Enhanced security and performance through permutation-byte key cipher with reduced-round AES

Jerico S. Baladhay, Alvincent E. Danganan, Edjie M. De Los Reyes, Heidilyn V. Gamido, Marlon V. Gamido
College of Computer Studies, Tarlac State University, Tarlac, Philippines

Article Info

Article history:

Received May 27, 2024
Revised Aug 27, 2025
Accepted Sep 27, 2025

Keywords:

Advanced encryption standard
Bit permutation
Encryption
Key expansion
Round keys

ABSTRACT

This paper introduces the permutation-byte key cipher with reduced-round advanced encryption standard (PBKC-RRAES), a novel enhancement of the AES designed to significantly improve both security and performance. The proposed algorithm integrates key modifications; i) replacing the computationally intensive MixColumns function with an efficient bit permutation technique that achieves superior diffusion while reducing computational overhad by eliminating complex matrix multiplication operations. This substitution enhances security through improved bit-level scrambling patterns, while simultaneously accelerating processing speed through simpler bitwise operations; ii) the addition of AddRoundKey operations between cipher states, iii) enhanced byte substitution operations and round constant additions in the key schedule algorithm before key expansion, and iv) reducing rounds from 10 to 6. These innovations yield heightened sensitivity to plaintext changes, evidenced by a 54.214% avalanche effect, surpassing the standard 50% threshold. Performance evaluations reveal PBKC-RRAES operates 26.90% improvement in encryption time and a 22.73% improvement in decryption time than standard AES, alongside throughput enhancements of 39.48% in encryption and 31.27% in decryption compared to the original AES, critical improvements for bandwidth-constrained applications. These results demonstrate that PBKC-RRAES is a robust and effective alternative for cryptographic applications, particularly beneficial for real-time video streaming, secure cloud storage, mobile payment systems, and IoT device where both security and processing effectivity are paramount.

Corresponding Author:

Jerico S. Baladhay
College of Computer Studies, Tarlac State University
Tarlac, Philippines
Email: jsbaladhay@tsu.edu.ph

1. INTRODUCTION

The rapid advancement of information technology has significantly transformed communication from traditional to digital formats, leading to increased concerns about the confidentiality of information during its transmission [1], [2]. As digital communication becomes more prevalent, the need to secure data from unauthorized access is paramount, particularly during transmission between the source and the intended recipient [3]. Encryption is a well-established method for safeguarding data before it is transmitted over networks, converting it into a format that is unreadable to unauthorized individuals and only decipherable by those with the correct decryption key [4], [5].

However, the fact that despite the widespread adoption of encryption techniques, existing algorithms like the advanced encryption standard (AES) are facing increasing scrutiny due to potential vulnerabilities arising from advances in processing power [6] and execution techniques [7]. Modern computing capabilities, including parallel processing, specialized hardware accelerators, and emerging computational paradigms, continue to challenge the long-term security assumptions of current cryptographic standards. Additionally, the increasing sophistication of cryptanalytic techniques and the growing computational resources available to potential adversaries necessitate continuous evaluation and enhancement of existing encryption methods [8]. As a result, there is a growing need to enhance these cryptographic methods to ensure they can continue to provide robust security against emerging threats [9].

The AES, while widely adopted for its security, contains specific structural limitations that present opportunities for enhancement [10], [11]. Most notably, the MixColumns operation requires computationally expensive matrix multiplications that create processing bottlenecks [12], and the fixed 10-round structure may be unnecessarily conservative when combined with improved diffusion techniques [13], [14].

Despite its widespread use and recognized effectiveness, AES is not without limitations. The algorithm’s inherent simplicity, while a strength, also leaves it vulnerability to advances in processing and execution techniques that could compromise its security [15], [16]. As such, there is an increasing consensus on the need to enhance and update cryptographic methods to address emerging threats [17], [18]. There is a critical need to improve the security of cryptographic algorithms like AES while maintaining or even enhancing their performance, especially in medium to large-scale applications.

AES comprises four primary transformation functions: SubBytes, ShiftRows, MixColumns, and AddRoundKey [19]. Several studies have focused on enhancing the performance and security of the AES algorithm [20]. One study improves the performance of AES by replacing the MixColumns function with a bit permutation technique [21], which enhances the speed of encryption and decryption while reducing memory usage. Another study improves the performance by incorporating a precomputed table in the bit permutation process [22], which can manage medium-to-large scale file sizes [23]. A different study enhances the diffusion property of AES through two modifications [24]: the first modification involves the AES key schedule, where additional permutation operations are applied to the cipher key before it is expanded; the second modification occurs in the AES cipher rounds, introducing additional key permutation operations between states to achieve faster diffusion rates and better randomness of the encrypted data, subsequently reducing the number of rounds from 10 to 6 using reduced-round modified AES (RRMA) [25].

However, these existing modifications address either performance or security improvements in isolation. Performance-focused approaches using bit permutation [21], [22] lack enhanced diffusion properties, while security-focused modifications [24], [25] compromise computational efficiency. No existing study successfully integrates bit permutation with enhanced key scheduling and round reduction to achieve simultaneous improvements in both domains.

In this paper, we build upon these previous efforts by proposing permutation-based key cipher (PBKC)-reduced round advanced encryption standard (RRAES), which addresses specific AES limitations through targeted modifications; i) replacing the computationally intensive MixColumns function with a bit permutation technique to eliminate matrix multiplication overhead while maintaining diffusion quality, ii) introducing additional AddRoundKey operations between states to compensate for the reduced computational complexity with enhanced confusion properties, iii) adding byte substitution and round constant operations in the key schedule to strengthen key-dependent transformations without increasing round count, and iv) reducing rounds from 10 to 6, made feasible by the improved diffusion from our combined modifications to create a customized AES variant, referred to as the permutation-byte key cipher with reduced-round AES (PBKC-RRAES). The proposed PBKC-RRAES algorithm aims to achieve enhanced security and improve performance in terms of encryption and decryption processes across various file sizes.

2. METHOD

In this section, the methods, techniques, and processes of the modified algorithm are discussed. The algorithm incorporates the following operations to modify the AES: the bit permutation technique, a replacement of the MixColumn operation, and the application of the RRMA algorithm. The goal of the study is to strengthen the security and improve the performance of encryption and decryption processes for all file types and scale sizes.

2.1. Bit permutation technique

To improve the performance of a cryptographic algorithm, a customized AES encryption method is employed. In this adaptation, the bit permutation transformation replaces the conventional MixColumns transformation utilized in AES. Unlike the MixColumns transformation, the bit permutation transformation involves simple bit position adjustments, devoid of intricate mathematical computations. To execute the bit permutation during encryption, the state within the ShiftRows operation is manipulated according to the following steps [21]:

a. The state values for each column are extracted.
b. Column 0 is segmented into four rows, with each state represented as (x, 0), where x denotes the row number and consists of 8 bits, forming a 4×8 matrix. Here, ((x, 0), b) signifies the row, column, and bit number within each state.
c. The 4×8 matrix is alternatively represented as ‘a’.
d. The transpose of each block matrix is obtained.
e. The value of a’(x, y) is derived from a row-wise concatenation of bit values from the transposed block, where x represents the column value in ShiftRows, and y denotes the block number within the partitioned 4×2 matrix.

Additionally, a precomputed table is incorporated to optimize the performance of the modified algorithm, thus the operation of the technique is the following: the precomputed table maps input bit positions to output positions. It’s used to effectively rearrange the bits in the state matrix during encryption and decryption. The permutation mapping function retrieves the output position for each input bit position. The bit permutation and inverse bit permutation functions utilize this table to perform the permutation process.

2.2. Reduced-round modified advanced encryption standard algorithm

Figure 1 depicts an adapted rendition of the AES algorithm, elucidating its encryption procedure and key expansion. This customized algorithm integrates significant alterations aimed at bolstering both security and effectivity of performance. Through comprehensive adjustments, it seeks to optimize cryptographic strength while maintaining swift computational performance.

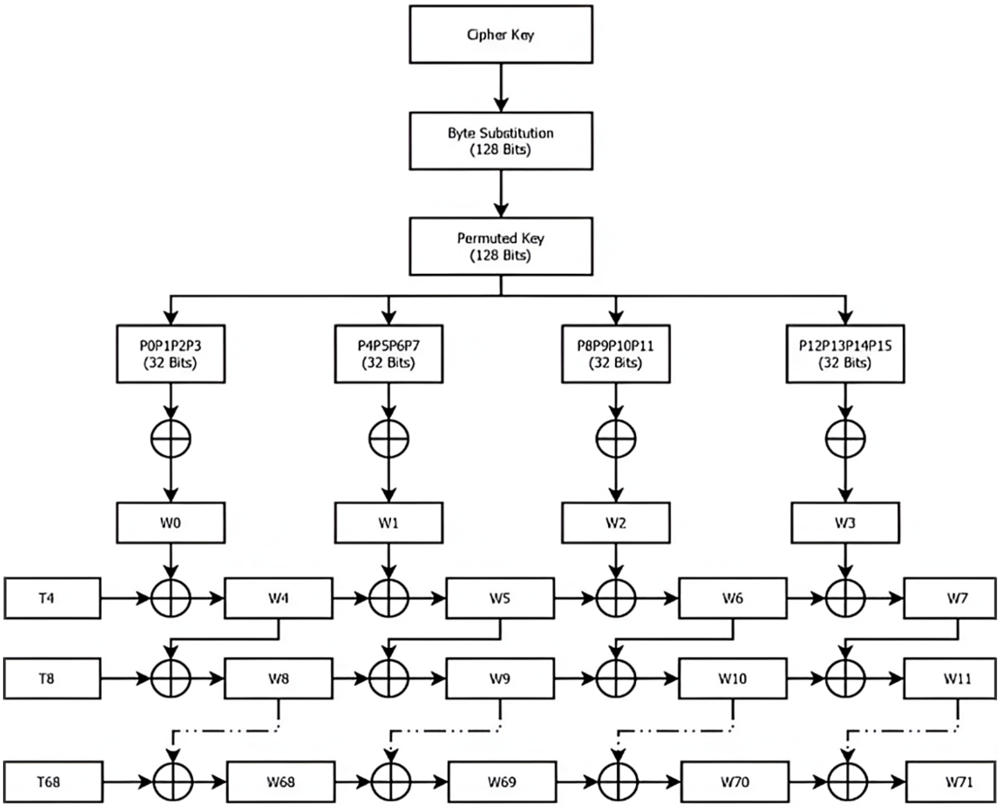


Figure 1. Cipher round algorithm and key schedule algorithm [24]

Encryption process:

1. Round 0: initial transformation
- AddRoundKey: the plaintext is XORed with the initial key
- With round key 0 (W0-W3)

2. Rounds 1 to 5: main rounds

Each round uses corresponding round keys:
- Round 1: uses W4-W7
- Round 2: uses W8-W11
- Round 3: uses W12-W15
- Round 4: uses W16-W19
- Round 5: uses W20-W23

Each of these rounds involves the following operations:
- SubBytes: a non-linear substitution step where each byte is replaced with another byte according to a lookup table (S-box).
- AddRoundKey: each byte of the state is XORed with a round key derived from the original cipher key.
- ShiftRows: a transposition step where each row of the state is shifted cyclically a certain number of steps.
- ModAddRoundKey: this appears to be a modular addition with the round key, possible implying a different type of key mixing step compared to the standard AddRoundKey.
- Bit permutation–involves bit position adjustments during encryption through extracting and segmenting state values into a 4×8 matrix, then transposing and re-concatenating these bits. This process rearranges the bit values to derive new state representations, and ensuring enhanced encryption complexity.

3. Round 6: final round: i) SubBytes, ii) ModAddRoundKey, iii) ShiftRows, and iv) AddRoundKey.
4. Output: the output after the final round is the ciphertext.
5. The decryption process reverses the encryption steps: starting with the inverse of the final round, it applies inverse SubBytes, ModAddRoundKey, ShiftRows, and AddRoundKey operations. Following this, it reverses the main rounds by undoing the bit permutation and applying the inverse operations for SubBytes, ShiftRows, and AddRoundKey. Finally, it reverses the initial transformation by undoing the XOR operation with the initial key. These steps collectively restore the original plaintext from the ciphertext.

2.3. Key expansion

The 128-bit cipher key undergoes byte substitution and permutation to create initial words (W0, W1, W2, and W3). Subsequent words (W4 to W71) are generated using XOR operations and modular additions for use in encryption rounds.

In conventional AES-128, the key expansion generates 44 words (W0 to W43) to produce 11 round keys (one initial key plus 10 round keys for 10 rounds). In contrast, PBKC-RRAES generates 28 words (W0 to W27) to produce 7 round keys (one initial key plus 6 round keys for 6 reduced rounds). This reduction is made possible by the enhanced diffusion properties achieved through the bit permutation and additional key operations, which provide equivalent security with fewer rounds.

The following are the processes:

a. Cipher key: the original 128-bit cipher key is provided as the initial input for the key schedule process.
b. Byte substitution: the cipher key undergoes a byte substitution step. This process is similar to the SubBytes step used in the encryption rounds of block ciphers like AES. It involves replacing each byte of the key with another byte according to a predefined substitution box (S-box).
c. Permuted key: after substitution, the key is permuted to enhance diffusion. The 128-bit substituted key is rearranged according to a fixed permutation pattern to produce the permuted key.
d. Key schedule generation:

- Division into words: the 128-bit permuted key is divided into four 32-bit words: P0P1P2P3, P4P5P6P7, P8P9P10P11, and P12P13P14P15.
- Initial words (W0 to W3): these words are XORed with predefined constant values to generate the initial words (W0, W1, W2, and W3) for the key schedule.
- Subsequent words (W4 to W71): the remaining words (W4 to W71) are generated through a series of operations involving XOR and modular additions. The process typically involves combining the previous words and applying transformations to produce a new word. For instance, W4 might be derived by XORing W0 with a transformation of W3, and similarly for subsequent words.

e. Usage in encryption rounds: the expanded key schedule, consisting of words W0 to W71, is used across different rounds of the encryption process. Each round of encryption utilizes specific words from the key schedule to perform operations such as substitution, permutation, and mixing, ensuring that each round effectively contributes to the overall security of the encryption.

2.4. Permutation-byte key cipher with reduced-round algorithm

The PBKC-RRAES algorithm as shown in Figure 2 enhances the traditional AES by incorporating additional steps such as modular addition (ModAddRoundKey) and bit permutation, which increase the complexity and security of the encryption process. The encryption sequence starts with an AddRoundKey step, followed by SubBytes, another AddRoundKey, ShiftRows, ModAddRoundKey, BitPermutation, and additional rounds of AddRoundKey, SubBytes, ModAddRoundKey, and ShiftRows, ending with a final AddRoundKey step to produce the ciphertext. The decryption process reverses these steps with corresponding inverse operations: AddRoundKey, InvSubBytes, AddRoundKey, InvShiftRows, InvModAddRoundKey, InvBitPermutation, and repeated inverse operations, concluding with a final AddRoundKey to recover the plaintext. This approach aims to reduce the number of rounds needed while maintaining strong security through more complex operations. The algorithm provides a balance of performance and robustness, ensuring accurate recovery of the original plaintext from the ciphertext.

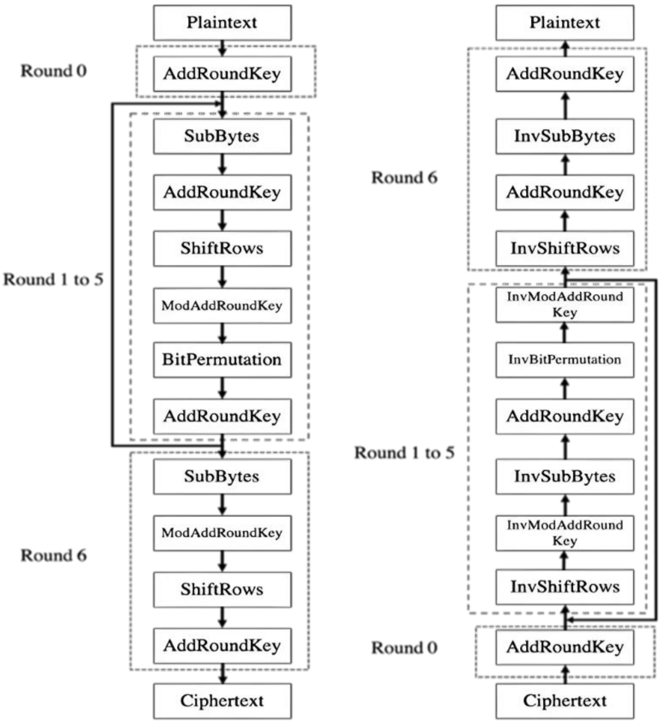


Figure 2. PBKC-RRAES algorithm overview

3. RESULTS AND DISCUSSION

This section discusses the results of the experiments conducted for the modified algorithm. It covers the outcomes of various tests, including the observation of ciphertext changes upon flipping a single plaintext bit, the assessment of the avalanche effect, where a recommended security threshold of 50% or higher is suggested, the analysis of encryption and decryption performance, focusing on time consumption or speed, and the evaluation of throughput to gauge the effectiveness of the modified algorithm's performance. Additionally, it is noted that the modified algorithm was developed using JavaScript and the server-side framework Node.js, JavaScript's dynamic, event-driven nature, and extensive ecosystem enable rapid, versatile development, while Node.js's asynchronous, non-blocking I/O model, and unified language environment enhance efficiency and consistency in server-side applications [26]. Together, they streamline development processes, making them ideal for complex algorithmic implementations.

3.1. Single bit flipping on advanced encryption standard and modified advanced encryption standard ciphertext

The tables illustrate the impact of flipping one bit in plaintext on the ciphertext using conventional and modified AES encryption methods. Table 1 lists the plaintext samples and their one-bit flipped counterparts. Table 2 shows the results of conventional AES encryption, with the percentage change in ciphertext ranging from 42.18% to 50%, indicating significant sensitivity to minor plaintext changes. Table 3 presents the modified AES results, where the percentage change in ciphertext ranges from 50.78% to 59.37%, demonstrating an increased sensitivity compared to conventional AES. This comparison highlights the enhanced diffusion properties of the modified AES algorithm. This demonstrates that the methods and techniques applied are effective, thereby strengthening and maximizing security.

Table 1. Input plaintexts

Samples	Plaintexts	Plaintexts with one bit flipped
1	0000000011111111	0 **1** 00000011111111
2	1010101010101010	10101010101 **9** 1010
3	aaaaabbbbccccddd	aaaaabb **1** bccccddd
4	abcdef123456abcd	a **f** cdef123456abcd
5	a1b2c3d4e5f60708	**8** 1b2c3d4e5f60708
6	f0e1d2c3b4a59687	f0e1d2c3b4a5968 **a**
7	1234567890abcdef	123456 **f** 890abcdef
8	eeeeeffff0000111	eeeeeffff0 **6** 00111
9	2222333344445555	2222 **d** 33344445555
10	f00dcafef00dcafe	**c** 00dcafef00dcafe

Table 2. Conventional AES ciphertext results

Samples	Encrypted plaintexts	Encrypted plaintexts with one bit flipped	%
1	3d050c6477180e2ac053dfeb3c071cbf	79560e4c4058f21dc376c5538d62ec4f	42.18
2	2e18068d36f00bbfe76f1f887192ccb7	afd8d7b4db69f72b803e0186cca6e1fc	49.21
3	89458db16d9ecd70dcb999f8ff03efc9	e507472183a3f00d9a50805836d628d0	50
4	a91e9959306f799f05c281699ee50f5a	3c4333a8308ff80ed9ee2a3b590de036	48.43
5	231ae0ae99827203e880ff1df328a5cf	02b07aa0eddfc51d211a29148c97ac4e	50.78
6	f75a563da1d74d9264102818e578e1b5	6fbf5fad309a8c6d5581e278c9593e0f	46.09
7	840c58af684a28a9295827eefa57fff1	8e7cd973803702a371390374ec34d91c	42.96
8	fe6a10a9b50fdea79eddb15e89ba982a	c84906ad70d127bf24b6a113ca740688	46.87
9	4f1cf7650c2cf7ed12210265a017a2eb	7fb9fff684edfc536cec17f27fbc7a28	50
10	2c0f1467ff9f381152dda7b32341ff52	35376dfbf232d890b6684c21023de682	46.09

Table 3. Modified AES ciphertext results

Samples	Encrypted plaintexts	Encrypted plaintexts with one bit flipped	%
1	e11ad3cd9d00a2086c794c0a95f58dda	1c59c2b6415cdb229798b5076b147b3b	59.37
2	3c7ccda8644f55d86c1b9c87ab3d86a7	ea10f2d107851c1a4708372737426b72	54.68
3	48318a804271853ee940cf1f5b51e2d7	051323f7b4377ad9438957430392 93a2	54.68
4	028763684f480e554a88fcfd7a285dc8	76e05029c53bfee320b9ebdfb3a38097	50.78
5	9b257f4ab68963907939a3a2c188b07b	56e3bf9959cd8429088554fa47904b1c	56.25
6	8a4c5129808e7e2936494c937be4d14d	f1724a7be0c962ca5c2165a418df6e67	51.56
7	810f1dab53b64004172b692298d4215d	292869071041dd4ff176455ae52825e0	54.68
8	414e04fc7cd127f4f5f69e3fc6db458e	8b6f61744627f687690d141d1d8a9f7c	51.56
9	4e68e63ff98fd9d6e36717e423371502	b0051bb366ea46e4ebbc8fa71ac5cc27	55.46
10	b09c70ba02ad6d6ec277f6063e4a9143	aca7752f5d726f2b4fb9096648d527e3	53.12

3.2. Avalanche effect

Figure 3 demonstrates that the modified algorithm achieves an avalanche effect of 54.214%, surpassing the expected value of 50%. This high degree of sensitivity to input changes underscores the algorithm’s enhanced diffusion capabilities. Consequently, the modified algorithm substantially bolsters the security of the ciphertext, as it ensures that small alterations in the plaintext result in significantly different ciphertexts. These findings affirm the robustness and performance of the modified AES algorithm in strengthening cryptographic security measures. It is evident that reaching such a high value of the avalanche effect not only proves the algorithm's security but also implies that it is highly resistant to differential cryptanalysis attacks. This enhanced diffusion makes it extremely difficult for attackers to predict or manipulate the output, thereby ensuring the integrity and confidentiality of the encrypted data. The implication of this result is that the modified algorithm can provide a higher level of security for sensitive information, making it suitable for applications requiring stringent cryptographic protections. To quantify the avalanche effect, we often use [27]:

$$Avalanche\ Effect\ (\%) = \frac{Number\ of\ changed\ output\ bits}{Total\ number\ of\ output\ bits}\ x\ 100 \tag{1}$$

In Figure 4 illustrates the avalanche effect when a single bit in the plaintext is flipped, showing the comparative results of the modified AES algorithm versus conventional AES. Each bar represents the avalanche percentage across 10 different samples. The data reveals that the modified AES consistently achieves higher avalanche percentages across all samples compared to conventional AES, reinforcing the robustness of the modified algorithm in maintaining high sensitivity to input changes. This highlights the improved diffusion properties and enhanced security provided by the modified AES algorithm.

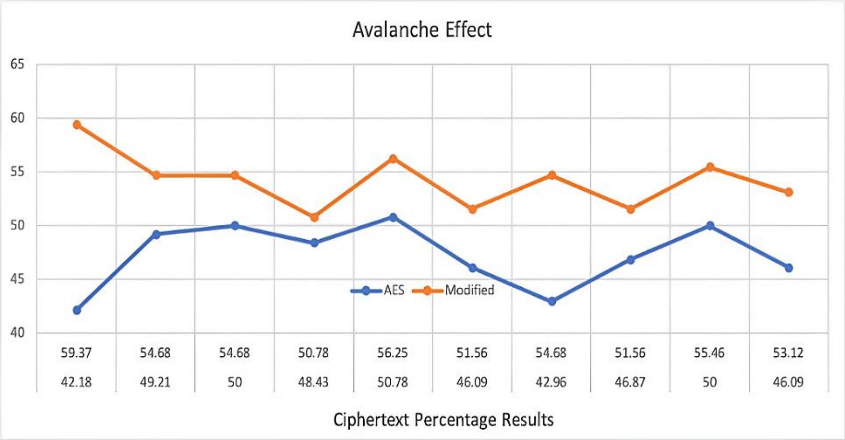


Figure 3. Avalanche effect

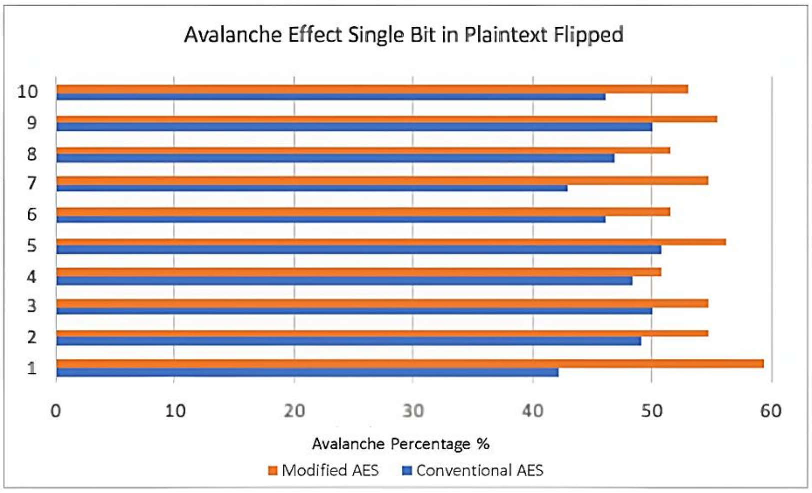


Figure 4. Avalanche effect bar graph: single bit in plaintext flipped

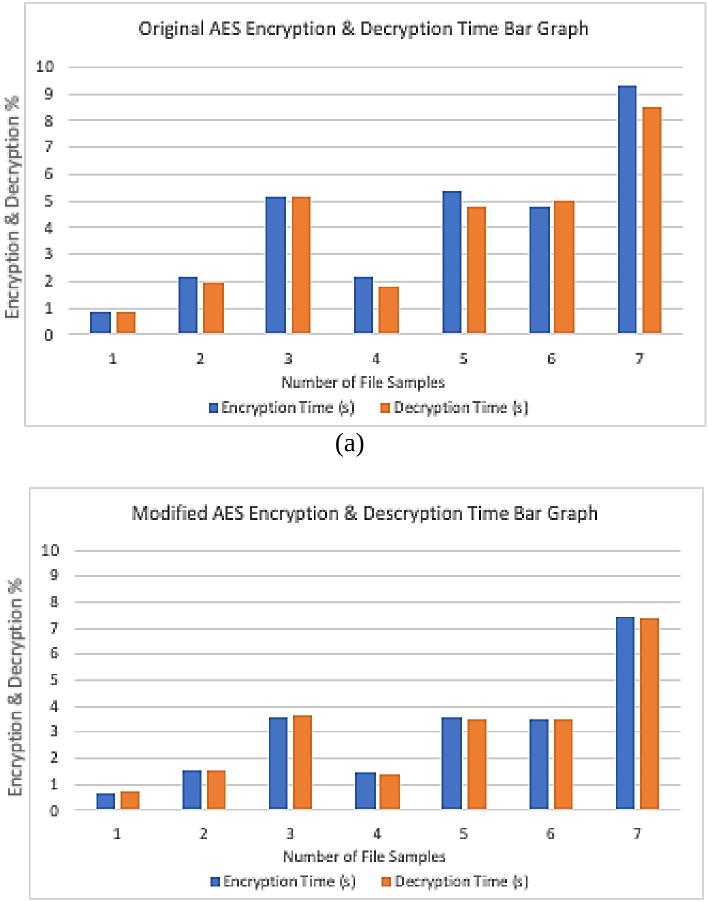
3.3. Time consumption of the original AES vs modified AES

In the performance comparison of the modified and original algorithms, Table 4 demonstrates that files of various sizes can effectively utilize the new algorithm. These percentages were calculated by taking the difference between the original and modified times, dividing by the original time, and then multiplying by 100. The modified AES performs faster than the original AES, as shown in the encryption and decryption time columns. The results indicate that the modified AES improved its encryption performance by 26.90% and its decryption performance by 22.73%. This clearly shows that, despite enhanced security, the algorithm's modifications also improved performance. Additionally, the observation indicates that as the file size increases, the performance of the modified algorithm holds steady, demonstrating the potential to be particularly effective for medium to larger files.

Table 4. Time consumption of conventional AES and modified AES

File type	File information	Size (MB)	Original AES		Modified AES	
			Encryption time (s)	Decryption time (s)	Encryption time (s)	Decryption time (s)
.txt	1000000 characters	1	0.86	0.87	0.68	0.70
.jpg	5072×6761 pixels	2.1	2.20	1.97	1.522	1.53
.jpg	5072×6761 pixels	5.3	5.15	5.16	3.57	3.64
.mp3	1:26 (min)	2	2.14	1.84	1.44	1.40
.mp3	3:37 (min)	5.2	5.37	4.78	3.54	3.52
.mp4	1280×720 resolution 30 (sec)	5.3	4.77	5.00	3.52	3.53
.mp4	1280×720 resolution 1:02 (min)	10.5	9.32	8.53	7.44	7.40

Figure 5(a) visualizes the time consumption of the original AES algorithm for both encryption and decryption across various file sizes and types, showing that as the file sizes increases, the time required for these processes also increase. While effective, the original AES requires a significant amount of time to process larger files, especially in complex formats like '.mp4'. Conversely, Figure 5(b) illustrates that the modified AES algorithm consistently outperforms the original AES in terms of speed, showing lower encryption and decryption times across all tested file sizes and types. The improved speed is particularly notable with larger files, where the modified AES shows a significant reduction in processing time compared to the original AES, making it more suitable for applications requiring faster data encryption and decryption.



(a)

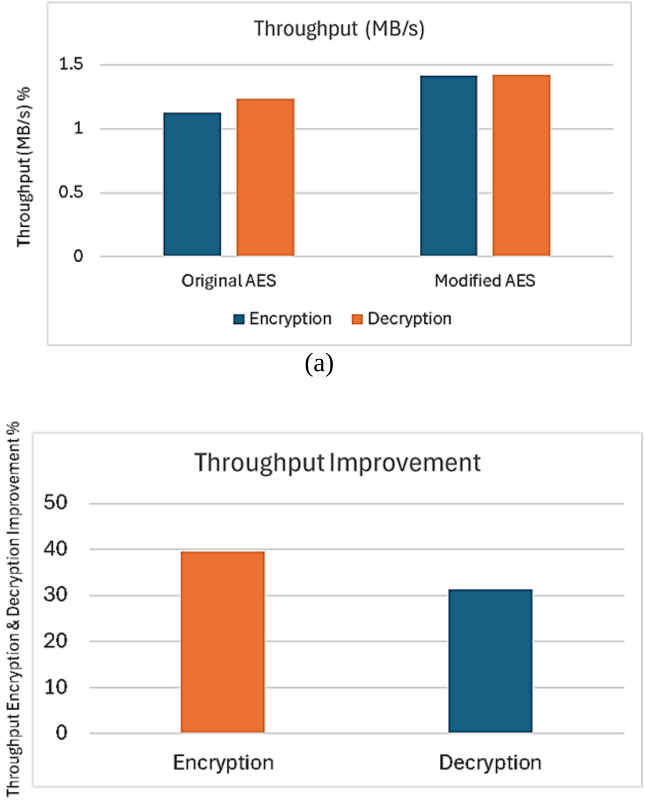
(b)

Figure 5. Comparison of encryption and decryption time performance; (a) original AES encryption and decryption time bar graph and (b) modified AES (PBKC-RRAES) encryption and decryption time bar graph

3.4. Throughput

In the evaluation of throughput as presented in Figure 6, the performance of the modified AES algorithm was compared to the standard AES algorithm. Figure 6(a) illustrates the direct throughput comparison in MB/s, showing both encryption and decryption performance for original AES and modified AES across all tested files. While the visual difference appears modest due to the scale, Figure 6(b) clearly demonstrates the significant percentage improvements achieved, with encryption throughput improving by 39.48% and decryption throughput by 31.27%. This metric provides a clear indication of the enhanced performance achieved through the modifications. Furthermore, the throughput of the modified AES is greater than the original, highlighting its performance in handling encryption and decryption tasks more rapidly. This improvement is particularly advantageous for applications requiring real-time data processing or handling large volumes of data, as it significantly reduces the time and computational resources needed for cryptographic operations. Additionally, the enhanced throughput suggests that the modified algorithm can sustain high performance even under heavy loads, making it a robust solution for modern encryption needs. Throughput was calculated by dividing the file size by the encryption or decryption time for each algorithm.

The percentage difference was computed using the formula (Modified AES–AES/AES) multiplied by 100. Calculating the improvement by Improvement (%)=[(Modified Throughput-Original Throughput)÷Original Throughput]×100.



(a)

(b)

Figure 6. Throughput performance analysis of PBKC-RRAES; (a) throughput comparison between original AES and Modified AES (MB/s) and (b) percentage improvement in throughput for encryption and decryption operations

3.5. Security features comparison

Table 5 presents a comprehensive comparison of security features between conventional AES and the PBKC-RRAES algorithm. The analysis demonstrates that PBKC-RRAES achieves superior security metrics while maintaining computational efficiency. Most notably, the algorithm exhibits a 15.4% improvement in avalanche effect (54.214% vs ~47%), indicating enhanced diffusion properties. Despite reducing rounds from 10 to 6, PBKC-RRAES maintains robust security through strategic enhancements including byte substitution with permutation in the key schedule, dual round key operations (AddRoundKey and ModAddRoundKey), and improved sensitivity to input changes (50.78-59.37% bit change). Furthermore, the replacement of MixColumns' O(n²) complexity with bit permutation's O(n) complexity significantly reduces computational overhead while preserving cryptographic strength, validating the algorithm's dual achievement of enhanced security and improved performance.

Table 5. Security features comparison between conventional AES and PBKC-RRAES

Security feature	Conventional AES	PBKC-RRAES	Enhancement
1. Avalanche effect	~47% average	54.214%	+15.4% improvement
2. Number of rounds	10 rounds	6 rounds	Optimized with enhanced diffusion
3. Key schedule	Standard expansion	Byte substitution+permutation before expansion	Enhanced key-dependent transformations
4. Round key operations	AddRoundKey only	AddRoundKey+ModAddRoundKey	Additional confusion layer
5. Sensitivity to input	Standard	Enhanced (50.78–59.37%-bit change)	Higher unpredictability
6. Computational complexity	O(n^2) for MixColumns	O(n) for bit permutation	Reduced complexity

3.6. Modified advanced encryption standard comparison

In Tables 6 and 7, we present a state-of-the-art comparison of the avalanche effect, throughput, and encryption/decryption rates for different modified AES algorithms. Due to the unique experimental setup of the study, direct comparisons with other works were not always feasible. Where data is unavailable or not applicable, or not applicable with the same system requirements for the experiment [28], we have indicated this. The presented values are normalized where possible to facilitate meaningful comparisons [29], and we provide a critical discussion of these findings in the subsequent sections.

Table 6. Comparison of different AES algorithms

Algorithm	Avalanche effect %	Throughput %	Encryption rate %	Decryption rate %	File sizes/type
PBKC-RRAES	54.21	43.07	26.90 (s)	22.73 (s)	Mixed medium-to-large
RRPBAES	52.92	31.12	76.76 (ms)	55.46 (ms)	Video files of large sizes
RRMAES	50.06	1.29	1.27 (ms)	1.21 (ms)	Small mixed files

Table 7. Different modified AES algorithm test environments

Algorithm	Test environments	Notes
PBKC-RRAES	Small, medium, and large files (MB)	Performs with enhanced security and medium-to-large files and all file types, emphasizing the benefits of its design for high-throughput scenarios.
RRPBAES [22]	Medium and large files (KB/MB)	Performs better on video files which influences its encryption and decryption rates to be higher, especially with the unique data patterns of video content.
RRMAES [25]	Small files (KB)	Optimized for smaller files with faster processing speeds but tailored to different data sizes.

The comparison of different modified AES algorithms reveals notable improvements in encryption and decryption performance. PBKC-RRAES, with a throughput of 39.48%, shows enhanced encryption speed, making it ideal for mixed medium-to-large files, while its encryption and decryption rates (26.90% and 22.73%) highlight its efficiency in secure data processing while maintaining enhanced security in higher level. RRPBAES excels in handling large video files, with an impressive 76.76% encryption rate and a 55.46% decryption rate, making it suitable for real-time applications. RRMAES, though having lower throughput, is optimized for small files, offering secure processing tailored to smaller data sizes. The avalanche effect across all algorithms indicates high sensitivity to input changes, ensuring robust security. Each algorithm is best suited for specific file types and environments, reflecting a balance between speed and security based on the intended use case. While efficiency is a relevant factor in evaluating algorithm performance, it was not within the scope of this study, which focused specifically on assessing the speed of encryption and decryption, as well as security enhancements. Future research will aim to address efficiency considerations, providing a more comprehensive evaluation of these modified AES algorithms.

4. CONCLUSION

The PBKC-RRAES algorithm proposed in this study significantly enhances both security and performance compared to the standard AES. By innovating within the standard AES framework, replacing the MixColumns function with a bit permutation technique, incorporating additional AddRoundKey operations between cipher states, adding byte substitution operations, and round constant additions in the key schedule algorithm before key expansion, and reducing the rounds from 10 to 6, PBKC-RRAES achieves a 54.214% avalanche effect (exceeding the standard 50% threshold), 26.90% faster encryption, and 22.73% faster decryption, with throughput improvements of 39.48% and 31.27% respectively. These modifications not only strengthen cryptographic security but also eliminate computational bottlenecks, making the algorithm a robust and efficient alternative for securing various data types and sizes.

The findings suggest that PBKC-RRAES can serve as a highly effective cryptographic solution, particularly in applications requiring a balance of high security and performance. The improvements in both security measures and processing speed underscore its suitability for real-time data encryption, cloud storage, and other security-critical environments.

While PBKC-RRAES demonstrates significant improvements, certain limitations merit consideration. The algorithm requires thorough validation against comprehensive cryptanalytic attacks beyond the scope of this study. Hardware implementation optimization remains unexplored, particularly for FPGA and ASIC deployments where the bit permutation operations could potentially achieve even greater performance gains. Additionally, integration with existing cryptographic protocols and standards requires further investigation to ensure seamless adoption in production systems.

Future research should address these limitations by focusing on precomputing permutation tables and S-box values to expedite the encryption and decryption processes. Additionally, exploring parallelization techniques, such as distributing bit permutation and key schedule operations across multiple processing cores, could significantly enhance performance, especially in large-scale data encryption scenarios. Hardware-specific implementations and formal security proofs would further validate PBKC-RRAES for deployment in critical infrastructure. The demonstrated combination of enhanced security metrics and substantial performance improvements positions PBKC-RRAES as a compelling evolution of AES for next-generation cryptographic applications. As organizations face increasingly sophisticated threats while demanding higher processing efficiency, PBKC-RRAES offers a practical solution that addresses both requirements without compromise, potentially influencing future standardization efforts in symmetric key cryptography.

FUNDING INFORMATION

Publication funding by Tarlac State University.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Jerico S. Baladhay	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓
Alvincent E. Danganan		✓		✓	✓	✓		✓		✓	✓	✓		✓
Edjie M. De Los Reyes	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	✓
Heidilyn V. Gamido	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
Marlon V. Gamido	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis

I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting

Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [JSB], upon reasonable request.

REFERENCES

[1] D. Awschalom *et al.*, "Development of Quantum Interconnects (QuICs) for Next-Generation Information Technologies," *PRX Quantum*, vol. 2, no. 1, pp. 1-21, Feb. 2021, doi: 10.1103/PRXQuantum.2.017002.

[2] G. Aceto, V. Persico, and A. Pescapé, "A Survey on Information and Communication Technologies for Industry 4.0: State-of-the-Art, Taxonomies, Perspectives, and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 4, pp. 3467–3501, 2019, doi: 10.1109/COMST.2019.2938259.

[3] X. Chen, K. Gatsis, H. Hassani, and S. S. Bidokhti, "Age of Information in Random Access Channels," *IEEE Transactions on Information Theory*, vol. 68, no. 10, pp. 6548–6568, Oct. 2022, doi: 10.1109/TIT.2022.3180965.

[4] H. M. Ghadirli, A. Nodehi, and R. Enayatifar, "An overview of encryption algorithms in color images," *Signal Processing*, vol. 164, pp. 163–185, Nov. 2019, doi: 10.1016/j.sigpro.2019.06.010.

[5] C. Tiken and R. Samli, "A Comprehensive Review About Image Encryption Methods," *Harran Üniversitesi Mühendislik Dergisi*, vol. 7, no. 1, pp. 27–49, Apr. 2022, doi: 10.46578/humder.1066545.

[6] N. Pundir, S. Aftabjahani, R. Cammarota, M. Tehranipoor, and F. Farahmandi, "Analyzing Security Vulnerabilities Induced by High-level Synthesis," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 18, no. 3, pp. 1–22, Jul. 2022, doi: 10.1145/3492345.

[7] K. Wang, Y. Yan, and C. Zhu, "Exploiting wavelet transform and support vector machine algorithm to perform side channel attacks on advanced encryption standard (AES)," in *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, Dec. 2019, pp. 1–7, doi: 10.1145/3371425.3371447.

[8] D. Ramakrishna and M. A. Shaik, “A Comprehensive Analysis of Cryptographic Algorithms: Evaluating Security, Efficiency, and Future Challenges,” *IEEE Access*, vol. 13, pp. 11576–11593, 2025, doi: 10.1109/ACCESS.2024.3518533.

[9] R. Pitale, K. Tajane, P. Mahajan, N. Nehate, A. Mulimani, and D. Lokhande, “Cryptographic algorithm development and application for encryption and decryption,” in *Proceedings of the 5th International Conference on Information Management & Machine Intelligence*, 2023, pp. 1–7, Nov. doi: 10.1145/3647444.3647853.

[10] L. E. Kane, J. J. Chen, R. Thomas, V. Liu, and M. McKague, “Security and Performance in IoT: A Balancing Act,” *IEEE Access*, vol. 8, pp. 121969–121986, 2020, doi: 10.1109/ACCESS.2020.3007536.

[11] S. Sheikhpour, A. Mahani, and N. Bagheri, “Reliable advanced encryption standard hardware implementation: 32- bit and 64-bit data-paths,” *Microprocessors and Microsystems*, vol. 81, p. 103740, Mar. 2021, doi: 10.1016/j.micpro.2020.103740.

[12] N. Mouha, “Review of the advanced encryption standard,” National Institute of Standards and Technology (U.S.), Gaithersburg, MD, NIST IR 8319, Jul. 2021, doi: 10.6028/NIST.IR.8319.

[13] S. Devi and H. D. Kotha, “AES encryption and decryption standards,” *Journal of Physics: Conference Series*, vol. 1228, no. 1, pp. 1-11, May. 2019, doi: 10.1088/1742-6596/1228/1/012006.

[14] D. Punia and B. Singh, “Speed optimization of the AES algorithm using pipeline hardware architecture,” in *Proceedings of the 4th International Conference on Communication and Electronics Systems, ICCES 2019*, Coimbatore, India, Jul. 2019, pp. 2070–2074, doi: 10.1109/ICCES45898.2019.9002086.

[15] J. Kaur, S. Lamba, and P. Saini, “Advanced encryption standard: attacks and current research trends,” in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2021*, Greater Noida, India, Mar. 2021, pp. 112–116, doi: 10.1109/ICACITE51222.2021.9404716.

[16] L. Giner *et al.*, “Generic and Automated Drive-by GPU Cache Attacks from the Browser,” in *ASIA CCS '24: Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, 2024, pp. 128–140, doi: 10.1145/3634737.3656283.

[17] B. Sarkar, A. Saha, D. Dutta, G. De Sarkar, and K. Karmakar, “A Survey on the Advanced Encryption Standard (AES): A Pillar of Modern Cryptography,” *International Journal of Computer Science and Mobile Computing*, vol. 13, no. 4, pp. 68–87, Apr. 2024, doi: 10.47760/ijcsmc.2024.v13i04.008.

[18] M. Ruan, H. Gao, and Y. Wang, “Secure and privacy-preserving consensus,” *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4035–4049, Oct. 2019, doi: 10.1109/TAC.2019.2890887.

[19] M. Bedoui, H. Mestiri, B. Bouallegue, B. Hamdi, and M. Machhout, “An improvement of both security and reliability for AES implementations,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 9844–9851, Nov. 2022, doi: 10.1016/j.jksuci.2021.12.012.

[20] N. G. Zinabu and S. Asferaw, “Enhanced Security of Advanced Encryption Standard (ES-AES) Algorithm,” *American Journal of Computer Science and Technology*, vol. 5, no. 2, pp. 41-48, 2022, doi: 10.11648/j.ajcst.20220502.13.

[21] H. V. Gamido, A. M. Sison, and R. P. Medina, “Modified AES for text and image encryption,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 3, pp. 942–948, 2018, doi: 10.11591/ijeecs.v11.i3.pp942-948.

[22] J. S. Baladhay and E. M. D. L. Reyes, “AES-128 reduced-round permutation by replacing the MixColumns function,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 3, pp. 1641–1652, Mar. 2024, doi: 10.11591/ijeecs.v33.i3.pp1641-1652.

[23] M. K. Yang and J.-S. Jeong, “Optimized Hybrid Central Processing Unit–Graphics Processing Unit Workflow for Accelerating Advanced Encryption Standard Encryption: Performance Evaluation and Computational Modeling,” *Applied Sciences*, vol. 15, no, 7, 2025, doi: 10.3390/app15073863.

[24] E. M. D. L. Reyes, A. M. Sison, and R. P. Medina, “Modified AES cipher round and key schedule,” *Indonesian Journal of Electrical Engineering and Informatics*, vol. 7, no. 1, pp. 28–35, Mar. 2019, doi: 10.11591/ijeei.v7i1.652.

[25] E. M. D. L. Reyes, A. M. Sison, and R. P. Medina, “File encryption based on reduced-round AES with revised round keys and key schedule,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 897–905, Nov. 2019, doi: 10.11591/ijeecs.v16.i2.pp897-905.

[26] K. I. D. Kyriakou and N. D. Tselikas, “Complementing JavaScript in High-Performance Node.js and Web Applications with Rust and WebAssembly,” *Electronics*, vol. 11, no. 19, pp. 1-17, Oct. 2022, doi: 10.3390/electronics11193217.

[27] J. Kaur and K. R. R. Kumar, “Analysis of Avalanche effect in Cryptographic Algorithms,” in *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2022*, pp. 1–4, Oct. 2022, doi: 10.1109/ICRITO56286.2022.9965127.

[28] B. E. H. H. Hamouda, “Comparative Study of Different Cryptographic Algorithms,” *Journal of Information Security*, vol. 11, no. 03, pp. 138–148, 2020, doi: 10.4236/jis.2020.113009.

[29] I. Ogundoyin, “Comparative Analysis and Performance Evaluation of Cryptographic Algorithms,” *UNIOSUN Journal of Engineering and Environmental Sciences*, vol. 4, no. 1, Mar. 2022, doi: 10.36108/ujees/2202.40.0140.

BIOGRAPHIES OF AUTHORS

Jerico S. Baladhay is a dedicated researcher who has been actively involved in the field since 2020 and also a faculty member at Tarlac State University under the Department of College of Computer Studies. He holds a Master's degree in Information Technology at Tarlac State University. His expertise includes software development, specializing in web applications. Additionally, he is a law student at Tarlac State University, actively engaging in the intersection of legal studies and technology within this prestigious institution. Significantly, his scholarly endeavors have manifested in notable contributions to the domain, elucidated through the publication of scientific papers within the realms of computer science, information technology, and the intricate domain of data security and encryption algorithms. His scholarly pursuits are distinctly anchored in the investigation of data security paradigms and the development of encryption algorithms. He can be contacted at email: jsbaladhay@tsu.edu.ph.

Dr. Alvincent E. Danganan has been a faculty member at Tarlac State University's College of Computer Studies in the Philippines since 2003. From 2013 to 2016, he was the Chairperson of the Department of Information Systems and also served as the Chairperson of the College Extension Service during the same period, managing projects and presentations that received institutional recognition. Additionally, he was the Chairperson of the Computer Science Department from 2019 to 2020. He is currently the Dean of the College of Computer Studies at Tarlac State University. He also serves as one of the area coordinators for the Central Luzon chapter of the Philippines Society of Information Technology Educators. His research focuses on data mining, machine learning, and data analytics, and he has published articles on data mining in Scopus-indexed journals. He can be contacted at email: avdanganan@tsu.edu.ph.

Dr. Edjie M. De Los Reyes currently holds the esteemed position of Associate Professor IV at Tarlac State University, boasting a dedicated service of two decades in academia. With a strong academic background, previously serving as Associate Dean from 2014 to 2016. His extensive expertise is evidenced by a variety of certifications, including Cisco Certified Network Associate, Cisco Certified Academic Instructor, Microsoft Office Specialist, and Electronic Data Processing Specialist-Programmer. Actively engaged in academic and research circles, he is a respected member of esteemed organizations such as the Philippine Society of Information Technology Educators (PSITE), Philippine Schools Universities and Colleges Computer Education and Systems Society (PSUCCESS), and the International Association of Multidisciplinary Research. Moreover, he has made significant contributions to the academic community, publishing numerous research papers in Scopus-indexed journals, with a notable focus on data security. He can be contacted at email: emdelosreyes@tsu.edu.ph.

Dr. Heidilyn V. Gamido holds a doctorate in information technology from the Technological Institute of the Philippines, Quezon City, achieved through the CHED K-12 transition program scholarship. In 2006, she earned her Masters of Engineering with a major in Information and Communications from Pai Chai University in Daejeon, South Korea, where she studied on a scholarship. Prior to this, she completed her Bachelor of Science in Information Technology at Saint Louis University in Baguio City, Philippines, graduating in 2002. Currently, she serves as a Professor IV within the College of Computer Studies at Tarlac State University, while also fulfilling the role of Director at the Management Information Systems Office since 2014. Alongside her academic accomplishments, she has acquired skills certifications such as MOS, ICDL, and Network Security Associate. Her research interests span across diverse areas including data security, image processing, and information systems. She can be contacted at email: htvgamido@tsu.edu.ph.

Dr. Marlon V. Gamido bringing over 25 years of academic expertise, he currently holds the position of Associate Professor V and Director of the Management Information Systems Office at Tarlac State University. He obtained a Master of Science in Information Technology from Hannam University in Daejeon, Korea, supported by a CHED Scholarship. His leadership roles include serving as the Vice President for Administration and Finance from 2020 to 2022, Dean of the College of Computer Studies from 2014 to 2020, and Director of Management Information Systems from 2005 to 2014 and again from 2016 to 2019. Notably, he represented the Philippines at the United Nations Asian and Pacific Training Center for ICT for Development (UN APCICT/ESCAP). Additionally, he is a licensed electrical engineer and an active participant in the Tarlac ICT Council. His research interests span across various areas including ICT for development, educational technologies, IT audit, enterprise architecture, and project management. He can be contacted at email: mvgamido@tsu.edu.ph.