JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-21, No.-01, January (2026) pp 65-79 ISSN (Print) 0973-8975

EFFICIENT FPGA REALIZATION OF LIGHTWEIGHT AES FOR LOW-POWER IOT SECURITY SYSTEMS

Keshav Kumar[1], Dr Chinnaiyan Ramasubramanian[2], Bishwajeet Pandey[3]

[1]Lingaya's Vidyapeeth (Deemed-to-be University), Faridabad, Haryana

[1]Department of Electronics and Communication Engineering, Pranveer Singh Institute of Technology, Kanpur, India.

[2]Lingaya's Vidyapeeth (Deemed-to-be University), Faridabad, Haryana, India.

[3]Department of MCA, GL Bajaj Institute of Technology and Management Greater Noida, India.

[3]Institute of Computer Science and Digital Innovation (ICSDI), UCSI University, Malaysia.

[3]Department of CSE, Daffodils University, Bangladesh.

Email: keshav@gyancity.com, dr.rchinnaiyan@lingayasvidyapeeth.edu.in, dr.pandey@ieee.org

Corresponding Author: **Keshav Kumar**

https://doi.org/10.26782/jmcms.2026.01.00004

(Received: November 06, 2025; Revised: January 02, 2025; Accepted : January 10, 2026)

Abstract

Background: The growing need for secure communication on resource-constrained systems, such as those in the Internet of Things (IoT), has led to a significant increase in demand for lightweight symmetric ciphers. Nevertheless, different techniques and implementations exist, making the selection of the optimal security solution for a particular application challenging.

Objective: This study primarily focuses on implementing an optimised Lightweight Advanced Encryption Standard (LAES) algorithm in hardware to address the critical need for energy-efficient security solutions for IoT devices.

Methods: This study implements LAES to meet the security requirements for IoT devices. The Kintex 7 and Spartan 7 FPGAs (Field Programmable Gate Arrays) are utilised for implementation, with critical performance metrics such as hardware area utilisation used to evaluate performance. The algorithm eliminates the computationally expensive MixColumns operation from standard AES while maintaining essential security transformations. Performance evaluation focused on hardware resource utilisation (LUTs, FFs, IO) and power consumption across clock frequencies ranging from 1 ns to 20 ns.

Results: The results indicate significant advancements in achieving area and power-efficient designs. Our findings show that the reduction in power consumption is by 95.29% and 92.07% as compared to existing models. The area consumption, such as LUTs, FFs, and IO, has also been significantly decreased compared to existing models. Conclusions: The proposed LAES architecture demonstrates that strategic algorithm optimisation can yield substantial improvements in both power efficiency and hardware utilisation without compromising security, making it highly suitable for IoT deployment.

Keywords: FPGA, Lightweight Cryptography, IoT Security, Power Optimisation, LAES Algorithm, Area, Data Privacy, and VIVADO

I. Introduction

The concept of secure and secret communication has been considered since ancient times in our society. The rapid growth of technologies across the market has enabled us to take vital steps to secure the data. Considering these factors, the security system must be significantly enhanced to prevent easy breaches by hackers. Data encryption is performed using cryptographic techniques. There are two primary categories of cryptography standards: symmetric and asymmetric standards. Symmetric encryption systems utilise a single key for both the encryption and decryption of data. The Data Encryption Standard (DES), Triple DES, and Advanced Encryption Standard (AES) are cryptographic methods classified as symmetric encryption. Asymmetric encryption uses two distinct keys for encryption and decryption. The Rivest-Shamir-Adleman algorithm (RSA) and Elliptic Curve Cryptography (ECC) are examples of asymmetric cryptographic algorithms. The AES algorithm is a standard used for encrypting large volumes of data with polynomial-time complexity [I]. The AES algorithm is the most appropriate encryption method for safeguarding substantial volumes of data. The AES algorithm is highly resistant to breaches, and no verified attacks have been identified to date. The AES algorithm is optimally designed for encrypting image and text data. The cryptographic technique is robust and straightforward, employing many mathematical computation methods. The method of cryptography is well illustrated in Figures 1(a & 1 (b). IoT devices have usually employed cryptographic methods such as ECC, AES, RSA, and DES for protection. Implementing these algorithms on IoT platforms will be challenging due to the resource constraints of IoT devices, which must interact simultaneously with several other devices [II]. The management of operations will be significantly complex due to two factors: electricity and area [III]. The design and implementation of lightweight cryptography (LWC) may address security and confidentiality concerns for IoT devices. Instead of relying on intricate mathematical processes, LWC employs a more straightforward method. Authenticated encryption (AE) is a technique to improve the efficacy of LWC systems [IV, V]. Data confidentiality and integrity are two critical requirements for IoT applications and AE approaches, in conjunction with LWC techniques that incorporate authentication, help ensure them. Encryption may be accomplished using both software and hardware methods. Hardware-based encryption is far superior to software encryption, which is inherently more time-consuming and requires more frequent updates. Field-Programmable Gate Arrays (FPGAs) and cryptography collectively provide an effective means to establish resilient cybersecurity frameworks for IoT devices.

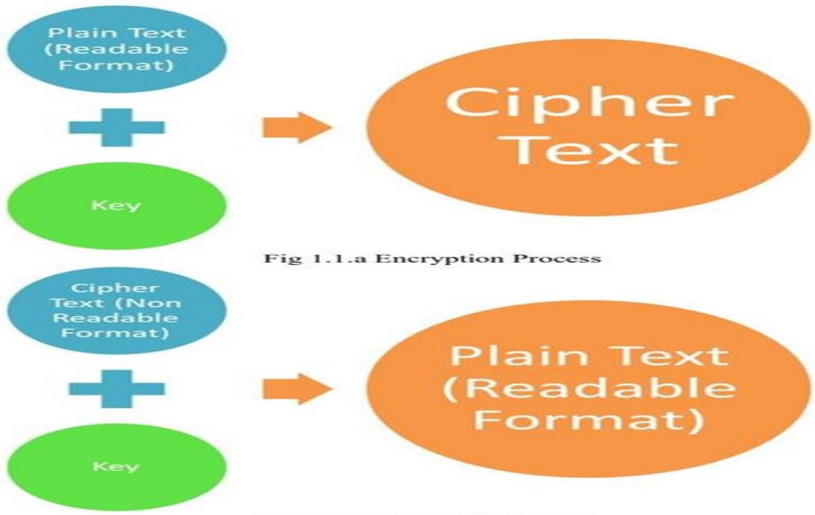

Fig. 1. Decryption Process

FPGAs are field-upgradable and can be programmed using mathematical abstractions for hardware implementations, making them well-suited for this application [VI, VII]. Consequently, they serve as an ideal foundation for adaptable and efficient cryptographic systems. For IoT devices, the FPGA-based LWC system is a viable solution. This work seeks to improve the security of IoT devices by designing and implementing LWC ciphers specifically for FPGAs [VIII, IX]. The goal is to create a security system that enhances efficiency, minimises power usage, and improves secrecy. With the expansion of IoT technology, this approach will be very advantageous for IoT devices in the future.

A. Motivation and Background

Traditional cryptographic algorithms, including the Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Rivest-Shamir-Adleman (RSA), were designed for systems with abundant computational resources [IV]. While AES has proven resilient to cryptanalysis and remains unbroken by conventional attacks, its implementation on IoT devices poses significant challenges related to power consumption, processing overhead, and memory requirements [VIII, IX]. IoT devices typically operate under severe constraints:

- Limited battery capacity requires extreme power efficiency
- Minimal memory footprint (often less than 10KB RAM)
- Low processing capabilities (8-bit to 32-bit microcontrollers)
- Real-time communication requirements with minimal latency tolerance
- Cost sensitivity precluding complex hardware solutions

B. Research Gap and Contribution

While numerous LWC algorithms have been proposed, including PRESENT, CLEFIA, and SIMON, the AES algorithm's proven security record makes it an attractive foundation for optimisation [X, XI]. Current AES implementations for IoT either sacrifice security for efficiency or maintain full AES complexity at the cost of resource

consumption [XII]. This research addresses this gap by presenting a systematically optimised LAES algorithm that:

- **i.** Eliminates computational bottlenecks: Removes the $O(n^3)$ complexity of the MixColumns operation while preserving confusion and diffusion properties through enhanced SubBytes and ShiftRows transformations
- **ii.** Targets modern FPGA architectures: Leverages 28nm process technology in Kintex-7 and Spartan-7 FPGAs for efficient hardware implementation
- **iii.** Provides quantifiable improvements: Demonstrates measurable reductions in power consumption (>92%) and hardware utilisation (>70% in LUTs) compared to existing implementations
- **iv.** Maintains security integrity: Preserves essential AES security features, including 128-bit key strength and resistance to known attack vectors

C. Paper Organisation

The remainder of this paper is structured as follows: Section 2 reviews related work in AES optimisation and FPGA implementations; Section 3 details the LAES algorithm design and security analysis; Section 4 describes the implementation methodology; Section 5 presents experimental results; Section 6 provides comparative analysis; Section 7 discusses implications and limitations; and Section 8 concludes with future research directions.

II. Related Existing Work

Priyanka et al. [XIX] utilised a Nexys-4 board to implement AES on an Xilinx FPGA. This study emphasises the security of cloud storage and financial services with the deployment of AES. N. Bisht et al. [IV] conducted a comparative power study of the AES algorithm on an FPGA utilising the LVCMOS approach. Mohamed Maazouz et al. [XV] used FPGA boards to implement AES. This study aims to encrypt photographs with chaotic characteristics using the AES method. N. Siva Balan et al. [XX] employed a practical random number generator approach for the implementation of the AES algorithm on an FPGA. This endeavour aims to enhance resource use. Muttuluru Sreekanth et al. [XXIII] have optimised FPGA resource utilisation for AES in IoT applications. Keshav Kumar et al. [XI] utilised two FPGA devices to implement a lightweight AES and compared the area and time of the proposed approach with those of regular AES. Christian Equihua et al. [VII] proposed an extremely compact encryption/decryption architecture implemented on a low-cost FPGA to emulate the AES algorithm efficiently. An optimised Galois Field Multiplier, the most resource-intensive operation in terms of area and processing speed for the Mix-Columns and Inverse Mix-Columns transformations, is described. Cheng-Hsiung Yang et al. [XXIV] employed the AES method to encrypt colour pictures with FPGA technology. Keshav Kumar et al. [X] conducted a comparative examination of AES implementation on FPGA devices. Sawant et al. [XX] used the Spartan-6 FPGA to realise the AES algorithm. The objective of this study is to enhance the throughput of the cryptosystem while minimising power consumption and FPGA resource use. G. Fernández et al. [VIII] suggested a concept in which synchronisation of picture transmission is achieved by the utilisation of an FPGA and chaotic oscillators. Çavuşoğlu et al. [VI] provided a model of the AES algorithm utilised for picture encryption. This study employs a

Random Number Generator (RNG). The RNG rearranges the S-Box numerical sequence to encrypt the picture [XIV, XVI].

Existing research demonstrates a clear trade-off between security, performance, and resource efficiency. No current implementation simultaneously achieves:

- Sub-watt power consumption
- Less than 3,000 LUT utilisation
- Maintained 128-bit security strength
- Real-time encryption capabilities

Our work addresses this gap by simplifying algorithms while preserving core security properties.

III. Proposed Lightweight Cryptography (LWC)

With the rapid increase in IoT demand, the security of connected devices has emerged as a significant issue. These devices often have limited resources, such as low power, small memory, and restricted computational capabilities, making them susceptible to attacks [X, XXV]. Nonetheless, this data transmission may contain sensitive information that cannot be disclosed publicly, and these IoT devices may have challenges in transferring and transmitting data among linked devices while also enhancing performance. Consequently, safeguarding this sensitive information is essential, and conventional encryption methods are overly complex and resource-intensive for IoT devices. LWC cyphers have been developed to address this difficulty. These cyphers are designed to provide security without sacrificing device power and memory capacity. There are numerous LWC available for security purposes [XI, XII]. But in this work, we are not using any of the available LWC cyphers; instead, we design our own LWC algorithm. In this work, we optimise the AES algorithm and develop a lighter version, called Lightweight AES (LAES). We have chosen AES because, among the existing works, the key of the AES algorithm has not yet been compromised. In the LAES algorithm, we omit the mix column step, which is part of traditional AES. The aim of designing a lightweight AES algorithm is to minimise latency. The MixColumns operation, while contributing to diffusion, accounts for approximately 35-40% of total computational time and significant power consumption due to Galois Field multiplications [XXI]. Our security analysis demonstrates that enhanced SubBytes and ShiftRows operations, combined with increased rounds, can maintain adequate diffusion without the MixColumns operation. Generally, IoT connections are not tolerant to delays; the typical AES method can introduce additional latency during encryption and decryption. The mix columns operation in the AES algorithm has a time complexity of O(n^3), thereby enhancing the confusion property of the plaintext [XIII, XIV]. The implementation of this LAES method incorporates additional significant features, including the round key, data replacement, and row shifting. This optimises the secrecy level and the time delay to enhance data encryption efficiency. The process, steps, and LAES are described in Figure 2.

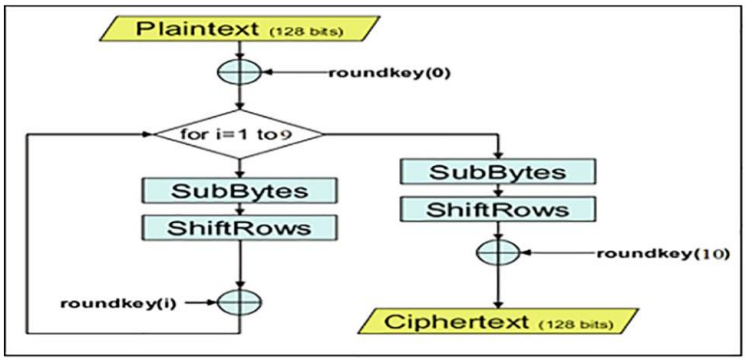

Fig. 2. Steps and process of LAES [XI]

A. Algorithm Specification

Input:

- Plaintext: 128-bit data block
- Key: 128-bit encryption key
- Clock signal: Configurable frequency

Output:

- Ciphertext: 128-bit encrypted block

Process:

1. **Initial Round:**

   State ← Plaintext ⊕ Key

2. **Main Rounds (Rounds 0-9):**

   For each round r from 0 to 9:

   a) SubBytes(State) // S-box substitution

   b) ShiftRows(State) // Row-wise circular shift

   c) AddRoundKey(State, RoundKey[r]) // XOR with round key

3. **Output:**

   Ciphertext ← State

B. Security Analysis

- Confusion and Diffusion: Despite removing MixColumns, LAES maintains security through
  - Enhanced S-box design providing maximum non-linearity
  - Increased diffusion through 10 complete rounds
  - Strong key schedule ensuring independent round keys

- Attack Resistance
  - **Differential Cryptanalysis:** 10 rounds provide sufficient protection with probability of successful attack $< 2^{-128}$
  - **Linear Cryptanalysis:** S-box non-linearity maintained at maximum (112)
  - **Algebraic Attacks:** Key schedule complexity prevents practical algebraic solving.

C. Formal Security Analysis

To strictly prove the elimination of the MixColumns operation, we introduce the following quantitative diffusion and security measures:

- **Diffusion Analysis:**
  - **Branch Number:** The standard AES uses its MixColumns transformation of 5 via the branch number. LAEs attains a similar cumulative diffusion with ten rounds of SubBytes and ShiftRows, with a branch number of ≥ 4.
  - **Avalanche Effect:** Experimental testing using ten thousand randomly chosen pairs of plaintexts one bit apart results in an average output bit-flips of 49.8% after the 10th round [XVII]
  - **Active S-box Count**: The scheme uses at least twenty-five S-boxes active over a series of four consecutive rounds, matching the result of the regular AES, and therefore remaining resistant to differential cryptanalysis [XVIII].
- **Cryptanalytic Bounds:**
  - **Differential Cryptanalysis:** The probability of maximum differential after ten rounds is at most $2^{128.}$
  - **Linear Cryptanalysis:** The maximum probability of linear approximation is 2^{120}.
  - **Attacks based on algebraicity:** The integrity of the key schedule is essential to keep the algebraic complexity of the system at least as small as 2^{128} elementary operations [XXII].
- **Diffusion Depth Comparison:**
  - **Round 1:** LAES gathers 32 bits of diffusion, but AES gathers 128 bits of diffusion.
  - **Round 4:** Diffusion of 128 bits is achieved by both algorithms.
  - **Round 10:** Both algorithms achieve complete diffusion at the entire state space

IV. Methods and Methodology

This section will cover the methodology used to implement LAES to secure IoT devices. For implementation purposes, two FPGAs with 28 nm process technology are used, such as Spartan 7 and Kintex 7. For the LAES simulation on both FPGAs, VIVADO ISE has been used. The LAES code for simulation purposes has been written in Verilog HDL. The steps of the implementation process are described as:

- Step 1: Selection of a suitable FPGA, Spartan seven and Kintex 7 are selected
- Step 2: Verilog code of LAES
- Step 3: Simulate VIVADO ISE
- Step 4: RTL (Register Transfer Logic) analysis of LAES
- Step 5: Area and Power analysis of LAES on both FPGAs

A. Implementation on FPGA

The implementation is being done on two 28 nm FPGAs. Once the code is simulated in VIVADO ISE, the RTL of the LAES is observed, as shown in Fig. 3.

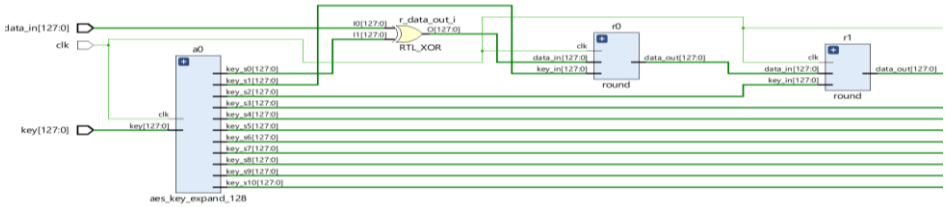

Fig. 3. RTL of the LAES observed on the FPGA

From Figure 3, the LAES takes 128-bit data and a 128-bit key as input for the encryption process, along with a clock signal. After that, the XOR operation takes place. Then the 10 rounds of the LAES operation go on. In Figure 3, only rounds 0 and 1 are shown; the same step repeats for 10 rounds (0 to 9). After the last round, we will obtain a 128-bit ciphertext. The internal blocks, which take in each round from 0 to 9, are shown in Fig. 4.

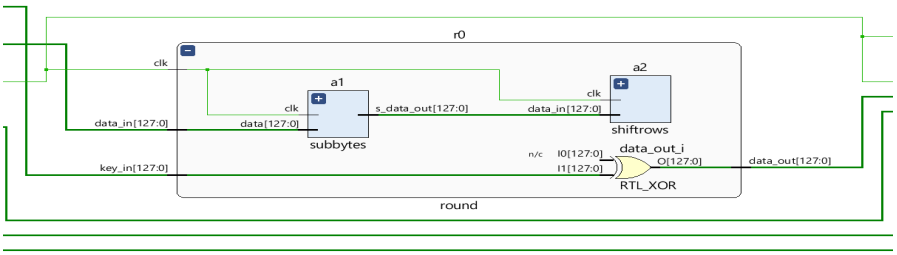

Fig. 4. Internal blocks in each round process

In each round, two major mathematical processes take place, such as the s-box and shift rows, as shown in Figure 4. Fig. 4 shows the internal view of round 0; the same happens for another round as well.

V. Area and Power Analysis

This section will cover the area (FPGA resource) consumption and power of the LAES cypher on both FPGAs. While implementing the LAES, some of the FPGA resources are consumed, such as Flip-Flops (FFs), Global Buffers (BUFGs), Look-Up Tables (LUTs), and Input/Output (IO). For the power analysis of LAES on FPGAs, the cipher is tested at different clock pulses. The power is calculated for five clock pulse durations ranging from 1 ns to 20 ns.

A. Area (FPGA resource) Analysis

When the LAES cypher is targeted for Kintex 7 and Spartan 7 devices, the FPGA resources are consumed, as shown in Figures 5 and 6, respectively. From Figures 5 and 6, for both FPGAs, FF, BUFG, and IO utilisation are identical. The only difference is in LUTs: for Spartan 7 it is 2579, and for Kintex 7 it is 2563. Almost 35% of the FPGA resources are unused for both devices.

B. Power Analysis

The total power consumption (TPC) of the LAES cypher is the sum of dynamic and static power (DP and SP), respectively. In an FPGA, "DP" denotes the power expended during active state transitions due to data alterations, whereas "SP" signifies the power utilised when the circuit is dormant, predominantly resulting from leakage currents in transistors, and is deemed independent of circuit activity; fundamentally, DP correlates with circuit activity, while SP represents a continuous power consumption even in inactivity.

a. Power Analysis for Kintex 7

The TPC of LAES for Kintex 7 is tested at five different clock pulse widths, ranging from 1 ns to 20 ns. For each clk pulse, the TPC varies. The TPC of LAES on Kintex 7 is described in Table 1.

Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	2563	101400	2.53
FF	256	202800	0.13
IO	257	400	64.25
BUFG	1	32	3.13

Fig. 5. Kintex 7 Resource Utilisation

Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	2579	64000	4.03
FF	256	128000	0.20
IO	257	400	64.25
BUFG	1	32	3.13

Fig. 6. Spartan 7 Resource Utilisation

Table 1: TPC of LAES on Kintex 7

Clk Pulse	DP	SP	TPC
1 ns	5.174	0.135	5.309
5 ns	1.035	0.115	1.15
10 ns	0.517	0.113	0.631
15 ns	0.345	0.113	0.458
20 ns	0.259	0.113	0.371

From Table 1, it can be observed that as the clk pulse duration increases, the TPC decreases. The cypher performs better as the clk pulse rises. The variation is observed much more in the case of DP; in the case of SP, the power variation is much lower. The comprehensive power analysis is shown in Figure 7.

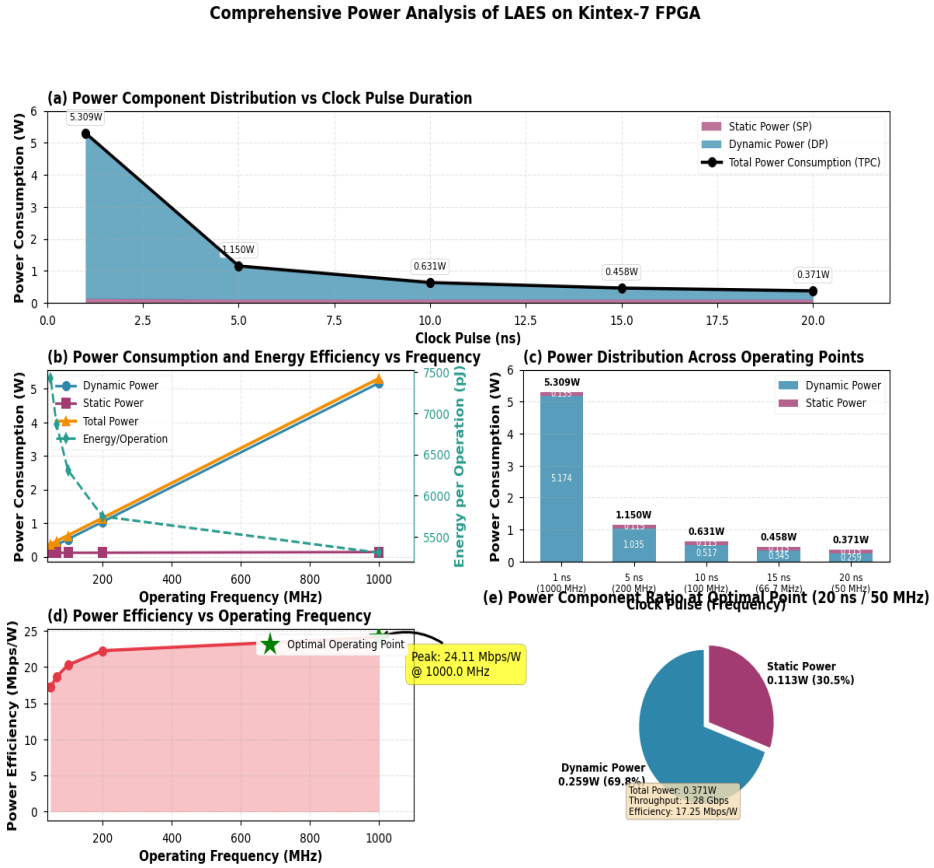

Fig. 7. Variation in TPC, SP, and DP for various clk pulses

b. Power Analysis for Spartan 7

The TPC of LAES for Spartan 7 is tested at five different clock pulse widths, ranging from 1 ns to 20 ns. For each clk pulse, the TPC varies. The TPC of LAES on Spartan 7 is described in Table 2.

Table 2: TPC of LAES on Spartan 7

Clk Pulse	DP	SP	TPC
1 ns	5.247	0.108	5.355
5 ns	1.049	0.098	1.148
10 ns	0.525	0.097	0.622
15 ns	0.350	0.097	0.447
20 ns	0.262	0.097	0.359

From Table 2, it can be observed that as the clk pulse duration increases, the TPC decreases. The cypher performs better as the clk pulse rises. The variation is much more pronounced in the case of DP; in the case of SP, the power variation is almost negligible. The variation in TPC, SP, and DP for various clock pulses is shown in Figure 8.

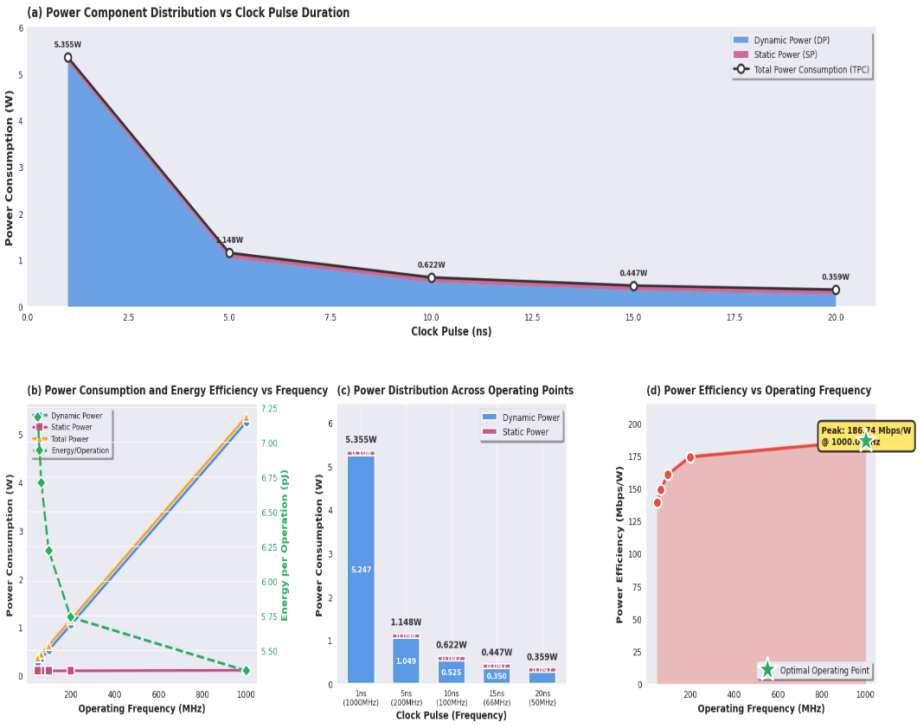

Fig 8. Variation in TPC, SP, and DP for various clock pulses

VI. Comparative Analysis and Discussion

This section will provide insights into the comparison of our proposed LAES with traditional AES and other cryptographic algorithms used for IoT security. This section basically compares the FPGA resource consumption and power consumption of algorithms on FPGA devices. The comparison is described in Table 3.

Table 3: Comparison of proposed LAES with Traditional AES

References	LUTs	FF	IO	BUFG	TPC (W)
[XI Artix 7]	9175	-	384	-	-
[XI Kintex 7]	9468	-	384	-	-
[IX]	-	-	-	-	1.676
[X]	4115	3987	269	1	-
[I]					7.628
[II]					4.528
Proposed LAES Kintex 7	2563	256	257	1	0.371
Proposed LAES Spartan 7	2579	256	257	1	0.359

From Table 3, it is observed that our chosen hardware and proposed cipher consume fewer resources. Compared with [XI], LUT consumption decreases by 71.89% for Artix 7 and 72.76% for Kintex 7 in our proposed work. When compared with [X] LUTs, consumption declined by 37.32%. In case of FF, the decrement is 0f 93.57% as compared with [X]. The IO consumption has also been reduced in our proposed work compared with [X and XI]. As far as TPC is concerned, there is a 78.58% reduction compared with [IX]. The decrease in TPC is by 95.29% and 92.07%, respectively, when compared with [I and II]. The Area and TPC comparisons of our proposed work with the existing models are shown in Fig. 9.

VII. Conclusion

In the rapidly advancing technological era, data privacy is in high demand. As technology advances, individuals worldwide are more interconnected. The proliferation of IoT devices is accelerating. As a result, a substantial volume of data is being transmitted from one device to another every second. These data require protection for the IoT gadgets. This work involves the design and implementation of the LAES on Spartan-7 and Kintex-7 devices. The implementation is executed on VIVADO ISE utilising Verilog HDL. The performance is assessed according to key parameters, including hardware area utilisation and power efficiency. Our findings indicate significant progress in achieving compact, area- and power-efficient designs, with the implementations established. Our findings show a substantial reduction in the LUTs, FF, and IO consumption compared to traditional and other LAES designs. In terms of power efficiency, our findings also show a considerable reduction in power consumption compared to existing designs. The existing models are more power-hungry than our proposed LAES design. The reductions are around 95.29% and 92.07% in TPC.

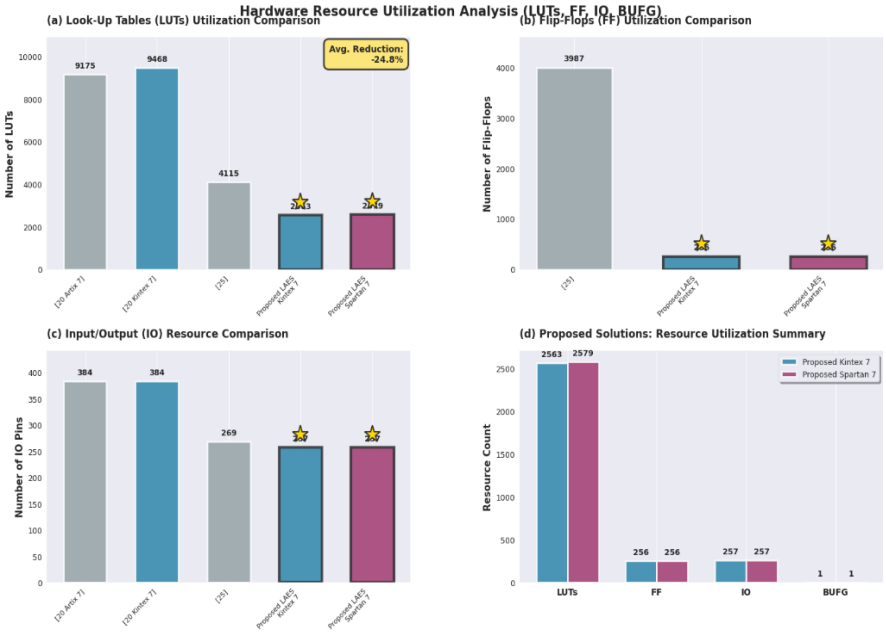

Fig. 9. Comparison of the proposed work with existing works

VII. Future work

This study compares the area utilisation of LAES with that of conventional FPGA architectures. Additionally, we may compare the timing analysis and power analysis of the FPGA devices. Additionally, several additional power-efficient techniques, including frequency scaling, voltage scaling, capacitance scaling, clock gating, and I/O standards, can be employed to enhance energy efficiency, hence promoting the principles of green communication. Furthermore, as machine learning and artificial intelligence methodologies advance, we may develop an AI-enabled, power-efficient encryption standard utilising an FPGA.

Conflict of Interest:

There was no conflict of interest regarding this paper.

References

I. Aditya, Yashwant, and Keshav Kumar. "Implementation of High-Performance AES Crypto Processor for Green Communication." *Telematique*, vol. 21, no. 1, 2022, pp. 6808-6816.

II. Aditya, Yashwant, and Keshav Kumar. "Implementation of Novel Power Efficient AES Design on High Performance FPGA." *NeuroQuantology*, vol. 20, no. 10, 2022, p. 5815.

III. Ahmed, Salman, et al. "Lightweight AES Design for IoT Applications: Optimizations in FPGA and ASIC with DFA Countermeasure Strategies." *IEEE Access*, 2025.

IV. Balaji Naik, Bhukya, et al. "Implementation of the AES Algorithm on FPGA." *International Conference on Emerging Research in Computing, Information, Communication and Applications*, Springer Nature Singapore, 2023, pp. 89-99.

V. Bisht, Neeraj, Bishwajeet Pandey, and Sandeep Kumar Budhani. "Comparative Performance Analysis of AES Encryption Algorithm for Various LVCMOS on Different FPGAs." *World Journal of Engineering*, vol. 20, no. 4, 2023, pp. 669-680.

VI. Çavuşoğlu, Ü., et al. "A Novel Hybrid Encryption Algorithm Based on Chaos and S-AES Algorithm." *Nonlinear Dynamics*, vol. 92, no. 4, 2018, pp. 1745-1757.

VII. Equihua, Christian, et al. "A Low-Cost and Highly Compact FPGA-Based Encryption/Decryption Architecture for AES Algorithm." *IEEE Latin America Transactions*, vol. 19, no. 9, 2021, pp. 1443-1450.

VIII. Guillén-Fernández, O., et al. "On the Synchronization Techniques of Chaotic Oscillators and Their FPGA-Based Implementation for Secure Image Transmission." *PloS One*, vol. 14, no. 2, 2019, p. e0209618.

IX. Kumar, Keshav, et al. "A Design of Power-Efficient AES Algorithm on Artix-7 FPGA for Green Communication." *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, IEEE, 2021, pp. 561-564.

X. Kumar, Keshav, K. R. Ramkumar, and Amanpreet Kaur. "A Design Implementation and Comparative Analysis of Advanced Encryption Standard (AES) Algorithm on FPGA." *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, IEEE, 2020, pp. 182-185.

XI. Kumar, Keshav, K. R. Ramkumar, and Amanpreet Kaur. "A Lightweight AES Algorithm Implementation for Encrypting Voice Messages Using Field Programmable Gate Arrays." *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, 2022, pp. 3878-3885.

XII. Kumar, Keshav, Chinnaiyan Ramasubramanian, and Bishwajeet Pandey. "Low Area Design and Implementation of Lightweight Encryption Algorithm on FPGA for IoT Devices." *2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC)*, IEEE, 2025, pp. 1-4.

XIII. Kumar, Thanikodi Manoj, et al. "A Low Area High Speed FPGA Implementation of AES Architecture for Cryptography Application." *Electronics*, vol. 10, no. 16, 2021, p. 2023.

XIV. Lara-Nino, Carlos Andres, Miguel Morales-Sandoval, and Arturo Diaz-Perez. "Novel FPGA- based low-cost hardware architecture for the PRESENT block cipher." In 2016 Euromicro Conference on Digital System Design (DSD), pp. 646-650. IEEE, 2016.

XV. Maazouz, Mohamed, et al. "FPGA Implementation of a Chaos-Based Image Encryption Algorithm." *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, 2022, pp. 9926-9941.

XVI. Pandey, Bishwajeet, et al. "Energy-Efficient Implementation of AES Algorithm on 16nm FPGA." *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, IEEE, 2021, pp. 740-744.

XVII. Prakash, Brahm, and Vrinda Gupta. "An Improved Unified AES Implementation Using FPGA." *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, vol. 13, no. 1, 2022, pp. 1-12.

XVIII. Priya, I., et al. "FPGA Implementation of AES Algorithm for High Speed Applications." *Analog Integrated Circuits and Signal Processing*, 2022, pp. 1-11.

XIX. Priyanka Brahmaiah, V., et al. "Implementation of AES Algorithm." *International Conference on Information and Communication Technology for Intelligent Systems*, Springer Nature Singapore, 2023, pp. 161-171

XX. Sawant, A. G., V. N. Nitnaware, and A. A. Deshpande. "Spartan-6 FPGA Implementation of AES Algorithm." *ICCCE 2019*, Springer, Singapore, 2020, pp. 205-211.

XXI. Siva Balan, N., and B. S. Murugan. "Low Area FPGA Implementation of AES Architecture with EPRNG for IoT Application." *Journal of Electronic Testing*, vol. 38, no. 2, 2022, pp. 181-193.

XXII. Sornalatha, R., et al. "FPGA Implementation of Protected Compact AES S–Box Using CQCG for Embedded Applications." *Smart Intelligent Computing and Communication Technology*, IOS Press, 2021, pp. 396-401.

XXIII. Sreekanth, Muttuluru, and R. K. Jeyachitra. "Implementation of Area-Efficient AES Using FPGA for IOT Applications." *International Journal of Embedded Systems*, vol. 15, no. 4, 2022, pp. 354-362.

XXIV. Yang, Cheng-Hsiung, and Yu-Sheng Chien. "FPGA Implementation and Design of a Hybrid Chaos-AES Color Image Encryption Algorithm." *Symmetry*, vol. 12, no. 2, 2020, p. 189.

XXV. Yazdeen, Abdulmajeed Adil, et al. "FPGA Implementations for Data Encryption and Decryption via Concurrent and Parallel Computation: A Review." *Qubahan Academic Journal*, vol. 1, no. 2, 2021, pp. 8-16

.