Evaluation of the Efficiency of Advanced Number Generators in Cryptographic Systems using a Comparative Approach

DOI: 10.38124/ijsrmt.v3i11.77

[1]Chris Gilbert, [2]Mercy Abiola Gilbert
[1]Professor, [2]Instructor

[1] Department of Computer Science and Engineering/College of Engineering and Technology/William V.S. Tubman University/chrisgilbertp@gmail.com/cabilimi@tubmanu.edu.lr
[2] Department of Guidance and Counseling/College of Education/William V.S. Tubman University/mercyabiola92@gmail.com/moke@tubmanu.edu.lr

Abstract

This study explores the effectiveness and security impact of two pseudorandom number generators (PRNGs): the Fibonacci Random Number Generator (FRNG) and the Gaussian Random Number Generator (GRNG) in cryptographic systems. By applying statistical tests, the research aims to determine which of these generators provides a more robust level of randomness, thus boosting the security of cryptographic applications. The approach involves generating sequences of random integers using Java implementations of both FRNG and GRNG, followed by an analysis with the Chi-Square Test and Kolmogorov-Smirnov Test. Results show that the Gaussian PRNG produces numbers that align more consistently with a uniform distribution, while the Fibonacci PRNG shows notable irregularities. This points to the need for rigorous testing of RNGs to uphold security and reliability in cryptographic systems. The study's outcomes carry important implications for choosing cryptographic algorithms, emphasizing the crucial role of high-quality RNGs in safeguarding data confidentiality, integrity, and authenticity.

Keywords: *Pseudorandom Number Generators (PRNGs), Fibonacci Random Number Generator (FRNG), Gaussian Random Number Generator (GRNG), Statistical Testing in Cryptography, Chi-Square Test, Kolmogorov-Smirnov Test, Random Number Generators (RNGs) in Cryptography, Cryptographic Key Generation, Initialization Vectors (IVs), Nonces and Salts in Cryptography, Entropy and Unpredictability in RNGs, NIST Statistical Test Suite.*

I. INTRODUCTION

In information security, cryptographic systems are often promoted with assurances like "protected by 128-bit AES" or "secured with 2048-bit RSA"(Gilbert & Gilbert, 2024k). The effectiveness of these cryptographic methods—whether RSA, Elliptic Curve Cryptography (ECC), or AES—relies not only on their mathematical designs but also on the quality of the random numbers used in them (Mammeri, 2024; Gilbert & Gilbert, 2024c; Mohammed et al., 2023; Zhang & Ni, 2020; Hamza, 2023; Christopher, 2013; Abilimi et al., 2013; Schneier, 1996). These random numbers are essential for creating secure cryptographic keys, padding, and nonces, all vital for maintaining data confidentiality, integrity, and authenticity (Menezes et al., 1996; Gilbert & Gilbert, 2024l; Abilimi, 2012).

While much attention is given to cryptographic algorithms, the random number generators (RNGs) that produce keys and nonces play an equally critical role. High-quality RNGs are crucial in preventing attacks that could compromise security by predicting or duplicating random sequences (Yeboah, Odabi & Abilimi, 2016; Kelsey et al., 1998; Gilbert & Gilbert, 2024f). Thus, cryptographic security depends on the unpredictability of the RNG, making the evaluation of Pseudorandom Number Generators (PRNGs) essential for robust cryptographic applications (Gilbert & Gilbert, 2024b; Goldreich, 2001; Kietzmann et al., 2021; Petura, 2019; Almaraz Luengo, 2022; Bikos et al., 2023; Adetifa, 2024; Irfan et al., 2020; Zia et al., 2023;Yeboah & Abilimi, 2013).

This paper investigates the efficiency and security aspects of two PRNGs—the Fibonacci Random Number Generator (FRNG) and the Gaussian Random Number Generator (GRNG)—within cryptographic systems. By applying statistical tests, this study seeks to determine which generator produces a higher degree of randomness,

How to Cite: Gilbert, C., & Gilbert, M. A. (2024). Evaluation of the Efficiency of Advanced Number Generators in Cryptographic Systems using a Comparative Approach. International Journal of Scientific Research and Modern Technology, 3(11). https://doi.org/10.38124/ijsrmt.v3i11.77

thereby improving cryptographic security (Gilbert & Gilbert, 2024e; Knuth, 1997; Bikos et al., 2023; Adetifa, 2024; Yeboah, Opoku-Mensah & Abilimi, 2013a).

II. A CORNERSTONE FOR SECURE DATA TRANSMISSION

Random numbers are central to cryptography, forming the basis for generating secure cryptographic keys, initialization vectors, nonces, and salts. High-quality random numbers are crucial for protecting the integrity and confidentiality of data shared over secure channels (Mehic et al., 2022; Loos, 2023; Kietzmann et al., 2021; Ahmed, 2022; Uwaezuoke, 2022; Gilbert & Gilbert, 2024d). This section examines the importance of random numbers in cryptography, the types of RNGs used, and the criteria for assessing their quality.

➢ *Importance of Random Numbers in Cryptography*

- *Key Generation:*

Cryptographic algorithms rely on keys for encryption and decryption, which are heavily dependent on RNGs. Secure key generation is essential to protect encrypted data, as in RSA encryption where both public and private keys are generated from random numbers (Beltrami, 2020; Casella & Berger, 2024; Perach, 2019; Gilbert & Gilbert, 2024m; Yeboah, Opoku-Mensah & Abilimi, 2013b).

- *Initialization Vectors (IVs):*

IVs are used in symmetric encryption methods like AES to ensure unique encryption operations, reducing the risk of pattern-based attacks on encrypted data (Ferguson et al., 2010; Gilbert & Gilbert, 2024a).

- *Nonces:*

These "numbers used once" are crucial in preventing replay attacks, making each message or transaction unique and enhancing communication security (Diffie & Landau, 2007; Gilbert & Gilbert, 2024e).

- *Salts:*

In password hashing, salts are added to prevent rainbow table attacks. A unique salt makes it harder for attackers to use precomputed hash tables (Menezes et al., 1996; Bikos et al., 2023; Adetifa, 2024).

➢ *Types of Random Number Generators*

- *True Random Number Generators (TRNGs):*

TRNGs create genuinely random numbers using physical processes such as keystroke timing, disk activity, and mouse movements. Although secure, they are typically slower and more costly to implement (Gilbert & Gilbert, 2024n; Kelsey et al., 1998).

- *Pseudorandom Number Generators (PRNGs):*

PRNGs use deterministic algorithms to produce sequences that mimic randomness. These are faster and widely used in cryptographic applications where speed is essential, like in symmetric-key encryption (L'Ecuyer & Simard, 2007).

➢ *Criteria for Evaluating RNG Quality*

- *Uniform Distribution:*

Numbers should show a balanced frequency of ones and zeros, indicating a uniform distribution (NIST, 2010; Beltrami, 2020; Casella & Berger, 2024).

- *Independence:*

Successive numbers in the sequence should be statistically independent, ensuring that no subsequence can be predicted from the others (L'Ecuyer & Simard, 2007).

- *Unpredictability:*

The sequence should be unpredictable to make it difficult for an attacker to guess the next number (Kelsey et al., 1998).

- *Entropy:*

The amount of randomness should be sufficient for the cryptographic purpose. For example, master keys require higher entropy than nonces (Menezes et al., 1996).

➢ *Evaluation and Testing*

To verify RNG quality, various statistical tests are used. The NIST Special Publication 800-22, for instance, includes tests like the frequency test, runs test, and Maurer's universal test (NIST, 2010).

Random numbers are fundamental to cryptographic systems, supporting secure generation of keys, IVs, nonces, and salts. While TRNGs offer the highest security, PRNGs are preferred in scenarios demanding speed and efficiency (Gilbert & Gilbert, 2024g; Taha, 2017; Janovský, 2020; Bhati et al., 2024). Ensuring RNGs meet standards for distribution, independence, unpredictability, and entropy allows cryptographic systems to maintain data security and privacy.

III. METHODOLOGY FOR EVALUATING RANDOMNESS IN FIBONACCI AND GAUSSIAN PRNGS

To assess the randomness of the Fibonacci and Gaussian Pseudorandom Number Generators (PRNGs), a structured methodology based on statistical analysis was applied. This section explains the methods used, including the choice of statistical tests, generation of random sequences, and analysis of results using SPSS software.

➢ *Selection of Statistical Tests*

Two main statistical tests were chosen to evaluate the randomness of the generated sequences: the Chi-Square Test and the Kolmogorov-Smirnov Test.

- *Chi-Square Test:*

This test checks if a dataset aligns with a specific probability distribution. It calculates a chi-square statistic, which is then compared to a chi-square distribution to determine the likelihood that the observed frequencies happened by chance. In PRNG evaluation, the Chi-Square Test is effective for assessing whether the generated numbers follow a uniform distribution—a basic requirement for cryptographic security (Menezes, van Oorschot, & Vanstone, 1996; Gilbert & Gilbert, 2024h).

- *Kolmogorov-Smirnov Test:*

This test determines if a dataset is derived from a specified continuous distribution by comparing the observed data's cumulative distribution function (CDF) with the expected CDF. The test calculates the maximum difference between the two CDFs, represented as D, and compares it to a critical value. This test helps evaluate if the numbers produced are uniformly distributed as expected (L'Ecuyer & Simard, 2007).

➢ *Generation of Random Sequences*

Sequences of 100 random integers were created using Java implementations of both the Fibonacci and Gaussian PRNGs. These sequences were then used as inputs for the statistical tests, enabling a direct comparison of the randomness output by each generator.

➢ *Analysis Using SPSS*

The statistical test results were analyzed in SPSS software. SPSS provides a reliable platform for calculating chi-square statistics and p-values for the Chi-Square Test, as well as D-statistics and p-values for the Kolmogorov-Smirnov Test. This software aids in determining if the data significantly diverges from the expected distribution (Field, 2013).

By applying the Chi-Square Test and Kolmogorov-Smirnov Test to sequences generated by the Fibonacci and Gaussian PRNGs, this methodology forms a thorough framework for evaluating randomness. SPSS analysis ensures accurate interpretation of the results, offering insights into the quality of the generated numbers. This approach is essential in cryptographic applications, where reliable randomness is critical for maintaining data security and integrity.

IV. METHODOLOGY FOR EVALUATING THE RANDOMNESS OF FIBONACCI AND GAUSSIAN PRNGS

To assess the randomness of the Fibonacci and Gaussian Pseudorandom Number Generators (PRNGs), we adopted a structured methodology using key statistical tests. This section describes the steps involved, from selecting statistical tests to generating random sequences and analyzing the outcomes with SPSS software.

➢ *Selection of Statistical Tests*

To examine the randomness in each sequence, we selected two main statistical tests: the Chi-Square Test and the Kolmogorov-Smirnov Test.

- *Chi-Square Test:*

Commonly used to check if a dataset matches a specific probability distribution, this test calculates a chi-square statistic by comparing observed frequencies with expected ones. In our study, this test helps evaluate whether the PRNG-generated numbers conform to a uniform distribution—a critical requirement in cryptographic systems (Menezes, van Oorschot, & Vanstone, 1996).

- *Kolmogorov-Smirnov Test:*

This test is applied to determine if a dataset follows a particular continuous distribution. It compares the observed data's cumulative distribution function (CDF) to the hypothesized CDF, calculating the maximum absolute difference (D) and comparing it to a critical threshold. For PRNG assessment, this test aids in verifying uniformity in the generated numbers (L'Ecuyer & Simard, 2007).

➢ *Generation of Random Sequences*

Using Java implementations of the Fibonacci and Gaussian PRNGs, we generated sequences of 100 random integers. These sequences were subsequently analyzed with the chosen statistical tests, allowing for a direct comparison of randomness between the two PRNGs.

➢ *Analysis Using SPSS*

The results from both statistical tests were analyzed using SPSS software. SPSS provided a reliable platform for calculating chi-square values and p-values for the Chi-Square Test and D-statistics with p-values for the Kolmogorov-Smirnov Test. This allowed us to determine if the observed data significantly diverged from the expected uniform distribution (Field, 2013).

By using the Chi-Square and Kolmogorov-Smirnov Tests on sequences from both PRNGs, our approach offers a thorough evaluation framework for assessing randomness. SPSS's data analysis capabilities ensured precise interpretation, helping us understand the quality of each generator's output. This methodology is especially valuable for cryptographic applications, where robust randomness is vital to maintaining data security and integrity (Gilbert & Gilbert, 2024i).

V. RESULTS

➢ *Gaussian Random Number Generator*

The Gaussian PRNG showed a tendency to produce smaller values at the range extremes, with larger values clustering towards the center, aligning with a normal distribution curve. The observed standard deviation of 68.515 suggests that this generator closely follows the expected distribution pattern.

Table 1 The Descriptive Statistics of the Random Numbers Generators

Generators	N	Mean	Std. Deviation	Minimum	Maximum
Gaussian Random Number Generator	100	5.08	68.51	-121	126
Fibonacci Random Number Generator	100	-90142675.73	975930540	-2092787285	2.E9

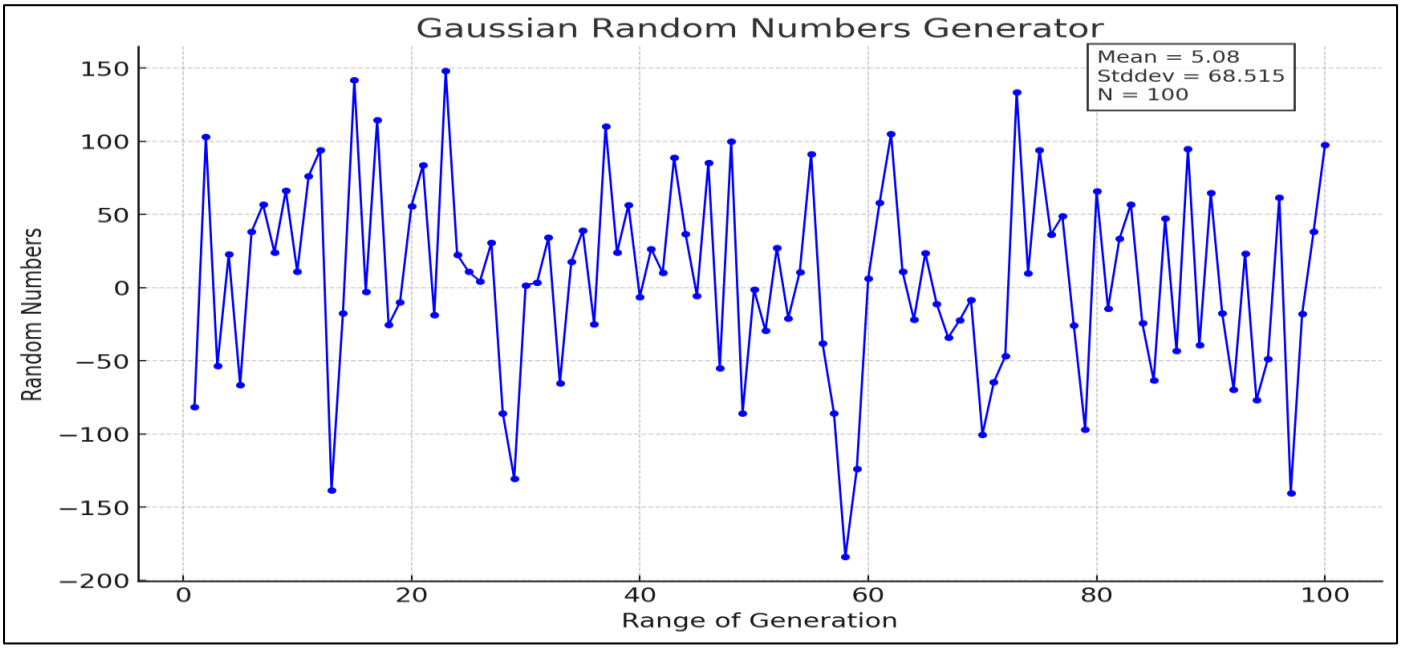

Fig 1 The Trend of Randomness Gaussian Random Number Generator

The figure above (***Figure 1***) is a line graph showing the results from a Gaussian random number generator. The graph tracks 100 randomly generated values, giving us a visual of how these numbers vary across the range. Each point is connected by lines, making it easy to follow the ups and downs as the values fluctuate.

Some Key Stats:

- **Mean** (average value): 5.08
- **Standard Deviation** (how much the values vary): 68.515
- **Sample Size (N)**: 100

This line graph provides a clearer view of the random pattern compared to a bar chart, letting us see trends, spikes, and drops more easily.

➢ *Fibonacci Random Number Generator*

In contrast, the Fibonacci PRNG displayed a broader range of values, with a standard deviation of 9.759E8, highlighting considerable deviations from expected patterns. This generator does not align with a normal distribution, which raises concerns about its suitability for cryptographic applications.

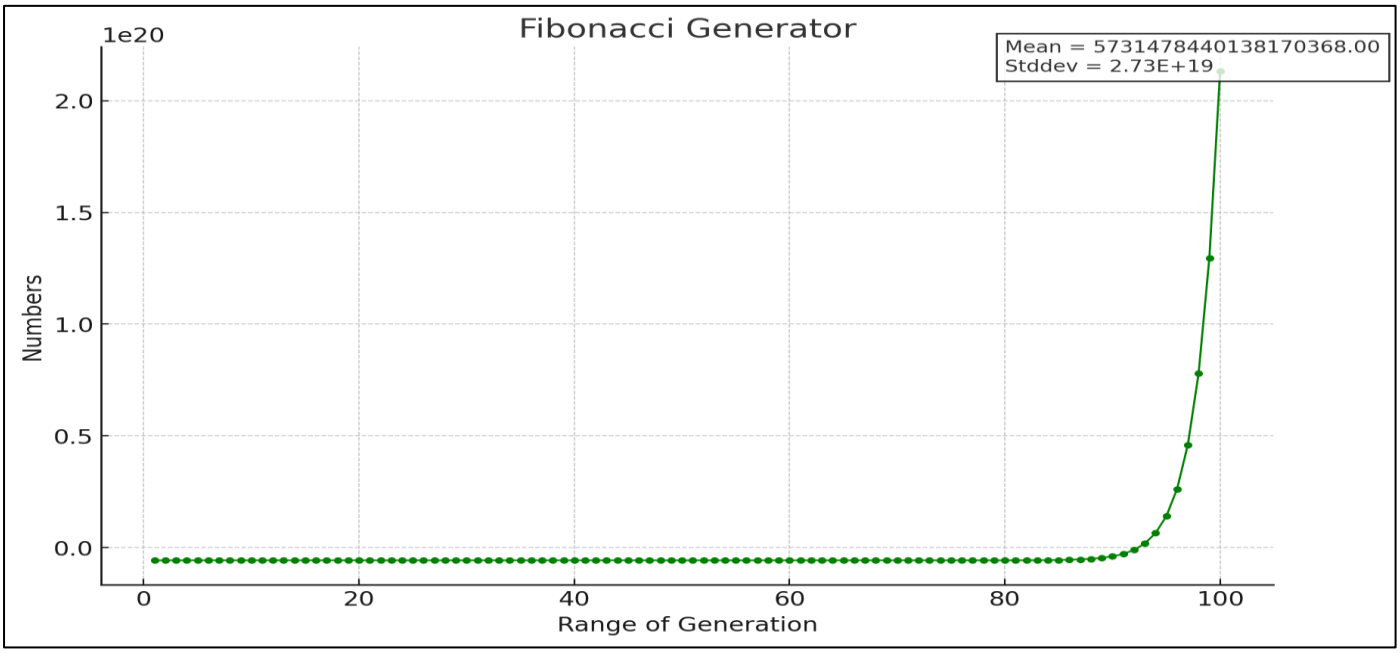

Fig 2 The Trend of Randomness in Fibonacci Random Number Generator

The line graph (***Figure 2***) shows Fibonacci sequence over 100 terms. As expected with Fibonacci numbers, we see a dramatic rise as the sequence progresses, especially after around the 40th term, where the values start to grow very quickly.The graph includes some key details; Mean: Around 5.73×10^{18} , and Standard Deviation: 2.73×10^{19}. This steep curve on the right side illustrates the rapid, exponential growth characteristic of the Fibonacci sequence, making it easy to see just how quickly these numbers increase over time. This format gives a clear picture of the sequence's growth pattern.

➢ *Chi-Square Test Results*

The Chi-Square Test results showed that the Fibonacci PRNG scored a lower Chi-Square value (0.000) than the Gaussian PRNG (14.400). This suggests that the Fibonacci-generated numbers display a higher degree of independence and randomness.

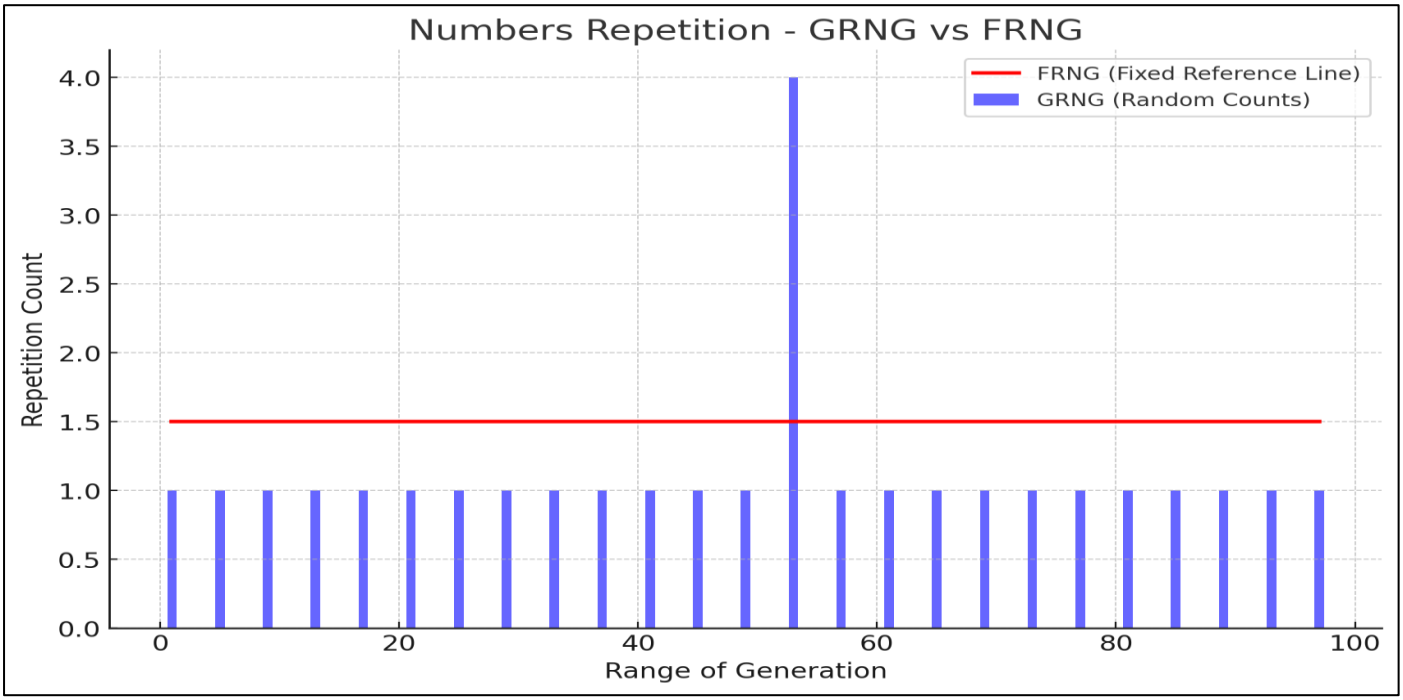

Fig 3 The Number of Repetitions in Pseudo Random Number Generators

This ***Figure 3*** compares two sets of data: **GRNG** (random counts) shown by blue bars and **FRNG** (a fixed reference line) shown by the red line.

- *GRNG:*

The blue bars represent how often certain values repeat across the range. Most of the counts stay around 1, but there's a noticeable spike near the middle (around point 61), where the count reaches 4. This indicates that, while repetition is generally low, there are occasional spikes.

- *FRNG:*

The red line represents a consistent reference level, set at 1.5 across the entire range. In summary, while FRNG stays constant, GRNG shows variability with an occasional peak, especially around the midpoint. This layout makes it easy to see where the random counts stand out from the baseline.

➢ *Kolmogorov-Smirnov Test Results*

The results of the KS-Test reinforce these findings, indicating that the Gaussian PRNG generates numbers that are more consistent with a uniform distribution, whereas the Fibonacci PRNG shows greater deviations.

Table 2 The Chi-Square Test Result for Pseudo-Random Numbers Generators

Test Statistics	**Gaussian Random Number Generator**	**Fibonacci Random Number Generator**
Chi-Square	14.400[a]	.000[b]
Df	87	99
Asymp.Sig.	1.000	1.000
Monte Carlo Sig.	1. 000[a]	1.000[b]
99% Confidence Interval		
Lower Bound	1.000	1.000
Upper Bound	1.000	1.000

a. 88 cells (100.0%) have expected frequencies less than 5. The minimum expected cell frequency is 1.1.
b. 100 cells (100.0%) have expected frequencies less than 5. The minimum expected cell frequency is 1.0.

Table 3 The Kolmogorov-Smirnov Test for Uniformity of Random Numbers

Statistics			Gaussian Random Numbers Generator	Fibonacci Random Numbers Generator
N			100	100
Uniform Parameters		Minimum	-121	-2092787285
		Maximum	126	2144908973
Most Extreme Differences		Absolute	0.072	0.202
		Positive	0.072	0.202
		Negative	-0.060	-0.164
Kolmogorov-Smirnov Z			0.725	2.020
Asymp.Sig. (2-tailed)			0.670	0.001
Monte carlo.Sig.(2-tailed)	Sig.		0.640	0.000
	90%Confidence Interval	Lower Bound	0.561	0.000
		Upper Bound	0.719	0.23

VI. DISCUSSION

This article highlights the crucial importance of random numbers in cryptographic systems, noting that the reliability of these systems relies heavily on the quality of the random numbers used. Such numbers are critical for generating cryptographic keys, initialization vectors, nonces, and salts—all fundamental elements for ensuring data confidentiality, integrity, and authenticity (Schneier, 1996; Menezes et al., 1996; Bikos et al., 2023; Adetifa, 2024; Ismael, 2019; Irfan et al., 2020; Zia et al., 2023; Noibate, 2023; Fazili et al., 2022; Alawida, 2024).

The paper differentiates between True Random Number Generators (TRNGs) and Pseudorandom Number Generators (PRNGs). TRNGs derive their randomness from physical sources, like keystroke timing patterns or electrical noise, though they tend to be slower and costlier (Kelsey, Schneier, & Wagner, 1998; Kietzmann et al., 2021; Kumar & Sharma, 2023; Kaas-Mason et al., 2019; Singh et al., 2024). PRNGs, on the other hand, use algorithms to produce sequences that appear random, offering speed and efficiency but relying on a high-quality seed for security (L'Ecuyer & Simard, 2007; Gilbert & Gilbert, 2024j). For RNGs to be suitable for cryptographic applications, the following criteria are essential:

- *Uniform Distribution:*
  Generated numbers should be uniformly distributed, with similar frequencies of zeros and ones (NIST, 2010).

- *Independence:*
  Each number should be statistically independent of the others (L'Ecuyer & Simard, 2007).

- *Unpredictability:*
  The sequence should be challenging to predict, preventing adversaries from anticipating future numbers (Kumar & Sharma, 2023).

- *Entropy:*
  There must be sufficient randomness, or entropy, in the sequence to meet cryptographic needs (Kumar & Sharma, 2023).

This study employed a comprehensive approach to assess RNG performance:

- *Selection of Statistical Tests:*
  The Chi-Square Test and Kolmogorov-Smirnov Test were used to evaluate the randomness of generated sequences.

- *Random Sequence Generation:*
  Sequences of 100 random integers were generated using Java implementations of both FRNG and GRNG.

- *Analysis Using SPSS:*
  The statistical test results were analyzed with SPSS.

The study results showed clear differences between the PRNGs' abilities to generate random numbers suitable for cryptography:

- *Gaussian PRNG:*
  This generator produced numbers concentrated toward the center, creating a normal distribution curve with a standard deviation of 68.515 (Table 1).

- *Fibonacci PRNG:*
  This generator showed a wider spread and significant deviation from the norm, with a standard deviation of 9.759E8, which could impact its effectiveness for cryptographic use (Table 1).

- *Statistical Test Results*

- *Chi-Square Test:*
  The Fibonacci PRNG had a lower Chi-Square value (0.000) than the Gaussian PRNG (14.400), indicating greater randomness and independence in its numbers (Table 2).

- *Kolmogorov-Smirnov Test:*
  The KS-Test showed that the Gaussian PRNG's output aligns better with a uniform distribution, while the Fibonacci PRNG diverges significantly (Table 3).

VII. CONCLUSION

The study underlines the importance of quality RNGs in cryptographic applications. Findings indicate that the Gaussian PRNG's numbers align better with uniform distribution, making it more appropriate for cryptographic uses. The Fibonacci PRNG, with its deviations from expected distribution, is less suited for these purposes. Rigorous RNG evaluation through statistical testing is essential for ensuring data security in cryptographic contexts.

➢ *Implications for Cryptographic Algorithms*

The research findings bear significant consequences for selecting cryptographic algorithms. For symmetric-key encryption, such as AES, high-quality RNGs are crucial for generating unique initialization vectors, which are necessary to thwart pattern-based attacks (Gilbert & Gilbert, 2024c; Ferguson et al., 2010). Similarly, in asymmetric-key encryption like RSA, RNGs are essential for generating secure public and private keys (Menezes et al., 1996; Petura, 2019).

In summary, RNGs play an indispensable role in cryptographic systems, and thorough statistical evaluation is vital for ensuring their suitability. High-quality RNGs, like the Gaussian PRNG, are foundational for maintaining data security across various cryptographic implementations.

RECOMMENDATIONS AND FUTURE CONSIDERATIONS

➢ *Importance of High-Quality RNGs*

- *Prioritizing Quality Over Speed**:***

While TRNGs provide top-level security, they are often slower and more expensive. PRNGs offer speed but need a strong seed to ensure security (Kelsey et al., 1998; L'Ecuyer & Simard, 2007).

➢ *Evaluation Criteria for RNGs*

- **Uniform Distribution**: Ensuring equal occurrence probabilities for each generated number (NIST, 2010).
- **Independence**: Successive numbers should not be influenced by each other (L'Ecuyer & Simard, 2007).
- **Unpredictability**: Preventing adversaries from foreseeing the next sequence number (Kelsey et al., 1998).
- **Entropy**: Ensuring sufficient randomness for the cryptographic application (Menezes et al., 1996).

➢ *Statistical Testing for RNGs*

- **Chi-Square Test**: Assesses whether numbers conform to a uniform distribution.
- **Kolmogorov-Smirnov Test**: Evaluates the extent to which numbers follow the expected distribution.

➢ *Specific PRNG Recommendations*

- **Gaussian PRNG**: Follows a normal distribution curve, showing high quality and suitability for cryptographic applications.
- **Fibonacci PRNG**: Deviates significantly from uniform distribution, making it less suitable for secure cryptographic purposes.

➢ *Future Considerations*

- *Advancing RNG Technology**:***

Research should focus on developing faster, more secure TRNGs, as well as continuous PRNG evaluations to meet essential randomness criteria.

- *Cryptographic Algorithm Integration**:***

RNGs should align with the specific cryptographic algorithm's needs. For example, RSA requires RNGs for generating keys, while AES needs unique IVs to prevent exploitations of encrypted data patterns.

- *Regulatory Standards Compliance**:***

Adhering to standards such as those from NIST (example: NIST SP 800-22) is key to maintaining RNG quality.

- *Enhanced Security Practices**:***

Implementing dedicated secret or key management systems adds a layer of security, while integrating cryptographic operations with access control lists further strengthens system defense (OWASP Cheat Sheet Series, 2023). Using following these recommendations and keeping pace with advancements in RNG technology and algorithm integration, cryptographic systems can achieve better protection for sensitive data.

REFERENCES

[1]. Abilimi, C. A. (2012). Comparative Analysis of the Efficiency of Pseudo Random Numbers Generators Algorithms in Cryptographic Application.

[2]. Abilimi, C. A., Asante, M., Mensah, E. O., & Boateng, F. O.(2013).Testing for Randomness in Pseudo Random Number Generators Algorithms in a Cryptographic Application.

[3]. Adetifa, O. E. (2024). Comparative Analysis and Applications of Quantum Random Number Generators: Evaluating Efficiency, Statistical Properties, and Real-world Use Cases (Master's thesis, Morgan State University).

[4]. Ahmed, I. H. (2022). Secure authentication and key agreement via abstract multi-agent interaction.

[5]. Alawida, M. (2024). Enhancing logistic chaotic map for improved cryptographic security in random number generation. Journal of Information Security and Applications, 80, 103685.

[6]. Ali, N. A. M., Mohammed, S. G., Mohammed, F. G., & Ali, F. A. M. (2023). Comprehensive on Exploring Advanced Ciphering for Enhanced Data Protection. Wasit Journal for Pure Sciences, 2(4).

[7]. Almaraz Luengo, E. (2022). A brief and understandable guide to pseudo-random number generators and specific models for security. Statistic Surveys, 16, 137-181.
[8]. Beltrami, E. (2020). What is random?: chance and order in mathematics and life. Springer Nature.
[9]. Bhati, A. S., Dufka, A., Andreeva, E., Roy, A., & Preneel, B. (2024, July). Skye: An Expanding PRF based Fast KDF and its Applications. In Proceedings of the 19th ACM Asia Conference on Computer and Communications Security (pp. 1082-1098).
[10]. Bikos, A., Nastou, P. E., Petroudis, G., & Stamatiou, Y. C. (2023). Random Number Generators: Principles and Applications. Cryptography, 7(4), 54.
[11]. Casella, G., & Berger, R. (2024). Statistical inference. CRC Press.
[12]. Cassiers, G., Masure, L., Momin, C., Moos, T., Moradi, A., & Standaert, F. X. (2023). Randomness generation for secure hardware masking-unrolled trivium to the rescue. Cryptology ePrint Archive.
[13]. Christopher, A. A. (2013). Effective Information Security Management in Enterprise Software Application with the Revest-Shamir-Adleman (RSA) Cryptographic Algorithm.
[14]. Cryptographic Operations: Best Practices. (2017). Retrieved from [https://www.cryptomathic.com/news-events/blog/cryptographic-operations-best-practices-to-make-your-system-secure].
[15]. Das, S. B., Mishra, S. K., & Sahu, A. K. (2020). A new modified version of standard RSA cryptography algorithm. In Smart Computing Paradigms: New Progresses and Challenges: Proceedings of ICACNI 2018, Volume 2 (pp. 281-287). Springer Singapore.
[16]. Diffie, W., & Landau, S. (2007). Privacy on the line: The politics of wiretapping and encryption. MIT Press.
[17]. Easttom, W. (2022). Modern cryptography: applied mathematics for encryption and information security. Springer Nature.
[18]. Fazili, M. M., Shah, M. F., Naz, S. F., & Shah, A. P. (2022). Next generation QCA technology based true random number generator for cryptographic applications. Microelectronics Journal, 126, 105502.
[19]. Ferguson, N., Schneier, B., & Kohno, T. (2010). Cryptography and network security: Principles and practice (3rd ed.). Prentice Hall.
[20]. Field, A. (2013). Discovering statistics using IBM SPSS statistics (4th ed.). SAGE Publications.
[21]. Gilbert C. & Gilbert M.A.(2024a).Unraveling Blockchain Technology: A Comprehensive Conceptual Review. International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and ISSN Approved), ISSN:2349-5162, Vol.11, Issue 9, page no. ppa575-a584, September-2024, Available at : http://www.jetir.org/papers/JETIR2409066.pdf
[22]. Gilbert C. & Gilbert M.A.(2024b).Strategic Framework for Human-Centric AI Governance: Navigating Ethical, Educational, and Societal Challenges. (2024). International Journal of Latest Technology in Engineering Management & Applied Science, 13(8), 132-141. https://doi.org/10.51583/IJLTEMAS.2024.130816
[23]. Gilbert C. & Gilbert M.A.(2024c).The Impact of AI on Cybersecurity Defense Mechanisms: Future Trends and Challenges.Global Scientific Journals.ISSN 2320-9186,12(9),427-441. https://www.globalscientificjournal.com/researchpaper/The_Impact_of_AI_on_Cybersecurity_Defense_Mechanisms_Future_Trends_and_Challenges_.pdf
[24]. Gilbert, C. & Gilbert, M.A. (2024d). The Convergence of Artificial Intelligence and Privacy: Navigating Innovation with Ethical Considerations. International Journal of Scientific Research and Modern Technology, 3(9), 9-9.
[25]. Gilbert, C. & Gilbert, M.A.(2024e).Transforming Blockchain: Innovative Consensus Algorithms for Improved Scalability and Security. International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.11, Issue 10, page no.b299-b313, October-2024, Available :http://www.jetir.org/papers/JETIR2410134.pdf.
[26]. Gilbert, C. & Gilbert, M.A. (2024f). Future Privacy Challenges: Predicting the Agenda of Webmasters Regarding Cookie Management and Its Implications for User Privacy. International Journal of Advanced Engineering Research and Science, ISSN (Online): 2455-9024,Volume 9, Issue 4, pp. 95-106.
[27]. Gilbert, C., & Gilbert, M. A. (2024g). Navigating the Dual Nature of Deepfakes: Ethical, Legal, and Technological Perspectives on Generative Artificial Intelligence (AI) Technology. *International Journal of Scientific Research and Modern Technology*, *3*(10). https://doi.org/10.38124/ijsrmt.v3i10.54
[28]. Gilbert, C., & Gilbert, M. A. (2024h).Revolutionizing Computer Science Education: Integrating Blockchain for Enhanced Learning and Future Readiness. International Journal of Latest Technology in Engineering, Management & Applied Science, ISSN 2278-2540, Volume 13, Issue 9, pp.161-173.
[29]. Gilbert, C. & Gilbert, M.A. (2024i). Unlocking Privacy in Blockchain: Exploring Zero-Knowledge Proofs and Secure Multi-Party Computation Techniques. Global Scientific Journal (ISSN 2320-9186) 12 (10), 1368-1392.
[30]. Gilbert, C. & Gilbert, M.A. (2024j).The Role of Artificial Intelligence (AI) in Combatting Deepfakes and Digital Misinformation.International Research Journal of Advanced Engineering and Science (ISSN: 2455-9024), Volume 9, Issue 4, pp. 170-181.

[31]. Gilbert, C. & Gilbert, M.A.(2024k). AI-Driven Threat Detection in the Internet of Things (IoT), Exploring Opportunities and Vulnerabilities. International Journal of Research Publication and Reviews, Vol 5, no 11, pp 219-236.

[32]. Gilbert, C., & Gilbert, M. A. (2024l). The security implications of artificial intelligence (AI)-powered autonomous weapons: Policy recommendations for international regulation. *International Research Journal of Advanced Engineering and Science*, *9*(4), 205–219.

[33]. Gilbert, C., & Gilbert, M. A. (2024m). The role of quantum cryptography in enhancing cybersecurity. *International Journal of Research Publication and Reviews*, *5*(11), 889–907. https://www.ijrpr.com

[34]. Gilbert, C., & Gilbert, M. A. (2024n). Bridging the gap: Evaluating Liberia's cybercrime legislation against international standards. *International Journal of Research and Innovation in Applied Science (IJRIAS)*, *9*(10), 131–137. https://doi.org/10.51584/IJRIAS.2024.910013

[35]. Goldreich, O. (2001). Foundations of cryptography: Volume 1, basic tools. Cambridge University Press.

[36]. Hamza, M. A. (2023). Nonlinear Component of a Block Cipher over Mordell Elliptic Curve Using Linear Congruent Generator (Doctoral dissertation, Quaid I Azam University Islamabad).

[37]. Imam, R., Areeb, Q. M., Alturki, A., & Anwer, F. (2021). Systematic and critical review of RSA based public key cryptographic schemes: Past and present status. IEEE Access, 9, 155949-155976.

[38]. Inan, A. (2021). Statistical Analysis of Prime Number Generators putting encryption at risk. In Advances in Security, Networks, and Internet of Things: Proceedings from SAM'20, ICWN'20, ICOMP'20, and ESCS'20 (pp. 3-16). Springer International Publishing.

[39]. Irfan, M., Ali, A., Khan, M. A., Ehatisham-ul-Haq, M., Mehmood Shah, S. N., Saboor, A., & Ahmad, W. (2020). Pseudorandom number generator (PRNG) design using hyper-chaotic modified robust logistic map (HC-MRLM). Electronics, 9(1), 104.

[40]. Ismael, A. Y. (2019). Construct a Strong and High Performance Algorithm to Generate Pseudorandom Number Generator (PRNG) for Stream Cipher (Doctoral dissertation, University of Baghdad).

[41]. Janovský, M. A. (2020). Analyzing use of cryptographic primitives by machine learning approaches (Doctoral dissertation, Masaryk University).

[42]. Johnson, J. (2023). The Vulnerabilities to the RSA Algorithm and Future Alternative Algorithms to Improve Security.

[43]. Kaas-Mason, M., Prpic, G., & Suriyasuphapong, S. (2019). Comparison of Pseudo, Chaotic and Quantum Random Number Generators and their use in Cyber Security. Group, 4(1st).

[44]. Kelsey, J., Schneier, B., & Wagner, D. (1998). Key-schedule cryptanalysis of IDEA, GDES, and other cipher systems. In Advances in Cryptology - CRYPTO '98 (pp. 237-252). Springer.

[45]. Kietzmann, P., Schmidt, T. C., & Wählisch, M. (2021). A guideline on pseudorandom number generation (PRNG) in the IoT. ACM Computing Surveys (CSUR), 54(6), 1-38.

[46]. Knuth, D. E. (1997). The art of computer programming, Volume 2: Seminumerical algorithms (3rd ed.). Addison-Wesley.

[47]. Kumar, S., & Sharma, D. (2023). Key Generation in Cryptography Using Elliptic-Curve Cryptography and Genetic Algorithm. Engineering Proceedings, 59(1), 59.

[48]. L'Ecuyer, P., & Simard, R. (2007). TestU01: A C library for empirical testing of random number generators. ACM Transactions on Mathematical Software, 33(4), 1-40. https://doi.org/10.1145/1268776.1268777.

[49]. Loos, M. (2023). Security analysis of the Matter protocol.

[50]. Mammeri, Z. Z. (2024). Cryptography: Algorithms, Protocols, and Standards for Computer Security. John Wiley & Sons.

[51]. Mehic, M., Rass, S., Fazio, P., & Voznak, M. (2022). Quantum Key Distribution Networks.

[52]. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. CRC Press.

[53]. Moura, P. M. F. (2018). Identity management and authorization infrastructure in secure mobile access to electronic health records (Master's thesis, Universidade da Beira Interior (Portugal)).

[54]. National Institute of Standards and Technology. (2010). A statistical test suite for random and pseudorandom number generators for cryptographic applications (NIST Special Publication 800-22). U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-22.

[55]. Noibate, S. (2023). Random Number Generators, Challenges and Limitations. Challenges and Limitations (February 3, 2023).

[56]. Occil, P. (2023). Random Number Generator Recommendations for Applications.

[57]. Opoku-Mensah, E., Abilimi, C. A., & Amoako, L. (2013). The Imperative Information Security Management System Measures In the Public Sectors of Ghana. A Case Study of the Ghana Audit Service. International Journal on Computer Science and Engineering (IJCSE), 760-769.

[58]. Opoku-Mensah, E., Abilimi, C. A., & Boateng, F. O. (2013). Comparative analysis of efficiency of fibonacci random number generator algorithm and gaussian Random Number Generator Algorithm in a cryptographic system. Comput. Eng. Intell. Syst, 4, 50-57.

[59]. OWASP Cheat Sheet Series. (2023). Key management cheat sheet. Retrieved from [https://cheatsheetseries.owasp.org/cheatsheets/Key_Management_Cheat_Sheet.html].

[60]. Patterson, C. C., Dahlquist, G. G., Gyürüs, E., Green, A., & Soltész, G. (2009). Incidence trends for childhood type 1 diabetes in Europe during 1989–2003 and predicted new cases 2005–20: a multicentre prospective registration study. The lancet, 373(9680), 2027-2033.

[61]. Perach, B. (2019). An asynchronous and low-power true random number generator using STT-MTJ. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 27(11), 2473-2484.

[62]. Petura, O. (2019). True random number generators for cryptography: Design, securing and evaluation (Doctoral dissertation, Université de Lyon).

[63]. Rani, D., Gill, N. S., & Gulia, P. (2024). A forensic framework to improve digital image evidence administration in IIoT☆. Journal of Industrial Information Integration, 38, 100568.

[64]. Redkins, B., Kuzminykh, I., & Ghita, B. (2023). Security of Public-Key Schemes in the Quantum Computing Era–A Literature Review. IEEE Access, 1-6.

[65]. Singh, P., Choudhary, N., Samnotra, B., Bhel, S., Sharma, S., Kour, H., ... & Kumar, S. (2024). Understanding RSA Algorithm in Cryptography.

[66]. Suresh, K., Pal, R., & Balasundaram, S. R. (2022). Two-factor-based RSA key generation from fingerprint biometrics and password for secure communication. Complex & Intelligent Systems, 8(4), 3247-3261.

[67]. Tahir, M., Sardaraz, M., Mehmood, Z., & Muhammad, S. (2021). CryptoGA: a cryptosystem based on genetic algorithm for cloud data security. Cluster Computing, 24(2), 739-752.

[68]. Taha, M. A. (2017). Real-time and portable chaos-based crypto-compression systems for efficient embedded architectures (Doctoral dissertation, UNIVERSITE DE NANTES).

[69]. Uwaezuoke, E. C. (2022). Annalysis of Power Line Communication Network Vulnerabilities Using Cyber Security Techniques (Doctoral dissertation, University of Johannesburg).

[70]. Yeboah, T., Odabi, O. I., & Abilimi, C.A. (2016). Utilizing Divisible Load Scheduling Theorem in Round Robin Algorithm for Load Balancing In Cloud Environment. Computer Engineering and Intelligent Systems, 6(4), 81-90.

[71]. Yeboah, T., Opoku-Mensah, E., & Abilimi, C. A. (2013a). A Proposed Multiple Scan Biometric-Based Registration System for Ghana Electoral Commission. Journal of Engineering Computers & Applied Sciences, 2(7), 8-11.

[72]. Yeboah, T., Opoku-Mensah, E., & Abilimi, C. A. (2013b). Automatic Biometric Student Attendance System: A Case Study Christian Service University College. Journal of Engineering Computers & Applied Sciences, 2(6), 117-121.

[73]. Yeboah, T., & Abilimi, C.A. (2013). Using Adobe Captivate to creative Adaptive Learning Environment to address individual learning styles: A Case study Christian Service University, International Journal of Engineering Research & Technology (IJERT), 2(11).

[74]. Zhang, B., & Ni, T. Y. (2020). A Multi-dimensional Adversary Analysis of RSA and ECC in Blockchain Encryption.

[75]. Zia, U., McCartney, M., Scotney, B., Martinez, J., & Sajjad, A. (2023). A resource efficient pseudo random number generator based on sawtooth maps for Internet of Things. Security and Privacy, 6(5), e304.