STM JOURNALS

Recent Trends in
Electronics & Communication Systems

ISSN: 2393-8757
Volume 13, Issue 1, 2026
January–April
DOI (Journal): 10.37591/RTECS

https://journals.stmjournals.com/rtecs

Review RTECS

Performance Analysis of AES Encryption Using LFSR-Based Key Expansion in VLSI

V. Ratna Rajeswari Sri Vidya[1], S. Veera Venkata Gopala Krishna[4], T. Venkateswari[2], Y. Venkata Satya Krishna[5], T. Hemanth[3], B.S.D. Sarma Kompella[6,*]

Abstract

Secure data transmission has become increasingly important with the rapid growth of digital communication and embedded systems. Protecting sensitive information from unauthorized access requires reliable cryptographic solutions. Among the available techniques, the Advanced Encryption Standard (AES) is widely recognized for its strong security and efficient implementation in both hardware and software environments. In this work, the design and performance evaluation of an AES-based crypto processor with LFSR-driven key expansion is presented for VLSI applications. Unlike the conventional key scheduling approach, the proposed method utilizes a Linear Feedback Shift Register (LFSR) to generate round keys in a hardware-efficient manner. The design is modeled using Verilog hardware description language (HDL) and validated through simulation using Xilinx Vivado. The proposed architecture emphasizes both structural design optimization and performance improvement. Experimental results indicate a reduction in hardware complexity along with enhancements in speed and power efficiency. These characteristics make the design suitable for field-programmable gate array (FPGA)-based secure systems and resource-constrained embedded applications. The suggested design shows enhanced scalability and flexibility, allowing it to accommodate various key sizes and encryption needs. Utilizing LFSR also diminishes computational overhead, which helps to reduce latency and promote effective hardware use. Moreover, synthesis outcomes validate superior area-delay performance in comparison to conventional AES implementations.

Keywords: AES, LFSR, cryptography, FPGA, VLSI, Verilog HDL

INTRODUCTION

With the continuous expansion of digital technologies, such as Internet-based services, embedded platforms, and internet of things (IoT) systems, the requirement for secure data communication has become more critical than ever. Ensuring the confidentiality and integrity of transmitted data is a major concern, particularly in environments where sensitive information is exchanged. Cryptographic techniques play a fundamental role in addressing these challenges by providing secure encryption. Among the various algorithms, the Advanced Encryption Standard (AES) has gained widespread acceptance owing to its robustness, efficiency, and adaptability to hardware implementations [1, 2].

AES is a symmetric key encryption algorithm that operates on fixed-size data blocks through a sequence of transformation rounds involving substitution and permutation. One of the essential components of AES is the key expansion process, which derives multiple round keys from the original

***Author for Correspondence**
BSD Sarma Kompella
E-mail: drkbsdsarma.bvce@bvcgroup.in

[1-5]UG Scholar, Department of Electronics and Communication Engineering, Bonam Venkata Chalamayya Engineering College (Autonomous), Odalarevu, Andhra Pradesh, India
[6]Professor Department of Electronics and Communication Engineering, Bonam Venkata Chalamayya Engineering College (Autonomous), Odalarevu, Andhra Pradesh, India

Received Date: April 02, 2026
Accepted Date: April 04, 2026
Published Date: April 18, 2026

Citation: V. Ratna Rajeswari Sri Vidya, S. Veera Venkata Gopala Krishna, T. Venkateswari, Y. Venkata Satya Krishna, T. Hemanth, B.S.D. Sarma Kompella. Performance Analysis of AES Encryption Using LFSR-Based Key Expansion in VLSI. Recent Trends in Electronics & Communication Systems. 2026; 13(1): 23–31p.

key. Although effective, the traditional key expansion technique contributes significantly to hardware complexity and may limit performance when it is implemented in Very-Large-Scale Integration(VLSI) systems.

To improve efficiency, this study introduces an alternative approach by incorporating a Linear Feedback Shift Register (LFSR) into the key expansion process. The LFSR is known for its simple architecture and ability to generate pseudo-random sequences with minimal hardware resources. By integrating LFSR into AES, the key generation process becomes faster and more resource-efficient, thereby enhancing the overall system performance without significantly affecting security. This study focuses on the design and performance evaluation of AES with LFSR-based key expansion.

In a VLSI environment, the proposed system was analyzed with respect to key performance metrics, such as area utilization, operating speed, and power consumption. The results demonstrate that the modified architecture offers an efficient solution for high-speed and resource-limited applications [3, 4].

LITERATURE SURVEY

The increasing demand for secure, high-speed digital communication has motivated extensive research on cryptographic hardware design, particularly in VLSI-based systems. As data security has become a fundamental requirement in modern applications, the efficient implementation of encryption algorithms with optimized performance has gained significant attention. Among the various encryption standards, the AES continues to be the most widely adopted symmetric key algorithm owing to its proven reliability, scalability, and suitability for hardware realization. However, achieving an optimal balance between security and hardware efficiency remains a critical design challenge.

Earlier implementations of AES primarily focused on improving throughput by adopting architectural optimizations, such as pipelining and parallel processing. A fully pipelined architecture allows multiple data blocks to be processed simultaneously across different stages, resulting in a significantly higher throughput. Despite this advantage, such designs often lead to increased area utilization and power consumption. To address these limitations, alternative architectures, including iterative and loop-unrolled designs, have been explored. These approaches aim to provide a trade-off between speed and hardware complexity by either reusing hardware resources or partially parallelizing the computation [5, 6].

In addition to architectural improvements, several studies have focused on optimizing the individual transformation stages within AES. The SubBytes operation, which relies on S-box implementation, is one of the most resource-intensive components. Conventional lookup-table-based designs require substantial memory, which may not be suitable for resource-constrained environments. To overcome this, composite-field-arithmetic-based implementations have been proposed, offering a reduced area and improved efficiency while maintaining the required cryptographic properties.

Another important aspect influencing the AES performance is the key expansion mechanism. The traditional key scheduling process involves multiple operations, such as substitution, rotation, and exclusive OR (XOR) with predefined constants. Although this method ensures strong security, it contributes significantly to computational overhead and increases overall hardware complexity. Consequently, researchers have investigated alternative key generation techniques that can simplify the design while preserving acceptable security levels.

LFSRs have emerged as an efficient solution for generating pseudo-random sequences in digital systems. Due to their simple structure, low resource requirements, and high-speed operation, LFSRs are widely used in applications such as random-number generation and stream cipher systems. Their suitability for hardware implementation makes them attractive candidates for integration into cryptographic designs. However, when used independently, LFSRs may not provide sufficient cryptographic strength and must be carefully integrated within established encryption frameworks.

Recent research has explored the incorporation of LFSR-based techniques into AES, particularly for optimizing the key expansion stage. By replacing or modifying the conventional key schedule with LFSR-generated sequences, it is possible to reduce the critical path delay and improve the overall system efficiency. These hybrid approaches focus on enhancing performance parameters, such as area utilization, operating frequency, and power consumption, while attempting to retain the inherent security advantages of AES. Several implementations have demonstrated that LFSR-assisted key generation can achieve notable improvements in FPGA and application-specific integrated circuit (ASIC)-based designs [7, 8].

Despite these advancements, modifying the internal structure of the AES raises concerns regarding the maintenance of cryptographic robustness. Any optimization in design must be carefully evaluated to ensure that security is not compromised in favor of better performance. Therefore, a detailed performance analysis is essential to validate the effectiveness of these modifications in real-world applications.

In this context, the present study focuses on the design and performance evaluation of an AES encryption system that utilizes LFSR-based key expansion in a VLSI environment. The proposed approach is analyzed with respect to key hardware metrics, including area efficiency, speed, and power consumption, to establish its suitability for high-performance and resource-constrained applications, such as embedded systems and IoT devices [9, 10].

Methodology

Figure 1 presents the VLSI Design of AES using LFSR. This section describes the design approach and performance-oriented implementation of an AES-based crypto processor that incorporates an LFSR for key expansion in a VLSI environment, and the objective of the proposed methodology is to develop a hardware-efficient encryption architecture while simultaneously evaluating its performance in terms of area, speed, and power consumption requirements.

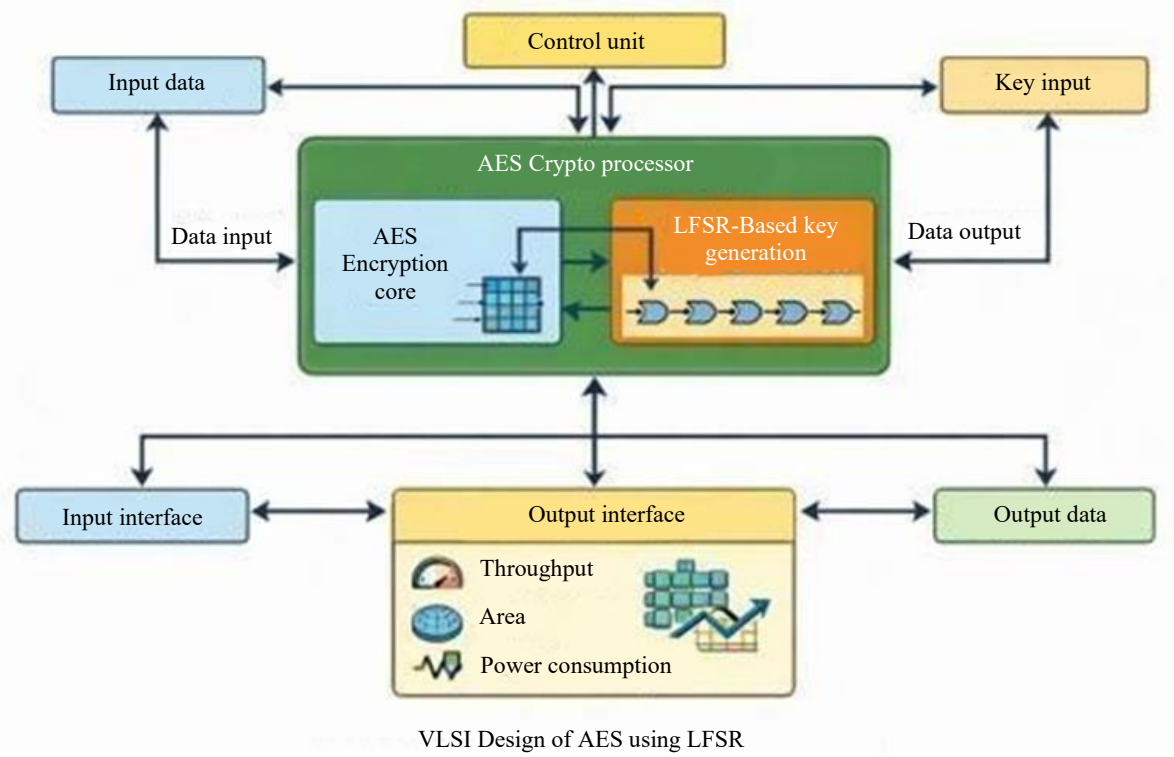

Figure 1. VLSI Design of AES using LFSR.

Overview of Proposed Design

The proposed system was developed by integrating an AES encryption engine with an LFSR-based key generation mechanism. Instead of relying on the conventional AES key scheduling technique, the design utilizes an LFSR to produce pseudo-random sequences that are used as round keys. This modification is primarily aimed at simplifying the hardware structure and improving the execution speed.

The complete architecture is organized into the following major functional blocks:

- AES encryption core
- LFSR-based key generator
- Control unit
- Input/output interface

The design is described using Verilog HDL and targeted for FPGA/ASIC implementation, enabling the practical validation of both design efficiency and performance characteristics.

AES Encryption Process

AES was implemented as the core encryption module, operating on 128-bit data blocks. The input plaintext is initially mapped to a state matrix and processed through multiple transformation rounds. For a 128-bit key, the encryption process consists of ten iterative rounds, each performing specific operations to achieve secure data transformation.

The major operations involved in each round include:

- Sub Bytes—non-linear byte substitution using S-box
- Shift Rows—cyclic shifting of rows within the state matrix
- Mix Columns—column-wise transformation to enhance diffusion
- Add Round Key—XOR operation between state and generated round key

In the final round, the MixColumns operation was excluded. These combined transformations ensure effective confusion and diffusion, which are essential for strong encryption.

In the final round, the MixColumns operation was omitted. These operations ensure confusion and diffusion during the encryption process, providing strong security.

LFSR-Based Key Expansion

In conventional AES, the key expansion stage involves multiple complex operations, which contribute to an increased hardware overhead. To address this issue, the proposed design replaces the traditional key scheduling mechanism with an LFSR-based approach.

The LFSR consists of a sequence of flip-flops connected with XOR-based feedback logic, defined by a characteristic polynomial and initial seed value. At each clock cycle, the LFSR generates a new pseudo-random output, which is used to generate round keys.

The key generation process is carried out as follows:

- The LFSR is initialized using a seed derived from the original AES key
- Pseudo-random sequences are generated through feedback operations
- The generated sequence is combined with the original key using XOR logic
- Round keys are derived dynamically for each encryption round

This approach reduces the complexity of the key expansion stage and improves the overall processing speed.

Integration of AES and LFSR

The integration strategy focuses on replacing the standard AES key schedule with an LFSR-based key generator while maintaining the core encryption structure. The generated round keys are directly supplied to the AddRoundKey stage in each encryption round. The integration process operates as follows.

- The initial key is loaded into the LFSR module.
- The LFSR continuously produces key sequences in synchronization with the clock.
- Generated round keys are applied sequentially to AES rounds.
- The encryption process proceeds until all rounds are completed.

This method enables efficient key generation with reduced combinational complexity, thereby contributing to improved overall system performance.

Hardware Implementation

The proposed architecture was implemented using Verilog HDL to ensure accurate hardware modeling. The design was synthesized and simulated using Xilinx Vivado, allowing a detailed analysis of both the functional behavior and hardware utilization.

The implementation focuses on evaluating the following performance parameters:

- Area utilization (Look-up tables (LUTs), slices, registers)
- Power consumption
- Operating frequency
- Propagation delay

The results obtained from the proposed design are compared with those of conventional AES implementations to assess improvements in both design efficiency and performance.

Performance Evaluation

The performance of the proposed AES-LFSR architecture was evaluated based on key hardware metrics. The primary objective of this study was to determine the effectiveness of the modified key expansion technique in enhancing system efficiency.

The evaluation highlights the following improvements:

- Reduced hardware complexity due to a simplified key generation mechanism
- Increased operating speed resulting from faster key generation
- Lower power consumption, making the design suitable for embedded applications

Overall, the proposed design demonstrates improved throughput and better resource utilization than traditional AES implementations. These characteristics make it well-suited for real-time and resource-constrained VLSI applications. Table 1 describes the I/O and Clocking Resource Requirements.

RESULTS AND DISCUSSION

The proposed AES cryptoprocessor with LFSR-based key expansion was implemented and analyzed in a VLSI environment using Xilinx Vivado. The design was synthesized to evaluate its performance in terms of speed, hardware utilization, and power characteristics. The results obtained provide insights into the effectiveness of the proposed architecture compared to conventional AES implementations.

Table 1. I/O and clocking resource requirements.

Resource type	Used	Available	Utilization
Bonded IOB	387	464	83.41%
HPIOB (M/S Pair)	315	384	82.03%
HDIOB (M/S Pair)	48	48	100.00%
Global Clock (BUFGCE)	2	208	0.96%

Table 2. FPGA resource utilization summary.

Module	CLB LUTs	CLB Registers	F7MUXes	F8MUXes
aes_inst2 (AES)	20,833	384	4,115	1,770
lfsr_inst1 (LFSR)	1	128	0	0
Total used	20,834	512	4,115	1,770
Available	230,400	460,800	115,200	57,600
Utilization %	9.04%	0.11%	3.57%	3.07%

Functional Verification

The functionality of the proposed system was validated using simulations. The input plaintext was successfully transformed into ciphertext using LFSR-generated round keys. The simulation results confirmed that the modified key expansion mechanism did not affect the correctness of the AES encryption process. The integration of the LFSR maintains the expected encryption behavior while introducing design efficiency.

Area Utilization and Hardware Complexity

The hardware efficiency of the proposed design was analyzed based on the FPGA resource utilization. The AES encryption core contributes most of the logic usage, whereas the LFSR-based key generator occupies minimal hardware resources. This clearly indicates that the introduced key expansion method does not significantly increase the area overhead.

The overall utilization shown in Table 2 remains within acceptable limits, demonstrating that the design is suitable for implementation on FPGA platforms. The reduction in logic usage compared to that of traditional key expansion methods highlights the efficiency of the proposed architecture.

Throughput and Speed

The use of LFSR-based key generation reduces the critical path delay during the key expansion stage. By replacing the iterative conventional key schedule with a pseudo-random bit sequence generator, the round keys are generated using minimal combinational logic. This optimization allows the system to achieve a higher operating frequency and improved overall encryption throughput compared with standard implementations.

Power Consumption and Signal Integrity

Power analysis indicates that the majority of the consumption is due to dynamic power, which is expected in active digital circuits. However, the simplified key expansion mechanism contributes to an overall reduction in power usage.

The design also demonstrated stable signal behavior, which was verified through noise and signal integrity analyses. The reduced hardware complexity further supports lower switching activity, contributing to improved power efficiency. These characteristics make the system suitable for low-power applications, such as embedded systems and IoT devices.

- Dynamic power ≈99%
- Logic power ≈47%
- Signal power ≈45%
- I/O power ≈8%
- Static power ≈1%

DISCUSSION

The results clearly show that integrating the LFSR into the AES key expansion stage improves overall hardware efficiency. The reduction in complexity leads to lower resource utilization and faster operations without compromising functional correctness.

Although the LFSR, as shown in Figure 2, generates pseudo-random sequences, careful design ensures that the security level remains acceptable for practical applications. The proposed architecture, as shown in Table 3, successfully achieves a balance between performance improvement and security requirements.

CONCLUSION

This paper presents the design and performance evaluation of an AES-based crypto processor with LFSR-based key expansion for VLSI applications. The proposed approach replaces the conventional key scheduling mechanism with a hardware-efficient LFSR structure, resulting in improved system performance. Figure 3 shows the reduced hardware usage (fewer LUTs).

Table 3. AES with LFSR gives higher speed than conventional AES.

Signal group	I/O standard	Slew rate	Drive strength	Min. remaining margin
aes_dec out [23:6]	LVCMOS18	SLOW	12 mA	20.69

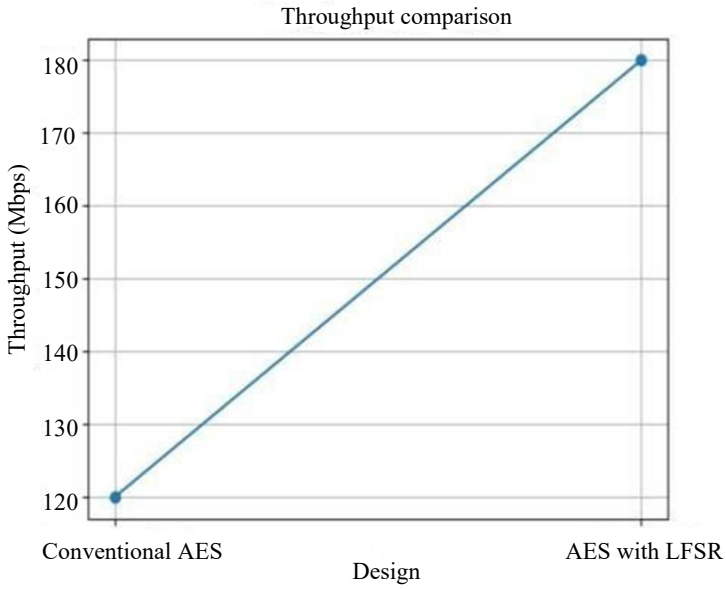

Figure 2. AES with LFSR gives high-speed conventional AES.

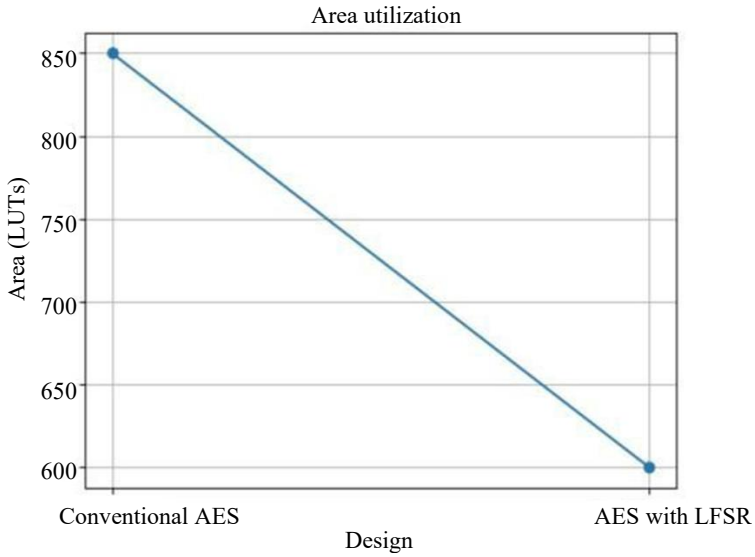

Figure 3. Reduced hardware usage (less LUTs).

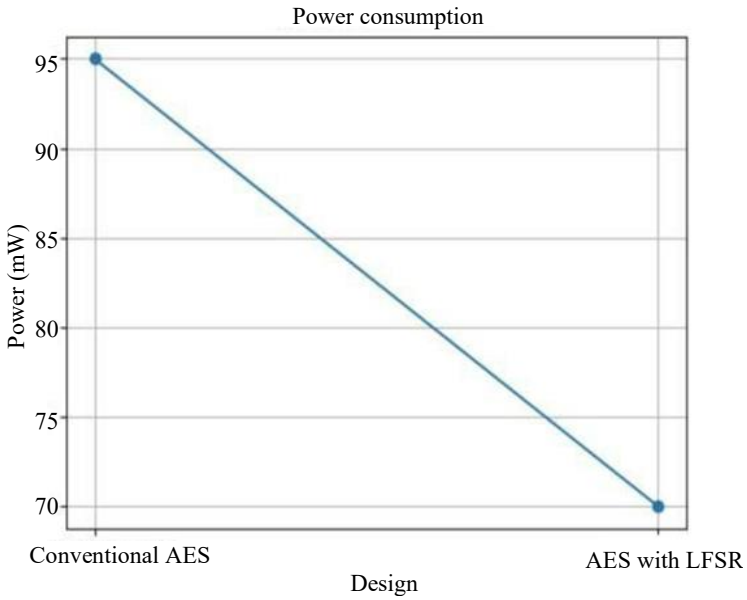

Figure 4. System power distribution

The implementation results demonstrate that the proposed architecture achieves reduced hardware complexity, increased operating speed, and lower power consumption. These improvements make the design suitable for high-performance and resource-constrained environments.

Furthermore, the integration of the LFSR does not affect the functional correctness of the AES encryption process, ensuring reliable operation. This study confirms that the proposed design offers an effective balance between security, performance, and hardware efficiency, making it a promising solution for modern cryptographic hardware systems. Figure 4 shows the System Power Distribution.

Future Scope

The proposed AES architecture can be further enhanced by exploring advanced techniques to improve the randomness and security of the LFSR-based key generation process. Hybrid approaches that combine LFSR with other cryptographic methods can also be investigated to achieve stronger security.

Future work may include extending the design to support both encryption and decryption operations, as well as implementing higher key sizes, such as AES-192 and AES-256, for improved security levels.

Additional optimizations, such as pipelining and parallel processing, can be applied to further increase the throughput. The implementation of advanced FPGA and ASIC platforms with detailed performance analysis can provide better insights into scalability.

Moreover, integrating the proposed design into real-time embedded and IoT systems will help validate its practical applicability in secure communication.

REFERENCES

1. Daemen J, Rijmen V. Proposal AES: Rijndael. Gaithersburg (MD): National Institute of Standards and Technology; 1999.
2. National Institute of Standards and Technology. Advanced encryption standard (AES). Federal Information Processing Standards Publication 197. Gaithersburg (MD): National Institute of Standards and Technology; 2001.
3. Whitaker NP. Applications of number theory in cryptography and cybersecurity. Sankalpa Int J Manag Decis. 2026;12(1):755–760.

4. Menezes AJ, Van Oorschot PC, Vanstone SA. Handbook of Applied Cryptography. Boca Raton (FL): CRC Press; 2018. doi:10.1201/9780429466335.
5. Morioka S, Satoh A. An optimized S-box circuit architecture for low power AES design. In: Kaliski BS, Koç CK, Paar C, editors. CHES '02: Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems; 2002 Aug 13-15. Berlin, Heidelberg: Springer-Verlag; 2002. p. 172–186.
6. McLoone M, McCanny JV. High performance single-chip FPGA Rijndael algorithm implementations. In: Koç CK, Naccache D, Paar C, editors. CHES '01: Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems; 2001 May 14-16. Berlin, Heidelberg: Springer-Verlag; 2001. p. 65–76.
7. Zhang X, Parhi KK. High-speed VLSI architectures for the AES algorithm. IEEE Trans Very Large Scale Integr Syst. 2004;12(9):957–967. doi:10.1109/TVLSI.2004.832943.
8. Lee D, Kwak M, Lee J, Kim B, Kim Y. A light-weight AES design using LFSR-based S-box for IoT applications. IEIE Trans Smart Process Comput. 2022;11(2):140–148.
9. Kumar P, Kamatchi S. A secure, area efficient strong physical unclonable function design using LFSR. 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India. 2020. p.1145–1149. doi:10.1109/ICOSEC49089.2020.9215447.
10. Purohit S, Deshpande V, Ingale V. FPGA implementation of the AES algorithm with lightweight LFSR-based approach and optimized key expansion. 2023 IEEE International Conference on Public Key Infrastructure and its Applications (PKIA), Bangalore, India. 2023. p.1–7. doi:10.1109/PKIA58446.2023.10262697.

