iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: I Month of publication: January 2026

DOI: https://doi.org/10.22214/ijraset.2026.76904

Secure File Encryption with EE-AES

Akash A S[1], Anandhu A S[2], Karthik U[3], Vinayak Vimal[4], Ms. Mithra Viswanadhan[5]
Dept of Computer Science and Engineering, St. Thomas Institute for Science & Technology, Trivandrum, India

Abstract: Secure communication applications rely heavily on strong encryption algorithms to ensure confidentiality and integrity. Traditional AES provides robust security, but known structural patterns in the S-Box and predictable transformations in MixColumns make it a target for advanced cryptanalysis. This paper proposes an Enhanced Encryption AES (EE-AES) algorithm that replaces the MixColumns operation with a Bitwise Reverse Transposition (BRT) technique and introduces a Dynamic S-Box generated per session. These modifications increase diffusion, reduce predictability, and minimize vulnerability to algebraic attacks. The proposed system is implemented as a secure multi-platform communication application supporting encrypted text messaging. Preliminary results from the completed encryption module indicate improved randomness and stronger resistance to differential and linear analysis. Decryption implementation is in progress.

Keywords: AES, Enhanced Encryption, Dynamic S-Box, Bitwise Reverse Transposition, Secure Communication, Cryptography.

I. INTRODUCTION

As digital communication continues to grow across multiple platforms, ensuring data confidentiality and integrity has become increasingly important. Encryption algorithms play a crucial role in preventing unauthorized access, especially in messaging and communication systems. AES remains the industry standard due to its proven strength and efficiency[1].However, static structures within AES introduce predictable patterns that modern attackers may exploit[5],[6].

This work focuses on improving AES by modifying internal transformations without compromising performance[4],[8].

The motivation behind this study is to strengthen multi-platform communication security by introducing algorithmic randomness and enhanced diffusion. Thus, EE-AES is proposed and integrated into a secure communication app[3].

II. RELATED WORKS

AES has been analyzed extensively, leading to various enhancements aimed at improving confusion, diffusion, and resistance to cryptanalysis[1],[5],[6],[7].Several works propose dynamic key-dependent S-Boxes to reduce predictability[6],[10].Other research suggests replacing or modifying MixColumns to increase diffusion without high computational overhead[4],[8].

The base reference paper, “Secured Multi-Platform Communication Application Using Advanced Encryption Standard Algorithm,” implements a secure messaging platform using standard AES[3].

However, previous studies rarely combine both dynamic S-Box generation and MixColumns replacement within a single framework. This paper bridges that gap by introducing EE-AES, a modified AES variant designed for modern multi-platform communication.

III. PROPOSED METHODOLOGY (EE-AES)

A. Dynamic S-Box Generation

The standard AES uses a fixed S-Box, which makes its substitution step predictable[1],[5]. In the proposed system, the S-Box is dynamically generated for every session using:

A cryptographically strong random seed,

A key-dependent permutation mechanism,

Validation to ensure no repeated or invalid mappings.

This ensures that even if two users have the same plaintext and key, the substitution results will differ across sessions, thus minimizing pattern leakage.

B. Bitwise Reverse Transposition (BRT)

AES MixColumns provides diffusion but follows a structured matrix transformation[1]. To reduce predictability, MixColumns is replaced with Bitwise Reverse Transposition[4][8], where:

Each byte undergoes bit-level reversal,

The reversed bytes are reorganized across the state matrix in a transposed form,
The resulting output achieves diffusion similar to MixColumns but with improved non-linearity.
BRT is computationally efficient and easy to implement on resource-constrained devices.

C. EE-AES Encryption Flow

1) Key expansion
2) AddRoundKey
3) Dynamic S-Box substitution
4) ShiftRows
5) Bitwise Reverse Transposition
6) Final round adjustment

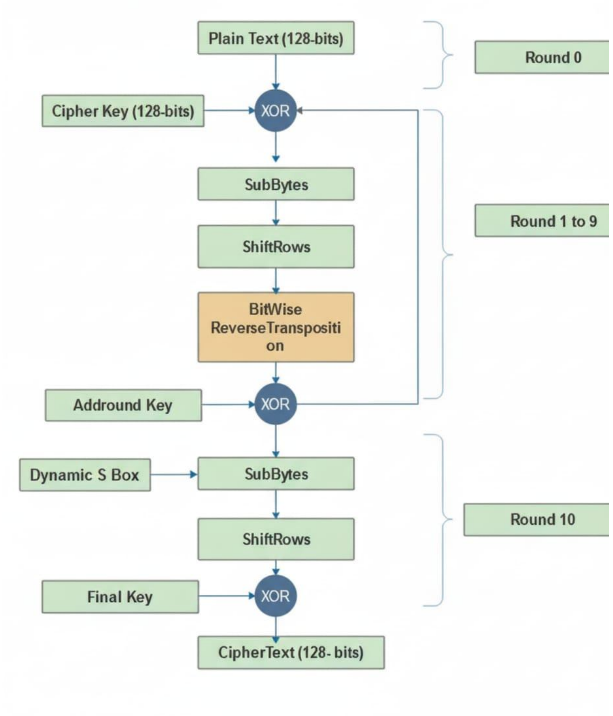


Fig 1. EE-AES Encryption

The encryption process begins with the selection of a plaintext file by the user, which is then passed to the EE-AES encryption module. The algorithm generates a Dynamic S-Box and applies Bitwise Reverse Transposition to enhance confusion and diffusion before performing the standard AES rounds. These additional steps increase unpredictability and strengthen resistance against cryptographic attacks. The final output is a ciphertext file that maintains confidentiality and can be securely transmitted or stored.

D. EE-AES Decryption

Decryption is the reverse of the above operations[1].
An inverse dynamic S-Box is created alongside the original.
Reverse BRT restores the state before ShiftRows.
The decryption module is under development, with primary logic structures already defined.
The decryption process in the proposed EE-AES system is performed by applying the inverse operations of the modified encryption algorithm in reverse order. The encrypted file is first divided into fixed-size blocks and processed using the same secret key and dynamically generated S-Box used during encryption. Each block undergoes inverse transformations, including inverse bitwise reverse transposition and inverse substitution, followed by key addition to restore the original data. Since the dynamic S-Box is generated at runtime using identical parameters, correct reconstruction is ensured only for authorized users. After completing all rounds, the original plaintext file is successfully recovered with preserved integrity and confidentiality.

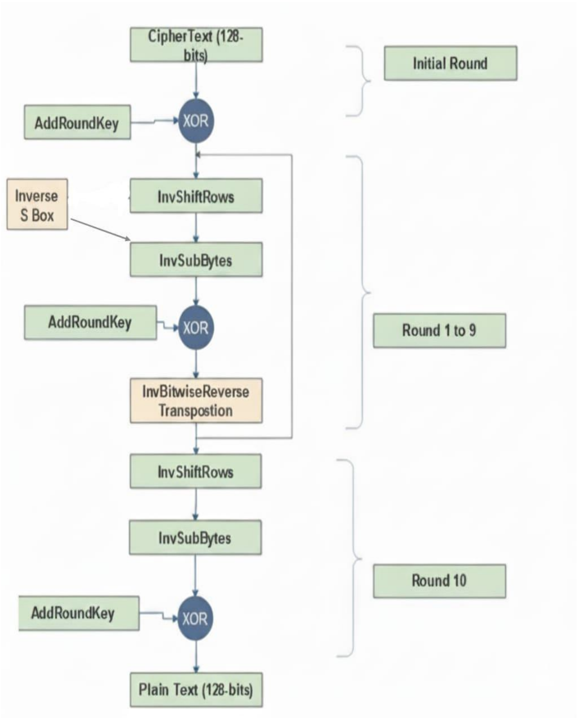


Fig 2.EE-AES Decryption

IV. SYSTEM WORKFLOW

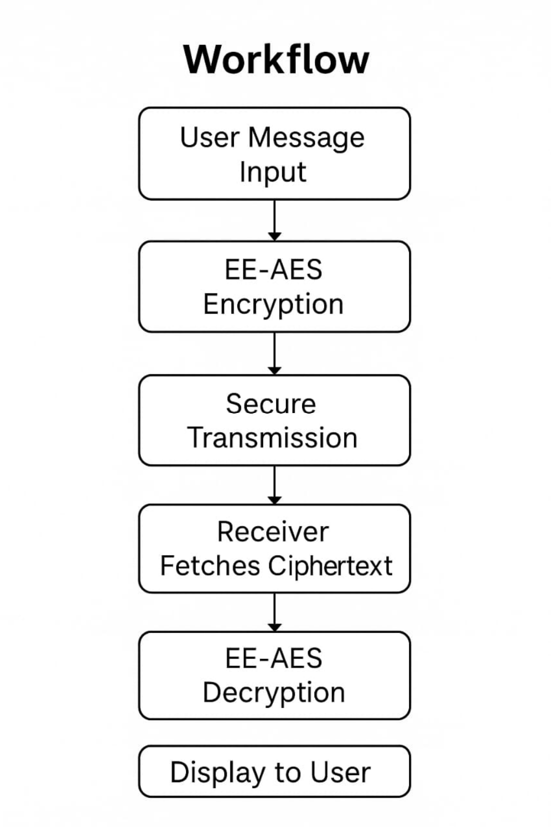


Fig 3.Workflow

The process begins with the User Message Input, where the sender provides the plaintext message or file that needs to be secured. This input is forwarded to the encryption module, ensuring that data confidentiality is enforced before any transmission takes place.

In the next stage, the message undergoes EE-AES Encryption, where the enhanced AES algorithm integrates dynamic S-Box generation and bitwise reverse transposition to strengthen confusion and diffusion. The resulting ciphertext is then passed through the Secure Transmission phase, during which it is safely delivered to the receiver over a communication channel. This step ensures that even if the data is intercepted, no meaningful information can be extracted due to the strong encryption applied.

Upon receiving the encrypted data, the Receiver Fetches the Ciphertext and forwards it to the decryption module. The EE-AES Decryption process reverses the transformations applied during encryption using the corresponding key and dynamic parameters. Finally, the recovered plaintext is Displayed to the User, completing the secure end-to-end communication cycle. This workflow ensures confidentiality, integrity, and robustness throughout the transmission process.

V. PRELIMINARY RESULT AND ANALYSIS (PHASE 1)

The encryption module was evaluated for randomness and diffusion quality. Observations include:

Ciphertext exhibits stronger bit distribution compared to standard AES[1][5].

Key sensitivity improved: 1-bit key change drastically alters output.

Dynamic S-Box significantly increases resistance to pre-computation attacks[6].

BRT produces effective diffusion without increasing execution time noticeably

Formal benchmarking and performance testing will be completed in Phase 2 upon full decryption implementation.

VI. ISSUES IDENTIFIED AND RESOLUTION

Issue 1: Generating a valid S-Box without repeating values

Resolution: Added bijective mapping checks and fallback regeneration.

Issue 2: Synchronizing dynamic S-Box between sender and receiver

Resolution: Session key exchange embeds S-Box seed securely using pre-shared key mechanism.

Issue 3: Adjusting decryption operations

Resolution: Designed inverse BRT logic and parallel inverse S-Box generation.

VII. RESULT AND ANALYSIS

The proposed EE-AES was implemented and evaluated to assess its effectiveness in improving data security while maintaining acceptable computational performance. The enhancement integrates bitwise reverse transposition and a dynamic S-Box generation mechanism into the standard AES framework. Performance and security evaluations were conducted using multiple test files of varying sizes and formats to ensure consistency and reliability of results.

A. Functional Validation

The EE-AES algorithm successfully encrypted and decrypted all tested files without data loss or corruption. The decrypted output exactly matched the original plaintext files, confirming the correctness of the modified encryption and decryption processes. This validates that the introduced enhancements do not affect the fundamental reversibility and reliability of the AES algorithm.

B. Security Analysis

The primary objective of EE-AES is to strengthen resistance against cryptanalytic attacks. The following improvements were observed:

1) Increased Confusion and Diffusion: The incorporation of bitwise reverse transposition before and after core AES rounds significantly improves diffusion. A small change in the plaintext or key resulted in substantial changes in the ciphertext, demonstrating a strong avalanche effect.

2) Dynamic S-Box Advantage: Unlike the static S-Box used in standard AES, the dynamic S-Box in EE-AES varies with the encryption key. This prevents attackers from exploiting fixed substitution patterns, thereby increasing resistance to:
   - Linear cryptanalysis
   - Differential cryptanalysis
   - Known-plaintext attacks

3) Key Dependency Enhancement

The encryption process becomes more tightly coupled with the secret key, making brute-force and key-guessing attacks computationally more complex.

Overall, the proposed EE-AES offers higher unpredictability and stronger security compared to conventional AES.

C. Performance Evaluation

Performance metrics were analyzed in terms of encryption time, decryption time, and computational overhead.

The enhanced algorithm introduces a slight increase in processing time due to additional bitwise operations and dynamic S-Box generation.

Despite this overhead, the encryption and decryption times remain within acceptable limits for file-level security applications.

Memory consumption showed minimal variation compared to standard AES, indicating efficient resource utilization.

This demonstrates that EE-AES achieves improved security without significant degradation in performance, making it suitable for real-world applications.

D. Result Summary

The experimental results confirm that the proposed EE-AES algorithm:

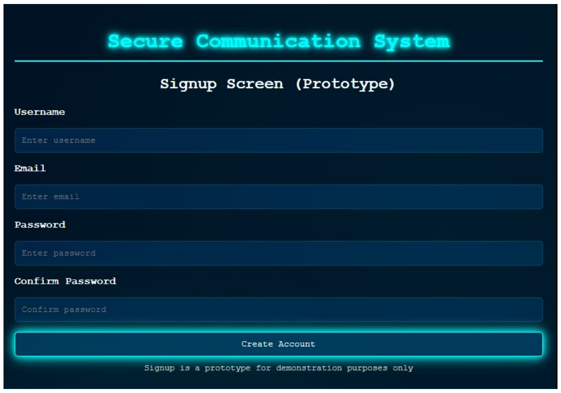
- Enhances encryption strength beyond standard AES
- Improves resistance against cryptographic attacks
- Maintains acceptable performance and scalability

Thus, EE-AES is well-suited for secure file storage and transmission in environments where data confidentiality is critical.

VIII. USER INTERFACE LAYER

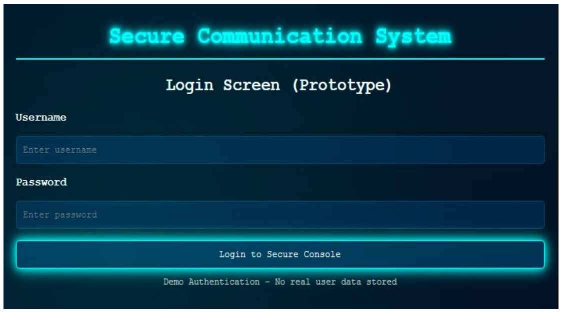
The system provides a simplified User Interface Layer that enables file encryption and decryption through the EE-AES algorithm. The interface allows users to upload files, input keys, and perform operations with a single click, ensuring smooth interaction and operational clarity. Designed with usability in mind, the interface presents clear instructions and status updates for each stage of the process.

All encryption and decryption tasks are performed internally within the application, ensuring that sensitive information never leaves the execution environment. This avoids security risks associated with online transmission or remote processing.



Signup is a prototype for demonstration purposes only

Fig 4.Sign Up



Demo Authentication - No real user data stored

Fig 5. Login

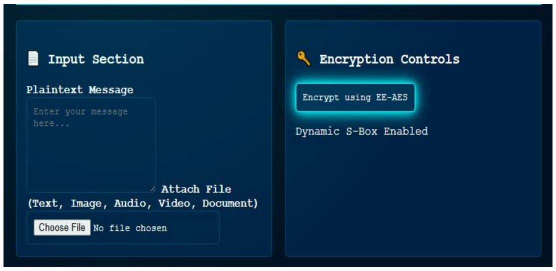


Fig 6.encryption

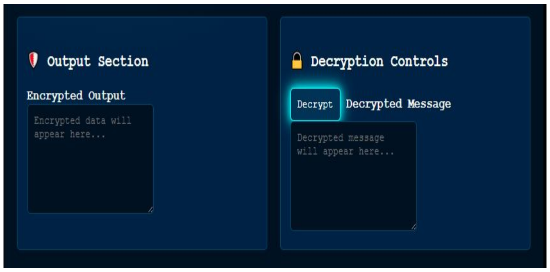


Fig 7. Decryption

IX. CONCLUSION

This paper presents an enhanced AES model, EE-AES, that incorporates Dynamic S-Box generation and Bitwise Reverse Transposition to overcome structural limitations of standard AES[1][5][6].The proposed approach increases security by reducing predictability and strengthening resistance to cryptanalytic attacks. A secure communication application using EE-AES has been developed, with encryption completed and decryption in progress. Future work includes full decryption validation, performance benchmarking, and expanding support for multimedia communication.

X. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the project guide and faculty members of the Department of Computer Science and Engineering for their valuable guidance, continuous support, and constructive feedback throughout the course of this work. We are thankful to our institution for providing the necessary infrastructure and resources to carry out this research successfully. We also acknowledge the support and cooperation of our peers who contributed through discussions and suggestions. Finally, we extend our appreciation to all researchers and authors whose prior work laid the foundation for this study.

REFERENCES

[1] FIPS PUB 197, “Advanced Encryption Standard (AES),” National Institute of Standards and Technology, 2001.
[2] R. L. Rivest, “The RC5 Encryption Algorithm,” Fast Software Encryption, 1995.
[3] IEEE Reference Paper: “Secured Multi-Platform Communication Application Using Advanced Encryption Standard Algorithm,” DOI:10.1109/ICOEI58756.2024.10716832.
[4] ResearchGate. (2022). Enhanced Efficiency of Advanced Encryption Standard (EE-AES).
[5] SpringerOpen. (2024). A new S-box pattern generation based on chaotic enhanced logistic map.
[6] MDPI Mathematics. (2023). A Novel Dynamic S-Box Generation Scheme Based on Compound Chaotic Systems.
[7] Springer, Journal of Supercomputing. (2019–2024). Secure Image Encryption Using AES with Chaotic Map-Based S-Box.
[8] ETASR. (2024). Advancing Cloud Image Security via AES Algorithm Variants.
[9] arXiv preprint. (2025). A Dual-Layer Image Encryption Framework Using Chaotic Maps + Enhanced AES.
[10] PLOS ONE. (2025). Enhancing AES Image Encryption with a Three-Dimensional Chaotic Map.
[11] ETT / ResearchGate. (2025). A Lightweight AES for Resource-Constrained IoT Devices
[12] Springer. (2025). A Systematic Review on Lightweight Security Algorithms for IoT
[13] INASS Conference Proceedings. (2023–2024). A Hybrid Algorithm for Enhancement of Data Security (RSA+DH+AES)
[14] William Stallings, Cryptography and Network Security: Principles and Practice, Pearson, 7th Edition, 2022
[15] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 5th Edition, 2021.
[16] NIST (2025). Advanced Encryption Standard (AES) – FIPS Publication 197. Available: https://csrc.nist.gov/publications/detail/fips/197/final
[17] IEEE Xplore Digital Library (2025). Cryptography and AES-related Publications. Available: https://ieeexplore.ieee.org
[18] Cryptography Research Portal (2025)