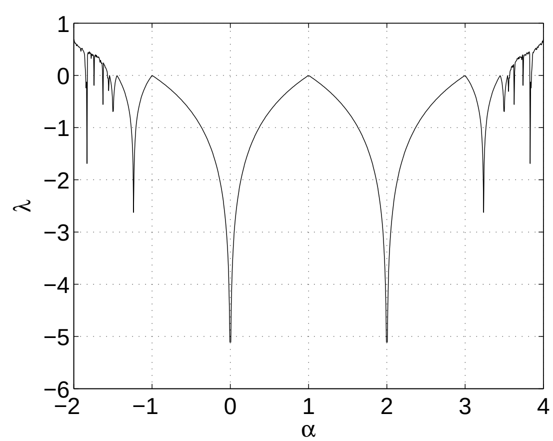

Fig. 4. Lyapunov exponent of the Logistic map.

3. Proposed Pseudo-Random bit generator

Before discussing the approach to generate pseudo-random series, we introduce some basic concepts of random number generators (RNG), as in [2]

- Definition 1: A True Random Number Generator (TRNG) is characterized by the fact that its output cannot be reproduced. This generator is based on physical processes like semiconductor noise. In cryptography, TRNG is often necessary for generating session keys but not for stream ciphers.
- Definition 2: A (General) Pseudo-random Number Generator (PRNG) generates sequences which are computed from an initial seed value, note that PRNG's are not random in a true sense because of their pseudo-random series are computed in a completely deterministic way. A common requirement of PRNG's is that they possess good statistical properties, meaning their output approximates a sequence of true random numbers.
- Definition 3: A Cryptographically Secure Pseudo-random Number Generator (CSPRNG) is a special type of PRNG which possesses the following additional property: a CSPRNG is a PRNG but is unpredictable. This means given n consecutive bits of the key stream, there is no polynomial time algorithm that can predict the next bit s_{n+1} with better than 50 % chance of success. Another property of CSPRNG is that given the above sequence, it should be computationally infeasible to compute any preceding bits s_{n-1}, s_{n-2}.

In this work we are proposing an algorithm to generate a CSPRNG based on two time series of the logistic map, starting from arbitrary initial condition, we have used two different values of the parameter α and 3 units of memory for each time series. The block diagram of the proposed CSPRNG is shown in Figure 5

For the time series $M1$ we have fixed a bifurcation parameter α equal to 4 and iterate the logistic map equation with an arbitrary initial condition $x_0 \in (0, 1)$. In order to remove the shape of the logistic map in the phase space, the time series $M1$ is given by adding: the current iteration x_i, the iteration with a delay of 5 units x_{i-5} and the iteration with a delay of 10 units x_{i-10}. Finally the values of the time series $M1 \in (0, 1)$ have been limited by the operation mod 1, this is shown as follows:

$$M1_i = x_{i-10} + x_{i-5} + x_i, \text{ mod } 1. \tag{2}$$

On the other hand, M2 time series have been performed in a similar process but now with the parameter α equal to -2. Now the time series M2 is given by adding: the current iteration x_i, the iteration with a delay of 6 units x_{i-6} and the iteration with a delay of 10 units x_{i-10}, and the values of the time series $M2 \in (0, 1)$ are bounded with the operation mod 1, this is shown as follows:

$$M2_i = x_{i-10} + x_{i-6} + x_i, \text{ mod } 1. \tag{3}$$

Finally, the time series z is obtained by mixing these two times series $M1$ and $M2$, as follows:

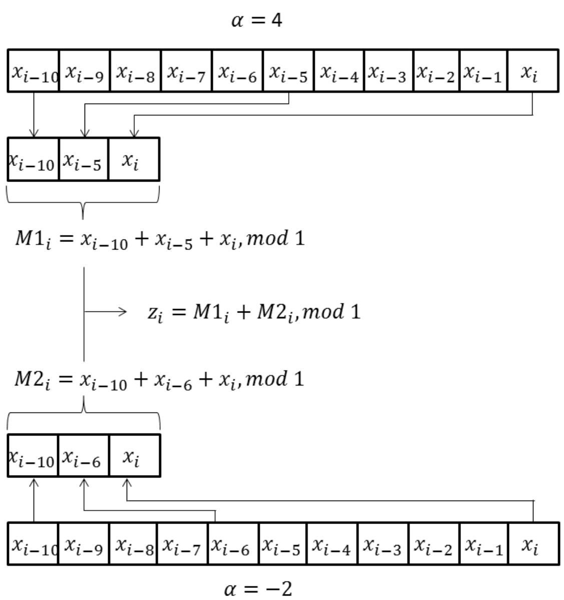

Fig. 5. Block diagram of the proposed Cryptographically Secure Pseudo-random Number Generator (CSPRNG).

$$Z_i = M1_i + M2_i, \text{ mod } 1. \tag{4}$$

The combination of two time series represented by Z_i destroys the structure of the chaotic map used, this can be seen in the phase space in Figure 6. In order to get the binary time series of Z_i, $s_i(Z_i) \in \{0, 1\}$, with the same probability to obtain zeros or ones, thus the process for getting the binary series is as follows:

$$s_i = \begin{cases} 0 & 0 < Z_i \leq 0.5; \\ 1 & 0.5 < Z_i < 1. \end{cases} \tag{5}$$

Remark: If a designer wants to use other values of the bifurcation parameter, fist must analyze the Lyapunov exponent to ensure the asymptotic independence of two trajectories.

For security reasons and in order to increase the keyspace, we use the high sensitivity on the initial condition and bifurcation parameter, we iterate the algorithm 200 times without considering its output bits. Therefore, the encryption process starts by applying the XOR logic function to the bits of the plaintext and the keystream, so this way ciphertext bits are generated. The decryption process is completely the same as the encryption process.

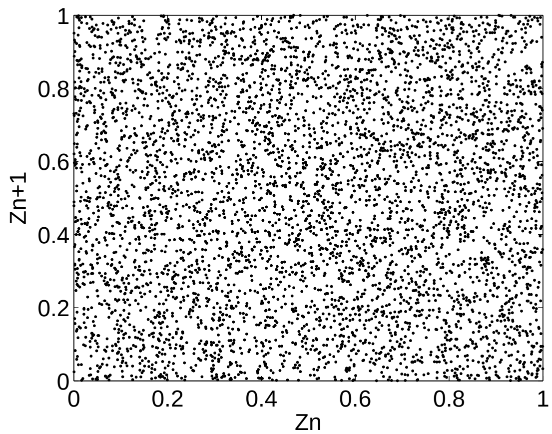

Fig. 6. Phase space from the time series of the proposed CSPRNG.

4. The Statistical Suite Test

To characterize pseudo-random bit series and show that the proposed approach is safe for use in cryptography is important to analyze series with a variety of statistical tests. These statistical tests determine whether the sequence possesses specific characteristics as those truly random sequences would be exhibited. There are several options available for analyzing the randomness of the pseudo-random bit generators for example the suite developed by Beker and Piper [30], the Gustafson's suite [31] or the DIEHARD suite [32], however the most used test for military and commercial purposes is the defined by the NIST [33] that contains a sufficient number of independent statistical tests, which detect any deviation from the randomness, in other words with this suite we have a theoretical reference distribution of statistic determined by mathematical methods, in addition the results of statistical testing must be interpreted with some care and caution to avoid incorrect conclusions about a specific generator. First we need to define a significance level σ. Typically, it is chosen in the range [0.001, 0.01], by default $\sigma = 0.01$ and indicates that one would expect one sequence in 100 sequences to be rejected by the test if the sequence was random.

The NIST has adopted two ways to interpret empirical results in this paper we used the examination of the proportion of sequences that pass a statistical test. For this we need the confidence interval defined as follows:

$$(1-\sigma) \pm 3\sqrt{\frac{1-(1-\sigma)}{m}}, \tag{6}$$

where $\sigma = 0.01$ and m is the sample size of sequences in this case we take 2000 sequences and each has 1000000 of elements. If the proportion falls outside of this interval, then there is evidence that the data is non-random. Now, time series generated by Eq.(5) are analyzed by the statistical tests of NIST suite and the results are as follows:

- Frequency (Monobit) Test: This test verifies whether the number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence. The test assesses the closeness of the fraction of ones to $\frac{1}{2}$. We have obtained that 1974 sequences have passed the test and 26 have failed the test, that is, the number of ones in series is approximately the same that the number of zeroes.
- Frequency Test within a Block: This test determines whether the frequency of ones in an M-bit block is approximately $\frac{M}{2}$, as would be expected under an assumption of randomness. For block size M=1, this test degenerates to the Frequency (Monobit) test. We set the value of blocks with M=128, our results have shown that 1974 sequences have passed the test and 26 sequences have failed the test then the number of ones within a block is about the same.
- Runs Test: A run of length k consists of exactly k identical bits and is bounded before and after with a bit of the opposite value. This test determines whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test determines whether the oscillation between such zeros and ones is too fast or too slow. We have obtained that 1980 sequences have passed the test and 20 sequences have failed the test.
- Test for the Longest Run of Ones in a Block: This test determines whether the length of the longest run of ones within the tested sequence is consistent with the length of the longest run of ones that would be expected in a random sequence. Note that an irregularity in the expected length of the longest run of ones implies that there is also an irregularity in the expected length of the longest run of zeroes. Therefore, only a test for ones is necessary. For this test the size of the block M depends on the length of the sequence; in this case M=10^4 and the results have shown that 1981 sequences have passed the test and 19 sequences have failed the test.
- Binary Matrix Rank Test: This test checks for linear dependence among fixed length substrings of the original sequence, for this test is necessary to construct square matrices of 1024 elements, so we have obtained that 1978 sequences have passed the test and 22 sequences have failed the test, then the generator produces linearly independent sequences.
- Discrete Fourier Transform (Spectral) Test: This test detects periodic features (i.e., repetitive patterns that are near each other) in the tested sequence that

would indicate a deviation from the assumption of randomness. For this test we have used the Discrete Fourier Transform in order to detect peak heights. The results have shown that 1966 sequences have pass the test and 34 sequences have failed the test, then we have obtained series with non-periodicity.

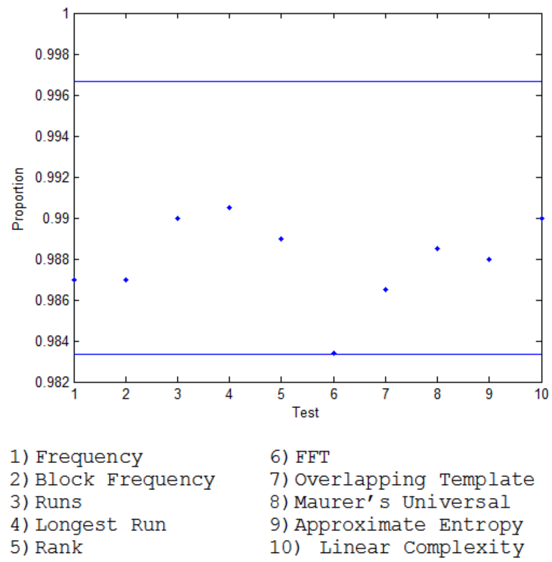

Fig. 7. Confidence interval of Part 1 of the results.

- Non-overlapping Template Matching Test: This test and the Overlapping Template Matching test use an M-bit window to search for a specific M-bit pattern with the purpose of detect generators that produce too many occurrences of a given non-periodic (aperiodic) pattern. If the pattern is not found, the window slides one bit position. If the pattern is found, the window is reset to the bit after the found pattern, and the search resumes. For this test we have used M=9, and the templates and the results are shown as follows: a)000000001, 1981 sequences have passed and 19 sequences have failed the test, b)000100111, 1984 sequences have passed and 16 sequences have failed the test, c)001010011, 1985 sequences have passed and 15 sequences have failed the test, d)010001011, 1985 sequences have passed and 15 have failed the test.
- Overlapping Template Matching Test: The difference between this test and the Non-overlapping test is that when the pattern is found, the window slides only one bit before resuming the search. We have obtained that 1973 sequences have passed the test and 27 sequences have failed the test using a window M=9.
- Maurer's Universal Statistical Test: This test detects whether or not the sequence can be significantly compressed without loss of information. We have obtained

that 1977 sequences passed the test and 23 sequences have failed the test, if a sequence is significantly compressible then it is considered to be non-random.

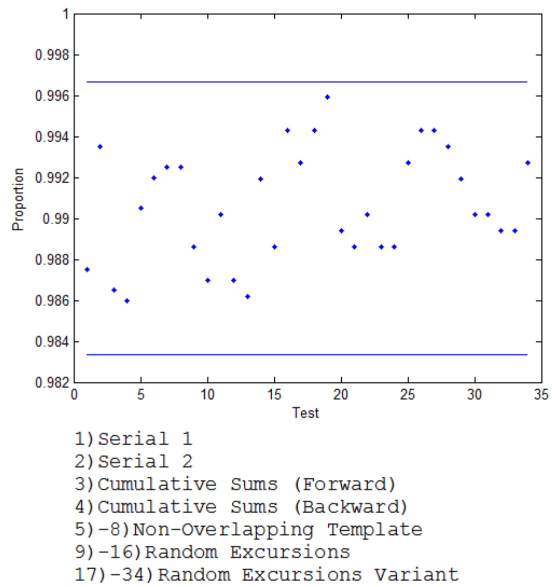

Fig. 8. Confidence interval of Part 2 of the results.

- Linear Complexity Test: This test determines whether or not the sequence is complex enough to be considered random, random sequences are characterized by longer Linear Feedback Shift Register with this test we check the length of a LFSR. For this test we have obtained that 1980 sequences have passed and 20 sequences have failed the test. An LFSR that is too short implies non-randomness.
- Serial Test: This test determines whether the number of occurrences of the 2^M M-bit overlapping patterns is approximately the same as would be expected for a random sequence. Random sequences have uniformity; that is, every m-bit pattern has the same chance of appearing as every other M-bit pattern. Note that for M=1, the Serial test is equivalent to the Frequency test. For this test we have used M=16, the NIST uses two different algorithms to compute this test we have obtained that 1975 sequences have passed and 25 sequences have failed the test and for the other algorithm we have had that 1987 sequences have passed and 13 sequences have failed the test.
- Approximate Entropy Test: This test compares the frequency of overlapping blocks of two consecutive/adjacent lengths (m and m+1) against the expected result for a random sequences. For this test we have taken the block M=10 and have obtained that 1976 sequences have passed and 24 sequences have failed the

test.

- Cumulative Sums (Cusum) Test: This test determines whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of that cumulative sum for random sequences. For compute this test the NIST has adopted two ways (forward and backward) and in both cases the sum could be near zero, we have obtained the following results: a)Forward, 1973 sequences have passed and 27 sequences have failed the test, b)Backward, 1987 sequences have passed and 13 sequences have failed the test.
- Random Excursions Test: This test determines if the number of visits to a particular state within a cycle deviates from what one would expect for a random sequence. This test is actually a series of eight tests, one test and conclusion for each of the states: -4, -3, -2, -1 and 1, 2, 3, 4. We have obtained the following results: a)-4, 1977 sequences have passed and 23 sequences have failed the test, b)-3, 1974 and 26 sequences have passed and failed the test, respectively, c)-2, 1980 and 20 sequences have passed and failed the test, respectively, d)-1, 1974 and 26 sequences have passed and failed the test, respectively, e)1, 1972 and 28 sequences have passed and failed the test, respectively, f)2, 1983 and 17 sequences have passed and failed the test, respectively, g)3, 1977 and 23 sequences have passed and failed the test, respectively, h)4, 1988 and 12 sequences have passed and failed the test, respectively.
- Random Excursions Variant Test: This test detects deviations from the expected number of visits to various states in the random walk. This test is actually a series of eighteen tests, one test and conclusion for each of the states: -9, -8, ..., -1 and 1, 2, ..., 9. Our results have shown to be good and they are presented as follows (state, passed sequences, failed sequences): a)(-9, 1985, 15), b) (-8, 1988, 12), c) (-7, 1991, 9), d) (-6, 1978, 22), e) (-5, 1977, 23), f) (-4, 1980, 20), g) (-3, 1977, 23), h) (-2, 1977, 23), i) (-1, 1985, 15), j) (1, 1988, 12), k) (2, 1988, 12), l) (3, 1987,13), m) (4, 1893, 17), n) (5, 1980, 20), o) (6, 1980, 20), p) (7, 1978, 22), q) (8, 1978, 22), r) (9, 1985, 15).

The above test are summarized in the tables 1 and 2, the first column indicates the name of the test, the second and third columns contain the number of sequences that passed and failed, respectively, and the last column shows the proportion sequences that passing the test. In Figures 7 and 8, it is clear that the portion for each test lies inside the confidence interval, hence the proposed PRBG is Cryptographically Secure according the test of the NIST.

5. Conclusion

In this paper we have presented a theoretical analysis for negative and positive values in the parameter of the logistic map. Also, the analysis of equilibrium points is included as well as the stability. The corresponding bifurcation diagram was obtained and Lyapunov exponent analysis indicates that the logistic map presents

Table 1. Part 1 of the results of the suite of statistical tests.

Statistical Test	No. sequences success	No. sequences failure	Proportion sequences passing the test
Frequency (Monobit) Test	1974	26	0.9870
Frequency Test within a Block (Block=128)	1974	26	0.9870
Runs Test	1980	20	0.9900
Test for the Longest Run of ones in a Block	1981	19	0.9905
Binary Matrix Rank Test	1978	22	0.9890
Discrete Fourier Transform (Spectral) Test	1966	34	0.9834
Overlapping Template Matching Test (Block=9)	1973	27	0.9865
Maurer's Universal Statistical Test	1977	23	0.9885
Approximate Entropy Test (Block=10)	1976	24	0.9880
Linear Complexity Test (Block=500)	1980	20	0.9900

chaotic behavior when the parameter takes certain positive and negative values. Besides we present a Pseudo Random Bit Generator based on two time series of the logistic map, in order to obtain a more complex sequence we have used a delay in the map, finally we have shown that the pseudo-random sequences satisfy all the tests of the NIST suite. Thus these satisfactory results can be used to generate stream ciphers.

Acknowledgments

M. García-Martínez is a doctoral fellows of CONACYT (Mexico) in the Graduate Program on Control and Dynamical Systems at DMAp-IPICYT.
E. Campos-Cantón acknowledges CONACYT for the financial support through project No. 181002.

Table 2. Part 2 of the results of the suite of statistical tests.

Statistical Test	No. sequences success	No. sequences failure	Proportion sequences passing the test
Serial Test 1 (Block=16)	1975	25	0.9875
Serial Test 2 (Block=16)	1987	13	0.9935
Cumulative Sums (Cusum) Test			
a)Forward	1973	27	0.9865
b)Backward	1987	13	0.9935
Non-overlapping Template Matching Test(Block=9)			
a)	1981	19	0.9905
b)	1984	16	0.9920
c)	1985	15	0.9925
d)	1985	15	0.9925
Random Excursions Test			
a)-4	1977	23	0.9886
b)-3	1974	26	0.9870
c)-2	1980	20	0.9902
d)-1	1974	26	0.9870
e)1	1972	28	0.9862
f)2	1983	17	0.9919
g)3	1977	23	0.9886
h)4	1988	12	0.9943
Random Excursions Variant Test			
a)-9	1985	15	0.9927
b)-8	1988	12	0.9943
c)-7	1991	9	0.9959
d)-6	1978	22	0.9894
e)-5	1977	23	0.9886
f)-4	1980	20	0.9902
g)-3	1977	23	0.9886
h)-2	1977	23	0.9886
i)-1	1985	15	0.9927
j)1	1988	12	0.9943
k)2	1988	12	0.9943
l)3	1987	13	0.9935
m)4	1893	17	0.9919
n)5	1980	20	0.9902
o)6	1980	20	0.9902
p)7	1978	22	0.9894
q)8	1978	22	0.9894
r)9	1985	15	0.9927

References

1. A. J. Menezes, P.C. Van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography* (CRC Press, 1997).
2. C. Paar, J. Pelzl, *Understanding Cryptography* (Springer, 2010).
3. G. Alvarez, S. Li, *Int. J. Bifurc. Chaos* **16**(8), 2129 (2006).
4. L. Kocarev, G. Jakimoski, T. Stojanovski, U. Parlitz, *From chaotic maps to encryption schemes — Proc. IEEE Int. Symp. Circuits and Systems* 4, p. 514 (1998).
5. T. Yang, L. O. Chua, *IEEE Trans. Circuits Syst.-I*, **43**(9), 817 (1996).

6. J. Y. Chen, K. W. Wong, L. M. Cheng, J. W., *Chaos* **13**(2), 508 (2003).
7. O. Morgul, M. Feki, *Phys. Lett A* **251**(3), 169 (1999).
8. S. M. Shahruz, A. K. Pradeep, R. Gurumoorthy, *J. Sound Vibrat.* **250**(4), 762 (2002).
9. H. Dedieu, M. P. Kennedy, M. Hasler, *IEEE Trans. Circuits Syst. II* **40**(10), 634 (1993).
10. S. Wang, J. Kuang, J. Li, Y. Lou, H Lu, G. Hu, *Phys. Rev. E* **66**(6),065202 (2002).
11. T. Gao, Z. Chen, *Phys. Lett. A*, **372**(4), 394 (2008).
12. X.Y. Wang, X.J. Wang, *Int. J. Modern Physics C* **19**(5), 813 (2008).
13. X.Y. Wang, Y.X. Xie, *Int. J. Modern Physics C* **23**(3), 1250024 (2012).
14. A. Akhshani, S. Behnia, A. Akhavan, S.-C Lim, Z. Hassan, *Int. J. Modern Physics C* **21**(2), 275 (2010).
15. A. Kanso, N. Smaoui, *Chaos* **40**(5), 2557 (2009).
16. W. Xing-yuan, Y. Qing, *Commun. Nonlinear Sci. Numer. Simulat.* **14**(2), 574 (2009).
17. G. Tang, X. Liao, Y. Chen, *Chaos* **23**(2), 413 (2005).
18. N. Masuda, G. Jakimoski, K. Aihara, L. Kocrev, *IEEE Trans Circuits Syst. I* **53**(6), 1341 (2006).
19. A. Akhavana, A. Samsudina, A. Akhshanib, *Journal of the Franklin Institute* **348**(8), 1797 (2011).
20. S. Oishi, H. Inoue, *Transactions of the Institute of Electronics and Communication Engineers of Japan E*, **65**(9), 534 (1982).
21. P.F. Verhulst, *Correspondance mathmatique et physique* **10**, 113 (1838).
22. M. J. Fiegenbaum, *J. Stat. Phys* **21**(6), 669 (1979).
23. M. J. Fiegenbaum, *Physica D* **7**(1-3), 16, (1983).
24. S. N. Elaydi, *Discrete Chaos* (Chapman & Hall, 2000).
25. S. Lynch, *Dynamical Systems With Applications* (Birkhauser/Springer, Boston 2010).
26. R. A. Holmgren, *A First Course in Discrete Dynamical Systems* (Springer-Verlag, New York 1996).
27. E. Campos-Canton, R. Femat, A.N. Pisarchik, *Commun. Nonlinear Sci. Numer. Simulat.* **16**(9), 3457 (2011).
28. M. García-Martínez, I. Campos-Cantón, E. Campos-Cantón and S. Celikovsky, *Nonlinear Dyn.*, DOI 10.1007/s11071-013-1007-4, (2013).
29. D. S. Dendrinos and M. Sonis, *Ann. Reg. Sci.* **27**(4), 297 (1993).
30. Beker H, Piper F. *Cipher systems: the protection of communications* (Van Nostrand Reinhold, New York 1982).
31. Gustafson H., Dawson E., Nielsen L,. Caelli W. *Computers and Security* **13**(8), 687 (1994).
32. G. Marsaglia, *DIEHARD Statistical Tests* http://www.stat.fsu.edu/pub/diehard/.
33. A. Rukhin et al, *NIST special publication* 800-22 (2010).