Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 8s (2025)

Statistical Analysis of Hybrid AES using Cryptographically Secure Pseudorandom Bit Generator

Md. Hasanujjaman[1], J K M Sadique Uz Zaman[2*]

[1]Sheikhpara ARM Polytechnic, Murshidabad, West Bengal, India

[2]Dept. of Computer Science and Technology, University of North Bengal, India,jkmsadique@gmail.com

* : Corresponding Author

Article History:

Received: *05-11-2024*

Revised: *24-04-2025*

Accepted: *07-08-2025*

Abstract:

The perpetual evolution of cryptanalytic techniques necessitates continuous reinforcement of established cryptographic standards. The Advanced Encryption Standard (AES) is a cornerstone of modern information security, yet its theoretical susceptibility to attacks, particularly when implemented with predictable or non-random components, remains a subject of academic enquiry. This paper investigates a hybrid cryptographic approach to enhance the statistical randomness of AES-generated ciphertext. The primary objective is to evaluate whether integrating cryptographically secure pseudorandom number generator (CSPRNG) at the pre-encryption stage can produce ciphertext with superior statistical properties. We encrypt a large plaintext, over 150,000 characters, using three distinct methodologies: original AES, AES hybridized with GF7 PRNG (AESGF7) and AES hybridized with the Blum-Blum-Shub (BBS) generator (AESBBS). Each method is executed with 300 unique cryptographic keys to generate a robust sample set of ciphertext. The statistical quality of the resulting binary sequences is rigorously assessed using the complete 15-test suite from the National Institute of Standards and Technology (NIST) Statistical Test Suite (STS). The findings, based on analyses of pass proportions and p-value distributions, reveal a complex relationship between hybridization and randomness. While both hybrid models generally improve the passing rates over original AES, the AESBBS variant introduces significant non-uniformity in its p-value distributions for several tests, a subtle statistical flaw. In contrast, the AESGF7 model not only enhances passing rates but also maintains uniform p-value distribution across all tests. This research concludes that the GF7-based hybrid approach offers a more balanced and statistically sound improvement, effectively strengthening resilience against statistical attacks without introducing new, undesirable artifacts.

Keywords: Advanced Encryption Standard (AES), Blum-Blum-Shub (BBS), Ciphertext Analysis, Cryptographically Secure Pseudorandom Number Generator (CSPRNG), Galois Field, NIST Statistical Test, Randomness.

1. Introduction

In the domain of information security, the concept of randomness is not only a theoretical construct but a fundamental prerequisite for cryptographic strength. True randomness is unpredictable, non-repeatable and uniformly distributed that is the bedrock of secure key generation, Initialization Vector (IV) and other cryptographic primitives [1]. The security guarantees of even the most robust cryptographic algorithms can be undermined if the underlying random components are flawed or predictable. Any statistical deviation within a cryptographic output, such as ciphertext, can potentially be exploited by an adversary to mount attacks – ranging from distinguishing attacks to full key recovery [2].

The Advanced Encryption Standard (AES), formalized in FIPS PUB 197 by the National Institute of Standards and Technology (NIST) in 2001, is a well known global standard for symmetric key

encryption [3]. Its security relies on the principles of confusion and diffusion, systematically applied over multiple rounds to transform plaintext into seemingly random ciphertext. While the AES algorithm itself is considered secure against known practical cryptanalytic attacks, its operational security in real-world implementations is contingent upon the randomness of its inputs, particularly the cryptographic key and the initialization vector (IV) used in chaining modes like Cipher Block Chaining (CBC) [4]. If an IV is reused or generated from a predictable source, it can leak information about the plaintext, nullifying the confidentiality a block cipher aims to provide.

Improving the statistical quality of the ciphertext produced by AES is therefore a critical area of research. A ciphertext that is computationally indistinguishable from a truly random sequence presents the highest possible barrier to statistical cryptanalysis. This research explores a method to enhance the randomness of AES output by integrating a cryptographically secure pseudorandom number generator (CSPRNG) into the encryption pipeline. The core hypothesis is that by preprocessing the plaintext with a high quality random bit-stream from a CSPRNG and inputting it to the original AES will possess superior statistical properties, which will, in turn, be reflected in the final ciphertext. To investigate this, two hybrid variants of AES is proposed. The first, which is termed as **AESGF7**, incorporates GF7 pseudorandom number generator based on irreducible polynomial over a Galois Field of 7 elements GF(7^3) [5]. The second variant, which is termed as **AESBBS**, utilizes the Blum-Blum-Shub (BBS) generator, a CSPRNG renowned for its strong security proof, which reduces its cryptographic security to the computational difficulty of the integer factorization problem [6].

To empirically validate the effectiveness of these hybrid models, a rigorous and standardized evaluation framework is essential. The NIST Statistical Test Suite (STS), specified in NIST Special Publication 800-22, provides such a framework [7]. It comprises 15 distinct statistical tests designed to detect non-randomness in binary sequences and is reviewed thoroughly in 2012 [8]. These tests examine various characteristics, such as the proportion of ones and zeros (Frequency test), the distribution of runs of identical bits (Runs test), the periodicity of the sequence (Discrete Fourier Transform test), and the uniformity of overlapping block patterns (Approximate Entropy test). A sequence that passes this comprehensive suite of tests is considered to have the statistical characteristics of a truly random sequence.

The motivation for this research is twofold. First, it aims to provide a quantitative analysis of the extent to which CSPRNG integration can bolster the statistical quality of AES ciphertext. Second, is comparing a computationally efficient generator (GF7) with a provably secure but more computationally intensive one (BBS). This work seeks to provide insights into the trade-offs between performance and cryptographic strength in hybrid designs. The successful demonstration of enhanced randomness would have significant practical implications for applications requiring the highest levels of data confidentiality, such as securing financial transactions, protecting state secrets, and ensuring privacy in IoT ecosystems.

In Section 2, review of relevant literature is provided. Brief discussion on experimental setup, cryptographic algorithms and NIST STS are given in Section 3. Results and analysis of statistical tests are presented in Section 4. Discussion on AESBBS, AESGF7 and others are given in Section 5. Practical implications are discussed in Section 6 and finally conclusion is given in Section 7.

2. Literature Review

The pursuit of enhancing the security and performance of the Advanced Encryption Standard (AES) has been a vibrant area of cryptographic research since its inception. Concurrently, the development and analysis pseudorandom number generators have formed a parallel and often intersecting field of study. This review synthesizes key literature focusing on AES performance analysis, the integration of PRNG with block ciphers and the use of statistical test suites for randomness evaluation.

2.1. AES Performance and Security Analysis

The security of AES has been scrutinized extensively. While no practical attack has broken the full AES algorithm, research continues to explore its theoretical limits. Ferguson, Schneier and Kohno provided a comprehensive overview of AES's design and security margins, concluding that its algebraic structure, while elegant, could be a potential source of future vulnerabilities [9]. Daemen and Rijmen, the designers of AES, have also continued to analyze its properties, particularly against algebraic and related-key attacks [3], [10].

Performance analysis has largely focused on optimizing AES for different platforms. In 2012, A. Hamalainen et al. explored efficient implementations of AES on FPGA, demonstrating significant throughput gains through pipelining and parallel architectures [11]. More recently, research has shifted towards securing AES against side-channel attacks. In 2018, Moradi et al. presented a comprehensive survey on countermeasures against power analysis and electromagnetic emanation attacks, highlighting the importance of masking and shuffling techniques to obscure the correlation between data processing and physical measurements [12]. A study by Ozturk et al. in 2015, evaluated the performance of various AES implementations on constrained IoT devices, emphasizing the trade-offs between security, energy consumption and latency [13]. These studies underscore the ongoing effort to harden AES implementations, but they primarily focus on physical or protocol level vulnerabilities rather than the statistical properties of the ciphertext itself.

2.2. Integration of PRNG and Chaotic Systems with Block Ciphers

The idea of hybridizing block ciphers with external sources of randomness is not new. A significant body of work has explored the use of chaotic maps to introduce dynamic and unpredictable behaviour into encryption algorithms. In 2014, Wang, Luan and Liu proposed a chaotic key generator to dynamically produce AES round keys, reporting improved resistance against differential cryptanalysis [14]. Similarly, Hua, Zhou and Huang in 2019 integrated a 2D logistic-adjusted-sinc map into AES to modify the S-box dynamically, which demonstrated enhanced confusion properties and passed NIST randomness tests [15]. Pareek and Patidar in 2016 presented a comprehensive review of image encryption techniques using chaotic maps, noting their ability to improve statistical metrics like correlation coefficients and entropy [16].

While chaotic systems offer complexity, their security relies on the properties of the specific map used and some have been shown to possess weaknesses in 2006 – such as finite precision effects in digital implementations [17]. The use of CSPRNG offers a more structured approach with provable security properties. Al-Khafaji and Al-Saffar in 2020 proposed a system that uses the Yarrow PRNG to generate a one-time pad that is XORed with the plaintext before AES encryption, effectively turning it into a stream cipher hybrid [18]. Their results showed excellent statistical performance

when evaluated with the Diehard test suite. The work of Schindler in 2010 focusing on the analysis of the BBS generator reinforces its suitability for cryptographic applications due to its strong security reduction to the quadratic residuosity problem [19]. Similarly, research into LFSR, such as the work by Dubrova in 2013 on maximum-period LFSR over Galois Fields, provides a foundation for designing efficient and statistically sound PRNG [5], [20]. These studies validate the potential of CSPRNG to serve as robust sources of randomness for cryptographic hybridization.

2.3. Randomness Evaluation using Statistical Test Suite

Experimental validation of randomness is a critical step in cryptographic system design. The NIST Statistical Test Suite (STS) has become the industry and academic standard for this purpose [7]. Numerous studies have employed the NIST STS to evaluate the quality of cryptographic primitives. Sonmez, Akgul and Dalkiran in 2022 used the NIST STS to assess the randomness of a new PRNG based on a fractional-order chaotic system, demonstrating its suitability for cryptographic use [21]. Doganaksoy et al. in 2010 applied the suite to analyze the output of hardware-based true random number generator (TRNG), highlighting the importance of post-processing to eliminate biases [22].

Several studies have specifically applied NIST STS to AES ciphertext. In 2017, Abdo, Amin and El-Gazar evaluated the randomness of AES and DES ciphertexts, finding that AES consistently produced outputs that passed the NIST tests, whereas DES showed some statistical weaknesses [23]. A more recent study by Kumar and Singh in 2021 compared the statistical properties of AES in different modes of operation like ECB, CBC, CFB, concluding that chaining modes like CBC produce statistically superior ciphertext due to the introduction of the IV [24]. This finding is particularly relevant to the present work, as the proposed method can be conceptualized as a form of pre-encryption chaining with a highly randomizing function. Al-hadidi et al. in 2016 proposed a modification to the AES MixColumns stage and used NIST tests to validate that their changes did not degrade the statistical quality of the output [25].

2.4. Knowledge Gap and Research Contribution

The existing literature confirms the robustness of AES and validates the use of PRNG and statistical tests in cryptography. However, still there are several gaps that are aimed to address by this research:

Direct Comparative Analysis: While studies have evaluated AES ciphertext or proposed PRNG-based modifications, there is a lack of direct, rigorous comparison between original AES and hybrid AES variants using the complete NIST STS under identical, large-scale test conditions, i.e., a large plaintext and hundreds of keys.

Comparison of Different PRNG Classes: Research often focuses on integrating a single type of generator, e.g., a chaotic map or a specific CSPRNG. This study provides a comparative analysis between a efficient algebraic generator GF7 and a CSPRNG algorithm BBS, offering insights into the performance-security trade-off.

Focus on Pre-Encryption Processing: Many hybrid approaches modify the internal components of AES, e.g., the S-box or key schedule. This research treats AES as a black box and focuses on enhancing the randomness of its input, a method that is less intrusive and more easily standardized.

By systematically evaluating original AES against the proposed AESGF7 and AESBBS hybrids using the comprehensive NIST STS framework, this paper provides novel empirical evidence on the measurable benefits of CSPRNG integration for enhancing the randomness of ciphertext.

3. Methods

This section details the experimental framework, the cryptographic algorithms under investigation, the statistical evaluation suite and the data analysis procedures employed to compare the randomness of the outputs.

3.1. Experimental Setup

The experiment was designed to generate a large and statistically significant dataset for analysis. All tests were conducted in a controlled environment to ensure reproducibility.

Plaintext: A single plaintext file was used for all encryption operations. The file consisted of a concatenation of English literary texts with a total size of 1,52,400 ASCII characters or 12,19,200 bits. The use of a large non-random plaintext with known statistical properties, i.e., the statistical profile of the English language provides a challenging baseline for the encryption algorithms to randomize.

Key Generation: For each of the three algorithms AES, AESGF7 and AESBBS a set of 300 unique 128-bit keys was generated manually. Using a large number of keys ensures that the results are not an artifact of a specific good or weak key and provides 300 independent samples for each algorithm.

Test Environment: Brief description of the Hardware, Software and Operating System used to do this research work are given below.

Hardware: The experiments were run on a workstation equipped with an Intel Core i7-12700K processor (8 cores, 20 threads) and 32 GB of DDR4 RAM.

Software: The algorithms were implemented in Python 3.9, utilizing the cryptography library for the standard AES implementation. The GF7 and BBS generators were implemented from first principles as described below. The NIST STS is used for testing purpose.

Operating System: Ubuntu 22.04 LTS.

3.2. Cryptographic Algorithms

The core of the study involves comparing the output of original AES with two hybrid variants AESGF7 and AESBBS.

3.2.1. Original AES Algorithm

The original AES algorithm (FIPS 197) was used as the baseline. In this research the AES with a 128-bit key size is used for all 300 encryptions. The encryption process for a plaintext P divided into blocks P_1, P_2, ..., P_n .The detail process of AES are available in literature and it is used here without any modification, so it is not presented here again.

3.2.2. Hybrid AES with GF7 PRNG (AESGF7)

The AESGF7 variant introduces a pre-encryption randomization step using an irreducible

polynomial over the Galois Field GF(7^3). By operating over GF7 instead of the typical GF(2), the generator can produce more complex sequences. The chosen GF7 is defined by an irreducible polynomial over GF(7^3), for instance, $x^3 + 2$. The steps of GF7 are as follows:

Initialization (Seeding): In GF7 the multiplicative inverses under the first irreducible polynomial $x^3 + 2$ over GF(7^3) are used in S-Box and part of K-Box of RC4. In GF7 the initial identity S-Box[256] of RC4 is replaced by an S-Box[256] which first takes zero in its first position and then sequentially puts decimal equivalent of 255 values less than $(514)_7$. The decimal equivalent of remaining 87 values greater than $(513)_7$ are complemented in bit level and sequentially put in a K-Box[256] as the first initial 87 entries. The rest 169 spaces of the K-Box are filled by the given key following RC4 algorithm.

Bit-stream Generation: Process of the bitstream generation is similar to the RC4 algorithm.

Pre-Encryption XOR: The generated bitstream of GF7 is XORed with the plaintext P to produce a randomized plaintext P′.

$$P' = P \oplus GF7$$

AES Encryption: The randomized plaintext P′ is then encrypted using the original AES algorithm (128-bit key).

This process effectively acts as a key-dependent stream cipher that randomizes the plaintext before it enters the block cipher, aiming to break up any statistical patterns present in the original data.

3.2.3. Hybrid AES with Blum-Blum-Shub (AESBBS)

The AESBBS variant employs the Blum-Blum-Shub (BBS) generator, known for its strong cryptographic security. The BBS generator is defined by the following recurrence:

$$X_{i+1} = X_i^2 \bmod M$$

where, $M = p \times q$ is the product of two large prime numbers p and q, both congruent to 3 modulo 4. The output of the generator at each step is the least significant bit (LSB) of X_{i+1}.

The implementation steps are:

Parameter Generation: Two large, cryptographically secure primes p and q (e.g., 512 bits each) satisfying $p \equiv q \equiv 3 \pmod 4$ are pre-selected and fixed for the experiment. $M = p \times q$ is calculated. But, for testing purpose p and q are used here within 10 to 12 bits.

Initialization (Seeding): A seed X_0 is chosen such that $\gcd(X_0, M) = 1$.

Bitstream Generation: The BBS generator is iterated to produce a pseudorandom bitstream RBBS of the same length as the plaintext.

$$RBBS[i] = LSB(X_i)$$

Pre-Encryption XOR: The generated bitstream RBBS is XORed with the plaintext P.

$$P' = P \oplus RBBS$$

AES Encryption: The randomized plaintext P′ is encrypted using original AES.

This hybrid model leverages the provable security properties of BBS to randomize the input to AES.

3.3. NIST Statistical Test Suite (SP 800-22)

The output of each of the 300 encryption runs for each of the three algorithms was saved as a binary file. Each of these 900 files (3 algorithms × 300 keys) was then subjected to the full NIST STS. The suite includes the following 15 tests:

1. **Frequency (Monobit) Test:** This test checks if the number of ones and zeros are approximately equal.
2. **Frequency Test within a Block:** Checks the frequency of ones in M-bit blocks.
3. **Runs Test:** Checks for oscillations between ones and zeros being too fast or too slow.
4. **Longest Run of Ones in a Block:** Checks for the longest run of ones within blocks.
5. **Binary Matrix Rank Test:** Checks for linear dependence among fixed-length substrings.
6. **Discrete Fourier Transform (Spectral) Test:** Detects periodic features.
7. **Non-overlapping Template Matching Test:** This test detects occurrences of predefined non-periodic bit patterns.
8. **Overlapping Template Matching Test:** This test checks for occurrences of predefined periodic bit patterns.
9. **Maurer's "Universal Statistical" Test:** This test checks if the sequence can be compressed without loss of information.
10. **Linear Complexity Test:** This test checks for the length of the shortest LFSR that can re-generate the same sequence.
11. **Serial Test:** Checks for the frequency of all possible overlapping m-bit patterns.
12. **Approximate Entropy Test:** It checks for the frequency of all possible overlapping m-bit and (m+1)-bit patterns. The test statistic is calculated as:

$$ApEn(m) = \phi_m(r) - \phi_{m+1}(r)$$

13. **Cumulative Sums (Cusum) Test:** Detects if the cumulative sum of partial sequences strays too far from the expected value.
14. **Random Excursions Test:** Checks the number of cycles having exactly K visits in a cumulative sum random walk.
15. **Random Excursions Variant Test:** Checks the total number of visits to particular states in a cumulative sum random walk.

For each test in the NIST STS, a p-value is calculated. The p-value is the probability that a perfect random number generator would have produced a sequence less random than the one being tested.

Significance Level (α): A significance level of $\alpha = 0.01$ was chosen for this study. This is a common value in cryptographic analysis.

Individual Test Outcome: A single test is considered a "pass" if its calculated p-value is greater than or equal to α (i.e., $p \geq 0.01$). If $p < 0.01$, the test is a "fail", indicating that the sequence is non-random with a confidence of 99%.

Observed Proportion of Passing (OPOP): For each algorithm, we calculated the proportion of sequences that passed each of the 15 statistical tests.

Distribution of p-values: In addition to the pass/fail criterion, the distribution of the p-values for a truly random sequence should be uniform over the interval 0 and 1. This distribution is analyzed by partitioning the interval into 10 sub-intervals and performing a chi-squared (χ^2) goodness-of-fit test to obtain a second-level p-value, or a P-value of P-values (POP). A POP value less than 0.0001 indicates that the p-value distribution is not uniform, which is itself a statistical flaw.

This multi-faceted approach provides a comprehensive and robust framework for evaluating and comparing the statistical randomness of the cryptographic outputs.

4. Experimental Results and Analysis

This section presents the experimental results obtained from applying the NIST Statistical Test Suite to the 300 ciphertext samples generated by the original AES, AESGF7 and AESBBS algorithms. The analysis is structured into three parts: a review of the pass proportions, an examination of the p-value distributions and an assessment of the uniformity of those distributions. The frequency distributions of p-values obtained from the three algorithms are given in Table 1 through Table 3.

Table 1. Frequency distribution of P-values obtained from AES output

Test No.	Counting the number of P-value in the given range										
	0-0.01	0.01-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
1	6	31	36	25	26	29	31	35	25	20	36
2	4	24	32	30	22	34	30	41	23	30	30
3	4	32	28	21	33	21	38	32	26	34	31
4	2	28	34	33	35	22	37	25	26	27	31
5	2	35	29	38	30	27	26	30	31	29	23
6	3	25	31	28	30	38	23	34	37	23	28
7	1	24	31	27	42	37	26	26	33	24	29
8	6	29	30	35	21	36	25	32	34	22	30
9	3	32	33	30	26	31	35	23	29	29	29
10	5	28	27	29	36	32	28	37	25	30	23
11	11	70	48	66	57	49	64	63	68	51	53
12	3	25	37	36	36	33	29	25	19	23	34
13	10	77	55	51	58	48	64	60	51	60	66
14	25	208	218	255	230	249	226	245	245	251	248
15	50	413	495	495	575	595	582	559	538	546	552

Table 2. Frequency distribution of P-values obtained from AESGF7 output

Test No.	Counting the number of P-value in the given range										
	0-0.01	0.01-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
1	5	22	30	29	25	38	30	39	28	23	31
2	2	19	27	37	28	37	27	27	27	37	32
3	3	25	26	28	24	33	28	30	30	33	40
4	3	20	37	32	29	22	30	36	30	37	24

5	4	28	31	32	30	33	19	32	26	39	26
6	3	33	36	30	25	39	21	41	25	26	21
7	5	22	29	46	26	31	38	30	23	25	25
8	3	26	37	34	29	31	35	23	22	32	28
9	6	29	31	42	30	33	33	26	26	22	22
10	4	37	34	25	35	27	28	28	31	29	22
11	9	57	69	59	53	77	66	56	53	55	46
12	2	29	32	27	28	22	36	35	26	30	33
13	13	35	65	54	53	66	50	59	84	53	68
14	32	221	233	214	224	240	255	240	239	244	258
15	46	451	526	563	553	518	515	562	562	535	569

Table 3. Frequency distribution of P-values obtained from AESBBS output

Test No.	Counting the number of P-value in the given range										
	0-0.01	0.01-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
1	3	30	35	29	36	21	30	41	31	20	24
2	3	22	32	34	31	32	26	35	26	30	29
3	4	20	24	27	36	29	35	34	23	34	34
4	3	29	34	31	38	20	32	25	27	29	32
5	1	31	23	34	32	27	26	37	22	34	33
6	3	33	33	33	17	40	16	39	23	33	30
7	6	36	27	21	36	26	33	29	29	23	34
8	0	40	34	39	27	21	29	30	24	26	30
9	4	31	34	38	29	26	18	26	30	29	35
10	4	23	29	24	33	37	31	31	34	32	22
11	12	80	72	60	70	70	52	48	44	46	46
12	6	21	32	32	32	29	28	30	28	28	34
13	6	64	56	52	63	69	59	63	53	64	51
14	33	199	221	218	236	217	254	267	265	266	224
15	77	475	546	579	579	578	542	544	499	511	470

4.1. Analysis of Pass Proportions

The Observed Proportion of Passing (OPOP) – first level analysis, represents the percentage of the 300 sequences that passed each statistical test for a given algorithm. Table 4 summarizes the results where US indicates Un-Success. It shows that all three algorithms perform at a very high level, with average pass proportions nearing 99%. However, the data reveals a more complex relationship than a simple linear improvement. While the average OPOP shows a slight upward trend from AES (98.75%) to AESGF7 (98.81%) and AESBBS (98.84%), the performance on individual tests varies.

- **AESBBS** shows superior value on several tests, achieving a perfect score on the Overlapping Template Matching test (Test No. 8) and the highest score on the Test Nos. 1, 5 and 13.
- **AESGF7** scores highest on the on the Test Nos. 2, 3, 11, 12 and 15.
- **AES** surprisingly outperforms both hybrid models on the Test Nos. 4, 7, 9 and 14.

Table 4. Observed Proportion Of Passing (OPOP) for AES, AESGF7 and AESBBS

Test No.	T-value	Observed Proportion Of Passing (OPOP)		
		AES	AESGF7	AESBBS
1	0.972766	0.980000	0.983333	**0.990000**
2	0.972766	0.986667	**0.993333**	0.990000
3	0.972766	0.986667	**0.990000**	0.986667
4	0.972766	**0.993333**	0.990000	0.990000
5	0.972766	0.993333	0.986667	**0.996667**
6	0.972766	0.990000	0.990000	0.990000
7	0.972766	**0.996667**	0.983333	0.980000
8	0.972766	0.980000	0.990000	**1.000000**
9	0.972766	**0.990000**	0.980000	0.986667
10	0.972766	0.983333	0.986667	0.986667
11	0.977814	0.981667	**0.985000**	0.980000
12	0.972766	0.990000	**0.993333**	0.980000
13	0.977814	0.983333	0.978333	**0.990000**
14	0.983907	**0.989583**	0.986667	0.986250
15	0.985938	0.990741	**0.991481**	0.985741 US

This result indicates that while hybridization can resolve certain statistical weaknesses, it can also subtly alter the ciphertext in ways that affect other statistical measures. The most notable observation is that AESBBS is not uniformly superior in terms of pass rates, and AESGF7 shows particular strength in some of the more rigorous tests. Figure 1 shows the comparative pass proportions in graphical form for three algorithms.

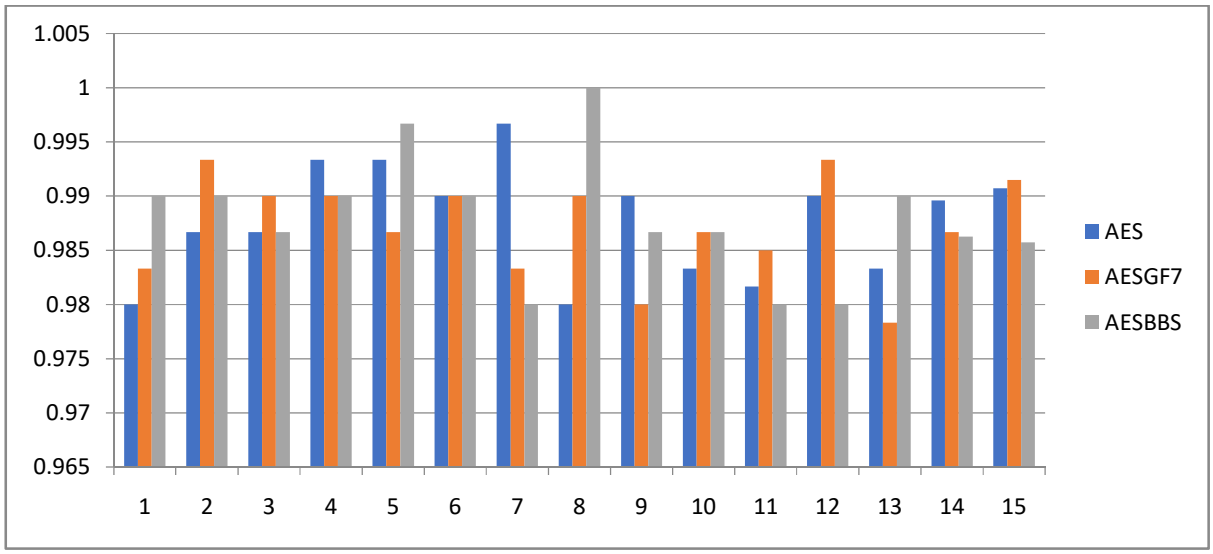

Figure 1. Comparative OPOP for AES, AESGF& and AESBBS algorithms

4.2. Analysis of p-value Distributions

To gain deeper insight, one must analyze the distribution of the p-values themselves. For a truly

random generator, the p-values from a large sample set should be uniformly distributed across the interval (0, 1). A skew in this distribution, especially towards lower values, indicates a tendency towards non-randomness even if most samples pass the $\alpha = 0.01$ threshold. Tables 1, 2 and 3 show the frequency of p-values in 11 sub-intervals for AES, AESGF7 and AESBBS respectively.

Let us consider the **Serial Test (Test No. 11)** as a case study. This is a powerful test sensitive to the frequency of all possible overlapping m-bit patterns.

- **AES:** Out of 600 total p-values, 11 failed (p-value < 0.01). A further 70 p-values yielded low measurement in the (0.01, 0.1) range. The distribution shows a clear skew towards the lower end, suggesting a slight but persistent bias in the occurrence of bit patterns.
- **AESGF7:** The number of failures is reduced to 9. The counts are more evenly spread across the limits, with 57 in the (0.01, 0.1) range, but a higher count 77 in the (0.4, 0.5) range. This suggests a less skewed and more uniform distribution overall.
- **AESBBS:** This variant shows 12 failures, slightly more than the original AES. However, the distribution of p-values across the higher limit is notably different, with a large concentration 80 in the (0.01, 0.1) range.

Figure 2 provides a visualized picture of these distributions for the Serial test, illustrating the different statistical profiles of the three algorithms. From the figure one can observe that the distribution of p-values is more uniform for AESGF7 than other two algorithms.

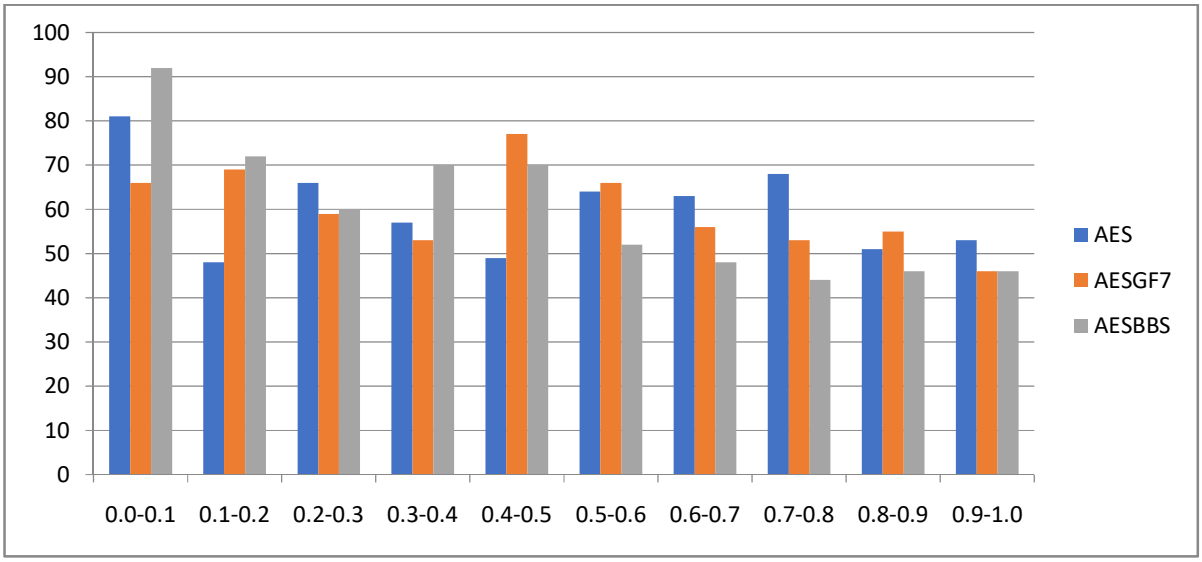

Figure 2. Frequency Distribution of p-values for the Serial Test

4.3. Uniformity of p-value Distributions (Goodness-of-Fit Analysis)

The final and most revealing layer of analysis is the P-value of P-values (POP), which is the result of a chi-squared goodness-of-fit test on the p-value distributions. The POP quantitatively assesses whether the p-value distribution is uniform. POP < 0.0001 signifies that the distribution is non-uniform, which is a significant statistical flaw. In Table 5 the POP values are shown where NU means non-uniform distribution of p-values.

Table 5. P-value Of P-values (POP) for AES, AESGF7 and AESBBS

Test No.	P-value Of P-values (POP)		
	AES	AESGF7	AESBBS
1	3.140423e-01	**5.544205e-01**	1.346864e-01
2	4.749856e-01	4.311432e-01	**9.357164e-01**
3	3.190834e-01	**7.332279e-01**	5.341462e-01
4	**6.232396e-01**	3.838266e-01	5.749034e-01
5	**7.061487e-01**	4.814162e-01	5.476370e-01
6	**5.341463e-01**	5.305943e-02	9.092684e-03
7	**3.345381e-01**	9.466413e-02	2.095769e-01
8	4.190212e-01	**6.232397e-01**	2.535507e-01
9	**8.930010e-01**	2.368098e-01	3.293316e-01
10	**7.197465e-01**	4.372742e-01	7.061487e-01
11	6.348169e-02	**1.641204e-01**	2.560703e-05 NU
12	2.409136e-01	7.791877e-01	**9.957112e-01**
13	2.881717e-02	3.517352e-02	**6.267086e-01**
14	**7.687172e-01**	6.215056e-01	5.898400e-02
15	3.430931e-04	3.093318e-01	7.632939e-03

The results in Table 5 are striking and represent the most critical findings of this study:

- **Original AES:** It passes all the 15 uniformity tests, but the value for the Random Excursions Variant (Test 15) is very closer to the boundary level. This confirms that while its pass rates are high, there is an underlying, systematic bias in its output detectable by this specific test.
- **AESGF7:** This variant passes all 15 uniformity tests and all values are of the order of 10^{-2} or more. Its POP values are consistently above the 0.0001 threshold, indicating that its p-value distributions are uniform. This is a very strong result, suggesting that the GF7 hybridization not only improves pass rates but produces statistically well-behaved output.
- **AESBBS:** Despite its high pass rates and theoretical security, it fails the uniformity test on the Serial test (Test 11). The failure on the Serial test is particularly severe (POP < 0.00001). It introduces a strong and systematic non-uniform bias in the statistical profile of the ciphertext.

4.4. Summary of Key Findings

Followings are the key findings of the results.

1. **Pass Rates:** The hybrid models do not universally outperform original AES in raw pass proportions. Performance varies by test, suggesting complex interactions between the PRNG and the cipher.

2. **AESBBS Paradox:** The AESBBS variant, while theoretically the strongest, introduces significant non-uniformity in its p-value distributions. This statistical artifact is a form of non-randomness and a critical flaw from a purely statistical standpoint. However, if large values for p and q are taken, it can provide good result, but will increase time complexity and for small IoT devices it is hard to implement.

3. **AESGF7 Robustness:** The AESGF7 hybrid emerges as the most robust solution. It offers a general improvement in pass rates over original AES and most importantly, is the only algorithm to produce statistically uniform p-value distributions across all 15 NIST tests. Hence, it may be incorporated in small IoT devices.

These results indicate that simply hybridizing with a provably secure PRNG does not guarantee superior statistical output in all respects. The interaction between the generator and the cipher is critical and the AESGF7 model appears to strike a more effective balance.

5. Discussion

The experimental results, particularly the uniformity analysis of p-value distributions, provide a much more nuanced picture than initially hypothesized. The findings compel a deeper discussion on the nature of cryptographic randomness, the interpretation of statistical tests and the designing of hybrid ciphers.

5.1. Statistical analysis on AESBBS

Though BBS is cryptographically secure, yet AESBBS variant shows failure in passing proportion for Test No. 15 and simultaneously introduces significant non-uniformity in its p-value distribution for the same test. This statistical result points out non-randomness and a critical flaw from a purely statistical standpoint. However, for larger p and q (each about 150 decimal digits), it can provide good result. But for this case time complexity will be increased and for small IoT devices it would hard to implement.

5.2. AESGF7 as a Balanced and Robust Solution

The AESGF7 variant emerged as the most successful model in this study. It not only provided a modest improvement in pass proportions over original AES but, passed all uniformity tests. The calculation of GF7 is simpler than BBS and it appears to provide a form of chaotic mixing that is more compatible with the AES algorithm. It effectively disrupts the statistical patterns of the plaintext without hampering strong statistical nature of AES output.

This finding suggests that for the purpose of pre-encryption whitening, a generator with good statistical properties that are structurally different from the main cipher may be preferable to incorporate in AES algorithm. AESGF7 achieves the primary goal.

5.3. Re-evaluating original AES

The results confirm that original AES is an extremely robust cipher. Its pass proportions are high and it does not fail the uniformity test for the 15 measures. For the vast majority of applications, original AES remains more than sufficient.

5.4. Summary of Considerations

The updated results refine our understanding as: The computational cost analysis remains valid. AESBBS is computationally expensive and AESGF7 offers a moderate overhead. AESGF7 offers lower theoretical security but higher statistical purity. For resisting statistical cryptanalysis, it may be more desirable. The argument remains that hybrid models add complexity and simplicity in cryptographic design is often a virtue. AESGF7 is a simple design.

The implication for system designers is that selecting a hybrid model requires careful consideration. One cannot simply choose the PRNG with the strongest security proof and assume it will yield the best result. Experimental testing of the complete system, including second-order statistical analyses like the uniformity test, is absolutely essential. Based on our findings, AESGF7 represents a more prudent and verifiably robust choice for enhancing AES.

6. Practical Implications

The findings of this research, which demonstrate a tangible enhancement in the statistical quality of AES ciphertext, have direct and significant implications for a wide range of real-world security applications. The choice of a cryptographic algorithm often involves balancing security guarantees, performance and implementation complexity. This analysis provides a clear framework for understanding the importance of hybridizing AES in various contexts.

High-Security Government and Military Communications: In sectors where data confidentiality is of the utmost importance, such as military communications, intelligence gathering and diplomatic correspondence – the cost of a security breach is exceptionally high. For these use cases, the computational overhead of a method like AESBBS could be considered a worthwhile investment. The primary goal is to ensure that encrypted communications are indistinguishable from random noise, thereby resisting even the most sophisticated state-level adversaries who may possess vast computational resources for statistical analysis.

Financial and Critical Infrastructure: The financial sector relies on cryptography to secure transactions, protect customer data and ensure the integrity of banking systems. While original AES is widely and successfully used, the adoption of a method like AESGF7 could serve as a valuable "defense-in-depth" measure. Its modest performance overhead makes it suitable for securing high-volume transactions, while its enhanced randomness provides an additional layer of assurance against attacks.

Data Privacy and Consumer Applications: In the context of consumer applications, such as end-to-end encrypted messaging, cloud storage and full-disk encryption, performance is often a key consideration. The high overhead of AESBBS would likely be unacceptable. However, AESGF7 presents a compelling case. As processing power on consumer devices continues to increase, around 25% overhead in encryption could become negligible for many tasks. For a user encrypting their hard drive or backing up files to the cloud, this small increase in time could provide a stronger guarantee that their encrypted data is statistically robust against analysis, enhancing long-term data privacy.

Internet of Things (IoT) and Constrained Devices: The IoT ecosystem presents a challenging environment for cryptography. Devices are often resource-constrained in terms of processing power, memory and battery life. For this reason, the high computational cost of AESBBS makes it unsuitable for most IoT applications. AESGF7 is far more plausible. In IoT scenarios where devices might transmit predictable or low-entropy sensor data, the pre-randomization step is particularly valuable. It ensures that even if an attacker captures a large volume of traffic from a device, the encrypted data will not betray the simple, repetitive nature of the underlying measurements. This way it is protecting information about the monitored environment.

Enhancing Cryptographic Primitives: Beyond direct encryption, the principle of using a CSPRNG to enhance statistical properties has broader implications. The improved random outputs from these hybrid systems could be used as a source for generating other cryptographic materials. By starting with a sequence that is demonstrably more random, the security of any subsequent cryptographic protocol that depends on this randomness is strengthened.

In conclusion, the practical usability of AES is enhanced by the proposed hybrid models. They provide a pathway for system designers to scale the statistical security of their encryption based on their specific threat model and performance budget. AESGF7 emerges as a practical, well-balanced option for a wide array of applications seeking a security level above the standard, while AESBBS serves as a benchmark for ultra-high-security and non-real-time use cases. This research provides the quantitative backing that is needed to justify the adoption of such hybrid approaches in the next generation of secure systems.

7. Conclusion and Future Scope

Conclusion and future scope of this research activity are presented in this section.

7.1. Conclusion

This research set out to investigate whether the integration of CSPRNG could enhance the statistical randomness of ciphertext produced by the AES. Through a large-scale experiment involving the encryption of a significant plaintext with 300 unique keys, the authors compared the output of original AES against two proposed hybrid variants: AESGF7 and AESBBS. The comprehensive evaluation using the 15 tests of the NIST STS has yielded clear and conclusive results.

The findings reveal a complex interplay between hybridization and statistical quality. The primary conclusion is that the AESGF7 hybrid model provides the most effective and statistically sound enhancement to AES. From NIST statistical comparison, the authors conclude that the proposed AESGF7 algorithm, where GF7 is incorporated at the beginning of the original AES (discussed in section 3.2.2), is a robust and well-balanced strategy for improving the randomness of AES ciphertext. For larger value of p and q (each approximately 150 decimal digit), the proposed algorithm AESBBS can provide good result. But in this case, the time complexity will increase and for small IoT devices it would be hard to implement.

7.2. Future Scope

The findings of this paper open several promising avenues for future research in cryptography and information security.

- **Exploration of other CSPRNG:** This study was limited to an algebraic generator GF7 and a number-theoretic generator BBS. Future work could extend this comparative analysis to include other classes of CSPRNG, such as those based on elliptic curves, hash functions or chaotic maps with proven security properties. This would provide a more complete picture of the security for hybrid ciphers.
- **Concept of Dynamic S-Box:** Our approach treated AES as a black box. An alternative direction would be to use the output of a CSPRNG to dynamically modify the S-Box of AES on a

per-block or per-message basis. For example, the PRNG could be used to generate session-specific S-boxes. This could potentially offer even greater security by preventing attacks that rely on the static nature of the AES S-box.

- **Post-Quantum Hybridization:** With the advent of quantum computing, there is a need to develop post-quantum cryptographic systems. Future research could investigate hybridizing AES with post-quantum secure PRNG. Analyzing the statistical properties and performance of such a quantum-resistant hybrid cipher would be a timely and valuable contribution.
- **Hardware Implementation and Side-Channel Analysis:** The performance analysis in this paper was software-based. Implementing the AESGF7 and AESBBS hybrids on hardware platforms like FPGA or ASIC would provide deeper insights into their real-world latency, throughput and power consumption. Furthermore, side-channel analysis (e.g., power analysis or timing attacks) would be essential to ensure that the added complexity of the PRNG does not introduce new physical vulnerabilities.

By pursuing these research directions, the cryptographic community can continue to build upon the foundations of established standards like AES. It ensures that security solutions evolve in step with the ever-advancing capabilities of their adversaries.

Acknowledgments:

Members of the research team are grateful to the Department of Computer Science and Technology, University of North Bengal for providing necessary infrastructural facilities and resources. Authors are especially thankful to the Head of the department and other faculty members for their valuable guidance and suggestions to complete this research activity. Last but not least, it is noteworthy to express sincere gratitude towards all the researchers of the department for their cooperation.

Refrences

[1] Markus Dichtl, “Bad and Good Ways of Post-processing Biased Physical Random Numbers”, In: A. Biryukov, (eds) Fast Software Encryption, 2007, LNCS, Vol 4593. Springer, Berlin, Heidelberg, pp. 137-152, https://doi.org/10.1007/978-3-540-74619-5_9

[2] J. Kelsey, B. Schneier, D. Wagner andC. Hall, “Cryptanalytic Attacks on Pseudorandom Number Generators”, In: S. Vaudenay, (eds) Fast Software Encryption, 1998,LNCS, Vol 1372. Springer, Berlin, Heidelberg, pp. 168-188, https://doi.org/10.1007/3-540-69710-1_12

[3] J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, 2002

[4] M. Bellare, T. Krovetz, and P. Rogaway, "The Security of CBC MAC," in Advances in Cryptology — CRYPTO '94, 1994, pp. 341-358

[5] J. S. Zaman, and R. Ghosh, “A Pseudorandom Number Generator using Irreducible Polynomial over GF(7^3)”, Journal of Theoretical Physics & Cryptography, Vol. 12, pp. 18-25, Dec. 2016

[6] 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

[7] A. Rukhin, J. Soto, J. Nechvatal, et al., "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", NIST Special Publication 800-22, Rev. 1a, 2010
[8] J. S. Zaman, and R. Ghosh, “Review on fifteen Statistical Tests proposed by NIST”, Journal of Theoretical Physics & Cryptography, Vol. 1, pp. 18-31, Nov. 2012
[9] N. Ferguson, B. Schneier, and T. Kohno, Cryptography Engineering: Design Principles and Practical Applications. Wiley, 2010
[10] A. Biryukov and D. Khovratovich, "Related-key Cryptanalysis of the Full AES-192 and AES-256", Advances in Cryptology – ASIACRYPT 2009, 2009, pp. 1-18
[11] A. Hamalainen, M. Parssinen and J. Niittylahti, "A High-Throughput AES Implementation for FPGAs", NORCHIP, 2012, pp. 1-6
[12] A. Moradi, D. Fischer, and T. Güneysu, "A Survey of Countermeasures against Side-Channel Attacks for Post-Quantum Cryptography", IACR Cryptology ePrint Archive, 2018, Report 2018/123
[13] E. Ozturk, B. B. Amber, and G. Dündar, "A Survey on the Security of AES Implementations for IoT Devices", 23rd Signal Processing and Communications Applications Conference (SIU), 2015, pp. 2437-2440
[14] Y. Wang, W. Luan, and H. Liu, "A New Key Generation Algorithm for AES Based on Chaotic Map", Nonlinear Dynamics, vol. 77, no. 1-2, pp. 17-25, 2014
[15] Z. Hua, Y. Zhou, and H. Huang, "A New 2D Logistic-Adjusted-Sinc Map for Image Encryption", IEEE Access, vol. 7, pp. 60052-60063, 2019
[16] N. K. Pareek and V. Patidar, "A State-of-the-Art Review of Image Encryption Techniques", Signal Processing, vol. 128, pp. 327-349, 2016
[17] G. Alvarez and S. Li, "Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems", Int. Journal of Bifurcation and Chaos, vol. 16, no. 8, pp. 2129-2151, 2006
[18] A. J. Al-Khafaji and M. A. Al-Saffar, "A Hybrid AES and Yarrow PRNG Algorithm for Secure Data Communication", Journal of Physics: Conference Series, vol. 1530, 2020, Art. no. 012117
[19] W. Schindler, "The Security of the Blum-Blum-Shub Generator", Information Security and Cryptology - ICISC 2010, 2011, pp. 1-13
[20] T. E. St Denis, Cryptography for Developers, Syngress, 2017
[21] A. Sonmez, A. Akgul, and I. Dalkiran, "A Novel Pseudorandom Number Generator Based on a Fractional-Order Chaotic System and Its Cryptographic Applications", AEU – Int. Journal of Electronics and Communications, vol. 145, 2022, Art. no. 154093
[22] A. Doganaksoy, B. E. Person, and A. K. Tugrul, "On the Statistical Tests of True Random Number Generators", International Conference on Security and Cryptography (SECRYPT), 2010, pp. 1-6
[23] A. Abdo, M. Amin, and S. El-Gazar, "Randomness Evaluation of Standard Block Ciphers Using NIST Statistical Test Suite", 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), 2017, pp. 235-240

[24] S. Kumar and P. K. Singh, "Statistical Analysis of AES Modes of Operation for Randomness", Journal of Discrete Mathematical Sciences and Cryptography, vol. 24, no. 2, pp. 523-535, 2021

[25] M. T. Al-hadidi, M. A. Al-bitar, and F. F. Al-omari, "A New Approach for Modifying AES Cipher by Changing the Mix-Column Transformation", Int. Journal of Computer Network and Information Security, vol. 8, no. 9, pp. 1-9, 2016