Misan Journal of Engineering Sciences ISSN-E: 2957-4250
Vol. 5, No. 1, 28 June 2026 ISSN: 2957-4242

Hash Function-Based Multi-Layer AES Key Expansion Framework

Bushra Jaber M.Jawad*[1]

[1]*University of Kerbala, Kerbala-Iraq.*
**Corresponding author E-mail: bushra.j@uokerbala.edu.iq*

(Received 22 Feb, Revised 8 June, Accepted 8 June)

Abstract: The advanced encryption standard (AES) is one of the most commonly used encryption algorithms because of its effective encrypted capabilities and strong crypto-security. Due to the growing demand for stronger cryptographic protection, researchers have been finding alternative enhanced key management and key expansion methods that will improve randomness, nonlinearities, and statistical security properties within the framework of AES.In this paper, we proposed a hybrid SHA3−AES framework that utilizes three complementary techniques for strengthening AES. SHA3 is utilized to preprocess the master key to produce a hardened key with higher entropy; dynamic external key evolution employs iterative hashing with SHA3 to produce round dependent keys, thus reducing the number of deterministic relationships that exist between round keys; and SHA3 is also incorporated into AES's internal key expansion process to provide increased non-linearity and complexity for generating round keys. The objective of the proposed framework is to enhance diffusion characteristics of the process and increase randomness of the encryption process while preserving the original AES structure. Security analysis indicates that integrating SHA3 improves key unpredictability and strengthens random behavior through dynamic key dependency and continuous key evolution. Furthermore, experimental observations show that the proposed framework provides substantial improvements to the statistical and security metrics (i.e., entropy, avalanche effect, and randomness distribution) when compared to the traditional implementations of AES. Results indicate that the proposed SHA3-AES model provides a second layer of security for applications with the need for a higher level of security against advanced analytical attacks.

Keywords: Symmetric Encryption; Key Expansion; AES; Hash Function; SHA3

1. Introduction

With the rapid development of cloud computing, digital communication systems, and Internet-Based Applications, the need for highly secure and efficient methods of encrypting sensitive information to protect against the new and emerging cyber threat has escalated [1], [2], [3]. Advanced Encryption Standard (AES) is the most commonly used cryptographic standard because it provides the highest level of data security, is computationally efficient, and can be applied across a wide range of software and hardware platforms [4], [5], [6], [7]. As a result, AES has become widely used in many different areas such as secure communications, cloud storage, banking and finance, and IoT environments. However, the development of new techniques for cryptanalysis and the advancement of high-performance computing have prompted researchers to look for ways to possibly improve upon the current AES internal structure, focusing on the key expansion mechanism. The current key schedule of AES relies on deterministic operations (i.e. RotWord, SubWord, XOR, and Rcon) for producing the round keys that will be used for encrypting/decrypting the data using a given master key [8], [9], [10]. While these operations provide sufficient diffusion and efficiency for creating the round keys, they could also introduce structural dependencies and/or mathematical relationships among the round keys due to their deterministic nature, which could lead to the feasibility of conducting advanced analytical attacks (i.e., differential and linear cryptanalysis).

DOI: https://doi.org/10.61263/mjes.v5i1.279
This work is licensed under a Creative Commons Attribution 4.0 International License.

More recently, researchers working on the expansion of the AES key have focused on improving security, performance, and randomness using sophisticated computational and structural methods. Today's techniques have been designed to overcome the deterministic nature of traditional key design by using random, parallel, and hybrid cryptographic architectures.

In [11], authors introduced a one-dimensional chaotic map based key expansion technique to increase the unpredictability of the ring keys and the resilience to statistical attacks. However, since the model introduces an external chaotic behavior into the traditional AES key scheduling without changing it, the efficiency of removing all structural dependencies is limited. In addition, chaotic systems may introduce a sensitivity in parameter selection and implementation complexity and may therefore not be optimal for lightweight environments.

A multi-chaos key expansion model for AES is proposed to improve dissemination and resilience to cryptanalysis attacks. However, multiple chaotic generators introduce an increased computational overhead, while still concentrating mainly on external random number generation to counter any internal correlations between the ring keys [12]. The AES-ECC hybrid encryption model is developed with a dynamic S-box mechanism to increase the security of cloud services; while this model increases the resilience to different types of external attack, it introduces additional computational complexity and is almost exclusively focused on increasing the encryption layer, rather than the actual process of scaling the AES key(s) [13]. In addition, In [14], the authors developed an alternative key expansion method for AES, based on dynamic access, to increase the security of files sent over university networks. Although this method provides more flexibility in terms of application and functionality than traditional AES key scheduling techniques, it has a minimal impact on the AES key expansion process itself and does not use strong non-linear primitives (e.g. transforms based on hash characters) that could increase security. A low latency DDBT-based AES key expansion unit is proposed, resulting in lower latency and increased hardware efficiency. Although the unit offers many hardware optimization benefits, it has no direct influence on randomness and internal non-linearity and does not provide a means to defeat analytical attack methods [15]. According to the literature, significant progress has been made in improving the key expansion function of AES using chaotic systems, hybrid encryption techniques, dynamic structures, and optimized hardware. However, there are still many unexplored avenues of research in this field. Current methods for increasing the size of AES keys rely mainly on generating external randomness; focus on application-specific increases in the size of AES keys or on hardware optimization only, limiting the development of integrated lightweight hash functions directly to the process of pinning the external and internal AES key expansion functions. However, authors typically limit improvements to an isolated aspect of the function, namely randomness or efficiency, without providing a comprehensive framework for simultaneously increasing entropy in the master key, making the keys ring-dependent, and decreasing the predictability between round keys or enhance the non-linearity of the internal AES key schedule.

To address these limitations, a new research paper proposes a hybrid AES-SHA3 framework that integrates the SHA3 hash function into several layers of the AES Key Management process. The proposed methodology consists of three different techniques: pre-processing external keys with SHA3, generating dynamic round-dependent keys with SHA3 iterative hashing, and generating internal key expansion using SHA3. The integrated design aims to minimize the correlation between round keys, increase the diffusion and randomness properties, and improve the resilience to differential and linear cryptanalysis while maintaining the AES's performance.

The main contributions of this research can be summarized as follows:

- Proposing a hybrid AES–SHA3 cryptographic framework integrating SHA3 into both external and internal AES key management processes.
- Introducing a SHA3-based external key preprocessing mechanism to improve entropy and randomness of the master key.
- Developing a dynamic round-dependent key evolution model using iterative SHA3 hashing to reduce deterministic relationships among round keys.
- Designing an internal SHA3-based AES key expansion mechanism to increase non-linearity and

strengthen cryptographic complexity.

- Evaluating the proposed framework using multiple statistical security metrics including entropy, avalanche effect, randomness distribution, and differential probability analysis.

The remainder of this paper is organized as follows. Section II presents the proposed multi-level key AES SHA3-based. Section III the security analysis of the proposed model. Section IV presents experimental results and performance evaluation. Finally, Section V concludes the paper and outlines future research directions.

2.Proposed Work

In this section, we present the cryptographic framework that enhances the extensiveness of the AES key by integrating SHA3 into several key management layers. The aim is to increase key randomness,non-linearity and resilience against a range of statistical and analytical attacks.

The methodology is composed of three different mechanisms. External Key Preprocessing, Dynamic External Key Evolution and Internal Key Extraction using SHA3, as illustrated in Figure 1.

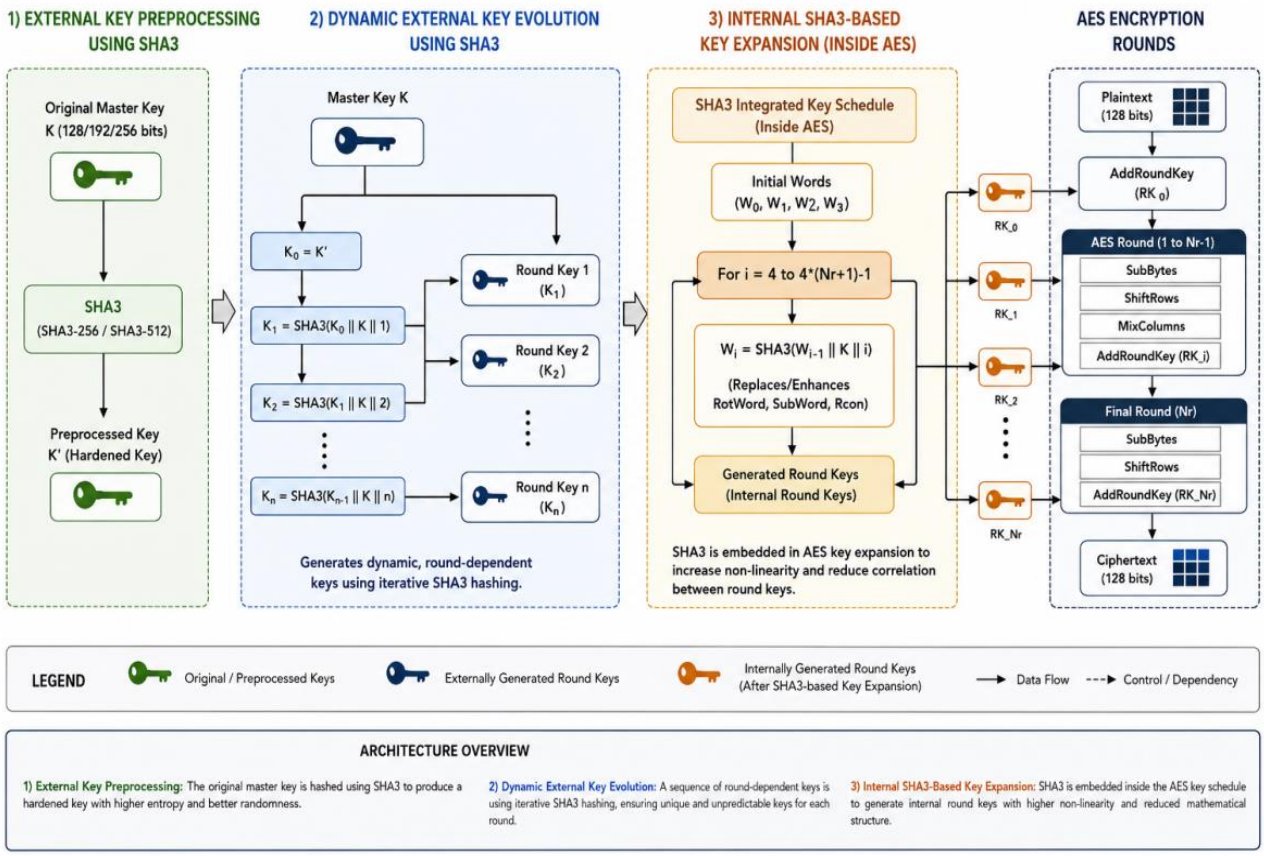


Fig.1 Block Diagram of the Proposed Framework

2.1 External Key Preprocessing Using SHA3

The secret key of AES encryption algorithm is first started with a SHA3 hash function (such as SHA3-256 or SHA3-512). The result of a hash function is treated as a derived key, as it is considered more statistically secure.

Let the original secret key denoted by K. The modified key K' as follows in Equation 1:

$$K' = \text{SHA3}(K) \tag{1}$$

This transformation increases entropy of the input key and removes the structure or correlation of the original key. Significantly, this step does not change the AES internal structure, but improves the quality of the cipher input.

2.2 Dynamic External Key Evolution Using SHA3

A second stage is proposed, the dynamic external key generation scheme, in which several round-dependent keys are calculated outside the AES key scheduling. The iterative hashing process generates a string of different keys, Ki, instead of using a simple static key, by: Ki−1, which is calculated in Equation2:

$$K_i = \text{SHA3}(K_{i-1} \| K \| i) \quad (2)$$

where:

- K_{i-1}, is the previous round key,
- K, is the master key,
- i, is the round index,
- ‖ denotes concatenation.

Each round key is dependent on previously generated round values and is non-linearly dependent on preceding ones. Consequently, the round keys are effectively decorrelated, providing a higher degree of security to the overall system. Yet, this phase continues to operate externally to the internal key expansion process of AES.

2.3 Internal SHA3-Based Key Expansion

In the final stage, the AES key schedule is directly modified by the SHA-3 function, which makes it different from the regular key schedule. The generation of new round keys incorporates SHA3, which adds additional diffusing effects to the new round keys.
The modified key expansion process can be expressed as in Equation 3:

$$Wi = \text{SHA3}(W_{i-1} \| K \| i) \quad (3)$$

where:

- W_i represents the generated key word,
- W_i = SHA3(W_{i-1}) is the previous key word,
- K is the master key,
- i is the round number.

The key schedule becomes more complex due to the non-linear integration of mechanisms, as each subsequent round key is more dependent on all previous round keys than the standard AES key expansion process.

2.4 Overall Framework Summary

The scheme integrates three distinct techniques through a three-layer framework: first, to strengthen the master key by pre-processing data using SHA3, second, to dynamically generate round keys using an external iterative hashing method, and third, to embed SHA3 in the AES internal key schedule for further cryptographic diffusion and non-linearity. Overall, statistical randomness is improved, the avalanche effect is maximized, and resistance to cryptanalysis is enhanced.

3. Experimental Setup

The experimental aimed at improving key generation of the AES algorithm, setup including two subsections: Hardware and Software Specifications and Metrics for Performance Measurement.

3.1 Hardware and Software Specifications

The proposed method is implemented on a laptop with an Intel i7 core processor and 16GB RAM, operating system is Windows 11Pro.

The choice of software tools also greatly affects the evaluation process. So, to simulate the proposed framework, we used an Integrated Development Environment (IDE) Pycharm which is specifically designed for Python programming, which allows efficient synthesis, implementation and simulation of the developed algorithms. The testing framework includes both proprietary solutions and open source libraries to measure performance effectively.

3.2 Evaluation Performance and Security of the Proposed Framework

When evaluating the effectiveness of the AES algorithm with proposed key expansion techniques, several key metrics are essential, including:

3.2.1 Performance Metrics

To measure the efficiency of the propsed keys are used performance metrics include: Encryption speed and throughput.[16]

- **Encryption speed**

Encryption speed or encryption time is the time required for encryption, and the time needed to convert plaintext to ciphertext.[17]

- **Throughput**

The Throughput is calculated by using the Equation 4.

$$Throughput = \frac{Plaintext\ size(byte)}{Encryption\ time(ms)} \quad (4)$$

3.2.2 Security Metrics

We used many statistical tests to measure randomness and unbredictable of proposed keys:

- **Entropy**

Entropy is a measure used to assess the degree of randomness or unpredictability in data. In cryptography, the goal is for data to be: Unpredictable, randomly distributed and difficult to guess The higher the entropy value, the more random the data and the stronger it is cryptographically.In binary data or ciphertext, entropy measures the degree of variation in the distribution of values. If the data contains a recurring pattern, the entropy is low and If all values are distributed relatively evenly, the entropy is high. The most commonly used equation is the Equation 5 for entropy [18], [19].

$$H(X) = -\sum_{i=1}^{n} p(x_i)\ \log_2 p(x_i) \quad (5)$$

- $H(X)$ = Entropy.
- $p(xi)$ = Probability of x.
- log2 = Binary logarithm.
- $\sum$ = Sum of all probabilities.

Therefore, in cryptography tests: the highest possible value is approximately 8 because a byte contains 8 bits.

- **Monobit Test**

The Monobit Difference is one of the randomness tests used to measure the "balance" of binary bits (0 & 1) in result sets from encryption algorithms, key generation, or from random numbers. Its purpose is to determine whether the number of 1s is close to the number of 0s. In good random data, the ratio of zeros to ones should be very close, approximately: [20], [21]
Number of 1s ≈ Number of 0s

The difference is calculated as follows:

$$Monobit\ Difference = | N_1 - N_0 | \quad (6)$$

Where:

- N_1 = Number of bits with a value of 1
- N_0 = Number of bits with a value of 0
- $|\cdot|$ = Absolute value

A small value indicates a balanced distribution and better randomness of the data. If the value is too large, this may indicate a bias in the data, meaning that the algorithm produces more 1s than 0s or vice versa.

- **Chi-Square Test**

The chi-squared test is a statistical test used to measure how much the actual distribution of data differs from the expected distribution. In cryptography and randomness, this test is used to determine whether the distribution of bytes or bits generated by an algorithm is close to a perfect random distribution.

The chi-square test compares:

- Observed values
- Expected values

Then it measures the difference between them. The basic equation is: [22]

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (7)$$

Where:

- χ^2 = chi-squared value
- Oi = observed frequency
- Ei = expected frequency
- $\sum$ = sum of all classes

- **Avalanche Effect**

The Avalanche Effect is a very important metric in cryptography that measures how sensitive an algorithm is to small changes. The basic idea: a minor change in the input message such as flipping one bit, results in a significantly different hash value. Avalanche effect wherein at least 50% of the bits of the hash value are flipped by a minor change in the input [23], [24] [25]. The equation is:

$$Avalanche\ Effect = \frac{Number\ of\ Changed\ Bits}{Total\ Bits} \times 100 \quad (8)$$

Where:

- Number of Changed Bits = The number of bits that changed between the two outputs
- Total bits = The total number of bits
- The result is a percentage

4. Results And Discussion

4.1 Comparing KDF and PBDFK with SHA3

The results in Table 1 show that all three methods possess good cryptographic properties, but with significant differences in performance and security efficiency.

Table 1: Comperison between SHA3, KDF and PBDFK

Metrics	PBKDF2	HKDF	SHA256
Average Time	0.038	0.001	0.0009
Entropy	7.94	7.94	7.96
Chi-Square	247	229	242
Monobit Difference	6	62	19
Avalanche Effect	46.4%	50%	49.6%

SHA-256 exhibited the fastest encryption time and the highest entropy, indicating a high level of randomness.

HKDF displayed the best avalanche effect (50%), which is nearly ideal, and a lower chi-square value than the other methods, indicating a balanced statistical distribution and good strength in propagating changes within the output.

PBKDF2 was the slowest (0.038) due to its reliance on security iterations, but achieved a low monobit difference (19), indicating good bit balance.

Overall:

- SHA256 is the best in terms of speed and randomness.
- HKDF is the best in terms of statistical strength and avalanche effect.
- PBKDF2 offers higher security against guessing attacks, but at the expense of performance.

Therefore, HKDF offers the best overall balance between efficiency and security performance, while SHA-256 is best suited for high-speed applications, and PBKDF2 is best suited for password protection.

4.2 Evaluating Performance

The results in Figure 3 indicate that encryption time increases gradually with increasing data size in both methods, a predictable behavior given the increasing amount of data to be processed.

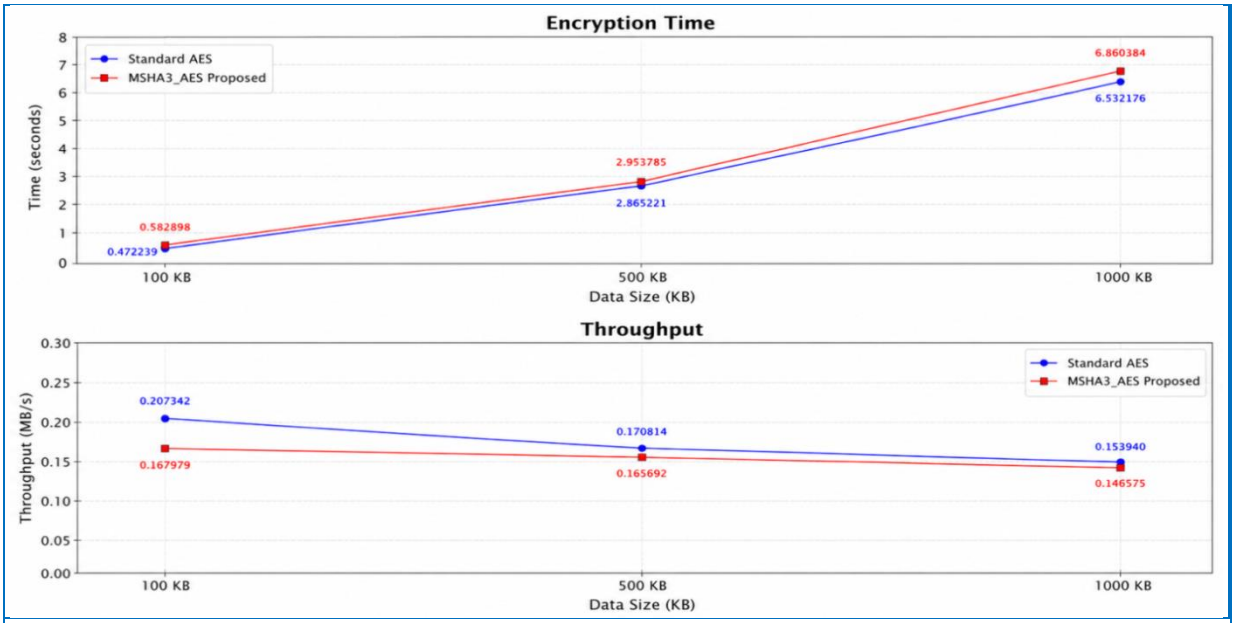


Fig. 2 Performance tests of the proposed framework

The proposed SHA3-based method exhibited a slightly higher execution time compared to the conventional AES algorithm. This is attributed to the additional operations involved in implementing SHA3 within the encryption mechanism or key generation, which adds a small computational cost to the encryption process.

Also, concerning throughput, the traditional AES algorithm was slightly faster than the SHA3 algorithm; since there is a negative correlation between total execution time and throughput (i.e., the more time it takes to encrypt something; the less data can be encrypted per unit time) the amount of difference in throughput for either encryption method had a relatively negligible effect on overall system performance (especially when working on very large amounts of data).

Despite this slight performance decrease, the differences between the two methods remain relatively small, particularly at large datasets where performance converges more closely. Therefore, the proposed method can be considered to achieve an acceptable trade-off between performance and security, as the slight increase in encryption time and the limited decrease in Throughput come in exchange for the improvements shown by previous results in measures of randomness and statistical propagation such as Entropy, Bit Difference Probability and Chi-Square, indicating an enhancement of the system's security characteristics without imposing a high performance cost.

4.2 Statistical tests of Proposed Framework

Figure 3 includes four ststistical tests conducted for proposed framework,

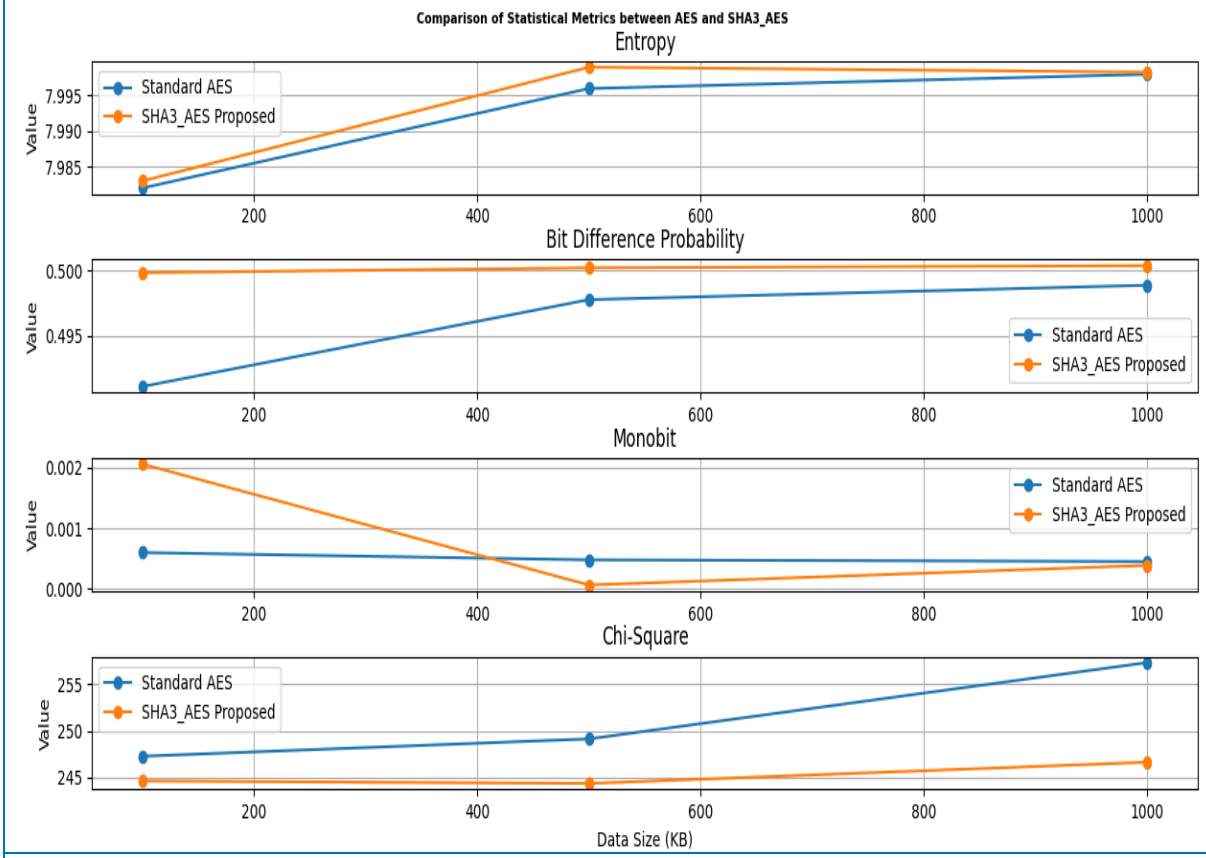


Fig. 3 Statistical tests of the proposed framework

According to the results in Figure 4, the proposed SHA3_AES system generated better (or relatively similar) statistical and security tests in relation to the Standard AES. This indicates that the proposed system produced randomly distributed data with more strength (i.e., encryption) and did not negatively impact system stability.

- **Entropy Analysis:** Entropy values are very close to the optimal value (8), indicating a high level of randomness within both encrypted payload types. However, the SHA3_AES was marginally superior with respect to most data sizes (i.e., superior bit distribution and fewer predictable patterns) and its values remained stable despite the increase in the total amount of data. Therefore, this demonstrates that the SHA3_AES preserves randomness at scale satisfactorily.
- **Bit Difference Probability Analysis:** The SHA3_AES system's performance was much closer to the optimal value (0.5) than Standard AES. This corresponds with the strong Avalanche Effect (that is, the output of an algorithm is unique due to an inconsequential input being changed (e.g., data or key), thereby resulting in very close correlations between both systems' outputs to approximately 50 % of total output. This closes the door for statistical and differential attacks. While both systems were relatively close, the SHA3_AES's performance produced more statistically exact results than Standard AES.
- **Monobit Analysis:** The Monobit values are really low in both systems. This means that the ciphertext has a mix of zeros and ones which is important for making it random. The new system has lower Monobit values especially when the sizes are bigger. This shows that the new system is better at making ciphertext and reducing patterns.
- **Chi-Square Analysis:** The Chi-Square test shows that the proposed system has values than the standard AES at all sizes. This means that the proposed system makes ciphertext that looks like random bytes. The lower Chi-Square value also means that the new system is better at hiding patterns in the ciphertext.

Generally**:** The results confirm that integrating SHA-3 with AES enhances the statistical features of the ciphertext, principally the randomness, bit difference rate, and resistance to statistical analysis. Furtheremore, the differences, although sometimes minor, are consistent and recurring across different data sizes, which supports the effectiveness of the proposed system and its potential implementation as a more secure encryption model when is compared to traditional AES.

5. Conclusions

The experimental results show that the proposed AES-SHA3 framework consistently achieves higher entropy, better bit variance probabilities, and lower chi-square values than the standard AES, while maintaining the same monobit properties across different data sizes. These improvements are due to increased randomness, increased avalanche behaviour and increased sensitivity to statistical analysis attacks. The proposed system therefore provides an effective improvement over traditional AES and is suitable for security-sensitive applications such as secure communications and IoT security systems. It is confirmed that increasing the SHA-3 hash by adding a key or by changing the key generation process will significantly increase the overall power of the cryptographic techniques, without compromising performance.

Author Contributions: The authors contributed to all parts of the current study.

Funding: This study received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] D. Kumar Sharma, N. Chidananda Singh, D. A. Noola, A. Nirmal Doss, and J. Sivakumar, “A review on various cryptographic techniques & algorithms,” *Materials Today: Proceedings*, vol. 51, no. xxxx, pp. 104–109, 2021, doi: 10.1016/j.matpr.2021.04.583.

[2] R. Yadav, “Analysis of Cryptography in Information Technology,” *Interantional Journal of Scientific Research in Engineering and Management*, vol. 07, no. 03, pp. 1–6, 2023, doi: 10.55041/ijsrem18379.

[3] H. Baha Eldin Hassan Hamouda, "Comparative study of different cryptographic algorithms," *Journal of Information Security*, vol. 11, no. 4, pp. 138–148, 2020, doi: formation Security, 11, 138-148. https://doi.org/10.4236/jis.2020.113009.

[4] U. H. Shaikh, M. M. Abbas, S. A. Lahad, M. Razi, and M. Shaikh, "A Comparative Survey of Symmetric and Asymmetric Key Cryptography Algorithms," *2nd International Multidisciplinary Conference on Emerging Trends in Engineering Technology-2024 (2nd IMCEET-2024)*, no. October, pp. 257–262, 2024.

[5] M. N. Alenezi, H. Alabdulrazzaq, and N. Q. Mohammad, "Symmetric encryption algorithms: Review and evaluation study," *International Journal of Communication Networks and Information Security*, vol. 12, no. 2, pp. 256–272, 2020.

[6] K. Assa-Agyei and F. Olajide, "A Comparative Study of Twofish, Blowfish, and Advanced Encryption Standard for Secured Data Transmission," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, pp. 393–398, 2023, doi: 10.14569/IJACSA.2023.0140344.

[7] A. Ghosh, "Comparison of Encryption Algorithms : AES , Blowfish and Twofish for Security of Wireless Networks," *International Research Journal of Engineering and Technology (IRJET)*, no. June, pp. 4656–4659, 2020, doi: 10.13140/RG.2.2.31024.38401.

[8] M. Divya, "Research on Cryptography and Network Security," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 05, no. 01, pp. 1532–1536, 2023, doi: 10.56726/irjmets33258.

[9] M. B. Yassein, S. Aljawarneh, E. Qawasmeh, W. Mardini, and Y. Khamayseh, "Comprehensive study of symmetric key and asymmetric key encryption algorithms," *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, vol. 2018-Janua, pp. 1–7, 2017, doi: 10.1109/ICEngTechnol.2017.8308215.

[10] A. Olutola and M. Olumuyiwa, "Comparative Analysis of Encryption Algorithms," *European Journal of Technology*, vol. 7, no. 1, pp. 1–9, 2023, doi: 10.47672/ejt.1312.

[11] Y. Li, "Application of high performance one-dimensional chaotic map in key expansion algorithm," *2023 3rd International Conference on Electronic Information Engineering and Computer, EIECT 2023*, pp. 85–91, 2023, doi: 10.1109/EIECT60552.2023.10442043.

[12] K. Assa-Agyei, K. Owa, F. Olajide, and T. Al-Hadhrami, "A Multi-Chaotic Key Expansion for Advanced Encryption Standard (AES) Algorithm," *2024 International Conference on Computing, Networking and Communications, ICNC 2024*, pp. 711–717, 2024, doi: 10.1109/ICNC59896.2024.10556263.

[13] M. N. J. R. , Nagaraj M Lutimath, "An Enhanced AES-ECC model with Key Dependent Dynamic S-Box for the Security of Mobile Applications using Cloud Computing," *Journal of Electrical Systems*, vol. 20, no. 2, pp. 2735–2746, 2024, doi: 10.52783/jes.2050.

[14] R. Astillero, "Optimizing Advanced Encryption Standard (AES) for Enhanced File Security in University Networks Using Dynamic Key Expansion," *Technologique: A Global Journal on Technological Developments and Scientific Innovations*, vol. 3, no. 1, pp. 51–59, 2024, doi: 10.62718/vmca.tech-gjtdsi.3.1.sc-1124-004.

[15] X. Zheng, H. Yan, Z. Peng, and X. Zhang, "Low-Delay AES Key Expansion Units Based on DDBT Structure †," *Electronics (Switzerland)*, vol. 14, no. 2, 2025, doi: 10.3390/electronics14020384.

[16] B. J. M. Jawad, "The Effect of Block Cipher Modes on Confidentiality and Authentication in Network Communication Security," no. 14, pp. 25–43, 2026, doi: 10.52113/3/eng/mjet/2026-14-0.

[17] Bushra Jaber M.Jawad and S. Al-alak, "Design and Implementation of Multi-key Blowfish and CAST Algorithm: Comparative Study with CBC, CFB and CTR Modes," *Wasit Journal of Computer and Mathematics Science*, vol. 2, no. 4, pp. 87–98, 2023, doi: 10.31185/wjcms.203.

[18] A. K. Bairwa and S. Joshi, "Mutual authentication of nodes using session token with fingerprint and MAC address validation," *Egyptian Informatics Journal*, vol. 22, no. 4, pp. 479–491, 2021, doi: 10.1016/j.eij.2021.03.003.

[19] M. E. Manaa and R. H. Jwdha, "A proactive data security scheme of files using Minhash technique," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 24, pp. 8421–8433, 2018.

[20] G. Yellapu, "Necessity of frequency (Monobit) test for pseudo random bit generators," vol. 2667, 2018, doi: 10.1080/02522667.2016.1231969.

[21] Z. Mengdi, Z. Xiaojuan, Z. Yayun, and M. Siwei, "Overview of Randomness Test on Cryptographic Algorithms Overview of Randomness Test on Cryptographic Algorithms," 2021, doi: 10.1088/1742-6596/1861/1/012009.

[22] N. H. Munshi, P. Das, and S. Maitra, "Chi-Squared Test Analysis on Hybrid Cryptosystem," *Micro and Nanosystems*, vol. 14, no. 1, p. 2022, 2022, doi: 10.2174/1876402913666210508235706.

[23] M. Imdad, S. N. Ramli, and H. Mahdin, "An Enhanced Key Schedule Algorithm of PRESENT-128 Block Cipher for Random and Non-Random Secret Keys," *Symmetry*, vol. 14, no. 3, pp. 1–22, 2022, doi: 10.3390/sym14030604.

[24] A. On, A. Effect, and I. N. Cryptography, "ANALYSE ON AVALANCHE EFFECT IN CRYPTOGRAPHY ALGORITHM," in *European Proceedings of Multidisciplinary Sciences EpMS e-ISSN:*, 2022, pp. 610–618. doi: 10.15405/epms.2022.10.57.

[25] D. Upadhyay, N. Gaikwad, M. Zaman, and S. Sampalli, "Investigating the Avalanche Effect of Various Cryptographically Secure Hash Functions and Hash-Based Applications," *IEEE Access*, vol. 10, no. October, pp. 112472–112486, 2022, doi: 10.1109/ACCESS.2022.3215778.