International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 10, October 2024)

Secure and Efficient Lightweight AES Algorithms for IoT Applications

[1]Praveen Kumar Bajpeyi, [2]Dr. Tarun Verma
[1]Research Scholar, [2]Professor
[1,2]Department of Electronics and Communication Engineering
[1,2]Lakshmi Narain College of Technology, Bhopal, India

***Abstract*— With the rapid growth of the Internet of Things (IoT), ensuring data security in resource-constrained devices has become a critical challenge. The Advanced Encryption Standard (AES), a widely adopted cryptographic algorithm, provides robust data security but often demands significant computational resources. This presents a challenge for IoT devices with limited power, memory, and processing capabilities. In response, lightweight versions of AES have been developed to meet the specific needs of IoT environments. These lightweight AES algorithms maintain the strong security properties of standard AES while optimizing efficiency in terms of speed, energy consumption, and resource usage. This paper presents a comprehensive review of lightweight AES algorithms tailored for IoT applications, focusing on their design principles, performance, and trade-offs. We also analyze their applicability in various IoT use cases, emphasizing the importance of balancing security and efficiency in IoT deployments.**

Keywords*— *Secure, Lightweight, AES Algorithms, IoT.

I. INTRODUCTION

The Internet of Things (IoT) has emerged as a transformative technological paradigm, interconnecting billions of devices worldwide. From smart homes and wearable devices to industrial automation and healthcare systems, IoT has the potential to revolutionize daily life and industries. However, the ubiquity of connected devices also introduces significant security risks, as IoT networks often handle sensitive data and operate in critical environments. Ensuring the security and privacy of data in such systems is paramount, particularly when these devices are highly constrained in terms of computational power, energy resources, and memory capacity.

Among the most commonly used cryptographic standards is the Advanced Encryption Standard (AES), which was selected by the National Institute of Standards and Technology (NIST) in 2001 as a robust, highly secure encryption method. AES offers strong resistance against most known cryptographic attacks, making it the gold standard for securing communications in a wide range of applications, from financial services to government data protection. However, the design of AES, while highly secure, was optimized for systems with ample resources. As IoT devices are often constrained by hardware limitations such as low processing power, limited battery life, and small memory footprints, standard AES can be too resource-intensive for direct application in many IoT scenarios.

This challenge has given rise to the development of lightweight cryptographic algorithms, specifically lightweight variants of AES, which are designed to provide a balance between security and efficiency. These lightweight AES algorithms aim to maintain the high security standards of AES while reducing the algorithm's complexity and resource consumption to make it suitable for IoT devices. The need for such algorithms is especially acute in applications where devices operate autonomously for extended periods without direct human intervention, such as in remote sensing or environmental monitoring systems, where frequent battery replacement is not feasible.

The primary goal of lightweight AES algorithms is to reduce the number of operations, memory requirements, and power consumption, without compromising the cryptographic strength of the encryption. By doing so, these algorithms enable secure communication across IoT networks without overwhelming the limited capabilities of the devices involved. For instance, lightweight AES variants might simplify the number of rounds in the encryption process, optimize key schedules, or utilize hardware accelerators that are better

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 10, October 2024)

suited for constrained environments. Despite these optimizations, designing lightweight AES algorithms is not without its challenges, as reducing resource usage often introduces trade-offs in areas such as encryption speed or potential vulnerability to side-channel attacks.

As the IoT landscape continues to evolve, the demand for secure and efficient cryptographic solutions will only increase. This is particularly true as IoT applications extend into critical infrastructure sectors such as healthcare, transportation, and energy, where data security is not only a matter of privacy but also of safety. In such environments, lightweight AES algorithms can play a crucial role by enabling secure, real-time communication while conserving the limited resources available to IoT devices. The importance of these algorithms extends beyond merely reducing resource consumption; they are vital to ensuring that even the most resource-constrained devices can maintain high standards of data security in increasingly complex IoT networks.

II. PROPOSED METHODOLOGY

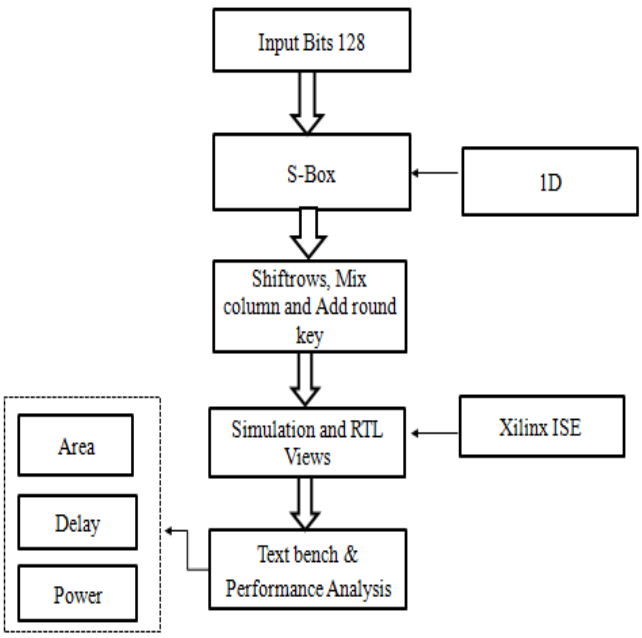


Figure 1: Flow chart

The flowchart you've provided outlines the process for implementing and analyzing a cryptographic algorithm, most likely AES (Advanced Encryption Standard), on a hardware platform using tools like Xilinx ISE (a tool for FPGA development). Here's a detailed description of the process step-by-step:

1. Input Bits 128

- **Description**: The algorithm begins by taking a 128-bit input, which is the block size for AES encryption. AES operates on 128-bit blocks, and this step involves receiving or formatting the input data into 128 bits for processing.

2. S-Box

- **Description**: The **S-Box (Substitution Box)** is a non-linear substitution step in AES. It is used for substituting bytes in the input data according to a predefined table, improving security by introducing confusion into the data. In this flow, the 128-bit input is passed through the S-Box transformation.
- **1D**: Likely refers to a one-dimensional lookup table used to implement the S-Box, which performs byte substitution on each byte of the input data.

3. Shiftrows, Mixcolumn, and Add Round Key

- **Description**:
  - **ShiftRows**: This is a step in AES where the rows of the state (the input after S-Box) are cyclically shifted by different offsets. It adds diffusion to the algorithm.
  - **MixColumns**: In this step, columns of the state are mixed using a mathematical transformation, enhancing diffusion.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 10, October 2024)

- **AddRoundKey**: This step XORs the state with a round key derived from the original encryption key. This step ties the encryption process to the secret key, ensuring security.

4. Simulation and RTL Views

- **Description**:
  - **Simulation**: At this stage, the AES algorithm (which includes the S-Box, ShiftRows, MixColumns, and AddRoundKey) is simulated to verify that the encryption process is working correctly.
  - **RTL Views**: RTL (Register Transfer Level) design is a hardware description of the AES algorithm. It represents the hardware implementation of the cryptographic process. Viewing RTL allows for insight into how the logic is designed in terms of flip-flops, logic gates, and connections.

5. Xilinx ISE

- **Description**: This refers to the use of **Xilinx ISE** software, which is used for synthesizing and implementing the AES algorithm on FPGAs (Field Programmable Gate Arrays). It converts the RTL description into a hardware implementation on a specific FPGA device.

6. Text Bench & Performance Analysis

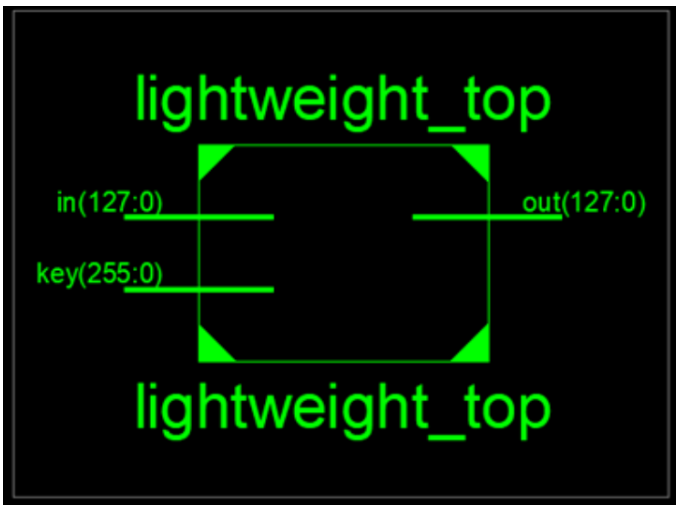
- **Area**: The amount of FPGA resources (e.g., logic gates, flip-flops) used to implement the AES algorithm.
- **Delay**: The time taken for the algorithm to process the input data and generate the output (latency).
- **Power**: The amount of power consumed by the hardware implementation of AES, which is crucial for IoT applications where power consumption is often limited.

7. Overall Flow:

- The flow begins with a 128-bit input that goes through the S-Box for substitution.
- Then, the data is transformed further using ShiftRows, MixColumns, and AddRoundKey.
- The entire process is simulated and synthesized into hardware using RTL design in Xilinx ISE.
- Finally, the performance of the AES implementation is analyzed by measuring area, delay, and power using a test bench.

III. SIMULATION AND RESULTS

Nano AES Security Algorithm implementation has multiple sub-modules inside it both at the Encryption and Decryption end, based on the internal operations of the algorithm.



International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 10, October 2024)

Figure 2: Top View

In figure 2, top view of proposed Nano-AES lightweight cryptography algorithm, where 128-bit input, 128-bit output and 256 Encryption and 256 Decryption key taken.

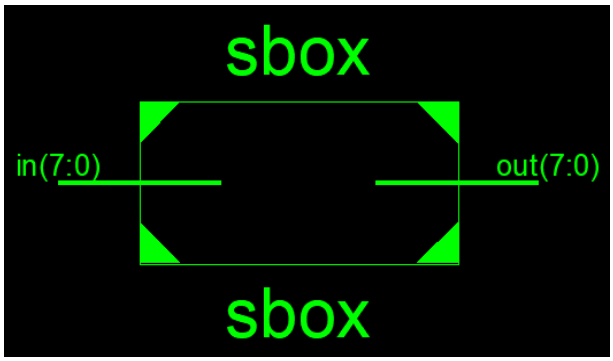


Figure 3: Top module of proposed 1D S-box

Figure 3 is showing the top module of the proposed 1-dimensional sub-byte box. Here 8-bit input is giving to the Sbox and its generating 8-bit output after operation of s-box.

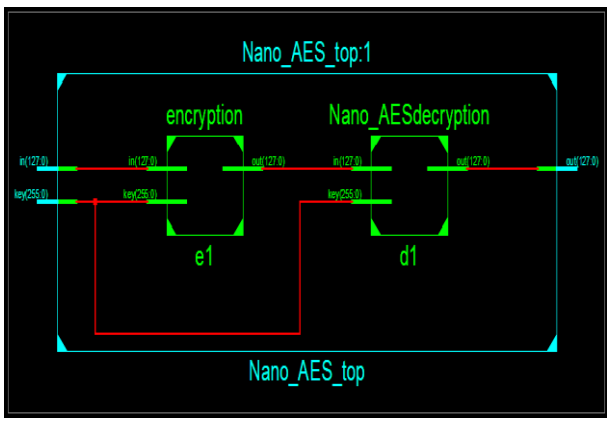


Figure 4: RTL view of Encryption and Decryption Process

The figure 4 is showing the RTL view of encryption and decryption process. The 128-bit input data is encrypted by the 256-bit key and at the output side it is decrypted by same 256 bit key and original data is recovered.

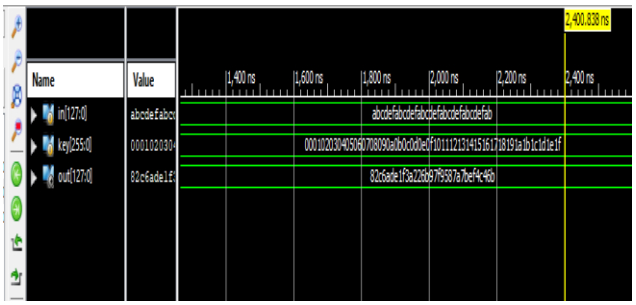


Figure 5: Encryption process

Figure 5 presents the encryption process of the proposed Nano-AES lightweight cryptography algorithm.

Input – abcdefabcdefabcdefabcdefabcdefab

Key-
h000102030405060708090a0b0c0d0e0f1011121314
15161718191a1b1c1d1e1f

Output - 82c6ade1f3a226b97f9587a7bef4c46b

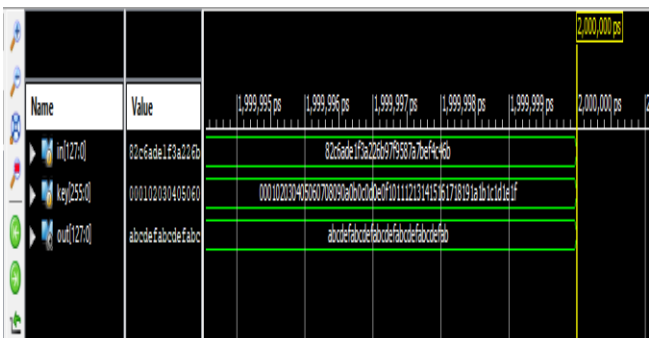


Figure 6: Decryption Process

Figure 6 presents the decryption process of the proposed Nano-AES lightweight cryptography algorithm.

Input – 82c6ade1f3a226b97f9587a7bef4c46b

Key-
h000102030405060708090a0b0c0d0e0f1011121314
15161718191a1b1c1d1e1f

Output – abcdefabcdefabcdefabcdefabcdefab

Table 1: Result Comparison

Sr No	Parameters	Previous Result	Proposed Result
1	Method	AES	Nano AES
2	Input bit	128	128
3	S Box	4 x 4	1X8
4	Chip area	13340	13016
5	Frequency	238.10 MHz	284.10 MHz
6	Delay	41.99ns	3.52ns

IV. CONCLUSION

This paper presents novel high performance of nano AES for lightweight implementation in IoT & AI Devices. Both methods utilize a 128-bit input size, but the Nano AES achieves greater efficiency with a more compact 1x8 S-Box configuration compared to the 4x4 S-Box in standard AES. This optimization leads to a reduced chip area, with the Nano AES occupying 13016 units versus 13340 units for the traditional AES, making it more suitable for space-constrained environments. The Nano AES also operates at a significantly higher frequency of 284.1 MHz compared to 238.10 MHz, allowing for faster data processing. Most notably, the delay is reduced from 41.99 ns in the traditional AES to just 3.52 ns in the Nano AES, indicating a substantial improvement in processing speed. These results underscore the Nano AES's suitability for high-performance, resource-efficient applications in IoT and AI devices.

REFERENCES

1. P. Y. Cheng, Y. C. Su and P. C. P. Chao, "Novel High Throughput-to-Area Efficiency and Strong-Resilience Datapath of AES for Lightweight Implementation in IoT Devices," in IEEE Internet of Things Journal, vol. 11, no. 10, pp. 17678-17687, 15 May15, 2024, doi: 10.1109/JIOT.2024.3359714.
2. S. Kumar *et al.*, "SHC: 8-bit Compact and Efficient S-Box Structure for Lightweight Cryptography," in *IEEE Access*, vol. 12, pp. 39430-39449, 2024, doi: 10.1109/ACCESS.2024.3372388.
3. X. He, Y. Bai, Y. Liu, L. Du, Z. Wang and Y. Du, "Low-Latency PAE: Permutation-Based Address Encryption Hardware Engine for IoT Real-Time Memory Protection," in *IEEE Internet of Things Journal*, vol. 11, no. 7, pp. 12319-12330, 1 April1, 2024, doi: 10.1109/JIOT.2023.3333203.
4. R. Huang, A. Aljuffri, S. Hamdioui, K. Ma and M. Taouil, "Securing an Efficient Lightweight AES Accelerator," *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Exeter, United Kingdom, 2023, pp. 841-848, doi: 10.1109/TrustCom60117.2023.00121.
5. J. Vimalkumar, H. R. Babu and B. M, "FPGA Implementation of Modified Lightweight 128-Bit AES Algorithm for IoT Applications," *2023 IEEE International Symposium on Smart Electronic Systems (iSES)*, Ahmedabad, India,

2023, pp. 306-309, doi: 10.1109/iSES58672.2023.00069.

6. P. Satyanarayana, N. Sriramdas, B. Madhavi, A. M, N. V. Phani Sai Kumar and V. Gokula Krishnan, "Enhancement of Security in IoT Using Modified AES Algorithm for IoT Applications," *2023 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, Theni, India, 2023, pp. 380-386, doi: 10.1109/ICSCNA58489.2023.10370606.
7. GS Rajput, R Thakur, R Tiwari "VLSI implementation of lightweight cryptography technique for FPGA-IOT application" Materials Today: Proceedings, 2023, ISSN 2214-7853, doi: 10.1016/j.matpr.2023.03.486.
8. S. Purohit, V. Deshpande and V. Ingale, "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, pp. 1-7, doi: 10.1109/PKIA58446.2023.10262697.
9. M. Gong, Z. Zheng and H. Li, "An Inverter-based Lightweight Digital True Random Number Generator Circuit for IoT Device," *2022 10th International Symposium on Next-Generation Electronics (ISNE)*, Wuxi, China, 2023, pp. 1-3, doi: 10.1109/ISNE56211.2023.10221587.
10. M. Nooruddin and D. Valles, "An Advanced IoT Framework for Long Range Connectivity and Secure Data Transmission Leveraging LoRa and ASCON Encryption," *2023 IEEE World AI IoT Congress (AIIoT)*, Seattle, WA, USA, 2023, pp. 0583-0589, doi: 10.1109/AIIoT58121.2023.10174401.