Commun Nonlinear Sci Numer Simulat 19 (2014) 101–111

Contents lists available at SciVerse ScienceDirect

Commun Nonlinear Sci Numer Simulat

journal homepage: www.elsevier.com/locate/cnsns

Pseudo random number generator based on quantum chaotic map

A. Akhshani [a,*], A. Akhavan [b], A. Mobaraki [c], S.-C. Lim [a], Z. Hassan [a]

[a] School of Physics, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia
[b] School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia
[c] University College of Science and Technology (UCST), Orumieh, Iran

ARTICLE INFO

Article history:
Received 1 October 2012
Received in revised form 28 April 2013
Accepted 6 June 2013
Available online 19 June 2013

Keywords:
Quantum map
Pseudo-random number generator
Non-periodicity
Statistical complexity
Cryptography
Differential attack
Quantum key distribution

ABSTRACT

For many years dissipative quantum maps were widely used as informative models of quantum chaos. In this paper, a new scheme for generating good pseudo-random numbers (PRNG), based on quantum logistic map is proposed. Note that the PRNG merely relies on the equations used in the quantum chaotic map. The algorithm is not complex, which does not impose high requirement on computer hardware and thus computation speed is fast. In order to face the challenge of using the proposed PRNG in quantum cryptography and other practical applications, the proposed PRNG is subjected to statistical tests using well-known test suites such as NIST, DIEHARD, ENT and TestU01. The results of the statistical tests were promising, as the proposed PRNG successfully passed all these tests. Moreover, the degree of non-periodicity of the chaotic sequences of the quantum map is investigated through the Scale index technique. The obtained result shows that, the sequence is more non-periodic. From these results it can be concluded that, the new scheme can generate a high percentage of usable pseudo-random numbers for simulation and other applications in scientific computing.

1. Introduction

The nature of randomness has attracted an increasing amount of interest in recent years. Many applications require random input. Sources of random numbers can be broadly divided into two classes. The first of these is the pseudo-random number generators (PRNGs) and the second is true random number generators (TRNGs). The primary difference between random and pseudo-random numbers is that pseudo-random numbers are necessarily periodic whereas truly random numbers are not. Also, pseudo-random number generators are deterministic algorithms. In most of the scientific fields, the first one is particularly desirable feature for some applications, such as simulations of stochastic processes, statistical sampling and performance evaluation of computer algorithms and Monte Carlo simulation. True random number generators are further classified into physical and non-physical. This kind of random number generator is often called non-deterministic random number generator since the next number to be generated cannot be determined in advance. Many true random number generators are relatively slow. This paper is focused on PRNGs.

Cryptography is the art of protecting information from any unauthorized access. The central aim of cryptography is to enable two parties to communicate in a secure manner. Cryptographic methods that do not utilize quantum laws fall under classical cryptography.

* Corresponding author.
E-mail address: a.akhshani@yahoo.com (A. Akhshani).

1007-5704/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cnsns.2013.06.017

Quantum communication, the branch of quantum information provides several examples of communication protocols which cannot be implemented securely only classical communication. The most widely known of these is quantum cryptography, which allows secure key exchange between parties sharing a quantum channel subject to an eavesdropper. The best known protocol of this type is quantum key distribution (QKD). Quantum key distribution involves the communication of a secure cryptographic key between two parties, Alice and Bob, in remote locations by exploiting the laws of quantum mechanics [1]. QKD allows Alice and Bob to make this exchange over a quantum channel, even if it is controlled by the eavesdropper. In fact, Alice and Bob share a pseudo-random number generator that is used to generate a sheared secret key. Bennett and Brassard had provided the first example of the QKD protocol in 1984 [2]. Their protocol is called BB84. Quantum cryptography systems have been demonstrated operating at speeds of up to 1.25 GHz. True random number generators are not available that operate at this speed, so these systems must use pseudo-random number generators [3].

This is an intrinsic weakness in the system because it relies on a random number generator [4]. The NIST system [5], for example, relies on a Mersenne Twister [6] pseudo-random number generator to create Alice's random bits. The Mersenne Twister is computationally fast and is often used for Monte Carlo simulations, but from a security perspective it is considered unsuitable for cryptography because it only requires observation of 624 iterates to determine all of the parameters needed to predict its next output bit [6]. However, the current standard in cryptographically secure random bits is the Blum Blum Shub (BBS) algorithm [7]. The security of the BBS algorithm is based on the difficulty of factoring prime numbers. But quantum cryptography is important precisely because it does not rely on the difficulty of factoring composite numbers, so choosing PRNG based on such condition seems counter-productive.

One obvious solution would be to use a more secure PRNG. Quantum chaos theory seems to be a tool that can be used to improve the quality of pseudo-random number generators. The word quantum chaos refers to quantum systems which in the classical limit show chaotic dynamics. However, the usage of this phrase even for systems like atomic nuclei, which do not possess a classical limit, is now quite wide spread. Also, chaos in systems with discrete phase spaces is called pseudo-chaos or quantum chaos [8]. Besides quantum mechanics, there are examples of pseudo-chaos abound in digital computers, which behave like discrete classical dynamical systems [9,10]. One aim of the field of quantum chaos is the study of quantum versions of classically chaotic systems. As in classical chaos theory, simple chaotic quantum maps [11] have turned out to provide deep insight into the nature of quantum chaos. Quantum maps have been much studied in the last 25 years as convenient toy models of "quantum chaos" [12,13].

In this paper, a novel pseudo-random number generator based on the quantum chaotic map [14] is proposed. In fact, this quantum map is the logistic map with additive noise that arises from the very lowest-order quantum corrections [15,16]. Note that, the PRNG merely relies on the equations used in the quantum chaotic map. To ensure that a random number generator is secure, its output must be statistically proven unpredictable and indistinguishable from a true random sequence. Several tests are used in order to test the randomness of the presented algorithm. These tests include TestU01 [17], DIEHARD [18], NIST statistical test suite [5] and Entropy test suite [19]. To apply statistical random tests such as SP800-22 and DIEHARD, a sufficiently large size of data is required. If the statistical tests are conducted on small size samples, then tests will yield an inaccurate inference. The tests of TestU01 are grouped into three batteries, small crush, crush, and big crush, which are used to test the quality of the proposed PRNG. The new PRNG passed all tests in TestU01 including linear complexity tests that all linear feedback shift-register (LFSR) and generalized feedback shift-register (GFSR)-based random number generators fail (see [17] for more details). Furthermore, in order to pass random number statistical tests, there is no post-processing procedure which makes it an extremely simple generator. The presented quantum based PRNG passes all the standard statistical tests; therefore, it can be used for any application that requires randomness such as cryptographic applications.

Although there exist very powerful and stringent benchmark for testing PRNGs, but the importance of the main statistical characteristics of a chaotic map should be considered. In this vein, a few quantifiers for measuring the main statistical properties of chaotic PRNGs are proposed. They use mainly two kinds of procedures: (1) quantifiers based on information theory [20–22], (2) quantifiers based on recurrence plots [23,24]. The quantifiers based on information theory are Normalized Shannon Entropy and statistical complexity measure [25]. For the quantifiers based on recurrence plots, several measures to quantify the recurrence plots' characteristics are presented [24]. Because of the "small-scale" structures the visual impact produced by the recurrence plot is insufficient to compare the quality of different PRNGs [25].

In this paper, the randomness of the proposed PRNG is successfully verified by statistical complexity and the normalized Shannon entropy. Moreover, in order to the study of non-periodicity in the chaotic sequences of the quantum map, the Scale index analysis based on Continuous Wavelet Transform (CWT) is carried out [26]. Also, the proposed PRNG is highly resistive against different types of attacks such as brute-force attacks and differential attacks.

2. Quantum chaotic map

This section briefly reviews of quantum logistic map. Quantum logistic map is presented by Goggin et al. in 1990 [14]. In their rich work, a kicked quantum system coupled to a bath of oscillators and a logistic map with very lowest-order quantum corrections is derived. In order to studying the effects of quantum correlations on a dissipative system they start with the Hamiltonian of a kicked quantum system coupled to a bath [14]. The operators used were the well-known boson creation ($a^\dagger$) and annihilation (a) operators. In fact, the dynamics of a dissipative quantum map which quantum corrections effectively add noise, become more classical as the dissipation (β) increased. The effects of quantum corrections were made by

introducing $a = \langle a\rangle + \delta a$, where δa represents a quantum fluctuation about $\langle a\rangle$ [14]. This chaotic map is governed by the following equations:

$$\begin{cases} x_{n+1} = r(x_n - |x_n|^2) - ry_n, & (1.a)\\ y_{n+1} = -y_n e^{-2\beta} + e^{-\beta} r[(2 - x_n - x_n^\star)y_n - x_n z_n^\star - x_n^\star z_n], & (1.b)\\ z_{n+1} = -z_n e^{-2\beta} + e^{-\beta} r[2(1 - x_n^\star)z_n - 2x_n y_n - x_n]. & (1.c) \end{cases} \tag{1}$$

where $x = \langle a\rangle$, $y = \langle \delta a^\dagger \delta a\rangle$ and $z = \langle \delta a \delta a\rangle$. In this dynamical system β is dissipation parameter, $x^\star$ and $z^\star$ are complex conjugates of x and z respectively. If, however, we set the initial values to be real numbers then all successive values will also be real. The range of the parameters as follows: $x \in [0,1]$, $y \in [0,0.1]$, $z \in [0,0.2]$, $x^\star = x$, $z^\star = z$, $\beta \in [6,\infty)$ and $r \in [0,4]$. It should be noted that, in this study, the intermediate values of β and y_n, $z_n \neq 0$ are considered. Eq. (1.a) has the same form as the logistic map with additive noise. In fact, the noise here is a measure of the strength of quantum corrections. Eq. (1) reduce to the classical, one dimensional logistic map when the quantum corrections y_n and $z_n \longrightarrow 0$. The quantum map exhibits a period doubling route to chaos.

3. Degree of non-periodicity

In this section, first the Scale index which is presented by Benìtez et al. [26] is briefly reviewed, then in order to detect and study non-periodicity in the chaotic sequences of the quantum map, the scale index analysis is carried out.

The Scale index technique is based on the continuous wavelet transform and the wavelet multi-resolution analysis [35]. Because of the non-stationary nature of chaotic sequences wavelets are more suitable to study non-periodicity [36]. The Continuous Wavelet Transform (CWT) of f at time u and scale s is defined as [35]:

$$Wf(u,s) := \langle f, \psi_{u,s}\rangle = \int_{-\infty}^{+\infty} f(t)\psi_{u,s}^*(t)\mathrm{d}t, \tag{2}$$

The *scalogram* of $f, \mathcal{S}$, is defined as follows:

$$\mathcal{S}(s) := \|Wf(u,s)\| = \left(\int_{-\infty}^{+\infty} |Wf(u,s)|^2 \mathrm{d}u\right)^{\frac{1}{2}}.$$

$\mathcal{S}(s)$ is the energy of the continuous wavelet transform of f at scale s. The scalogram is a useful tool for studying a signal, since it allows the detection of its most representative scales or frequencies [26,36]. Also, the *innerscalogram* of f at a scale s can be defined by:

$$\mathcal{S}^{\text{inner}}(s) := \|Wf(s,u)\|_{J(s)} = \left(\int_{c(s)}^{d(s)} |Wf(s,u)|^2 \mathrm{d}u\right)^{\frac{1}{2}},$$

where $J(s) = [c(s), d(s)] \subseteq I$ is the maximal subinterval in I for which the support of $\psi_{u,s}$ is included in I for all $u \in J(s)$. As regards the length of $J(s)$ depends on the scale s, so that the values of the inner scalogram at different scales cannot be compared. Therefore, the inner scalogram should be normalized as follows [26]:

$$\overline{\mathcal{S}}^{\text{inner}}(s) = \frac{\mathcal{S}^{\text{inner}}(s)}{(d(s) - c(s))^{\frac{1}{2}}}.$$

The Scale index of f in the scale interval $[s_0, s_1]$ can be defined by: quotient

$$i_{\text{scale}} := \frac{\mathcal{S}(s_{\min})}{\mathcal{S}(s_{\max})},$$

where $s_{\max}$ is the smallest scale such that $\mathcal{S}(s) \leqslant \mathcal{S}(s_{\max})$ for all $s \in [s_0, s_1]$, and $s_{\min}$ the smallest scale such that $\mathcal{S}(s_{\min}) \leqslant \mathcal{S}(s)$ for all $s \in [s_{max}, s_1]$. Note that for compactly supported signals only the normalized inner scalogram will be considered [26]. From its definition, the scale index i_{scale} is such that $0 \leqslant i_{\text{scale}} \leqslant 1$ and it can be interpreted as a measure of the degree of non-periodicity of the signal: the scale index will be zero or close to zero for periodic sequences and close to one for highly non-periodic sequences [26].

Since the scale index gives a measure of the degree of non-periodicity of the signal so that, this can be used to specify which values of the chaotic map parameters are best for generation of pseudo-random number sequences.

Figs. 1 and 2 show the bifurcation diagram and the Lyapunov exponents of the quantum map for different values of control parameter (r) and the dissipation parameter (β). As shown in Figs. 1 and 2, for $r > 3.8$ the system can exhibit chaotic behavior. There is a periodic attractor when $r = 3.85$. In a certain range of values of the control parameter, $r > 3.85$, full chaotic behavior can be seen. In Fig. 3, the scale index analysis of the map is presented. From Figs. 1–3 it is obvious that, there is a correspondence between the regions where the Lyapunov exponent is positive, the chaotic regions of the bifurcation diagram and the regions where the Scale index is positive. From the definition, for highly non-periodic signal the scale index will be close to 1. So that, from Fig.3 it can be concluded that, the best values of the control parameter (r) and dissipation

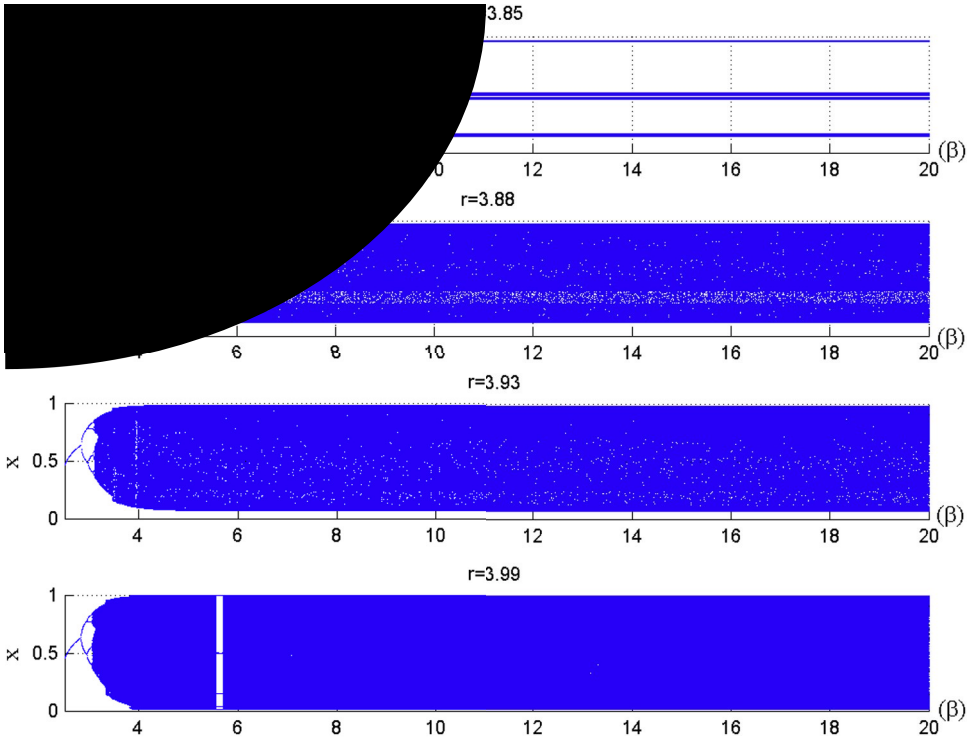

Fig. 1. Bifurcation diagram of the quantum map for different control parameter (r).

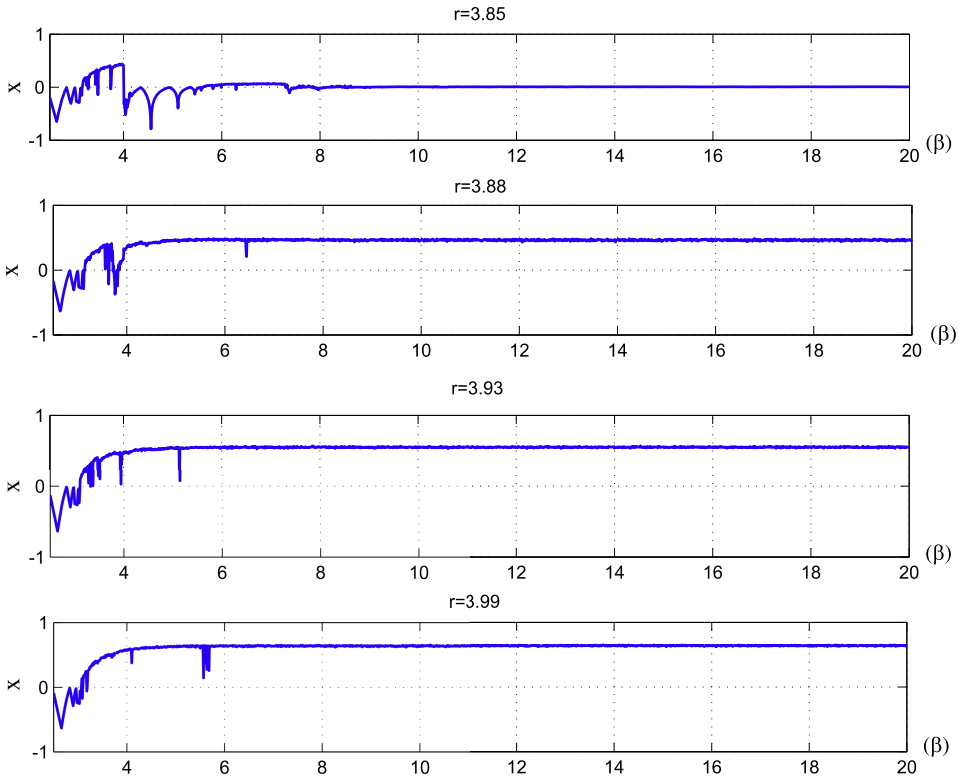

Fig. 2. The Lyapunov exponent of the quantum map for different control parameter (r).

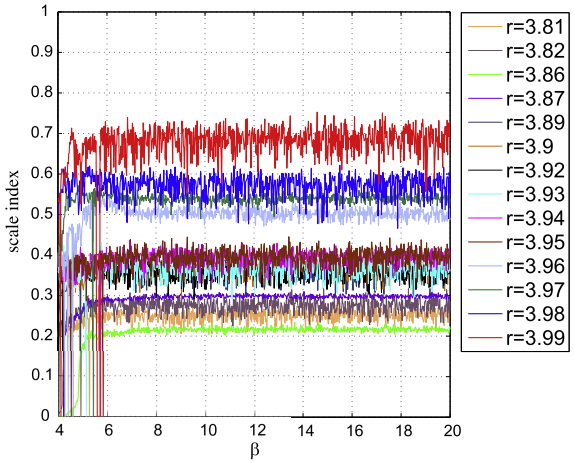

Fig. 3. The Scale index of the quantum map for different control parameter (r).

parameter (β) are $r = 3.99$ and $\beta \geqslant 6$ respectively. When these values are considered, the Scale index becomes maximum ($i_{scale} \approx 0.7$) and remain at this value for all $\beta \geqslant 6$. Thus, the sequence in this state is highly non periodic and it can be used for any PRNG purposes.

3.1. Comparison of the non-periodicity

In this section, the comparison between quantum map and other high dimensional chaotic map such as Henon map and Rössler system is presented. Fig. 4 shows the scale index and bifurcation diagrams of Henon map and Rössler system. It is apparent from comparison of Figs. 3 and 4 that, the scale index of quantum map is higher than other two chaotic maps. So that, it can be concluded that, the generated sequence of the quantum map is more non-periodic.

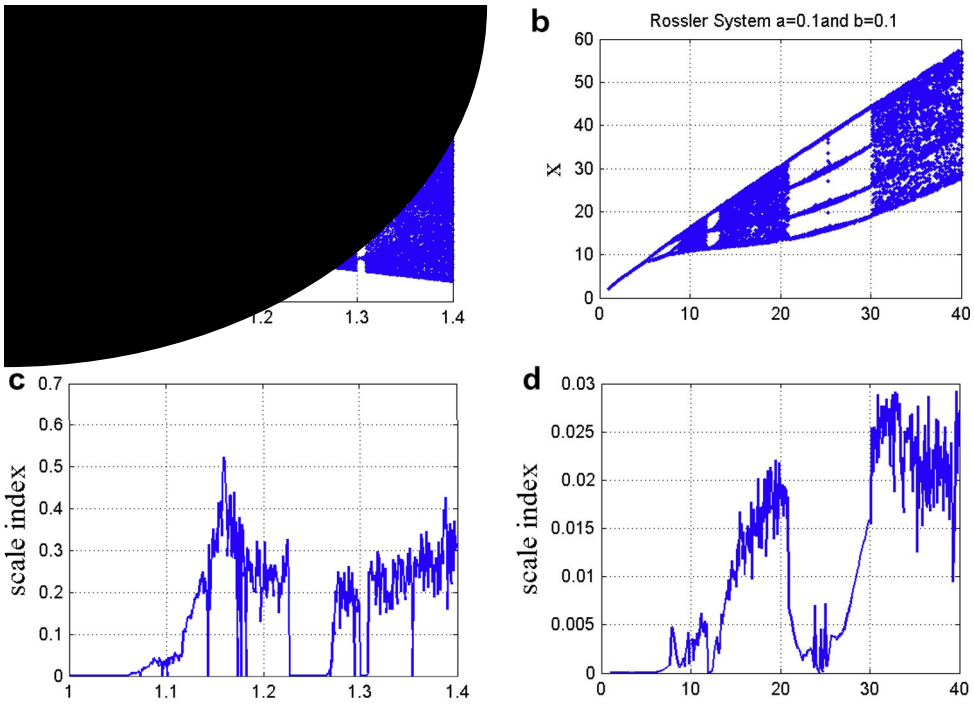

Fig. 4. The Scale index and bifurcation diagrams of the Henon map and the Rössler system.

4. Statistical complexity measure

Complexity is a measure of off-equilibrium "order". Statistical complexity measures (SCM) were proposed as quantifiers of the degree of physical structure in a signal [20,27,28]. They are null for total random processes. In this section based on the method of Ref. [29] the statistical complexity of the presented algorithm is calculated. The intensive statistical complexity measure ($C_J[P]$) can be considered as a quantity that characterizes the probability distribution P associated with the time series generated by the dynamical system [29]. It quantifies not only randomness but also the presence of correlational structures [29,28]. Eventually statistical complexity can be used to study the intricate structures hidden in the dynamics. In between these two special instances, a wide range of possible degrees of physical structure exist, degrees that should be reflected in the features of the underlying probability distribution. The measure of complexity C_J recently introduced in [29] the so-called the intensive SCM, is defined as:

$$C_J[P] = Q_J[P, P_e].H_S[P],$$

where with the probability distribution P we associate the entropic measure $H_S[P] = \frac{S[P]}{S_{max}}$, with $S_{max} = S[P_e] (0 \leqslant H_S \leqslant 1)$.

P_e is the equilibrium distribution and S is the Shannon entropy. The disequilibrium Q_J is defined in terms of the Jensen–Shannon divergence [29,21] and is given by:

$$QJ[P, P_e] = Q_0 S\left[\frac{(P + P_e)}{2}\right] - \frac{S[P]}{2} - \frac{S[P_e]}{2}.$$

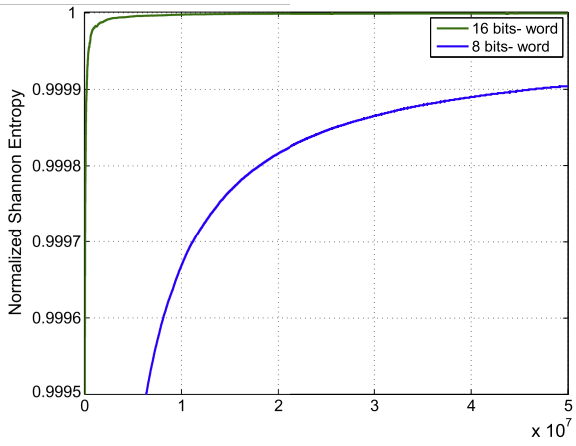

Fig. 5. Normalized Shannon entropy (H_S) for the proposed PRNG.

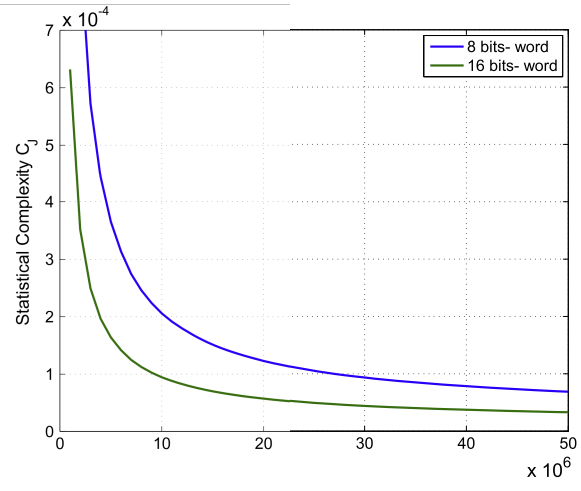

Fig. 6. Intensive statistical complexity measure (C_J) for the proposed PRNG.

With Q_0 being the normalization constant ($0 \leqslant Q_J \leqslant 1$). Thus the disequilibrium Q_J is an intensive quantity. If the PRNG is an extremely good generator we can expect that "no attractor" will be reconstructed, that is, it will be quite reasonable to obtain a homogeneity cloud of points with a tendency to "fill" the d-dimensional space [29]. Consequently, the associated permutation probability distribution will be $P \approx P_e$, so $H_S \approx 1$ and $C_J \approx 0$ and for periodic sequences will have $H_S \approx 0$ and $C_J \approx 0$ [29]. Based on the calculations mentioned above, the normalized entropy H_S and the intensive statistical complexity C_J as functions of the number of 8 bits and 16 bits-words are shown in Figs. 5 and 6.

As can be seen from the figures, when the number of words of the analyzed sequence increases, the statistical complexity and the normalized entropy tend to 0 and 1 respectively. It can be concluded that, the randomness of the proposed PRNG is successfully verified by statistical complexity and the normalized Shannon entropy.

5. Proposed algorithm

In this section, we discuss how to construct the pseudo-random number generator based on quantum chaotic map and analyze its properties. A good random number generator must have some properties such as good distribution, long period, portability and etc. In order to achieve a fast throughput and facilitate hardware realization, 32-bit precision representation with fixed point arithmetic is assumed. The steps of algorithm for generating N pseudo-random 32-bit numbers are as follows:

- Step 1: Import the keys, which are the control parameters and initial conditions.
- Step 2: Specify the required length of sequence.
- Step 3: In order to avoid transient effect the map, iterate the quantum chaotic map 1000 times, Eq. (1), using control parameters and initial conditions provided in step (1) and the first 1000 initial conditions are disposed.
- Step 4: Normalize the new value (initial condition) of the x in Eq. (1.a) using the following equation.

$$x_{n+1} = x_{n+1} \times 1000 - floor(x_{n+1} \times 1000).$$

The concept of normalization of initial condition x in Eq. (1.a) is because of nature of distribution of the initial condition values, in order to take the maximum advantage of the complexity of the map, we try to eliminate three most significant digits from the initial conditions, making it more complex and uniform. (in the above equation, the constant value 1000 is considered as the normalization factor, which can vary for more optimization.).

- Step 5: Generate and output the random number using following equation from the normalized x_{n+1}.
  Final Random Integer = floor $(x_{n+1} \times (2^{32} - 1))$.
  In this step, the goal is to convert the values of the x_{n+1} from floating values into decimal values and normalize it in the range of 32 bit integer numbers.
- Step 6: If the length of sequence is not satisfied, return to Step (3) otherwise stop.

6. Correlation analysis

In this section, in order to investigate the sensitivity of the chaotic map to very small changes in initial conditions and also the correlation between sequences produced with nearby keys [30,31], we have performed the correlation coefficients test as follows:

(1) $x = 0.62352345$; a sequence with 1,000,000 numbers is generated, then a new sequence by a very small changing of the initial condition $x' = 0.6235234500000001$ is generated.
(2) $y = 0.0152345$; a sequence with 1,000,000 numbers is generated, then a new sequence by a very small changing of the initial condition $y' = 0.015234500000001$ is generated.
(3) $z = 0.0352345$; a sequence with 1,000,000 numbers is generated, then a new sequence by a very small changing of the initial condition $z' = 0.035234500000001$ is generated.

The correlation coefficients C_{xy} for each pair of sequences are computed according to the method described in [32–34]. Let the two sequences $x = [x_1, ,x_N]$ and $y = [y_1, ,y_N]$ we have:

$$C_{xy} = \frac{\sum_{i=1}^{N}(x_i - \bar{x}).(y_i - \bar{y})}{\left[\sum_{i=1}^{N}(x_i - \bar{x})^2\right]^{1/2}\left[\sum_{i=1}^{N}(y_i - \bar{y})^2\right]^{1/2}},$$

where $\bar{x} = \sum_{i=1}^{N}\frac{x_i}{N}$ and $\bar{y} = \sum_{i=1}^{N}\frac{y_i}{N}$ are the mean values of x and y respectively. A strong correlation occurs between two sequences for $C_{xy} \simeq \pm 1$ and no correlation corresponds to $C_{xy} = 0$. The corresponding data are listed in Table 1. Based on the analysis of the data in Table 1, we can draw the conclusion: there is no correlation between the generated sequences and also the chaotic map is very sensitive to very small changes in all initial conditions.

Table 1
Correlation coefficients of three pairs of pseudo random sequences.

$x = 0.62352345$	$x' = 0.6235234500000001$	C_{xy}=−0.0028
$y = 0.0152345$	$y' = 0.015234500000001$	C_{xy}=0.0009
$z = 0.0352345$	$z' = 0.035234500000001$	C_{xy}=−0.0017

7. Tests for randomness

In this section, several tests carried out to examine the randomness of the presented algorithm, these tests are DIEHARD [18], NIST [5], ENT [19]. Moreover, the most stringent testing by TestU01 [17] is applied to verify the statistical properties of the proposed system. TestU01 has flexible parameters, and hence is suitable to implement the adaptive version of statistical tests, unlike DIEHARD and NIST where the sample size is fixed. As to tests by TestU01, there are three different crush type batteries; the smallest, smallcrush, which evaluates about 2^{29} of random numbers, 4 GiByte of data; an intermediate one, simply called crush, evaluates around 2^{35} random numbers, 256 GiByte of data; and the most stringent of the crushes, bigcrush, where the quantity of random numbers needed for evaluation lies around 2^{38} numbers corresponding to 1.5 TiByte of data. For each test, a *P*-value is calculated. If *P*-value is within the range, $[10^{-4}, 1 - 10^{-4}]$, the associated test is a success. Any *P*-values lying outside this range is considered as failure. According to Tables 2–6 which present NIST, DIEHARD, ENT and TestU01 tests results respectively, the introduced PRNG passes all the tests. These experimental results lead us to conclude that quantum based pseudo-random number generator is a very good and reliable PRNG and can be used in simulation as well as security fields [2,37–39].

7.1. Key space analysis

Random number generators are used in generating cryptographic keys. In the proposed scheme, the following parameters can be used as encoding keys. The keys are chosen as follows: $x = 0.62352345$, $y = 0.0152345$, $z = 0.0352345$, $r = 3.99$ and β=10. The parameters are chosen from the chaotic region of the control parameters and initial conditions. In any case, the designer of any chaotic cryptosystem should conduct a study of chaotic regions of the parameter space from which valid keys, i.e., parameter values leading to chaotic behavior, can be chosen [37]. Only keys chosen from the black region of bifurcation diagram are suitable enough [40]. Bifurcation diagram is used to describe any sudden changes in the dynamics of the system. The bifurcation diagram of the mentioned chaotic map is given in Fig. 1. As it can be seen in Fig. 1, within the black regions, there are not any periodic windows, so the entire black region is suitable for robust keys. The figure is obtained by plotting, for several β (dissipation parameter) values, the N successive iterations of the map (here $N = 80{,}000$). As shown in Fig. 1, for small value of dissipation parameter (β) the system is stable. The period doubling transition to chaos is occurred by

Table 2
Results of the SP800–22 tests suite for the 32-bit proposed PRBG.

Test name	*P*-value	Result
Frequency	0.602458	Success
Block-frequency	0.253551	Success
Cumulative sums		
Forward	0.213309	Success
Reverse	0.275709	Success
Runs	0.407091	Success
Long runs	0.60239	Success
Rank	0.876060	Success
FFT	0.671779	Success
Non-periodic templates	0.275709	Success
Overlapping templates	0.602458	Success
universal	0.602458	Success
ApEn	0.350485	Success
Serial		
P-value 1	0.378138	Success
P-value 2	0.568055	Success
Linear complexity	0.110952	Success
Approximate entropy ($m = 10$)	0.350485	Success
random-excursions		
X = −4	0.574903	random-excursions
X = −3	0.153763	random-excursions
X = −2	0.383827	random-excursions
X = −1	0.191687	random-excursions
X = 1	0.739918	random-excursions
X = 2	0.494392	random-excursions
X = 3	0.040108	random-excursions
X = 4	0.023545	random-excursions

Table 3
Results of the SP800-22 tests suite for the 32-bit proposed PRNG.

Random excursions variant (state X)		
X = −9	0.040108	Success
X = −8	0.350485	Success
X = −7	0.319084	Success
X = −6	0.455937	Success
X = −5	0.955835	Success
X = −4	0.262249	Success
X = −3	0.213309	Success
X = −2	0.122325	Success
X = −1	0.058984	Success
X = 1	0.096578	Success
X = 2	0.171867	Success
X = 3	0.534146	Success
X = 4	0.494392	Success
X = 5	0.467553	Success
X = 6	0.911413	Success
X = 7	0.883171	Success
X = 8	0.455937	Success
X = 9	0.191832	Success

Table 4
DIEHARD tests suite for the 32-bit proposed PRNG.

Test name	Average value	Result
Birthday spacing	0.846501	Success
Overlapping permutation	0.144367	Success
Binary rank 31×31	0.928232	Success
Binary rank 32×32	0.830982	Success
Binary rank 6×8	0.128879	Success
Bitstream	0.59207	Success
OPSO	0.7334	Success
OQSO	0.40855	Success
DNA	0.5137	Success
Count the ones 01	0.169695	Success
Count the ones 02	0.186592	Success
Parking Lot	0.441949	Success
Minimum distance	0.526178	Success
3DS spheres	0.465835	Success
Squeeze	0.537076	Success
Overlapping sum	0.890220	Success
Runs	0.773666	Success
Craps	0.27023	Success

Table 5
Max grade of ENT test suite.

Test name	Average value	Result
Entropy	7.999995	Success
Arithmetic mean	127.7714	Success
Monte Carlo	3.140621114	Success
Chi-square	255.19	Success
Serial correlation coefficient	0.000108	Success

Table 6
TestU01 test suite for the 32-bit proposed PRNG.

Battery	Parameters	Number of statistics	Result
SmallCrush	Standard	15	Pass
Crush	Standard	144	Pass
BigCrush	Standard	160	Pass

increasing β in the dynamical system. From the cryptographic point of view, the size of the key space should not be smaller than 2^{128} to provide a high level of security [41]. In this specific algorithm, there are several control parameters and control parameters which the sensitivity of the algorithm to them is tested using the statistical test suits and also the bifurcation diagram. The size of the key space therefore would be all the conditions that these keys can be applied: $x\in[0,1]$, $y\in[0,0.1]$, $z\in[0,0.2]$, $\beta\in[6,\infty)$ and $r\in[0,4]$. This algorithm is applied in C++ with 10^{-16} precision, so that roughly the range of each of the parameters can be calculated as below:

$$(10^{16})\times(10^{15})\times(10^{15})\times(4\times10^{16})\times(4\times10^{16})=1.6\times10^{79}\simeq2^{262}.$$

As we want to make sure that implementation will not cause any precision related problem related, so that we assume that the precision is 10^{-14} which is a very safe testimony and the key space in this case would be 2^{236}. It seems that, the size of key space compared with is large enough to resist all kinds of brute-force attacks.

7.2. Guess-and-determine and distinguishing attacks

"Guess and Determine" and "Distinguishing" against stream ciphers are presented in Refs. [42,43]. The presented algorithm is a cryptographic pseudo-random number generator rather than a stream cipher (although it can be applied to design a stream cipher). The algorithm is based on the chaotic map; unlike the linear feedback shifts which are based on linear systems, so that it is not really applicable to apply the guess and determine and distinguishing attacks on it. The chaotic map is fully coupled and it is not possible to analysis each section of the map separately and guess the parameter. Moreover, as it can be seen from the results of the correlation coefficient analysis (see Table 1), the map is extremely sensitive to its initial conditions. Furthermore, the proposed PRNG passed all tests in TestU01 including linear complexity tests that all linear feedback shift-register (LFSR) and generalized feedback shift-register (GFSR)-based random number generators fail [17].

7.3. Differential attack

In order to study the security of the presented algorithm against different types of attacks, randomness of the numbers generated is examined using several batteries such as Diehard, NIST and TestU01. Moreover to ensure the resistance of the algorithm against the differential attack, the "Sum of Absolute Difference" (SAD) analysis is carried out in this paper.

In the differential attack, which mostly is applicable on block ciphers, effect of very small changes on the plaintext and their corresponding cipher text is analyzed. But as in PRNGs there is no input plaintext, thus we have applied the same analysis on the initial seeds which are at the same time keys for the random number generator.

In this section, we have generated three large sequence (S_1, S_2 and S_3) of random numbers (with 50,000,000 members) and applying a very small change (10^{-15}) in the initial conditions, (x, y and z) the resulting sequences (S'_1, S'_2 and S'_3) are also generated and used for the Sum of Absolute Difference analysis. In this test, the sum of absolute difference between each pair of sequences (S_1 and S'_1, S_2 and S'_2 and finally S_3 and S'_3) is calculated using following formula: [38,39].

$$d=\sum_{i=1}^{N}|S(e_i)-S(e'_i)|.$$

The results of mean of d are listed in Table 7. According to the results of the sum of absolute difference, the ideal value of mean sum of absolute difference of two uniform random sequences is $\frac{2}{3}$ of their mean value which in this case is $\frac{2}{3}$ of $\frac{1}{2}$ [44]. The test result demonstrates sensitivity of the presented algorithm to the keys hence it can be concluded that the presented algorithm is highly resistive against differential attack.

7.4. Analysis of speed

We have analyzed the speed of the proposed algorithm on an Intel Core i5-2467 M CPU@1.60GI-Iz 1.60 GHz with 4 GB Running on Microsoft Window 8 Professional, using Microsoft Visual C++ Ultimate compiler. The mean speed is 873.05 *Mbits/S*. Also according to the algorithm and the chaotic map, number of multiplication used in this algorithm per each byte is 5 and the number of cycles needed per each random number generated is about 180 [48], which indicates that, the proposed algorithm is fairly fast. Also the IEEE 754–2008 standard for the floating numbers has been used for all the variables in this algorithm, which makes the system standard and the same on all the machines applying this universal standard.

Table 7
Mean values of the absolute difference (d).

$x = 0.62352345$	$x' = 0.623523450000001$	Mean of d = 0.3353801
$y = 0.0152345$	$y' = 0.015234500000001$	Mean of d = 0.3357431
$z = 0.0352345$	$z' = 0.035234500000001$	Mean of d = 0.3300391

8. Conclusion

The goal of this work is to show that the quantum chaotic map can be used as a random number generator and more generally, as a source of entropy [45,46]. Actually, we can never have perfect generators in the real world.

This work is the first attempt, to our knowledge, of exploring the quantum chaotic map as a random number generator. Though chaotic orbits of discrete-time maps are non-periodic in nature, because of finite precision of digital computers the orbits actually turn out to be periodic. So that, the average period of an orbit of a three-dimensional map can be expected to be longer than that of a one dimensional map. For this purpose, a wavelet based analysis called the Scale index is carried out [26]. The obtained result shows that, the chaotic orbit of the quantum map is more non-periodic. In this paper, we have proposed a new pseudo-random number generator, which exploits the interesting properties of three-dimensional quantum logistic map such as statistical complexity [47]. To evaluate the randomness and uniformity, three different statistical tests including NIST, DIEHARD and ENT test suites are employed. Moreover, the most stringent testing by TestU01 is applied to verify the statistical properties of the proposed system. The statistical test results show that, the proposed PRNG passes all the standard statistical tests in NIST, DIEHARD, ENT and TestU01 test suites. These experimental results lead us to conclude that our generator is a very good and reliable PRNG. It offers a sufficient level of security for a whole range of applications in computer science. It seems that, this PRNG can also be used for secure QKD. Also the application of this random number generator in Quantum Monte Carlo and cryptography is immediate.

References

[1] Katz J, Lindell Y. Introduction to modern cryptography. Boca Raton: Chapman and Hall/CRC Press; 2007.
[2] Bennett C, Brassard G. Quantum cryptography: public-key distribution and coin tossing. In: Proceedings of IEEE international conference on computers, systems and signal processing, Bangalore, India: December; 1984. p. 175–9.
[3] Jeffrey E. Advanced quantum communication systems, Ph.D. Thesis; 2007.
[4] Rogers DJ. Broadband Quantum Cryptography. Morgan, Claypool Publishers; 2010.
[5] National Institute of Standards and Technology, Computer code available at <http://csrc.nist.gov/rng/SP800-22b.pdf>.
[6] Matsumoto M, Nishimura T. ACM Trans Model Comput Simul 1998;8(1):3.
[7] Blum L, Blum M, Shub M. SIAM J Comput 1986;15(2):364383.
[8] Chirikov BV, Vivaldi F. Physica D 1999;129:223.
[9] Chirikov BV. Pseudochaos in statistical physics. In: Infeld E, Zelazny R, Galkowski A, editors. Proceedings of the international conference on nonlinear dynamics, chaotic and complex systems. Cambridge University Press; 1997. p. 149.
[10] Chirikov BV. Open Syst Inf Dyn 1997:4241.
[11] Berry MV, Balazs NL, Tabor M, Voros A. Ann Phys 1979;122:26.
[12] Graffi S, Degli Esposti M, editors. The mathematical aspects of quantum maps. Lecture notes in physics, vol. 618. Springer; 2003.
[13] Haake F. Quantum signatures of chaos. Springer; 2000.
[14] Goggin ME, Sundaram B, Milonni PW. Phys Rev A 1990;41:5705.
[15] Graham R. Phys Rev Lett 1989;62:1806.
[16] Graham R. Europhys Lett 1987;3(3):259.
[17] L'Ecuyer P, Simard R. ACM Trans Math Software 2007;33(4) [Article 22].
[18] Marsaglia G. Computer code DIEHARD; 1997, available at, <http://stat.fsu.edu/pub/diehard/>.
[19] Walker J. ENT. A pseudo random number sequence test program; 1998, available at, <http://www.fourmilab.ch/random/>.
[20] López-Ruiz R, Mancini HL, Calbet X. Phys Lett A 1995;209:321.
[21] Lamberti PW, Martin MT, Plastino A, Rosso OA. Phys A 2004;334:119.
[22] Rosso OA, Larrondo HA, Martin MT, Plastino A, Fuentes MA. Phys Rev Lett 2007;99:154102.
[23] Eckmann J, Oliffson Kamphorst S, Ruelle D. Europhys Lett 1987;4:973.
[24] Marwan N, Romano MC, Thiel M, Kurths J. J Phys Rep 2007;438:237.
[25] De Micco L, Larrondo HA, Plastino A, Rosso OA. Phil Trans R Soc A 2009;367:3281.
[26] Beñıtez R, Bolós VJ, Ramìrez ME. Comput Math Appl 2010;60:634.
[27] Shiner JS, Davison M, Landsberg PT. Phys Rev E 1999;59:1459.
[28] Martin MT, Plastino A, Rosso OA. Phys Lett A 2003;311:126.
[29] Larrondo HA, González CM, Martín MT, Plastino A, Rosso OA. Phys A 2005;356:133.
[30] Lee P-H, Chen Y, Pei S-C, Chen Y-Y. Comput Phys Commun 2004;160:187.
[31] Patidar V, Pareek NK, Purohit G, Sud KK. Commun Nonlinear Sci Numer Simul 2010;15:2755.
[32] Pareek NK, Patidar V, Sud KK. Digital Signal Proc 2013;23:894.
[33] Pareek NK, Patidar V, Sud KK. Int J Net Sec 2010;10(1):32.
[34] Patidar V, Sud KK. Electron J Theor Phys 2009;6(20):327.
[35] Mallat SA. Wavelet tour of signal processing. Academic Press; 1999.
[36] Chandre C, Wiggins S, Uzer T. Physica D 2003;181:171.
[37] Akhshani A, Mahmodi H, Akhavan A. A novel block cipher based on hierarchy of One-dimensional composition chaotic maps. IEEE international conference on image processing; 2006. p. 1993–6.
[38] Akhavan A, Samsudin A, Akhshani A. Chaos Solitons Fract 2009;42:1046.
[39] Akhshani A, Behnia S, Akhavan A, Jafarizadeh MA, Abu Hassan H, Hassan Z. Chaos Solitons Fract 2009;42:2405.
[40] Alvarez G, Li S. Int J Bifurcation Chaos 2006;16(8):2129.
[41] Ecrypt II yearly report on algorithms and keysizes; 2010. <http://www.ecrypt.eu.org/documents/D.SPA.13.pdf>.
[42] Ahmadi H, Eghlidos T. Inf Secur IET 2009;3(2):66.
[43] Coppersmith D, Halevi S, Jutla C. In: 22nd Annual international cryptology conference, Santa Barbara, California. Advances in Cryptology – CRYPTO 2002; 2442. p. 515–32.
[44] Johnsonbaugh R. Discrete mathematics. 5th ed. Prentice Hall; 2000.
[45] Li H, Zhang J. Commun Nonlinear Sci Numer Simul 2009;14:4304.
[46] Nagaraj N, Vaidya PG. Chaos 2009;19:033102.
[47] Akhshani A, Akhavan A, Lim SC, Hassan Z. Commun Nonlinear Sci Numer Simulat 2012;17:4653.
[48] Muller J-M, Brisebarre N, de Dinechin F, Jeannerod C-P, Lefèvre V, Melquiond G, Revol N, et al. Handbook of floating-point arithmetic. Boston: Birkhauser; 2009.