Transport and Communications Science Journal, Vol. 75, Issue 04 (05/2024), 1644-1658

Transport and Communications Science Journal

IMPLEMENT QUANTUM RANDOM NUMBER GENERATION ON THE IBM QUANTUM COMPUTER PLATFORM

Nhu Quynh Luc*, Van Anh Le

Academy of Cryptography Techniques, 141 Chien Thang Road, Tan Trieu, Thanh Tri, Hanoi, Viet Nam

ARTICLE INFO

TYPE: Research Article
Received: 30/03/2024
Revised: 06/05/2024
Accepted: 07/05/2024
Published online: 15/05/2024
https://doi.org/10.47869/tcsj.75.4.14
* *Corresponding author*
Email: quynhln@actvn.edu.vn; lucnhuquynh69@gmail.com

Abstract. Random numbers are a crucial component of any encryption activity in modern cryptography. Quantum Random Number Generators (QRNGs) produce truly random output strings to replace pseudo-random ones. The principle of QRNG relies on measuring qubit states, which excel in quantum computing applications, particularly on IBM's quantum computing platform. To construct a random number generator, the authors utilized IBM Q Experience's Qiskit quantum development toolkit. We developed QRNG applications on IBM quantum computers (7-qubit, 16-qubit, and 127-qubit) and tested the program's functionality on these quantum computing platforms. The quality assessment of the random strings was conducted according to NIST and AIS-31 standards. For NIST standards, to achieve good quality, the output string must reach a minimum of 1,593,088 bits to pass 16 tests per SP800-22 standard. According to AIS-31 standards, to achieve good quality, the output string must reach a minimum of 8,000,000 bits to pass 8 tests of the standard.

Keywords: AIS-31, Hadamard gate, Measurement, NIST SP 800-22, QRNG, Qubit.

1. INTRODUCTION

The random number generator (RNG) is indispensable in cryptography technique systems to safeguard information confidentiality because the key being generated by RNG is a sequence of random numbers [1]. RNG is divided into two categories: (1) pseudo-random number generators (PRNG) and (2) true random number generators (TRNG) [2].

(1) Currently, PRNG is commonly used in cryptographic systems. It produces sequences of numbers that are nearly random but not genuinely random (essentially, pseudo-random sequences) [3]. PRNG operates based on optimized algorithms and provides deterministic statistical characteristics in their output. Therefore, PRNG can achieve high throughput and easily integrate into cryptographic products [4]. However, PRNGs fall short in terms of security; if an attacker possesses the seed value, the attacker can predict the next output [4].

(2) Compared with PRNG, TRNG generators address the limitations of PRNG generators by using highly uncertain random sources rooted in physical phenomena (nature ensures randomness, independence, and co-performance - not a random stopping sequence) [5]. The unpredictability of TRNG stems from its exploitation of inherent uncertainty, making the next state unguessable. This necessitates deriving the actual random sequence from a tangible physical source. In particular, the random source containing physical processes plays a central role in the TRNG generator, because the quality of the random number in the output depends closely on the quantity entropy in this block.

Recently in the period from 2019-2024, with the development of electronic technology, the quantum random number generator (QRNG) has garnered significant attention and research interest from scientists and the scientific community [6]. The using of a Quantum random number generator (QRNG) aims to leverage the advantages of optical properties proposed by utilizing various random sources, including (1) Novel optical devices targeting single photon detection [7], [8]; (2) Devices measuring vacuum state fluctuations noise [9], [10]; (3) A laser phase-oscillator device [11] and amplify spontaneously emitted noise [12],... A QRNG generator typically comprises two primary components: (1) a random source and (2) a measurement unit. With quantum technology, the random source is obtained from the superposition state in the underlying measurement whose measurement results are unpredictable to obtain random numbers due to Heisenberg's principle. On the other hand, propelled by significant advancements in quantum computing, QRNG have undergone refinement and adaptation to synergize with evolving technologies [13].

Presently, several major companies and corporations have introduced cloud-based quantum computers, offering users more efficient execution of quantum programs computers [14]. Notably, QRNG generators, highlighted in numerous studies, excel in generating random numbers, particularly for cryptographic purposes. Quantum superposition states, fundamental in quantum computing, play a crucial role in application development. Quantum superposition states, as elucidated by the author team, involve inputting a qubit state through a Hadamard gate to obtain a superposition state (a superposition of two basic states $|0\rangle$ and $|1\rangle$ [15]). Subsequently, measuring the output qubit yields 0 and 1 with equal probabilities. Most of the testing and assessing the quality of random numbers generated using quantum technology have not been proposed to be promulgated into standards [16]. Currently, most research is utilizing the NIST and AIS-31 standards [17] to assess the quality of random numbers.

In this research, the author's team centered on investigating and scrutinizing the design of a QRNG utilizing optical methods tailored to meet cryptographic standards. Subsequently,

executed and simulated the QRNG's random number generation process on an IBM quantum computer. Accordingly, the quality of generated random numbers is evaluated using the SP 800-22 standard (a publication of the National Institute for Standards and Technology) and AIS-31 (a publication of the German Federal Office for Information Security (BSI)).

2. RESEARCH RELATED TO QRNG GENERATORS

2.1. The general principle of quantum random number generator (QRNG)

The majority of random number generators adhere to a fundamental principle that can be segmented into three key blocks: (1) Entropy Source; (2) Measurement; and (3) Post-processing. According to publications [6] and [18], the operating process principle is understood as follows:

- ✓ Input is a source of entropy using physical processes as sources of entropy including randomness in mouse movements and keyboard typing, noise in electronic circuits, or some chaotic systems.
- ✓ Following the input from the entropy source, the next step involves measurement to extract randomness which relies on the precision of the measurement device employed. The higher the measurement quality the better the result of the random sequence number.
- ✓ Lastly, the output of the measurement system may consist of raw random numbers, then undergo processing through a post-processing module to enhance their statistical and security properties.

In the publication [6] several principles are proposed for creating diverse types of QRNGs based on distinct quantum random sources: encompass generating random numbers through the time arrival principle of emission states (utilizing qubit state measurements [19], [20]), employing photon detection or vacuum fluctuations ([21], [22]), utilizing phase modulation principles in spontaneous emission [9], developments in non-optical QRNGs or QRNGs independent of specific devices [23].

Table 1. Some methods for generating quantum random numbers.

Some methods for generating quantum random numbers	Limitations
QRNG based on radioactive decay	A radioactive source as a randomizer can present processing challenges.
QRNG based on electronic noise	In practice, it's difficult to distinguish between shot noise and thermal noise, hence extracting randomness from shot noise encounters this drawback.
QRNG is based on measuring qubits in a superposition state	The detectors will be inactive for a certain period, during which they cannot detect photons.
QRNG based on measuring photon arrival time	The accuracy of the measured arrival time will limit the random generation speed.
QRNG utilizing randomness in the zero-point oscillations of the electric field	The speed is limited by the rate of the noise reduction process.
QRNG utilizing Raman scattering phenomenon	Sensitive to temperature changes

Until now, several methods have been applied to construct various types of QRNGs based on different quantum noise sources. In a publication [24], Leilei Huang and colleagues integrated four types of distinct quantum random number generators on Alibaba's cloud server to enhance network security, including numbers derived from single-photon detection, photon counting detection, phase oscillations, and vacuum fluctuations. However, the platform had to run continuously for over a year to improve results. Alternatively, QRNGs formed from a superluminescent diode with temperature-enhanced effects have been studied numerically [25]. However, the limitation of this QRNG is the minimal entropy reduction due to increased temperature in the SLED, resulting in unsuccessful outputs in NIST statistical tests and post-processing procedures unable to counteract temperature increases. Overall, most QRNG generators currently deployed rely heavily on the operation of these physical devices and require post-processing procedures to improve the characteristics of the output sequence. In Table 1, the authors outlined the limitations of some existing methods for generating random numbers.

According to the publication [26], Tamura and colleagues have delineated the advantages, drawbacks, and applicability of QRNGs based on qubit state measurements, along with their operation on IBM quantum computers. With QRNG generators deployed on quantum computers, there is no need for any post-processing for the output random sequence. In this study, the author's team concentrated on researching, analyzing, and evaluating the effectiveness of implementing QRNGs based on qubit state measurements on IBM quantum computers, as extensively discussed in this research endeavor.

2.2. The QRNG generator principle based on Qubit state measurement

Figure 1 shows the actual QRNG generator based on Qubit state measurement, the principle is implemented by Qubit state measurement and generates random numbers by measuring qubits in a superposition state. The operation of this QRNG hinges on the principle of quantum entanglement, in conjunction with the measurement theorem, which posits that the quantum state will collapse into one of the basic states upon measurement. This collapse can be elucidated as follows: Initially, a photon (representing an initial qubit state) is prepared through a polarizing beam splitter (PBS) in a superposition state, wherein it splits into horizontal (H) and vertical (V) polarizations (as expressed by the equation $\frac{1}{\sqrt{2}}(|H\rangle + |V\rangle)$. The polarizing beam splitter (PBS) transmits horizontally polarized light and reflects vertically polarized light. Then, two single-photon detectors (SPDs) are used to generate random numbers. The operation of the QRNG based on measuring qubit states consists of two main steps:

- ✓ ***Step 1 (Creating qubits in a superposition state):*** The initial input here is a basis bit $|0\rangle$ transform passing through the Hadamard gate, resulting in a qubit existing in a quantum superposition state. Here, a qubit in a quantum superposition state will have the probability of appearing in the state $|0\rangle$ is $P(|0\rangle) = \left(\frac{1}{\sqrt{2}}\right)^{2,}$ and similarly, for the probability of the state $|1\rangle$ is $P(|1\rangle) = \left(\frac{1}{\sqrt{2}}\right)^{2}$. Consequently, the output states $|0\rangle$ or $|1\rangle$ is generated randomly, independently, and with equal probabilities. Kentaro Tamura and colleagues have proposed quantum random number generators capable of producing independent sequences with identical inputs, thereby mitigating the risk of predicting the generator's output from a fixed input [26].
- ✓ ***Step 2 (Measuring the qubit in a superposition state):*** After creating the quantum superposition state, a measurement is conducted, collapsing the superposition state into

one of the two basis states, either $|0\rangle$ or $|1\rangle$, with equal probabilities. Consequently, the output cannot be forecasted in advance (ensuring randomness, independence, and equiprobability of the output sequence). However, the publication [27] highlights the limitations of quantum technology, particularly the constraints of resources in quantum computers (IBM quantum computers and IBM quantum virtual machines). To generate extensive random number sequences, this process necessitates numerous repetitions, which, when executed using quantum simulation tools like Qiskit on the IBM platform, demand the support of high-performance computers and may consume considerable time [28].

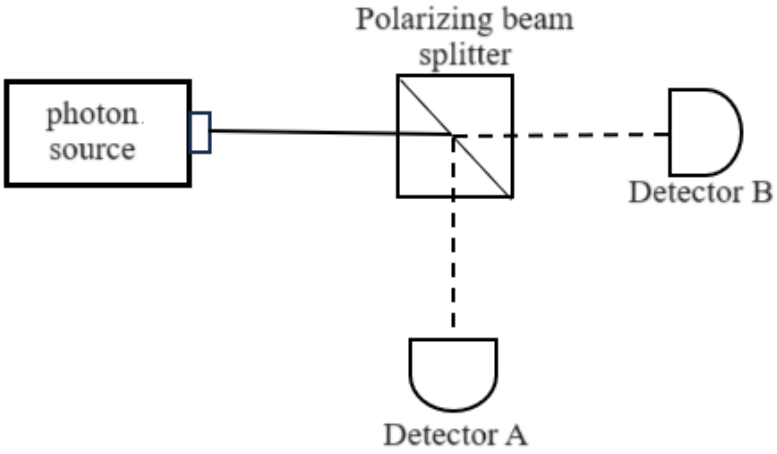


Figure 1. Practical QRNG based on qubit state measurement.

The QRNG generator based on Qubit state measurement has the advantage of a simple implementation method and the ability to be installed on current IBM quantum computers and quantum virtual computers. The process requires firing single photons, enabling the generation of one random bit per photon. Nonetheless, the speed of generating random sequences is contingent upon and restricted by detector performance. As a result, the speed of generating random numbers is constrained to tens of Mbps, falling short of the demands of high-speed applications such as high-speed quantum key distribution (QKD). Recently, many studies have also demonstrated the potential for manufacturing basic quantum gates. If the input state of the Hadamard gate is $|0\rangle$ or $|1\rangle$, the superposition state $|\psi\rangle$ generated by the Hadamard gate is respectively: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ with the probability of obtaining the state $|0\rangle$ and $|1\rangle$ are equal [15]. However, the QRNG based on measuring qubit states uses the Hadamard gate as the polarizing beam splitter (PBS) [27], [26]. However, QRNG based on measuring qubit states utilizes the Hadamard gate as the polarizing beam splitter (PBS), acting as a software application on a quantum computer.

3. RESULTS AND DISCUSSION

3.1. Design, construction, and implementation of the QRNG based on measuring qubit states

Based on the analyses conducted, the authors have determined that constructing a QRNG based on measuring qubit states represents a foundational approach that can be simulated and

executed on current IBM quantum computers without necessitating intricate physical devices. The construction of this QRNG by the author group involved the following three steps:

- ✓ **Step 1**: Prepare all qubits in a superposition state between two basic states, ensuring equal probabilities for both states, using the Hadamard gate.
- ✓ **Step 2**: Measurements on all qubits and gather the results.
- ✓ **Step 3**: Return to Step 1.

In the implementation and execution process of the QRNG generator, the authors' team utilized a computer with the following configuration: Intel(R) Core(TM) i5-1035G1 - 1.00GHz, 12GB RAM, connected to an IBM platform computer. The IBM platform computer had the following specifications: 32Qubit, Max shots = 20000, Max circuits = 300, Max qubits per pulse gate = 3, and Max channels gate = 9, accessed via the provided API code. In this study, the authors tested the operation of all three quantum circuits (7-qubit, 16-qubit, and 127-qubit) on the IBM quantum computer via API, as well as on a local computer with an existing quantum development toolkit installed. The input for the application was the basis qubit, constructed in a superposition state using Hadamard gates, exploiting the resulting probabilities to generate random numbers. The detailed dynamic flow of the modulo for the random number generator is depicted in Figure 2.

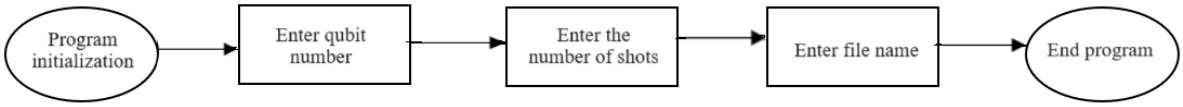


Figure 2. Operational model, simulating the quantum random number generator.

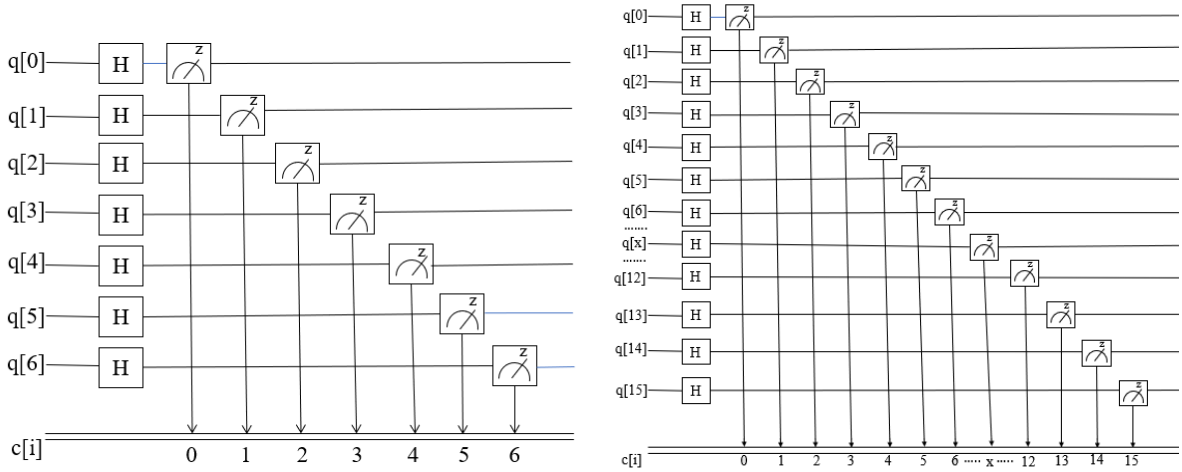
In this program, the authors have developed a quantum random number generator (QRNG) application based on measuring the qubit state. The number of input qubits determines the base circuit, which is limited to three values: 7-qubit, 16-qubit, and 127-qubit. The "shots" value represents the number of repetitions required to generate n sequences of random numbers, with each sequence's length corresponding to the input qubit value. For instance, if the input qubit value for generating the base circuit is 7-qubit and the shot value is 1, the program will produce a bit string comprising bits 0 and 1, with a length of 7. Similar to the 16-qubit and 127-qubit base circuits. The QRNG application is structured into three main modules:

- ✓ Modulo 1: QRNG Generator on the IBM quantum computer (7-qubit).
- ✓ Modulo 2: QRNG Generator on the IBM virtual computer (16-qubit).
- ✓ Modulo 3: QRNG Generator on the IBM virtual computer (127-qubit).

The QRNG (Quantum Random Number Generator) based on qubit state measurements operates with Hadamard gates serving as pivotal components akin to polarizing beam splitters (PBS). The number and intensity of photons fired significantly determine the operational speed of the random number generation circuit. When a qubit traverses a Hadamard gate, the output results in the qubit existing in a superposition state, with equal probabilities for the qubit $|0\rangle$ or qubit $|1\rangle$, each at a 50% likelihood. To generate a large output sequence, spanning up to millions of bits, continuous operation of the circuit is essential, achieved through repeated steps. At the output, all possible combinations may appear, with the probabilities of occurrence varying with each execution. The initialization program determines the number of bits in the random sequence, thereby dictating the required number of Hadamard gates. The

length of the random sequence, upon program conclusion, is calculated as follows: Output sequence length = number of qubits * number of firings. Upon providing the input parameters, the program generates an output file, saved under the designated name. Here are some results achieved by the QRNG generator based on qubit state measurements, operating with 3 modules, where the number of firings for the 3 modules is set to be the same.

a) Modulo QRNG (7-qubit) setup on the IBM computer (7-qubit).



a) The circuit schematic of the QRNG (7-qubit) b) The circuit schematic of the QRNG (16-qubit)

Figure 3. Circuit diagram of the QRNG generator based on qubit state measurements.

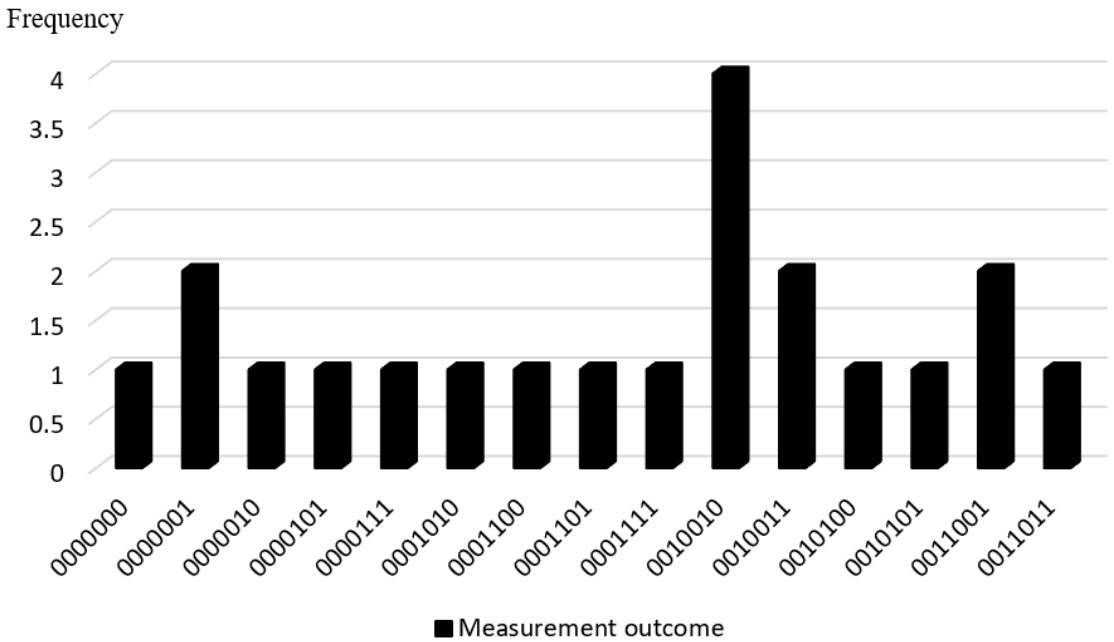


Figure 4. The occurrence ratio of actual random numbers in the 7-qubit quantum state.

Figure 3a depicts the detailed circuit design of the QRNG generator on the IBM quantum computer (7-qubit). For the QRNG circuit (7-qubit), it requires 7 Hadamard gates and 7

measurement operations. Therefore, to generate a sequence of random numbers with a length of 2^7 bits ranging from 0 to 2^7, the computer needs a 7-qubit quantum register. The results of the QRNG generator when operating with superposition states are illustrated in Figure 4. In this case, the occurrences of possible combinations from the corresponding 7-qubit quantum random number generation circuits are determined.

b) Modulo QRNG (16-qubit) setup on the IBM virtual computer (16-qubit).

Figure 3b illustrates the detailed circuit design of the QRNG generator on the IBM virtual quantum computer (16-qubit). In this setup, the QRNG circuit (16-qubit) requires 16 Hadamard gates and 16 measurement operations. Therefore, if generating a sequence of random numbers with a length of bit 2^{16} with values ranging from 0 to 2^{16}, the computer needs a 16-qubit quantum register. The results of the QRNG generator operating with superposition states are depicted in Figure 5. This figure illustrates the occurrences of possible combinations from the corresponding 16-qubit quantum random number generation circuits.

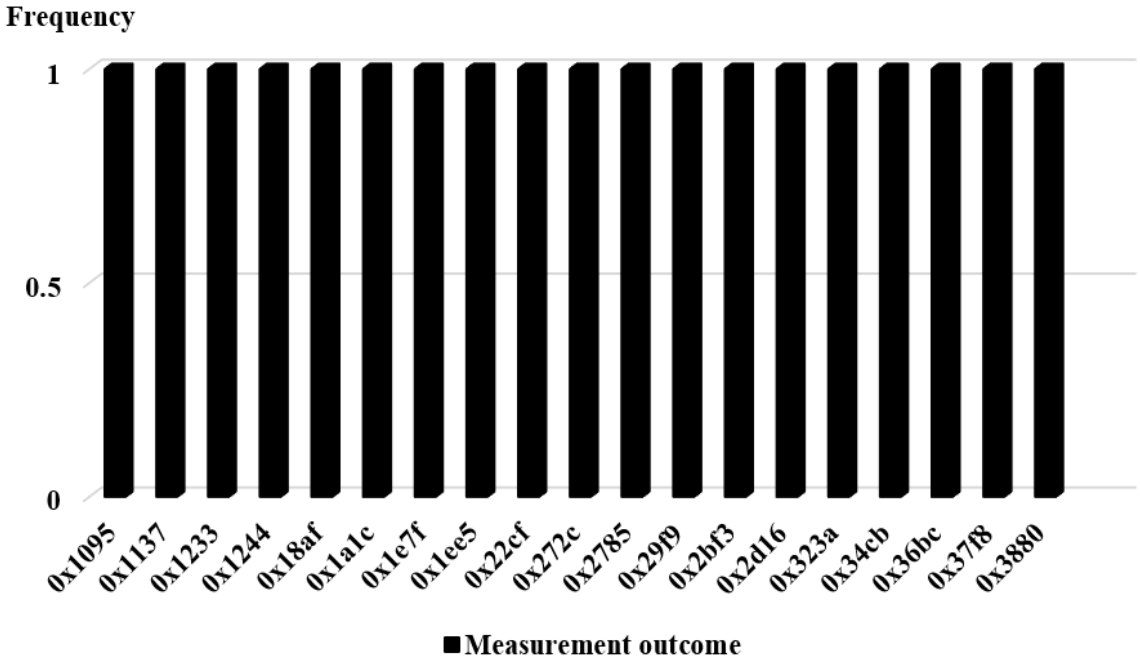


Figure 5. The occurrence ratio of actual random numbers in the 16-qubit quantum state.

c) Modulo QRNG (127-qubit) setup on the IBM virtual computer (127-qubit).

QRNG generator on the IBM virtual quantum computer (127-qubit) entails the utilization of 127 Hadamard gates and 127 measurement operations. Therefore, if generating a sequence of random numbers with a length of bits 2^{127} with values ranging from 0 to 2^{127}, the computer needs a 127-qubit quantum register. The results of the QRNG generator when operating with superposition states are illustrated in Figure 6. In this case, the occurrences of possible combinations from the corresponding 127-qubit quantum random number generation circuits are determined.

Figure 7 illustrates the result file containing the sequence of random numbers generated by the QRNG based on qubit state measurements. Throughout the experimental process, the authors conducted multiple trials, resulting in output data ranging up to 1 million qubits (for

quality assessment checks following NIST standards) and 8 million bits (for quality assessment checks following AIS-31 standards). The output of the QRNG is presented in the form of regular bits, constituting a sequence of bits 0 and bits 1.

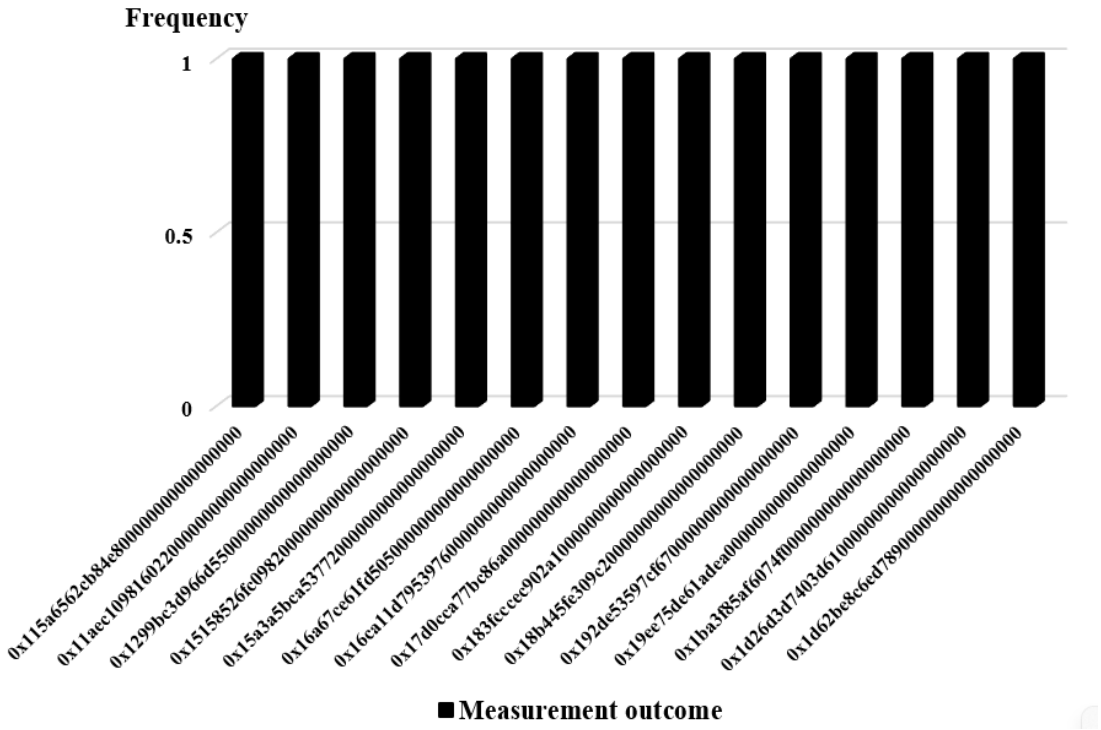


Figure 6. The occurrence ratio of actual random numbers in the 127-qubit quantum state.

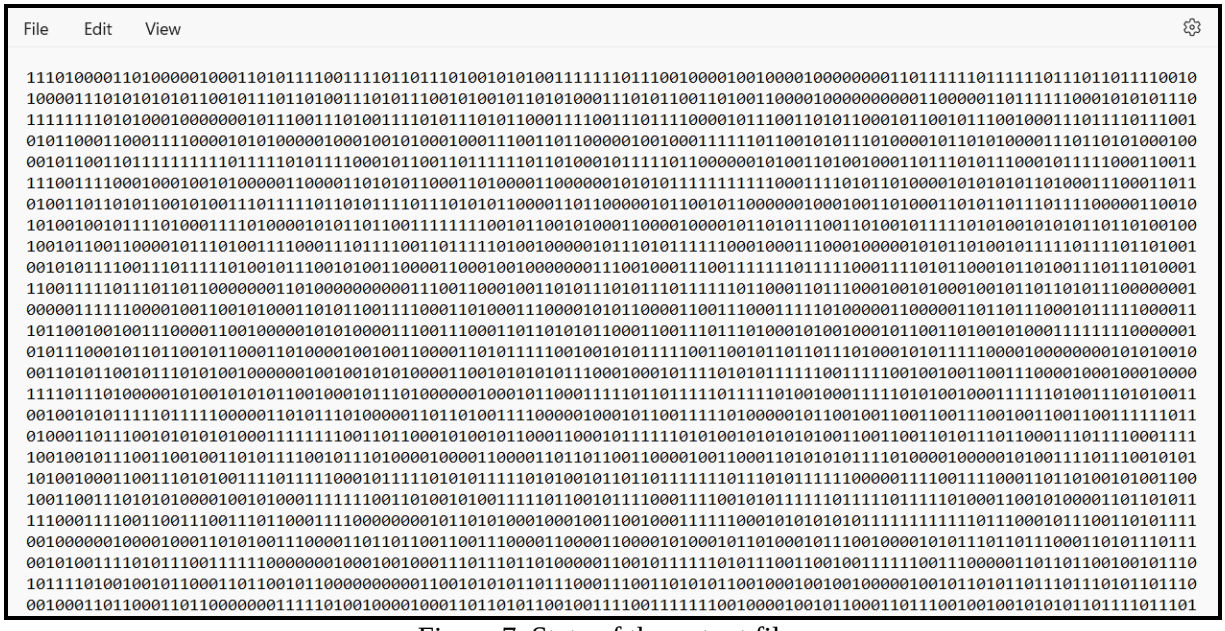


Figure 7. State of the output file.

Figure 8 illustrates clear differences in both the generation time and the generated random number sequences when using the same number of shots for modules constructed on IBM quantum computers (7-qubit), IBM virtual computers (16-qubit), and IBM virtual computers (127-qubit). As the number of shots increases, the time required to generate random numbers also noticeably increases.

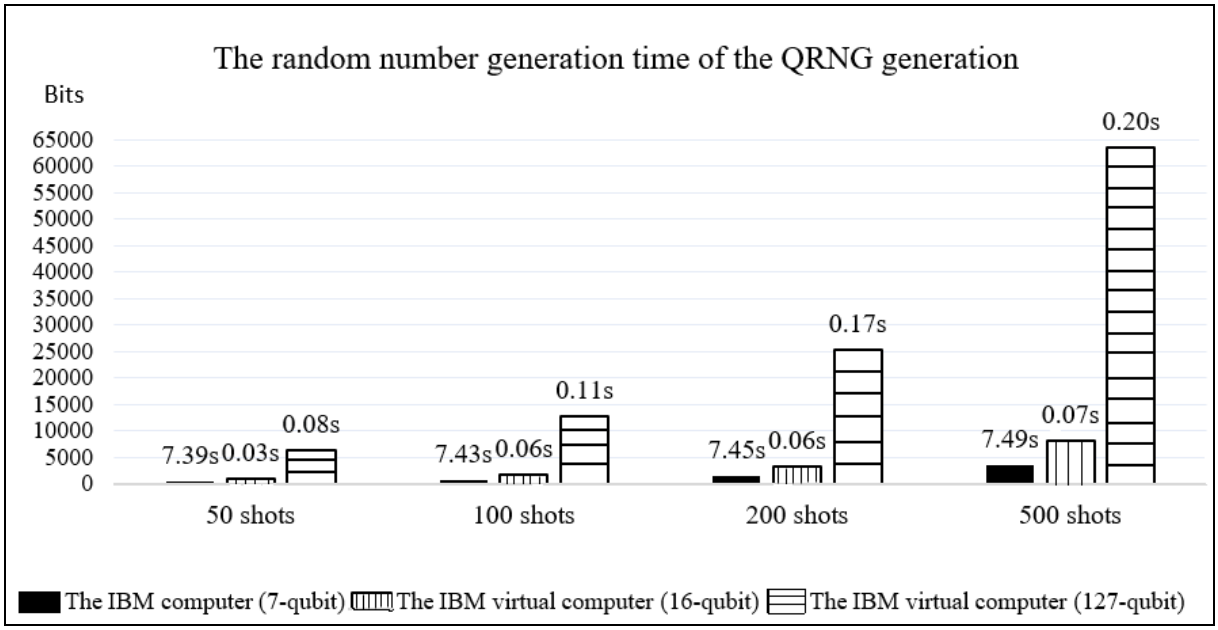


Figure 8. Execution time results of the QRNG on various IBM quantum computer platforms.

After conducting their analysis, the authors noted that with the same number of shots, the occurrence ratio of actual random numbers in the case of 127 qubits appeared to be the most uniform. This observation can be explained straightforwardly: while actual 7-qubit random numbers span values ranging from 0 to 2^7, the range for actual 127-qubit random numbers is significantly larger, spanning from 0 to 2^{127}. Consequently, the occurrence ratio of actual random numbers in the 127-qubit scenario is the most uniform among the three scenarios. Additionally, the authors observed differences in execution time when running the QRNG program on a virtual machine (a local computer with Qiskit installed, an open-source software development kit that runs notably faster than sending jobs to IBM quantum computers for execution). This discrepancy stems from the time required to transmit and receive results from the quantum computer back to the user's computer [29], [30].

3.2. The achieved results can be evaluated using the NIST and AIS-31 standards.

According to the publication [18], [29], Vaishnavi Kumar and colleagues employed an input of 24 qubits and 65536 shots to generate random numbers, producing 24 × 65536 bits in a single execution. In contrast, the author's group conducted experiments using input qubit numbers of 7, 16, or 127 qubits in their QRNG modulo. However, due to computational constraints, the number of shots was limited to 20000 per execution. To generate a random number sequence of standard length for testing according to NIST or BSI evaluation standards, the authors utilized a loop in the QRNG modulo. This approach involved dividing the program into multiple runs if the number of shots exceeded 20000, ensuring that the shot

values for each run did not surpass the limit. Subsequently, the resulting values were concatenated into a unified sequence following a specific sequence. To evaluate the quality of the generated random number sequence, the authors utilized the Random Numbers Analyzer tool, developed in compliance with the standards of NIST (National Institute of Standards and Technology) and AIS-31 (Federal Office for Information Security in Germany).

Table 2. Quality check of the generated sequence according to NIST standards.

The standards of NIST (SP800-22)	**The IBM computer (7-qubit)**		**The IBM virtual computer (16-qubit)**		**The IBM virtual computer (127-qubit)**	
	p-value	**Results**	**p-value**	**Results**	**p-value**	**Results**
01. Frequency Test (Monobit)	0.5741	Pass	0.9060	Pass	0.6369	Pass
02. Frequency Test within a Block	0.6630	Pass	0.3722	Pass	0.2532	Pass
03. Run Test	0.2826	Pass	0.4715	Pass	0.3471	Pass
04. Longest Run of Ones in a Block	0.1444	Pass	0.037	Pass	0.4765	Pass
05. Binary Matrix Rank Test	0.2974	Pass	0.4245	Pass	0.2912	Pass
06. Discrete Fourier Transform (Spectral) Test	0.6662	Pass	0.5693	Pass	0.3540	Pass
07. Non-Overlapping Template Matching Test	0.9727	Pass	0.3812	Pass	0.8851	Pass
08. Overlapping Template Matching Test	0.9442	Pass	0.1416	Pass	0.9099	Pass
09. Maurer's Universal Statistical test	0.4865	Pass	0.9595	Pass	0.5672	Pass
10. Linear Complexity Test	0.1142	Pass	0.7531	Pass	0.9645	Pass
11. Serial test	0.7427	Pass	0.1065	Pass	0.5978	Pass
12. Approximate Entropy Test	0.3557	Pass	0.0873	Pass	0.0723	Pass
13. Cumulative Sums (Forward) Test	0.6155	Pass	0.7561	Pass	0.6970	Pass
14. Cumulative Sums (Reverse) Test	0.6155	Pass	0.7561	Pass	0.6970	Pass
15. Random Excursions Test	0.9795	Pass	0.5240	Pass	0.0463	Pass
16. Random Excursions Variant Test	0.9389	Pass	0.1056	Pass	0.8596	Pass

Table 2 provides a comprehensive overview of the results obtained from evaluating the quality of the random number sequences generated by the author's modulo program. In this evaluation, the output file data comprises 1,593,088 bits, and the authors conducted quality assessment checks on multiple output sequences with varying numbers of bits. The findings indicate that the evaluated data successfully passed 16 tests according to the SP800-22A standard of the National Institute of Standards and Technology (NIST).

Table 3 furnishes comprehensive evaluations regarding the quality of the random number sequences generated from the modulo to ensure compliance with the AIS-31 standard. As per the AIS-31 standard, the output file data should approximate 8 million bits. In this context, the output file data surpasses the 8 million bits threshold, with the authors conducting quality assessments on multiple sequences featuring random number sequence data of up to 11,151,616 bits. The findings indicate that for data files exceeding 8 million bits, all output files successfully passed 8 tests following the AIS-31 standard of the Federal Republic of Germany. It's notable that the NIST SP800-22 standard primarily employs statistical methods to evaluate the quality of random numbers, whereas the AIS-31 standard is tailored to assess the quality of actual random number sequences. The results obtained from applying tests from both the NIST and AIS-31 test suites by the BSI demonstrate that the random numbers generated by the QRNG generator application developed by the team fully satisfy the standards for authentic random numbers.

Table 3. Quality assessment according to AIS-31 standard.

Test methods (AIS-31)		**The IBM computer (7-qubit)**	**The IBM virtual computer (16-qubit)**	**The IBM virtual computer (127-qubit)**
Test T0	Disjointness test	Pass	Pass	Pass
Test T1	Monobit test	Pass	Pass	Pass
Test T2	Poker test	Pass	Pass	Pass
Test T3	Run test	Pass	Pass	Pass
Test T4	Long Run test	Pass	Pass	Pass
Test T5	Autocorrelation test	Pass	Pass	Pass
Test T6	Uniform distribution test	Pass	Pass	Pass
Test T7	Comparative multinomial test	Pass	Pass	Pass
Test T8	Entropy test	Pass	Pass	Pass

While employing the IBM Platform device as a quantum random number generator might not be pragmatic, generating quantum random numbers with this device could hold significant implications and remains an area of interest for future research endeavors, both within the team and the broader scientific community.

4. CONCLUSION

In this study, the authors conducted an analysis and evaluation of a quantum random number generator (QRNG) utilizing the IBM quantum computer across various configurations: 7-qubit, 16-qubit, and 127-qubit systems. Building upon this analysis, the authors proceeded to develop a QRNG application based on qubit state measurement specifically tailored for deployment on these IBM quantum computer platforms. The program underwent rigorous testing on each of these quantum computer configurations. The obtained results indicate that the QRNG program, based on qubit state measurement, successfully operates on these platforms, generating output sequences of considerable length, reaching up to 1,593,088 bits. To assess the quality of the generated random sequences, the authors employed both NIST and AIS-31 standards. According to the NIST standards, to achieve satisfactory quality, the output sequence must surpass a minimum length of 1,593,088 bits to pass 16 tests stipulated by the NIST SP800-22 standard. In contrast, following the AIS-31 standards, a minimum sequence length of 8,000,000 bits is required to achieve satisfactory quality and pass 8 tests mandated by the AIS-31 standard.

ACKNOWLEDGMENT

The authors acknowledge the Academy of Cryptography Techniques for supporting this work under grant number 08/2024/CS.

REFERENCES

[1]. K. Gu, X. Dong, L. Wang, Efficient traceable ring signature scheme without pairings, Advances in Mathematics of Communications, 14 (2020) 207–232. https://doi.org/10.3934/amc.2020016.
[2]. J. Sen Teh, W. Teng, A. Samsudin, J. Chen, A post-processing method for true random number generators based on hyperchaos with applications in audio-based generators, Frontiers of Computer Science, 14 (2020) 146405. https://doi.org/10.1007/s11704-019-9120-2.
[3]. L. Deng, D. Bowman, Developments in pseudo-random number generators, WIREs Computational Statistics, 9 (2017) 1404. https://doi.org/10.1002/wics.1404.
[4]. A. Shukla *et al.*, A True Random Number Generator for Probabilistic Computing using Stochastic Magnetic Actuated Random Transducer Devices, in 2023 24th International Symposium on Quality Electronic Design (ISQED), (2023) 1–10. https://doi.org/10.1109/ISQED57927.2023.10129319.
[5]. Ç. K. Koç, Ed., Cryptographic Engineering, Boston, MA: Springer US, (2009). https://doi.org/10.1007/978-0-387-71817-0.
[6]. V. Mannalath, S. Mishra, A. Pathak, A Comprehensive Review of Quantum Random Number Generators: Concepts, Classification and the Origin of Randomness, 22 (2023) 439. https://doi.org/10.1007/s11128-023-04175-y.
[7]. M. A. Wayne, P. G. Kwiat, Low-bias high-speed quantum random number generator via shaped optical pulses, Optics Express, 18 (2010) 9351. https://doi.org/10.1364/OE.18.009351.
[8]. M. Fürst, H. Weier, S. Nauerth, D. G. Marangon, C. Kurtsiefer, H. Weinfurter, High speed optical quantum random number generation, Optics Express, 18 (2010) 13029. https://doi.org/10.1364/OE.18.013029.
[9]. Y. Shen, L. Tian, H. Zou, Practical quantum random number generator based on measuring the shot noise of vacuum states, Physical Review A, 81 (2010) 063814.

https://doi.org/10.1103/PhysRevA.81.063814.
[10]. Q. Zhou, R. Valivarthi, C. John, W. Tittel, Practical quantum random number generator based on sampling vacuum fluctuations, 1 (2018) 1-6. http://arxiv.org/abs/1703.00559
[11]. B. Qi, Y.-M. Chi, H.-K. Lo, L. Qian, High-speed quantum random number generation by measuring phase noise of a single-mode laser, Optics Letters, 35 (2010) 312. https://doi.org/10.1364/OL.35.000312.
[12]. W. Wei, G. Xie, A. Dang, H. Guo, High-Speed and Bias-Free Optical Random Number Generator, IEEE Photonics Technology Letters, 24 (2012) 437–439. https://doi.org/10.1109/LPT.2011.2180521.
[13]. Y. Alexeev *et al.*, Quantum Computer Systems for Scientific Discovery, PRX Quantum 2, 2 (2021) 017001. https://doi.org/10.1103/PRXQuantum.2.017001.
[14]. J. Preskill, Quantum Computing in the NISQ era and beyond, 2 (2018) 79. https://doi.org/10.22331/q-2018-08-06-79.
[15]. Y. Wang, Quantum Computation and Quantum Information, Statistical Science, 27 (2012) 373-394. https://doi.org/10.1214/11-STS378.
[16]. L. E. Bassham *et al.*, A statistical test suite for random and pseudorandom number generators for cryptographic applications, Gaithersburg, 20899 (2010) 1-131. https://doi.org/10.6028/NIST.SP.800-22r1a.
[17]. W. Schindler, A Proposal for Functionality Classes for Random Number Generators, (2022) 1-239.
https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Certification/Interpretations/AIS_31_Functionality_classes_for_random_number_generators_e.html.
[18]. R. Biswas, D. Roy Talukdar, U. Roy, Verifying the Reliability of Quantum Random Number Generator: A Comprehensive Testing Approach, SN Computer Science, 5 (2024) 140. https://doi.org/10.1007/s42979-023-02323-w.
[19]. M. Wahl, M. Leifgen, M. Berlin, T. Röhlicke, H.-J. Rahn, O. Benson, An ultrafast quantum random number generator with provably bounded output bias based on photon arrival time measurements, Applied Physics Letters, 98 (2011) 171105. https://doi.org/10.1063/1.3578456.
[20]. S. Li, L. Wang, L.-A. Wu, H.-Q. Ma, G.-J. Zhai, True random number generator based on discretized encoding of the time interval between photons, Journal of the Optical Society of America A, 30 (2013) 124. https://doi.org/10.1364/JOSAA.30.000124.
[21]. B. Sanguinetti, A. Martin, H. Zbinden, N. Gisin, Quantum Random Number Generation on a Mobile Phone, Physical Review X, 4 (2014) 031056. https://doi.org/10.1103/PhysRevX.4.031056.
[22]. T. Symul, S. M. Assad, P. K. Lam, Real time demonstration of high bitrate quantum random number generation with coherent laser light, Applied Physics Letters, 98 (2011) 231103. https://doi.org/10.1063/1.3597793.
[23]. Arvind Krishna, 2022 Annual 10-k report, IBM Corporation, (2022). https://www.ibm.com/annualreport/assets/downloads/IBM_Annual_Report_2022.pdf
[24]. L. Huang, H. Zhou, K. Feng, C. Xie, Quantum random number cloud platform, npj Quantum Information, 7 (2021) 107. https://doi.org/10.1038/s41534-021-00442-x.
[25]. Y. Li *et al.*, Analysis of the effects of temperature increase on quantum random number generator, The European Physical Journal D, 75 (2021) 69. https://doi.org/10.1140/epjd/s10053-021-00087-7.
[26]. K. Tamura, Y. Shikano, Quantum Random Number Generation with the Superconducting Quantum Computer IBM 20Q Tokyo, Cryptology ePrint Archive, 30 (2020) 1–13. https://eprint.iacr.org/2020/078
[27]. V. Mannalath, S. Mishra, A. Pathak, A Comprehensive Review of Quantum Random Number Generators: Concepts, Classification and the Origin of Randomness, 22 (2023) 439. http://arxiv.org/abs/2203.00261
[28]. E. F. C. Amira Abbas, Learn Quantum Computation Using Qiskit, (2020). https://qiskit.org/textbook/
[29]. V. Kumar, J. B. B. Rayappan, R. Amirtharajan, P. Praveenkumar, Quantum true random number

generation on IBM's cloud platform, Journal of King Saud University - Computer and Information Sciences, 34 (2022) 6453–6465. https://doi.org/10.1016/j.jksuci.2022.01.015.
[30]. Y. Li *et al.*, Quantum random number generator using a cloud superconducting quantum computer based on source-independent protocol, Scientific Reports, 11 (2021) 23873. https://doi.org/10.1038/s41598-021-03286-9.