iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

# ENHANCED IMPLEMENTATION AND PERFORMANCE BENCHMARKING OF AES FOR SECURE DATA ENCRYPTION

**SATHEESH BANDI**
**B RAVITEJA**
**KATHA SRAVANI**
UG Student, J.B Institute of Engineering and Technology, Moinabad, RangaReddy, Telangana

**ABSTRACT**
With the increasing need for secure data transmission in computer networks, numerous cryptographic algorithms have been developed. The National Institute of Standards and Technology (NIST) introduced the Advanced Encryption Standard (AES) as a robust security standard to protect electronic data. Its specifications are outlined in Publication 197 of the Federal Information Processing Standards (FIPS). AES operates on 128-bit data blocks and utilizes 128-bit cipher keys (AES-128) for encryption and decryption.
This paper proposes an optimized AES-128 encryption model where round keys are dynamically generated alongside the encryption process. This unique pipelined design minimizes encryption delay for each round, significantly reducing the overall processing time of a plaintext block. As a result, the encryption throughput is improved, enhancing communication security and efficiency.

## INTRODUCTION

The Advanced Encryption Standard (AES) is a widely used encryption protocol in the United States, established in November 2001 under the Federal Information Processing Standard (FIPS) 197 and officially adopted as a federal standard in May 2002. Among the four encryption algorithms approved for government use, AES is the most recent. It differs from RSA, another popular encryption technique, as they belong to distinct cryptographic categories. Unlike RSA, which is primarily used for digital signatures and key exchange, AES is a symmetric encryption algorithm that processes data in 128-bit blocks for both encryption and decryption.
In binary terms, unlike decimal digits that range from 0 to 9, each bit in AES encryption can hold one of two values—0 or 1. The AES algorithm converts a 128-bit input block into an encrypted output block of the same size using a predefined encryption key. Since the same key is utilized for both encryption and decryption, AES is classified as a symmetric encryption method, ensuring secure and efficient data transmission.

.

## LITERATURE SURVEY

This survey explores advancements in the **Advanced Encryption Standard (AES)**, focusing on aspects such as **low power consumption, high security, enhanced performance, and improved efficiency**. Additionally, the feasibility of implementing AES in **VLSI environments** has been extensively studied.

**AES Algorithm and Its Security Enhancements**

The **National Institute of Standards and Technology (NIST)** introduced AES as a secure cryptographic standard. Multiple studies have analyzed its efficiency across different **modes of operation**, including **ECB, CBC, CFB, and OFB**, with comparisons to the **Triple DES (TDEA) algorithm**. These modes are designed to ensure data integrity by applying inverse cryptographic functions.

**Optimized Implementation of AES**

Researchers such as **Francois-Xavier Standaert, Gael Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat (2004)** examined the efficient implementation of **Rijndael Encryption** in **FPGA-based systems**. Their study introduced methodologies for optimizing **substitution boxes (S-boxes), key addition, and diffusion layers**, improving AES execution in hardware-based environments.
Similarly, **Farhadian A. and Aref M.R. (2009)** proposed an innovative approach to **S-box simplification and approximation**. Their technique utilized **power functions over finite fields**, streamlining cryptanalysis while maintaining encryption robustness. This approach has been successfully applied in AES-based encryption models like **Camellia and Shark**.

**Comparing AES with Other Symmetric Encryption Algorithms**

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

The study conducted by **Ramesh Babu, George Abraham, and Kiransinh Borasia (2012)** examined the role of symmetric key cryptography in securing **distributed systems**. Their comparative analysis of **DES and AES** concluded that AES outperforms **DES** in terms of **key length, block size, and security strength**, making it the preferred choice for modern encryption.

Additionally, **Ross Anderson, Eli Biham, and Lars Knudsen (1999)** proposed a **highly secure and efficient AES design**. Their research suggested the use of **128-bit block sizes and 256-bit keys**, achieving performance comparable to DES while offering enhanced security. Their work also explored alternative transformations such as **bitwise permutations and XOR substitutions**, leading to more robust encryption algorithms.

**The AES Encryption Process**

AES operates as a **symmetric key encryption algorithm**, processing **128-bit plaintext blocks** with three key length variations: **128, 192, and 256 bits**. The encryption process follows multiple iterations, where the number of rounds is determined by the key length—**10, 12, or 14 rounds** for **128, 192, and 256-bit keys**, respectively. The four fundamental transformation steps in AES include:

- **Sub Bytes:** A non-linear byte substitution operation.
- **Shift Rows:** A permutation process that shifts row data.
- **Mix Columns:** A transformation that strengthens diffusion.
- **Add Round Key:** The final stage where the generated round key is added to the data block.

This structured approach ensures high security and makes AES one of the most reliable encryption standards today.

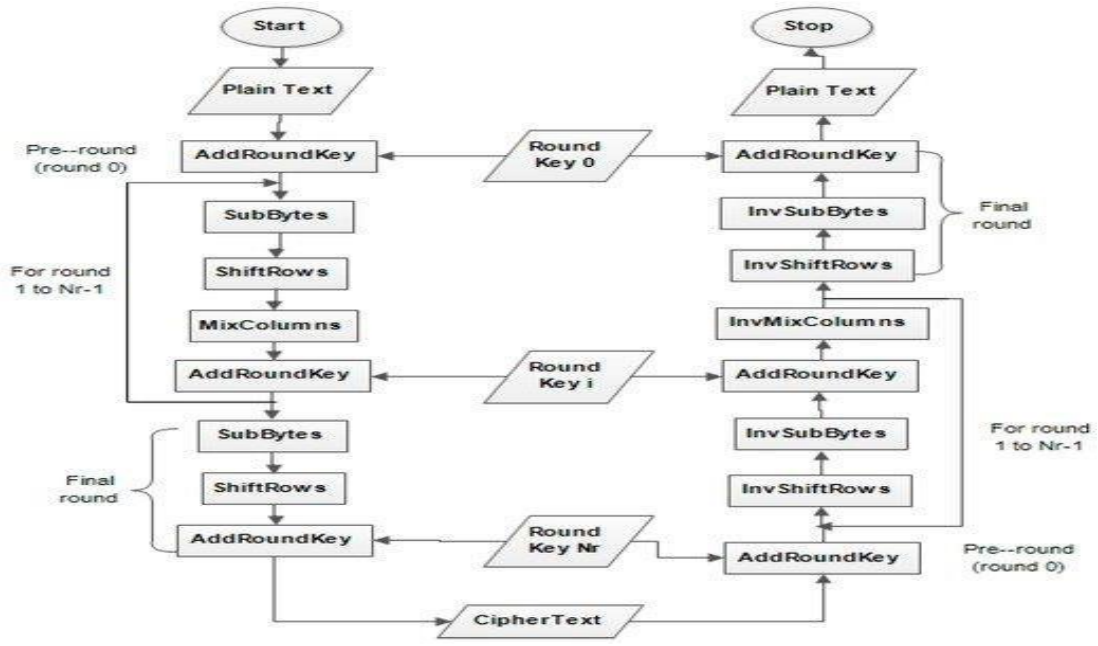


***Figure 1 Structure of the Advance Encryption Standard (AES)Algorithm***

## PROPOSED SYSTEM

The **AES encryption process** consists of two core components: **key scheduling** and **round transformation**. These elements play a crucial role in structuring the encryption mechanism efficiently. The encryption framework is divided into two essential modules: **Key Expansion** and **Key Selection**. Initially, the original key undergoes an expansion phase to generate multiple round keys. Once the appropriate round key is selected, the corresponding **plaintext block** proceeds to the **round transformation phase** for encryption.

To optimize data processing, the **encryption operand** is segmented into **32-bit units**, enabling streamlined data transmission and processing. This transformation improves the efficiency and performance of the encryption algorithm. **Figure 1.1** provides a detailed representation of the proposed algorithm's workflow.

.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

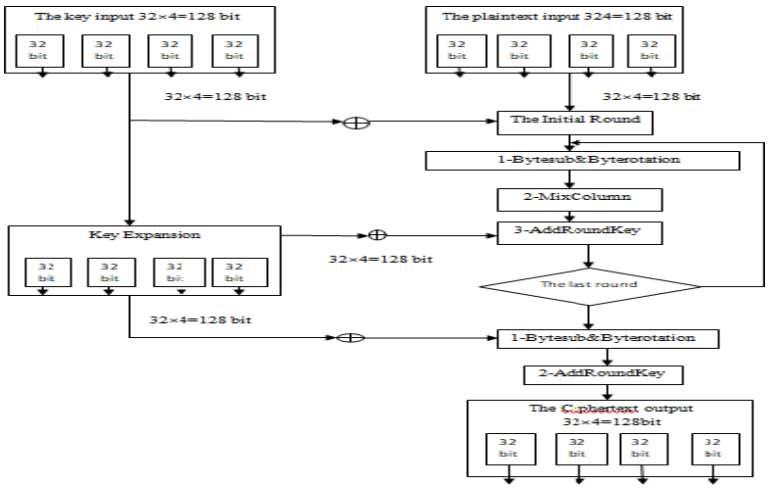


*Figure 2 The new improved structure of AES algorithm*

**The functionalities of different components within the above structure are outlined as follows**:

**The initial round of encryption:**

The encryption process begins by **loading four consecutive 32-bit plaintext packets** into the registers, forming a **128-bit data block**. Simultaneously, a clock signal manages the insertion of **four sequential 32-bit initial key packets** (128-bit key) into additional registers. The module then applies the **XOR operation** to combine the original key with the plaintext, initiating the encryption process.

**Round Transformation in the intermediate steps:**

The **round transformation** is a critical phase where **Sub Bytes** and **Mix Columns** operations are performed on **32-bit segments** independently. The round transformation module consists of **32 output ports** and **64 input ports** (32-bit plaintext and 32-bit key), enabling efficient encryption.

- **Sub Bytes Operation:** Implemented using a **Look-Up Table (LUT)**, where precomputed substitution values are stored in a **256×8-bit memory (1024 bits)** for efficient transformation.
- **Mix Columns Operation:** Based on mathematical principles of the **Galois Field GF(2⁸)**, requiring **multiplication and XOR operations** within each processing unit. Due to the commutative and associative properties of Galois field arithmetic, the **Mix Columns function** efficiently enhances diffusion.

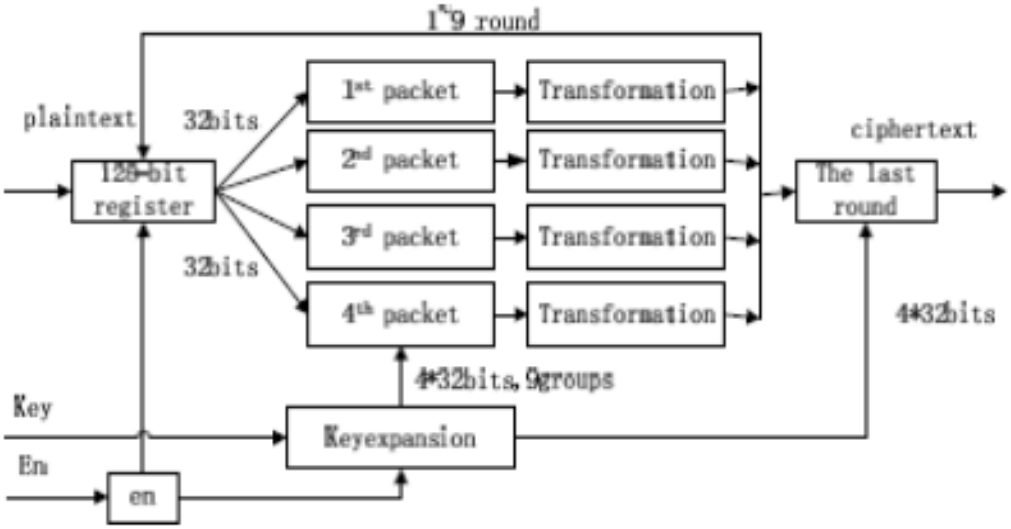


*Figure 3 round processing with AES*

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

**Pipelining in AES Encryption**

To enhance efficiency, the **128-bit data block** is **partitioned into four independent 32-bit packets**, allowing for parallel round transformations. This **pipelining approach** ensures seamless data processing, as unprocessed data is temporarily stored in a **128-bit register**. The register **synchronizes with the clock signal**, enabling a **continuous encryption cycle** for the **nine intermediate rounds** before the final cipher text is generated.

**The process of the last round**

**In the last encryption round, a** 128-bit processor **performs the final transformations, including** Shift Rows, Sub Bytes, and Mix Columns**. The encrypted** 128-bit intermediate data **is then processed using an** XOR operation with the final expanded key **(4×32-bit), obtained from the key expansion module. The final** 128-bit cipher text **is generated and subsequently divided into** four 32-bit packets**, which are processed based on an external control signal.**

**Key expansion and Key extraction**

The **key expansion module** functions similarly to traditional AES implementations but differs in its **data transmission approach**. The original and expanded keys are divided into **four 32-bit segments** before extraction, ensuring efficient key scheduling.

In **FPGA-based implementations**, the AES algorithm can be decomposed into fundamental operations such as **lookup-table searches and XOR functions**. Every encryption round begins with the **AddRoundKey operation**, integrating the round key with the data block. The encryption module executes its operations **simultaneously**, ensuring a fast and independent encryption process in FPGA architectures.

**RESULTS**

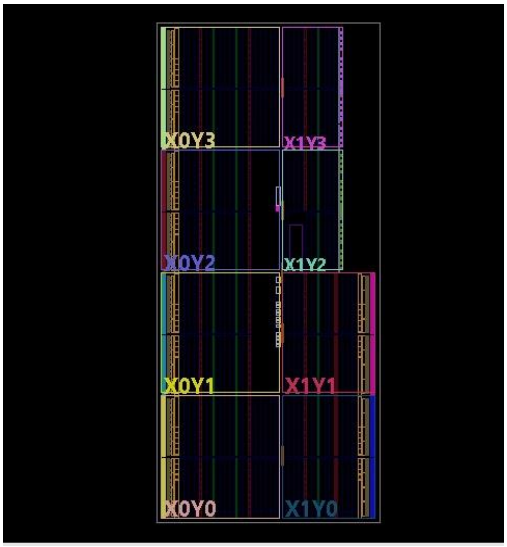


***Figure 3 Synthesized diagram of AES***

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

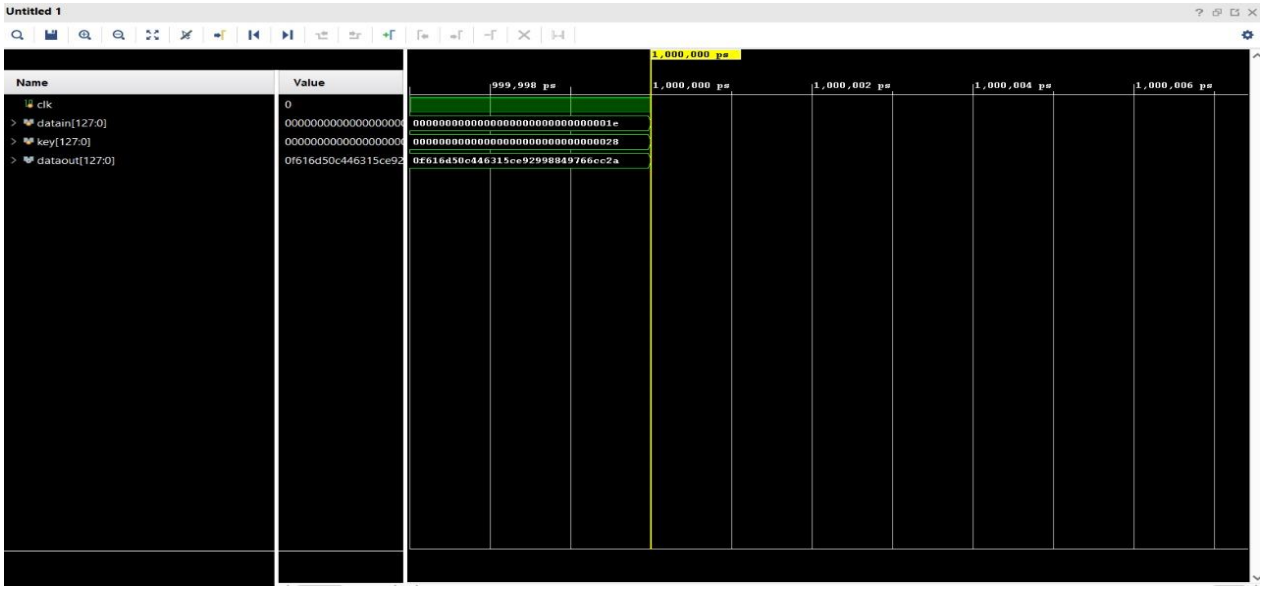


*Figure 3 Output of AES algorithm*

## CONCLUSION

This paper introduces an **optimized AES variant** designed for **resource-constrained environments**. The proposed model incorporates a **new Substitution Box (S-Box)**, which operates within the **Galois Field ($2^4$)** and is derived using a unique **affine transformation equation**. A key advantage of this **Modified AES (MAES)** is its ability to **reduce energy consumption**, thereby extending the battery life of **low-power devices**. Experimental results indicate that the **proposed MAES method enhances encryption efficiency by 18.35%**, leading to an **increase in the number of successfully transmitted packets**. Additionally, the approach achieves a **latency of 29.983 milliseconds**, demonstrating improved performance in secure data transmission.

## FUTURESCOPE

Future research will focus on addressing **security challenges** and **space complexity** to enhance the practicality of the proposed modifications. Additionally, we aim to explore a **multipath routing strategy** for efficiently transmitting **encrypted data to the sink node**. Further developments will include the **integration of Public Key Cryptosystems**, particularly **Elliptic Curve Cryptography (ECC)**, to achieve **improved security while maintaining efficiency** in terms of **packet transmission and latency**.

## REFERENCES

[1] Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). *Internet of Things (IoT): A literature review.* Journal of Computer and Communications, **3(05)**, 164.

[2] Wang, Y., Attebury, G., & Ramamurthy, B. (2006). *A survey of security issues in wireless sensor networks.* IEEE Communications Surveys & Tutorials.

[3] Veeramallu, B., Sahitya, S., & Lavanya Susanna, C. (2013). *Confidentiality in Wireless Sensor Networks.*

[4] Eisenbarth, T., & Kumar, S. (2007). *A survey of lightweight cryptography implementations.* IEEE Design & Test of Computers, **24(6)**.

[5] Banik, S., Bogdanov, A., & Regazzoni, F. (2015). *Energy efficiency analysis of lightweight block ciphers.* Selected Areas in Cryptography Conference, Springer, Cham.

[6] Bogdanov, A., et al. (2007). *PRESENT: An ultra-lightweight block cipher.* CHES, **Vol. 4727**.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

[7] Beaulieu, R., et al. (2015). *The SIMON and SPECK lightweight block ciphers.* Design Automation Conference (DAC), 52nd ACM/EDAC/IEEE.

[8] Suzaki, T., et al. (2012). *TWINE: A lightweight block cipher for multiple platforms.* Selected Areas in Cryptography, **Vol. 7707**.

[9] Shibutani, K., et al. (2011). *Piccolo: An ultra-lightweight block cipher.* CHES, **Vol. 6917**, 342-357.
[10] Wu, W., & Zhang, L. (2011). *LBlock: A lightweight block cipher.* Applied Cryptography and Network Security, Springer Berlin/Heidelberg.

[11] Feldhofer, M., Wolkerstorfer, J., & Rijmen, V. (2005). *AES implementation on a grain of sand.* IEE Proceedings-Information Security, **152(1)**, 13-20.

[12] Moradi, A., Poschmann, A., Ling, S., Paar, C., & Wang, H. (2011). *Compact and threshold implementation of AES.* Eurocrypt, **Vol. 6632**, 69-88.

[13] Hocquet, C., Kamel, D., Regazzoni, F., Legat, J.-D., Flandre, D., & Bol, D. (2011). *Harvesting the potential of nano-CMOS for lightweight cryptography: Ultra-low voltage AES coprocessor for RFID tags.* Journal of Cryptographic Engineering, **1(1)**, 79-86.

[14] Kerckhof, S., Durvaux, F., Hocquet, C., Bol, D., & Standaert, F.-X. (2012). *Towards green cryptography: Energy-efficient lightweight cipher comparison.* CHES 2012, **390-407**.

[15] Batina, L., et al. (2013). *Energy and performance analysis of lightweight block ciphers.* International Workshop on RFID Security and Privacy, Springer, Berlin, Heidelberg.

[16] Banik, S., Bogdanov, A., & Regazzoni, F. (2015). *Energy consumption analysis of lightweight block ciphers in FPGA.*