Controller of Certifying Authorities
Ministry of Electronics & Information Technology
Government of India

IEEE COMPUTER SOCIETY Bangalore Chapter

IAS IEEE INDUSTRY APPLICATIONS SOCIETY Linking Research to Practice Bangalore Chapter

4TH INTERNATIONAL CONFERENCE ON

PUBLIC KEY INFRASTRUCTURE AND ITS APPLICATIONS (PKIA 2023)

SEPTEMBER 8-9TH, 2023

FPGA Implementation of the AES Algorithm with Lightweight LFSR-Based Approach and Optimized Key Expansion

1st Samruddhi Purohit
Dept. of Electronics and Telecomm.
COEP Technological University
Pune, India
purohitsu19.extc@coeptech.ac.in

2nd Vaishnavi Deshpande
Dept. of Electronics and Telecomm.
COEP Technological University
Pune, India
deshpandevm19.extc@coeptech.ac.in

3rd Dr. Vaishali Ingale
Dept. of Electronics and Telecomm.
COEP Technological University
Pune, India
vvi.extc@coeptech.ac.in

Agenda

- Introduction
- Literature Survey
- Objectives
- Proposed Methodology
- Experimental Setup
- Results and Comparison
- References

Introduction

- In Information Age, the need for protecting information is more pronounced than ever.
- **Secure Communication** over wired and/or wireless internet.
- Cryptography : A basic necessity of communication and data exchange when it comes to **confidentiality , integrity, identification** and **authentication** of data between the exchanges.
- **Advanced Encryption Standard (AES)** : **Symmetric** encryption algorithm widely adopted due to its robust security, resistance to various cryptographic attacks.
- This work proposes an efficient implementation of AES algorithms viz. AES-128, AES-192 and AES-256 on **Intel DE-10 Lite FPGA** development board.
- The proposed LFSR-based approach uses the LFSRs for the Substitute bytes stage of the AES algorithm. Additionally, this work also proposes a round-wise key expansion algorithm for storage efficiency.

AES Variant	Key Size	Block Size	Number of Rounds	Round Key Size	Number of Possible Keys
AES-128	128 bits	128 bits	10	44 words	3.4×10^{38}
AES-192	192 bits	128 bits	12	52 words	6.2×10^{57}
AES-256	256 bits	128 bits	14	60 words	1.1×10^{77}

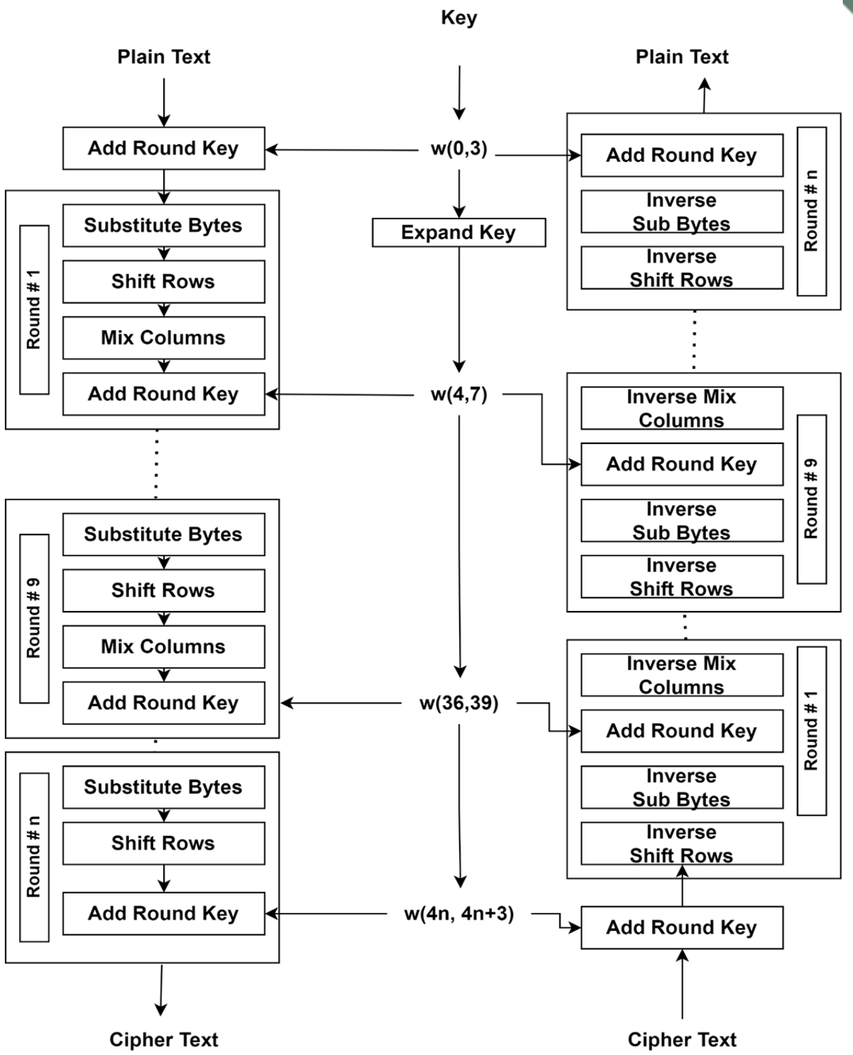
Table I

Literature Survey

Sr. No.	Title	Conclusion
1	Sousi, Ahmad-Loay, Dalia Yehya, and Mohamad Joudi. "Aes encryption: Study & evaluation." (2020).	Sousi et al. compare the old Data Encryption Standard and the Advanced Encryption Standard algorithms.
2	Lu, Zhengyi. "Analysis on AES encryption standard and safety." Third International Symposium on Computer Engineering and Intelligent Communications (ISCEIC 2022). Vol. 12462. SPIE, 2023.	Lu et al. conclude that the AES algorithm is resistant to brute force attacks, square attacks and differential cryptanalysis attacks. The authors propose the method of using temporal redundancy and changing the AES cycle, which effectively detects fault attacks.
3	F. J. D'souza and D. Panchal, "Advanced encryption standard (AES) security enhancement using hybrid approach," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017, pp. 647-652, doi: 10.1109/CCAA.2017.8229881.	D'Souza et al. propose a new approach for the AES algorithm which employs dynamic key generation and dynamic S-box generation. The authors conclude that this increases the confusion and diffusion in the data, hence increasing the complexity of the algorithm.
4	Buchanan, William J., Shancang Li, and Rameez Asif. "Lightweight cryptography methods." Journal of Cyber Security Technology 1.3-4 (2017): 187-201.	Buchanan et al. conducted an extensive review of widely adopted lightweight cryptography solutions designed for resource-limited devices. The authors analyze the advantages and drawbacks of these solutions to provide a comprehensive understanding of their strengths and limitations.
5	M. M. Wong and M. L. D. Wong, "New lightweight AES S-box using LFSR," 2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Kuching, Malaysia, 2014, pp. 115-120, doi: 10.1109/ISPACS.2014.7024436.	Wong et al. propose a lightweight cryptographic solution for the AES algorithm using the LFSRs and architectural optimization.
6	M. Botta, M. Simek and N. Mitton, "Comparison of hardware and software-based encryption for secure communication in wireless sensor networks," 2013 36th International Conference on Telecommunications and Signal Processing (TSP), Rome, Italy, 2013, pp. 6-10, doi: 10.1109/TSP.2013.6613880.	Botta et al. compare the software and hardware modes of encryption and conclude that hardware cryptography has lower energy consumption as compared to software cryptography. In addition, hardware encryption is faster and more secure than software encryption.
7	S. Kaur and R. Vig, "Efficient Implementation of AES Algorithm in FPGA Device," International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Sivakasi, India, 2007, pp. 179-187, doi: 10.1109/ICCIMA.2007.250.	Kaur et al. conclude that the AES algorithm can be efficiently implemented on an FPGA reconfigurable device. Either architecture optimization or algorithmic optimization, can be focused on, to boost the efficiency as well as the security of the algorithm.

Objectives

- Efficient implementation of AES algorithms viz. AES-128, AES-192 and AES-256 on Intel DE-10 Lite FPGA development board.
- Validating LSFR approach for run-time Substitute-Bytes stage of the AES algorithm.
- Optimized Round-wise Key Expansion algorithm for run-time Key Generation.



Proposed Methodology

1. LFSRs based Substitution Bytes Algorithm

a. In the original proposal of AES, Rijndael used particular irreducible polynomial to generate S-box.

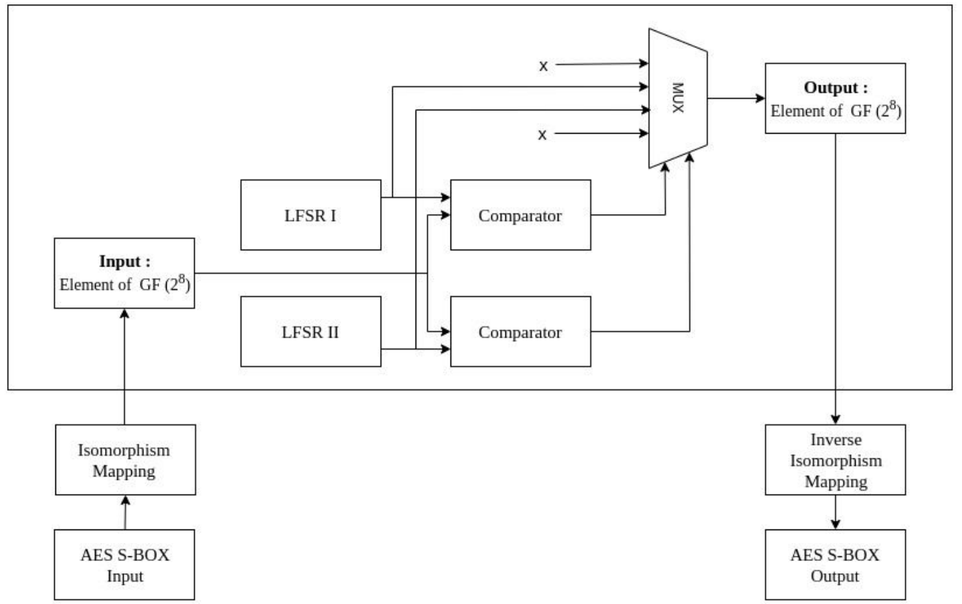
$$m(x) = x^8 + x^4 + x^3 + x + 1$$

a. For LFSRs to generate all unique states in the finite field GF(2^8), the primitive polynomial is chosen.

$$m'(x) = x^8 + x^4 + x^3 + x^2 + 1$$

S-box and inverse S-box construction :

- Find multiplicative inverse of input byte in the finite field GF(2^8) **using LFSRs.**
- Affine transformation.



Proposed Methodology

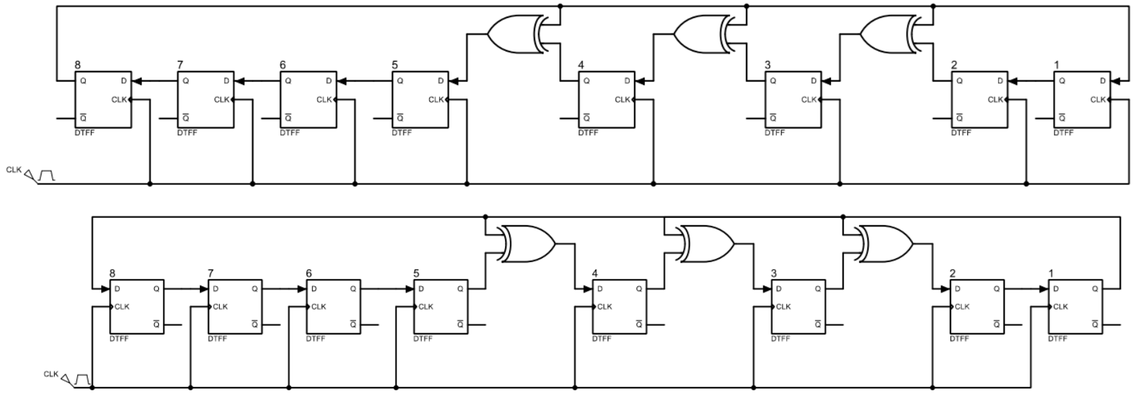
How LFSRs are used to find multiplicative inverse ?

- For 8-bit LFSR generated using primitive polynomial in the finite field GF(2^8):
  - $s(t+1) = T^1 s(t)$ where, T is the LFSR Transformation matrix (8 × 8)
  - s(0) b the initial seed to LFSR. Here, it is 1.
  - Repeats every 2^n -1 states.

    So, $s(t+2^n-1) = s(t)$

    $T^{(2^n - 1)} s(t) = s(t)$

    $T^{(2^n - 1)} = I$ (Identity element)



- Suppose we have to find a multiplicative inverse of s(p).
  - $s(p) = T^p s(0)$
  - $T^p T^{(2^n - 1 - p)} = I$; so, $s(2^n-1-p)$ will be the multiplicative inverse of s(p).
  - Thus, to find multiplicative inverse, we have to find p', such that $p + p' = 2^n - 1$.

- The above process requires 255 cycles to find multiplicative inverse. But we a using 2 LFSRs which will run in opposite directions. Thus, the multiplicative inverse can be found in 127 cycles.

Proposed Methodology

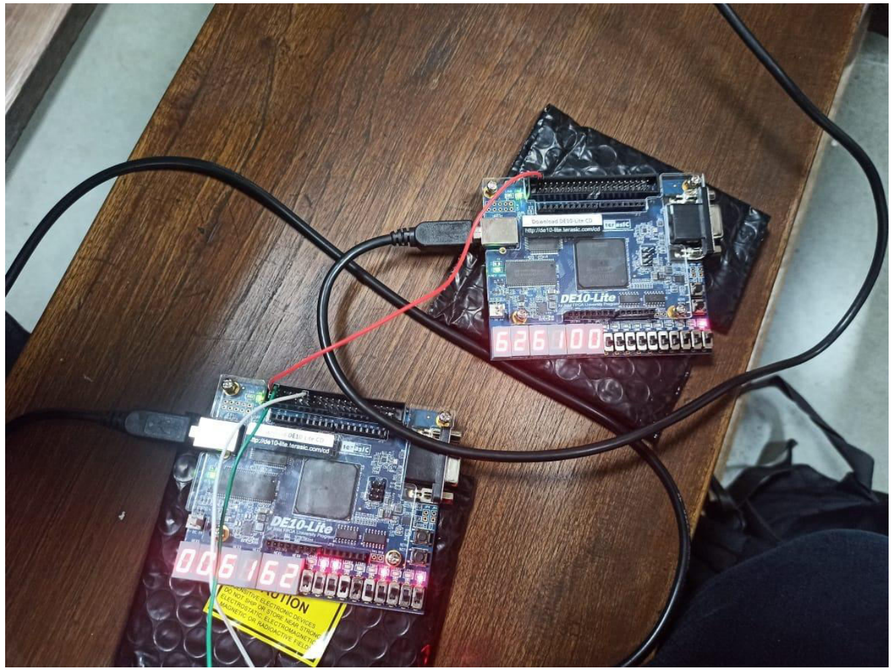
2. Optimized Round-wise Key Expansion Algorithm

- The conventional algorithm of key-scheduling in AES-128, AES-192 and AES-256 algorithms is to generate all the 'n' keys required for all rounds of AES algorithm at once at the start.
- It consumes a lot of memory to store all the keys.
- The idea is to generate the key at the start of every round, just prior to the requirement and reduce the memory storage.
- According to Table. I, in all types of AES algorithms the round key is 128 bits long.

- AES-128
  - 4-byte key is generated in each round.
  - The input key is w_0 to w_3.
  - The round key is :
    - If i mod 4 ≠ 0, $w_i = w_{(i-1)} + w_{(i-4)}$; otherwise $w_i = w_{(i-4)} + f(w_{(i-1)})$
- AES-192
  - 6-byte key is generated in each round.
  - The input key is w_0 to w_5.
  - The round key is :
    - If i mod 6 ≠ 0, $w_i = w_{(i-1)} + w_{(i-6)}$; otherwise $w_i = w_{(i-6)} + f(w_{(i-1)})$
- AES-256
  - 8-byte key is generated in each round.
  - The input key is w_0 to w_7.
  - The round key is :
    - If i mod 8 ≠ 0, $w_i = w_{(i-1)} + w_{(i-8)}$; otherwise $w_i = w_{(i-8)} + f(w_{(i-1)})$
    - If i mod 4 = 0, but I mod 8 ≠ 0, $w_i = w_{(i-8)} + g(w_{(i-1)})$

Experimental Setup

- The input plain text and the key is transmitted to FPGA-I using a serial communication protocol, Universal Asynchronous Receiver Transmission (UART), between the board and the host PC.
- The encryption algorithm of AES-128, AES-192, and AES-256 implemented on FPGA-I encrypts the input plain text using the key.
- The generated ciphertext and key is transmitted to FPGA-II.
- FPGA-II decrypts the ciphertext again to input plain text. Then, the original data is sent to the host PC using UART.



Results and Comparison

- The resulting design has been proven to use less hardware and to have reduced routing complexity, making it appropriate for small embedded devices.
- From the table of comparison, the device utilization in the implementation of the AES-128 algorithm has undergone a substantial reduction, decreasing from 91% to 29%.
- Implementation of AES-192 and AES-256 algorithms using the conventional S-Box approach exceeded the hardware limit of Intel DE-10 Lite FPGA development board. The proposed approach reduces the device utilization for AES-192 and AES-256 algorithms to 39% and 34% respectively.

Decryption						
	Conventional AES algorithm			Proposed AES algorithm		
	Total Combinational functions	Dedicated logic registers	Device utilization per FPGA	Total Combinational functions	Dedicated logic registers	Device utilization per FPGA
AES-128	46,511 /49,760 (93.5%)	312 /49,760 (1%)	46,560 /49,760 (93.5%)	15,259 /49,760 (31%)	3,279 /49,760 (7%)	15,494 /49,760 (31%)
AES-192	-	-	-	17,416 /49,760 (35%)	4,479 /49,760 (9%)	16,756 /49,760 (34%)
AES-256	-	-	-	20,900 /49,760 (42%)	4,884 /49,760 (10%)	19,904 /49,760 (40%)

Encryption						
	Conventional AES algorithm			Proposed AES algorithm		
	Total Combinational functions	Dedicated logic registers	Device utilization per FPGA	Total Combinational functions	Dedicated logic registers	Device utilization per FPGA
AES-128	45,327 /49,760 (91%)	340 /49,760 (1%)	45,368 /49,760 (91%)	14,137 /49,760 (28%)	3,540 /49,760 (7%)	14,295 /49,760 (29%)
AES-192	-	-	-	16,466 /49,760 (33%)	3,988 /49,760 (8%)	16,756 /49,760 (34%)
AES-256	-	-	-	19,087 /49,760 (38%)	4,884 /49,760 (10%)	19,389 /49,760 (39%)

References

- Varghese, Fredy, and P. Sasikala. ”A Detailed Review Based on Secure Data Transmission Using Cryptography and Steganography.” Wireless Personal Communications 129.4 (2023): 2291-2318.
- Bhagat, Vijesh, et al. ”Lightweight cryptographic algorithms based on different model architectures: A systematic review and futuristic applications.” Concurrency and Computation: Practice and Experience 35.1 (2023): e7425.
- Abdullah, Ako Muhamad. ”Advanced encryption standard (AES) algorithm to encrypt and decrypt data.” Cryptography and Network Security 16.1 (2017): 11.
- Hasija, Taniya, et al. ”A Survey on Performance Analysis of Different Architectures of AES Algorithm on FPGA.” Modern Electronics Devices and Communication Systems: Select Proceedings of MEDCOM 2021 (2023): 39-54.
- Mohammed, Abdalbasit, and Nurhayat Varol. ”A review paper on cryptography.” 2019 7th International Symposium on Digital Forensics and Security (ISDFS). 2019.
- Sousi, Ahmad-Loay, Dalia Yehya, and Mohamad Joudi. ”Aes encryption: Study evaluation.” (2020).
- Lu, Zhengyi. ”Analysis on AES encryption standard and safety.” Third International Symposium on Computer Engineering and Intelligent Communications (ISCEIC 2022). Vol. 12462. SPIE, 2023.
- Sinha, Abhishek Kumar, and N. Jayaraj. ”Performance Analysis of AES Cryptographic Algorithm.” NCRTS (2015).
- F. J. D’souza and D. Panchal, ”Advanced encryption standard (AES) security enhancement using hybrid approach,” 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017, pp. 647-652, doi: 10.1109/CCAA.2017.8229881.
- M. Botta, M. Simek and N. Mitton, ”Comparison of hardware and software-based encryption for secure communication in wireless sensor networks,” 2013 36th International Conference on Telecommunications and Signal Processing (TSP), Rome, Italy, 2013, pp. 6-10, doi: 10.1109/TSP.2013.6613880.
- Rodríguez-Henríquez, Francisco, et al. Cryptographic algorithms on reconfigurable hardware. Springer Science Business Media, 2007. S. Kaur and R. Vig, ”Efficient Implementation of AES Algorithm in FPGA Device,” International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)

Controller of Certifying Authorities
Ministry of Electronics & Information Technology
Government of India

G20 भारत 2023 INDIA

THANK YOU

भगत संगणन विकास केंद्र
सीडैक CDAC
॥ज्ञानादेव तु कैवल्यम्॥
CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

IEEE COMPUTER SOCIETY
Bangalore Chapter

IAS IEEE INDUSTRY APPLICATIONS SOCIETY
Linking Research to Practice
Bangalore Chapter