A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing

Vinod Patidar and K. K. Sud
Department of Basic Sciences, School of Engineering,
Sir Padmapat Singhania University, Bhatewar, Udaipur – 313 601, India
E-mail: vinod_r_patidar@yahoo.co.in

N. K. Pareek
University Computer Centre, Vigyan Bhawan, New Campus
M.L.S. University, Udaipur 313002, Rajasthan, India

Keywords: pseudo random, random, PRBG, random bit generator, logistic map, cryptography, stream cipher

Received: May 20, 2008

During last one and half decade an interesting relationship between chaos and cryptography has been developed, according to which many properties of chaotic systems such as: ergodicity, sensitivity to initial conditions/system parameters, mixing property, deterministic dynamics and structural complexity can be considered analogous to the confusion, diffusion with small change in plaintext/secret key, diffusion with a small change within one block of the plaintext, deterministic pseudo randomness and algorithmic complexity properties of traditional cryptosystems. As a result of this close relationship several chaos-based cryptosystems have been put forward since 1990. In one of the stages of the development of chaotic stream ciphers, the application of discrete chaotic dynamical systems in pseudo random bit generation has been widely studied recently. In this communication, we propose a novel pseudo random bit generator (PRBG) based on two chaotic logistic maps running side-by-side and starting from random independent initial conditions. The pseudo random bit sequence is generated by comparing the outputs of both the chaotic logistic maps. We discuss the suitability of the logistic map by highlighting some of its interesting statistical properties, which make it a perfect choice for such random bit generation. Finally, we present the detailed results of the statistical testing on generated bit sequences, done by the most stringent tests of randomness: the NIST suite tests, to detect the specific characteristics expected of truly random sequences.

Povzetek: Predstavljen je psevdo naključni generator bitov na osnovi kaotičnega pristopa.

1 Introduction

New rapid developments in the telecommunication technologies especially the Internet and mobile networks have extended the domain of information transmission, which in turn present new challenges for protecting the information from unauthorized eavesdropping. It has intensified the research activities in the field of cryptography to fulfill the strong demand of new secure cryptographic techniques [1, 2].

Recently researchers from the nonlinear dynamics community have noticed an interesting relationship between chaos and cryptography. According to that, many properties of chaotic systems such as: ergodicity, sensitivity to initial conditions/system parameters, mixing property, deterministic dynamics and structural complexity can be considered analogous to the confusion, diffusion with small change in plaintext/secret key, diffusion with a small change within one block of the plaintext, deterministic pseudo randomness and algorithmic complexity properties of traditional cryptosystems [3]. As a result of this close relationship several chaos-based cryptosystems have been put forward since 1990 [4]. These chaos-based cryptosystems can be broadly classified into two categories: analog and digital. Analog chaos-based cryptosystems are based on the techniques of control [5, 6] and synchronization [5, 6] of chaos. There are several ways through which analog chaos-based cryptosystems can be realized such as: chaotic masking [7-11], chaotic modulation [12-15], chaotic switching [16, 17], inverse system approach [18, 19] etc. On the other hand in digital chaos-based cryptosystems, chaotic discrete dynamical systems are implemented in finite computing precision. Again there are number of ways through which digital chaos-based cryptosystems be realized: block ciphers based on forward and/or reverse iterations of chaotic maps [4, 20-23], block ciphers based on chaotic round functions [24-27], stream ciphers implementing chaos-based pseudo random bit generators (PRBG) [28-33] etc.

The subject of the present manuscript is the generation of cryptographically secure pseudo random bit sequences, which can be further used in the development of fool-proof stream ciphers and its statistical testing. In the following paragraph, we briefly summarize a few efforts undertaken recently in this direction.

The first, relatively unnoticed, idea of designing a pseudo-random number generator by making use of chaotic first order nonlinear difference equation was proposed by Oishi and Inoue [34] in 1982 where they could construct a uniform random number generator with an arbitary Kolmogorov entropy. After a long gap, in 1993 Lin and Chua [35] designed a pseudo random number generator by using a second-order digital filter and realized it on digital hardware. In 1996 Andrecut [36] suggested a method for designing a random number generator based on logistic map and also compared the congruential random generators, which are periodic, with the logistic random number generator, which is infinite and aperiodic. In 1999 Gonzalez and Pino [37] generalized the logistic map and designed a truly unpredictable random function, which helped in the generation of truly random numbers. In 2001 Kolesov et al [38] developed a digital random-number generator based on the discrete chaotic-signal. The suggested digital generator employed the matrix method of chaotic-signal synthesis. Further, Kocarev [39] and Stojanovski et al [40] analyzed the application of a chaotic piecewise-linear one-dimensional map as random number generator. Li et al [32] did a theoretical analysis, which suggests that piecewise linear chaotic maps have perfect cryptographic properties like: balance in the defined interval, long cycle length, high linear complexity, good correlation properties etc. They also pointed out that bit streams generated through a single chaotic system are potentially insecure as the output may leak some information about the chaotic system. To overcome this difficulty, they proposed a pseudo random bit generator based on a couple of piecewise linear chaotic maps, which are iterated independently and the bit streams are generated by comparing the outputs of these chaotic maps. They also justified their theoretical claims through a few numerical experimentations on the proposed pseudo random bit generator. In 2003 Kocarev and Jakimoski [41] discussed the different possibilities of using chaotic maps as pseudo-random number generators and also constructed a chaos-based pseudorandom bit generator. In 2004 Fu et al [42] proposed a chaos-based random number generator using piecewise chaotic map. Further, a one-way coupled chaotic map lattice was used by Huaping et al [43] for generating pseudo-random numbers. They showed that with suitable cooperative applications of both chaotic and conventional approaches, the output of the spatiotemporal chaotic system can meet the practical requirements of random numbers i.e. excellent random statistical properties, long periodicity of computer realizations and fast speed of random number generations. This pseudo-random number generator can be used as an ideal synchronous and self-synchronizing stream cipher for secure communications. In 2005 Li et al [44] designed and analysed a random number generator based on a piecewise-linear map. A new pseudo-random number generator (PRNG) based on modified logistic map was proposed by Liu [45] and a design of a chaotic stream cipher using it was also suggested. Further, a chaotic random number generator was developed by Wang et al [46] and realized it by an analog circuit. In 2006, Wang et al [47] proposed a pseudo-random number generator based on z-logistic map, where the binary sequence through the chaotic orbit was realized under finite computing precision. Recently in 2007, Ergun and Ozogur [48] showed that the bit streams, generated from the stroboscopic Poincare map of a non-autonomous chaotic electronic circuit, pass the four basic tests of FIPS-140-2 as well as NIST tests suite. Very recently, Hu et al [49] proposed a true random number generator (which generates a 256-bit random number by computer mouse movement), where the authors used three chaos-based approaches namely: discretized 2D chaotic map permutation, spatiotemporal chaos and MASK algorithm to eliminate the effect of similar mouse movement patterns. The results have been tested through NIST tests suite. Recently, Patidar et al [50] proposed a pseudorandom bit generator based on the chaotic standard map and presented its testing analysis using the NIST as well as DIEHARD test suites. No failure has been observed in any of the tests of these two test suites.

In this paper, we propose a pseudo random bit generator (PRBG) based on two chaotic logistic maps. Most of the existing pseudo random bit generators [34-47] are based on a single chaotic system and there are known techniques in chaos theory to extract information about the chaotic systems from its trajectory, which makes such chaos-based pseudo random bit generators insecure [32]. However the proposed pseudo random bit generator is based on two chaotic systems running side-by-side, which of course increases the complexity in the random bit generation and hence becomes difficult for an intruder to extract information about the chaotic system. In the next section, we briefly introduce the logistic map, which is a basic building block of the proposed pseudo random bit generator and its properties, which make it a suitable choice for the generation of random bit sequences.

2 The logistic map

The logistic map is a very simple mathematical model often used to describe the growth of biological populations. In 1976 May [51] showed that this simple model shows bewildering complex behaviour. Later Fiegenbaum [52, 53] reported some of the universal quantitative features, which became the hallmark of the contemporary study of chaos. Because of its mathematical simplicity, this model continues to be useful test bed for new ideas in chaos theory as well as application of chaos in cryptography [4]. The simple modified mathematical form of the logistic map is given as:

$$X_{n+1} = f(X_n) = \lambda X_n (1 - X_n), \quad (1)$$

where X_n is a state variable, which lies in the interval [0, 1] and λ is called system parameter, which can have any value between 1 and 4.

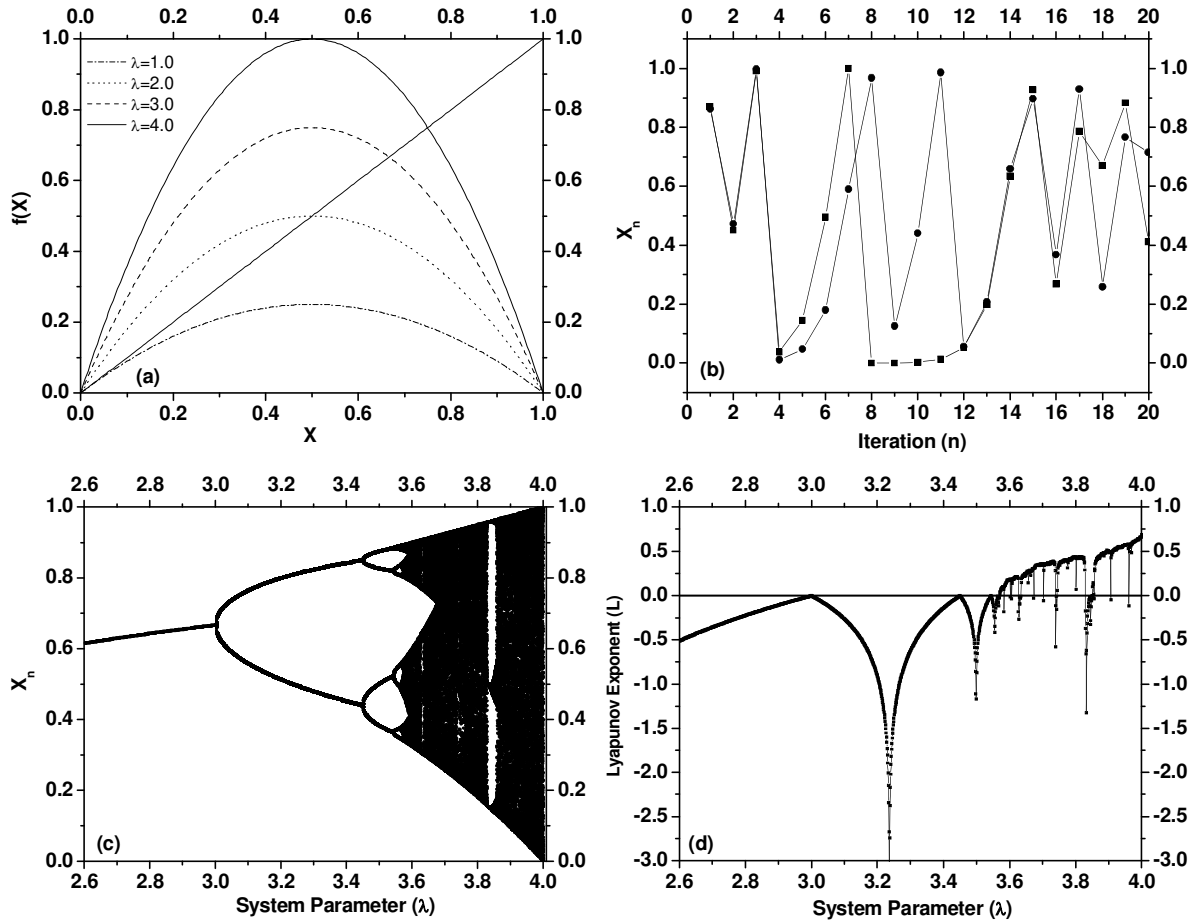

Figure 1: Behaviour of the logistic map: (a) map function $f(X) = \lambda X(1-X)$ for different values of parameter λ, (b) sensitivity on initial conditions for $\lambda = 4.0$, (c) bifurcation plot showing the qualitative changes in the dynamical behaviour as a function of parameter λ and (d) Lyapunov exponent (quantitative measurement of chaos) as a function of parameter λ.

In Figure 1(a), we have plotted the map function $f(X)$ as a function of X for different values of system parameter λ. It is clear that the map function $f(X)$ is symmetric about the mid point of the interval [0, 1]. This iterative map shows a strange complex behaviour for the system parameter values $\lambda > 3.5699...$, where map function never repeats its history. This peculiar behaviour is termed as chaos and more precisely, it can be described by the phrase 'sensitivity on initial conditions'. In Figure 1(b), we have depicted one such example of sensitivity on initial conditions for $\lambda = 4.0$. It is clear that the two trajectories of the logistic map starting nearby, soon diverge exponentially in the course of time and have no correlation between them. If we calculate the correlation coefficient for these two data sets (for $N =$ 1 to 10^6), it comes out equal to -0.000839 at the significance level of $\alpha = 0.01$, which confirms the completely uncorrelated behaviour of two trajectories, which are starting from almost same initial conditions. In Figure 1(c), we have summarized the complete dynamical behaviour of the logistic map by using the bifurcation plot: a plot illustrating the qualitative changes in the dynamical behaviour of the logistic map as a function of system parameter λ. It is also clear from the bifurcation diagram that the map function is surjective/onto in the complete interval [0, 1] only at $\lambda = 4.0$ i.e., each and every value of $f(X)$ in the interval [0, 1] is an image of at least one value of X in the same interval [0, 1]. The interval of surjectivity reduces as we decrease the value of λ form 4.0. In Figure 1(d), we have displayed the Lyapunov exponent ($L = \lim_{N\to\infty} \sum_{i=1}^{N} \ln | f'(X_i) |$, which is a quantative measure of chaos and a positive Lyapunov exponent indicates chaos) as a function of system parameter λ.

Invariant density measure and ergodicity: If we divide the complete range of state variable [0, 1] into a set of M equal sub-intervals and calculate the number that a trajectory visits a particular sub-interval i ($1 \le i \le M$), if it is m_i then the probability associated with the sub-interval i is $p_i = m_i / N$ (where N is the total number of trajectory points considered). A graph of

p_i as a function of i gives us the natural probability distribution or probability measure.

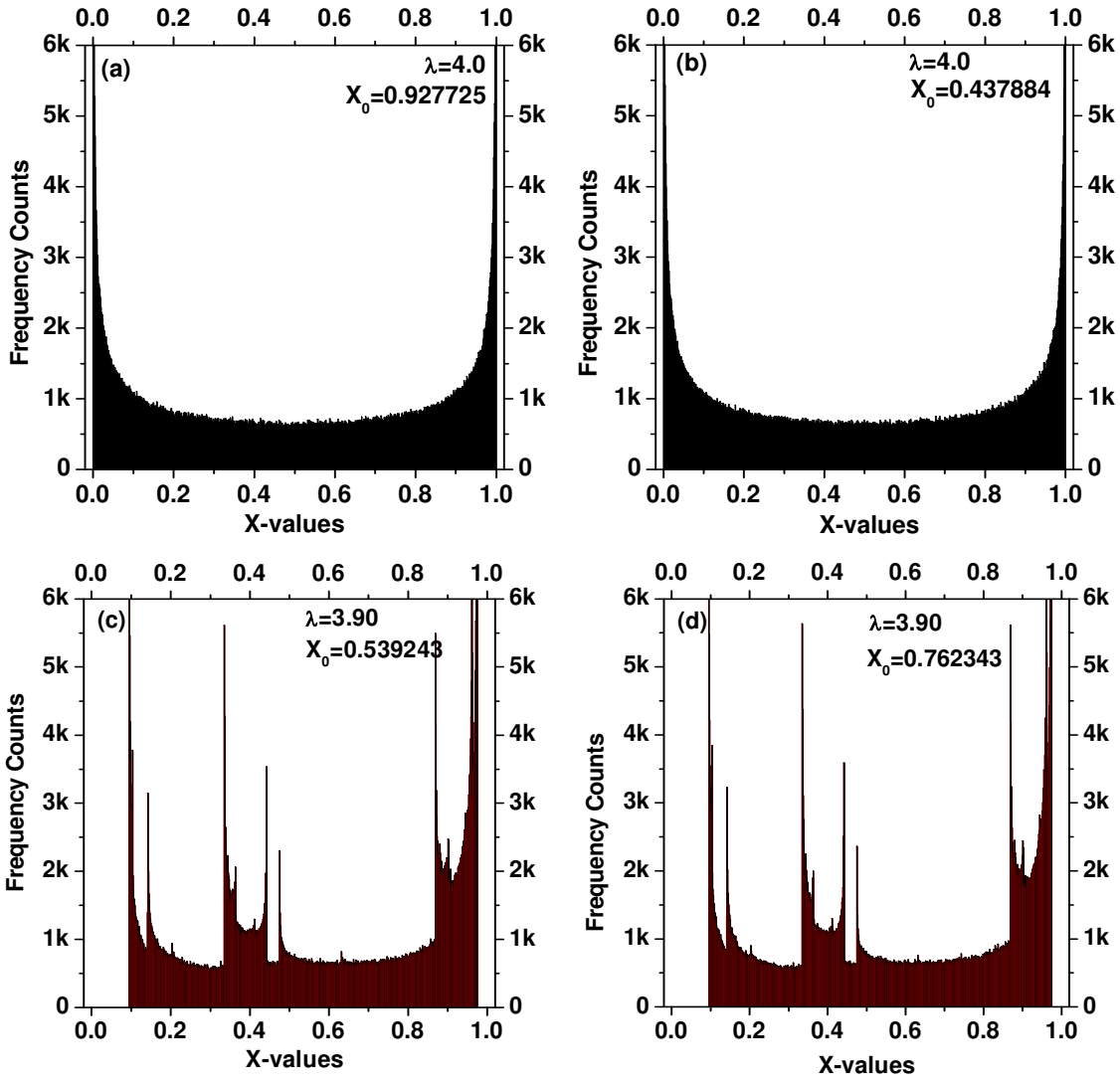

Figure 2: Probability distributions for the logistic map trajectories (a) and (b) for $\lambda = 4.0$ & (c) and (d) for $\lambda = 3.90$.

For a chaotic trajectory, this probability distribution does not depend on the starting point of the trajectory (if we observe the trajectory for a long enough duration) i.e., the probability measure is unchanged under the dynamics of the system, we term it as *invariant probability measure* or *invariant density measure*. It has been shown analytically that for the logistic map with system parameter $\lambda = 4$ the probability distribution is given by [54],

$$P(X) = \frac{1}{\pi\sqrt{X(1-X)}}. \tag{2}$$

If such an invariant distribution exists for a system then it allows us to replace the time averages by the spatial averages and the system is called *ergodic*. This ergodic property provides us a very simple way for calculating the average properties of the system. For example the average Lyapunov exponent for the logistic map with system parameter $\lambda = 4.0$ can be calculated with the help of above invariant probability distribution as:

$$L = \int_0^1 P(X)\,\lambda(X)\,dX, \tag{3}$$

here $\lambda(X)$ is the local Lyapunov exponent. Using (3) we have

$$L = \int_0^1 \frac{1}{\pi\sqrt{X(1-X)}} \ln|f'(X)|\,dX, \tag{4}$$

$$= \int_0^1 \frac{1}{\pi\sqrt{X(1-X)}} \ln|4(1-2X)|\,dX = \ln 2, \tag{5}$$

which is positive and confirms the chaotic nature of the logistic map at $\lambda = 4.0$ .In Figures 2(a) and 2(b), we have shown probability distributions for two different trajectories of logistic map starting from different initial conditions ($X_0 = 0.927725$ and 0.437884) with system parameter $\lambda = 4.0$. Here the interval [0, 1] has been divided into 1000 equal sub-intervals and total $N = 10^6$

points are used for each trajectory. Clearly both the distributions are same hence the logistic map exhibits unique invariant probability measure for $\lambda = 4.0$. It is also clear that the probability distributions are symmetric about the mid point of the interval [0, 1]. However in Figures 2(c) and 2(d), probability distributions are displayed for the two logistic trajectories starting form $X_0 = 0.835283$ and 0.582735 with the system parameter $\lambda = 3.90$. It is clear that the logistic map also exhibits invariant probability measure for $\lambda = 3.90$ but the distribution is not symmetric about the mid point of the interval [0, 1]. From Figure 2, one may also conclude that the logistic map has surjective character in the complete interval [0, 1] only very near to $\lambda = 4$. In the next section, we discuss the basic terminology for the random bit generation and details of the proposed pseudo random bit generator (PRBG).

3 The proposed PRBG

A *random bit generator (RBG)* is a device or algorithm, which outputs a sequence of statistically independent and unbiased binary digits. Such generator requires a naturally occurring source of randomness (non-deterministic). In most practical environments designing a hardware device or software programme to exploit the natural source of randomness and produce a bit sequence free from biases and correlation is a difficult task. In such situations, the problem can be ameliorated by replacing a random bit generator with a pseudo random bit generator (PRBG).

A *pseudo random bit generator (PRBG)* is a deterministic algorithm, which uses a truly random binary sequence of length *k* as input called seed and produces a binary sequence of length *l*>>*k*, called pseudo random sequence, which appears to be random. The output of a PRBG is not truly random; in fact the number of possible output sequences is at most a small fraction ($2^k/2^l$) of all possible binary sequences of length *l*. The basic intent is to take a small truly random sequence of length *k* and expand it to a sequence of much larger length *l* in such a way that an adversary can not efficiently distinguish between output sequence of PRBG and truly random sequence of length *l* [2].

In this paper, we are proposing a PRBG, which is based on two logistic maps, starting from random independent initial conditions $(X_0, Y_0 \in (0,1)$ and $X_0 \neq Y_0$)

$$X_{n+1} = \lambda_1 X_n (1 - X_n), \quad (6)$$

$$Y_{n+1} = \lambda_2 Y_n (1 - Y_n). \quad (7)$$

The bit sequence is generated by comparing the outputs of both the logistic maps in the following way:

$$g(X_{n+1}, Y_{n+1}) = \begin{cases} 1 & if \ X_{n+1} > Y_{n+1} \\ 0 & if \ X_{n+1} \le Y_{n+1} \end{cases}, \quad (8)$$

The set of initial conditions $(X_0, Y_0 \in (0,1)$ and $X_0 \neq Y_0)$ serves as the seed for the PRBG, if we supply the exactly same seed to the PRBG, it will produce the same bit sequence due to the above deterministic procedure. The schematic block diagram of the proposed PRBG is shown in Figure 3.

In a recent analytical study Li et al [32] showed that the binary sequences produced by comparing the outputs of two chaotic maps will have perfect cryptographic properties if following requirements are satisfied:

(i) Both the maps should produce asymptotically independent trajectories as $n \to \infty$,
(ii) both the maps are surjective on the same interval,
(iii) both the maps have unique invariant density distributions $P_1(x)$ and $P_2(x)$ and are ergodic on the defined interval,
(iv) either $P_1(x) = P_1(x)$ or $P_1(x)$ and $P_2(x)$ are symmetric about the mid point of the interval.

It is clear from the discussion of Section 2 that the logistic map exhibits all the above mentioned properties wherever it shows chaotic behaviour. In view of the condition (ii), we have to choose the same value of λ for both the chaotic maps (i.e., $\lambda_1 = \lambda_2 = \lambda$) to maintain its surjectivity in the same interval. However it would be most appropriate to choose λ very near to 4.0 to make available a large interval for the seed values X_0 and Y_0, which will in turn increase the key space of the stream cipher, where the proposed cipher is going to be used. It is also suggested that before choosing $\lambda_1 = \lambda_2 = \lambda$ other than 4.0, a careful analysis of Lyapunov exponent must be done to take care of the asymptotic independence of two trajectories (property (i)), larger the Lyapunov exponent lesser the correlation between the trajectories starting from almost same initial conditions.

In the next section, we mention various resources for statistical testing of PRBGs which are available to researchers from academia and industry who wish to analyze their newly developed PRBG. We also briefly introduce the resource (NIST tests suite), which we have used for analyzing the proposed PRBG and discuss the

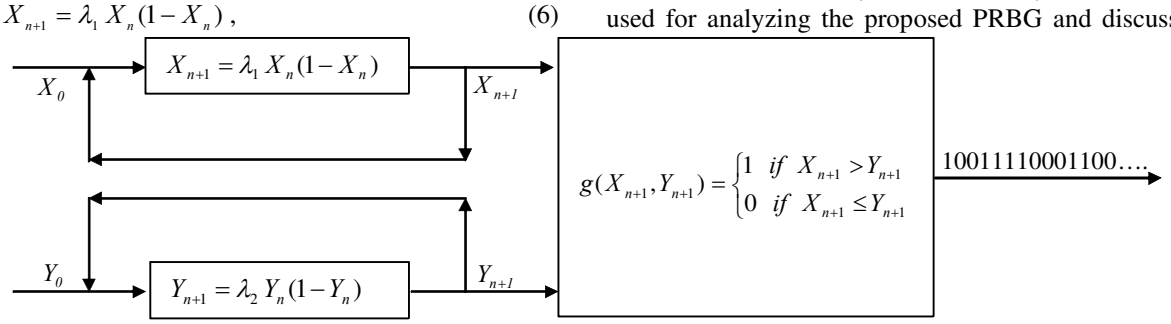

Figure 3: Schematic block diagram of the proposed pseudo random bit generator (PRBG).

results of our analysis in detail. It is to be noted here that the PRBG and the analysis proposed in [32] does not present any idea about the performance of PRBG in respect to the NIST test suite (whether successful or not). Hence we can not compare both the PRBGs in terms of their superiority/inferiority. However the idea of the present PRBG has emerged from the analytical study and properties reported in [32].

4 Statistical testing

In order to gain the confidence that newly developed pseudo random bit generators are cryptographically secure, they should be subjected to a variety of statistical tests designed to detect the specific characteristics expected of truly random sequences. There are several options available for analyzing the randomness of the newly developed pseudo random bit generators. The four most popular options are: (i) NIST suite of statistical tests [55], (ii) The DIEHARD suite of statistical tests [56], (iii) The Crypt-XS suite of statistical tests [57] and (iv) The Donald Knuth's statistical tests set [58]. There are different number of statistical tests in each of the above mentioned test suites to detect distinct types of non-randomness in the binary sequences. Various efforts based on the principal component analysis show that not all the above mentioned suites are needed to implement at a time as there are redundancy in the statistical tests (i.e., all the tests are not independent). The results also suggest that the NIST statistical tests suite contains a sufficient number of nearly independent statistical tests, which detect any deviation from the randomness [59]. Hence for analyzing the randomness of the proposed pseudo random bit generator (PRBG), we use the most stringent tests of randomness: the NIST suite tests. In the following subsection, we briefly mention the various statistical tests of NIST suite their focuses and purposes.

4.1 The NIST Tests Suite

The NIST tests suite is a statistical package comprising of 16 tests that are developed to test the randomness of (arbitrary long) binary sequences produced by either hardware or software based cryptographic random or pseudo random bit generators. These tests focus on a variety of different types of non-randomness that could exist in a binary sequence. Broadly, we may classify these sixteen tests into two categories: (i) non-parameterized tests and (ii) parameterized tests.

4.1.1 Non-parameterized tests

Frequency (monobit) test: The focus of the test is the proportion of zeroes and ones for the entire sequence. The purpose of this test is to determine 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 ½, that is, the number of ones and zeroes in a sequence should be the same.

Runs test: The focus of this test is the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits. A run of length k consists of exactly k identical bits and is bounded before and after with a bit of the opposite value. The purpose of the runs test is to determine 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.

Test for longest run of ones in a block: The focus of the test is the longest run of ones within M-bit blocks. The purpose of this test is to determine 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.

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.

Binary matrix rank test: The focus of the test is the rank of disjoint sub-matrices of the entire sequence. The purpose of this test is to check for linear dependence among fixed length substrings of the original sequence.

Cumulative sums test: The focus of this test is the maximal excursion (from zero) of the random walk defined by the cumulative sum of adjusted (-1, +1) digits in the sequence. The purpose of the test is to determine 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. This cumulative sum may be considered as a random walk. For a random sequence, the excursions of the random walk should be near zero.

Discrete Fourier transform (spectral) test: The focus of this test is the peak heights in the Discrete Fourier Transform of the sequence. The purpose of this test is to detect 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.

Random excursions test: The focus of this test is the number of cycles having exactly K visits in a cumulative sum random walk. The cumulative sum random walk is derived from partial sums after the (0,1) sequence is transferred to the appropriate (-1, +1) sequence. A cycle of a random walk consists of a sequence of steps of unit length taken at random that begin at and return to the origin. The purpose of this test is to determine 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 (and conclusions), one test and conclusion for each of the states: x = -4, -3, -2, -1 and +1, +2, +3, +4.

Random excursions variant test: The focus of this test is the total number of times that a particular state is visited (i.e., occurs) in a cumulative sum random walk. The purpose of this test is to detect deviations from the expected number of visits to various states in the random walk. This test is actually a series of eighteen tests (and

conclusions), one test and conclusion for each of the states: x = -9, -8, …, -1 and +1, +2, …, +9.

4.1.2 Parameterized tests

Frequency test within a block: The focus of the test is the proportion of ones within M-bit blocks. The purpose of this test is to determine whether the frequency of ones in an M-bit block is approximately M/2, as would be expected under an assumption of randomness. For block size M=1, this test degenerates to the Frequency (Monobit) test.

Approximate entropy test: The focus of this test is the frequency of all possible overlapping m-bit patterns across the entire sequence. The purpose of the test is to compare the frequency of overlapping blocks of two consecutive/adjacent lengths (m and $m+1$) against the expected result for a random sequence.

Linear complexity test: The focus of this test is the length of a linear feedback shift register (LFSR). The purpose of this test is to determine whether or not the sequence is complex enough to be considered random. Random sequences are characterized by longer LFSRs.

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.

Serial test: The focus of this test is the frequency of all possible overlapping m-bit patterns across the entire sequence. The purpose of this test is to determine 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.

Non-overlapping template matching test: The focus of this test is the number of occurrences of pre-specified target strings. The purpose of this test is to detect generators that produce too many occurrences of a given non-periodic (aperiodic) pattern. For this test and for the Overlapping Template Matching test, an m-bit window is used to search for a specific m-bit 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.

Overlapping template matching test: The focus of the Overlapping Template Matching test is the number of occurrences of pre-specified target strings. Both this test and the Non-overlapping Template Matching test use an m-bit window to search for a specific m-bit pattern. It differs from the non-overlapping template matching test in the sense that in this case when the pattern *is* found, the window slides only one bit before resuming the search.

For the detailed description of above mentioned 16 tests, we refer the readers to the NIST document [55].

4.2 Testing strategy

The NIST framework, like many statistical tests, is based on hypothesis testing. A hypothesis test is a procedure for determining if an assertion about a characteristic of a population is reasonable. In the present case, the test involves determining whether or not a specific sequence of zeroes and ones is random (it is called null hypothesis H_0).

For each test, a relevant randomness statistic be chosen and used to determine the acceptance or rejection of the null hypothesis. Under an assumption of randomness, such a statistic has a distribution of possible values. A theoretical reference distribution of this statistic under the null hypothesis is determined by mathematical methods and corresponding probability value (*P-value)* is computed, which summarizes the strength of the evidence against the null hypothesis. For each test, the *P-value* is the probability that a perfect random number generator would have produced a sequence less random than the sequence that was tested, given the kind of non-randomness assessed by the test. If a *P-value* for a test is determined to be equal to 1, then the sequence appears to have perfect randomness. A *P-value* equal to zero indicates that the sequence appears to be completely non-random. A significance level (α) be chosen for the tests and if $P-value \geq \alpha$, then the null hypothesis is accepted i.e., the sequence appears to be random. If $P-value < \alpha$, then the null hypothesis is rejected; i.e., the sequence appears to be non-random. Typically, the significance level (α) is chosen in the interval $[0.001, 0.01]$. The $\alpha = 0.01$ indicates that one would expect 1 sequence out of 100 sequences to be rejected. A $P-value \geq 0.01$ would mean that the sequence would be considered to be random with a confidence of 99 %.

For the numerical experimentations on the proposed pseudo random bit generator, we have generated 2000 (sample size $m = 2000$) different binary sequences (each sequence has been generated from a randomly chosen seed $X_0, Y_0 \in (0,1)$ with $X_0 \neq Y_0$ and $\lambda_1 = \lambda_2 = \lambda = 4.0$) each of length 10^6 bits and computed the *P-value* corresponding to each sequence for all the 16 tests of NIST Suite (in all we have computed total 48×2000 = 96000 *P-values*). All the computations have been performed in double precision floating point representation. We refer the readers to Rukhin et al [55] for the detailed mathematical procedure for calculating the *P-value* for each individual test of NIST suite. For the analysis of *P-values* obtained from various statistical tests, we have fixed the significance level at $\alpha = 0.01$. In Tables 1 and 2 respectively, we have summarized the results obtained after implementing non-parameterized and parameterized tests of NIST suite on the binary sequences produced by the proposed pseudo random bit generator.

4.3 Interpretation of results:

(i) *Uniform distribution of P-values:* For each test, the distribution of *P-values* for a large number of binary sequences ($m = 2000$) has been examined. Visually, it has been done by plotting the histograms, where we have divided the complete interval of *P-values* [0, 1] into 10

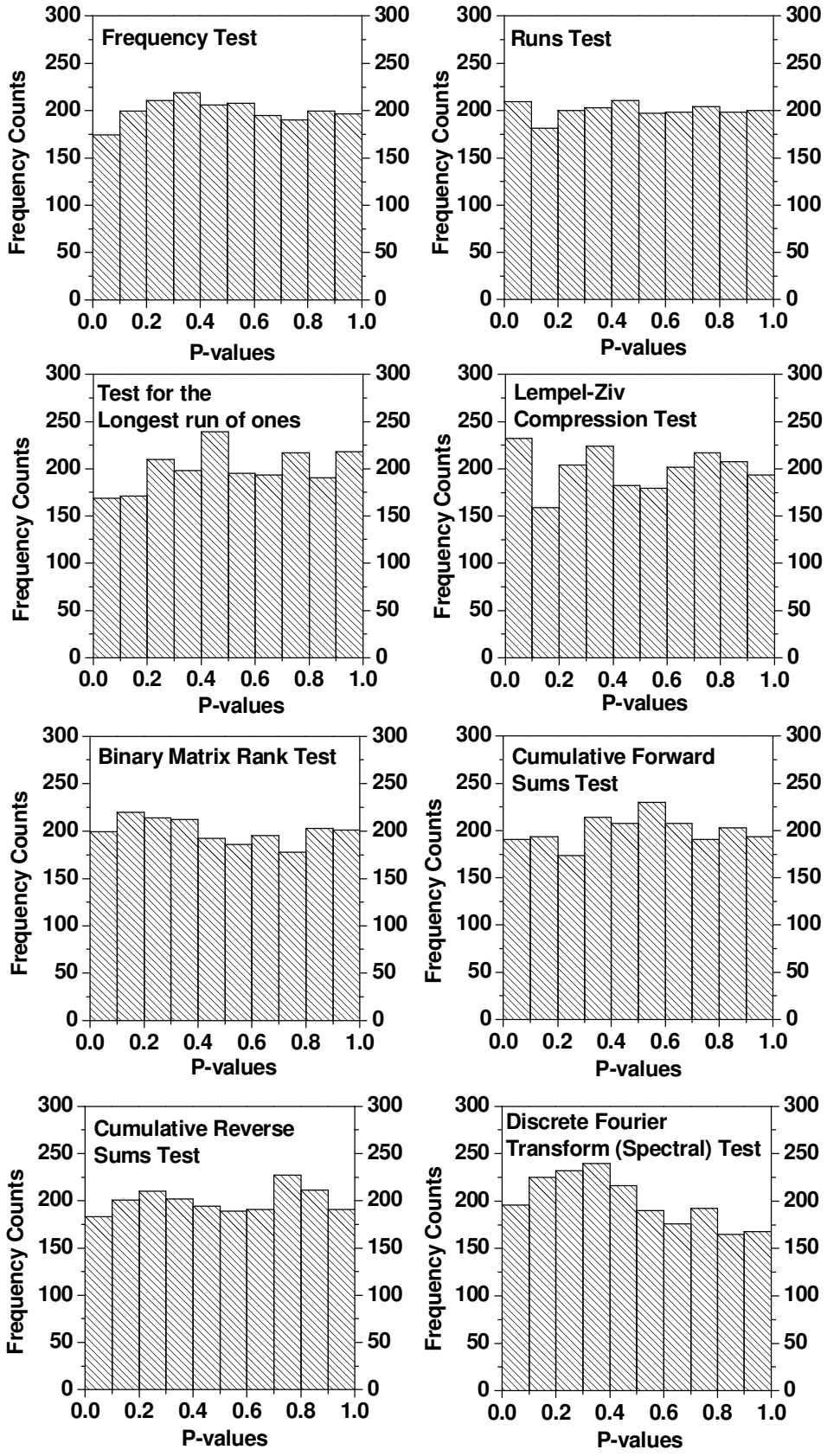

Figure 4: Histograms of *P-values* for non-parameterized tests of NIST suite.

equal sub-intervals and the *P-values* that lie in each subinterval has been counted and displayed. We have displayed the result of one such analysis in Figure 4 for some of the non-parameterized tests. It is clear from Figure 4 that the *P-values* for each statistical test are uniformly distributed in the complete interval of *P-values* i.e., [0, 1]. We obtain the similar results for the remaining non-parametric and parametric tests also.

The uniformity of the P-values has also been examined quantitatively via an application of χ^2 test and the determination of a *P-value* corresponding to the Goodness-of-Fit distributional test on the *P-values* obtained for each statistical test (i.e., a *P-value* of the *P-values*, which is denoted by $P-value_T$). The computation is as follows:

$$\chi^2 = \sum_{i=1}^{10} \left(f_i - \frac{m}{10} \right)^2 \Big/ \left(\frac{m}{10} \right), \quad (9)$$

where f_i is the number of P-values in the sub-interval i and m is the size of the sample, which is $m = 2000$ for the present analysis. The *P-value* of the *P-values* (i.e., $P-value_T$) is obtained from the χ^2 by using

$$P-value_T = igamc\left(\frac{9}{2}, \frac{\chi^2}{2} \right), \quad (10)$$

where *igamc*( ) is the incomplete Gamma function. If $P-value_T \geq 0.0001$ then the P-values are considered to be uniformly distributed.

The computed $P-value_T$ corresponding to each statistical test has been given in Tables 1 and 2. In Figures 5(a) and (b) respectively, we have graphically depicted the computed $P-value_T$ for each non-parameterized and parameterized test along with the threshold value (0.0001). It is clear that the computed $P-value_T$ for each test lies above the threshold value, which confirms the uniformity of the *P-values* for all the 16 tests of NIST suite.

(ii) *Proportions of the sequences passing the tests:* We have calculated the proportion of the sequences passing a particular statistical test and compared it with the range of acceptable proportion. The range of acceptable proportion is determined by using the confidence interval given by $\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}$, (11)

where m is the sample size and $\hat{p} = 1 - \alpha$, which are $m = 2000$ and $\hat{p} = 1 - 0.01 = 0.99$ for the present analysis. So the range of acceptable proportion is $[0.9833245, 0.9966745]$. The quantitative results of proportions are given the Tables 1 and 2 respectively for

Table 1: Non-parameterized tests results.

- Number of binary sequences tested (m): 2,000
- Length of each binary sequence: 1,000,000 bits
- Significance level (α) = 0.01
- The range of acceptable proportion is [0.9833245, 0.9966745]

- Null hypothesis (H_0): The binary sequence is random
- If $P-value \geq \alpha(0.01)$ then the null hypothesis (H_0) is accepted.
- If $P-value < \alpha(0.01)$ then the null hypothesis ( ) is rejected.
- If $P-value_T$ ($P-value$ corresponding to the Goodness-of-Fit distributional test on the $P-values$ obtained for a particular test i.e., a $P-value$ of the $P-values$) ≥ 0.0001 then $P-values$ can be considered uniformly distributed.

S. No.	Statistical Test	No. of sequences with $P-values \geq 0.01$ (Success) (m_p)	No. of sequences with $P-values < 0.01$ (Failure) (m_f)	χ^2 of distribution of $P-values$	$P-value$ corresponding to the goodness of fit ($P-value_T$)	Proportion of sequences passing the test (m_p / m)
1.	Frequency (monobit) test	1982	18	7.0152	0.635558	0.9910
2.	Runs test	1982	18	2.92	0.967382	0.9910
3.	Test for longest run of ones in a block	1979	21	21.07	0.0123432	0.9895
4.	Lempel-Ziv compression test	1977	23	22.34	0.00786166	0.9885
5.	Binary matrix rank test	1978	22	7.6	0.574903	0.9890
6.	Cumulative sums test					
	1) Forward sums test	1983	17	11.15	0.265567	0.9915
	2) Reverse sums test	1981	19	7.76	0.558502	0.9905
7.	Discrete Fourier transform (spectral) test	1990	10	32.55	0.000159908	0.9950
8.	Random excursions test					
	1) x = -4	1977	23	21.6	0.0102369	0.9885
	2) x = -3	1971	29	7.42	0.593478	0.9855
	3) x = -2	1983	17	20.74	0.0138562	0.9915
	4) x = -1	1977	23	25.54	0.00242845	0.9885
	5) x = 1	1980	20	20.35	0.0158711	0.9900
	6) x = 2	1990	10	20.93	0.0129649	0.9950
	7) x = 3	1983	17	19.91	0.018476	0.9915
	8) x = 4	1987	13	15.93	0.0683578	0.9935
9.	Random excursions variant test					
	1) x = -9	1991	9	17.31	0.044077	0.9955
	2) x = -8	1985	15	26.87	0.00146972	0.9925
	3) x = -7	1990	10	11.15	0.265567	0.9950
	4) x = -6	1988	12	10.795	0.290023	0.9940
	5) x = -5	1981	19	15.59	0.075953	0.9905
	6) x = -4	1974	26	13.78	0.130369	0.9870
	7) x = -3	1978	22	7.64	0.570792	0.9890
	8) x = -2	1979	21	11.45	0.24612	0.9895

various non-parameterized and parameterized statistical tests of NIST suite. In Figures 6(a) and (b) respectively, we have graphically depicted the computed proportions for each non-parameterized and parameterized test along with the confidence interval i.e., $[0.9833245, 0.9966745]$. It is clear that the computed proportion for each test lies inside the confidence interval; hence the tested binary sequences generated by the proposed PRBG are random with respect to all the 16 tests of NIST suite.

5 Conclusion

We have proposed a design of a pseudo random bit generator (PRBG) based on two chaotic logistic maps iterated independently starting from independent initial conditions. The pseudo random bit sequence is obtained by comparing the outputs of both the chaotic logistic maps. We have also tested rigorously the generated sequences using the NIST suite, which consists of 16 independent statistical tests devised to detect the specific characteristics expected of truly random bit sequences. The results of statistical testing are encouraging and show that the proposed PRBG has perfect cryptographic properties and hence can be used in the design of new stream ciphers.

References

[1] Schneier B. *Applied Cryptography-Protocols, algorithms and source code in C.* John Wiley & Sons, New York, USA, 1996.

[2] Menezes A.J., Oorschot P.C.V. and Vanstone S.A. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, 1997.

[3] Alvarez G. and Li S. *Some basic cryptographic requirements for chaos based cryptosystems*. Int. J. of Bifur. and Chaos, vol. 16, pp. 2129-2151, 2006.

[4] Pareek N. K., Patidar Vinod and Sud K. K. *Discrete chaotic cryptography using external secret key.* Physics Letters A, vol. 309, pp. 75-82, 2003.

[5] Boccaletti S., Grebogi C., Lai Y.-C., Mancini H. and Maza D. *The control of chaos: theory and applications.* Phys. Reports, vol. 329, pp. 103-197, 2000.

[6] Schuster H. G. (Ed.) *Hand book of chaos control.* Wiley-VCH Verlag, Weinheim, Germany, 1999.

[7] Kocarev, L., Halle, K. S., Eckert, K., Chua, L. O. & Parlitz, U. *Experimental demonstration of secure communications via chaotic synchronization.* Int. J. Bifurc. Chaos, vol. 2, pp. 709-713, 1992.

[8] Wu, C. W. & Chua, L. O. *A simple way to synchronize chaotic systems with applications to secure communications systems.* Int. J. Bifurc. Chaos, vol. 3, pp. 1619–1627, 1993.

[9] Cuomo, K. M., Openheim, A. V. & Strogatz, S. H. *Synchronization of lorenz-based chaotic circuits with applications to communications.* IEEE Trans. Circuits Syst. II, vol. 40, pp. 626–633, 1993.

[10] Morgul, O. & Feki, M. *A chaotic masking scheme by using synchronized chaotic systems*. Phys. Lett. A, vol. 251, pp. 169–176, 1999.

[11] Shahruz, S. M., Pradeep, A. K. & Gurumoorthy, R.

Table 2: Non-parameterized tests results

- Number of binary sequences tested (m): 2,000
- Length of each binary sequence: 1,000,000 bits
- Significance level (α) = 0.01
- The range of acceptable proportion is [0.9833245, 0.9966745]

- Null hypothesis (H_0): The binary sequence is random
- If $P-value \geq \alpha(0.01)$ then the null hypothesis (H_0) is accepted.
- If $P-value < \alpha(0.01)$ then the null hypothesis (H_0) is rejected.
- If $P-value_T$ ($P-value$ corresponding to the Goodness-of-Fit distributional test on the $P-values$ obtained for a particular test i.e., a $P-value$ of the $P-values$) ≥ 0.0001 then $P-values$ can be considered to be uniformly distributed.

S. No.	Statistical Test	No. of sequences with $P-values \geq 0.01$ (Success) (m_p)	No. of sequences with $P-values < 0.01$ (Failure) (m_f)	χ^2 of distribution of $p-values$	$P-value$ corresponding to the goodness of fit ($P-value_T$)	Proportion of sequences passing the test (m_p/m)
1.	Frequency test within a block (Block length = 10^4)	1977	23	8.79	0.456881	0.9885
2.	Approximate entropy test (Block length = 10)	1980	20	4.08	0.906069	0.9900
3.	Linear complexity test (Block length = 10^3)	1977	23	13.92	0.1252	0.9885
4.	Maurer's universal statistical test (No. of blocks = 7, Block length = 1280)	1978	22	9.37	0.403844	0.9890
5.	Serial test (Block length = 16)	1979	21	3.65	0.932904	0.9895
6.	Overlapping template matching test (Template length = 9)	1977	23	9.65	0.379555	0.9885
7.	Non-overlapping template matching test (Template length = 9)					
	1) Template = 000000001	1987	13	9.98	0.352107	0.9935
	2) Template = 000100111	1974	26	5.8	0.759756	0.9870
	1) Template = 001010011	1981	19	10.02	0.348869	0.9905
	2) Template = 010001011	1978	22	7.39	0.596584	0.9890
	3) Template = 011101111	1981	19	12.04	0.211064	0.9905
	4) Template = 101101000	1974	26	8.27	0.507182	0.9870
	5) Template = 110100100	1981	19	8.66	0.469232	0.9905
	6) Template = 111100000	1976	24	6.01	0.738917	0.9880

Design of a novel cryptosystem based on chaotic oscillators and feedback inversion. J. Sound Vibrat., vol. 250, pp. 762–771, 2002.

[12] Halle, K. S., Wu, C. W., Itoh, M. & Chua, L. O. *Spread spectrum communication through modulation of chaos in Chua's circuit*. Int. J. Bifurc. Chaos, vol. 3, pp. 469–477, 1993.

[13] Cuomo, K. M. & Openheim, A. V. *Circuit implementation of synchronized chaos with applications to communications*. Phys. Rev. Lett., vol.71, pp. 65–68, 1993.

[14] Chen, J. Y., Wong, K. W., Cheng, L. M. & Shuai, J. W. *A secure communication scheme based on the phase synchronization of chaotic systems*. Chaos, vol. 13, pp. 508–514, 2003.

[15] Yang, T. & Chua, L. O. *Secure communication via chaotic parameter modulation*. IEEE Trans. Circuits Syst. I , vol. 43, pp. 817–819, 1996.

[16] Dedieu, H., Kennedy, M. P. & Hasler, M. *Chaos shift keying: Modulation and demodulation of a chaotic carrier using self-synchronizing*. IEEE Trans. Circuits Syst. II , vol. 40, pp. 634–641, 1993.

[17] Parlitz, U., Chua, L. O., Kocarev, L., Halle, K. S. & Shang, A. *Transmission of digital signals by chaotic synchronization*. Int. J. Bifurc. Chaos , vol. 2, pp. 973–977, 1992.

[18] Feldmann, U., Hasler, M. & Schwarz, W. *Communication by chaotic signals: The inverse system approach*. Int. J. Circuit Theory Appl. , vol. 24, pp. 551–579, 1996.

[19] Zhou, H. & Ling, X. *Problems with the chaotic inverse system encryption approach*. IEEE Trans. Circuits Syst. I, vol. 44, pp. 268–271, 1997.

[20] Habutsu, T., Nishio, Y., Sasase, I. & Mori, S. *A secret key cryptosystem by iterating a chaotic map*. in Advances in Cryptology – EUROCRYPT'91, Lecture Notes in Computer Science, vol. 547, pp. 127–140, 1991 (Spinger-Verlag).

[21] Pareek N. K., Patidar Vinod and Sud K. K. *Cryptography using multiple one-dimensional chaotic maps*. Communications in Nonlinear Science and Numerical Simulation, vol. 10, pp. 715-723, 2005.

[22] Pareek N. K., Patidar Vinod and Sud K. K. *Image encryption using chaotic logistic map*. Image and Vision Computing, vol. 24, pp. 926-934, 2006.

[23] Fridrich, J. *Symmetric ciphers based on two-dimensional chaotic maps*. Int. J. Bifurc. Chaos, vol. 8, pp. 1259–1284, 1998.

[24] Tang, G., Liao, X. & Chen, Y. A novel method for designing S-boxes based on chaotic maps. Chaos Solitons Fractals, vol. 23, pp. 413–419, 2005.

[25] Kocarev, L., Jakimoski, G., Stojanovski, T. & Parlitz, U. *From chaotic maps to encryption schemes*. in Proc. IEEE Int. Symposium Circuits and Systems (ISCAS'98), vol. 4, pp. 514–517,1998.

[26] Guo, D., Cheng, L. M. & Cheng, L. L. *A new symmetric probabilistic encryption scheme based on chaotic attractors of neural networks*. Applied Intelligence, vol. 10, pp. 71–84, 1999.

[27] Jakimoski, G. & Kocarev, L. *Chaos and cryptography: Block encryption ciphers based on chaotic maps*. IEEE Trans. Circuits Syst. I, vol. 48,

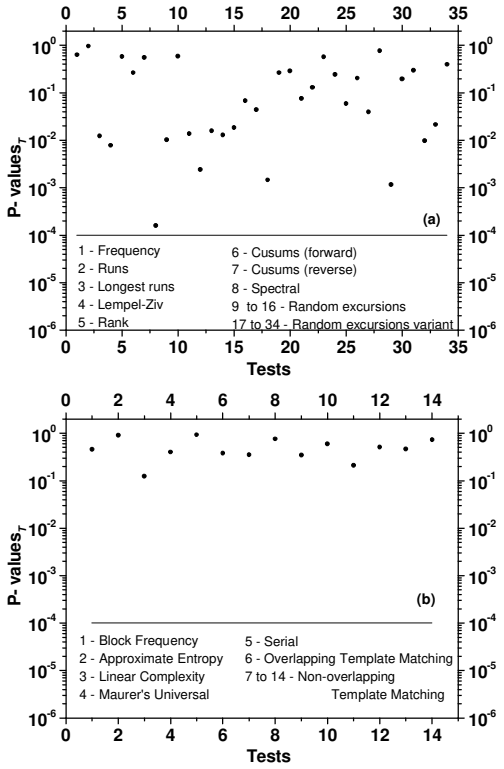

Figure 5: $P-value_T$ (i.e, *P-value* of the *P-values*) for (a) non-parameterized tests and (b) parameterized tests of NIST suite. The horizontal line represents the threshold value of $P-value_T$.

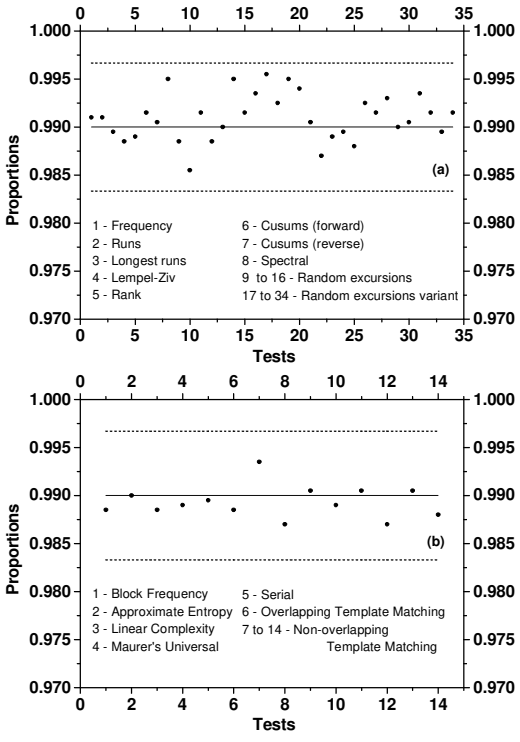

Figure 6: Proportions of the sequences passing the tests for (a) non-parameterized and (b) parameterized tests of NIST suite. The region between two horizontal dashed lines is the acceptable range of proportion.

pp. 163–169, 2001.

[28] Wolfram, S. *Cryptography with cellular automata*. in Advances in Cryptology – CRYPTO'85, Lecture Notes in Computer Science, vol. 218, pp. 429–432, 1985.

[29] Matthews, R. A. J. *On the derivation of a 'chaotic' encryption algorithm*. Cryptologia, vol. XIII, 29–42, 1989.

[30] Berstein, G. M. & Lieberman, M. A. *Method and apparatus for generating secure random numbers using chaos*. US Patent No. 5007087, 1991.

[31] Zhou, H. & Ling, X. *Generating chaotic secure sequences with desired statistical properties and high security*. Int. J. Bifurc. Chaos, vol. 7, 205–213, 1997.

[32] Li, S., Mou, X. & Cai, Y. *Pseudo-random bit generator based on couple chaotic systems and its application in stream-ciphers cryptography*. in Progress in Cryptology – INDOCRYPT 2001, Lecture Notes in Computer Science, vol. 2247, 316–329, 2001.

[33] Lee, P.-H., Pei, S.-C. & Chen, Y.-Y. *Generating chaotic stream ciphers using chaotic systems*. Chinese J. Phys. , vol. 41, 559–581, 2003.

[34] Oishi S. and Inoue H. *Pseudo-random number generators and chaos*. Transactions of the Institute of Electronics and Communication Engineers of Japan E, vol. 65, 534-541, 1982.

[35] Lin T. and Chua L. O. *New class of pseudo-random number generator based on chaos in digital filters*. International Journal of Circuit Theory and Applications, vol. 21, 473-480, 1993.

[36] Andrecut M. [1998] Logistic map as a random number generator, International Journal of Modern Physics B, vol. 12, 921-930.

[37] Gonzalez J. A..and Pino R. *Random number generator based on unpredictable chaotic functions*. Computer Physics Communications, vol.120, 109-114, 1999.

[38] Kolesov V. V., Belyaev R. V. and Voronov G. M. *A Digital random-number generator based on the chaotic signal algorithm*. Journal of Communications Technology and Electronics, vol. 46, pp. 1258-1263, 2001.

[39] Stojanovski T. and Kocarev L. *Chaos-based random number generators - Part I: Analysis.* IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 48, pp. 281-288, 2001.

[40] Stojanovski T., Pihl J. and Kocarev L. Chaos-*based random number generators - Part II: Practical realization.* IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 48, pp. 382-385, 2001.

[41] Kocarev L. and Jakimoski G. *Pseudorandom bits generated by chaotic maps*. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 50, pp. 123-126, 2003.

[42] Fu S. –M., Chen Z. –Y. and Zhou Y. –A. *Chaos-based random number generators*. Computer Research and Development, vol. 41, pp. 749-754, 2004.

[43] Huaping L., Wang S. and Gang H. *Pseudo-random number generator based on coupled map lattices*. International Journal of Modern Physics B, vol. 18, pp. 2409-2414, 2004.

[44] Li X. –M., Shen H. –B. and Yan X. –L. *Characteristic analysis of a chaotic random number generator using piece-wise-linear map.* Journal of Electronics and Information Technology, vol. 27, pp. 874-878, 2005.

[45] Liu J. *Design of a chaotic random sequence and its application*, Computer Engineering, vol.31, pp. 150-152, 2005.

[46] Wang Y., Shen H. and Yan X. *Design of a chaotic random number generator*. Chinese Journal of Semiconductors, vol. 26, pp. 2433-2439, 2005.

[47] Wang L., Wang F. –P. and Wang Z. –J. *Novel chaos-based pseudo-random number generator*. Acta Physica Sinica, vol. 55, pp. 3964-3968, 2006.

[48] Ergun S. and Ozoguz S. *Truly random number generators based on a non-autonomous chaotic oscillator*. AEU-International J. Electronics & Communications, vol. 62, pp. 235-242, 2007.

[49] Hu Y., Liao X., Wong K.-W. and Zhou Q. *A true random number generator based on mouse movement and chaotic cryptography*. Chaos Solitons and Fractals, vol. 40, pp. 2286-2293, 2009.

[50] Patidar Vinod and Sud K. K. *Anovel pseudo random bit generator based on chaotic standard map and its testing.* Electronic J. of Theoretical Physics, vol. 20, pp. 327-344, 2009.

[51] May R.M. *Simple mathematical models with very complicated dynamics*. Nature, vol. 261, pp. 459-467, 1976.

[52] Feigenbaum M. J. *The universal metric properties of nonlinear transformations.* J. Stat. Phys., vol. 21, pp. 669-706, 1979.

[53] Feigenbaum M. J. *Universal behaviour in nonlinear systems*. Los Alamos Science, vol. 1, pp. 4-27, 1980.

[54] Litchenberg A. J. and Lieberman M. A. *Regular and stochastic motion*. Springer Verlag, New York, USA, 1983.

[55] Runkin et al. *Statistical test suite for random and pseudo random number generators for cryptographic applications*. NIST special publication 800-22, 2001.

[56] Marsaglia G. *DIEHARD statistical tests, http://stst.fsu.edu/pub/diehard*, 1995.

[57] Gustafson H. et al. *A computer package for measuring the strength of encryption algorithms*, J. Computer Security, vol. 13, pp. 687-697, 1994.

[58] Knuth D. *The art of computer programming: semiemperical algorithms*. Addison Wesley, Reading, USA, 1998

[59] Sato J. *Statistical testing of random number generators*. Proceedings of 22nd National Information System Security Conference, http://csrc.ncsl.gov/nissc/1999/proceeding/papers/p24.pdf