Engineering Science & Technology
https://ojs.wiserpub.com/index.php/EST/

UNIVERSAL WISER PUBLISHER

Research Article

PiRATe: A Comprehensive Set of Statistical Tests for Evaluation of Quantum Random Number Generators

Soheil Hajibaba, AmirHosein Dadahkhani, Hamid Asgari, Majid Khodabandeh, Seyed Ahmad Madani*

Quantum Communication Group, Iranian Center for Quantum Technologies, Tehran, Iran
Email: seyed.ahmad.madani@gmail.com

Received: 26 December 2023; **Revised:** 9 April 2024; **Accepted:** 6 May 2024

Abstract: As true random numbers are essential for many applications, high-quality Random Number Generators (RNGs) are in high demand. Several different statistical test suites have been developed to evaluate the quality of RNGs. However, some programming skills are required to work with these statistical tests as they are not available as self-contained software and often lack a Graphical User Interface (GUI). In this work, we have developed PiRATe (Pi Randomness Assessment Test), an easy-to-use wrapper software for the assessment of RNGs using a comprehensive set of statistical tests, including DIEHARDER, NIST SP800-22, NIST SP800-90B, Entropy-Nonlinearity-Test (ENT), Borel Normality, and four Chaitin-Schwartz-Solovay-Strassen tests. We then used PiRATe to evaluate the quality of five different Quantum Random Number Generators (QRNGs), both commercial and experimental.

***Keywords*:** random numbers, statistical tests, quantum random number generator

1. Introduction

High-quality random numbers are essential in many important and practical areas including telecommunications [1]-[3], cryptography [4]-[5], simulations [6]-[7], games [8], and quantum technologies such as Quantum Key Distribution [9]-[11], Quantum Imaging [12]-[13], and Quantum Radar [14]-[15]. There are two approaches to the generation of random numbers: a software approach known as a Pseudo-Random Number Generator (PRNG), which uses mathematical algorithms to generate random numbers from an initial seed [16]-[17], and a hardware approach known as a True Random Number Generator (TRNG), which extracts random numbers from a physical process [18]-[19]. PRNGs may produce high-quality random sequences but by definition, they are (Turing) computable and hence predictable. The physical process behind TRNGs could be classical or quantum in nature [20]. While classical TRNGs employ complexity behind causality, a quantum TRNG known as a Quantum Random Number Generator (QRNG) employs the probabilistic nature of quantum mechanics to produce true random sequences that are unpredictable and incomputable. Random numbers from an ideal QRNG will be unbiased, however, it is hard to eliminate classical noise in practice [20]. The quality of the output from a Random Number Generator (RNG) (regardless of its type), can be assessed by a set of statistical tests [21]-[22]. Common test suites such as Dieharder [23] and NIST SP800-22 [24] compare the statistical distributions of predefined patterns in the given string to the distribution expected for an

DOI: https://doi.org/10.37256/est.5220244165

ideal random number generator. Other tests evaluate the performance of the output with respect to certain tasks and then compare this performance to that of other random number generators [25]. These tests are necessary to verify algorithmic randomness and thus incomputability of the random numbers [21]-[22].

Considering researches conducted at the Iranian Center for Quantum Technologies (ICQTs) [26]-[28] and the need for high-quality RNGs, we tried to gather a credible set of statistical tests (as a package) and apply them to different instruments and random number generation systems [29]. In this paper, we developed a wrapper software with Graphical User Interface (GUI) consisting of a comprehensive set of statistical tests including Dieharder [23], NIST SP800-22 [24], NIST SP800-90B [30], ENT [31], Borel Normality [25], [32], and Chaitin-Schwartz-Solovay-Strassen (CSSS) [21]-[22] tests. We then prepared random data from 5 different QRNGs, performed this set of statistical tests on them, and compared them.

This paper is organized as follows. In Section 2, we give a brief overview of different statistical tests employed, a description of the software structure and a description of different QRNGs used in this work. Then we report the evaluation results in Section 3. We give an outlook on the impacts and future applications of this software in Section 4. Finally, a conclusion is given in Section 5.

2. Materials and methods

2.1 *Statistical tests*

There are several statistical tests to evaluate the quality of a string of random numbers. Considering their applications and popularity, we decided to choose six of the most important packages [20]-[21]. Now we will briefly describe these six different packages of tests we have employed in this software.

2.1.1 *Dieharder*

Dieharder [23] is the most famous test suite for testing RNGs. It is based on an older Diehard [33] battery of tests and consists of 31 statistical tests. Each test in this suite evaluates a p-value that should be larger than the significance level. The significance level in the tests is $\alpha = 0.01$. The test is considered successful if all the p-values satisfy $0.01 \leq p - value \leq 0.99$.

2.1.2 *NIST SP800-22*

NIST SP800-22 [24] is the second famous statistical test suite which consists of 15 statistical tests. Similar to Dieharder, each test in this suite evaluates a p-value that should be larger than the significance level. The significance level in the tests is $\alpha = 0.01$. The test is considered successful if all the p-values satisfy $0.01 \leq p - value \leq 0.99$.

2.1.3 *NIST SP800-90B*

NIST SP800-90B [30] is an entropy assessment package that provides a standardized means of estimating the quality of a source of entropy. This test assures that a random sequence is independent and identically distributed (IID) and also estimates the min-entropy for the provided data.

2.1.4 *ENT*

ENT [31] is a series of basic statistical tests that evaluate the random sequence in some elementary features such as entropy [34], Chi-square Test [35], arithmetic mean, Monte Carlo value for Pi, and serial correlation coefficient [35]. The entropy is the information density of the random data, expressed as a number of bits per bit. In other words, entropy measures how many bits are required to construct a bit of data. Non-random bits have lower entropy as they can be described using fewer bits. The Chi-square test is a common test for the evaluation of data randomness and is extremely sensitive to errors in RNGs. The Chi-square distribution is calculated for the stream of bytes in the file and expressed as an absolute number and a percentage which indicates how frequently a truly random sequence would exceed the value calculated. The arithmetic mean is the sum of all 0 s and 1 s divided by the total length. This should be about 0.5 for

nearly random data. The deviation of the mean from this value indicates that the data is biased. This test uses successive sequences of bits as X and Y coordinates within a square and uses the Monte Carlo algorithm to calculate the value of Pi. The Monte Carlo value for Pi is a measure of the quality of a random number generator. The serial correlation coefficient measures the extent to which each byte in the string depends upon the previous byte. For random sequences, this value is close to zero.

2.1.5 *Borel normality*

A finite string is subjected to the Borel normality test [25], [32], which determines if all substrings of length appear with the expected probability of $2 - m$. A string x of length $|x|$ is said to be Borel normal if all integers m with $1 \leq m \leq \log_2 \log_2 |x|$ satisfy the following requirements:

$$\left(\max_{1 \leq j \leq 2^m} \left| \frac{N_j^m(x)}{|x|/m} - \frac{1}{2^m} \right| \right) \sqrt{\frac{|x|}{\log_2 |x|}} \leq 1 \tag{1}$$

Where $N_j^m(x)$ is the number of occurrences of the j th string selected from all binary strings with length m. The left side of the above equation is called the Metric Value. Although Borel normality is a prerequisite for algorithmic randomness and incomputability, it is insufficient [21], [22], [36]. In the outputs from QRNGs, Borel normality violations have been observed [21], [25], [37]. It has been suggested that these violations may have been caused by bias in the random strings [21], [37].

2.1.6 *CSSS*

Random sequences are used as witnesses in the CSSS tests to determine whether an integer n is composite or prime. These tests are based on the Solovay-Strassen primality test [38] and the Chaitin-Schwartz theorem [39]. For an integer n, the Solovay-Strassen test establishes a predicate $W(n, a)$ defined by

$$\left(\frac{a}{n}\right) \neq a^{(n-1)/2} \pmod{n} \tag{2}$$

Where a is a natural number between 1 and $(n - 1)$, and $\left(\frac{a}{n}\right)$ is the Jacobi symbol. $W(n, a)$ is called the Solovay-Strassen predicate. "If n is prime, then $W(n, a)$ is false for all $a \in [1, n - 1]$. Thus, if $W(n, a)$ is true for some $a \in [1, n - 1]$, then n is composite, and we call a a witness to the compositeness of n" [22].

The probabilistic outcome of the Solovay-Strassen test can be transformed into a rigorous proof of primality using the Chaitin-Schwartz theorem if potential witnesses are drawn from random strings. "Let s be a string of length m in 2-bit binary representation, and let n be an integer. Rewrite s into base $(n - 1)$, with digits $s = d_k\, d_{k-1} \ldots d_0$ over the alphabet $\{0, 1, \ldots, n - 1\}$, where k is given by the smallest integer that satisfies the inequality" [22].

$$k > \frac{\log(2^m - 1)}{\log(n - 1)} - 1 \tag{3}$$

Now the compound predicate is defined as

$$Z(n, s) = \neg W(n, 1 + d_0) \wedge \neg W(n, 1 + d_1) \wedge \ldots \wedge \neg W(n, 1 + d_{k-1}) \tag{4}$$

Where $W(n, 1 + d_i)$ is the Solovay-Strassen predicate.

"The Chaitin-Schwartz theorem states that for all sufficiently large c, if s is a random string containing $l(l + 2c)$ bits and n is an integer whose binary representation is l bits long, then $Z(n, s)$ is true if and only if n is prime." [22].

We can therefore conclude that random strings generally outperform nonrandom strings in testing primality.

Consequently, the CSSS tests can identify algorithmic randomness and incomputability. The four CSSS tests use random strings from the RNG source as large sets of potential witnesses for the primality test of sets of composite numbers. We can then compare the performance of strings from different QRNGs by measuring how well these strings perform in this primality-testing task. We used the CSSS tests as proposed and applied by Ref. [21] and code implemented by Ref. [22]. As in Ref. [21] and Ref. [22], we choose Carmichael numbers as test numbers, using the set of all Carmichael numbers up to 10^{21} (calculated in Ref. [40]). We look for statistically significant differences in the results from four CSSS tests by comparing the distributions of test results using the statistical analysis described in Ref. [22] and Ref. [21]. The statistical analysis includes the Kolmogorov-Smirnov test for two samples [41] and Welch's t-test [42], both with a significance level of 0.005.

2.2 *Software description*

In this section, we describe the PiRATe software architecture and functionalities. PiRATe is written in C++ and its source is available on GitHub [43]. The source has been compiled and tested successfully in Ubuntu 22.04.

2.2.1 *Software architecture*

PiRATe is a wrapper software which means it consists of several independent programs. Each program is associated with a certain set of statistical tests and is originally written by other researchers. There are two ways for PiRATe to access these programs:

1. Internal access
2. External access

While some of the programs are internally attached to PiRATe other programs have their executable files located in the **Tests**/directory. Thus every time a user runs a test in PiRATe, it automatically executes the corresponding test.

2.2.2 *Software functionalities*

PiRATe gets a random data file (in binary format) and performs a randomness assessment on it using several statistical tests. As shown in Figure 1, PiRATe has a simple graphical user interface (GUI) where the user can simply choose the random data file and select the intended test. After clicking the "Start Test" button, the output result from the selected test is automatically printed on the screen and can be saved later into a text file.

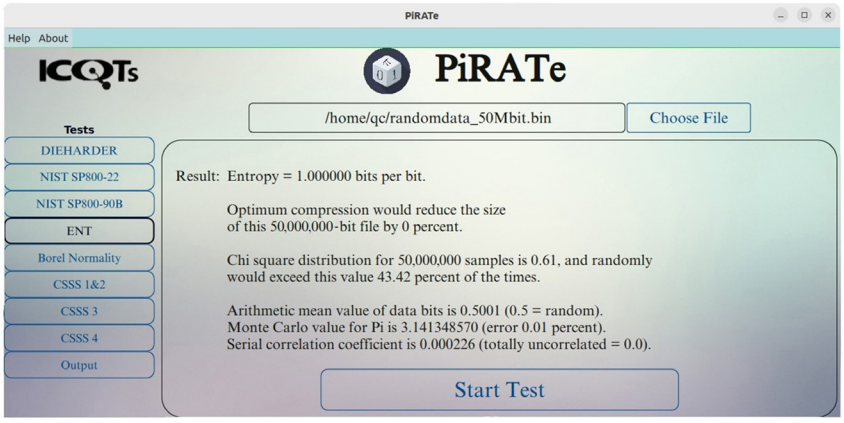

Figure 1. A view from PiRATe

2.3 *Quantum random number generators*

We prepared several random sequences with lengths of 50 Mbits and 1 Gbits from five QRNGs with different entropy sources based on: radioisotope decay [44]-[45], laser phase fluctuation [46]-[47], intensity in Charge-Coupled Device (CCD) [48], polarization entanglement source, and quantum fluctuations of the vacuum [49]-[50] (Table 1).

Table 1. List of QRNGs evaluated in this work

Short name	Full name	Manufacturer	Mechanism
EYL	EYL's quantum entropy chip	Everywhere in your life [46]	Radioisotope decay
F.H.E.S	Farhikhtegan-e Hami-e Elm-o-Sanaat QRNG	Farhikhtegan-e Hami-e Elm-o-Sanaat [48]	Laser phase fluctuation
SUT	Sharif University of Technology QRNG	Sharif University of Technology [49]	Intensity in CCD
FreeSpace	Free-space QKD key	Iranian center for quantum technologies	Polarization entanglement source
ANU	Australian National University QRNG	Australian National University [51]	Quantum fluctuations of the vacuum

We also prepared random sequences with a length of 2 Gbits from EYL [45], SUT [48], and ANU [52] for CSSS tests.

2.3.1 *EYL*

EYL is a commercial QRNG manufactured by EYL Co in South Korea [45]. It consists of an Entropy chip that extracts random data from radioisotope decay. The raw data is then used to seed a PRNG implemented in an FPGA. According to the manufacturer, EYL has passed NIST SP800-22, NIST SP800-90B, AIS31, and Diehard tests.

2.3.2 *F.H.E.S*

F.H.E.S is a commercial QRNG manufactured by F.H.E.S Co in Iran [47]. It extracts random numbers from Laser Phase fluctuations. It consists of a Distributed Feedback (DFB) laser as the light source, modulated using electrical pulses. The output light is sent to an unbalanced Mach-Zehnder interferometer and then enters a photodiode. The output signal is converted to digital using an Analog-to-Digital Converter (ADC) to obtain raw data. Then raw data is post-processed to obtain final random sequences.

2.3.3 *SUT*

SUT is an experimental QRNG located at the Sharif University of Technology (Information, Network, and Learning (INL) Lab) which contains a light-emitting diode (LED) and an image sensor. Due to quantum noise, the LED emits a random number of photons. The photons are captured and counted by the pixels of the image sensor (CCD), thus creating a series of raw random numbers. Finally, these raw random numbers are post-processed to achieve final random data [48].

2.3.4 *FreeSpace*

FreeSpace is the generated key from the first free-space entanglement-based Quantum key distribution (QKD) experiment (similar to Ref. [52]) in Iran. This experiment (Figure 2) was performed by ICQTs between Milad Tower and Azadi Tower in Tehran at a distance of about 6 km. The implemented protocol was BBM92 and the entangled photon source was a Sagnac-interferometer with PPKTP crystal. The entangled photon source produces pairs of polarization-entangled photons using the Spontaneous parametric down-conversion (SPDC) mechanism which is a quantum process and inherently random. Thus, we assumed that the raw generated keys from this experiment can be considered as random strings and be used for tests.

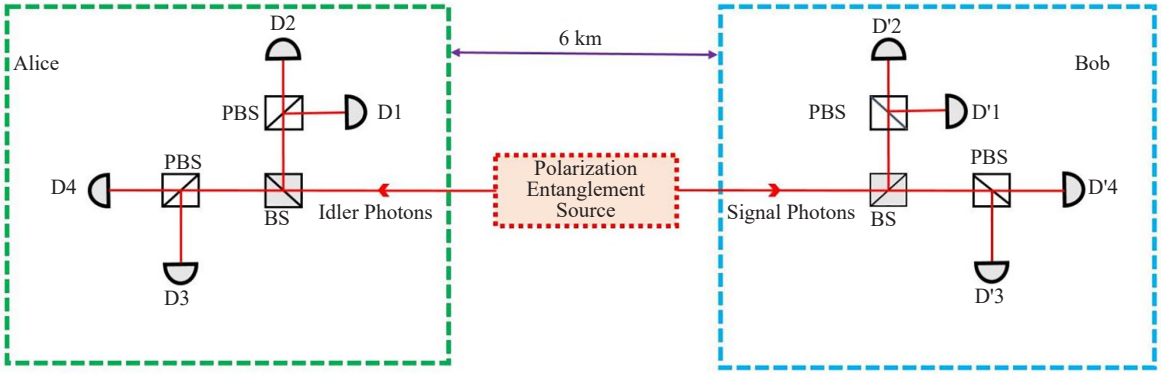

Figure 2. Schematics of free-space QKD experiment

2.3.5 *ANU*

The last QRNG is an experimental setup located at the Australian National University that generates random strings based on measurements of quantum fluctuations of the vacuum [49], [51]. The output from this QRNG is publicly available and its statistical randomness is confirmed in previous studies [22]. We used the outputs from this QRNG as a reference to compare our results.

3. Results

In this section, we present and analyze the statistical test results of the QRNGs obtained by PiRATe software.

3.1 *Dieharder*

Dieharder test results are presented in Table 2. As one can see, except for FreeSpace, all other QRNGs passed a good proportion of tests.

Table 2. Dieharder test results

QRNG	Number of passed tests
EYL	30/31
F.H.E.S	28/31
SUT	31/31
FreeSpace	4/31
ANU	28/31

3.2 *NIST SP800-22*

NIST SP800-22 test results for sequences of 50 Mbits of random data are listed in Table 3.

As one can see, EYL, F.H.E.S, SUT, and ANU passed the NIST SP800-22 tests. However, FreeSpace showed poor performance.

Table 3. NIST SP800-22 test results

NIST test item	EYL P-value	F.H.E.S P-value	SUT P-value	FreeSpace P-value	ANU P-value
Frequency	0.275709	0.851383	0.514124	0	0.834308
Block frequency	0.798139	0.779188	0.162606	0.000818	0.275709
Cumulative sums	0.595549	0.946308	0.554420	0	0.262249
Runs	0.574903	0.834308	0.262249	0	0.883171
Longest run	0.401199	0.816537	0.030806	0	0.275709
Rank	0.026948	0.983453	0.883171	0.171867	0.759756
FFT	0.000883	0.304126	0.115387	0.289667	0.534146
Non overlapping template	0.474986	0.12962	0.699313	0.058984	0.946308
Overlapping template	0.304126	0.383827	0.401199	0	0.383827
Universal	0.155499	0	0	0	0
Approximate entropy	0.366918	0.574903	0.236810	0	0.935716
Random excursions	0.939876	0.275709	0.275709	-	0.213309
Random excursions variant	0.874924	0.437274	0.048716	-	0.534146
Serial	0.759756	0.798139	0.171867	0.005358	0.249284
Linear complexity	0.739918	0.911413	0.574903	0.55442	0.759756

3.3 *NIST SP800-90B*

Results from NIST SP800-90B are presented in Table 4. As one can see only FreeSpace failed this test.

Table 4. NIST SP800-90B test results

QRNG	NIST SP800-90B test
EYL	Passed
F.H.E.S	Passed
SUT	Passed
FreeSpace	Passed
ANU	Passed

3.4 *ENT*

The testing results with ENT and ideal values of each parameter are presented in Table 5.

FreeSpace shows an Entropy below 1 bit per bit, which means that it can be compressed. Its Chi-square distribution is also out of the ideal range. The arithmetic mean value is a measure of bias in zeros and ones. It can be seen that FreeSpace is biased. The Monte Carlo value for Pi can be considered a quality factor. We can see that ANU is performing very well. The serial correlation coefficient is another important factor to ensure the lack of correlation, thus

the randomness of the data. EYL and ANU have the lowest serial correlation coefficient. As one can see, EYL, SUT, and ANU results are very close to ideal values.

Table 5. ENT test results

ENT test item	EYL	F.H.E.S	SUT	FreeSpace	ANU	Ideal value
Entropy (bits per bit)	1.000000	1.000000	1.000000	0.999996	1.000000	1.000000
F.H.E.S	42.21%	96.37%	8.75%	1.50%	23.84%	10%-90%
SUT	0.4999	0.5000	0.4998	0.4988	0.5001	0.5000
FreeSpace	3.142257786	3.137246620	3.141624227	3.145970336	3.141585213	3.1415926536
ANU	0.000051	0.000317	0.000234	0.004146	0.000049	0.000000

3.5 *Borel normality*

The metric value from the Borel Normality test and results are presented in Table 6. For passing this test, the metric should be equal to or less than 1, which is not the case for FreeSpace.

Table 6. Borel normality test results

QRNG	Metric value	Status
EYL	0.196995	Passed
F.H.E.S	0.319421	Passed
SUT	0.270778	Passed
FreeSpace	7.76125	Failed
ANU	0.204028	Passed

3.6 *CSSS*

As FreeSpace random numbers did not pass the previous tests, we only performed the CSSS tests on EYL, SUT, and ANU. The objective of these tests is to detect any statistically significant difference between different RNGs.

The first CSSS test results for original (Figure 3a) and complemented bits (Figure 3b) show the minimum number of witnesses required to confirm the compositeness of all Carmichael numbers up to 10^{21}. No statistically significant difference was observed between EYL, SUT, and ANU in the first CSSS test.

The second CSSS test results for original (Figure 4a) and complemented bits (Figure 4b) show the number of bits needed to witness the compositeness of all Carmichael numbers up to 10^{21}. No statistically significant difference was observed between EYL, SUT, and ANU in the second CSSS test.

The third CSSS test results for original (Figure 5a) and complemented bits (Figure 5b) show the number of bits needed to witness the compositeness of all Carmichael numbers up to 10^{21}, using the Chaitin-Schwartz compound predicate. No statistically significant difference was observed between EYL, SUT, and ANU in the third CSSS test.

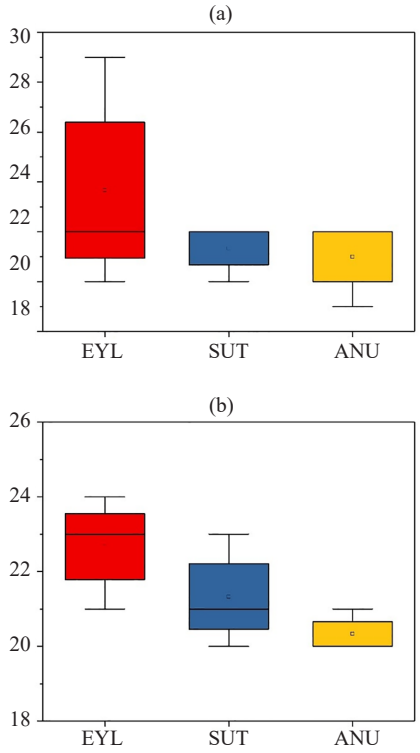

Figure 3. Results of the first CSSS test for the (a) original and (b) complemented bits

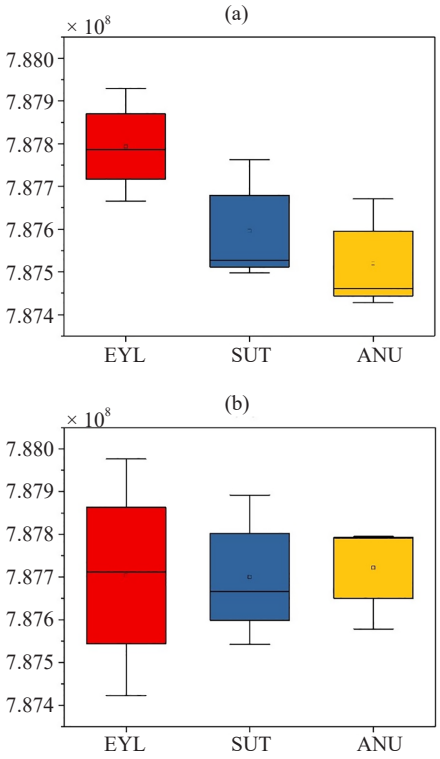

Figure 4. Results of the second CSSS test for the (a) original and (b) complemented bits

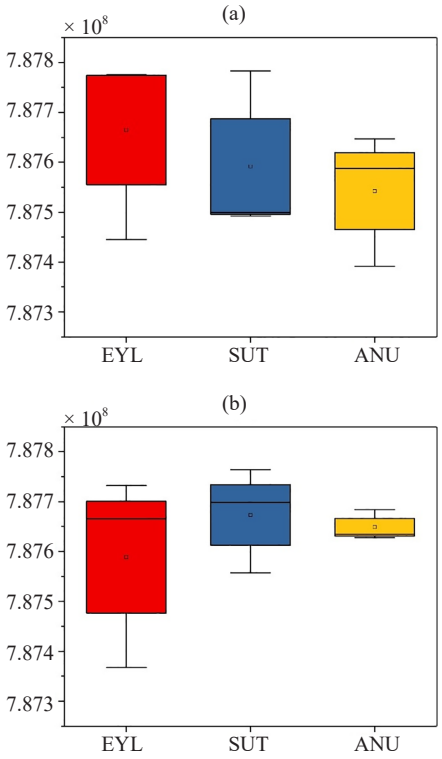

Figure 5. Results of the third CSSS test for the (a) original and (b) complemented bits

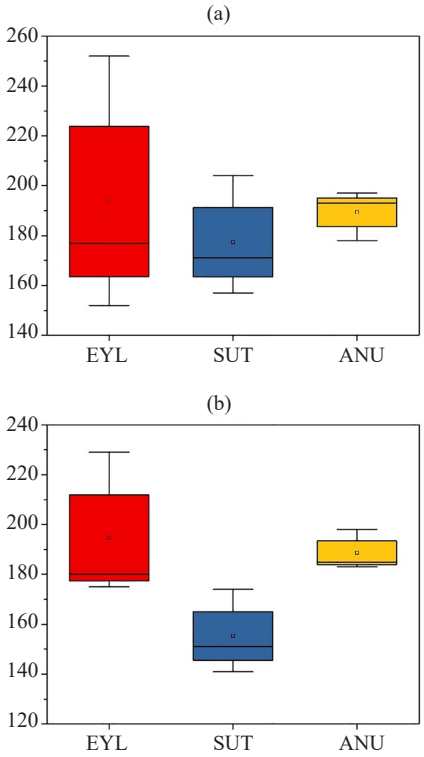

Figure 6. Results of the fourth CSSS test for the (a) original and (b) complemented bits

CSSS test results for original (Figure 6a) and complemented bits (Figure 6b) show how many times the Chaitin-Schwartz theorem has been violated for 26 test numbers including all odd composite numbers less than 100 in addition to the smallest Carmichael number (561), accumulated over repeated passes through a given random string with an incremental starting offset. No statistically significant difference was observed between EYL, SUT, and ANU in the fourth CSSS test.

We did not find any statistically significant differences in the results from the CSSS tests. As Ref. [22] showed, the output from ANU QRNG is not algorithmic random, thus we can infer that there is not any evidence of algorithmic randomness and incomputability in the output from these QRNGs.

4. Discussion

4.1 *QRNGs*

We applied various statistical test suites to five QRNGs. In order to quantitatively compare the QRNGs, we assign them a test score. For every test they pass, they earn *a* +1 score, then we divide the total score to the maximum possible score. Some test results can not be interpreted as pass/fail. These tests are excluded from the scoring.

As one can see from Figure 7, four of five QRNGs passed more than 90 percent of tests. EYL, F.H.E.S, SUT, and ANU show good randomness in terms of lacking biases and correlations. Considering the overall results, they can be used for cryptography applications. However, one should note that these tests are not complete as some of them overlap with each other and many of them have vulnerability [53]-[56].

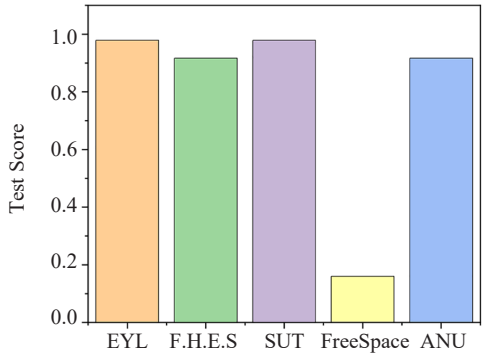

Figure 7. A comparison of QRNGs in terms of the proportion of passed tests

The results of FreeSpace tell us that the experiment was noisy and there was a bias in the detectors. Therefore, for further improvements, one should eliminate the noises and biases.

4.2 *Statistical tests*

Hypothesis tests are indeed fundamental in assessing the performance and reliability of RNGs. The aim of hypothesis testing in this context is to ascertain whether the output sequence generated by an RNG adheres to the expected characteristics of true randomness.

In statistical terms, the null hypothesis (H_0) posits that the RNG produces numbers that are indistinguishable from truly random numbers, while the alternative hypothesis (H_a) suggests that the RNG's output deviates from true randomness in some way. Through hypothesis testing, we seek to gather evidence to either support or refute the null hypothesis.

Several statistical techniques are employed in these hypothesis tests to compare the observed output of the RNG with what would be expected under the assumption of randomness. Some of the common techniques include:

Chi-Square Test: This test evaluates the uniformity of the distribution of numbers generated by the RNG. It compares the observed frequencies of numbers in different categories (e.g., bins) with the expected frequencies if the numbers were generated randomly. Deviations from expected frequencies may indicate non-random behavior.

Kolmogorov-Smirnov Test: The Kolmogorov-Smirnov test assesses whether the cumulative distribution function (CDF) of the generated numbers matches the expected uniform distribution. It quantifies the largest discrepancy between the empirical CDF of the generated numbers and the theoretical CDF of a uniform distribution.

Runs Test: The runs test examines the patterns of consecutive numbers in the generated sequence. It checks whether the number of runs (sequences of consecutive increasing or decreasing values) deviates significantly from what would be expected in a random sequence. Deviations may suggest non-random behavior.

Autocorrelation Test: Autocorrelation measures the correlation between elements of a time series at different lags. In the context of RNG testing, autocorrelation tests examine whether there are significant correlations between successive numbers in the generated sequence. The lack of significant autocorrelation is a characteristic of randomness.

Entropy Analysis: Entropy is a measure of randomness. Entropy analysis evaluates the amount of information contained in the generated sequence. High entropy indicates high randomness, while low entropy suggests patterns or predictability in the sequence.

These tests collectively provide insights into various aspects of the randomness of RNG output, helping assess their performance and reliability. Additionally, it's important to conduct multiple tests and consider their results collectively to obtain a comprehensive evaluation of the RNG's randomness properties.

4.3 *Software*

As mentioned before, random numbers are essential for many applications. It leads to increased demands for high-quality RNGs with high bit-generation rates. Although there are several schemes for the generation of random sequences, many of these are not truly random. Therefore, it is important to have standard routines for the evaluation of RNGs.

Evaluation of random data is not straightforward. As there is not a single notion of randomness, several statistical tests have to be employed to ensure the randomness of a sequence. These tests measure different aspects of randomness and are usually provided by different institutions. As they are available in sparse code forms, gathering them requires some programming knowledge. Thus, a comprehensive software package that easily gathers all these tests would be beneficial.

With these considerations, we gathered these tests into a single software package with a user-friendly GUI.

PiRATe helps researchers easily evaluate the randomness of their (Quantum) Random Number Generators. For instance, we have used PiRATe in our recent paper [29] to evaluate two different QRNGs. Nonetheless, this software is by no means complete and can be upgraded to include new statistical tests for randomness assessment.

The implemented statistical test programs are inefficient and may have bugs. This subject affects the final efficiency of PiRATe. However, the efficiency can be improved by reprogramming the original test programs.

5. Conclusions

In conclusion, we gathered a comprehensive set of statistical tests and developed an easy-to-use software to evaluate the quality of random strings from QRNGs and verify their randomness. We used this software to assess five different QRNGs and compared the results. Our results show that although the commercial EYL QRNG, the experimental SUT QRNG, and ANU QRNG do not produce algorithmic random sequences, their output is acceptable, especially for applications in cryptography where the lowest correlations are required. Due to the limited amount of random data, we could not evaluate the algorithmic randomness of F.H.E.S. However, it passed most of the tests. We also showed that the keys generated from FreeSpace QKD do not have good randomness, as they failed all statistical tests.

Author contributions

Software and analysis, S. H. and A. H. D.; writing-original draft preparation, S. H. and A. H. D.; Data preparation and collection H. A. and M. Kh; writing-review and editing, S. H.; supervision, S. A. M. All authors have read and agreed to the published version of the manuscript.

Data availability

Data used in this work are not available publicly, however, they can be provided upon reasonable requests.

Acknowledgements

We thank R. G. E. Pinch for providing us with the set of Carmichael numbers [40]. We are grateful to the authors of Refs. [21]-[22] for making publicly available their test code, and to the Centre for Quantum Computation and Communication Technology at the Australian National University for making publicly available their random strings. We also thank F.H.E.S co and INL Lab (Sharif University of Technology) for providing us with their random data.

Conflicts of interest

The authors declare no competing financial interest.

References

[1] H. C. A. van Tilborg, "Shannon's model," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer, 2011, pp. 1194-1195.
[2] D. Torrieri, *Principles of Spread-Spectrum Communication Systems*. Berlin/Heidelberg, Germany: Springer, 2005.
[3] H.-N. Hung, P.-C. Lee, and Y.-B. Lin, "Random number generation for excess life of mobile user residence time," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 3, pp. 1045-1050, 2006.
[4] M. Stipcevic, "Quantum random number generators and their applications in cryptography," in *Proceedings of advanced photon counting techniques VI*. Baltimore, Maryland, United States: SPIE, SPIE Defense, Security, and Sensing, 2012, pp. 20-34.
[5] A. J. Acosta, T. Addabbo, and E. Tena-Sánchez, "Embedded electronic circuits for cryptography, hardware security and true random number generation: An overview," *International Journal of Circuit Theory and Applications*, vol. 45, no. 2, pp. 145-169, 2017.
[6] M. N. Metropolis, and S. Ulam, "The monte carlo method," *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335-341, 1949.
[7] R. Motwani, and P. Raghavan, "Randomized algorithms," *ACM Computing Surveys (CSUR)*, vol. 28, no. 1, pp. 33-37, 1996.
[8] M. Du, Q. Chen, L. Liu, and X. Ma, "A blockchain-based random number generation algorithm and the application in blockchain games," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. Bari, Italy: IEEE, 2019, pp. 3498-3503.
[9] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 145-195, 2002.
[10] N. Walenta, M. Soucarros, D. Stucki, D. Caselunghe, M. Domergue, M. Hagerman, R. Hart, D. Hayford, R. Houlmann, M. Legré, T. McCandlish, J.-B. Page, M. Tourville, and R. Wolterman, "Practical aspects of security certification for commercial quantum technologies," in *Electro-optical and Infrared Systems: Technology and Applications XII; And Quantum Information Science and Technology*. Toulouse, France: SPIE, 2015, pp. 199-209.
[11] A. Martin, B. Sanguinetti, C. C. W. Lim, R. Houlmann, and H. Zbinden, "Quantum random number generation for 1.25-GHz quantum key distribution systems," *Journal of Lightwave Technology*, vol. 33, no. 13, pp. 2855-2859,

2015.
[12] G. Brida, M. Genovese, and I. Ruo Berchera, "Experimental realization of sub-shot-noise quantum imaging," *Nature Photonics*, vol. 4, no. 4, pp. 227-230, 2010.
[13] O. S. Magaña-Loaiza, and R. W. Boyd, "Quantum imaging and information," *Reports on Progress in Physics*, vol. 82, no. 12, pp. 124401, 2019.
[14] M. Lanzagorta, "Quantum radar," in *Synthesis Lectures on Quantum Computing*. Morgan & Claypool Publishers, 2012.
[15] D. Luong, B. Balaji, C. W. S. Chang, V. M. A. Rao, and C. Wilson, "Microwave quantum radar: An experimental validation," in *2018 International Carnahan Conference on Security Technology (ICCST)*. Montreal, QC, Canada: IEEE, 2018, pp. 1-5.
[16] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on Computing*, vol. 15, no. 2, pp. 364-383, 1986.
[17] F. James, "A review of pseudorandom number generators," *Computer Physics Communications*, vol. 60, no. 3, pp. 329-344, 1990.
[18] B. Barak, R. Shaltiel, and E. Tromer, "True random number generators secure in a changing environment," in *Cryptographic Hardware and Embedded Systems-CHES 2003: 5th International Workshop, Cologne*. Germany: Springer, Proceedings 5, 2003, pp. 166-180.
[19] M. Stipčević, and Ç. K. Koç, "True random number generators," in *Open Problems in Mathematics and Computational Science*, Ç. K. Koç, Ed. Berlin: Springer, 2014, pp. 275-315.
[20] M. Herrero-Collantes, and J. C. Garcia-Escartin, "Quantum random number generators," *Reviews of Modern Physics*, vol. 89, no. 1, pp. 015004, 2017.
[21] A. A. Abbott, C. S. Calude, M. J. Dinneen, and N. Huang, "Experimentally probing the algorithmic randomness and incomputability of quantum randomness," *Physica Scripta*, vol. 94, no. 4, pp. 045103, 2019.
[22] J. T. Kavulich, B. P. Van Deren, and M. Schlosshauer, "Searching for evidence of algorithmic randomness and incomputability in the output of quantum random number generators," *Physics Letters A*, vol. 388, pp. 127032, 2021.
[23] Duke University, Trinity College of Arts & Sciences, "Dieharder test." [Online]. Available: https://webhome.phy.duke.edu/~rgb/General/dieharder/. [Accessed Oct. 10, 2021].
[24] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, James Dray, and S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Gaithersburg, MD, USA: US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2001.
[25] C. S. Calude, *Information and Randomness: An Algorithmic Perspective*. Berlin, Germany: Springer Science & Business Media, 2013.
[26] A. Motazedifard, S. A. Madani, and N. Vayaghan, "Measurement of entropy and quantum coherence properties of two type-I entangled photonic qubits," *Optical and Quantum Electronics*, vol. 53, no. 7, pp. 378, 2021.
[27] A. Motazedifard, S. A. Madani, J. J. Dashkasan, and N. S. Vayaghan, "Nonlocal realism tests and quantum state tomography in Sagnac-based type-II polarization-entanglement SPDC-source," *Heliyon*, vol. 7, no. 6, 2021.
[28] A. Motazedifard, and S. A. Madani, "High-precision quantum transmittometry of DNA and methylene-blue using a frequency-entangled twin-photon beam in type-I SPDC," *OSA Continuum*, vol. 4, no. 3, pp. 1049-1069, 2021.
[29] S. Hajibaba, A. Dadahkhani, and S. A. Madani, "Fabrication and evaluation of high-quality and low-cost quantum random number generators," *Optics Continuum*, vol. 1, no. 7, pp. 1572-1578, 2022.
[30] M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish, and M. Boyle, "Recommendation for the entropy sources used for random bit generation," *NIST Special Publication*, vol. 800, no. 90B, pp. 102, 2018.
[31] John Walker, "ENT, A Pseudorandom Number Sequence Test Program," Jan 28th, 2008. [Online]. Available: https://www.fourmilab.ch/random/. [Accessed Oct. 10, 2021].
[32] M. Émile Borel, "Les probabilités dénombrables et leurs applications arithmétiques," *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, vol. 27, no. 1, pp. 247-271, 1909.
[33] G. Marsaglia, A. Zaman, and W. W. Tsang, "Toward a universal random number generator," *Statistics & Probability Letters*, vol. 9, no. 1, pp. 35-39, 1990.
[34] R. W. Hamming, "*Coding and Information Theory.*" Prentice-Hall, Inc., 1986.
[35] D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*, *Volume 2*. Addison-Wesley Professional, 2014.
[36] D. G. Champernowne, "The construction of decimals normal in the scale of ten," *Journal of the London*

Mathematical Society, vol. 1, no. 4, pp. 254-260, 1933.
[37] A. C. Martínez, A. Solis, R. Díaz Hernández Rojas, A. B. U'Ren, J. G. Hirsch, and I. Pérez Castillo, "Advanced statistical testing of quantum random number generators," *Entropy*, vol. 20, no. 11, pp. 886, 2018.
[38] R. Solovay, and V. Strassen, "A fast Monte-Carlo test for primality," *SIAM Journal on Computing*, vol. 6, no. 1, pp. 84-85, 1977.
[39] G. J. Chaitin, and J. T. Schwartz, "A note on monte carlo primality tests and algorithmic information theory," *Communications on Pure and Applied Mathematics*, vol. 31, no. 4, pp. 521-527, 1978.
[40] R. G. Pinch, *The carmichael numbers up to 10^{21}*. 2 Eldon Road, Cheltenham, Glos GL52 6TU, U.K.
[41] W. J. Conover, *Practical Nonparametric Statistics.* John Wiley & Sons, 1999.
[42] B. L. Welch, "The generalization of 'STUDENT'S' problem when several different population varlances are involved," *Biometrika*, vol. 34, no. 1-2, pp. 28-35, 1947.
[43] S. Hajibaba, "Pi Random Assessment Test," soheilgreen/PiRATe. [Online]. Available: https://github.com/soheilgreen/PiRATe. [Accessed Oct. 10, 2021].
[44] H. Schmidt, "Quantum-mechanical random-number generator," *Journal of Applied Physics*, vol. 41, no. 2, pp. 462-468, 1970.
[45] "Quantum Random Number Generator," EYL, Inc, *eylpartners.com*. Available: https://www.eylpartners.com/index.php/quantumrandom-number-generator/. [Accessed Nov. 10, 2021].
[46] H. Guo, W. Tang, Y. Liu, and W. Wei, "Truly random number generation based on measurement of phase noise of a laser," *Physical Review E*, vol. 81, no. 5, pp. 051137, 2010.
[47] F. H. E. S, Inc. Iran, Tehran. [Online]. Available: https://fanavartech.com/. [Accessed Oct. 12, 2021].
[48] "True Random Number Generator," *random-number.org*. Available: https://random-number.org/. [Accessed Nov. 10, 2021].
[49] T. Symul, S. M. Assad, and P. K. Lam, "Real time demonstration of high bitrate quantum random number generation with coherent laser light," *Applied Physics Letters*, vol. 98, no. 23, 2011.
[50] J. Y. Haw, S. M. Assad, A. M. Lance, N. H. Y. Ng, V. Sharma, P. K. Lam, and T. Symul, "Maximization of extractable randomness in a quantum random-number generator," *Physical Review Applied*, vol. 3, no. 5, pp. 054004, 2015.
[51] "ANU QRNG-Quantum Random Numbers," *qrng.anu.edu.au*. Available: https://qrng.anu.edu.au/. [Accessed Nov. 3, 2021].
[52] K. J. Resch, M. Lindenthal, B. Blauensteiner, H. R. Böhm, A. Fedrizzi, C. Kurtsiefer, A. Poppe, T. Schmitt-Manderbach, M. Taraba, R. Ursin, P. Walther, H. Weier, H. Weinfurter, and A. Zeilinger, "Distributing entanglement and single photons through an intra-city, free-space quantum channel," *Optics Express*, vol. 13, no. 1, pp. 202-209, 2005.
[53] E. A. Luengo, B. A. Olivares, L. J. G. Villalba, and J. Hernandez-Castro, "Further analysis of the statistical independence of the NIST SP 800-22 randomness tests," *Applied Mathematics and Computation*, vol. 459, pp. 128222, 2023.
[54] E. A. Luengo, B. A. Olivares, L. J. G. Villalba, and J. Hernández-Castro, "Weaknesses in ENT battery design," *Applied Sciences*, vol. 12, no. 9, pp. 4230, 2022.
[55] S.-J. Kim, K. Umeno, and A. Hasegawa, *Corrections of The NIST Statistical Test Suite for Randomness*. arXiv [Preprint]. 2004.
[56] C. Georgescu, E. Simion, A.-P. Nita, and A. Toma, "A view on NIST randomness tests (in) dependence," in *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI 2017)*. Targoviste, Romania: IEEE, vol. 1, no. 638, pp. 1-4, 2017.