# Hardware Implementation of BORON Cipher in CBC Mode for RFID Tags

*A Thesis Submitted in Partial Fulfilment of the Requirement for the Award of the Degree of*

**MASTER OF TECHNOLOGY**

in VLSI Design

**Submitted By**
DIXITA GUPTA
601662006

**Under Supervision of**
**Mrs. Manu Bansal**
Assistant Professor


ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY
(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB
JUNE, 2018

# DECLARATION

I, Dixita Gupta hereby declare that the work presented in this thesis entitled **"Hardware Implementation of BORON Cipher in CBC Mode for RFID Tags"**, in partial fulfilment of the requirement for the award of degree of Master of Technology (VLSI Design), submitted at ECED, Thapar Institute of Engineering & Technology (Deemed to be University), Patiala is an authentic record of work carried out under supervision of Mrs. Manu Bansal, Assistant Professor, ECED, Thapar Institute of Engineering and Technology from July, 2017 to June, 2018. The matter presented in this work has not been submitted either in part or full to any other university or institute for the award of any other degree.

Date: 14/6/18

**Dixita Gupta**
601662006

It is certified that the above statement made by the student is correct to best of my knowledge and belief.

Date : 14/6/18

**Mrs. Manu Bansal**
Assistant Professor
Electronics And Communication Engineering Department
Thapar Institute Of Engineering & Technology
(A Deemed To Be University), Patiala, Punjab

# ACKNOWLEDGEMENT

First of all, I would like to express my sincere gratitude to **Mrs. Manu Bansal**, Assistant Professor, Electronics and Communication Engineering Department, Thapar Institute of Engineering and Technology, Patiala for her patient guidance and constant support throughout this report.

I am also thankful **to Dr. Alpana Agarwal**, Head of the Department, ECED, TIET, Patiala for providing us with an adequate environment and lab facilities during the work.

Finally, the acknowledgment would be incomplete without express my gratitude to each and every individual who rendered his/her support towards the successful completion of this work.

The study has indeed helped me in exploring the knowledge and avenue related to my area of research and surely it is going to help me in near future.

**Dixita Gupta**

**601662006**

# ABSTRACT

In today's era, RFID tags allow giving digital identity to almost every object in the world. These devices find their applications in various day to day activities such as logistic chain management, access control, Inventory management, RFID smart cards, price tagging, various health monitoring devices etc. The communication in these devices is vulnerable to various attacks such as replay attacks, Denial of Service attacks, man-in-the-middle attack etc. To provide security against these attacks cryptographic algorithms for providing security are required to be embedded in these tags. The conventional cryptographic algorithms such as AES and ECC provide a highly secure encryption and authentication mechanism. But these algorithms require a larger area for their implementation. According to NIST, a low-cost RFID tag requires 1000-10,000 gate equivalents (GE) for its implementation. Out of these gates, around 200-2000 Gate Equivalents(GEs) are reserved for security. Since these tags are constrained in terms of area, memory, and power, so the conventional algorithms such as AES, ECC do not perform well in these tags. National Institute of Standards and Technology (NIST) has recommended using lightweight cryptographic algorithms for these low -cost devices.

This thesis work attempts to provide encryption-authentication scheme for low -cost RFID tags by adhering to the area constraints for security in these tags. For encryption, a lightweight block cipher BORON has been used as an encryption scheme. The 64-bit implementation of BORON required Gate Equivalents nearly equal to 2000 GE. To provide authentication some extra circuitry is required and hence exceeding the Gate Equivalents beyond 2000. So, a lightweight implementation has also been done by reducing the number of s-boxes and then embedding an authentication mechanism to it. Authentication has been done using an arbiter PUF. The experimental results show that for this lightweight implementation, encryption-authentication together required 106 FPGA slices and 3190 GEs thereby making this scheme suitable for providing security in low-cost RFID tags.

# TABLE OF CONTENTS

**Sr No .** **Chapter Title** **Page No.**

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| AES | : | Advanced Encryption Standard |
| ASIC | : | Application Specific Integrated Circuit |
| DES | : | Data Encryption Standard |
| DoS | : | Denial of Service. |
| CBC | : | Cipher Block Chaining |
| ECC | : | Elliptic Curve Cryptography |
| EPC | : | Electronic Product Code |
| FPGA | : | Field Programmable gate arrays |
| GE | : | Gate Equivalents |
| IEC | : | International Electrotechnical Commission |
| ISO | : | International Standard Organisation |
| IV | : | Initialization Vector |
| LUT | : | Look-Up Table |
| NIST | : | National Institute of Standards and Technology |
| OCR | : | Optical Character Recognition |
| PUF | : | Physical Unclonable Function |
| RFID | : | Radio Frequency Identification |
| SPN | : | Substitution and Permutation Network |
| VR | : | Voice Recognition |

# CHAPTER 1

# INTRODUCTION

## 1.1 AUTOMATIC IDENTIFICATION

Automatic identification (Auto-ID) deals with the identifying and locating the physical objects automatically by electronic transfer of data without any human interaction. The main goal of using automatic identification is to increase the efficiency and cost reduction by eliminating the need of human labour at entering data and thereby reducing the error caused by human [1]. There are enormous Auto-ID solutions such as Chip Cards, Biometrics, Radio Frequency Identification (RFID) tag, Voice recognition *etc.* available in market as shown in Figure 1.1.

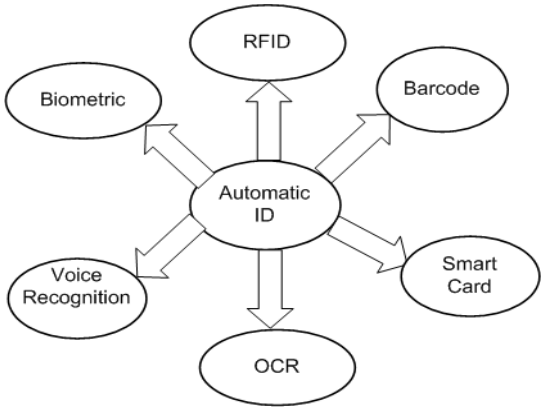



Figure 1.1 Automatic Identification Solutions

Table 1.1 provides the features of various automatic identification mechanisms. Among all of the auto-id solutions, RFID tags do not require a physical contact between the reader and the card. The RFIDs can connect to a reader wirelessly over a certain distance when an electromagnetic is provided by the reader. Due to the absence of human interaction, a large number of objects are identified in a lesser time, low cost and with higher reliability thereby making RFID technology suitable for applications such as tracking, electronic tagging, supply chain management, object monitoring *etc.*

Table 1.1 Features of Various Auto-ID Solutions [2]

| **Parameter** | **Barcode** | **OCR** | **VR** | **Biometrics** | **RFID** |
|---|---|---|---|---|---|
| Data Size(bytes) | 1-100 | 1-100 | N/A | N/A | 16-64 K |
| Readability by machine | Good | Good | Complex | Complex | Good |
| Readability by people | Partially | Easy | Easy | Difficult | Impossible |
| Affected by dirt/moisture | Strongly | Strongly | N/A | N/A | No influence |
| Initial Cost | Very Low | Medium | Very High | Very High | Medium |
| Max Distance reader/carrier | 0.5 cm | < 1 cm | 0.5cm | Direct Contact | 0.5m |

## 1.2 ADVANTAGES OF RFID [2]

- RFID readers do not require a line of sight to access data from the RFID tags. Therefore, with RFID systems data can be read over a long ranging from few centimetres to a few hundred meters.
- RFID systems can read and write different sizes of data from/to the tag, based on the type and the memory availability of the tag.
- Since RFID system do not require any human interaction for identification, this reduces the human error and the overall cost.
- RFID technology provides an additional feature of embedding a sensor in an RFID tag. Therefore it opens the door to sense various environment factors (temperature, moisture, humidity) and store them along with identification information in these tags.

## 1.3 RFID HISTORY AND STANDARDIZATION [3]

The passive communication technology used in RFID was first presented by Henry Stockman's seminal paper "Communication by means of Reflected Power" in 1948. IFF (Identified Friend or Foe)was the first RFID application developed by British Royal Air force during World War II. The Electronic Article Surveillance (EAS) was the first commercial RFID application that were used as theft prevention system in late 1960s. In typical EAS systems, a magnetic device was embedded in a product and it would be deactivated when an items was purchased. The first patent for active RFID tag with rewritable memory is claimed to have been received by Mario W. Cardullo on January 23, 1974. The major progress in RFID was seen in years 1980s and 1990s in different parts of the world. In 1987, the first RFID -based toll collection system was setup in Alesund, Norway. Table 1.2 describes the evolution of RFID over the decades.

Table 1.2 The Evolution of RFID Technology Over the Decades

| **The Decade of RFID** | |
|---|---|
| **Decade** | **Event** |
| 1940-1950 | RADAR redefined and used, major World War II development effort .RFID invented in 1948. |
| 1950-1960 | Early exploration of RFID technology, laboratory experiments. |
| 1960-1970 | Development of theory of RFID. Start of application field trials |
| 1970-1980 | Explosion of RFID Development. Tests of RFID accelerate. Very early adopter implementations of RFID. |
| 1980-1990 | Commercial application of RFID enter mainstream. |
| 2000-present | RFID explosion continues |

The main standards for RFID have been mainly developed by International Standard Organisation (ISO) and the International Electrotechnical Commission (IEC) [4]. The various existing standards for RFID are ISO 14223,11785,11784 for animal identification, ISO 18000 for item management, EPC HF1 for Electronic Product Codes, ISO 1443,15693 for contactless smart cards, IS0 18000-7 for item management (active tags).

## 1.4 RFID APPLICATIONS

The RFID tags gives many advantages since they have an ability to be read by the reader if they pass near a reader even if they are covered by some object or hidden inside some box or a container. These advantages allow the suppliers of various sectors to provide a secure and enduring support for their customers. It involves anti terrorism measures in airports, error free RFID in medical instruments and anti-counterfeiting RFID for drugs. This section gives few applications of RFID tags. The most important RFID applications focus on health care, ticketing, logistics, supply chain management, tolls, identification system *etc.* [5]. Figure 1.2 gives the percentage of RFID usage in various fields [2].

- ***Logistic and supply Chain Management***: This is the most famous application of RFID tags. The RFID tags are attached to each product, item and tool. These tags are read/tracked by RFID readers from the manufacturers. This allows the manufactures to get a good demand signals from their customers. RFID technology a better quality of service by helping the retailers to deliver right product at right time at right place. It also helps in maximizing the sales and the profits. A Motorola RFID plan has been deployed in the supply chain management of Megatrux, a top 100 logistic company.

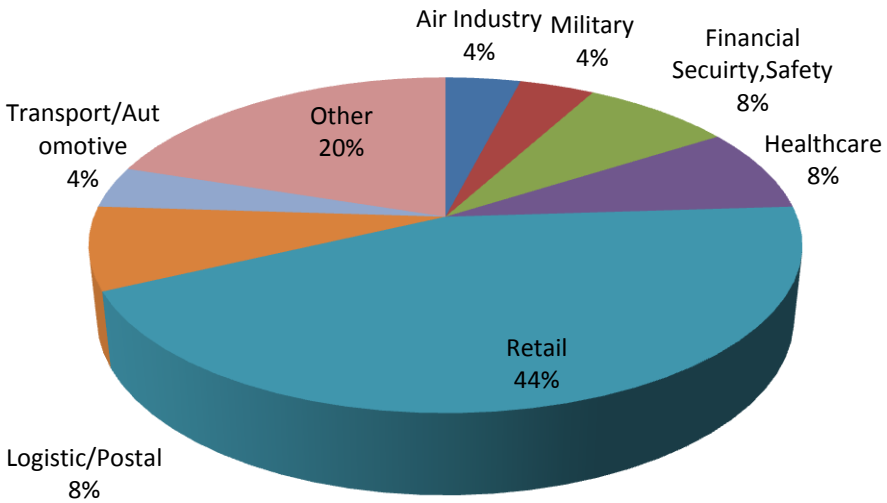



Figure 1.2 Global RFID Market Value in 2016

- ***Healthcare*:** There is a demand of high accuracy in healthcare in various aspects like drug distribution, identification and inventory management [6].
- ***Security*:** Security and personal identification are the broad and major applications of RFID. The most common use of RFID is in identification card to control building access. The USA Passports has RFID tags embedded in them. The RFID tags provide a more reliable and efficient storage of identification information in comparison to magnetic strips. Some high end security application allow the identification of people by embedding RFID tags under their skin to use in various fields like financial security, military and government security to control access to secure areas.
- ***Toll System and Payment Application*:** The RFID toll systems allows the vehicles to check in and checkout automatically in secure and convenient environment. But, vehicles must be queued up in line and must pass one by one through the toll gate. There are various other payment applications such as contact-less credit cards. But these cards require high level of security.
- **Tracking Applications:** There are various tracking applications making use of RFID technology such as
    - **Animal Tracking:** This application allows the user to keep track of livestock for preventing disease outbreak. It can also be used by the pet owners to keep track of their pets if they are lost.
    - **People Tracking:** This application is used mostly in jails and hospitals. In hospitals, this is required to monitor patients needing special or mental care and to monitor the new born babies.
    - **Asset Tracking:** This application is used in services like postal services and vehicle traffic monitoring.

## 1.5 RFID SYSTEM OVERVIEW [7]

The basic idea behind the RFID systems is that you mark the items with the tags. These tags contain a transponder that emits the identification number that can only be read by specialized RFID readers. RFID tags store some identification numbers. Identification number can be anything from customer number to product Stock Keeping Unit code. A reader retrieves the information about the identification numbers from database. RFID tags also contain a writable memory that can be used to store information to transfer to various RFID users in different locations. This helps to track the movement of the item and make this information transparent to each RFID reader.

The RFID systems basically consist of three main components

- A tag/transponder
- A middleware
- A reader.

- **RFID Tag**: RFID tags are data carrying part of an RFID system. The main purpose of the RFID tag is to carry the identification information of an item to which it is attached to. These tags are placed/embedded in an item to provide a unique identity to it. The data contained by these labels can either be as short as bits or it can be a gathering of extensive information, for example, personality code of a creature, restorative data of a man and so on.
- **RFID Reader**: RFID reader is a device that is used to transmit and receive the data through radio waves. The reader consists of a powerful antenna and a power supply to surround itself with an electromagnetic field to actuate the tag and retrieve /read the information contained in them. The identification data read by the RFID reader is processed by the software system, known as the **RFID middleware.** The RFID middleware manages readers, as well as filters and formats the RFID raw tag data so that they can be accessed by the various interested enterprise applications Hence, the middleware is a key component for managing the flow of information between tag readers and enterprise applications

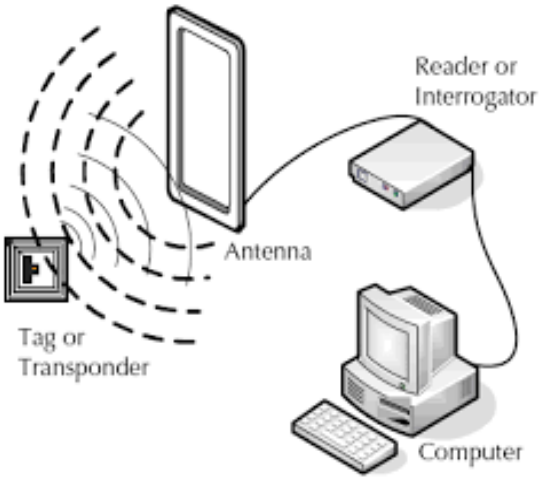



Figure 1.3 An RFID system

## 1.6 CLASSIFICATION OF RFID TAGS

RFID tags can be classified in two categories based on their power and memory requirements.

### 1.6.1 Power Source Based

Silicon based RFID tags are basically classified in three categories based on their power supply: semi-active tags, passive tags and active tags

**1. Passive RFID Tags [8]**

Passive RFID tags are the simplest RFID tags that do not have their own power supply and they cannot initiate any communication with the reader. The passive tag derives its power from energy wave transmitted by reader and it gives response to reader's radio frequency emission. Therefore, the passive RFID tags relies completely on its reader for their power requirements. Compared to all the other tags, these tags are cheaper and are smaller in size but their cover range is smaller. Due to the absence of battery source, these tags can be used for a large time period. Due to all these features, these tags

are most commonly used tags in various applications. Table 1.3 gives the advantages and disadvantages of passive RFID tags.

Table 1.3 Advantages and Limitations of Passive RFID Tags

| Advantages | Limitations |
|---|---|
| o Small size and cost as compared to active and semi passive tags.<br>o Lightweight<br>o Low cost<br>o Resistant to harsh environment conditions | o Requires presence of interrogator to work.<br>o Limited memory to store data<br>o Low read range ( few inches to few metres) |

**2. Active RFID Tags [8]**

An active RFID tag has an onboard power supply i.e. a battery and an active transmitter. The presence of an on board battery allow the active tags to initiate the communication and actuate themselves without any reader. These tags can cover a large range as compared to other tags because of the presence of a power source. The lifetime of these tags is restricted by the capacity of the battery. These tags can communicate over hundreds of feet and has a large memory of up to 3 kilobytes. In comparison to other types of tags, the active tags are more costly. Basically two different types of active RFID tags are available: transponders and beacons

- Transponders: A system that uses an active RFID tags, the signal is first sent by the reader and then the active transponder will send a signal back with the relevant information. Transponder tags provide high efficiency as they conserve battery life when tag is out of read range. Active RFID tags are commonly used in secure access control and toll booth payment systems.
- Beacons: In a system where beacon tags are used, the tag will not wait for reader's signal, but will send out information every 3-5 seconds. Beacon tags are commonly used in oil and gas industries, mining, cargo tracking applications. Active tag beacons can read through hundreds of metres .To deal with the harsh environmental conditions, most of the active RFID tags are enclosed in a rugged shell. Because of the size of the enclosed battery, Active RFID tags are much larger and bulkier than passive RFID tags. Table 1.4 lists the advantages and disadvantages of active RFID tags.

Table 1.4 Advantages and Limitations of Active RFID tags

| Advantages | Disadvantages |
|---|---|
| o More memory, can store more data | o Sensitive to harsh environment. |
| o Reduced power from interrogators | o Limited battery life . |
| o Longer read range (upto 100+ feet) | o Large size and bulky. |

**2. Semi-Passive RFID tags**

Semi-passive tags also have their own power supply. These tags cannot transmit the signal when their battery gets discharged. The difference between active tags and passive tags is that unlike passive tags, these tags do not have any active transmitter. These tags use backscattering technique to transmit the information. In this technique, RF energy from the reader is collected and altered to transfer the data in a reader understandable manner. Table 1.5 gives the comparison of active, semi-passive and active RFID tags.

Table 1.5 Comparison of Passive, Semi-Passive, and Active RFID tags

| Type of the Tag | Passive | Semi-passive | Active |
|---|---|---|---|
| Power Supply | Energy | Battery Source | Battery Source |
| Maximum Range | 10m | >100m | >100m |
| Memory | Read-Only | Read-Only | Read-Write |
| Application | EPC | Tolls | Asset-Tracking |

### 1.6.2 Memory Based

Based on the type of memory, RFID tags can be classified as

- *Read Only Memory Tags*: These tags allow only read operation to recuperate the data.
- *Read/Write Memory Tags*: These tags allow both reading and writing operations.

## 1.7 SECURITY CHALLENGES IN RFID TAGS

With the rapid use of RFID in various fields and industries, the security issues surrounding RFID and the challenges of providing security services, to meet the cost and interoperability requirements of the business process, with a resource limited device have been of important concern in academics, government, and industrial applications. Although RFID tags can offer a variety of security mechanisms, yet there is some possibility of attacks on the RFID systems. The various attacks on RFID tags are listed below

- **Eavesdropping:** This threat is the main privacy concern in RFID tags. Eavesdropping occurs when an unauthorized RFID reader listens to conversations between a tag and reader and obtains the important data one for the correct tag family and frequency-while the tag is be-

ing read by an authorized RFID reader. Since clear text communication is used in most of the RFID systems, eavesdropping is a simple but is an efficient method for an attacker to obtain sensitive information on the collected tag data. This threat becomes very serious when sensitive information like data of credit/debit card is exchanged on the channel without any encryption algorithm [9].

- **Snooping**: Snooping refers to unauthorised reading of tag's information. Snooping is said to have occurred when the tag's data is read by an illegitimate reader without the proprietor's knowledge. This attack occurs because most of the tags transmit their data in memory without authentication.
- **Replay Attacks:** This attack occurs when an eavesdropped information is replayed by an unauthorized user communication to achieve the same results that an original reader and tag would have achieved [10]. An application may be that an illegal device playbacking the authentication between the reader and the tag, fooling the both to pass the verification.
- **Man-in-the-Middle Attack/Relay Attacks:** A man-in-the-middle attack takes place during the transmission of a signal. In this type of attack, a device is placed by an attacker between the reader and the tag to intercept the information between two nodes. After this, original information is modified and then sent to the other end. The modified information through the middle device will have some delay and hence these attacks are also called relay attacks.
- **Counterfeiting and Spoofing Attacks:** On receiving information about the tag's identity, an attacker can clone the tag. This identity information is then used to access the RFID system to pretend as an authorized reader. This is called spoofing. The effective measure to prevent these attacks is to use a two-way authentication system for mutual authentication between the tag and the reader.
- **Denial of Service Attack:** This attack causes system to not work in an appropriate manner. An attacker blocks the reader from reading the tag by making use of a blocking tag. Denial of Service tags are a threat to all the modern communication systems.

To resolve all these attacks, cryptography algorithms are employed in theses tags to provide information security.

## 1.8 CRYPTOGRAPHY IN RFID TAGS

The channel connecting the RFID reader and the database are wired links that are assumed to be secure as both the reader and the server apply strong cryptographic protocols. But, on the other hand a wireless channel exists between the tag and the reader. This wireless communication is vulnerable to a large number of malicious attacks. So, there emerges a need for cryptography to provide security in RFID tags. Figure 1.4 provides an overview of security enhanced RFID system. An additional crypto module is added to RFID tags to provide the information security. A low-cost RFID may require a

total gate count of 1000 to 10,000 Gate Equivalents (GE). Out of these, around 200-1000 gate equivalents are reserved for security purposes [11, 12]. So, a cryptographic algorithms requiring maximum of 2000 GEs providing an adequate security needs to be embedded in these tags.

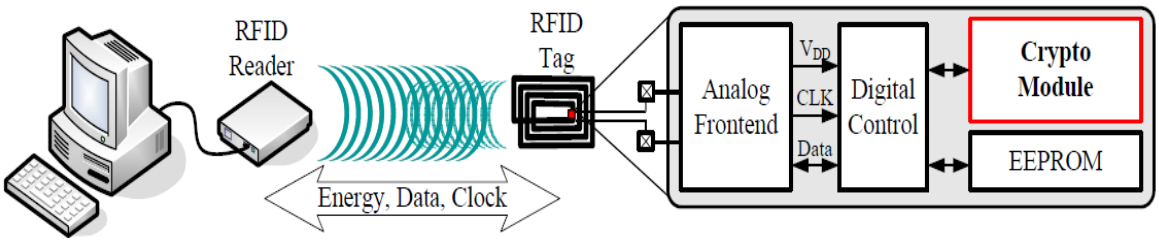



Figure 1.4 A Security Enhanced RFID System [11]

## 1.9 OVERVIEW OF CRYPTOGRAPHY

Cryptography is the science of concealing the original message for providing secrecy in information security. The word "cryptography" is coined from two Greek words, "Krytpos" and "Graphene" meaning secret writing. Cryptography is the study of various techniques for keeping the information between two parties private in presence of third party. Figure 1.5 gives the basic block diagram for cryptography. Main terms related with cryptography are given below:

- **Plaintext:** Plaintext is the original message that has to be encrypted. Different algorithms and protocols are used to encrypt data such that it hides the actual information.
- **Key**: It is used in encryption process along with the plaintext.
- **Encryption:** Encryption is the process of converting the original data into a secret message confidential form at transmitter side.
- **Cipher text:** The encrypted data which is in intelligible form is termed as cipher text.
- **Decryption:** It is the process of converting encrypted message back to original message at receiver side.

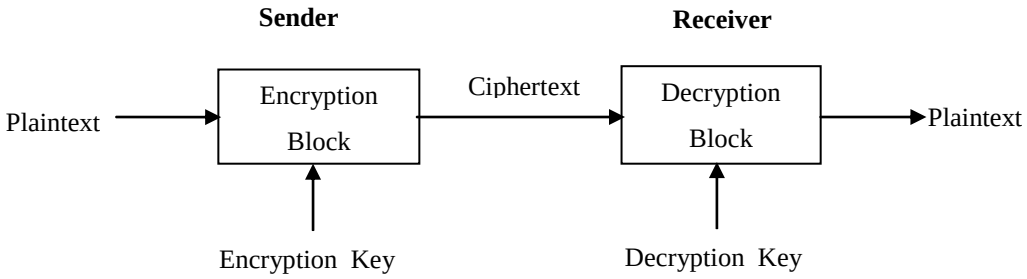



Figure 1.5 Basic Block Diagram for Cryptography

### 1.9.1 OBJECTIVES OF CRYPTOGRAPHY [13]

The main goal of cryptography is to provide authentication, integrity and security to the transmitted message.

- **Authentication:** Authentication refers to the verification of user's identity. At the receiver end it is checked whether the received data is transmitted by an authorized user or any wrong identity. It ensures that any kind of intruder doesn't have access to sensitive data.
- **Confidentiality:** Confidentiality ensures that any third party or an adversary does not get any access to data. Symmetric and asymmetric cryptographic techniques are applied to achieve confidentiality. Only the authorized user can encode or decode data. Cryptographic key is applied to encrypt the intelligible information and by using either same or different cryptographic key, recipient decrypts that message.
- **Non-Repudiation:** This ensures that data has been actually sent by the transmitter and received by the receiver. It means that the data has been sent and received by the parties that are claiming to send and receive data.
- **Integrity:** It ensures that the data received at receivers end is same as the data sent from transmitter end. It also ensures that any alteration can be done by the authorized party only. Recipient utilizes the same procedure as used at sender side to create a digest from the message for comparing it with the original message. This procedure ensures the integrity in data.

### 1.9.2 Types Of Cryptography

Based on the type of key, a cryptographic algorithm can be classified in symmetric and asymmetric cryptography. Figure 1.6 gives the classification of cryptographic algorithms.

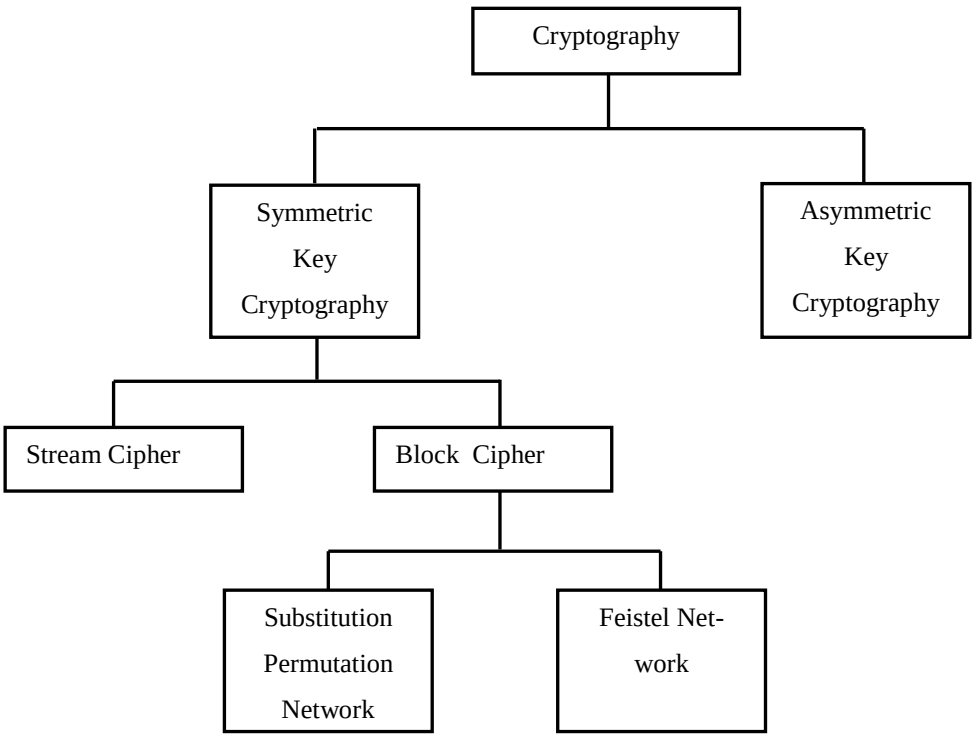



Figure 1.6 Classification of Cryptography Algorithms

**1. Symmetric Key Cryptography:** Symmetric key encryption algorithms are the oldest cryptographic techniques used to have a secure communication. These algorithms make use of use of same key to encrypt and decrypt the data. This type of cryptography is also termed as private key cryptography [14]. The encryption and decryption process are just reverse of each other. The main shortcoming of this type of cryptography is that all the authorised parties need to exchange the key for encryption process before the decryption process. The main point of concern in Symmetric Key Cryptography is the way in which key is shared for encryption and decryption. Figure 1.7 gives the basic block diagram for symmetric key Cryptography. Various examples of private key encryption algorithms are DES, AES, Blowfish, Welch-Gong Based Stream Cipher *etc.*

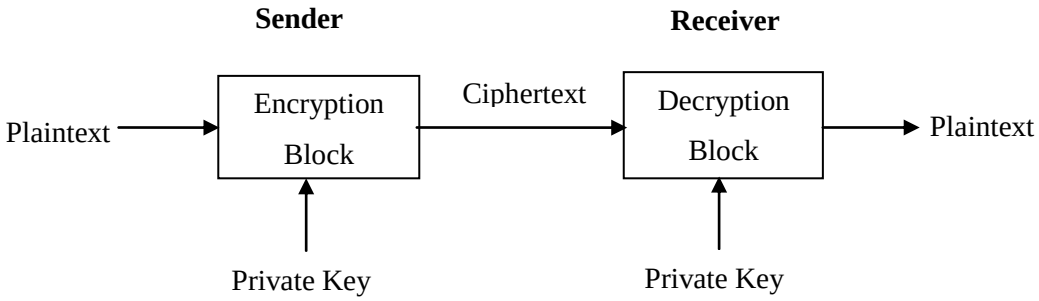



Figure 1.7 Block Diagram for Symmetric Key Cryptography

The symmetric algorithms are relatively fast as their operations are simpler also only single key is required to encrypt and decrypt the data. The main limitation of these algorithms is that they do not provide authentication and hence require an another algorithm to authenticate [15]. Some examples of symmetric cryptography algorithms such as Advanced Encryption Standard(AES), PRESENT, Data Encryption Standard(DES), Blowfish etc.

Symmetric Cryptographic algorithm can be further classified into two categories i.e. Stream Cipher and Block Cipher

**i. Stream Ciphers**

Stream Ciphers process one bit at a time. Stream ciphers imply infinite length pseudorandom sequence of bits as a key to encrypt data. Cryptographic algorithm is applied to each bit of data with corresponding bits of key stream to produce cipher-text bit. Stream ciphers generally imply two operations: Mixing and XORing [16]. The main advantage of theses ciphers is that these are less susceptible to chosen plaintext attacks as each time a different bit is produced. They are best suited in case of unknown data as encryption of one bit do not depend on surrounding bits.

**ii. Block Ciphers** [16]

Block ciphers involve the encryption of a data block by applying key and algorithm. In this, encryption and decryption takes place on a block of data (generally 64 bits or 128 bits). Block ciphers cannot be modified easily and are appropriate for the encryption of large data size. Due to a large number of bits in a block, these block ciphers require a high memory to process the bulk data. Since a bulk of

data is processed, so even a single bit change may affect the whole block. The block ciphers are susceptible to plaintext attacks as identical plaintext-key pairs produce identical ciphertext.

Block ciphers can be further divided into two categories:

- Feistel Networks
- Substitution Network

- ***Feistel Network:*** Feistel networks are for construction on any symmetric block ciphers. In feistel networks the data block is partitioned into two halves i.e. left half (L) and right half (R). It is a round based structure where an internal round function is iterated repeatedly. Figure 1.8 illustrates the typical fiestel network for cryptographic algorithm. In each round encryption function depends on the right half (R) and key (K) is applied to left half keeping right half (L) unchanged. The XORing of output F (K, R) is done with the right half. At the end of each step permutation layer swaps the modified L and unchanged R. For the next round R becomes L and L become R of current round. These substitution and permutation stages and number of rounds are specified by the algorithm design. Figure 1.8 gives the basic block diagram for Fiestel Network. In feistel network, encryption and decryption is same except for the reversal of key generation. This is the biggest advantage of feistel networks. Due to similarity between encryption and decryption, Feistel networks require a lesser area.

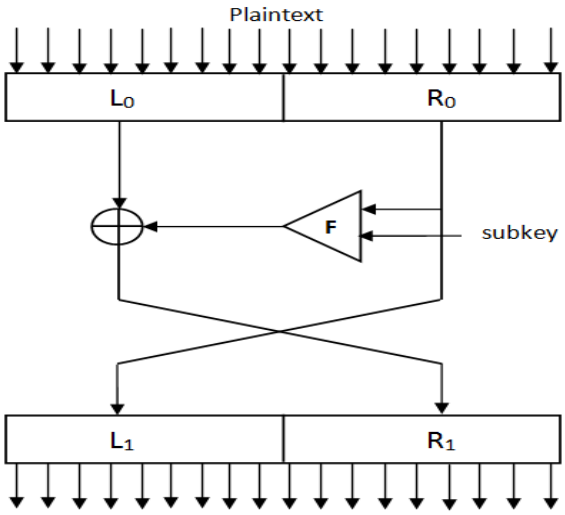



Figure 1.8 Block Diagram of Fiestel Network

- ***Substitution Permutation Network*** **(SPN):** Substitution permutation is an another type making a use of substitution and permutation layer for processing the ciphertext. It involves the passing of plaintext and key through a s-layer followed by a player to make a higher degree of confusion in ciphertext. Figure 1.9 illustrates the basic block diagram of a substitution and permutation network. As the name specifies, substitution boxes substitute the small block of bits with the other small block of bits but size of both the blocks is same. Permutation layer is

just shuffling of bits by routing to create diffusion. Output of S-Box of one round is taken and shuffled to different positions bitwise. Player efficiently distributes the data to as many S-BOXes as possible. Distinct round keys are also produced for each round. Ciphertext is produced after a particular number of rounds as specified by the algorithm. Table 1.6 provides the comparative analysis of Block and Stream ciphers.

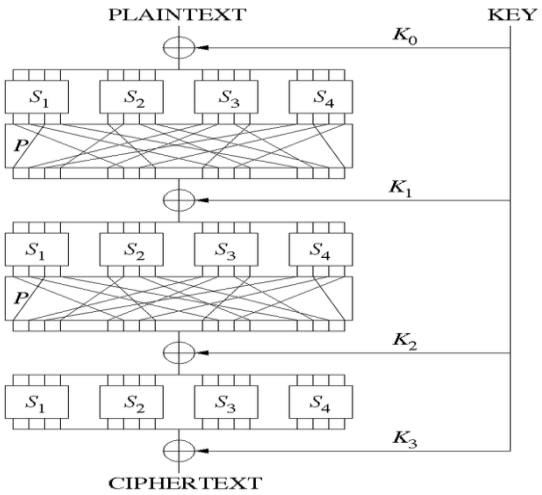



Figure 1.9 Block Diagram of a Substitution Permutation Network

Table 1.6 Comparison of Block and Stream Cipher

| Block Cipher | Stream Cipher |
|---|---|
| ○ Encryption of a particular block depends on the previous blocks. | ○ Encryption of a particular bit is independent of its surrounded bits. |
| ○ Key used for encrypting whole block of data is same. | ○ Distinct key is used to encrypt each block of data. |
| ○ Padding is required in case of short blocks. | ○ Bits are continuously processed as a chain. |
| ○ Since a complete block of bits is processed at a same time, so block ciphers are relatively slower and require a large processing time | ○ Only a single bit is processed at a time. Therefore stream ciphers are less complex and fast |
| ○ Examples: IDEA, AES, DES, SERPENT, PRESENT, BORON *etc.* | ○ Examples: TRIVIUM, GRAIN, RABBIT, FISH, RC4 *etc.* |

2. **Asymmetric Key Cryptography:** The idea of asymmetric key cryptography was first projected by Diffie and Hellmann in 1976. This type of cryptography make use of public key for encryption and private key (secret key) for decryption. Since keys for encryption and decryption is different, these

algorithms are called asymmetric key algorithms. It is also termed as public key cryptography. The main property of these algorithms is the difficulty of finding the secret key while everyone knows the public key [13]. Keys used for encryption are circulated publicly but decryption keys are only known to the recipient. Therefore, everyone can encrypt a message with the particular party's public key but only that particular party can decrypt the message. In this way, no one else can recover the message. Figure 1.10 illustrates the block diagram for asymmetric key cryptography. Examples of asymmetric key algorithms are Rivest Shamir Algorithm (RSA), Digital Signature algorithms, ECC *etc*.

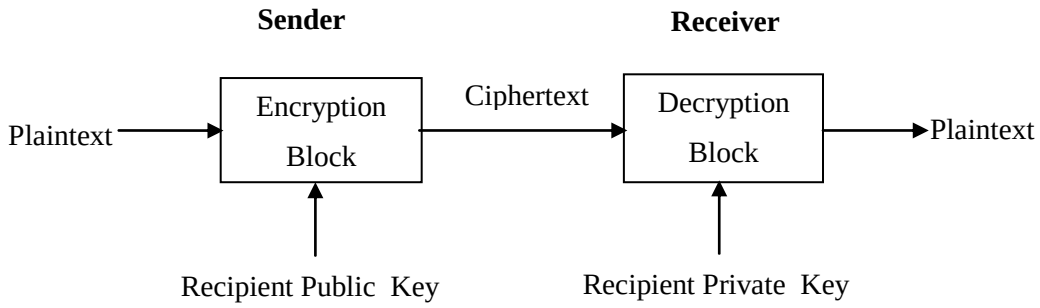



Figure 1.10 Block Diagram for Asymmetric Key Cryptography

The main advantage of two different keys is that these two keys are interchangeable, means a sender can encrypt the data with its private key and receiver can decrypt the data with sender's public key. This property is used is used to provide authentication and non-repudiation. But these require a large key size for exhaustive searching, brute force attack on public key. Also these ciphers rely on complex mathematical computation and hence they are more resource and power hungry in comparison to symmetric key algorithms [17].

### 1.9.3 Lightweight Cryptography [12]

The deployment of small computing devices such as RFID tags, wireless sensor nodes ,smart cards etc is becoming more common these days. The shift from desktop computers to these tiny devices brought a wide range of new security and privacy concerns. It is very challenging to apply conventional cryptographic algorithms as they require a larger area, a larger power and high computation. So National Institute of Standard and Technology came up with a research idea for the lightweight ciphers for these tiny devices.

Lightweight cryptography is a sub field of cryptography that was tailored with an aim to provide solutions to resource-constrained devices. In 2013, NIST initiated a lightweight cryptography project to study the performance of conventional algorithms on lightweight devices and to develop the lightweight algorithms for these devices.

Lightweight Cryptography targets devices that can be implemented on a broad spectrum of hardware and software. On the high end of this device spectrum there are server and desktop computer followed

by mobiles and tablets as shown in Figure 1.11. Since these devices has a larger memory, larger area, requires a greater security and hence conventional algorithms performs pretty well in these devices. Therefore, it filters out the need for lightweight cryptography in these devices. On the lower end of the spectrum, there are devices such as embedded systems followed by RFID tags and wireless sensor nodes at the bottom end. These bottom end devices are often realised on Application Specific Integrated Circuit (ASIC) to satisfy some of the most stringent implementation constraint. For the passive RFID tags not having battery source, only a limited amount of power is available from environment. Such devices require cryptographic algorithms that should not only consume less Gate Equivalents but should also stick to stringent power and timing requirement.

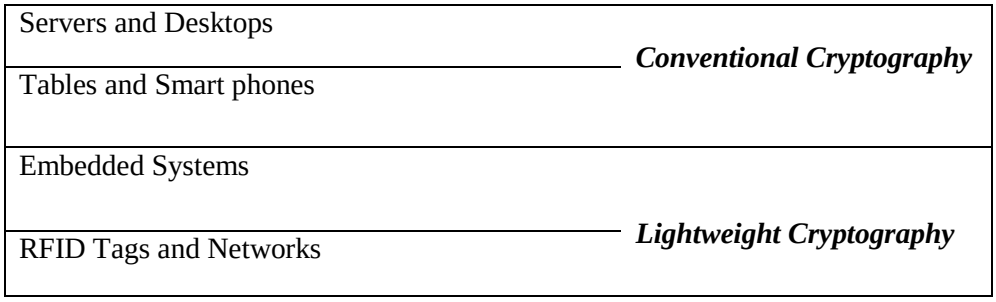


Figure 1.11 Device Spectrum [6]

### *1.9.3.1 Lightweight Cryptographic Primitives*

Over the last decade, a large number of lightweight cryptographic primitives including block ciphers, stream ciphers, hash functions and Message Authentication Codes (MAC) have been developed to provide security in these low-cost devices. The lightweight ciphers differs from conventional algorithms in such a manner that these ciphers are not intended for a wide variety of applications. The performance benefits of lightweight block ciphers over conventional cryptographic algorithms have been achieved using design choices such as

- ***Smaller Block Size:*** To save the memory, the lightweight block ciphers have been designed with the lesser block size than AES (e.g. 64-bits or 80-bits rather than 128-bits).The reduction in the block size reduces the limit on number of plaintext blocks to be encrypted.
- ***Smaller key sizes:*** Some block ciphers uses the smaller key size (less than 96 bits) for efficiency .For e.g. PRESENT uses a key size of 80-bits.
- ***Simpler Rounds:*** The operations used in lightweight block ciphers are usually simpler than those of conventional algorithms. In lightweight designs, a 4bit-4it S-BOX is used instead of 8-bit S-BOX. This reduction in the size of S-box leads to the reduction in area. For e.g. 4-bit S-Box in PRESENT required 28 Gate Equivalents whereas AES s-box required 395 gate equivalents [18]. Since the key size is small and the rounds are simpler in lightweight cryptographic algorithms, so in order to increase the security, number of rounds have to be increased (AES has 10 rounds but PRESENT has 32 rounds, HIGHT has 25 rounds).

- ***Simpler Key Scheduling:*** Most of the lightweight ciphers uses simple key scheduling that can generate the sub round keys on the fly.

Some examples of NIST recommended lightweight ciphers are: PRESENT, CLEFIA, PICOLLO, TWINE, PRINCE *etc*. PRESENT is one of the first lightweight block cipher designed for resource constrained devices.

In this work, a new lightweight block cipher BORON has been implemented along with a Physical Unclonable Function (PUF) based Authentication for resource constrained devices.

## 1.10 ORGANISATION OF DISSERTATION

**Chapter 2** surveys the various authentication and encryption schemes that have been worked upon for RFID tags in the recent years. **Chapter 3** identifies the gaps in the study and proposes a new light-weight authentication encryption scheme. **Chapter 4** gives the brief overview of encryption and authentication method that has been opted for this work. **Chapter 5** gives the implementation results. **Chapter 6** suggests some future directions that can be worked upon further.

# CHAPTER 2

# LITERATURE REVIEW

Since RFID tags have become an important aspect in today's digital world providing identity to almost all the objects in this world. These tags are vulnerable to various attacks as specified in section 1.7. Therefore, various cryptographic techniques have been proposed to provide the security in these low cost RFID tags. This section reviews the various cryptographic solutions for theses RFID tags.

**Martin Feldhofer, *et al*.** [18], authors proposed an encryption and authentication based protocol using Advanced Encryption Standard(AES) as a cryptographic approach for RFID tags. The protocol was embedded in ISO/IEC 18000 standard for RFID. The implementation has been done for 128 bits with 0.35μm technology. For reduced area and power, the datapath was reduces to 8-bits.The implementation result showed that the design required a total of 1,016 clock cycle for encryption of 128-bits . The hardware complexity was estimated to be around 3595 gate equivalents.

**Andrey Bogdanov, *et al*.** [19], presented an ultra-lightweight block cipher PRESENT for resource constrained devices such as smart cards, RFID tags etc. PRESENT cipher is a symmetric cipher supporting 64-bit block length with two key sizes of 80-bit and 128-bit. PRESENT provides a good security with an avalanche effect of 49.2%. The roundbased architecture was implemented while generating the keys on the fly. The implementation results showed that it required 1570 Gate Equivalents with 80-bit key size.

**Andrey Bogdanov, *et al***. [20], introduced a concept of building a hash function for authentication using PRESENT and AES algorithms to be used in RFID protocols. The author described a hash function that give 64-bit and 128 bit outputs. The implementation was done for both round based and serial architecture. The synthesis was done using Synopsys Design Compiler version Z-2007.03-SP5 with UMC 180nm technology. The results showed that hash with round based architecture consumed 4500 gate equivalents for PRESENT and 9800 gate equivalents for AES.

**Gregor Leander, et *al*.** [21], proposed a serialized lightweight architecture using DES (DESL) for RFID tags. The ASIC implementation of this architecture with 0.18μm technology reported that it required 2168 gate equivalents, 144 cycles to encrypt a block of 64-bit, 5.55 KB/s of throughput at 100KHz.

**Deukjo Hong**, ***et al*.** [22], presented a lightweight block cipher HIGHT with 64-bit block length and a key sie of 128 bits. The hardware realisation of this block cipher consumed 3048 Gate Equivalents with 0.25μm technology.

**Yadollah Eslami, *et al.*** [23], introduced hardware implementation of both public and private key algorithms (AES, DES, ECC) in a single universal processor for use in smart card applications with required power and performance specifications. DES is for providing best past compatibility, ECC has best encryption efficiency among public key algorithms and AES is used for its high throughput and security. The resulting cryptographic processor of three algorithms occupies 2.25mm$^2$ area in 0.18μm

technology which is 9% of the available area of smart cards. Area overhead of the crypto processor is reduced by using FeRAM instead of non-volatile SRAM memory.

**Chae Hoon Lim, *et al.*** [24]**,** presented a new lightweight block cipher m-Crypton with three key variants i.e. ( 64 bits,96 bits,128 bits) for low cost RFID tags. The hardware implementation was based on 1cycle/round architecture. The implementation required around 4500 to 5100 gates for encryption and decryption and 4000 gates for encryption only depending on key size. The results showed that the hardware complexity of m-Crypton is still well suited for low-cost RFID tags and sensors.

**Gaurav Bansod, *et al.*** [25]**,** presented a new lightweight block cipher ANU for low-cost devices. ANU cipher has 25 rounds and it supports two key variants 80bit/128 bit. The hardware implementation of ANU cipher required only 984 Gate Equivalents for 128-bit key. On comparing the performance comparison of ANU cipher with other lightweight ciphers it was observed that it consumed 30.24% lesser gate equivalents than PRESENT, 6.60% lesser GEs than SIMON, 17.12% lesser GEs than SPECK and 29.98% lesser GEs than PICOLLO.

**Gaurav Bansod, *et al.*** [26]**,** presented an ultra lightweight, low power BORON block cipher to provide security in small device such as wireless sensor nodes and RFID tags. BORON is an SPN network having 64-bit block size and two key sizes of 80-bit and 128-bit. Its design helped generating a large number of active S-boxes in less number of rounds, preventing it from linear and differential attacks. The experimental results showed that BORON required 1939 gate equivalents for 128-bit key and 1626 gate equivalents for an 80-bit key. Power consumed by the BORON cipher is also less as compared to other block ciphers. BORON also has a 297.43%, 96.00%, 33.36%, 88.27% higher throughput that PRESENT, LED, KLEIN, HUMMINGBIRD cipher. All these features of BORON make it suitable to be used in applications requiring a lesser footprint area and low power.

Table 2.1 gives the comparison of various lightweight ciphers for RFID tags.

Table 2.1 Comparison of Various Lightweight Cryptographic Ciphers

| Cipher | Key Size | Block Size | Clocks per block | Throughput at 100KHz | Logic Process | Gate Equivalents |
|---|---|---|---|---|---|---|
| PRESENT-80 [19] | 80 | 64 | 32 | 200 | 0.18 µm | 1570 |
| PRESENT_128 [27] | 128 | 64 | 32 | 200 | 0.18 µm | 1886 |
| BORON_80 [26] | 80 | 64 | 25 | 256 | 0.18 µm | 1626 |
| BORON_128 [26] | 80 | 64 | 25 | 256 | 0.18 µm | 1929 |
| DESL [21] | 56 | 64 | 144 | 44.4 | 0.18µm | 2168 |
| m-Crypton [24] | 96 | 64 | 13 | 492.3 | 0.13 µm | 4500 |
| HIGHT [22] | 128 | 64 | 34 | 188.2 | 0.25 µm | 3048 |
| AES [18] | 128 | 128 | 1032 | 12.4 | 0.35 µm | 3400 |

Authors **Carsten Rolfes**, ***et al.*** [27], presented a parallel and serial architectures for PRESENT algorithm. In parallel architecture thirty one time the loop was unrolled .To reduce the critical path delay, flip flop as pipelined registers were added. This architecture consumed a higher chip area and power. To reduce the chip area for making PRESENT suitable to be embedded in low cost RFID tags a serialized architecture was implemented. The datapath was reduced to 4-bits using only one s-box per clock cycle. It required 20 clock cycle to initialise the data and 15 clock cycles for one round. The results showed that this implementation required 1000 GEs giving a throughput of 11.4 Kbps with a latency of 563 clock cycles**.**

**Luo, *et al.*** [28]**,** proposed an efficient hardware implementation of Elliptic Curve Cryptography to provide authentication in RFID tags with reduced number of registers, reduced frequency of operations. The authors also used the restructured formulas in order to meet the resource limitations of RFID tags. The implementation was done for 226 bits and it was observed that it required a total of 16,900 gates with an operating frequency of 1280 kHz. But the proposed hardware is very far from resource requirement of low cost RFID tags.

Various authors in [29-31] have worked towards making ECC as a lightweight algorithm by reducing the flexibility of ECC algorithm by reducing parameters such as using only one elliptic curve, selecting specific prime numbers or choosing specific field sizes. Comparison of the results of all these works on ECC have been summarized in Table 2.2.

Table 2.2 Various Hardware Implementation Results for Elliptic Curve Cryptography

| Author | Bits | Gates | Technology (µm) | Operating Frequency (kHz) | Computation Time (ms) | Power (µW) |
|---|---|---|---|---|---|---|
| **Luo, *et al.*** [28] | 226 | 16900 | 0.18 | 1280 | N/A | 6.6 |
| **Kumar and Parr** [29] | 131 | 11,969 | 0.35 | 13560 | 18 | -- |
| **Gaubatz, *et al.*** [30] | 100 | 18,720 | 0.13 | 500 | 410.45 | <400 |
| **Lee, *et al.*** [31] | 163 | 12,506 | 0.13 | 1130 | 244.08 | 36.63 |

**E. Allen Michalski**, ***et al.*** [32], presented a hardware implementation of public key algorithm RSA with Montgomery multiplier design. The design for 1024-bit RSA core was synthesized for a 100 MHz Virtex2 Pro 100 FPGA platform. The implementation results showed that it required 12,791 FPGA slices and 12,162 FPGA slices for RSA and sub-pipelined RSA.

**Vinita Shadangi, *et al.*** [33]**,** proposed to use CBC-AES mode with multiple levels of encryption to provide security to digital images. In Cipher Block Chaining (CBC) mode, a unique block of cipher data is generated for the same block of input data. In order to make unique block of cipher data, an Initialization Vector (IV) of certain length had to be assigned to the very first block. It uses a chaining mechanism for encryption/decrypting a complete message. The encrypted text from first block is fed as an IV to the next block.

**Mihir Bellare, *et al.*** [34] explained the need of Initialisation Vector (IV) in CBC mode for generating a unique message every time. This IV was a random number generated by a Pseudo Random Number Generator (PRNG). Pseudo randomness is unpredictability in such a manner that given an initial sequence produced by a (PRNG), it is hard to predict the next bit in the sequence. For same seed, a same response will be produced.

**G. Edward Suh and Srinivas Devadas** [35, 36] proposed the concept of an arbiter PUF using a Multiplexers and an arbiter(latch) for authentication in RFID ICs. As it is impossible to model, copy or control the IC manufacturing process. So, the PUF implementation make these chips unique and unclonable.PUF implementation is well suited for these ICs as they require less cost (area) and security. 64 PUF chains producing 64-bit response were implemented for providing a non-volatile EPC code for RFID. The ASIC implementation showed that it required 0.02mm$^2$ of chip area with 0.18μm technology. The authors concluded that these can be easily embedded in passive tags with very little overhead in area.

**Jason H. Anderson** [37], implemented a 64-bit Physical Unclonable Function for Xilinx Virtex-5 65 nm FPGA board and implementation results showed that the 128-bit PUF uses less than 2% of Look Up tables in FPGA.

**J. J. Tay, *et al.*** [38], The author presented an area optimised round based architecture for PRESENT by adding a 64 bit encryption register in datapath. The s-box implementation was done in two ways one by using LUT based approach and other by using the minimal Boolean expression for s-box .Use of boolean expression resulted in an higher performance in terms of throughput. The proposed design used a 8 bit data path and consumed 62 FPGA slices giving 13.56 MHz, of maximum frequency and a higher throughput.

**Authors Mohamad Sbeiti, *et al.*** [39], presented an efficient round based architecture of PRESENT for low cost devices such as RFID tags. The cipher logic was implemented using three state FSM .The first round began after reset .Two registers 64 bit and 80 bit were used to select the appropriate value of text and key based on roundcounter. S-box implementation was done in two ways (Look Up Table and boolean expression).The results concluded that boolean expression resulted in higher throughput (516) and less gate count for (253) for PRESENT_80. But, throughput from LUT based technique was more for boolean based logic.

**Carlos Andres Lara-Nino**, ***et al.*** [40], proposed an area and power efficient architecture of PRESENT for low cost applications like RFID tags. The data was partitioned into for 16 bit blocks. Each word was stored in four separate 16 bit registers. The data in registers is shifted to right for reducing the size of MUX that was required to load data in the registers. Only four s-boxes were used in 16 bit data. path. The same 16 bit data structure was followed for key register by using 5 key register of 16 bit each. The proposed design gave a reduced area of 170 LUTS, latency of 133 clock cycles and a throughput of 123.86 Mbps for PRESENT_80, 220 LUTs, latency of 136 clock cycles and throughput of 99.13 Mbps for PRESENT_128.

**Panasayya Yalla, *et al*.** [41], worked on lightweight implementations(16-bit) of algorithms used in cryptography i.e. HIGHT and PRESNT for Field Programmable gate arrays. In this paper, implementation of HIGHT on FPGA use 117 slices and PRESENT use 91 slices only which are less than half the size of the Advanced Encryption Standard algorithm implementation (without using block Random Access Memories). Throughput over area ratio of PRESENT is 240 Kbps per slice i.e. similar to that of the AES algorithm and conversely HIGHT smashed them by far with 720 Kbps per slice. In addition, optimization techniques for lightweight implementations have also been introduced in this paper.

# CHAPTER 3

# PROBLEM FORMULATION AND OBJECTIVES

From the literature survey, it has been found that RFID tags are more prone to eavesdropping and replay attacks and security is a big concern [9, 10]. To overcome both the attacks, the symmetric algorithms are worked in CBC mode [42]. In the CBC (Cipher Block Chaining) mode, a random initialization vector (IV) is added which provide One-Time-Padding (OTP) in the encryption process and provide unique cipher for each block and authentication. In the symmetric cipher AES algorithm is best suited for the security purposes and has been recommended by recommended by NIST. The author in [18] has used AES algorithm for RFID tags and reported 3400 gates are used for implementation purposes. Next, to provide encryption and authentication, authors in [20] implemented AES with hash algorithm and their results showed that it required 9800 gate equivalent (GE) for AES-Hash**.** Authors in [33] has proposed to use AES in CBC mode for encryption/decryption to provide more security. Also Authors in [28] has used asymmetric algorithm ECC for authentication in RFID tags and results showed that it required 16,900 GE for its implementation. These primitive algorithms are not suitable for low-cost RFID tags, as an RFID tags has maximum of 2000 GEs available for security as stated in [11, 12]. So, there emerges a need for lightweight cryptographic algorithms to be implemented in these low-cost RFID tags.

The various lightweight ciphers such as PRESENT [19], HIGHT [22], DESL [21] have been deployed for providing cryptographic solutions to RFID tags. From comparison of various lightweight in Table 2.1 ciphers it has been observed that till now PRESENT is the best suited algorithm in for lightweight applications and is also recommended by NIST [12]. The security of lightweight cipher depends on the avalanche effect and the number of rounds. The PRESENT cipher has 49.2% avalanche effect and 32 rounds. Authors **Gaurav Bansod*, et al***. [26] have proposed a new lightweight algorithm BORON having higher avalanche effect higher throughput, lesser latency, higher avalanche effect than PRESENT cipher. In this work BORON cipher has been selected as an encryption scheme. Further, for CBC mode to generate an initialization vector, authors in [34] generated random number using Pseudo Random Number Generator(PRNG). In the literature, pseudorandom number is generated using LFSR (Linear Feedback Shift Register) and NFSR (Non-Linear Feedback Shift Register). The LFSR and NFSR registers required initial seed point to generate random number and storage required and overall security breaks if someone determines seed point. Hence, an arbiter PUF comes in picture which provides hardware security by generating a random response on the fly [36]. Keeping in mind the advantages of an arbiter PUF, it has been used to generate an IV in the proposed technique.

In the proposed technique, to provide encryption and authentication, BORON cipher has been made to work in CBC mode. The performance analysis of the implemented design has been done in terms of FPGA slices, latency, throughput, and gate equivalents.

**Methodology**

Figure 3.1 gives an overview to the methodology that has been followed throughout the course of entire work. The work started with analysing the need of security in RFID tags. The cipher that required maximum for 2000 GE for its implementation had to be chosen among all the existing ciphers. The software and hardware implementation of the chosen cipher has been done and the results were tested using the test vector provided in [26]. Further, the initialization vector has been generated using Arbiter PUF and its hardware implementation is done to make BORON work in the CBC mode.

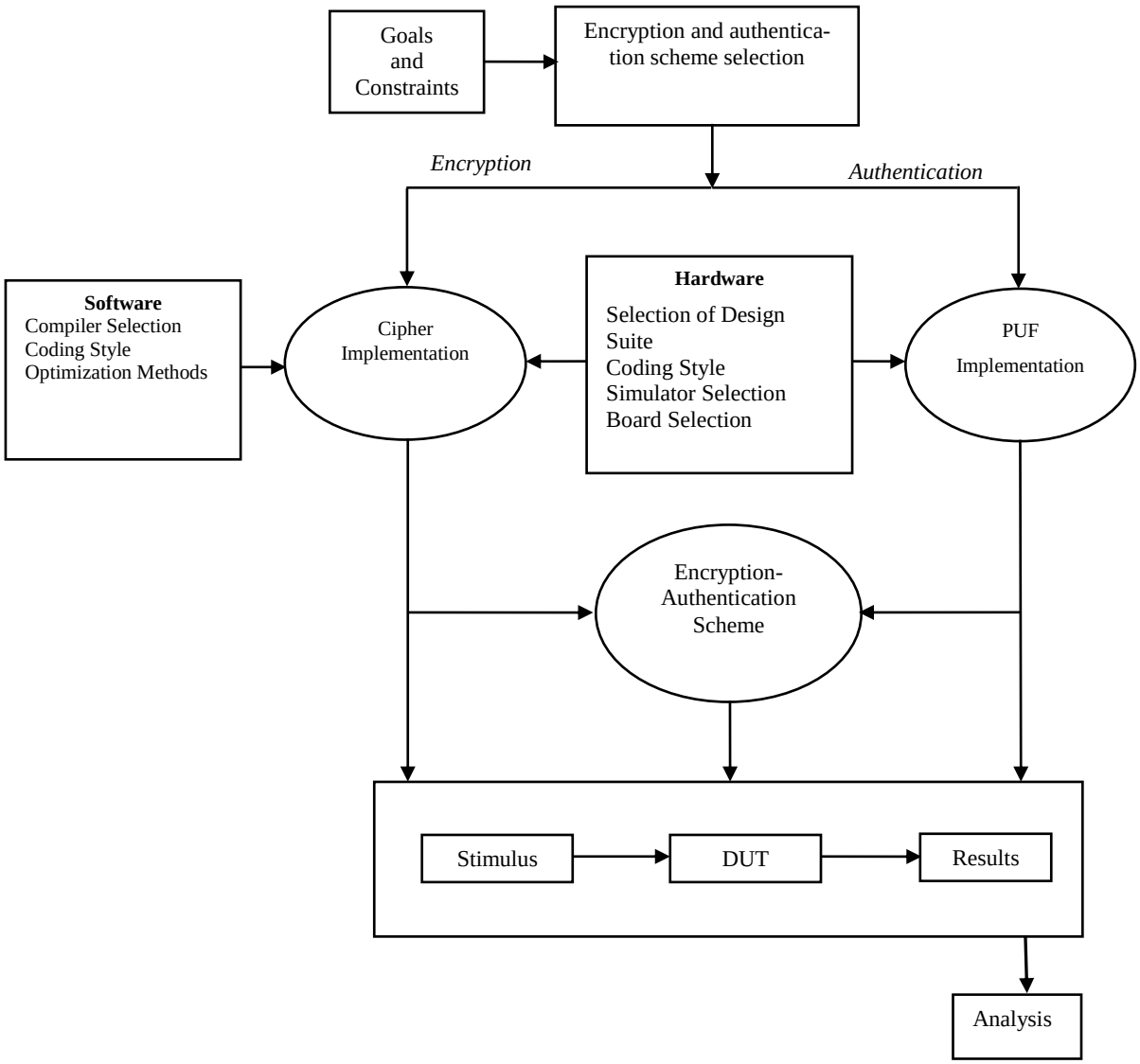



Figure 3.1 Overview of Methodology

# CHAPTER 4

# ENCRYPTION-AUTHENTICATION SCHEME

This chapter briefly describes the encryption-authentication scheme that has been worked upon to provide security in RFID tags. Figure 4.1 gives the block diagram of the adopted encryption-authentication scheme. In the scheme, the IV has been generated using arbiter PUF on the fly and has been input to BORON cipher in the first round of encryption. Section 4.1 and 4.2 give the brief overview of BORON and an Arbiter PUF.

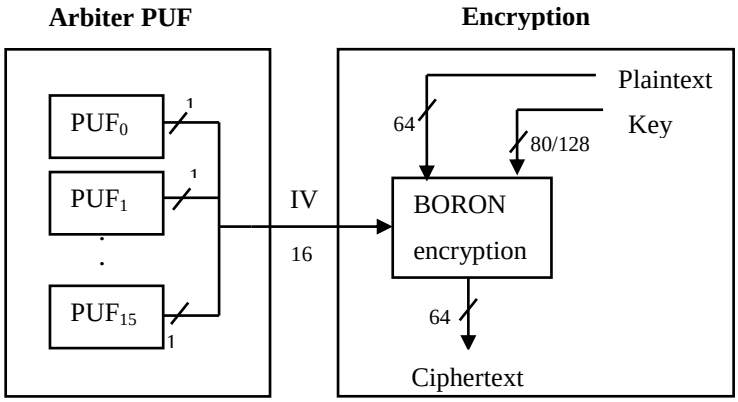



Figure 4.1 Block Diagram of Encryption-Authentication System

## 4.1 BORON Cipher [26]

BORON is a symmetric block cipher proposed by Gaurav Bansod, *et al.* [26]. BORON is an SP ( substitution permutation) network. The block length is 64 bits and it supports two key variants one with 80-Bit key referred as BORON_80 and other with 128 bits key referred as BORON_128. It consists of total of 24 rounds and last $25^{th}$ round for key mixing. The BORON Cipher consists of an S-Box which acts a non linear layer. The S-box layer is followed by a block shuffle of four bits. The shuffled bits are then passed to a round permutation layer followed by an XOR Operation. Twenty five keys(80bit/128 bit) are generated from(80bit/128bit) key register and these distinct keys are applied in each round of BORON cipher. One extra key is generated and XORed to produce final ciphertext. One round of BORON cipher undergoes following series of operations

- Add round_key
- S Box Layer
- P layer
- Key_scheduling

BORN has an optimized architecture, which requires less area and memory as compared to AES, DES etc. making it suitable for resource constrained environments such as RFID tags, Smart Cards, Wire-

less Sensors, IOTs. Till now only round based architecture is available in literature which takes 25 clock cycles for encrypting one block of data and consumed a total of 1939 gate equivalents for BORON_128 and 1626 for BORON_80.

### 4.1.1 Algorithmic Description of Roundbased Architecture

Figure 4.2 provides the top level algorithmic description of BORON block cipher.

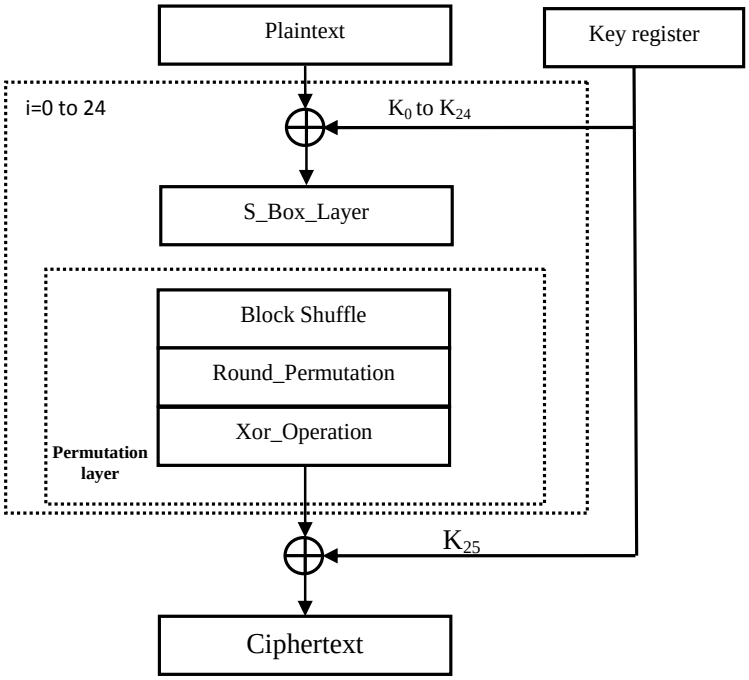



Figure 4.2 Basic Block Diagram of BORON Block Cipher

- **Add _Round_ Key**: Add round key performs an XOR operation with 64-bit plain text and 64-bit key extracted from 80/128bit key register. If $K_i$ denotes the subkeys with i ranging from 0 to 24 and A represent current state output $A \rightarrow a^{63}\, a^{62} \ldots\ldots\ldots\ldots\ldots\ldots a^{60}$,A is given as

$$A \rightarrow A \text{ XOR } K^i_j \quad (j=63 \text{ to } 0) \qquad (4.1)$$

- **S_BOX_Layer** A 4 bit to 4 bit bijectively mapped S_BOX is used in BORON Cipher design, S: $F_4^2 \rightarrow F_4^2$.The current state output($a^{63}\, a^{62} \ldots\ldots\ldots\ldots\ldots\ldots a^{60}$ ) from add round key() is divided into 16 4- bit blocks ($B^{15}\, B^{14} \ldots\ldots\ldots\ldots\ldots B^0$). Each of the $B^i$ is of 4 bit size given as

$$B^i = a^{(4*i+3)} \| a^{(4*i+2)} \| a^{(4*i+1)} \| a^{(4*i)}, \text{ with i ranging from 0 to 15.} \qquad (4.2)$$

Each $B^i$ is fed to 4-bit SBOX and is replaced form a corresponding 4-bit fixed value provided in the S-BOX Table 4.1.

**Table 4.1 BORON S_BOX LAYER** [26]

| $B^i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | b | C | d | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $s[B^i]$ | e | 4 | B | 1 | 7 | 9 | c | A | d | 2 | 0 | f | 8 | 5 | 3 | 6 |

- **Permutation Layer:** The permutation layer (P layer) of BORON cipher consists of three sub permutation layers namely
    i) Block_Shuffle
    ii) Round_permutation
    iii) Xor_Operation

**i. Block_Shuffle :** A 64-bit output from S_Box_Layer is divided into four blocks each 16-bit wide. The block shuffle layer takes one block of 16 bits as an input and gives shuffled 16 bits output. The block permutation layer divides each 16 bit input into 4 bits as $P=P_4^3 || P_4^2 || P_4^1 || P_4^0$. Block permutation is performed according to Table 4.2.

**Table 4.2 Block_Shuffle Layer of BORON Cipher** [26]

| **X** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P[X]** | 2 | 3 | 0 | 1 | 6 | 7 | 4 | 5 | 10 | 11 | 8 | 9 | 14 | 15 | 12 | 13 |

**ii. Round_Permutation:** A 64-bit output from block shuffle layer is divided into four blocks of 16-bit each. The round permutation layer performs left circular shift on each of the 16-bit block as per the value given in Table 4.3.

**Table 4.3 Round Permutation: Left Circular Shift Value** [26]

| **Block** | **0** | **1** | **2** | **3** |
|---|---|---|---|---|
| **Left Rotation by** | 1 bit | 4 bits | 7 bits | 9 bits. |

Figure 4.3 gives the complete insight for round permutation followed by block shuffling.

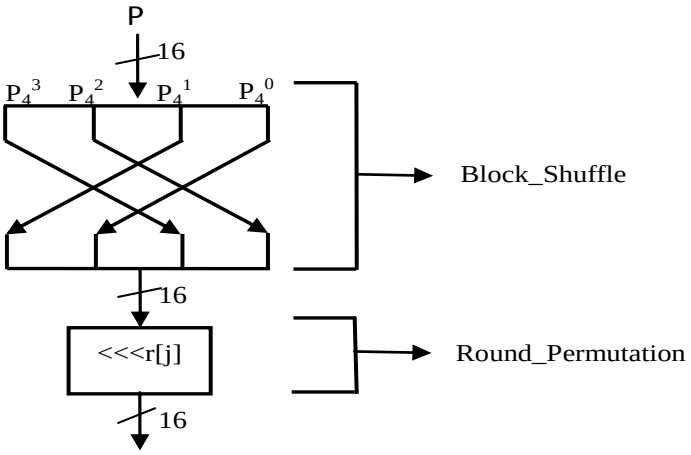



Figure 4.3 Block Shuffling followed by Round Permutation`

### iii. XOR_Operation

XOR_operation layer performs XOR operation on 16-bit inputs and produces 16-bit output. The 16-bit output is produced in following manner.

$$A_{64} \rightarrow (W^3 \oplus W^2 \oplus W^0) || (W^2 \oplus W^0) || \left( W^3 \oplus W^1 \right) || (W^3 \oplus W^1 \oplus W^0) \qquad (4.3)$$

Figure 4.4 provides the detailed description of each regular round in encryption.

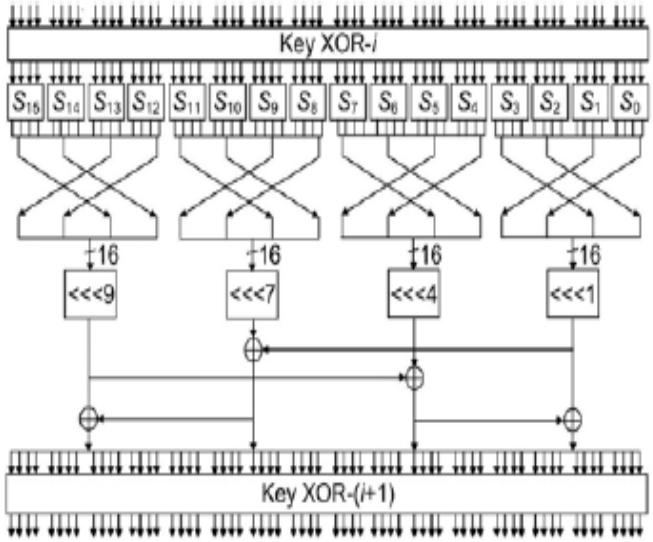



Figure 4.4 Series of Operations Performed in Each Round of Encryption [26]

### Key Scheduling

Key scheduling for BORON has been motivated form the key scheduling of PRESENT algorithm as presented in [19]. The key scheduling of BORON cipher consists of a total of 25 rounds and hence a total of 25 sub-keys.

***Key Scheduling for 128-bit keys***

The 128-bit key is stored in a KEY register and 64 Least Significant bits of the KEY register is extracted in following way

$$KEY=K_{127}K_{126}.....K_0 \quad (4.4)$$

$$K^i = K_{63}K_{62}.....K_0 \quad (4.5)$$

Once the 64-bits of key are extracted, KEY register is updated in following manner

1) $KEY[K_{127}K_{126}.....K_0] = KEY[K_{127}K_{126}.....K_0] <<<13$ (left rotation by 13 bits) (4.6)

2) $KEY[K_3\ K_2\ K_1\ K_0] = S\_BOX[K_3\ K_2\ K_1\ K_0]$ (4.7)

3) $KEY[K_7\ K_6\ K_5\ K_4] = S\_BOX[K_7\ K_6\ K_5\ K_4]$ (4.8)

4) $KEY[K_{63}\ K_{62}\ K_{61}\ K_{60}\ K_{59}] = KEY[K_{63}\ K_{62}\ K_{61}\ K_{60}\ K_{59}] \oplus$ roundcounter. (4.9)

Roundcounter is incremented after each round.

***Key Scheduling for 80-bit keys***

1) $KEY[K_{79}K_{78}.....K_0] = KEY[K_{79}K_{78}.....K_0] <<<13$ (4.10)

2) $KEY[K_3\ K_2\ K_1\ K_0] = S\_BOX[K_3\ K_2\ K_1\ K_0]$ (4.11)

4) $KEY[K_{63}\ K_{62}\ K_{61}\ K_{60}\ K_{59}] = KEY[K_{63}\ K_{62}\ K_{61}\ K_{60}\ K_{59}] \oplus$ roundcounter. (4.12)

### 4.1.2 S-Box implementation of BORON

There are two approaches for the implementation of the S-BOX i.e. LUT based approach and Combinational logic based approach.

**i. LUT Based Approach:** This approach uses a look up table for substitution. Case statement is applied to implement this approach. Each round uses 16 look up tables in parallel to substitute 4 bit values. Hardware implementation of this approach occurs through registers to store values.

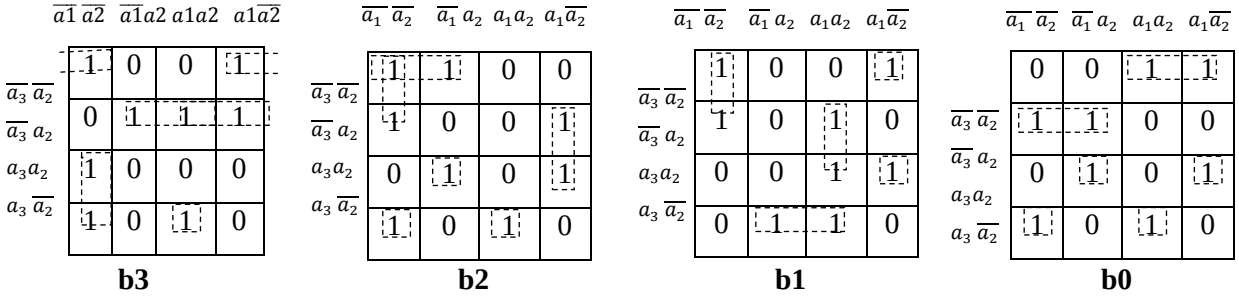
**ii. Combinational Logic Based Approach:** This new approach implements S-Box in the form of Boolean function. Each bit of the data is represented by a different Boolean function. Whole S-Box is derived from 4 SOP expressions derived as following.

Let $A=a_3a_2a_1a_0$ be the 4-bit input to slayer and $B=b_3b_2b_1b_0$ be the four bit output from slayer. Table 4.4 gives the truth table for deriving the Boolean expression for each output bit.

Table 4.4 Truth Table for 4-bit to 4-bit S-box

| $a_3$ | $a_2$ | $a_1$ | $a_0$ | $b_3$ | $b_2$ | $b_1$ | $b_0$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |




On solving Karnaugh map for b3, b2, b1, and b0, following minimum boolean expressions are obtained as given in equation 4.13, 4.14, 4.15 and 4.16.

$$b_0 = \overline{a_3}\ \overline{a_2}\ a_1 + \overline{a_2}\,a_1a_0 + \overline{a_3}\,a_2\overline{a_1} + a_2\overline{a_1}\ a_0 + a_3\ \overline{a_2}\ \overline{a_1}\ \overline{a_0} + a_3\ a_2a_1\ \overline{a_0} \quad (4.13)$$

$$b_1 = \overline{a_3}\ \overline{a_2}\ \overline{a_0} + \overline{a_3}\ \overline{a_1}\ \overline{a_0} + a_2a_1a_0 + a_3\overline{a_2}\ a_0 + a_3\ a_2a_1 \quad (4.14)$$

$$b_2 = \overline{a_3}\ \overline{a_2}\ \overline{a_1} + \overline{a_2}\ \overline{a_1}\ \overline{a_0} + \overline{a_3}a_2\overline{a_0} + a_3a_1a_0 + a_3a_2a_0 \quad (4.15)$$

$$b_3 = \overline{a_3}\ \overline{a_2}\ \overline{a_0} + \overline{a_3}\ a_1\ \overline{a_0} + \overline{a_3}a_2a_0 + a_3\overline{a_1}\ \overline{a_0} + a_3\overline{a_2}\ a_1a_0 \quad (4.16)$$

Hardwar implementation of these Boolean expressions requires 47 gates. In this work, an S-Box has been implemented using both combinational based approach as well as LUT based approach and results have been compared for both the approaches in terms of FPGA slices.

### 4.1.3 64-bit Datapath Architecture for BORON Cipher

The 64-bit datapath of round based architecture of BORON_128 and BORON_80 is depicted in Figure 4.5 and Figure 4.6. The Round based architecture consumes 1 clock cycle for each round. The Round based architecture uses 16-sboxes in parallel and P_layer which leads to data path of one 64 bit XOR, one P-layer, and 16 parallel S-boxes. Key scheduling consists of 64 bit state register to store internal state and 80/128 bit key register to store updated key. Key scheduling components includes 80/128 bit key register, 5 bit XOR, one/two S-box(s), 13 bit shifter and key is generated on the fly for each round.

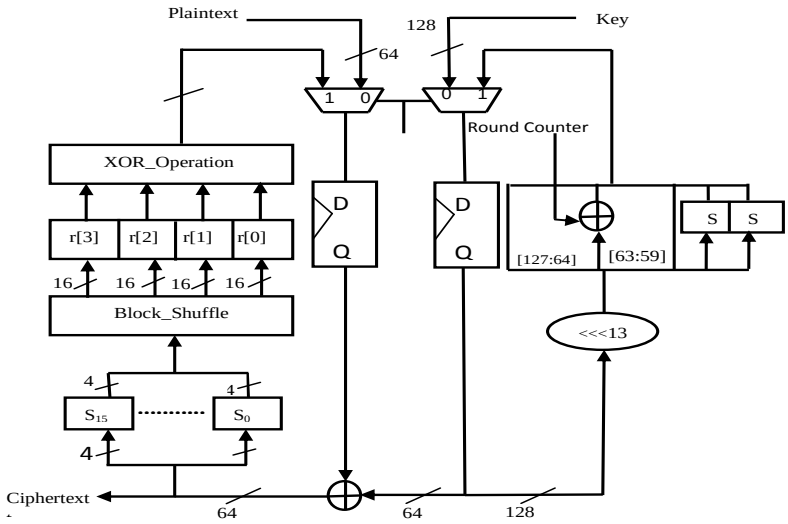



Figure 4.5 Round Based Architecture of BORON for 64-bit datapath and 128-bit key

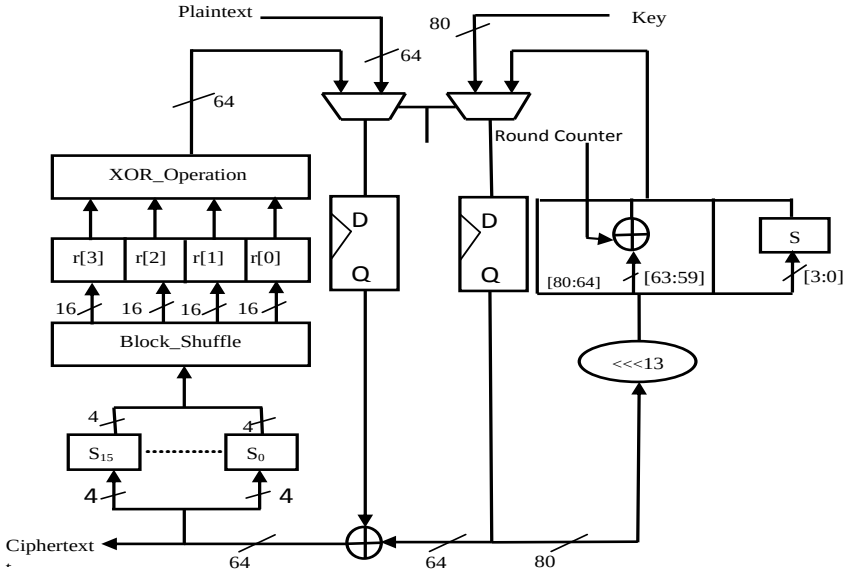



Figure 4.6 Round Based Architecture of BORON for 64-bit datapath and 80-bit key

### 4.1.4 16-bit Datapath Architecture for BORON

One of the most area consuming part of BORON are the 16 S-boxes that are used in parallel for substitution layer in each round of encryption. To reduce the chip area, the number of s-boxes are reduced

from 16 to 4 per round i.e. now the datapath structure has been reduced to 16-bit.This area saving came at the cost of longer computation time. Only 16-bits are processed per clock cycle. So, it requires a total of 9 clock cycles (4 clock cycles for plaintext and 5 cycles for key)for initialisation for BORON_80.For BORON_128 it requires 12 clock cycles (4 for plaintext and 8 for key)for initialisation. Since now only 4 S-boxes are used in parallel, and hence one regular round of encryption requires 4 clock cycles to complete. The overall latency for BORON_80 is 109 (9+25*4) clock cycles and for BORON_128 is ( 112 clock cycles). An additional 4 clock cycles would be required to write the data back to the memory. Figure 4.7 gives the 16-bit datapath for BORON cipher.

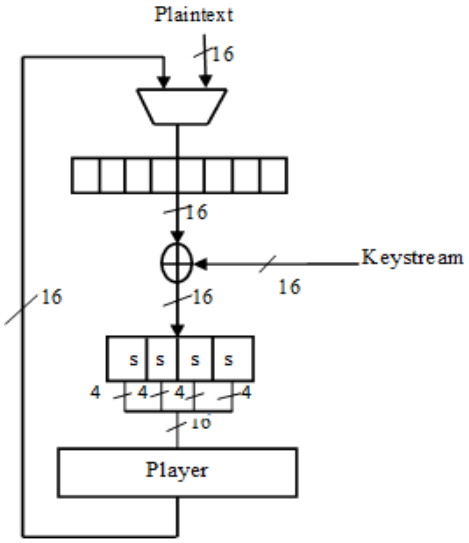



Figure 4.7 The 16-bit Datapath for BORON Cipher

## 4.2 ARBITER PHYSICAL UNCLONABLE FUNCTION (PUF)

A PUF is a function that maps a set of challenges to a set of responses based on complex physical system. The function can only be evaluated with the physical system, and is unique for each physical instance. Another advantage of PUFs is that they do not require any special manufacturing process or programming and testing steps. A PUF delay circuit using Multiplexers and an arbiter to generate an initialisation vector (nonance) is shown in Figure 4.8. The circuit has a multiple-bit input X and computes a 1-bit output Y based on the relative delay difference between two paths with the same chain length. The input bits determine the delay paths by controlling the multiplexers. Here, a pair of Multiplexers is controlled by the same input bit work as a switching box. The multiplexers pass through the two delay signals from the left side if the input control bit X[i] is zero. Otherwise, the top and bottom signals are switched. In this way, the circuit can create a pair of delay paths for each input X. To evaluate the output for a particular input, a rising signal is given to both paths at the same time, the signals race through the two delay paths, and the arbiter latch at the end decides which signal is faster. The output is one if the signal to the latch data input (D) is faster, otherwise output is 0. The single PUF chain is duplicated k times to produce k bits with single evaluation [35, 36].

In this work, the one bit PUF as shown in Figure 4.8 is duplicated 16 times to produce a 16-bit Initialisation Vector(IV) or nonce (secret number that is used only once). This initialisation vector is fed to the encryption algorithm in the very first round of encryption.

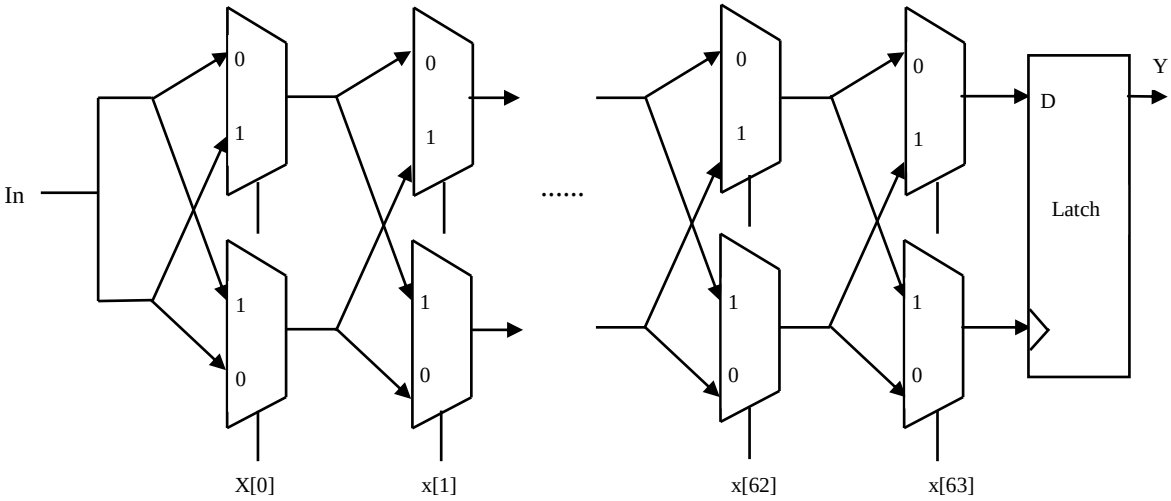



Figure 4.8 An arbiter PUF delay Circuit [35]

### 4.2.1 Advantages of PUF Over Conventional Authentication Scheme [36]

Since PUFs tap into the random variation that occurs during an the IC fabrication process, the secrets are intrinsic to the silicon itself, are extremely difficult to predict in advance to manufacturing, and are usually non-replicable from chip to chip. In this way a PUF technology provides a several advantages over the conventional approach of storing digital secrets for customizing each IC.

These PUFs increase physical security by generating temporary secrets that exist in a digital form only when a chip (FPGA board for this work) is powered up and running. This means that an adversary, instead of examining an IC's/Board's memory to read its stored secret would need to mount an attack while the chip/Board is running and using the secret exist in a digital form when a chip is powered on and running. This means that instead of merely examining an IC's memory to read the stored secret, an attack would have to mount an attack while the chip/board is running and using the secret for his malicious purpose. This is a harder proposition rather that reading a permanent memory.

# CHAPTER 5

# IMPLEMENTATION RESULTS

## 5.1 SOFTWARE IMPLEMENTATION

A C code has been implemented on GCC on LEON-3 SPARC V8 32-bit processor. Algorithm 1 provides the pseudo algorithm for BORON Cipher. For implementation of s_layer a loop unrolling and minimum variables has been used to increase the execution speed.

**Algorithm 1 Algorithmic overview of BORON Cipher**

**Input:** Plain text $M_{64} \rightarrow m_{63}\ m_{62} \ldots m_0$, S[16], P[4], r[4]
**Output:** Cipher text $C_{64}$

**for** i=0 to 24 **do**
$M_{64} \rightarrow m_{63}\ m_{62} \ldots m_0$

**for** j=0 to 3 **do**
$temp_{16} \rightarrow (M_{64} >>> 16.j) \oplus (K^i \gg 16.j)$
$temp_{16} \rightarrow S[temp_{16}]$
$B_{16}^{[j]} = 0$

**for** k=0 to 4 do
$W_{16}^{[j]} += ((temp_{16} \gg 4k) \& 0xF \ll 4\ P[k]$ *//Block_shuffle*
**end for**

$W_{16}^{[j]} = W_{16}^{[j]} <<< R[j]$ *//Round_permutation*
**end for**

$A_{64} \rightarrow X(W_{16}^3 . W_{16}^2 . W_{16}^0) \,||X(W_{16}^2 . W_{16}^0)\| X(W_{16}^3 . W_{16}^1) \,||\, X(W_{16}^3 . W_{16}^1 . W_{16}^0)$
*//X:XOR operation*
**end for**

**for** j=0 to 3 **do**
$C_{64} += (M_{64} \gg 16.j) \oplus (K_{64}^{25} \gg 16.j) \ll 16.j$
**end for**

**Result** : The experimental results showed that the BORON cipher required 22,558 clock cycles/block at 50MHz with a code size of 1638 bytes and gives a throughput for 141.856 Kbps.

## 5.2 HARDWARE IMPLEMENTATION

Synthesis has been done in XILINX VIVADO Design Suite 2016 using Verilog HDL. Synthesizable code has been generated for Digilent XC7A35T-1CPG236C BASYS-3 Board of Artix-7 family. Provided board has 20800 LUTs, 41600 Slice Registers, 8150 slices, 106 bonded IOBs, an internal clock of exceeding 400MHz, on chip analog to digital converter (XADC). 16 input switches, 16 out-

put LEDs, W5 clock pin, 5 user push buttons, three PMOD ports, 12 bit VGA output, USB-UART bridge.

The following metrics have been chosen for analysis of implemented design [43]

- ***Area***: Area has been reported in terms of total number of FPGA slices required for the design. The Gate Equivalents have been calculated on DC compiler using SCL 180nm by dividing the total area in($\mu m^2$) by the area of one NAND gate ($\mu m^2$).
- ***Maximum Frequency*:** There are various connections form input to output in a design. The slowest path in the design will set an upper bound on the clock frequency.
- ***Clock Cycles*:** Number of clock cycles required to compute the ciphertext.
- ***Throughput*:** Throughput is defined as the rate at which the ciphertext is produced with respect to time. Throughput is calculated as the total number of output bits divided by the clock cycles needed to produce the output and multiplied by the maximum frequency.

$$\text{Throughput} = \frac{\text{Block Size x Maximum Frequency}}{\text{Latency}} \quad (5.1)$$

For determining the security strength of the cipher, avalanche effect has been analysed.

- ***Avalanche Effect*:** Avalanche effect describes the change in the cipher text with a small variation either in plaintext bits or key bits. For encryption algorithms to be secured it is desirable that even a single bit change in the plaintext or key stream bits should vary as many bits in the output data**.**

### 5.2.1 Implementation Results for 64-bit round based architecture for BORON

A 64-bit roundbased architecture has been implemented for BORON block cipher for key size of 80-bit and 128-bit.S-BOX has been implemented using both the LUT based approach and combinational based approach. The key calculation has been done on the fly and hence no extra arrays were used to store the round keys. The simulation results has been checked using the test vectors given in [26].

- **Implementation Results for BORON_80**

Figure 5.1 gives the simulation waveform for BORON_80. The test vectors have been provided in Table 5.1. The synthesized results have been provided in Table 5.2.

Table 5.1 Test Vectors for BORON_80

| | |
|---|---|
| ***Plaintext*** | 0x0000_0000_0000_0000 |
| ***Keyvalue*** | 0x0000_0000_0000_0000_0000 |
| ***Ciphertext*** | 0x3cf7_ 2a8b_ 7518_ e6f7 |

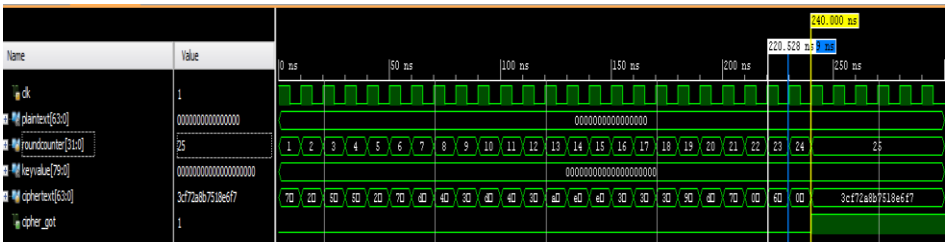



Figure 5.1 Simulation Waveform for 64-bit Roundbased Architecture for BORON_80

Table 5.2 Synthesis Results for 64-bit Roundbased Architecture for BORON_80

| **Parameter** | **Resources Available** | **SBOX (LUT)** | **SBOX(Boolean)** |
|---|---|---|---|
| LUTs | 20800 | 256 | 248 |
| Slice Registers | 41600 | 188 | 188 |
| FPGA Slices | 8150 | 91 | 87 |
| Clock Cycles | ---- | 25 | 25 |
| Maximum Frequency (MHz) | ---- | 177 | 167.28 |
| Throughput(Mbps) | ---- | 453.12 | 428.23 |
| Avalanche Effect | ---- | 50.78% | 50.78% |

Avalanche effect has been calculated using following plaintext and key pairs as shown in Table 5.3.

Table 5.3 Avalanche Effect Calculation for BORON_80

| **Plaintext** | **Key** | **Ciphertext** | **No. of bits changed** |
|---|---|---|---|
| | 0x0000_0000_0000_0000_0000 | 0x3cf7_ 2a8b_ 7518_ e6f7 | -- |
| 0x0000_0000_0000_0000 | 0x0100_0000_0000_0000_0000 | 0xfdf9_f345_3448_197a | 32 |
| | 0x0000_0000_0000_0000_0010 | 0xfe91_0aec_bee3_29b3 | 33 |

- **Implementation Results for BORON_128**

Figure 5.2 gives the simulation waveform for BORON_80. The test vectors have been provided in Table 5.4. The synthesized results for BORON_128 have been provided in Table 5.5.

Table 5.4 Test Vectors for BORON_128

| | |
|---|---|
| ***Plaintext*** | 0x0000_0000_0000_0000 |
| ***Keyvalue*** | 0x0000_0000_0000_0000_0000_0000_0000_0000 |
| ***Ciphertext*** | 0x3cf7_ 2a8b_ 7518_ e6f7 |

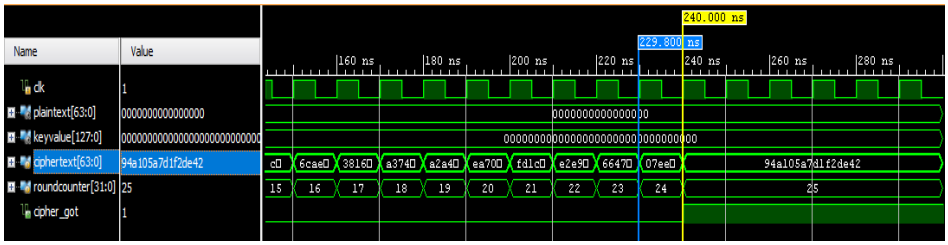



Figure 5.2 Simulation Waveform for 64-bit Roundbased Architecture for BORON_128

Table 5.5 Synthesis Results for 64-bit Roundbased Architecture for BORON_128

| Parameter | Resources Available | SBOX (LUT) | SBOX(Boolean) |
|---|---|---|---|
| LUTs | 20800 | 265 | 261 |
| Slice Registers | 41600 | 206 | 206 |
| FPGA Slices | 8150 | 98 | 93 |
| Clock Cycles | ---- | 25 | 25 |
| Maximum Frequency(MHz) | ---- | 166 | 149 |
| Throughput(Mbps) | ---- | 425 | 381.4 |
| Avalanche Effect | ---- | 54.68% | 54.68% |

Avalanche Effect has been calculated using following plaintext-key pairs in Table 5.6.

Table 5.6 Avalanche Effect Calculation for BORON_128

| Plaintext | Key | Ciphertext | No. of bits changed |
|---|---|---|---|
| 0x0000_0000_0000_0000 | 0x0000_0000_0000_0000_0000_0000_0000_0000 | 0x94a1_ 05a7_ d2f2_ de42 | -- |
| | 0x0000_0000_0000_0000_0000_0000_0000_0010 | 0x7946_ b520_ 9d6e_ c210 | 34 |
| | 0x0000_8000_0000_0000_0000_0000_0000_0000 | 0x2dcc_ 3b8d_ e115_ e67c | 36 |

From the synthesis results in Table 5.2 and 5.5 it has been observed that S-box approach implementation Boolean logic consumed 4.74% lesser FPGA slices than LUT based approach.

### 5.2.2 Hardware Implementation of an Arbiter PUF

To generate a 16-bit secret Initialisation Vector (IV), 16 1-bit PUF modules have been instantiated. Figure 5.3 and Figure 5.4 gives the RTL schematic for 16 bit chain of PUF and a single bit arbiter PUF. Figure 5.5 and Figure 5.6 gives the post implementation timing simulation waveform for an arbiter PUF with different challenge and response pair.

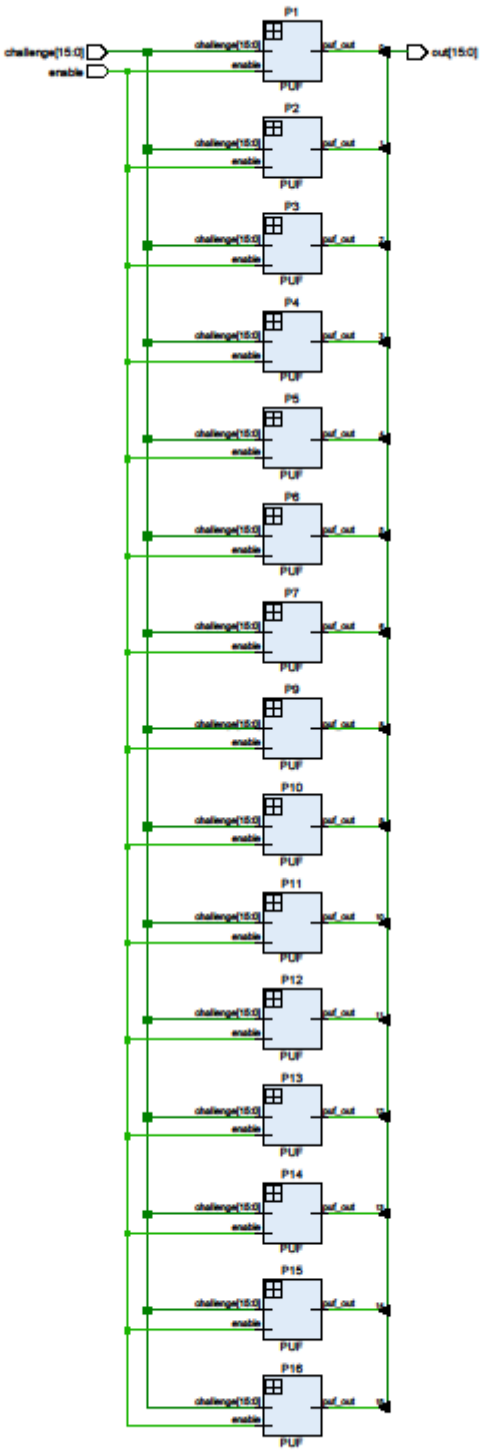



Figure 5.3 RTL Schematic for 16 1-bit PUF chains

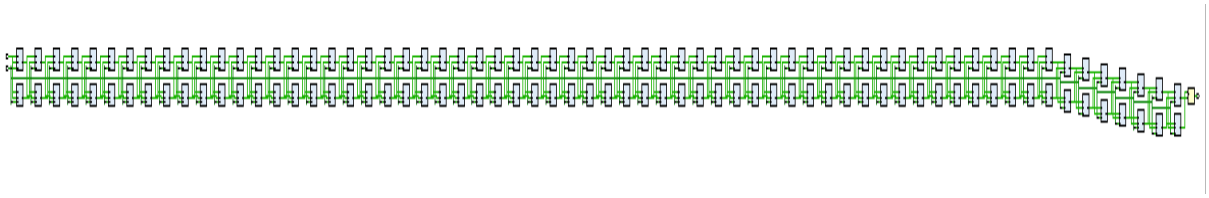


Figure 5.4 RTL Schematic for 1-bit Arbiter PUF

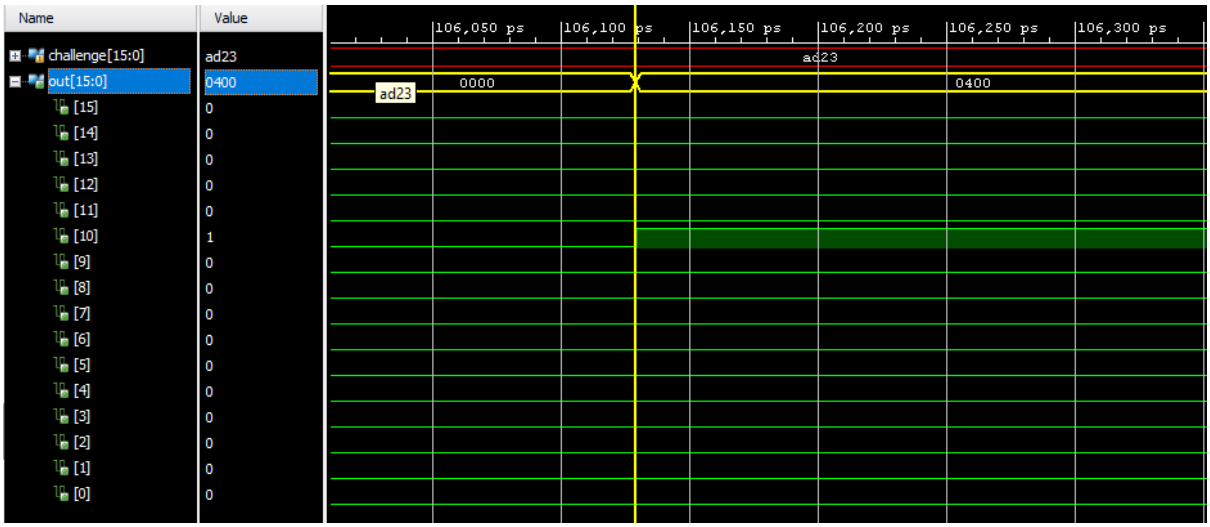



Figure 5.5 PUF Response for Challenge 1

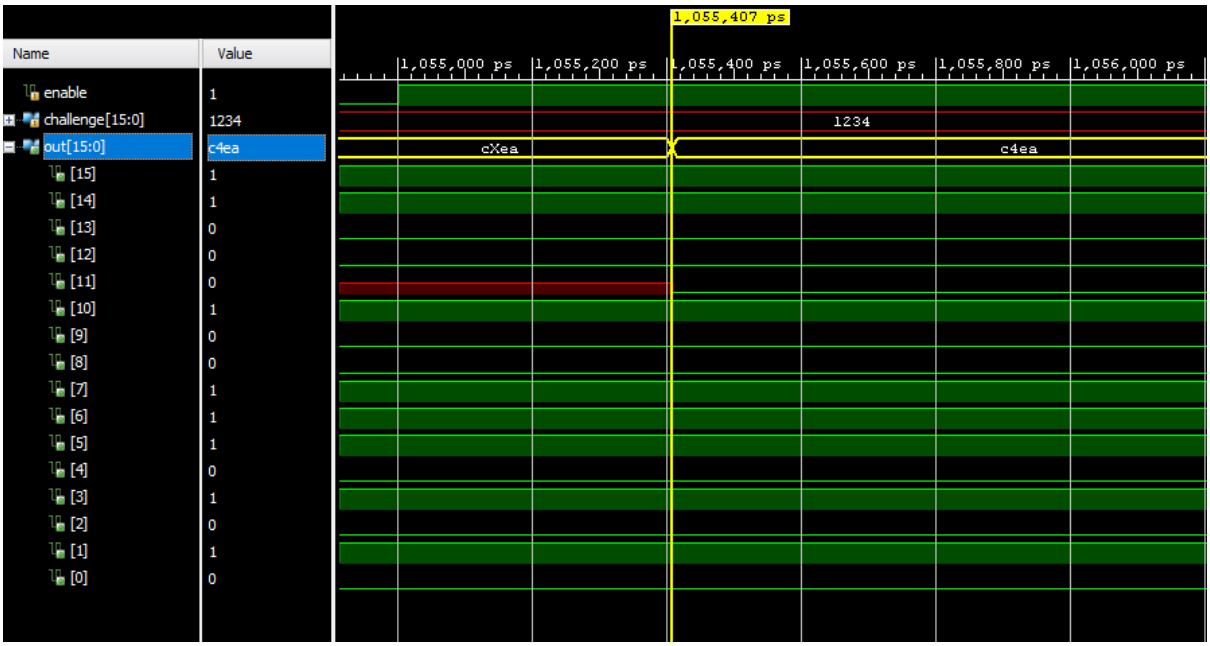



Figure 5.6 PUF Response for Challenge 2

On synthesising the design for Digilent XC7A35T-1CPG236C BASYS-3 board it has been observed that implementation of 1-bit PUF required 32 FPGA slices and 16-bit chain of PUF required 513 FPGA slices. All the design optimizations were turned off during the implementation. From the post

implementation timing simulation results it has been observed that based on internal delay, a different response was produced for a different challenges.

### 5.2.3 Implementation of 64-bit authentication encryption scheme

The designs of encryption unit (BORON_128) and an arbiter PUF were integrated to provide authentication by generating an Initialization vector (IV). The secret IV produced form an internal delay of PUF chains was XORred with the plaintext and key in the very first round of encryption. Figure 5.7 gives the implemented design for combined authentication and encryption scheme. Yellow coloured area shows the PUF circuit embedded in the main design.

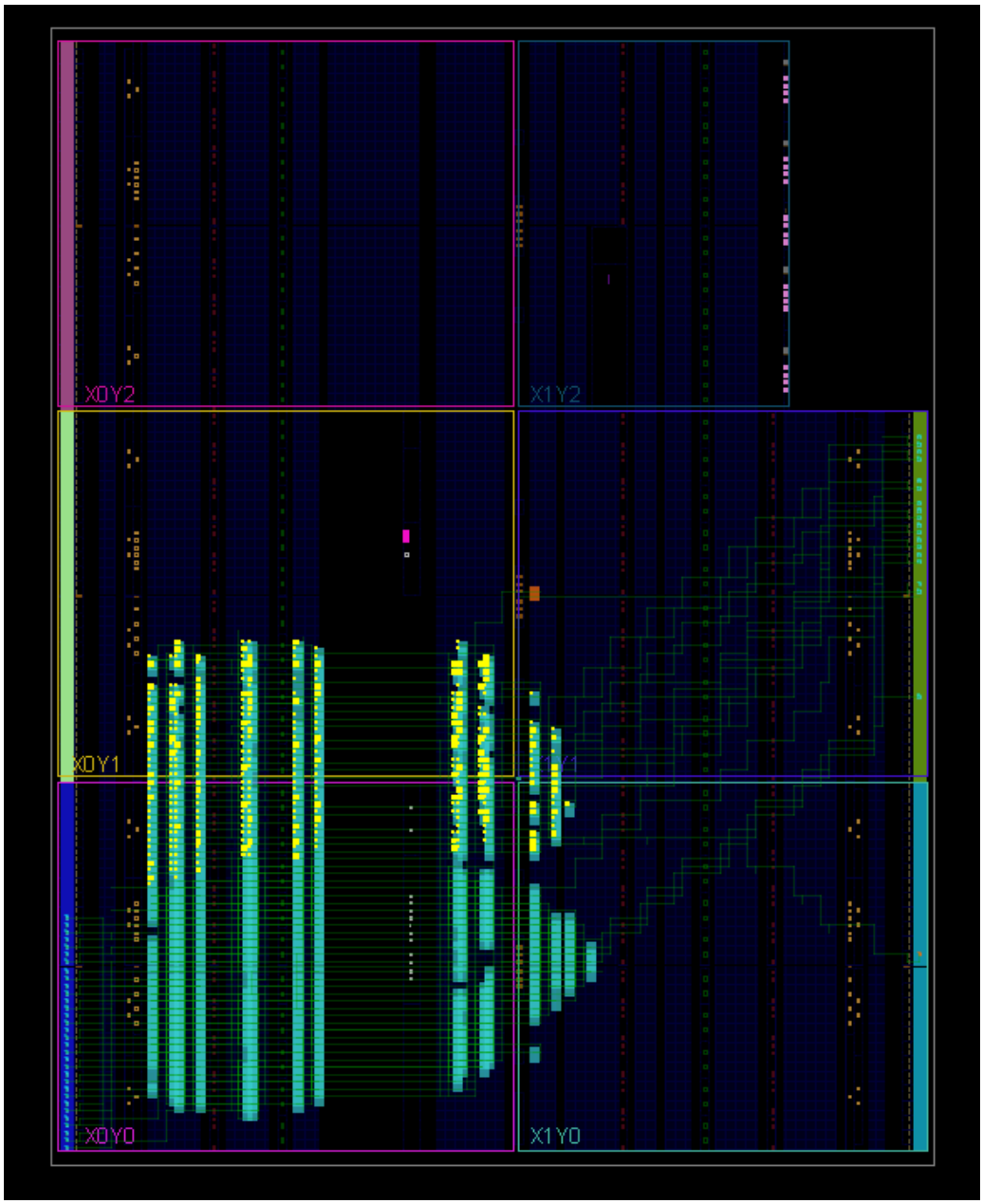



Figure 5.7 Implemented Design for 16-bit Authentication-Encryption Scheme

Figure 5.8 provides the post implementation timing simulation for the 64-bit encryption-authentication with the challenge 1.

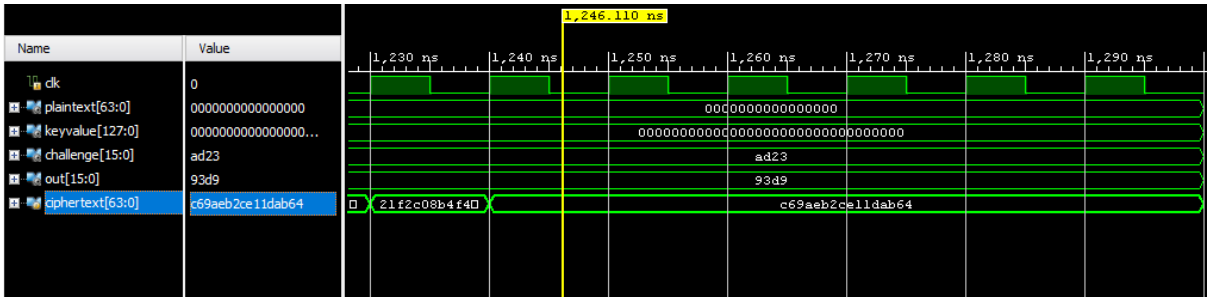



Figure 5.8 Simulation Waveform for 64-bit Encryption-Authentication Scheme with Challenge 1.

On synthesizing the design for the specified FPGA board, it has been observed that it required 417 LUTs, 210 slice registers and 113 FPGA slice for its implementation. The post implementation timing simulation results showed that again the different response was produced for the same challenges. On further synthesizing the design on Synopsys DC compiler with SCL-180nm technology, it has been observed that it required approximately 4321 GE for its implementation which is larger than the area available for security purposes low cost devices (2000 GEs). Therefore, a 16-bit implementation for BORON cipher has been implemented in section 5.2.4.

### 5.2.4 16-bit lightweight implementation results for BORON Cipher

For a reduced area implementation for RFID tags, a16-bit roundbased architecture has been implemented for BORON block cipher for key size of 80-bit and 128-bit.The 16-bit architecture requires 109 /113 clock cycles for encryption with BORON_80 and BORON_128. Figure 5.9 and 5.10 gives the simulation waveform for 16-bit datapath roundbased architecture for BORON_80 and BORON_128. Synthesis results for BORON_80 and BORON_128 have been provided in Table 5.7 and 5.8. The results have been tested against the same test vectors as in Table 5.1 and Table 5.4.

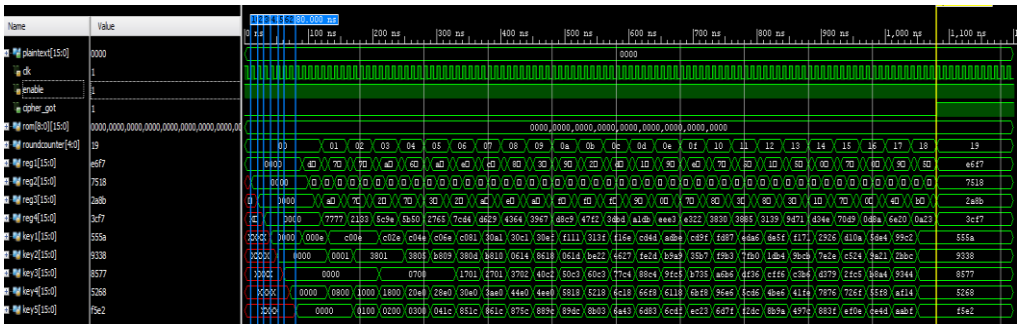


Figure 5.9 Simulation Waveform for 16-bit Datapath Roundbased Architecture for BORON_80

The upper 2 bytes of Ciphertext have been stored in reg4 and the lower two bytes have been stored in reg1.

**Ciphertext:** 0x 3CF7 2a8b 7518 e6f7

**Registers :** reg1 reg2 reg3 reg4

Table 5.7 Synthesis Results for 16-bit Roundbased Architecture for BORON_80

| Parameter | Resources Available | S-BOX (LUT) | SBOX (Boolean) |
|---|---|---|---|
| LUTs | 20800 | 202 | 198 |
| Slice Registers | 41600 | 162 | 162 |
| FPGA Slices | 8150 | 61 | 60 |
| Clock Cycles | ---- | 109 | 109 |
| Maximum Frequency(MHz) | ---- | 368 | 357 |
| Throughput(Mbps) | ---- | 236 | 228 |

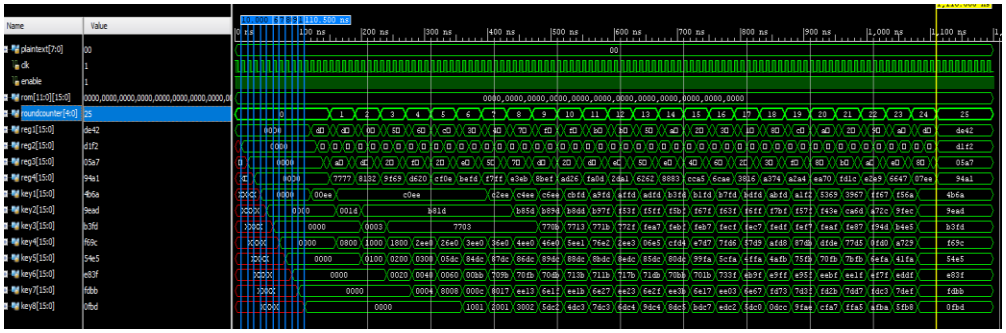


Figure 5.10 Simulation Waveform for the 16-bit Roundbased Architecture for BORON_128

Table 5.8 Synthesis Results for the 16-bit Roundbased Architecture for BORON_128

| Parameter | Resources Available | SBOX (LUT) | SBOX(Boolean) |
|---|---|---|---|
| LUTs | 20800 | 214 | 210 |
| Slice Registers | 41600 | 211 | 211 |
| FPGA Slices | 8150 | 78 | 75 |
| Clock Cycles | ---- | 112 | 112 |
| Maximum Frequency(MHz) | ---- | 283 | 254 |
| Throughput(Mbps) | ---- | 161.7 | 145.14 |

`On comparing 16-bit datapath with 64-bit datapath for BORON_80 and BORON_128, it has been observed that 16-bit datapath consumes for 31.03% lesser FPGA for BORON_80 and 19.35% lesser FPGA slices for BORON_128 than 64-bit datapath. The maximum Frequency has been increaed by 78.47% but due to an increased latency, the overall throughtput decreased by 162.6%.

### 5.2.5 Implementation results of 16-bit Authentication-Encryption Scheme

From the post implementation timing simulation of 16-bit encryption-authentication scheme, it has been observed that the design produced a different response for the same challenges provided in previous two implementations. The design consumed 307 LUTs, 256 slice registers and 106 FPGA slices. The Design Complier results showed that it required 3190 Gate Equivalents for their implementation. Figure 5.11 gives the simulation waveform for 16-bit encryption-authentication scheme.

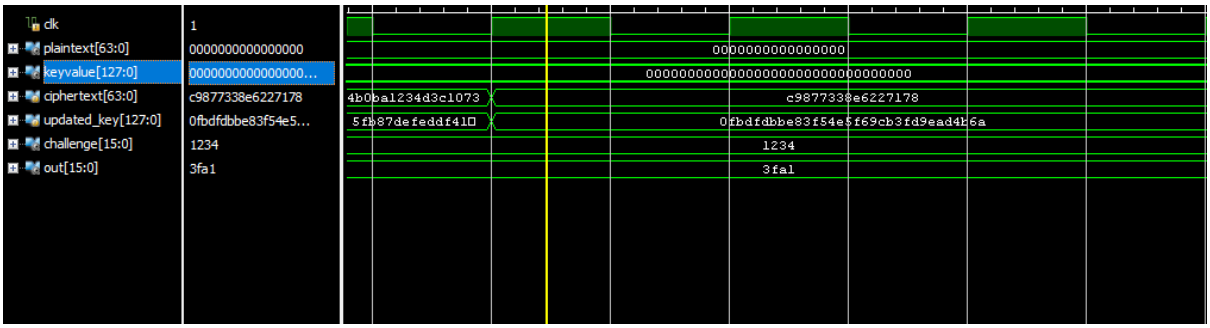



Figure 5.11 Simulation Waveform for the 16-bit Encryption-Authentication Scheme with Challenge 1

From the comaprison of 16-bit and 64-bit authtntication-encryption schemes, it has been observed that 16-bit scheme requires 6.19% lesser FPGA slices and 1131 less Gate equivalents than 64-bit encryption authentication scheme.

From the experimental results it can be concluded that encryption and authentication can be provided efficiently by implemeting 16-bit datapath for BORON cipher with 1190 (3190-2000) extra Gate Equivelents than the number of GE recommended by NIST.

# CHAPTER 6

# CONCLUSION AND FUTURE SCOPE

The software implementation has been done for the BORON cipher and from the results it has been found that it consumes large number of cycles as compared to primitive cryptography algorithm such as AES due to large number of rounds. Next, the hardware implementation of BORON cipher and its reduced data path has been done. In the reduced data-path, 16-bit design was presented in the work for both BORON_80 and BORON_128 and from the results it has been observed that that this design consumed **31.03%** and **19.57%** lesser FPGA slices for BORON_80 and BORON_128 than 64-bit data-path.

Next, a 16-bit Arbiter PUF hardware implementation has been done and integrated with both the datapath design for BORON and it has been observed that **113** and **106** FPGA slices were used for 64-bit and 16-bit encryption-authentication scheme. For estimating the total area, further synthesis of design has been done on DC compiler on SCL 180-nm technology. The results showed that **4321** and **3190** gate equivalents were required for 64-bit and 16-bit encryption authentication scheme. So, 16-bit encryption authentication required **1131** lesser GE than 64-bit scheme. But this lesser area has come at the cost of increased latency and decreased throughput. This 16-bit design is well suited for RFID tags as it provides both encryption and authentication with **1190** more GEs than permitted GEs (2000).Where as other ciphers such as AES for encryption only require **3400** GE [18].

## FUTURE SCOPE

Following are the directions which can be worked upon in near future.

- To generate the soft IP that can be directly called to encrypt and authenticate the message.
- With the aging affect, there can be delay variations. So in order to authenticate the Initialization Vector bits, an Error Correcting Code circuitry can be embedded in the design.

# REFERENCES


[1] K. Finkenzeller, "Introduction," *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition,* pp. 1-9, 2003.

[2] A. Khattab, Z. Jeddi, E. Amini, and M. Bayoumi, *RFID Security*: Springer, 2017.

[3] M. R. Rieback, B. Crispo, and A. S. Tanenbaum, "The evolution of RFID security," *IEEE Pervasive Computing,* vol. 5, pp. 62-69, 2006.

[4] S. A. Weis, "RFID (radio frequency identification): Principles and applications," *System,* vol. 2, pp. 1-23, 2007.

[5] K. Domdouzis, B. Kumar, and C. Anumba, "Radio-Frequency Identification (RFID) applications: A brief introduction," *Advanced Engineering Informatics,* vol. 21, pp. 350-355, 2007.

[6] W. Yao, C.-H. Chu, and Z. Li, "The use of RFID in healthcare: Benefits and barriers," in *RFID-Technology and Applications (RFID-TA), 2010 IEEE International Conference on*, 2010, pp. 128-134.

[7] M. Kaur, M. Sandhu, N. Mohan, and P. S. Sandhu, "RFID technology principles, advantages, limitations & its applications," *International Journal of Computer and Electrical Engineering,* vol. 3, p. 151, 2011.

[8] G. Borriello, "RFID: Tagging the world," *Communications of the ACM,* vol. 48, pp. 34-37, 2005.

[9] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Security in pervasive computing*, ed: Springer, 2004, pp. 201-212.

[10] O. Cheikhrouhou, "Secure group communication in wireless sensor networks: a survey," *Journal of Network and Computer Applications,* vol. 61, pp. 115-132, 2016.

[11] M. Feldhofer and J. Wolkerstorfer, "Strong crypto for RFID tags-a comparison of low-power hardware implementations," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, 2007, pp. 1839-1842.

[12] K. A. McKay, K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, *Report on lightweight cryptography*: US Department of Commerce, National Institute of Standards and Technology, 2017.

[13] R. Tripathi and S. Agrawal, "Comparative study of symmetric and asymmetric cryptography techniques," *International Journal of Advance Foundation and Research in Computer (IJAFRC),* vol. 1, pp. 68-76, 2014.

[14] E. Surya and C. Diviya, "A survey on symmetric key encryption algorithms," *International Journal of Computer Science & Communication Networks,* vol. 2, pp. 475-477, 2012.

[15] M. Agrawal and P. Mishra, "A comparative survey on symmetric key encryption techniques," *International Journal on Computer Science and Engineering,* vol. 4, p. 877, 2012.

[16] S. O. Sharif and S. Mansoor, "Performance analysis of stream and block cipher algorithms," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, 2010, pp. V1-522-V1-525.

[17] S. Chandra, S. Paira, S. S. Alam, and G. Sanyal, "A comparative survey of symmetric and asymmetric key cryptography," in *Electronics, Communication and Computational Engineering (ICECCE), 2014 International Conference on*, 2014, pp. 83-93.

[18] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2004, pp. 357-370.

[19] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw*, et al.*, "PRESENT: An ultra-lightweight block cipher," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2007, pp. 450-466.

[20] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, and Y. Seurin, "Hash functions and RFID tags: Mind the gap," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2008, pp. 283-299.

[21] G. Leander, C. Paar, A. Poschmann, and K. Schramm, "New lightweight DES variants," in *International Workshop on Fast Software Encryption*, 2007, pp. 196-210.

[22] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.-S. Koo*, et al.*, "HIGHT: A new block cipher suitable for low-resource device," in *International Workshop on Cryptographic Hardware and Embedded Systems*, 2006, pp. 46-59.

[23] Y. Eslami, A. Sheikholeslami, P. G. Gulak, S. Masui, and K. Mukaida, "An area-efficient universal cryptography processor for smart cards," *IEEE transactions on very large scale integration (VLSI) systems,* vol. 14, pp. 43-56, 2006.

[24] C. H. Lim and T. Korkishko, "mCrypton–a lightweight block cipher for security of low-cost RFID tags and sensors," in *International Workshop on Information Security Applications*, 2005, pp. 243-258.

[25] G. Bansod, A. Patil, S. Sutar, and N. Pisharoty, "An ultra lightweight encryption design for security in pervasive computing," in *Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2016 IEEE 2nd International Conference on*, 2016, pp. 79-84.

[26] G. Bansod, N. Pisharoty, and A. Patil, "BORON: an ultra-lightweight and low power encryption design for pervasive computing," *Frontiers of Information Technology & Electronic Engineering,* vol. 18, pp. 317-331, 2017.

[27] C. Rolfes, A. Poschmann, G. Leander, and C. Paar, "Ultra-lightweight implementations for smart devices–security for 1000 gate equivalents," in *International Conference on Smart Card Research and Advanced Applications*, 2008, pp. 89-103.

[28] P. Luo, X. Wang, J. Feng, and Y. Xu, "Low-power hardware implementation of ECC processor suitable for low-cost RFID tags," in *Solid-State and Integrated-Circuit Technology, 2008. ICSICT 2008. 9th International Conference on*, 2008, pp. 1681-1684.

[29] S. Kumar and C. Paar, "Are standards compliant elliptic curve cryptosystems feasible on RFID," in *Workshop on RFID security*, 2006, pp. 12-14.

[30] G. Gaubatz, J.-P. Kaps, E. Ozturk, and B. Sunar, "State of the art in ultra-low power public key cryptography for wireless sensor networks," in *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, 2005, pp. 146-150.

[31] Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede, "Elliptic-curve-based security processor for RFID," *IEEE Transactions on Computers,* vol. 57, pp. 1514-1527, 2008.

[32] E. A. Michalski and D. A. Buell, "A scalable architecture for RSA cryptography on large FPGAs," in *Field Programmable Logic and Applications, 2006. FPL'06. International Conference on*, 2006, pp. 1-8.

[33] V. Shadangi, S. K. Choudhary, K. A. K. Patro, and B. Acharya, "Novel Arnold Scrambling Based CBC-AES Image Encryption."

[34] M. Bellare, S. Goldwasser, and D. Micciancio, "“Pseudo-random” number generation within cryptographic algorithms: The DDS case," in *Annual International Cryptology Conference*, 1997, pp. 277-291.

[35] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th annual design automation conference*, 2007, pp. 9-14.

[36] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of PUF-based" unclonable" RFID ICs for anti-counterfeiting and security applications," in *RFID, 2008 IEEE International conference on*, 2008, pp. 58-64.

[37] J. H. Anderson, "A PUF design for secure FPGA-based embedded systems," in *Proceedings of the 2010 Asia and South Pacific Design Automation Conference*, 2010, pp. 1-6.

[38] J. Tay, M. Wong, M. Wong, C. Zhang, and I. Hijazin, "Compact FPGA implementation of PRESENT with Boolean S-Box," in *Quality Electronic Design (ASQED), 2015 6th Asia Symposium on*, 2015, pp. 144-148.

[39] M. Sbeiti, M. Silbermann, A. Poschmann, and C. Paar, "Design space exploration of present implementations for fpgas," in *Programmable Logic, 2009. SPL. 5th Southern Conference on*, 2009, pp. 141-145.

[40] C. A. Lara-Nino, M. Morales-Sandoval, and A. Diaz-Perez, "Novel FPGA-based low-cost hardware architecture for the PRESENT block cipher," in *Digital System Design (DSD), 2016 Euromicro Conference on*, 2016, pp. 646-650.

[41] P. Yalla and J.-P. Kaps, "Lightweight cryptography for FPGAs," in *Reconfigurable Computing and FPGAs, 2009. ReConFig'09. International Conference on*, 2009, pp. 225-230.

[42] L. Huai, X. Zou, Z. Liu, and Y. Han, "An energy-efficient AES-CCM implementation for IEEE802. 15.4 wireless sensor networks," in *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC'09. International Conference on*, 2009, pp. 394-397.

[43] B. J. Mohd, T. Hayajneh, and A. V. Vasilakos, "A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues," *Journal of Network and Computer Applications,* vol. 58, pp. 73-93, 2015.

# PUBLICATIONS

## *Published*

[1]. Dixita Gupta, Ajay Kumar, Kiranjit Kaur, Alpana Agarwal, Manu Bansal, "Hardware Trojans and Their Detection for Cryptography Algorithms and Open Issues", International Journal of Creative Research Thoughts (IJCRT), vol. 6, pp.728-731, 2018. (published)

## *Communicated*

[2]. Dixita Gupta, Ajay Kumar, Manu Bansal, Alpana Agarwal, "An Area Efficient 16-bit datapath for BORON Cipher", Journal of Electrical Engineering and Technology.

# 601662006

*by* Dixita Gupta

**Submission date:** 12-Jun-2018 12:55PM (UTC+0530)
**Submission ID:** 974955566
**File name:** 601662006_Dixita_Gupta_final.docx (1.94M)
**Word count:** 13269
**Character count:** 71399

601662006

ORIGINALITY REPORT

| 19% | % | 19% | % |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Ahmed Khattab, Zahra Jeddi, Esmaeil Amini, Magdy Bayoumi. "RFID Security", Springer Nature, 2017<br>Publication | 5% |
| 2 | Miaoqing Huang, Shiming Li. "A delay-based PUF design using multiplexer chains", 2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig), 2013<br>Publication | 1% |
| 3 | Srinivas Devadas, Edward Suh, Sid Paral, Richard Sowell, Tom Ziola, Vivek Khandelwal. "Design and Implementation of PUF-Based "Unclonable" RFID ICs for Anti-Counterfeiting and Security Applications", 2008 IEEE International Conference on RFID, 2008<br>Publication | 1% |
| 4 | Jaskaranbeer Kaur, Ajay Kumar, Manu Bansal. "Lightweight cipher algorithms for smart cards security: A survey and open challenges", 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), | 1% |

2017
Publication

| | | |
|---|---|---|
| 5 | Bouhouche, T., M. Boulmalf, M. Bouya, and M. Elkoutbi. "A new middleware architecture for RFID systems", Proceedings of 2014 Mediterranean Microwave Symposium (MMS2014), 2014.<br>Publication | 1% |
| 6 | "VLSI Design and Test", Springer Nature, 2017<br>Publication | 1% |
| 7 | Gurudatt Kulkarni, Rupali Shelke, Ramesh Sutar, Sangita Mohite. "“RFID security issues & challenges”", 2014 International Conference on Electronics and Communication Systems (ICECS), 2014<br>Publication | <1% |
| 8 | Lecture Notes in Computer Science, 2008.<br>Publication | <1% |
| 9 | M. E. Ajana, H. Harroud, M. Boulmalf, H. Hamam. "FlexRFID: A flexible middleware for RFID applications development", 2009 IFIP International Conference on Wireless and Optical Communications Networks, 2009<br>Publication | <1% |
| 10 | J. Landt. "The history of RFID", IEEE Potentials, 2005<br>Publication | <1% |