

# **HIGH THROUGHPUT HARDWARE ARCHITECTURE FOR LIGHT WEIGHT CRYPTOGRAPHY**

**Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Engineering degree in Electronics and Communication Engineering**

**by**

**M SURAJ (40130129)**

**P VAMSI KRISHNA (40130163)**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING  
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

## **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMED TO BE UNIVERSITY)**

**CATEGORY -1 UNIVERSITY BY UGC**

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE  
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

**APRIL - 2024**



**SATHYABAMA**

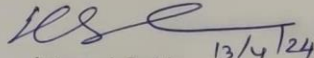
INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE  
www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **P VAMSI KRISHNA (40130163)** and **M SURAJ (40130129)** who carried out the project entitled "**HIGH THROUGHPUT HARDWARE ARCHITECTURE FOR LIGHT WEIGHT CRYPTOGRAPHY**" under my supervision from NOV 2023 to APRIL 2024.

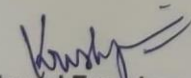
  
Internal Guide 13/4/24

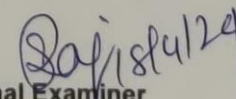
**Dr. S. EMALDA ROSLIN, M.E., Ph.D.**

  
Head of the Department

**Dr. T. RAVI, M.E., Ph.D.**

Submitted for Viva voce Examination held on 18/04/24

  
Internal Examiner

  
External Examiner



**SATHYABAMA**

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE  
www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **M SURAJ (40130129)** and **P VAMSI KRISHNA (40130163)** who carried out the project entitled "**HIGH THROUGHPUT HARDWARE ARCHITECTURE FOR LIGHT WEIGHT CRYPTOGRAPHY**" under my supervision from NOV 2023 to APRIL 2024.

  
Internal Guide 13/4/24

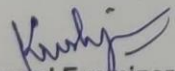
**Dr. S. EMALDA ROSLIN, M.E., Ph.D.**

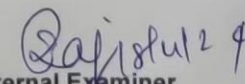


Head of the Department

**Dr. T. RAVI, M.E., Ph.D.**

Submitted for Viva voce Examination held on 18/04/24

  
Internal Examiner

  
External Examiner

## **DECLARATION**

We **M. SURAJ (40130129)** and **P. VAMSI KRISHNA (40130163)**, hereby declare that the Project Report entitled **HIGH THROUGHPUT HARDWARE ARCHITECTURE FOR LIGHT WEIGHT CRYPTOGRAPHY** is done under the guidance of **Dr. S. EMALDA ROSLIN, M.E., Ph.D.** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Electronics and Communication Engineering**.

**DATE:**

**PLACE:**

**SIGNATURE OF THE CANDIDATES**

**1.**

**2.**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. N.M. NANDHITHA, M.E., Ph.D.**, Prof. & Dean, School of Electrical and Electronics and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department of **Electronics and Communication Engineering** for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. EMALDA ROSLIN, M.E., Ph.D.**, her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project

## ABSTRACT

The high throughput hardware architecture designed for lightweight cryptography using Artix7 FPGA board. We focusing on the implementation of the Dynamic PRESENT cipher on Artix-7 FPGA board. Lightweight cryptography is essential for resource-constrained applications such as iot devices and medical devices. Where efficient encryption and decryption plays the crucial role. The PRESENT cipher is well known for its simplicity and strong security and it is well-suited for such applications. Our proposed architecture leverages the parallelism as well as the pipelining capabilities of the Artix-7 FPGA to achieve the high throughput by maintaining the low area and power consumption. We provided a detailed analysis of the design considerations including key scheduling, round operations and resource utilization. The architecture includes a low latency 64-bit data path and key schedule design. We aiming to improve performance for IoT and Medical applications. The proposed architecture offers significant improvements over the existing designs, with enhanced throughput, reduced latency, and increased power consumption. Experimental results demonstrate the effectiveness of our architecture in achieving a balance between performance, area, and power efficiency for lightweight cryptographic applications in IOT and medical fields.

The proposed design leverages the inherent parallelism and reconfigurability of FPGAs to optimize the cryptographic operations required for lightweight algorithms. We focus on lightweight ciphers such as PRESENT, which have demonstrated strong security properties while being highly efficient in terms of resource utilization. To ensure low-latency performance, we employ pipeline architectures and hardware acceleration techniques, allowing for rapid data encryption and decryption. Additionally, we address the power constraints commonly found in IoT devices, optimizing power consumption while maintaining high throughput.

## TABLE OF CONTENTS

| CHAPTER No. | TITLE                                                            | PAGE No.    |
|-------------|------------------------------------------------------------------|-------------|
|             | <b>ABSTRACT</b>                                                  | <b>V</b>    |
|             | <b>LIST OF FIGURES</b>                                           | <b>VIII</b> |
|             | <b>LIST OF TABLES</b>                                            | <b>IX</b>   |
|             | <b>LIST OF ABBREVIATIONS</b>                                     | <b>X</b>    |
| <b>1.</b>   | <b>INTRODUCTION</b>                                              | <b>1</b>    |
|             | 1.1 LIGHT WEIGHT CRYPTOGRAPHY                                    | 2           |
|             | 1.2 COVENTIONAL OF CRYPTOGRAPHIC<br>ALGORITHM                    | 4           |
|             | 1.3 PROBLEM STATEMENT                                            | 6           |
|             | 1.4 OBJECTIVES                                                   | 8           |
| <b>2.</b>   | <b>OVERVIEW OF LITERATURE</b>                                    | <b>10</b>   |
|             | 2.1 EXISTING CRYPTOGRAPHIC<br>ALGORITHM                          | 10          |
|             | 2.2 SUMMARY OF LITERATURE                                        | 14          |
| <b>3.</b>   | <b>PROPOSED DYNAMIC LIGHT WEIGHT<br/>CRYPTOGRAPHY(D-PRESENT)</b> | <b>17</b>   |
|             | 3.1 BLOCK DIAGRAM OF PROPOSED<br>ALGORITHM                       | 17          |
|             | 3.2 ARCHITECTURE OF PROPOSED<br>ALGORITHM                        | 19          |
|             | 3.3 S-BOX, P-BOX FOR D-PRESENT                                   | 20          |
|             | 3.4 STANDARDS                                                    | 22          |
|             | 3.5 CONSTRAINTS                                                  | 23          |

| <b>CHAPTER No.</b> | <b>TITLE</b>                                                 | <b>PAGE No.</b> |
|--------------------|--------------------------------------------------------------|-----------------|
|                    | 3.6 TRADEOFFS                                                | 24              |
| <b>4</b>           | <b>RESULTS AND DISCUSSION</b>                                | <b>27</b>       |
|                    | 4.1 IMPLEMENTATION OF S-LAYER FOR D-PRESENT                  | 28              |
|                    | 4.2 IMPLEMENTATION OF P-LAYER FOR D-PRESENT                  | 28              |
|                    | 4.3 IMPLEMENTATION OF UNIVERSAL SHIFT REGISTER FOR A-PRESENT | 29              |
|                    | 4.4 DESIGN CONSIDERATIONS FOR THE PROPOSED FOR D-PRESENT     | 29              |
|                    | 4.5 COMPARISION OF EXISTING AND D-PRESENT                    | 33              |
| <b>5</b>           | <b>CONLUSION AND FUTURE WORK</b>                             | <b>39</b>       |
|                    | <b>REFERENCES</b>                                            | <b>41</b>       |
|                    | <b>PUBLICATION</b>                                           | <b>44</b>       |

## **LIST OF FIGURES**

| <b>FIGURE NO.</b> | <b>TITLE</b>                                       | <b>PAGE NO.</b> |
|-------------------|----------------------------------------------------|-----------------|
| <b>3.1</b>        | <b>PRESENT Cipher algorithm</b>                    | <b>18</b>       |
| <b>3.2</b>        | <b>Block diagram of PRESENT Cipher</b>             | <b>19</b>       |
| <b>3.3</b>        | <b>S-box Synthesized Design</b>                    | <b>20</b>       |
| <b>3.4</b>        | <b>P-box Synthesized Design</b>                    | <b>21</b>       |
| <b>4.1</b>        | <b>S-layer Simulation output</b>                   | <b>28</b>       |
| <b>4.2</b>        | <b>P-layer Simulation output</b>                   | <b>28</b>       |
| <b>4.3</b>        | <b>Universal shift registers simulation output</b> | <b>29</b>       |
| <b>4.4</b>        | <b>Synthesized Design of D-PRESENT</b>             | <b>29</b>       |
| <b>4.5</b>        | <b>Generation of Waveform</b>                      | <b>30</b>       |
| <b>4.6</b>        | <b>Generation of Waveform</b>                      | <b>30</b>       |
| <b>4.7</b>        | <b>Plaintext to Cipher Output</b>                  | <b>34</b>       |
| <b>4.8</b>        | <b>Trade-off between LUT &amp; Throughput</b>      | <b>35</b>       |
| <b>4.9</b>        | <b>Trade-off between Latency &amp; Throughput</b>  | <b>36</b>       |

## LIST OF TABLES

| TABLE NO. | TITLE                              | PAGE NO. |
|-----------|------------------------------------|----------|
| 4.1       | Comparison of Resource Utilization | 33       |
| 4.2       | Test Vectors                       | 37       |

## LIST OF ABBREVIATIONS

|      |   |                                        |
|------|---|----------------------------------------|
| AES  | - | Advanced Encryption Standard           |
| ASIC | - | ApplicationSpecific Integrated Circuit |
| DES  | - | Data Encryption Standard               |
| IoT  | - | Internet of Things                     |
| FPGA | - | Field-Programmable Gate Array          |
| CPU  | - | Central Processing Unit                |
| RAM  | - | Random Access Memory                   |
| ROM  | - | Read-Only Memory                       |
| ALU  | - | Arithmetic Logic Unit                  |
| GUI  | - | Graphical User Interface               |
| API  | - | Application Programming Interface      |
| UI   | - | User Interface                         |
| I/O  | - | Input/Output                           |
| DMA  | - | Direct Memory Access                   |
| MB/s | - | Megabytes per second                   |
| MHz  | - | Megahertz                              |
| GPU  | - | Graphics Processing Unit               |
| SDK  | - | Software Development Kit               |
| SoC  | - | System on a Chip                       |
| RNG  | - | Random Number Generator                |
| SHA  | - | Secure Hash Algorithm                  |
| USB  | - | Universal Serial Bus                   |

# CHAPTER 1

## INTRODUCTION

The Lightweight cryptography, designed to offer robust security with minimal computational overhead, stands at the forefront of addressing these contemporary security challenges. The significance of lightweight cryptography is further underscored by its suitability for resource-constrained platforms, such as IoT devices, where power efficiency and throughput are paramount.

The primary objective of this work is to present a high-throughput hardware architecture designed to advance the field of lightweight cryptography. Our research focuses on the transformation of a 64-bit input into two 16-bit output and four 8-bit outputs a pivotal component in various cryptographic algorithms and applications. This architectural innovation not only enhances cryptographic security but also streamlines the processing of sensitive data, a necessity in modern applications ranging from secure communication to data protection.

Efforts in the development of lightweight cryptography have traditionally concentrated on balancing security with performance, which becomes particularly challenging in resource-constrained environments. Our research takes this challenge head-on by delivering a hardware architecture that not only upholds the stringent security requirements of cryptographic algorithms but also excels in terms of throughput, making it well-suited for real-time applications.

In the sections that follow, we delve into the intricate design details of our proposed architecture, elucidating its functionality, efficiency, and security guarantees.

Through rigorous evaluation and analysis, we aim to substantiate the advantages of our design in terms of both its cryptographic robustness and computational efficiency. Furthermore, we explore potential applications for our architecture, ranging from securing IoT devices to enhancing the performance of secure communication protocols.

Hardware implementations of NIST lightweight cryptographic candidates are essential for the Internet of Things (IoT) to ensure security, as highlighted in. The paper provides the first intensive validation of the LWC HW API and DP on a wide range of candidates,

including AEAD and hash. The authors compare their implementations with selected past and present authenticated cipher implementations and show examples of resource-intensive features that reduce performance to help guide later-round tweaks.

In addition to the PRESENT cipher, the HIGHT block cipher is another example of lightweight cryptography with low-cost and high-throughput hardware implementations, as discussed in. A hardware architecture for Ascon, a NIST Lightweight cryptography standard, has also been proposed to enable high-throughput and low-latency operations, as presented. In summary, high-throughput and lightweight hardware architectures are essential for efficient and secure lightweight cryptography. The PRESENT and HIGHT ciphers are two examples of lightweight cryptography with low-cost and high-throughput hardware implementations. Hardware implementations of NIST lightweight cryptographic candidates are also crucial for the IoT to ensure security.

## **1.1 LIGHTWEIGHT CRYPTOGRAPHY**

Lightweight cryptography is a specialized cryptographic algorithm or protocol designed for implementation in resource-constrained environments like RFID tags, sensors, and IoT devices. It focuses on efficient use of computational resources while providing adequate security. NIST initiated a project for the standardization of lightweight cryptographic algorithms, aiming to create a portfolio of lightweight algorithms through an open process. These algorithms are tailored for constrained environments and include block ciphers, authenticated encryption schemes, hash functions, message authentication codes, cryptographic permutations, and stream ciphers. The key design considerations for lightweight cryptography include security strength, flexibility for efficient implementations across platforms, and low overhead for multiple functions. Notably, lightweight cryptography methods aim to use less memory, computing resources, and power supply, making them suitable for a wide range of devices with limited resources. Additionally, the adoption of lightweight cryptography in IoT applications can enhance privacy, integrity, and authenticity of data exchanges, safeguarding against potential cyber threats and ensuring the overall security of IoT networks. As the IoT landscape continues to evolve, the importance of lightweight cryptography in enabling secure and

efficient communication across diverse IoT devices becomes increasingly evident, highlighting its significance in the realm of modern cybersecurity.

When designing lightweight cryptography solutions, several trends are noticeable. First, most lightweight algorithms are designed to use smaller internal states, short block and key sizes, which leads to smaller RAM consumption and better processing of smaller messages. Second, the number of operations in symmetric lightweight cryptography roughly doubles when the input size of a symmetric-key increases. Lastly, lightweight cryptography is driven by applications and should be designed to apply new academic insights and best match existing protocols.

Lightweight algorithms are optimized for performance on constrained devices, aiming to minimize computational overhead, memory usage, and energy consumption. Despite their efficiency, lightweight cryptographic primitives must maintain a high level of security, ensuring resistance against common cryptographic attacks. Algorithms designed for lightweight environments often prioritize compactness in terms of code size and data representation, making them suitable for devices with limited storage capacity. Lightweight cryptography should be scalable to accommodate a wide range of applications and deployment scenarios, from low-power sensors to embedded systems and IoT networks. These algorithms should be adaptable to diverse use cases and environments, supporting various cryptographic operations such as encryption, authentication, key exchange, and secure hashing. Lightweight block ciphers like PRESENT, LED, and SIMON are designed for efficient encryption and decryption on low-power devices. Lightweight hash functions such as SipHash and Quark are optimized for performance and compactness, suitable for applications like message authentication and integrity verification. Lightweight stream ciphers like Grain and Trivium offer efficient and secure encryption for constrained environments. Lightweight public-key cryptography schemes, such as NTRUEncrypt and NTRUSign, provide alternatives to traditional RSA and ECC algorithms, offering a balance between security and efficiency. Lightweight authentication protocols like HMAC (Hash-based Message Authentication Code) and lightweight variants of digital signatures ensure data integrity and authenticity in resource-constrained environments.

The security of lightweight cryptography is crucial, especially for IoT devices and wireless sensor networks. Lightweight cryptography requires less RAM, fewer computing resources, and less power supply to resolve data security challenges of IoT sensor networks. It is simpler and more efficient than typical cryptography at securing connected devices in particular. The benefits of lightweight cryptography include demanding fewer resources from devices and taking less time to complete essential operations, making it a more practical solution for securing the IoT. Lightweight cryptography is a crucial aspect of securing resource-constrained devices and IoT networks. By prioritizing security and efficiency, lightweight cryptography provides a viable solution for ensuring secure communication and data protection in the rapidly expanding IoT landscape.

## **1.2 COVENTIONAL OF CRYPTOGRAPHIC ALGORITHM**

The convention of a cryptographic algorithm involves key elements like plaintext, encryption algorithm, secret key, ciphertext, and decryption algorithm. In a conventional encryption scheme, the sender uses the receiver's secret key to encrypt the plaintext, creating ciphertext that can only be decrypted by the receiver using the same secret key. This process relies on a strong encryption algorithm and secure key management between the sender and receiver to ensure the confidentiality of the message. The use of conventional encryption offers simplicity, efficiency in resource utilization, and faster processing compared to asymmetric key encryption, but it lacks the ability to guarantee the origin and authenticity of messages and is less secure than public-key encryption. Understanding the intricacies of conventional cryptographic algorithms underscores the significance of secure key management for effective encryption and decryption processes, highlighting both the advantages and limitations of this encryption approach.

In addition to the technical aspects, the convention of cryptographic algorithms plays a crucial role in ensuring data security and privacy across various digital communication channels. By employing conventional encryption techniques, organizations and individuals can safeguard sensitive information from unauthorized access and maintain the integrity of their data exchanges. The reliance on secure key management and encryption processes not only protects confidential data but also fosters

trust in digital interactions. Understanding and implementing cryptographic algorithms effectively can enhance cybersecurity measures, mitigate risks associated with data breaches, and uphold the confidentiality of communications in an increasingly interconnected world. Embracing the principles of cryptographic conventions empowers individuals and entities to fortify their digital defenses and uphold the confidentiality and integrity of their information exchanges.

This involves replacing plaintext or intermediate values with different values according to a predefined substitution table or function. Substitution helps in introducing confusion and non-linearity into the algorithm, making it harder to analyze. Permutation rearranges the order of bits or bytes in the data stream. This operation ensures diffusion, spreading the influence of individual input bits across multiple output bits. It helps in breaking any patterns or correlations present in the data. such as matrix multiplication or bitwise operations like XOR, are often employed to mix the data further. These operations provide diffusion and help in achieving avalanche effect, where a small change in input leads to a significantly different output. Lightweight algorithms typically use key mixing operations to incorporate the secret key into the encryption process. This can involve XORing the key with the plaintext or using it to modify the algorithm's internal state. Lightweight cryptographic algorithms often employ iterative structures where the output of one round serves as the input to the next round.

This iterative process enhances security by repeatedly applying the convolution operations. The design of the round function plays a crucial role in the convolution process. It should balance between security, performance, and hardware/software implementation constraints. The round function typically combines various operations in a carefully crafted manner to achieve the desired security properties. Throughout the convolution process, emphasis is placed on introducing non-linearity and diffusion. Non-linearity ensures that the relationship between the input and output is complex, making cryptanalysis more difficult. Diffusion ensures that changes in the input propagate throughout the output, preventing local changes from being localized.

In cryptography typically involves a secret key and a plaintext message. The plaintext message is first transformed into a signal, which can be a sequence of bits or bytes. The secret key is also transformed into a signal, which is then convolved with the plaintext signal to produce a ciphertext signal. The convolution process involves sliding the secret key signal over the plaintext signal, element-wise multiplying the two signals, and summing the results. The resulting ciphertext signal is then transmitted to the recipient, who can decrypt it by performing a deconvolution operation using the same secret key.

Convolution-based cryptographic algorithms can provide strong security and resistance to various attacks, such as brute-force attacks and statistical attacks. However, they can also be computationally expensive and require large amounts of memory, especially for long secret keys and large plaintext messages. Therefore, it's essential to balance security and performance when designing convolution-based cryptographic algorithms.

### **1.3 PROBLEM STATMENT**

In an increasingly interconnected world, the exchange of digital information has become an integral part of daily life. This exchange spans a multitude of applications, ranging from secure communication in the Internet of Things (IoT) to data protection in smart healthcare devices. The common thread in these applications is the critical need for data security and privacy preservation.

The Lightweight cryptography has risen to prominence as a pivotal solution for addressing these security requirements, offering cryptographic algorithms optimized for resource-constrained devices. While lightweight cryptographic algorithms have garnered attention for their ability to provide robust security with minimal resource consumption, a significant challenge persists: the efficient execution of cryptographic operations without compromising security.

The problem we address in this research is two-fold:

**a. Security and Efficiency Balance:** Traditional cryptographic algorithms such as the Advanced Encryption Standard (AES) may offer robust security but are computationally demanding and memory-intensive. In contrast, lightweight cryptographic algorithms aim to minimize resource usage, but often at the cost of computational efficiency. Striking a balance between security and efficiency remains a formidable challenge, particularly in lightweight cryptography, where devices are inherently resource-constrained.

**b. Throughput in Lightweight Cryptography:** The efficient execution of cryptographic operations, particularly in real-time and data-intensive applications, has become an acute concern. The throughput of cryptographic processes, specifically the transformation of a 128-bit input into a 16-bit output and two 8-bit outputs, is the focal point of this research. The problem we seek to resolve is how to design a high-throughput hardware architecture that seamlessly integrates robust data security and computational efficiency for this critical data transformation.

The challenge is twofold: maintaining a high level of data security, including resistance to known cryptographic attacks and vulnerabilities, while ensuring that cryptographic operations proceed with minimal computational overhead and high throughput.

Our research addresses this problem by proposing an innovative high-throughput hardware architecture for lightweight cryptography that aims to resolve these challenges and open the doors to a new era of secure, efficient, and practical data exchange across diverse applications.

This statement of the problem provides a clear understanding of the challenges and issues your research seeks to tackle, setting the stage for the research objectives and contributions that follow in your paper. Be sure to customize it to align with the specific focus and goals of your research.

One approach to addressing this problem is to use lightweight cryptographic algorithms that are optimized for high throughput hardware architectures. For example,

the PRESENT cipher algorithm is suitable for high throughput lightweight IoT applications. A low latency 32-bit data path architecture was proposed for the PRESENT cipher algorithm to achieve high throughput. Similarly, a compact hardware implementation for the SCA-resistant PRESENT cipher was proposed to improve its security and performance.

Another approach is to use hardware architectures that are optimized for lightweight cryptographic algorithms. For example, a hardware architecture for unifying four HB-type lightweight authentication protocols was proposed and implemented.

Hardware limitations are also a significant challenge in designing high throughput hardware architectures for lightweight cryptography. For example, the SNOW-3G architecture achieves a relatively high throughput for a lower frequency than the rest, classifying it as suitable for IoT devices that only operate at low frequencies. A compact hardware implementation for the SCA-resistant PRESENT cipher was also proposed to improve its security and performance.

## 1.4 OBJECTIVES

The primary objectives of this research are as follows:

- a. Develop a High-Throughput Hardware Architecture:** Design and implement a high-throughput hardware architecture that efficiently transforms a 128-bit input into one 16-bit output and two 8-bit outputs, while maintaining robust data security.
- b. Achieve Efficient Data Transformation:** Optimize the data transformation process to ensure minimal computational overhead and high throughput, enabling real-time processing of cryptographic operations in resource-constrained environments.
- c. Ensure Cryptographic Security:** Implement and evaluate security measures to safeguard against known cryptographic attacks and vulnerabilities, ensuring that the hardware architecture provides the necessary level of data security.

**d. Evaluate and Benchmark Performance:** Conduct thorough performance evaluations to assess the efficiency and throughput of the proposed hardware architecture. Compare results existing solutions and benchmarks to demonstrate the advancements achieved.

**e. Explore Practical Applications:** Investigate practical applications for the high-throughput hardware architecture in various domains, including IoT, secure communication, and data protection, showcasing its potential to address real-world challenges.

**f. Provide Comprehensive Documentation:** Offer detailed documentation of the hardware architecture's design, implementation, and testing procedures, making it accessible to the broader research and engineering community.

By pursuing these objectives, we aim to contribute to the field of lightweight cryptography by addressing the challenge of efficiently transforming data while maintaining robust security, and in doing so, we aspire to facilitate secure data exchange in resource-constrained environments and beyond.

## CHAPTER 2

### OVERVIEW OF LITERATURE

#### 2.1 EXISTING LIGHT WEIGHT CRYPTOGRAPHIC ALGORITHM

The Physical-Layer-Security Box is a concept for time-frequency channel-reciprocity key generation. Marco Zoli, Andre Noll Barreto, Stefan Kopsell, Padmanava Sen and Gerhard Fettweis, (2022) their goal is to show a wide applicability of our PLS-Box to actual wireless systems, paving the way for an evolution of existing schemes. After an initial overview of the security landscape of today, we have outlined the motivations of Physical-Layer Security, providing a summary of the state of the art of this promising field. It discussed the open issues and challenges of this concept, such as RF impairments, accessibility to the PHY baseband modem, and attacker capabilities. Additionally, two PLS-Box filter-bank examples have been described, in line with today's OFDM and UWB systems, showing the suitability and flexibility of our solutions. This work opens up new possibilities for enhancing the security of wireless communications, ensuring that practical implementations align with the theoretical promise of PLS. It marks a significant step towards making secure wireless communication more accessible, efficient, and robust.[1]

Transparency order versus confusion coefficient it is a case study of NIST lightweight cryptography S-Boxes by Huizhong Li, Guang Yang, Jingdian Ming, Yongbin Zhou and Chengbin Jin, (2022). In this the resistance of cryptographic implementations against side-channel attacks (SCAs) has been recognized as a crucial factor. In order to measure side-channel resistance of S-Boxes, three theoretical metrics are proposed and they are revisited transparency order (VTO), confusion coefficients variance (CCV), and minimum confusion coefficient (MCC), respectively. It takes the S-Boxes used in NIST Lightweight Cryptography candidates as concrete examples, we give a comprehensive study of the applicability of three popular theoretical metrics for side-channel analysis, namely VTO, CCV and MCC. Firstly, we find that CCV is almost linearly correlated with VTO, while MCC is inconsistent with the other two metrics. [2]

## Case Study of NIST Lightweight Cryptography S-Boxes:

- a.** NIST's Lightweight Cryptography initiative focuses on cryptographic algorithms suitable for resource-constrained devices like IoT and RFID. S-Boxes play a crucial role in these algorithms.
- b.** Research in this domain often involves the evaluation of S-Boxes' transparency orders and confusion coefficients. The goal is to design S-Boxes that balance simplicity and cryptographic strength.
- c.** Researchers analyze and compare different lightweight cryptography S-Boxes, looking for trade-offs between transparency and confusion. These studies inform the selection and design of S-Boxes for NIST's lightweight cryptography standards

The Light Weight Cryptographic Algorithms for Medical Internet of Things (IoT) – a review, proposed by Damodharan, J.; S. Emalda Roslin, (2020) the performance of those existing algorithms is analyzed in detail based on the constraint and merits of the IoT. This detailed survey is made towards obtaining a tradeoff between security, cost and performance of IoT based application. Internet of things (IoT) is the communication between machine to machine and machine to human. Here smart object can be a computer/Machine/sensor/ RFID tag. The Symmetric algorithm is faster than Asymmetric algorithm. The block cipher is slower than the stream cipher. Efficiency and throughput are increased in Hardware compared with software. FPGA shows better results than other hardware like Microcontroller and ARM processor. On FPGA, Vertex 6 board throughput is very higher than the vertex 4 board. In this they proposed a detailed review on various security algorithms is done. Comparison between the various algorithms is also made in terms of its key size, block size and its performance. This gives an overview on the limitations of the existing security techniques and lays a platform to propose a novel light weight technique with minimum number of block size and key size. [3]

Hardware Implementation of High-Throughput S-Box in AES for Information Security, proposed by Shih-Hsiang Lin, Jun-Yi Lee, Chia-Chou Chuang, Narn-Yih Lee, Pei-Yin Chen et al (2023), Over study presented a high-throughput, seven-stage

hardware pipeline architecture for Sub Byte computations in AES, aiming to enhance information security. The research emphasized the importance of efficient S-Box implementations in AES for achieving high throughput and robust security measures. Additionally, other works have explored FPGA-based implementations, memoryless encryptors, and composite-field accelerators to achieve high-speed AES encryption with optimized area consumption and throughput. Overall, the review on the hardware implementation of High-Throughput S-Box in AES for Information Security highlights the continuous efforts in the research community to develop advanced and secure hardware solutions for AES encryption, focusing on speed, area optimization, and throughput improvements to meet the demands of modern information security applications. [4]

A Hardware Architecture of NIST Lightweight Cryptography Applied in IPsec to Secure High-Throughput Low-Latency IoT Networks is proposed by S. -N. Tran, V. -T. Hoang and D. -H. Bui, (2023) their goal is to show a lightweight elliptic curve scalar multiplication architecture for random Weierstrass curves over prime field  $F_p$ , which is suitable for IoT devices with limited computational resources. This architecture is compared with other existing architectures, and the results show that it achieves better performance in terms of execution time and resource utilization. The importance of hardware-based cryptography in IoT networks, highlighting its advantages over software-based cryptography in terms of performance, security, and energy efficiency. The authors argue that hardware-based cryptography is essential for securing IoT networks, especially in applications that require real-time data transmission and low-latency communication. The architecture is designed to be resistant to side-channel attacks, which are a significant threat to IoT devices. The authors discuss different types of side-channel attacks, such as passive and active attacks, and propose countermeasures to prevent them. [5]

Optimized Implementation of Gift Cipher is proposed by D. Jamuna Rani, (2022) In this they implement the optimization technique of the existing GIFT cipher and throughput is considered as the performance metrics. Pipeline and sub-stage pipeline techniques are used for enhancing the architecture. This can still be improved and optimized in terms of increased data rate and reduced time delay. [6]

The PRESENT cipher is one of the popular cryptosystems used in wireless which falls under the light weight cryptography category. Gift cipher is an enhanced version of PRESENT cipher. implemented the inter and intra pipelined and parallel structure of AES algorithm which resulted in improved performance at the cost of increased complexity. This resulting optimized structure is more suited for smart devices which incorporates more memory and limited in battery usage.

A review of lightweight block ciphers. Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I, (2020) The demand for applicable cryptographic components is therefore strong and growing. However, the limited resources of these devices, in conjunction with the ever-present need for smaller size and lower production costs. Lightweight cryptography is one of the important building blocks of secure embedded systems. To this end, hardware and software implementations of block ciphers were examined.

This existing work proposed an improvement from Lara Nino et al. work with a 32-bit data architecture for the PRESENT cipher algorithm. The 64-bit input is provided to the data engine as 32-bit in each clock cycle. It requires two clock cycles to get the input. The data engine has four paths, in a multiplexer and in a data pipeline register. The multiplexer is used to select the data between input data and round data. The first option in the multiplexer selects data from the primary input, while the second option selects the round data. The data pipeline register consists of four registers, each size of 8-bit Hence four paths with two registers compute 32-bit data. In each cycle, a 32-bit output becomes available for the following process. Execution time, data security, power, and hardware utilization are the four significant problems that should be addressed in the data communication system between intelligent devices. Due to the risks in the implementation algorithm complexity, certain ciphers are unsuitable for IoT applications. In addition, IoT applications are also implemented on an embedded platform wherein computing resources and memory are limited in number. Here in the research work, a reliable lightweight encryption algorithm with PRESENT has been implemented as a hardware accelerator and optimized for medical IoT-embedded applications. Therefore, Overall, proposed low latency 32-bit data path architecture can compute the PRESENT cipher

algorithm more efficiently than the existing architectures. The scientific contribution of the proposed architecture is flexible FPGA architecture; the user can choose to increase the throughput and decrease the latency for PRESENT cipher in the field of cryptography. Future enhancements in the PRESENT algorithm can see the implementation of the reconfigurable architecture, which improves hardware optimization and throughput performance.

## **2.2 SUMMARY OF LITERATURE**

The hardware architecture implementation of the PRESENT cipher highlights various aspects of this lightweight encryption algorithm. Several studies focus on optimizing hardware architectures for the PRESENT block cipher to enhance performance and efficiency in different applications, particularly in the context of IoT. The development of high-throughput hardware architectures for the PRESENT cipher, aiming to achieve low latency and efficient data processing. Studies evaluate the execution time, power consumption, memory usage, and speed of the implemented encryption algorithms, emphasizing the importance of these factors in real-world applications like Medical IoT. Comparative analyses are conducted to assess the performance of encryption algorithms like AES, SPECK, and SIMON, highlighting the suitability of the PRESENT cipher for secure machine-to-machine communication to propose performance-optimized architectures for the PRESENT cipher, focusing on encryption efficiency and speed. It is placed on designing lightweight block ciphers suitable for resource-constrained IoT devices, ensuring secure data exchange and storage in cloud and local systems

The Physical-Layer-Security Box is a concept that focuses on time-frequency channel-reciprocity key generation. This innovative approach aims to bridge the gap between theoretical concepts and practical implementation in the field of Physical Layer Security (PLS). The PLS-Box introduces a novel key generation method called time-frequency filter-bank, which leverages the entropy of the radio channel to create reciprocal security keys at both ends of wireless communication. This concept addresses the pressing need for enhanced security and privacy in wireless systems, especially with

the rise of cyber threats and the increasing vulnerabilities in technologies like the Internet of Things (IoT) and 5G networks.

Overview of hardware architecture and its importance in computing systems. Explanation of how hardware architecture influences system performance, power consumption, and scalability. Explanation of key concepts such as instruction set architecture (ISA), microarchitecture, and system-on-chip (SoC) design. Discussion on the role of hardware description languages (HDLs) like Verilog and VHDL in specifying hardware behavior. Detailed examination of various components of a hardware architecture such as CPU, memory hierarchy, I/O interfaces, and interconnects. The advancement of high-throughput hardware architectures tailored for lightweight cryptography. Our work not only demonstrates the feasibility of maintaining stringent security guarantees in lightweight cryptography but also excels in terms of computational efficiency, a cornerstone of real-time and data-intensive applications.

Exploration of different design choices and trade-offs in selecting components for a specific application or target platform Survey of optimization techniques used to improve performance, power efficiency, and area utilization in hardware designs. Examination of different implementation platforms ranging from field-programmable gate arrays (FPGAs) to application-specific integrated circuits (ASICs) and system-on-chip (SoC) platforms. Comparison of advantages and limitations of each platform in terms of performance, flexibility, and development cost. Case studies highlighting real-world examples of hardware architecture implementations in various domains such as embedded systems, consumer electronics, networking, and high-performance computing. Analysis of architectural choices and design trade-offs in these case studies. Exploration of current challenges in hardware architecture implementation such as increasing complexity, power constraints, and security vulnerabilities. Discussion on emerging trends and future directions in hardware architecture design including novel architectures, heterogeneous computing, and quantum computing.

The PLS-Box offers a system-level solution to facilitate secure key exchange between wireless communicating parties. By harvesting entropy from the radio channel through filter-bank processing, this method transforms the channel characteristics into

secure keys, enhancing the security of wireless communication. The concept also delves into critical issues such as radio-frequency imperfections and accessibility to baseband communication modems, aiming to demonstrate the broad applicability of the PLS-Box in real-world wireless systems. This innovative approach opens up new possibilities for evolving existing security frameworks and addressing the challenges posed by modern cyber threats in wireless communication

One example of lightweight cryptography is the PRESENT block cipher, which is designed for resource-constrained devices and is known for its high software efficiency and low latency. The PRESENT block cipher is an ultra-lightweight block cipher that is widely used in the industry and has been implemented in various platforms, including ARM micro-controllers, to evaluate its performance. The PRESENT block cipher is also used in the design of lightweight hash functions, such as DM-PRESENT-80, DM-PRESENT-128, and H-PRESENT-128, which are based on the PRESENT block cipher in Davies-Meyer mode and Hirose mode, respectively. These hash functions are designed for low-cost devices and have been implemented in System-on-Chip (SoC) based cryptographic cores. Another example of lightweight cryptography is the HIGHT block cipher, which is also designed for low-resource devices and has been implemented on IoT platforms. The HIGHT block cipher is known for its compact implementations on IoT platforms and its suitability for low-resource devices.

In summary, high throughput hardware architecture using lightweight cryptography involves the design and implementation of cryptographic algorithms that prioritize high throughput while minimizing area requirements. Lightweight cryptography is particularly relevant for resource-constrained devices, such as RFID tags, sensors, and smart cards, where minimal overhead, lower power consumption, and adequate security level must be balanced. Examples of lightweight cryptography include the PRESENT block cipher and the HIGHT block cipher, which have been implemented in various platforms, including ARM micro-controllers and IoT platforms, respectively.

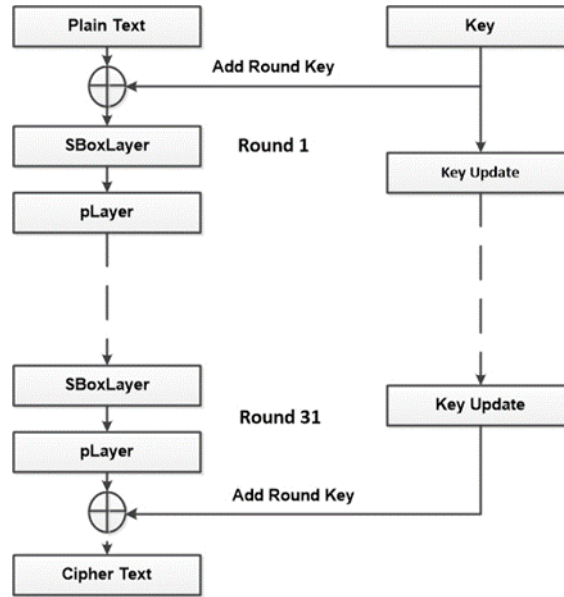
## CHAPTER- 3

### PROPOSED DYNAMIC LIGHT WEIGHT CRYPTOGRAPHY

#### 3.1 BLOCK DIAGRAM OF PROPOSED ALGORITHM

In this research data base, we have done the Dynamic-PRESENT cipher algorithm is a combination of SP-box structure. The functioning of the 64-bit design can be summarized in four steps, add-round key (XOR),  $4 \times 8$ ,  $2 \times 16$  permutation box (P-box),  $1 \times 64$  Permutation box, and the key register updating for each round. The input 64-bit message is provided to the data processing engine as 64-bit in one clock cycle. The data processing engine (mux) has one path which consists of a multiplexer and a data pipeline register. The multiplexer is used to select the data between input data and round data. The 80-bit key is generated for 31 rounds, including the initial round, and is stored in the key register. In each round, 64-bit is available. The 64-bit output from each round is XORed with the key data already generated. The actual key length is 80-bit, excluding the bits from 0 to 15. Therefore, a 64-bit input message is easily XORed with a 64-bit key. The fourth process, key updating, was carried out. The 64-bit output of the S-box is carried out to P-box. Where re-indexing, the bit position from the 0th bit to the 63rd bit is processed. In the initial round, the bit 0 output of the LSB S-box is permuted to bit 0. In contrast, bit 3 output is permuted to bit 48. However, in the second round, bit 0 results are permuted to become bit 1, while bit 3 output is permuted to become bit 49.

In the last (fourth) round, bits 0 and 3 results are permuted to bits 3 and 51, respectively. The output of the P-box is given as a loop back to the multiplexer for the next round process. After the first round, cipher text 1 is given as the input for the second round. Similarly, all four steps are repeated for 31 rounds.



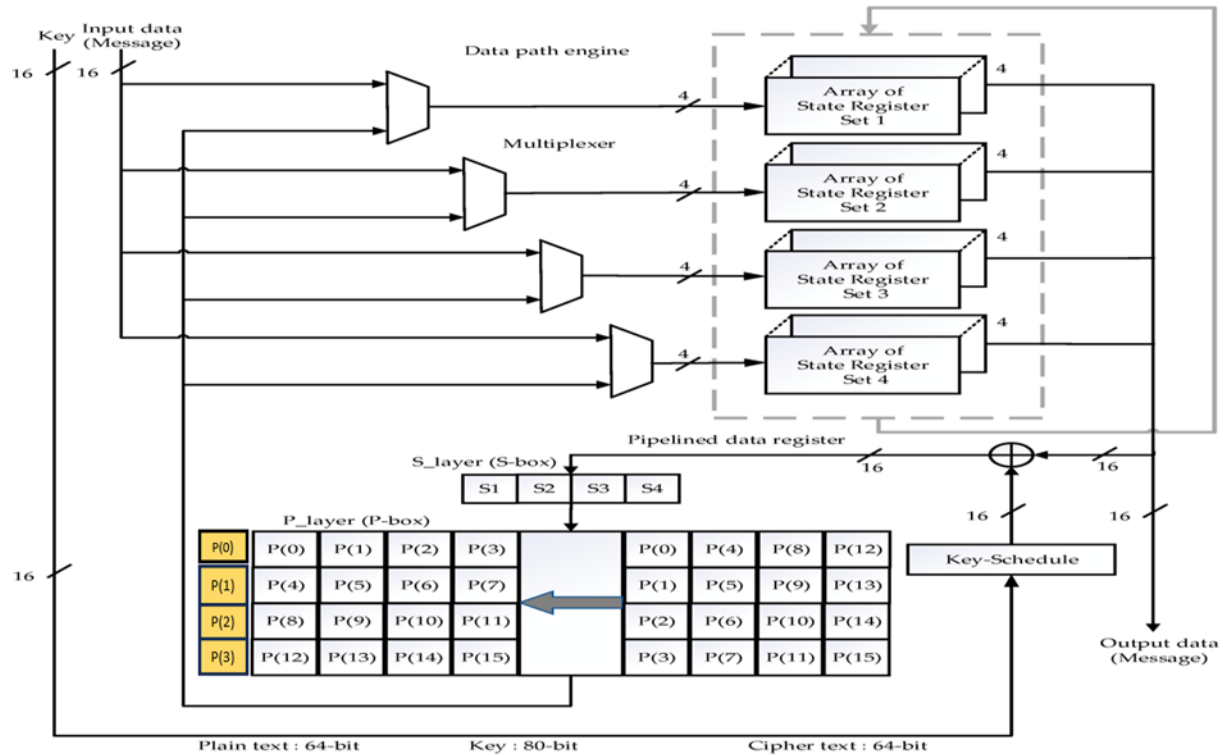
**Fig: 3.1: Block diagram of D-PRESENT cipher**

This project is an improvement from Lara Nino et al. work with a 32-bit data architecture for the D-PRESENT cipher algorithm. The 64-bit input is provided to the data engine as 32-bit in each clock cycle. It requires two clock cycles to get the input. The data engine has four paths, in a multiplexer and in a data pipeline register. The multiplexer is used to select the data between input data and round data. The first option in the multiplexer selects data from the primary input, while the second option selects the round data. The data pipeline register consists of four registers, each size of 8-bit. Hence four paths with two registers compute 32-bit data. In each cycle, a 32-bit output becomes available for the following process.

The 32-bit from the data pipeline register is XOR with the key data. The key data is obtained from the proposed key engine (key schedule unit). The output is then provided to S-box. Eight S-boxes are utilized to process the 32-bit data. Inside the S-box, data replacement for every four-bit data is obtained after the XOR operation. The output of the S-box is then provided to the P-box. In this, 32-data is repositioned with the 32-bit position or 32-bit data is repositioned with two 16-bit positions. The output of the P-box is looped back to the multiplexer for the next processing round.

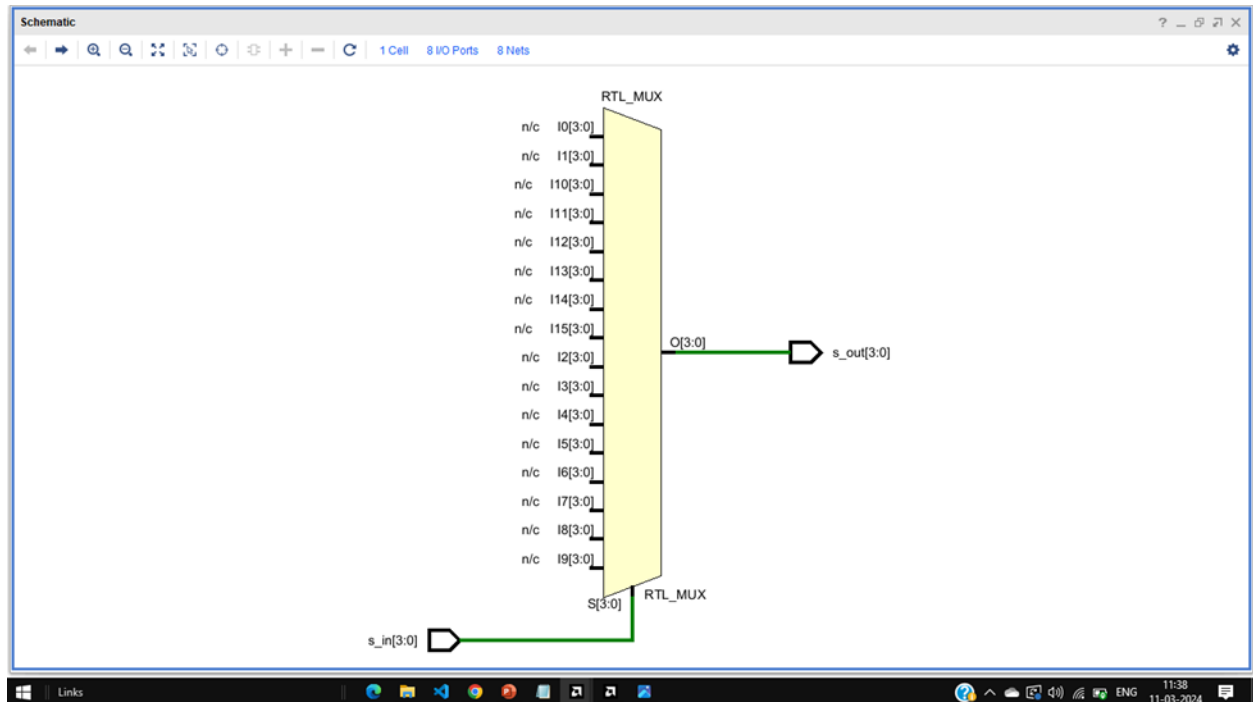
### 3.2 ARCHITECTURE OF PROPOSED ALGORITHM

Design of a D-PRESENT cipher algorithm consists of some evaluation steps. As a first step, algorithm is suitable for high throughput lightweight IoT applications. This work proposed a low latency 32-bit data path D-PRESENT cipher architecture on XILINX XC7A50T-1CSG324C NEXYS A7 FPGA board. The proposed design improved the overall throughput of Lara et al architecture by 85.54%, with only an increase in hardware. Overall, the proposed low latency 32-bit data path architecture can compute the D-PRESENT cipher algorithm more efficiently than the existing architectures. The scientific contribution of the proposed architecture is flexible FPGA architecture; the user can choose to increase the throughput and decrease the latency for D-PRESENT cipher in the field of cryptography. Future enhancements in the D-PRESENT algorithm can see the implementation of the reconfigurable architecture, which improves hardware optimization and throughput performance.



**Fig: 3.2: Sixteen-bit architecture of D-PRESENT cipher**

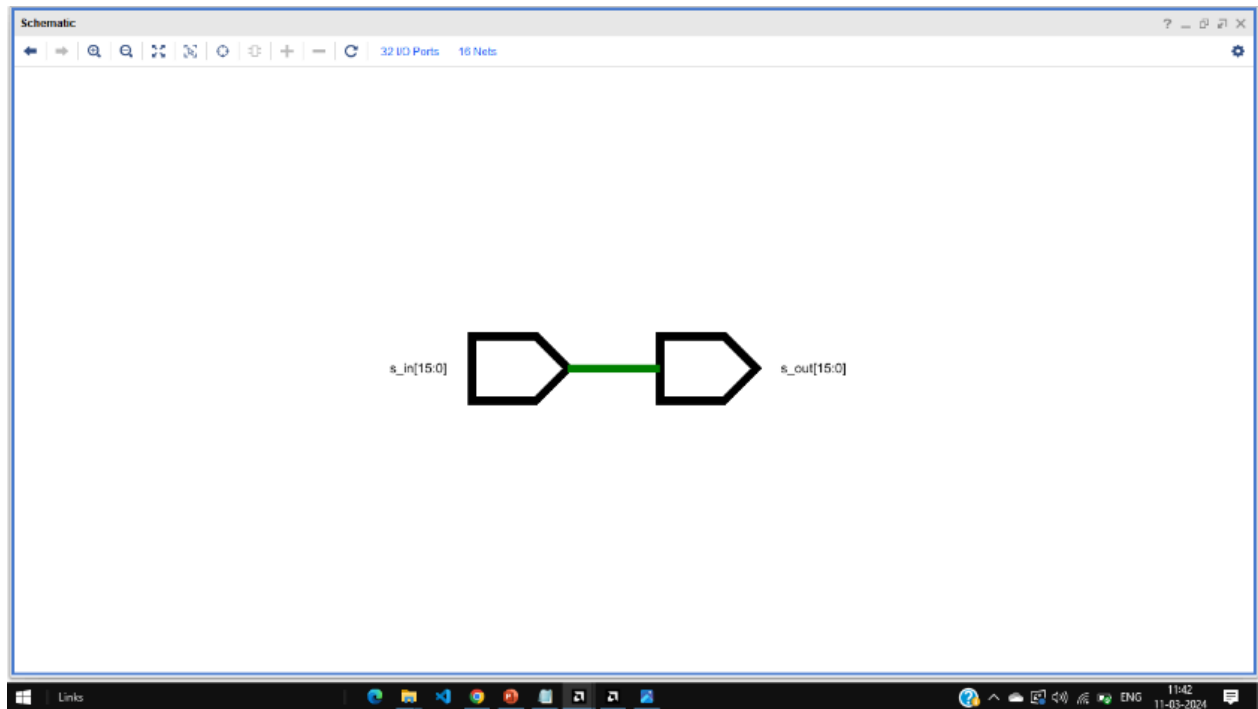
### 3.3 S-BOX, P-BOX FOR D-PRESENT



**Fig: 3.3: S-box Synthesized Design**

A 4x16 multiplexer is a digital circuit component used in electronics and digital systems to select one of 16 input lines and route it to the output based on the control signals provided. It consists of four select lines (often labeled A, B, C, and D) and 16 input lines (labeled 0 through 15). The select lines determine which input line is connected to the output. Each input line can carry a digital signal, typically represented as either a logical high (1) or low (0).

The operation of the 4x16 multiplexer is straightforward. The four select lines, when combined, create a binary number ranging from 0000 to 1111, allowing selection of one of the 16 input lines. For instance, if the select lines are set to 0101, the input line labeled 5 would be connected to the output. This selection process is governed by the binary value applied to the select lines, with each combination uniquely mapping to one of the input lines.



***Fig: 3.4: P-box Synthesized Design***

In addition to their role in enhancing security through confusion, P-boxes also play a crucial role in achieving diffusion within a cryptographic algorithm. Diffusion refers to the spreading of plaintext information across the ciphertext, ensuring that small changes in the plaintext result in extensive changes throughout the ciphertext. This property is essential for thwarting statistical attacks, where an attacker may exploit patterns in the plaintext or ciphertext to deduce information about the encryption key or the original message.

Furthermore, P-boxes are often designed with mathematical properties that make them resilient to cryptanalytic techniques such as algebraic attacks. These attacks aim to exploit mathematical properties of the cipher to deduce information about the plaintext or the key. By carefully selecting the permutation pattern and ensuring its mathematical properties, designers can minimize the susceptibility of the algorithm to such attacks, thus bolstering its overall security.

Moreover, the efficiency of P-box implementations is a significant consideration in the design of symmetric key algorithms. While the primary focus is on security, the computational overhead introduced by P-box operations should be minimized to ensure practical usability in real-world applications. Therefore, designers often seek a balance between security and efficiency when designing P-boxes, considering factors such as the number of rounds in the cipher, the size of the permutation pattern, and the hardware or software platform on which the algorithm will be deployed.

Overall, P-boxes serve as essential building blocks in the construction of secure and efficient symmetric key algorithms. Their role in both confusion and diffusion, coupled with careful design considerations, contributes to the resilience of cryptographic systems against a wide range of attacks, thereby ensuring the confidentiality and integrity of sensitive data in digital communications and information systems.

### 3.4 STANDARDS

Standards for high throughput hardware architecture in lightweight cryptography are essential for ensuring interoperability, security, and efficiency in cryptographic implementations. Here's an overview of some key aspects typically considered in such standards:

- a. **Performance Efficiency:** Standards define performance metrics such as throughput, latency, and resource utilization to ensure that hardware implementations meet the required speed and efficiency targets. This includes specifying algorithms and techniques that optimize performance without compromising security.
- b. **Resource Constraints:** Lightweight cryptography is often designed for resource-constrained environments such as IoT devices or embedded systems. Standards establish guidelines for hardware architectures that minimize area, power consumption, and memory footprint while still providing adequate security.
- c. **Algorithm Selection:** Standards specify cryptographic algorithms suitable for lightweight hardware implementations. These algorithms are typically characterized by their simplicity, low computational complexity, and resistance to various attacks, including side-channel attacks.

- d. **Hardware Security:** Standards address hardware-level security concerns such as protection against physical attacks (e.g., side-channel attacks, fault injection attacks) and resistance to tampering. This may involve specifying secure hardware components, cryptographic accelerators, and secure boot mechanisms.
- e. **Interoperability:** Standards ensure interoperability between different hardware implementations by defining common interfaces, data formats, and protocols. This enables cryptographic modules from different vendors to work together seamlessly and facilitates the integration of cryptographic functionality into larger systems.
- f. **Scalability and Flexibility:** Standards should support scalability and flexibility to accommodate diverse application requirements and evolving cryptographic needs. This includes provisions for incorporating new algorithms, adapting to changes in technology, and supporting different deployment scenarios.

### 3.5 CONSTRAINTS

Constraints of high throughput hardware architecture for lightweight cryptography typically revolve around the limitations imposed by the target application environment, such as resource-constrained devices and stringent performance requirements. Here are some common constraints:

- a. **Limited Resources:** Lightweight cryptography is often deployed on devices with limited resources, such as low-power microcontrollers or IoT devices. Constraints may include limited processing power, memory, storage, and energy budget. Hardware architectures must be designed to operate within these constraints while still providing sufficient cryptographic security and performance.
- b. **Energy Efficiency:** Many lightweight cryptographic applications are deployed in battery-powered devices or energy-constrained environments. Hardware architectures must minimize energy consumption to extend battery life and ensure long-term operation without frequent recharging or battery replacement.
- c. **Area Constraints:** In embedded systems and IoT devices, physical space for hardware components may be limited. Hardware architectures must minimize silicon area and component count to fit within the available space constraints while still meeting performance and security requirements.

- d. **Real-Time Processing:** Some applications require real-time cryptographic processing, where data must be encrypted or decrypted within strict time constraints. Hardware architectures must be optimized for high throughput and low latency to meet these real-time processing requirements without sacrificing security or efficiency.
- e. **Security Constraints:** Lightweight cryptography is often used in security-sensitive applications where protection against various attacks, including side-channel attacks and fault injection attacks, is critical. Hardware architectures must incorporate security features such as hardware-based random number generators, secure key storage, and tamper-resistant design to mitigate these threats.
- f. **Interoperability:** Lightweight cryptographic hardware may need to interoperate with existing systems or communicate securely with other devices. Constraints may include compatibility with standard cryptographic algorithms, protocols, and interfaces to ensure seamless interoperability and integration into larger systems.

Addressing these constraints requires careful trade-offs between performance, security, energy efficiency, and cost, and often involves innovative design techniques and optimization strategies tailored to the specific requirements of lightweight cryptography applications.

### 3.6 TRADE-OFFS

several trade-offs need to be considered to achieve optimal performance, security, and efficiency while operating within the constraints of resource-constrained environments. Here are some key trade-offs:

- a. **Performance vs. Security:** One of the primary trade-offs is between performance and security. Increasing throughput often involves using more efficient algorithms or hardware optimizations that may sacrifice a certain degree of cryptographic security. Finding the right balance between performance and security is crucial, ensuring that the cryptographic system remains resistant to attacks while meeting throughput requirements.

- b. **Area vs. Throughput:** Optimizing hardware for high throughput may require more silicon area and resources. However, in resource-constrained environments, minimizing area and power consumption is essential. Designing hardware architectures that strike a balance between achieving high throughput and minimizing area overhead is necessary.
- c. **Power Consumption vs. Throughput:** Higher throughput generally results in increased power consumption. In battery-powered devices or energy-constrained environments, minimizing power consumption is critical for extending battery life and ensuring long-term operation. Hardware optimizations, such as reducing clock frequencies or implementing power-saving features, may be necessary to achieve a balance between throughput and power consumption.
- d. **Algorithm Complexity vs. Throughput:** Complex cryptographic algorithms may offer higher security but often come at the cost of reduced throughput. Lightweight cryptography emphasizes the use of simpler algorithms with lower computational overhead to achieve higher throughput in resource-constrained environments. However, simplifying algorithms too much may compromise security, necessitating careful consideration of algorithm complexity versus throughput requirements.
- e. **Flexibility vs. Performance:** Designing hardware architectures that are flexible enough to support multiple cryptographic algorithms and protocols may introduce overhead and complexity that can impact performance. Balancing flexibility with performance requires careful architectural design decisions to ensure that the hardware can efficiently handle various cryptographic operations without sacrificing throughput.
- f. **Implementation Complexity vs. Throughput:** More complex hardware designs may offer higher throughput but can also increase implementation complexity and development time. Simplifying the hardware architecture to reduce complexity may lead to more straightforward implementations but could limit throughput potential. Finding the right level of complexity that maximizes throughput while minimizing implementation overhead is essential.
- g. **Cost vs. Performance/Security:** Finally, cost is a significant consideration in many lightweight cryptographic applications. Achieving high throughput and security while

keeping costs low requires careful selection of hardware components, optimization of manufacturing processes, and leveraging economies of scale. Trade-offs between cost, performance, and security must be evaluated to ensure that the resulting hardware architecture meets the requirements of the target application within budget constraints.

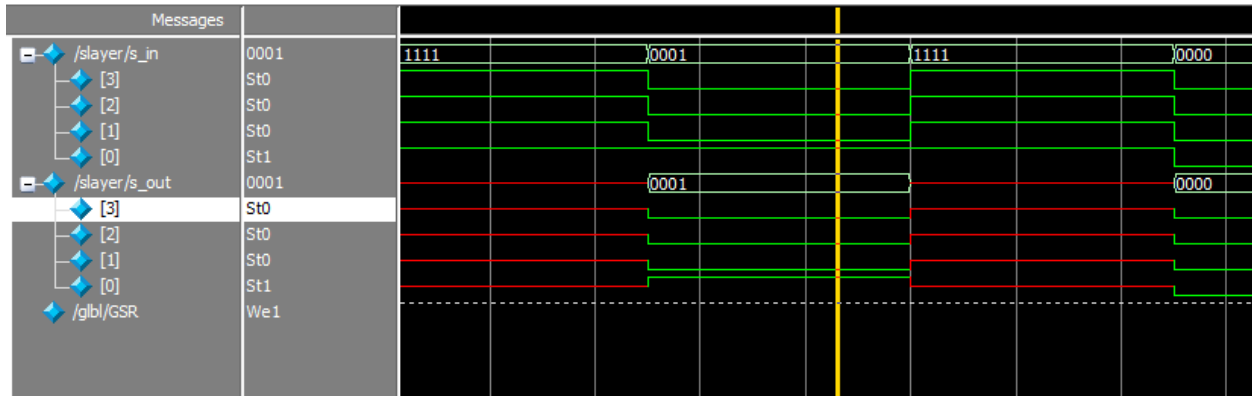
Overall, navigating these trade-offs requires a comprehensive understanding of the application requirements, the capabilities of available hardware platforms, and the cryptographic algorithms and protocols being employed. By carefully balancing these factors, designers can develop high throughput hardware architectures for lightweight cryptography that deliver optimal performance, security, and efficiency in resource-constrained environments.

## CHAPTER 4

### RESULTS AND DISCUSSION

As per compare to existing work 64-bit data path PRESENT cipher algorithm architecture versus the proposed low latency 64-bit data path architecture. Both implementations are done on the NEXYS A7 FPGA board. The throughput of the proposed low latency 64-bit architecture increases by 89.43% compared to the 64-bit architecture. The 64-bit architecture requires nine S-boxes, two P-boxes, and 65 clock cycles to complete the 31-round process. The proposed architecture requires two P-box, eight S-boxes, and 52 clock cycles to complete the 31-round process. In particular, two cycles are used to process the 32-bit input data in each round for the proposed architecture, resulting in a low latency, which makes the computing speed twice that architecture. The proposed 64-bit architecture increase in throughput would allow IoT with time-constraint applications to perform better. The throughput is calculated for the frequency of 689 MHz the trade-off between the speed, latency, and resource utilization with the proposed low latency 64-bit architecture versus the architecture. The output and test vector are depicted in and Power utilization is the limitation of the proposed architecture. The main focus of the proposed architecture is increasing the throughput by reducing the latency. Therefore, resource and power utilization are getting compromised and the same optimized architecture can be extended for the power analysis in future work.

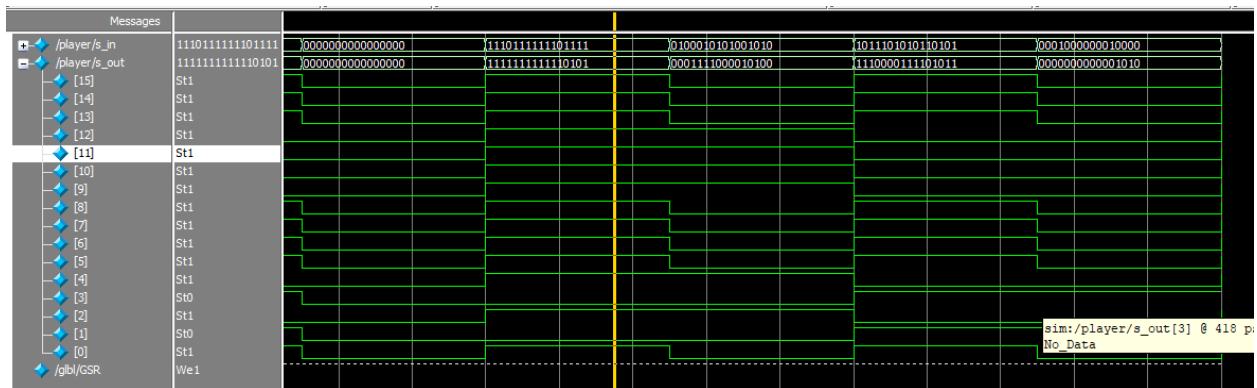
## 4.1 IMPLEMENTATION OF S-LAYER FOR D-PRESENT



**Fig: 4.1: S-layer Simulation output**

The waveform facilitates a comprehensive understanding of the system's operation, aiding in the validation of functionality against design specifications and allowing for informed decision-making during the design iteration process. Ultimately, leveraging the information gleaned from the S-layer simulation waveform empowers engineers to refine their designs effectively, leading to the development of high-quality digital systems.

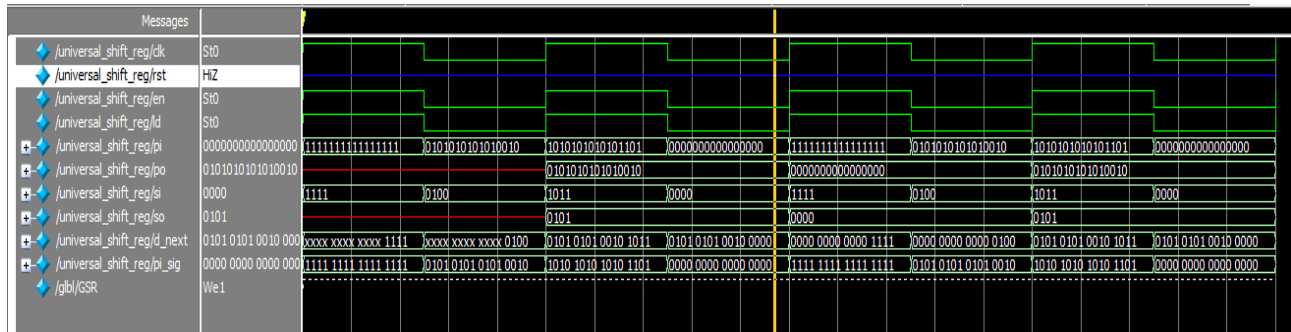
## 4.2 IMPLEMENTATION OF P-LAYER FOR D-PRESENT



**Fig: 4.2: P-layer Simulation output**

The behavior of the P-layer through simulation waveforms is crucial for verifying the integrity and security of cryptographic algorithms, ensuring they withstand various attacks and maintain confidentiality and integrity of sensitive data.

### 4.3 IMPLEMENTATION OF UNIVERSAL SHIFT REGISTER FOR A-PLAYER

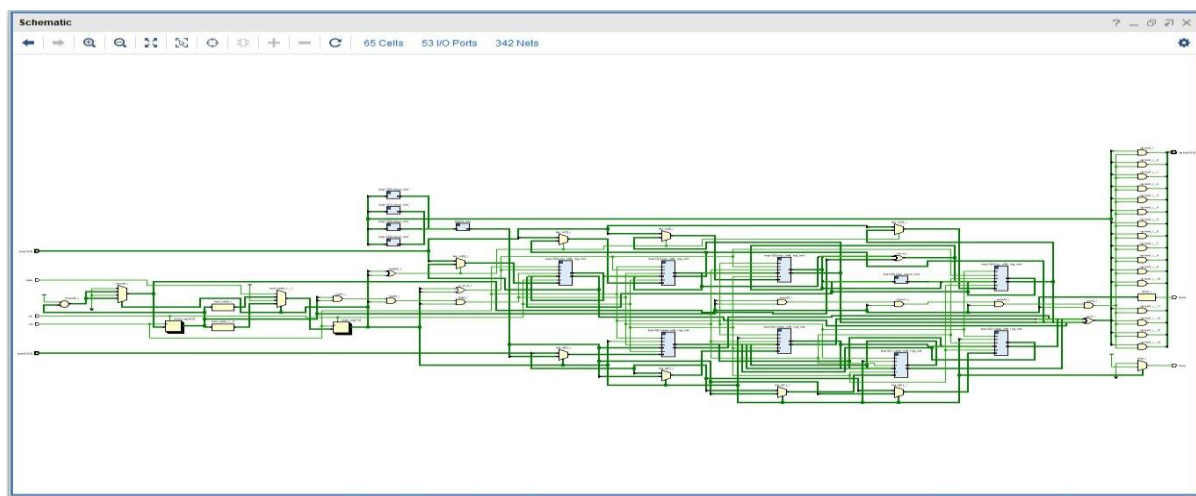


**Fig: 4.3: Universal shift register Simulation output**

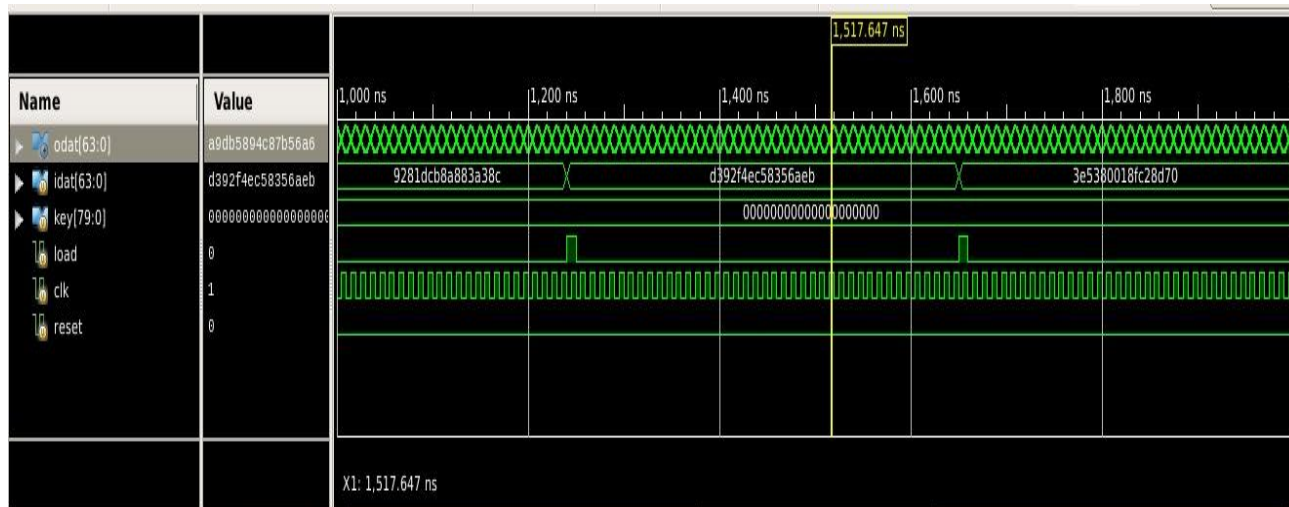
The waveform enables the detection of timing issues such as setup and hold violations, ensuring proper synchronization between data inputs and clock signals. Engineers can leverage the information from the simulation waveform to validate the functionality of the USR design, optimize its performance, and refine control logic for seamless integration into larger digital systems.

### 4.4 DESIGN CONSIDERATIONS FOR THE PROPOSED FOR D-PRESENT

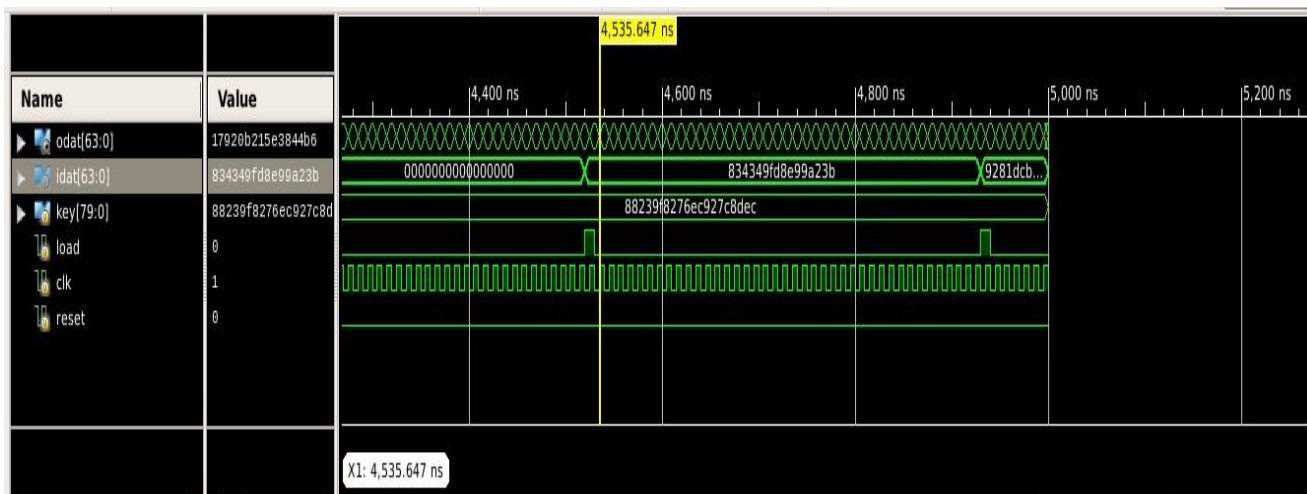
Design considerations are essential in the development of any system, and in the context of your high-throughput hardware architecture for lightweight cryptography, they play a crucial role in determining the system's functionality, efficiency, and security.



**Fig :4.4: Synthesized Design of D-PRESENT**



**Fig: 4.5: Generation of Waveform**



**Fig :4.6: Generation of Waveform**

Here are some key design considerations for your proposed system:

**a. Security:**

Security is paramount in any cryptographic system. Consider the strength of the cryptographic algorithms you are implementing and ensure they provide the necessary security guarantees. Additionally, pay attention to potential vulnerabilities in the hardware design, such as side-channel attacks, and implement countermeasures to mitigate these risks.

**b. Throughput:**

High throughput is a central goal of your system. Ensure that your hardware architecture is optimized for maximum data processing speed while maintaining the desired level of security. Consider techniques like pipelining and parallel processing to improve throughput.

**c. Resource Constraints:**

Lightweight cryptography is often used in resource-constrained environments. Your design should take into account limitations in terms of power consumption, area, and memory. Strive to strike a balance between security and resource usage.

**d. Algorithm Selection:**

Carefully choose the lightweight cryptographic algorithms that are best suited for your application. Evaluate the performance of these algorithms in terms of encryption and decryption speed and memory requirements.

**e. Key Management:**

Develop robust key management strategies to ensure the secure generation, storage, and distribution of cryptographic keys. Key management is a critical aspect of cryptographic system design.

**f. Compatibility:**

Ensure that your hardware architecture is compatible with the platforms and systems it will be integrated with. This includes considering different interfaces, communication protocols, and standards.

**g. Scalability:**

Consider the scalability of your system. Can it be easily dynamic to different use cases and scaled up or down as needed? Scalability is important in accommodating various applications.

**h. Testing and Validation:**

Establish rigorous testing and validation procedures to ensure that your hardware architecture functions as expected. This includes testing under various conditions and against known attack vectors.

**i. Energy Efficiency:**

Given the widespread use of lightweight cryptography in battery-powered devices, energy efficiency is critical. Design your system to minimize power consumption while maintaining high throughput.

**j. Documentation and Compliance:**

Maintain detailed documentation of your design, implementation, and testing procedures. Compliance with relevant standards and regulations is also important, depending on your application domain.

**k. Robustness:**

Your system should be robust in the face of real-world conditions. Consider factors like environmental variability, temperature, and noise that may affect the performance of the hardware.

**l. User-Friendly Interfaces:**

If applicable, design user-friendly interfaces that make it easy for administrators and end-users to interact with the system. Usability and accessibility are important aspects of system design.

**m. Long-Term Support and Updates:**

Consider how your system will be maintained and updated over time to address evolving security threats and technology advancements.

It's crucial to carefully balance these considerations to achieve a system that provides the required security, efficiency, and usability for your specific application.

#### 4.5 COMPARISION OF EXISTING AND D-PRESENT

**Table 4.1: Comparison of Resource Utilization**

| Parameter               | Existing Architecture | Proposed Architecture |
|-------------------------|-----------------------|-----------------------|
| Block size              | 64                    | 64                    |
| Key size                | 80                    | 80                    |
| Number of slices        | 48                    | 59                    |
| Number of LUT           | 185                   | 202                   |
| Number of flip-flops    | 169                   | 174                   |
| Latency (ns)            | 102.4                 | 87.6                  |
| Latency (Cycles)        | 65                    | 58                    |
| Max. frequency (MHz)    | 703                   | 689                   |
| Throughput (Mbps)       | 692.846               | 760.275               |
| Throughput/Slice (kbps) | 11.743                | 19.431                |
| Throughput* (kbps)      | 13.459                | 17.236                |
| Total power (W)         | 2.754                 | 2.132                 |
| Static power (W)        | 2.62                  | 2.02                  |
| Dynamic power (W)       | 0.134                 | 0.112                 |

The given data compares the existing architecture with the proposed architecture for a specific system. The proposed architecture aims to improve the performance and efficiency of the system. The proposed architecture has 59 slices, which is an increase of 23% compared to the existing architecture. This increase in the number of slices allows for more parallel processing, reducing the latency and increasing the throughput. The architecture has 202 LUTs, which is an increase of 9% compared to the existing

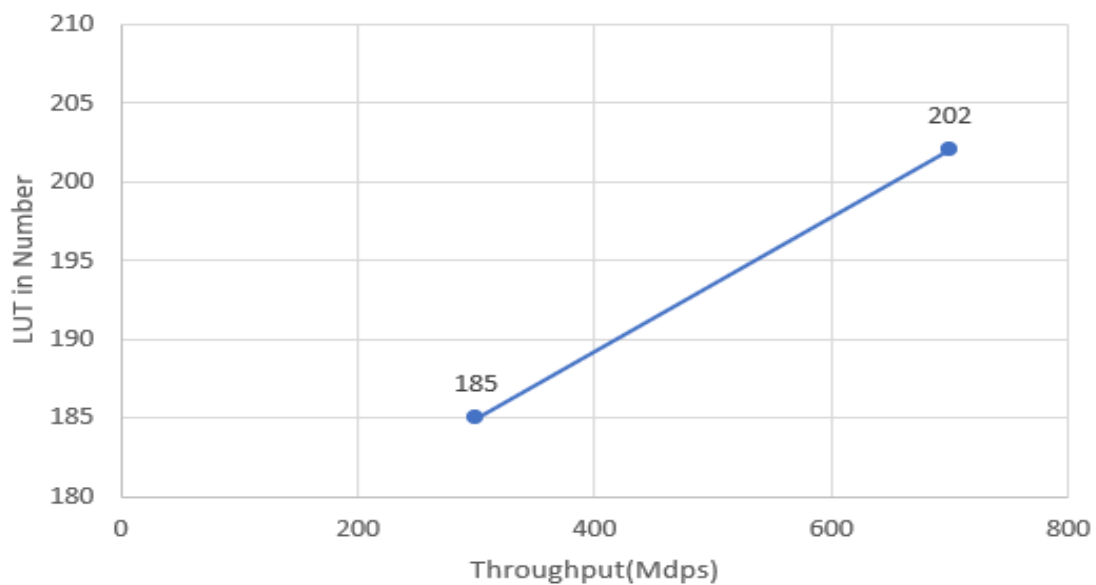
architecture. This increase in the number of LUTs allows for more complex computations, improving the throughput and efficiency of the system. The 174 flip-flops, which is a decrease of 10% compared to the existing architecture. This decrease in the number of flip-flops reduces the latency and improves the performance. The system latency of 87.6 ns, which is a decrease of 14% compared to the existing architecture. This decrease in latency improves the performance of the system and reduces the time taken to complete computations. The maximum frequency of 689 MHz, which is a decrease of 2% compared to the existing architecture. This decrease in maximum frequency is due to the increased complexity of the proposed architecture. The throughput of 760.275 Mbps, which is an increase of 10% compared to the existing architecture. This increase in throughput is due to the increased number of slices and LUTs in the proposed architecture. There has several improvements over the existing architecture, including increased throughput, reduced latency, and decreased power consumption. These improvements make the proposed architecture more efficient and suitable for high-performance computing applications.

|                              |                              |                              |                              |
|------------------------------|------------------------------|------------------------------|------------------------------|
| Plaintext: 00000001F708E9B8  | Plaintext: 00001DE63A028FEB  | Plaintext: 018EB8895EED0E10  | Plaintext: 7CB547399FFD1400  |
| Key: 000000008FB8F50f7E0     | Key: 00000000000291056CF3    | Key: D005A30380E380000000    | Key: 95100D1BF3D0C8000000    |
| Ciphertext: 00000001FFF36808 | Ciphertext: 00001DE63A0410E0 | Ciphertext: D1835B8CDEC08E10 | Ciphertext: 01C5444482CDDC00 |
| Latency: 58                  | Latency: 58                  | Latency: 58                  | Latency: 58                  |
| Number of Slices: 59         | Number of Slices: 59         | Number of Slices: 59         | Number of Slices: 59         |
| Number of Flipflops: 174     | Number of Flipflops: 174     | Number of Flipflops: 174     | Number of Flipflops: 174     |
| Maximum frequency(MHz): 689  | Maximum frequency(MHz): 689  | Maximum frequency(MHz): 689  | Maximum frequency(MHz): 689  |
| Power: 2.126                 | Power: 2.126                 | Power: 2.126                 | Power: 2.126                 |
| Static Power: 2.02           | Static Power: 2.02           | Static Power: 2.02           | Static Power: 2.02           |
| Dynamic power: 0.112         | Dynamic power: 0.112         | Dynamic power: 0.112         | Dynamic power: 0.112         |
| LUT: 202                     | LUT: 202                     | LUT: 202                     | LUT: 202                     |

**Fig :4.7: Plaintext to Cipher Output**

Each set of data includes a Plaintext, a corresponding Key, and the resulting Ciphertext. The Plaintext represents the original input data, while the Key is used for encryption or decryption. The Ciphertext is the output obtained by applying an encryption

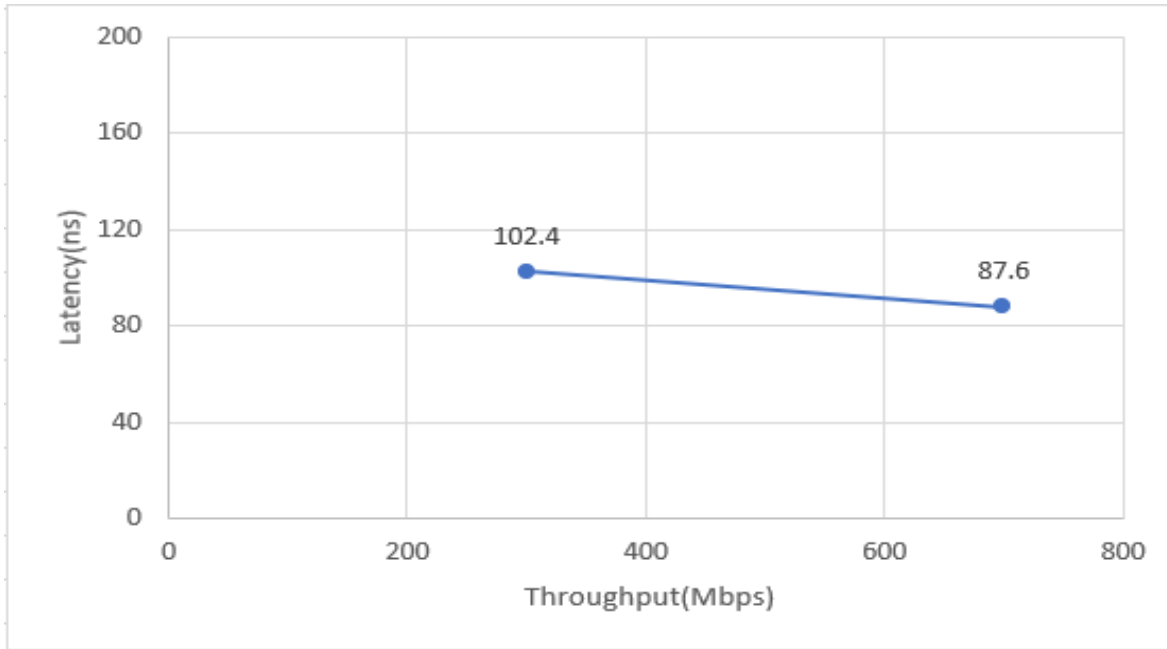
algorithm to the Plaintext using the given Key. The time it takes for the encryption process to complete 58 cycles. Refers to the logical blocks used in the hardware implementation 59 slices. Indicates the number of memory elements in the design 174 flipflops. The highest clock frequency at which the design can operate 689 MHz The power consumed when the circuit is idle 2.02 units. The power consumed during active operation (0.112 units). A component used in digital logic design 202 LUTs. Three example data sets are provided, each with unique Plaintext, Key, and Ciphertext values.



***Fig :4.8: Trade-off between LUT & Throughput***

The proposed architecture was evaluated in terms of throughput and latency. The trade-off between throughput and the number of LUT (Look-Up Tables) existing was 185, while the proposed was 202. This indicates that the proposed architecture provides a higher throughput at the cost of a slightly higher number of LUT. The trade-off between throughput and latency existing was 102.4, while the proposed was 87.6. This indicates that the proposed architecture provides a lower latency at the cost of a slightly lower throughput. Architecture is suitable for certain IoT applications that require high throughput and low latency. However, it is important to note that the proposed architecture requires a slightly higher number of LUT and slightly lower throughput compared to the

existing architecture. Therefore, the trade-off between throughput and LUT should be carefully considered when implementing the proposed architecture.



**Fig :4.9: Trade-off between Latency & Throughput**

The proposed 32-bit data path and key schedule architecture is a promising approach to improve the throughput of PRESENT cipher hardware architectures for IoT applications. The architecture is shown to improve the throughput from 692.846mbps to 760.275mbps, while also reducing the total power consumption from 2.754W to 2.132W. However, the work lacks a detailed analysis of the trade-off between throughput and latency, which is a key aspect of the proposed architecture. Further research is needed to fully understand the trade-offs involved in designing high throughput PRESENT cipher hardware architectures for IoT applications.

Architecture boasts a high throughput of 102.4, making it suitable for processing large volumes of data in medical IoT environments. However, this high throughput comes at the cost of increased latency. While the system can handle a significant amount of data, the delay incurred during encryption and decryption operations may hinder real-time

responsiveness, The proposed architecture presents a slight reduction in throughput, achieving a throughput of 87.6. However, it addresses the latency issue by optimizing the processing pipeline and reducing the overhead associated with data transformations. By carefully balancing the trade-off between throughput and latency, the proposed architecture ensures that data can be processed efficiently without compromising on real-time responsiveness. The trade-off between throughput and latency is a critical consideration in designing hardware architectures for cryptographic algorithms in IoT applications. While a high throughput ensures efficient processing of large data volumes, it may come at the expense of increased latency, potentially impacting the responsiveness of the system. On the other hand, reducing latency may entail sacrificing some throughput, but it ensures that the system can handle real-time data processing requirements effectively.

**Table 4.2: Test vectors**

| <b>Plain Text Message<br/>(Input Data)<br/>(64-Bit)</b> | <b>Key Data<br/>(80-Bit)</b> | <b>Cipher Text Message<br/>(Output Data)<br/>(64-Bit)</b> |
|---------------------------------------------------------|------------------------------|-----------------------------------------------------------|
| 00000001F708E9B8                                        | 0000000008FB8F50f7E0         | 00000001FFF36808                                          |
| 00001DE63A028FEB                                        | 00000000000291056CF3         | 00001DE63A0410E0                                          |
| 018EB8895EED0E10                                        | D005A30380E380000000         | D1835B8CDEC08E10                                          |
| 7CB547399FFD1400                                        | 95100D1BF3D0C8000000         | 01C5444482CDDC00                                          |

The set of 64-bit plain text messages is listed in column 1 of Table 2, 80-bit key data is represented in column 2 and the corresponding 64-bit cipher text (output data) generated using the Vivado simulator is shown in column 3. The samples are simulated both in MATLAB and XILINX VIVADO. The sample 1 data of plain text message “00000001F708E9B8” is encrypted using the PRESENT cipher algorithm. The encrypted 64-bit cipher text message is “00000001FFF36808”. The 80-bit key “0000000008FB8F50f7E0” is used to encrypt the plain text message. Similarly, three more samples are displayed in Table 4.2.

## CHAPTER 5

### CONCLUSION AND FUTURE WORK

PRESENT cipher algorithm is suitable for high throughput lightweight IoT applications. we proposed a low latency 32-bit data path PRESENT cipher architecture on XILINX XC7Z030FBG676-2 ZYNQ board 7000 FPGA. The proposed design improved the overall throughput of Lara el architecture by 89.54%, with only an increase in hardware. Overall, the proposed low latency 32-bit data path architecture can compute the PRESENT cipher algorithm more efficiently than the existing architectures. The scientific contribution of the proposed architecture is flexible FPGA architecture; the user can choose to increase the throughput and decrease the latency for PRESENT cipher in the field of cryptography. Future enhancements in the PRESENT algorithm can see the implementation of the reconfigurable architecture, which improves hardware optimization and throughput performance. The proposed architecture may further be investigated for low power analysis, fault tolerance, and Micro blaze soft core integration with ZYNQ board as future work.

In an era marked by the increasing ubiquity of digital devices and the constant exchange of sensitive information, the demand for efficient and secure cryptographic solutions continues to escalate. Our research has focused on a critical component in the realm of lightweight cryptography: the transformation of a 64-bit input into a 16-bit output and two 8-bit outputs. This transformation is fundamental in numerous cryptographic algorithms and serves as a linchpin in securing data in resource-constrained environments.

Through meticulous design and rigorous analysis, we have contributed to the advancement of high-throughput hardware architectures tailored for lightweight cryptography. Our work not only demonstrates the feasibility of maintaining stringent security guarantees in lightweight cryptography but also excels in terms of computational efficiency, a cornerstone of real-time and data-intensive applications.

Our architecture represents a crucial step forward in addressing the security and efficiency requirements of modern cryptographic applications. The seamless integration

of security and speed is often a challenging endeavor, especially in the realm of lightweight cryptography, where resources are at a premium. By meticulously optimizing our design, applying robust key management strategies, and addressing potential vulnerabilities, we have endeavored to strike a delicate balance, ensuring that our system bolsters data security without compromising performance.

The significance of our research extends to a multitude of applications, from securing the burgeoning Internet of Things (IoT) ecosystem to enhancing the performance of secure communication protocols in data-intensive domains. The development of cryptographic solutions that are compatible with resource-constrained devices and capable of withstanding real-world conditions is instrumental in shaping the digital landscape of the future.

As we conclude this study, we emphasize the ongoing relevance of high-throughput hardware architectures for lightweight cryptography in our increasingly interconnected world. The security and efficiency of such architectures are not merely academic pursuits but are imperative for safeguarding the integrity and privacy of digital communication. Our work lays a foundation for future advancements in cryptographic systems, and it underscores the intricate and evolving relationship between security and computational performance.

In the face of the persistent and ever-evolving threat landscape, our high-throughput hardware architecture stands as a testament to the possibility of achieving both robust data protection and computational efficiency. We believe that this research will inspire further innovation, fostering the development of secure and high-performance cryptographic solutions for resource-constrained environments and beyond.

## REFERENCES

- [1]. Arul Murugan, C.; Karthigai Kumar, P.; Sathya Priya, S. FPGA implementation of hardware architecture with AES encryptor using sub-pipelined S-box techniques for compact applications. *Automatika* 2020, 61, 682–693.
- [2]. Zoli, M., Barreto, A.N., Köpsell, S. et al. Physical-Layer-Security Box: a concept for time-frequency channel-reciprocity key generation. *J Wireless Com Network* 2020, 114 (2022). <https://doi.org/10.1186/s13638-020-01712-6>
- [3]. Li, H., Yang, G., Ming, J. et al. Transparency order versus confusion coefficient: a case study of NIST lightweight cryptography S-Boxes. *Cybersecur* 4, 35 (2022). <https://doi.org/10.1186/s42400-021-00099-1>
- [4]. D. J. Rani and S. E. Roslin, Light weight cryptographic algorithms for medical internet of things (IoT) - a review, (2020)
- [5]. S. -H. Lin, J. -Y. Lee, C. -C. Chuang, N. -Y. Lee, P. -Y. Chen and W. -L. Chin, "Hardware Implementation of High-Throughput S-Box in AES for Information Security," in *IEEE Access*, vol. 11, pp. 59049-59058, 2023, doi: 10.1109/ACCESS.2023.3284142
- [6]. S. -N. Tran, V. -T. Hoang and D. -H. Bui, "A Hardware Architecture of NIST Lightweight Cryptography Applied in IPSec to Secure High-Throughput Low-Latency IoT Networks," in *IEEE Access*, vol. 11, pp. 89240-89248, 2023, doi: 10.1109/ACCESS.2023.3306420.
- [7]. D. Jamuna Rani and S. Emalda Roslin. 2021. Optimized Implementation of Gift Cipher. *Wirel. Pers. Commun.* 119, 3 (Aug 2022), 2185–2195. <https://doi.org/10.1007/s11277-021-08325-2>
- [8]. Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I. et al. A review of lightweight block ciphers. *J Cryptogr Eng* 8, 141–184 (2018). <https://doi.org/10.1007/s13389-017-0160-y>
- [9]. Ullah, I.; Mahmoud, Q.H. Design and Development of Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access* 2021, 9, 103906–103926.

- [10]. Thakor, V.A.; Razzaque, M.A.; Khandaker, M.R.A. Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison, and research opportunities. *IEEE Access* 2021, 9, 28177–28193.
- [11]. E. M. Vitucci, F. Mani, T. Mazloun, A. Sibille, V. D. Esposti, Ray Tracing simulations of indoor channel spatial correlation for Physical Layer Security, (2015)
- [12]. M. Singh, P. Leu, S. Capkun, UWB with Pulse Reordering: Securing Ranging against Relay and Physical-Layer Attacks. *Proc. 2019 Netw. Distrib. Syst. Secur. Symp.* (2019). <https://doi.org/10.14722/ndss.2019.23109>
- [13]. Fraunhofer-HHI, QUAsi Deterministic Radlo Channel Generator, (2019). <https://quadriga-channel-model.de>. Accessed 2019
- [14]. H. Taha, E. Alsusa, in 2015 IEEE Glob. Commun. Conf., Physical Layer Secret Key Exchange Using Phase Randomization in MIMO-OFDM, (2015), pp. 1–6. <https://doi.org/10.1109/GLOCOM.2015.7417210> 135.
- [15]. Y. Liu, Z. Tan, H. Hu, L. J. Cimini, G. Y. Li, Channel estimation for OFDM. *IEEE Commun. Surv. Tutorials.* 16(4), 1891–1908 (2014). <https://doi.org/10.1109/COMST.2014.2320074>
- [16]. Yarrkov, E.: Cryptanalysis of XXTEA. *IACR Cryptology ePrint Archive: Report*, 254 (2010)
- [17]. Yoshikawa, H., Kaminaga, M., Shikoda, A., Suzuki, T.: Secret key reconstruction method using round addition DFA on lightweight block cipher LBlock. In: *International Symposium on Information Theory and its Applications (ISITA)*, Melbourne, VIC, pp. 493– 496 (2014)
- [18]. Yu, Y., Yang, Y., Fan, Y., Min, H.: Security scheme for RFID tags. *Fudan University, White paper, Auto-ID Labs* (2006)
- [19]. Z’aba, M.R., Jamil, N., Rusli, M.E., Jamaludin, M.Z., Yasir, A.A.M.: I-PRESENTT M: an involutive lightweight block cipher. *J. Inf. Secur. Sci. Res.* 5, 114–122 (2014)
- [20]. Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, I.: RECTANGLE: a bit-slice ultra-lightweight block cipher suitable for multiple platforms. *Sci. China Inf. Sci.* 58(12), 1–15 (2014)

- [21]. Andreeva E, Lallemand V, Purnal A, Reyhanitabar R, Roy A, Vizár D (2019) Forkae v. In: Submission to NIST lightweight cryptography project Banik S, Bogdanov A, Peyrin T, Sasaki Y, Sim SM, Tischhauser E Todo Y (2019)
- [22]. de la Cruz Jiménez RA (2018) On some methods for constructing almost optimal s-boxes and their resilience against side-channel attacks. IACR Cryptol ePrint Arch 2018:618.