LOW POWER VLSI ARCHITECTURES FOR CRYPTOGRAPHIC ALGORITHMS

A

Thesis submitted

for the award of the degree of

DOCTOR OF PHILOSOPHY

By

BHOOPAL RAO GANGADARI

Under the supervision of

Prof. SHAIK RAFI AHAMED

DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781 039, ASSAM, INDIA

Certificate

This is to certify that the thesis entitled "**Low Power VLSI Architectures for Cryptographic Algorithms**", submitted by **BHOOPAL RAO GANGADARI** (11610227), a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated:

Guwahati.

Dr. Shaik Rafi Ahamed
Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

Declaration

This is to certify that the thesis entitled "**Low Power VLSI Architectures for Cryptographic Algorithms**", submitted by **BHOOPAL RAO GANGADARI** (11610227), submitted by me to the *Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a original research work carried out by me under the supervision of **Prof. Shaik Rafi Ahamed**. The content of this thesis, in full or part, have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated:

Guwahati.

Bhoopal Rao Gangadari
Research Scholar
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

To

My guide **Prof. Shaik Rafi Ahamed**

for his guidance and inspiration

&

My **parents**, **wife** and **children**

for their blessings, love and support

Acknowledgements

This thesis would not have been possible without the immense help and support of several people in various measures. I would like to convey my acknowledgment to all of them.

First and foremost, I express my sincere gratitude to my research supervisor, Prof. Shaik Rafi Ahamed for providing me an opportunity to work under his guidance. It is very difficult to describe my feelings in words to acknowledge my supervisor for his continuous guidance in all aspects, constant motivation and support throughout the doctoral studies. I am very much thankful to him for transforming me from an unstructured form to a structured form in every aspect of my life and showing me a different path of life. It would be completely impossible for me to bring the research as well as the thesis to this form without the immense facilities provided by him in the VLSI Laboratory and the freedom of work he has given to me.

I am thankful to my doctoral committee members Prof. Roy Paily, Prof. Harshal Nemade and Dr. Amitabh Chatterjee for their encouragement and valuable suggestions on my work. I would like to thank faculty members and the office staffs of the Department of Electronics and Electrical Engineering, IIT Guwahati, for their help in carrying out this research work.

I am thankful to my friends Dheeraj Sinha and Satyabrata Dash for their assistance in writing my thesis. I am thankful to Tasleem Khan, Karam Singh, and all other members in the VLSI laboratory. I would like to thank my senior research scholar Dr. Surya Prakash.

I am thankful to my wife Harini for her sacrifice and support. I am heartily thankful to my daughters Anushka Likitha Rao, Anushri Rao and my son Akhilesh Rao for their love and support.

I attribute this achievement to my parents, brothers, sisters, wife and children for their constant blessings, support, silent prayers for my success and moreover, making me stand in this position.

BHOOPAL RAO GANGADARI

Abstract

With the advent of technology and portable devices for communications, cryptography algorithms are widely used in the modern days. Cryptographic algorithms have diverse applications to protect data from unauthorized attacks. This thesis proposes novel approaches for low power Very Large Scale Integrated Circuits (VLSI) architectures for Wireless Body Area Network (WBAN) cryptographic applications so as to improve the performance in terms of area utilization and energy consumption. Motivated by the fact that the present cryptographic algorithms consume high power and energy, there is need to develop low power architectures. Cryptography algorithms like Advanced Encryption Standard (AES), Camellia are widely used in RFID, secure communications, WBAN applications. The block cipher cryptographic algorithms like AES, Camellia are adopted in the latest WBAN standard IEEE 802.15.6 for cryptographic applications. The Substitution Box (S-Box) of these algorithms play a vital role in cryptography. The S-Box is realized using a standard polynomial equation and design is achieved using memory cells. These Look-Up-Table (LUT) based S-Boxes eventually occupy more area and hence consume high power, delay and energy. To overcome the limitation involved in the implementation of classical S-Box, in this thesis, we have proposed several efficient VLSI architectures for S-Box. This thesis investigates on the construction of S-Box using different irreducible polynomial equation. The S-Box can also be realized using Composite Field Arithmetic (CFA) which reduces hardware consumption and low power dissipation. This thesis then characterizes a special class of Cellular Automata (CA) based architectures for hardware implementation of S-Box. The special class of CA based S-Box for AES and Camellia algorithms are realized and implemented using CMOS technology libraries. In addition, we have also proposed hybrid linear cellular automata

and hybrid second order reversible cellular automata based low energy/ power architecture for encryption algorithms. The last part of this thesis deals with the security analysis of the proposed architectures against cyber-attacks. We have carried out the security analysis on the proposed encryption algorithms using cryptographic properties such as Non Linearity, Strict Avalanche Criteria, Correlation Immunity Bias and entropy. Results show that the proposed architectures are efficient in terms of low power dissipation, low energy consumption and secure against any cryptographic attacks.

Contents

List of Figures

List of Tables

List of Acronyms

BAN	Body Area Network
MAC	Medium Access Control Layer
PHY	Physical Layer
FIPS	Federal Information Processing Standards
NIST	National Institute of Standards and Technology
WBAN	Wireless Body Area Network
AES	Advanced Encryption Algorithm
DES	Data Encryption Algorithm
VLSI	Very Large Scale Integrated Circuits
ASIC	Application Specific Integrated Circuits
CMOS	Complimentary Metal Oxide Semiconductor
WBAN	Wireless Body Area Network
S-Box	Substitution Bytes
SR	Shift Rows
MC	Mix Columns
ARK	Add Round Keys
BDD	Binary Decision Diagram
FPGA	Field Programmable Gate Array
NTT	Nippon Telegraph and Telephone Corporation
NESSIE	New European Schemes for Signature, Integrity and Encryption
ISS	International Standard Specification
LUT	Look-Up-Table

CA	Cellular Automata
PCA	Programmable Cellular Automata
HLCA	Hybrid Linear Cellular Automata
$HRCA^2$	Hybrid Second Order Reversible Cellular Automata
CIB	Correlation Immunity Bias
NL	Non Linearity
SAC	Strict Avalanche Criterion
CFA	Composite Field Arithmetic
GF	Galois Fields
LPCA	Linear Programmable Cellular Automata
RCA^2	Second Order Reversible Cellular Automata

1

Introduction

Contents

1.1 Introduction

Nowadays data security plays a vital role not only for defense purpose but also for monitoring medical health and private communication systems. The convergence issues among different research areas such as e-health services [1] have been of interest to scientists and researchers. Wireless sensor networks (WSNs) have been studied in applications such as healthcare. WSN consists of intelligent sensor nodes that are capable of gathering and processing information in an environment and send the data to remote base station. Wireless Body Area Network (WBAN) has been considered as a special type of WSN. It consists of low power wireless sensor nodes which are placed or implanted on or inside a human body for continuous health monitoring. WBAN is a key technology to monitor episodic events or any other abnormal condition and can be used for ambulatory health monitoring. Additionally, it can also be used in diagnostic procedure, maintenance of chronic conditions, supervision of recovery from a surgical procedure and monitoring the effects of drug therapy. The IEEE has launched the Task Group (TG-6) which specially defines and sets the specification for WBAN. In recent years, WBANs have been studied extensively in literature. These studies are mostly focused on various technical issues of the WBANs. The latest IEEE 802.15 Task Group 6 has developed a communication standard optimized for low power devices and operation on, in or around the human body (though not limited to the humans). It provides a variety of applications including medical, consumer electronics, personal entertainment and others [2, 3]. In [4], the authors have provided a comprehensive survey on Wireless Body Area Network. The performance analysis of any wireless access network is an important step to evaluate the system. Since WBAN nodes are powered by small batteries, energy consumption also is one of the important issues in evaluating the system. The IEEE 802.15.6 standard specifies communication at Physical (PHY) layer and Medium Access Control (MAC) layer. The MAC sublayer supports three different PHY layers namely Narrow band (NB), Ultra-Wide band (UWB) and Human Body Communication (HBC). At the MAC sub-layer, IEEE 802.15.6 supports two different types of access mechanisms such as contention access and contention free access [5]. The message security services are to occur at the MAC sub layer and secret

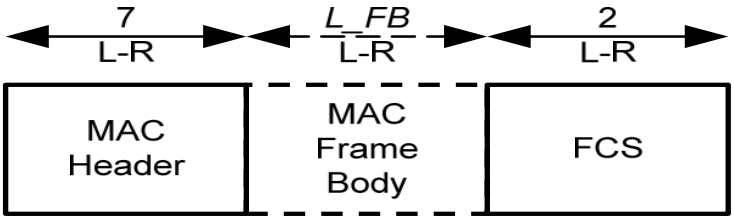

Figure 1.1: MAC Frame Format

keys are generated inside and/or outside the MAC sub layer. A MAC frame consists of a fixed length MAC header, a variable length MAC frame body and a fixed length Frame Check Sequence (FCS) fields as shown in Figure 1.1. The MAC header has length of 7 octets, MAC frame body is of 0 to 255 octets and FCS is of 2 octets. In order to protect the data from attackers, the information in MAC frame body is encrypted using cryptographic algorithms. Moreover, the IEEE 802.15.6 standard adopted Advanced Encryption Standard (AES), Camellia encryption algorithms for the purpose of data security at the MAC sublayer [3, 6, 7].

Cryptology is defined as secure communication in terms of cipher text. Cryptology can be classified into cryptography and cryptanalysis. The goal of cryptography is to protect the data from unauthorized attacks using cryptographic algorithms. Cryptanalysis mainly focuses on breaking the cryptographic systems and retrieving the secret information. Designing of the cryptographic system is a vital task. Cryptography ensures that the data is secure against all possible attacks. Cryptographic algorithms provide confidentiality, data integrity, authentication and non repudiation in data communication. However, the advancement in recent trends of cryptanalysis make the secure communication more prone to unauthorized access. The attackers of cryptography system try to manipulate and break the secret code by adopting the recent strategies of cryptanalysis. A secured cryptography system can sustain any type of attack. The plain text is transformed into cipher text using a secret key by a conventional encryption algorithm. A classification of cryptography algorithms has been shown in Figure 1.2. Cryptography algorithms are classified based on public key ciphers and symmetric (private) key ciphers. In symmetric key ciphers, the secret key is shared between the intended sender and the receiver. In

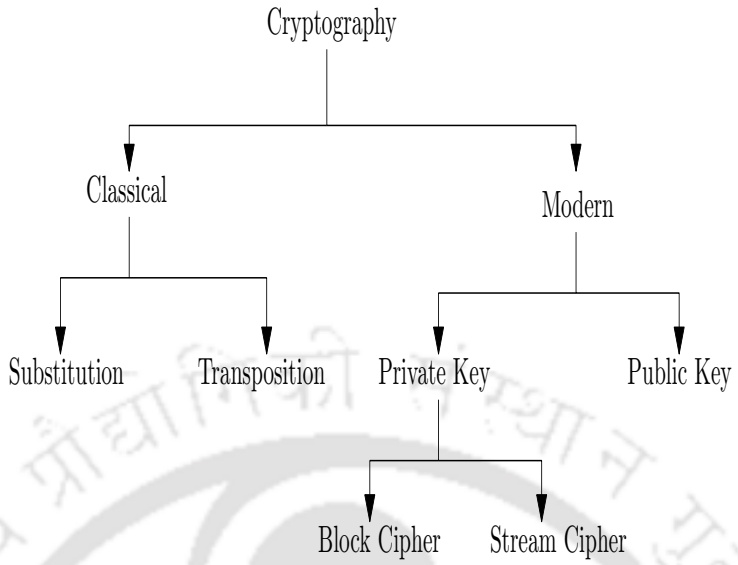

Figure 1.2: Types in Cryptography

public key ciphers, the secret key is shared between a group of users for encryption and decryption. The symmetric key ciphers can further be classified into two types such as block ciphers and stream ciphers. Block cipher algorithms are widely used in the world of cryptography which operate on a block of data for encryption and decryption. The Shannon principle of diffusion and confusion is applied in these block cipher algorithms. Confusion means substitution of data with another data and diffusion means to create an amount of diversification from input data to output cipher text. The Data Encryption Standard (DES), Advanced Encryption Standard (AES), Camellia are some of the encryption algorithms which fall in the category of block cipher algorithms. On the other hand in stream cipher algorithms, each bit of data is encrypted at an instant of time. A Vernam cipher is an example of stream cipher. The significant part of this thesis is focused on cryptographic algorithms that deal with block ciphers.

The DES algorithm was replaced by AES as standard algorithm for encryption and decryption of data [8,9]. The AES algorithm, which is developed by National Institute of Standards and Technology (NIST) has been standardized and adopted in the latest IEEE Standard 802.15.6 for Wireless Body Area Network (WBAN) application due to its

best performance and security [3, 6]. The AES algorithm shown in Figure 1.3, uses four transformations in each round, namely, the Substitution Bytes (S-Box), Shift Rows (SR), Mix Columns (MC) and Add Round Key (ARK), to generate cipher text over plain text in order to provide the desired level of security. The number of rounds of transformation depends upon secret key size. The latest IEEE 802.15.6 standard for WBAN application has recommended a secret key size of 128 bits for AES algorithm which results in 10 rounds of transformation [3, 6]. Out of these 10 rounds, first 9 rounds consist of four transformations S-Box, SR, MC and ARK, while the last round undergoes only three transformations S-Box, SR and ARK. In S-Box transformation, each byte of input data is substituted with another byte using a Look-Up-Table (LUT). The SR transformation is attained by shifting of elements by one byte in order to create diffusion in cipher text. MC transformation is used for attaining diffusion in the block cipher, using a column wise operation, where each column is expressed as a four term polynomial over GF(2^8) field and multiplied by the fixed polynomial $A(x) = (03H)x^3 + (01H)x^2 + (01H)x + (02H)$ with Modulo $x^4 + 1$. In ARK transformation, the ARK cipher keys are generated in the key expansion phase using bitwise XOR operation.

The second encryption algorithm adopted by IEEE 802.15.6 for WBAN is Camellia, which is jointly developed by Nippon Telegraph and Telephone Corporation (NTT) and Mitsubishi Electric Corporation in 2000 [3, 7]. Later, it has been recommended by New European Schemes for Signature, Integrity and Encryption (NESSIE). Camellia has also been adopted as International Standard Specification (ISS) [10]. Camellia is a feistel network based block cipher which use symmetric key to process data. The number of rounds of feistel network is determined by the length of the secret key. The secret key sizes of 128 ,192 and 256 bits result in number of rounds of 18, 24 and 24 respectively. The latest IEEE 802.15.6 standard for WBAN application has recommended a secret key size of 128 bits for Camellia algorithm which results in 18 rounds of feistel network [3, 6]. The operation with the key before and after the round functions is called pre-whitening and post-whitening respectively. After whitening process, the bits are passed through the blocks containing 6 feistel rounds followed by FL/FL^{-1} layer, as shown in Figure 1.4.

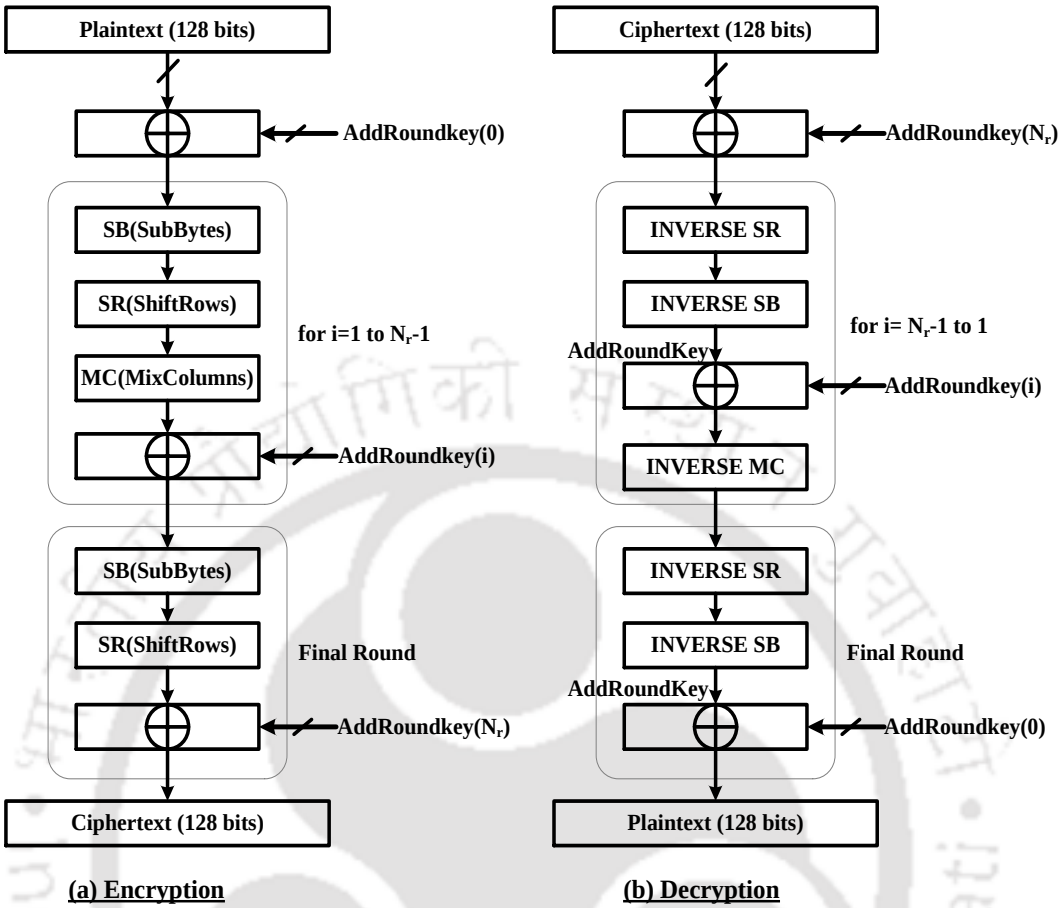

Figure 1.3: Process of AES Encryption and Decryption

There are 18 feistel rounds in total and 2-layers of FL/FL^{-1} function, the FL/FL^{-1} functions are placed after every 6 feistel rounds. The FL/FL^{-1} structure comprises of four S-Boxes. Each feistel round and FL/FL^{-1} layer requires a key of 64-bits in length where each one is different from others. These keys are generated from 128-bits key with a process called key scheduling. The decryption process of Camellia algorithm has a structure similar to the encryption process. The difference is that the order of keys are inversed as shown in Figure 1.5. The S-Box in AES and Camellia plays a vital role to create confusion in the cipher text. The function of S-Box used in cryptographic algorithms is to transform the input data into some secret data using precomputed Look-Up-Table (LUT).

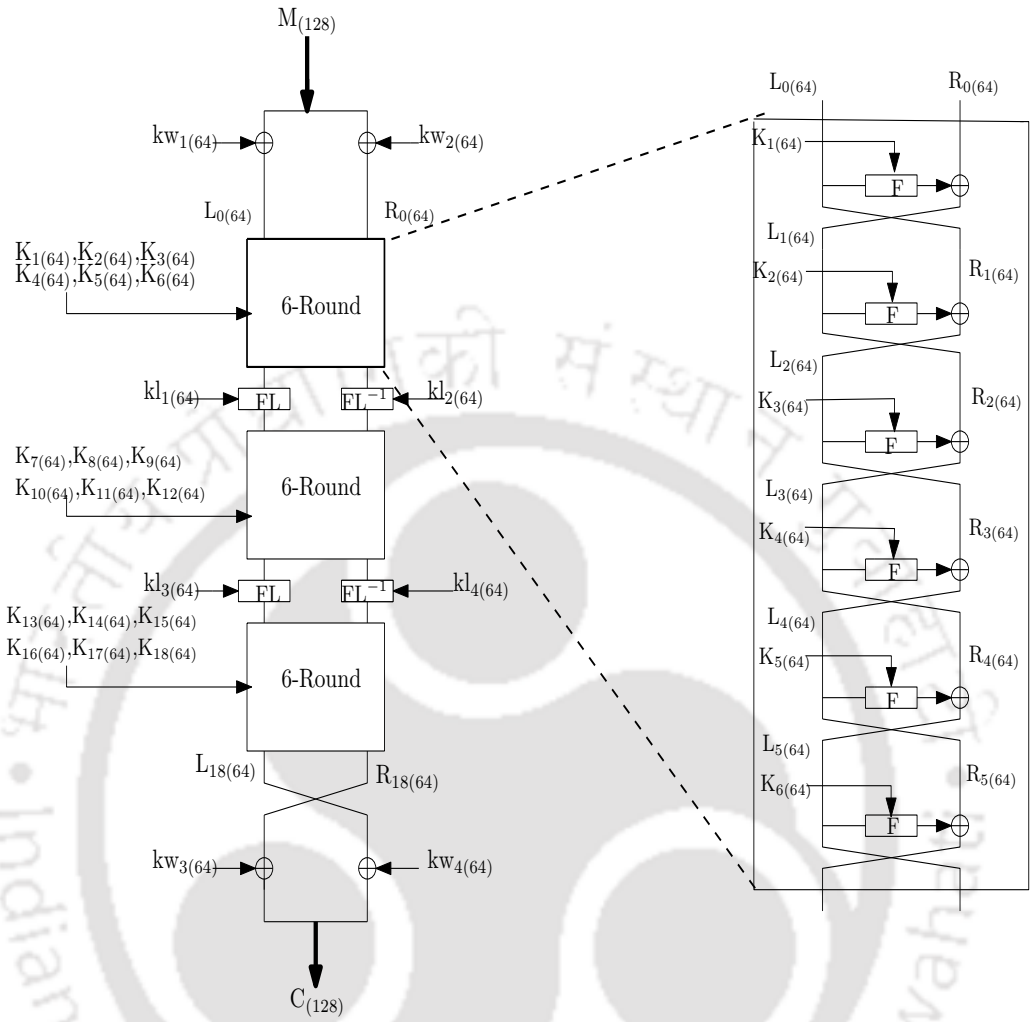

Figure 1.4: A block diagram of Camellia Encryption

1.2 Motivation

The works discussed so far mainly focused on high throughput architectures of encryption algorithms. However, in applications such as WBAN, there is a great demand for architectures with low power and less area [3, 11–13]. For example, the latest IEEE 802.15.6 WBAN standard demands a power consumption of less than 10 mW for both the MAC layer and PHY layer and data rates of 10 bps to 10 Mbps [11]. Several works have been reported in the literature to reduce the complexity of encryption algorithms using Cellular Automata (CA) [14]. However, no work so far has been reported to implement

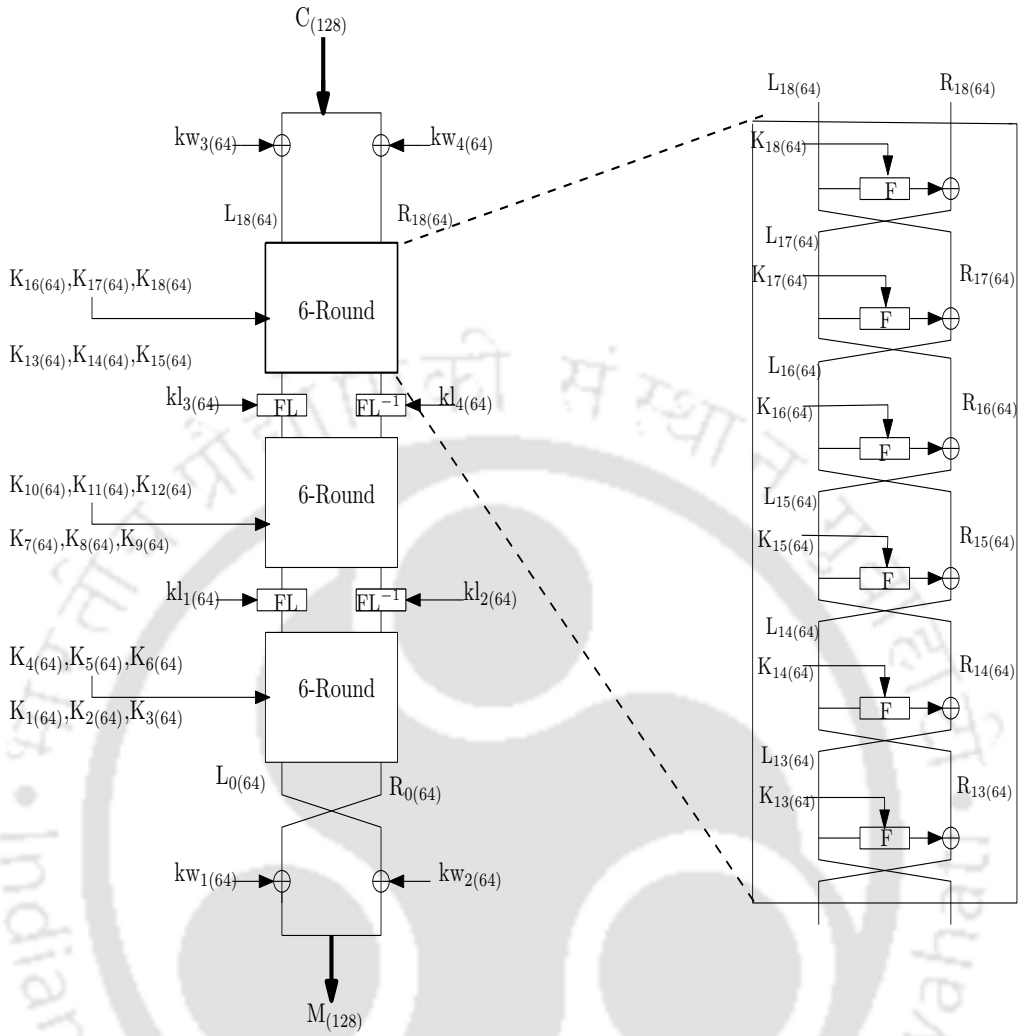

Figure 1.5: A block diagram of Camellia Decryption

CA based encryption algorithmd on hardware. In order to meet the requirement of IEEE 802.15.6 WBAN standard, in this thesis, we have exploited the concept of CA to obtain the low power architecture for AES and Camellia algorithms. As CA structure implementation on hardware utilizes simple logic gates, the S-Box can be realized on hardware using CA to attain low power architectures suitable for WBAN applications. The scope of the thesis is summarized in the next Section.

1.3 Scope of the Thesis

In this thesis, the features of programmable cellular automata and reversible cellular automata have been exploited to realize S-Box of AES and F function of Camellia algorithms. Simulation studies show that the proposed architectures can significantly reduce the power consumption compared to conventional LUT based S-Box realizations. The following contributions have been made in this thesis.

- **Evaluation of S-Box and Impelementation using Composite Field Arithmetic**: In this work, the construction of S-Box achieved using 30 different irreducible polynomial equations has been discussed. The classical LUT based S-Box is designed and constructed using AES standard polynomial equations. In addition, the security provided by construction of S-Box using different irreducible polynomial has been verified with cryptographic properties such as Strict Avalanche Criterion (SAC), Correlation Immunity Bias (CIB), Non Linearity (NL) and entropy. The S-Box construction using CFA technique has also been exploited here.

- **Low Energy Architecture for S-Box**: In this work, the theory of cellular automata has been applied in the construction of S-Box. The S-Box architecture for AES has been designed using PCA and RCA^2. The level of security provided by the S-Box has been verified using cryptographic properties. Simulation studies show that the proposed S-Box architectures using PCA and RCA^2 utilize less chip area, results in low power dissipation and less energy consumption.

- **Construction of CA based Encryption Algorithm**: In this work, low power encryption algorithm has been developed using hybrid linear cellular automata (HLCA) and hybrid second order reversible cellular automata ($HRCA^2$). The security provided by the proposed encryption algorithms has been examined using cryptographic properties. The proposed architectures have been implemented in hardware using 0.18 μm and 0.13 μm technology libraries.

- **Low Power Architectures for F function**: The F function architecture of Camellia encryption algorithm has been designed using linear programmable cel-

lular automata (LPCA) and RCA^2. The security provided by the proposed S-Boxes against cryptanalysis has been verified using cryptographic properties.

1.4 Organization of the Thesis

This thesis emphasizes on cryptography algorithms like AES, Camellia which are widely used in RFID, secure communications. The latest WBAN standard IEEE 802.15.6 has adopted algorithms like AES, Camellia for cryptographic application. In these algorithms, the S-Box eventually consumes more power, incurs more computational delay and energy. To overcome the limitations involved in the implementation of classical S-Box, in this thesis, we have proposed several efficient VLSI architectures for S-Box. In addition, we have also proposed HLCA and $HRCA^2$ based low energy/ power architecture for encryption algorithms. Security analysis has been carried out to test the suitability of the proposed architectures against cyber-attacks. It has been observed that proposed architectures are efficient to overcome cyber-attacks. The rest of the dissertation has been organized as follows:

- **Chapter 2**: This Chapter presents a literature survey and the motivation of the proposed work, major research activities in reduction of power dissipation and energy consumption of cryptographic algorithms. The chapter ends by summarizing the overview of its contents and contributions of the thesis.

- **Chapter 3**: This Chapter investigates the construction of S-Box using different irreducible polynomial equations. The classical S-Box has been constructed using standard irreducible polynomial equation. It has been proved that the S-Box construction can be achieved using different irreducible polynomial. Security provided by the S-Box construction with different irreducible polynomials has been verified in terms of cryptographic properties. The classical S-Box has been replaced by compact S-Box using CFA in order to reduce the power consumption and hardware requirement. The proposed CFA based S-Box architecture has been designed and verified on FPGA.

- **Chapter 4**: This Chapter explains novel S-Box architecture for AES algorithm using PCA and RCA^2. The level of security provided at the end output of the S-Box has been verified in terms of cryptographic properties such as NL, SAC, CIB and entropy. Simulation studies also show that the proposed S-Box architecture with PCA and RCA^2 has reduced area, power and energy requirement.

- **Chapter 5**:This Chapter focuses on architecture for encryption algorithms using HLCA and $HRCA^2$ to reduce power and energy consumption. The proposed encryption algorithm architectures are designed and implemented on CMOS technology libraries. Security provided by the proposed encryption algorithm has been examined with the help of cryptographic properties.

- **Chapter 6**: The Chapter deals with the F function architectures of Camellia encryption algorithm using LPCA and RCA^2. The security provided by the proposed F function against cryptanalysis has been verified using cryptographic properties like NL, SAC, CIB and entropy. The proposed F function architectures using LPCA and $HRCA^2$ have been simulated and synthesized using Cadence RTL Compiler.

- **Chapter 7**: This Chapter summarizes the work presented in this dissertation and highlights the main contributions of the work. It also discusses future recommended research problems for cryptographic algorithms.

2

Literature Survey

Contents

2.1 Review on S-Box Hardware Realization

As S-Box plays a vital role in the cryptographic algorithms such as AES and Camellia, several works have been proposed at algorithmic and architectural level for efficient realization of S-Box. S-Box is designed with the multiplicative inverse over the Galois Field GF (2^8) using the irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$, followed by an affine transformation [3, 6]. The S-Boxes are conventionally designed by using constant values. Hence, the structure is rigid in nature with large number of memory cells. Moreover, these rigid architectures are not secure enough against differential cryptographic attacks [15]. In order to enhance the speed and avoid unbreakable delay, the S-Boxes are designed and implemented using Composite Field Arithmetic (CFA), which involves decomposition of Galois Field GF (2^8) to GF $((2^4)^2)$ or GF $(((2^2)^2)^2)$, using isomorphic mapping functions [16]. Wong et al. have presented all the possible CFA S-Box architectures based on their field representation. Algebraic properties have been used for direct conversion of GF (2^8) to GF $((2^4)^2)$ [17]. In [18], a low power compact S-Box for AES using CFA and Algebraic Normal Form (ANF) with fine-grain pipelining has been implemented. The S-Box implementation on hardware using Binary Decision Diagram (BDD) and Cipher Block Chaining (CBC) mode achieved a throughput of 10 Gbps [19]. In [20], T-Boxes, which are capable to store the round keys generated in key expansion module have been used for S-Box implementation. The T-Box implementation on hardware could achieve a throughput of 1.45 Gbps. In [21], a multistage sub-pipelined Integrated Box (I-Box) architecture for AES has been implemented on FPGA, which achieved higher throughput of 10.66 Gbps and a minimum area of 7884 Slices. However, there is trade off between speed and delay incurred in these architectures.

The AES algorithm with Cipher Block Chaining (CBC) and Interleaved Cipher Block Chaining (ICBC) modes are used for disk encryption [22]. The FPGA based implementation of AES encryption processor has been presented in [23]. The software counter measure against cache attack results in a difficulty to differentiate data for different encryption algorithms [24]. The AES algorithm with key size of 128 bits has been implemented on FPGA with Micro-blaze soft core processor as an interface between hardware and soft-

ware [25]. The FPGA Artix 7 Nexys 4 kit has been used to implement the AES algorithm by configuring it with the help of Software Development Kit (SDK) in Xilinx ISE design suite [25]. In [26], the AES encryption chip with block length of 128, 192, 256 bits and secret key sizes of 128, 192, 256 bits have been implemented using 0.18μm CMOS standard cell library. The chip is capable to achieve a maximum of throughput 2.29 Gbps with a reduced power consumption of 56mW. In [27], the architecture of AES algorithm has been realized on hardware using sub-pipelining technique that could achieve a throughput of 21.56 Gbits/s. In [28], the AES architecture with feedback encryption mode using area optimization approach has been implemented on hardware. This architecture consumes 3.1k gates with a throughput of 121 Mbps, at operating frequency of 153 MHz. The AES algorithm using iterative mode processor has been implemented in [29] on hardware and achieved a throughput of 500 Gbps. In [30], the AES algorithm using pipelining and loop unrolling principle has been implemented using 0.18μm CMOS ASIC technology. It could achieve throughput in the range of 30 to 70 Gbps. In [31], in order to facilitate file sharing and key management, the cryptographic object store system has been implemented on FPGA and ASIC. In [32], the key optimization technique has been used for AES algorithm and the corresponding architecture has been implemented on FPGA, which is suitable for biomedical applications. In [33], the Rivest Shamir Adleman (RSA) public-key standard architecture along with AES algorithm has been used for encryption. The AES algorithm used for encryption of plain text and the secret key of AES has been encrypted using RSA algorithm, which allows to offer a double layer of security to the information. This hybrid cryptographic architecture has also been implemented on FPGA and then synthesized using Xilinx ISE Design Suite. In [34], the AES algorithm has been designed and implemented for network video encryption applications. The design of Customized Stream Advanced Encryption Standard (CSAES) has been used for secure channel coding and Combined Low Density Parity Check (LDPC) code to increase the security level [35].

As the operations involved in S-Box of Camellia algorithm are more computation intensive, several works have been proposed to efficiently implement the S-Box. Camellia encryption algorithm has been implemented on hardware in order to analyze the perfor-

mance in terms of device utilization using two different FPGA devices (Altera Stratix II and Cyclone II), which has been discussed in [36]. The Camellia algorithm implemented on LUTs of Cyclone II has been compared with the wider LUT of Stratix II. The F function plays a vital role in Camellia algorithm for encryption, decryption and key scheduling. The construction and architecture of the F function, especially the S-Box in F function has been well studied. The key scheduling module for 8-round Camellia with 128 bits secret key exhibits resistance against middle attacks [37]. In [38], compact iterative Camellia algorithm with 128 bit secret key achieved a throughput of 426 Mbps without key scheduling and 388 Mbps with key scheduling. A more detailed description of the LUT based S-Box F function and the requirement of SAC property for the S-box has been reported in [39]. The LUT based S-Box F function architecture of Camellia algorithm implementation on FPGA (Xilinx Spartan-3 XC3S505 device) using DRAMs and shift registers could achieve a throughput of 18.41Mbps with 318 slices [40]. In [41], the S-Box of Camellia has been implemented using CFA in order to enhance the speed of S-Box and avoid unbreakable delay. Although, the CFA implementation decreases the computation delay, it increases the power consumption. Several works have been reported in literature to implement the Camellia algorithm using 8051 and Pentium processor [42, 43]. The Camellia algorithm using pipelining architecture has been realized on Virtex FPGA that could achieve a throughput of 32 Gbits/s [44]. In [45], the throughput has been further enhanced to 33.25 Gbits/s using unrolling technique. However, this increase in throughput has been achieved at the cost of increase in hardware complexity and thereby area and power. The hardware implementation of Camellia algorithm using dynamic partial reconfiguration technique has achieved low energy consumption of 453.6 μJ and while reducing power dissipation to 205 mW [46]. An image has been encrypted using Camellia block cipher algorithm combined with the logistic chaotic map in order to provide more security [47]. The key scheduling of Camellia encryption algorithm has been used to generate round keys for pre-whitening while Logistic chaotic map technique has been used to generate the round keys for post-whitening [47].

2.2 CA and related research

The concept of CA has been introduced in 1940 by von Neumann and Ulam to study the biological processes of self reproduction [48]. This concept has been extended to engineering applications by Stephan Wolfram in [49]. CA has been used to generate pseudo-random numbers in [50]. The challenges faced in using CA to solve practical problems and several CA transformations has been discussed in [51]. In [52], the CA with umbrella Cellular Automata Transform (CAT) has been used for data compression applications. The selected rules and pertinent keys of CA have been used to yield better information packing. In [53], watermarking algorithm in which the original image is disassembled with two-dimensional CAT has been discussed. The basic binary watermark has been embedded into certain sub-band and each sub-band is further subdivided into non-overlapped blocks. The coefficient of each block and the bits of watermark are different from the original host image. This watermarking algorithm using CAT is robust to the common watermark attacks. A CA is said to be 1^{st} order one dimensional CA if the next configuration is the function of defined rule and neighborhood configuration cells [54]. An image encryption system has been developed using one dimensional cellular automata (group cellular automata). This encryption system based on symmetric key cryptography in which group cellular automata rules have been used for encryption and decryption, has been reported in [55]. In [56], the theory of CA is mainly focused on study of reversibility and dynamic nature of the system. In [57], the number of reversible rules of one dimensional CA system has been derived using Lemma, each output state of the defined rule can be expressed using Boolean function. If the function is invertible, the rule is reversible which is a desired property in cryptography. If the rule in CA is expressed using EXOR, and/or EXNOR logic, it is called as additive CA. Additive CA is used in VLSI testing, error correcting codes and data encryption. If all the cells in CA evolve using the same deterministic rule, then the CA is called a uniform CA. S.Nandi et al. have implemented a block cryptosystem based on additive CA with group properties [58]. The work [59] has proposed an encryption algorithm based on second order cellular automata that could achieve speed of execution higher than additive CA. The cryptography algorithm based on Multi-Granularity Reversible Cellular

Automata (MG-RCA) has been implemented, in which all the CA cells have different granularities. The granularity have been adjusted dynamically by split recombination behavior during the process of encryption and decryption. The cryptographic systems are resistant to brute force attack, differential attacks and hence capable of achieving better level of security [60]. A parallel image encryption algorithm based on Elementary Cellular Automata (ECA), in which certain rules are capable of generating state variables that satisfy the encryption requirements has been proposed in [61]. Through parallelism of ECA, the original image is divided into two sub-images, each sub-image is encrypted by a separate processor. This encryption technique when used for gray scale image achieved a high execution speed. In this encryption algorithm, two approaches have been used for image encryption. In the first experiment, a secret key was used for image encryption while in the second experiment used two secret keys. Using a secret key with a larger space achieved faster execution than that of two secret keys with smaller space. In [62], an image encryption algorithm based on Reversible Cellular Automata (RCA) has been presented. It could be observed that RCA offer better level of security against common attacks, such as, differential attacks and statistical attacks. In [63], an image encryption algorithm based on a hybrid model of CA, efficiently protected the encrypted image from the unauthorized attacks has been presented. In [64], color image encryption algorithm using logistic chaotic map has been studied, where the generated pixels are used to create confusion and diffusion. This encryption algorithm is executed in two steps. First, the original image pixels are XORed with randomly generated data having values between 0 and 255. The XORed data is permuted by a shuffling engine. Secondly, the diffusion of the permuted data is achieved with the help of ECA based diffusion mechanism. The decoding of the original information is not possible without the knowledge of appropriate rules of CA. This approach of encryption is resistant against brute force attack, statistical attacks and differential attacks, which comes out to be the key features of the algorithm. In [65], a gray scale image has been encrypted using CA and level of security has been examined using cryptographic properties, such as, entropy, correlation, differential and error metrics.

3

Evaluation of S-Box and Implementation using Composite Field Arithmetic

Contents

Objective The objective of this chapter is the construction of S-Box for AES algorithm. The security provided by the cryptographic algorithms depends mainly upon the algebraic construction of the S-Box. The cryptographic properties namely CIB, SAC, NL and entropy are used to evaluate the level of security provided by the S-Box. The traditional S-Box is constructed using standard irreducible polynomial equation. However, result show that the other irreducible polynomial equations offer better level of security compared to that of standard polynomial equation used in AES algorithm. This chapter also emphasizes on construction of S-Box for AES algorithm using CFA in Galois Field (GF) $(((2^2)^2)^2)$. The CFA technique uses decomposition methodology. However, an isomorphic mapping function for CFA is to map the GF (2^8) to its composite field of GF$(((2^2)^2)^2)$. There are eight such mapping functions for construction of S-Box using CFA. The proposed CFA based S-Box hardware implementation show reduction in area by 50% and low power consumption compared to the classical S-Box. The chapter concludes with a brief summary of the S-Box implementation and evaluation techniques.

3.1 Introduction

As noted in Section 1.1, the latest IEEE 802.15.6 standard for WBAN has recommended AES as block cipher algorithm. The S-Box of AES algorithm is traditionally designed using memory cells, which consume quite a large amount of power. In order to overcome this limitation, we have proposed an efficient architecture of S-Box, using CFA for AES algorithm, which reduces the overall hardware complexity and power consumption, compared to the memory based techniques. Simulation results show that the level of security provided by S-Box with different irreducible polynomial equations is comparable with existing techniques.

3.2 Advanced Encryption Standard

AES is a symmetric key cryptographic algorithm which in each round uses four transformations, namely, Substitution Bytes (S-Box), Shift Rows (SR), Mix Columns (MC) and Add Round Key (ARK), to generate cipher text over plain text in order to provide the

desired level of security. The rounds of transformation (N_r) used in the AES algorithm can be determined using the relation $N_r = \frac{S_k}{32} + 6$, where S_k is the key size. For WBAN application, the latest IEEE Standard 802.15.6 has recommended a secret key size of 128 bits for encryption and decryption, which results in 10 rounds of transformations [66]. The initial size of the round key is the 128 bit, which is XORed with the input data to generate subsequent ARK in the key expansion phase. Out of these 10 rounds, the first 9 rounds comprises of S-Box, SR, MC and ARK transformations, whereas the last round performs only the three transformations S-Box, SR and ARK, shown in Figure 1.3. In each round of encryption process, the algorithm performs S-Box, SR, MC and ARK operation on a 4×4 array of bytes called a state, described in the following subsections.

3.2.1 Substitution Bytes

In S-Box transformation, each byte of the input state is substituted by another byte using a precomputed Look Up Table(LUT). The S-Box is computed by multiplicative inverse over the Galois finite field GF (2^8), using the irreducible polynomial $p(x) = x^8 + x^4 + x^3 + x + 1$, followed by an affine transformation. Mathematically, the affine transformation of S-Box in matrix form is as follows:

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{3.1}$$

Traditionally, the classical S-Box is implemented using memory cells which can store the 256 possible values in an 8×8 array of bits. For input data of 128 bits, a total number of sixteen LUT based S-Boxes are required for the AES algorithm. The LUT based S-Box in hexadecimal form has been shown in Table 3.1. For example, if the input data is $a5$,

then the substituted value of S-Box is determined from the Table 3.1 by the intersection of row a and column 5, which results in 06.

Table 3.1: LUT based S-Box

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fd	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

3.2.2 Shift Rows

In SR transformation, the first row remains unchanged and the subsequent three rows are shifted cyclically to the left by 1, 2 and 3 bytes respectively. This transformation creates diffusion in the cipher text.

3.2.3 Mix Columns

The MC transformation operates on column by column, where, each column consists of four term polynomials over GF(2^8) and Modulo (x^4+1) is multiplied with a fixed polynomial $A(x) = (03H)x^3+(01H)x^2+(01H)x+(02H)$. In matrix form, the MC transformation can be expressed as

$$S^1(x) = A(x) \otimes S(x). \tag{3.2}$$

$$\begin{bmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{bmatrix} = \begin{bmatrix} 02H & 03H & 01H & 01H \\ 01H & 02H & 03H & 01H \\ 01H & 01H & 02H & 03H \\ 03H & 01H & 01H & 02H \end{bmatrix} \begin{bmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{bmatrix} \tag{3.3}$$

where $0 \leq C < 4$.

3.2.4 Add Round Key

In ARK transformation, each round key consists of 4 byte words denoted by w_i, generated from key expansion. Key expansion block generates a total of $4(N_r+1)$ number of ARKs. In the initial phase of AES algorithm, the first round key is the initial 128 bits of secret key. The subsequent round keys are calculated iteratively using SubWord, RotWord and Rcon. Each ARK produces 4 words as output from the key expansion block, denoted by ARK $i = (w_{4i},\ w_{4i+1},\ w_{4i+2},\ w_{4i+3})$, where, $i = 0\ to\ N_r$. SubWord is a nonlinear transformation in which each byte of secret key is substituted using S-Box. The Rotation Word (RotWord) is a cyclic left shift of each byte in a word by one byte. Rcon is an array of constant words with the left most byte being non-zero.

3.3 Algebraic construction of S-Box for AES algorithm

S-Boxes are used in modern symmetric key cryptography systems [67]. The function of S-Box is to map input bits onto output bits using predefined LUT. From the knowledge of Galois field, we can say that if p is a non-zero element of a principle ideal domain R, then $\frac{R}{p}$ is a field if p is irreducible. Hence for a prime p and $q = p^n$, we can denote the finite field of order q as GF(q). According to this theorem, for a prime p, GF(q^n) is constructed by using a generating polynomial $m(x)$ of degree n as

$$GF(q^n) = \frac{GF(p)[x]}{m(x)} \tag{3.4}$$

where $m(x)$ is the generating polynomial of degree n in GF(q). The standard irreducible polynomial equation for AES is $m(x) = x^8 + x^4 + x^3 + x + 1$. The S-Box of AES operate at the byte level over GF(2^8) fields, where each byte is denoted as b, then every $b \in$ GF(2^8) can be illustrated by a polynomial of degree 7 as:

$$a_8x^7 + a_7x^6 + a_6x^5 + a_5x^4 + a_4x^3 + a_3x^2 + a_2x + a_1 \tag{3.5}$$

where $a_i \in$ GF(2) and $b \in$ GF(2^8). In this field, the addition is defined by the XOR operation and the multiplication is represented by the polynomial multiplication modulo the generating polynomial. Thus S-Box is basically a mapping function in Galois field GF

$:2^8 \rightarrow 2^8$. The S-Box is constructed by substituting each element with the multiplicative inverse over the GF(2^8) using the irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$ and followed by an affine transformation. In other words, each element undergoes the operation $x :\rightarrow Ax^{-1} + b$. Here, $A \in GL_8(2)$ is the general linear group of degree 8 over GF(2) and $b \in$ GF(2^8) is called the translation vector where A and b are defined as follows:

$$\mathrm{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad and\ b = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Computing the above steps, the S-Box is realized with 256 elements. It is worth mentioning that usage of multiplicative inverse alone without the affine transformation, would make the S-Box and subsequent cipher more vulnerable to cryptanalytic attacks (linear, differential and interpolation attacks). The affine transformation is chosen to lend complexity to the encryption process being carried out, as it has a complicated algebraic expression. Furthermore, the role of translation vector b is also important since it ensures that the S-Box has no fixed points (that is S-Box $(a) \neq a$) and no opposite fixed points (that is S-Box $(a) \neq \bar{a}$). The S-Box, apart from providing non linearity to the cipher, is also a computation intensive operation in AES. Due to its mathematical complexity, the S-Box tends to dominate the hardware resources in the implementation of AES algorithm. Hence, it is very important to design compact and low power S-Box for AES algorithm.

3.4 Cryptographic Properties

The ideal characteristics of a S-Box depend on cryptographic properties, namely CIB, SAC, NL and entropy. If an S-Box satisfies these cryptographic properties, we can say that

the S-Box is cryptographically secure [68]. In order to examine the S-Box for cryptographic properties, the 2^8 output bits are transformed into a single output bit for Boolean function $f_i : B^n \rightarrow B$, where $i \in (1, m)$. In a S-Box, $f : B^n \rightarrow B^m$ and hence there exists m number of functions $\mu = \{f_1, f_2, \cdots \cdot f_m\}$. The truth table representation of S-Box in polarity form is written as $f_k(x) = (-1)^{f(x)}$.

$$f_\beta(x) = (\alpha_1 f_1(x) \oplus \alpha_2 f_2(x) \oplus \alpha_3 f_3(x) ... \oplus \alpha_m f_m(x)) \tag{3.6}$$

The Boolean function f_β is a linear combination of m functions $f_i(x)$, $i \leq m$, where $\alpha_i \in B^m$ are coefficients of the linear function.

3.4.1 Strict Avalanche Criterion (SAC)

SAC has been introduced by Webster and Tavares [69]. As per this criterion, if one bit input is changed in a Boolean function, then half of the output bits should get changed [68]. For a Boolean function, if f is to satisfy SAC, the following condition should be satisfied, $f(x) \oplus f(x \oplus \alpha)$ should be balanced, where the Hamming weight of α is 1 and SAC is denoted by Γ.

$$dSAC_f = max_{1 \leq i \leq n} |2^{n-1} - \sum_{x \in B^n} f(x) \oplus f(x \oplus c_i^n)| \tag{3.7}$$

B^n consist of all the possible input in the n variable function which is basically 2^n different inputs c_i^n consisting of all the element in B^n whose Hamming weight is 1 .

$$\Gamma = max(dSAC_{\mu_i}) \tag{3.8}$$

3.4.2 Entropy

This property gives us the amount of information in the input bits when output bits are already known [70]. If the function is $f : B^n \rightarrow B$, then the entropy is represented by H.

$$H(P_i) = P_i log_2 \frac{1}{P_i} + (1 - P_i) log_2 \frac{1}{1 - P_i} \tag{3.9}$$

where P_i is the fraction of 1s in the output side.

The $(i, j)^{th}$ input/output bit to bit entropy $H(x_i/\mu_j)$ is computed and the parameter

defined by

$$H = min[H\ \frac{(x_i)}{(\mu_j)}][i \in \{1, n\}, j \in \{1, m\}] \tag{3.10}$$

Where $H(x_i/\mu_j)$ is the entropy corresponding to the probability $P(x_i/\mu_j)$

$$H = min(H_{\mu_i}) \tag{3.11}$$

3.4.3 Non Linearity (NL)

The NL of a Boolean function is the minimum distance of the function to set of affine functions. It is denoted by Ψ and is mathematically represented as:

$$N_f = min[d(f, g)],\ where\ g \in A_n \tag{3.12}$$

where A_n is the set of all the affine functions.

$$d(f, g) = 2^{n-1} - 2^{-1}(\langle \eta, \beta \rangle) \tag{3.13}$$

where η, β represent the binary sequence of f, g respectively and $\langle \eta, \beta \rangle$ define the scalar product of sequence , Hence, for a function $f : B^n \rightarrow B$

$$N_f = 2^{n-1} - 2^{-1}[max(\langle \eta, \beta_j \rangle)] \tag{3.14}$$

Where β_j belongs to sequence of all linear function.

$$\Psi = min(N_{\mu_i}) \tag{3.15}$$

3.4.4 Correlation Immunity Bias (CIB)

A Boolean function is said to satisfy CIB of order m, if it is statistically independent of combination of any m input bits. Mathematically, if m input bits are fixed then we can get ${}^nC_m 2^m$ g functions and CIB is represented by Φ.

$$CIB_f(m) = max|2^m * W(g_j) - W(f)| \tag{3.16}$$

where $W(g_j)$belongs to the Hamming weight of all the possible functions keeping m bits in the function f fixed. $W(f)$ corresponds to the Hamming weight of function f.

$$\Phi = max(CIB_{\mu_i}) \tag{3.17}$$

3.5 Analysis of S-Box with different irreducible polynomial equations and Affine matrices

S-Box maps each input element using expression $(Ax^{-1}+b)$, where $A \in GL_8(2)$ and $b \in$ GF(2^8). This simple algebraic expression is the basic element for the construction of S-Box. We have proposed different polynomial equations to vary the design components without affecting the nature of this S-Box. This can be achieved in three ways - changing the underlying field to isomorphic field, changing the affine matrix A and changing the translation vector b. The level of security provided by the S-Box is examined using cryptographic properties [71]. Moreover, it is not possible practically to realize different irreducible polynomial equations on hardware. Irreducible polynomial equations are considered, analyzed, studied in terms of security in order to make the cipher more secure and resistant to various kinds of attacks. From Table 3.2, the level of security of S-Box for AES are evaluated using cryptographic properties such as CIB, SAC, NL and entropy. For studying the effect of varying the design components of S-Box on the cryptographic properties of the cipher, the aforesaid parameters are taken into consideration. We now vary the design components to analyze the corresponding change in the cryptographic properties [72]. It is clear from Table 3.2, that the irreducible polynomial $m(x) = x^8 + x^6 + x^5 + x + 1$ using cryptographic properties has achieved the best values. The reason for achieving the best values for $m(x)$ is that the values of m(x) depend on the vector conjugates. The vector conjugate mapping preserves the algebraic properties and additive inverses. Each irreducible polynomial $m(x)$ has set of eight components to form an ordered set of GF(2^8) vector conjugates. The vector conjugate of $m(x) = x^8 + x^6 + x^5 + x + 1$ attains the best values for the cryptographic properties has been proved in a Lemma and theorem provided in [73]. However, the construction of S-Box can be achieved using other irreducible polyno-

Table 3.2: Values of Cryptographic Properties for AES S-Boxes

Irreducible polynomial	NL	Entropy	CIB	SAC
$x^8+x^4+x^3+x+1$ **(AES)**	**112**	**0.98**	**16**	**16**
$x^8+x^7+x^5+x^4+1$	101	0.99	14	16
$x^8+x^6+x^5+x^4+1$	**110**	**0.99**	**16**	**16**
$x^8+x^4+x^3+x^2+1$	95	0.99	16	16
$x^8+x^7+x^6+x^4+x^3+x^2+1$	105	0.98	16	16
$x^8+x^6+x^5+x^4+x^2+x+1$	98	0.99	14	16
$x^8+x^5+x^3+x+1$	103	0.99	16	16
$x^8+x^7+x^5+x^3+1$	**109**	**0.99**	**14**	**16**
$x^8+x^7+x^6+x^5+x^4+x^3+1$	108	0.98	16	16
$x^8+x^5+x^4+x^3+x^2+x+1$	96	0.99	14	16
$x^8+x^7+x^6+x^4+x^2+x+1$	106	0.99	16	16
$x^8+x^5+x^3+x^2+1$	105	0.99	16	16
$x^8+x^6+x^5+x^3+1$	107	0.99	16	16
$x^8+x^7+x^4+x^3+x^2+x+1$	85	0.99	16	12
$x^8+x^7+x^6+x^5+x^4+x+1$	**112**	**0.98**	**14**	**16**
$x^8+x^5+x^4+x^3+1$	**111**	**0.99**	**16**	**12**
$x^8+x^7+x^5+x+1$	102	0.99	14	16
$x^8+x^7+x^3+x+1$	**110**	**0.98**	**16**	**16**
$x^8+x^6+x^5+x^2+1$	106	0.99	12	16
$x^8+x^6+x^3+x^2+1$	105	0.98	16	16
$x^8+x^7+x^5+x^4+x^3+x^2+1$	**111**	**0.98**	**16**	**16**
$x^8+x^6+x^5+x^4+x^3+x+1$	**110**	**0.98**	**16**	**12**
$x^8+x^6+x^4+x^3+x^2+x+1$	100	0.99	16	16
$x^8+x^7+x^6+x^5+x^4+x^2+1$	104	0.99	14	16
$x^8+x^7+x^3+x^2+1$	**108**	**0.99**	**14**	**16**
$\boldsymbol{x^8+x^6+x^5+x+1}$	**112**	**0.99**	**14**	**12**
$x^8+x^7+x^6+x^5+x^2+x+1$	89	0.98	16	16
$x^8+x^7+x^6+x^3+x^2+x+1$	105	0.99	16	16
$x^8+x^7+x^2+x+1$	98	0.98	16	12
$x^8+x^7+x^6+x+1$	92	0.99	14	16

mial equations. The function $f : B^n \rightarrow B^m$ of S-Box for different irreducible polynomials decides the level of security provided against cryptographic attacks [71]. A mathematical test for irreducibility of a polynomial is achieved by using Rabin's Test. It states that a polynomial $C \in \mathrm{GF}(p)[x]$ of degree d is irreducible if and only if $(X^p)^d = x \ Modulo \ C$. The values observed with the cryptographic properties can be enhanced by changing the isomorphic fields. The change of underlying field to an isomorphic field has been achieved using different irreducible polynomials with the same degree. The number of irreducible polynomial equations of degree n over $\mathrm{GF}(p)$ is given by

$$\frac{1}{n} \sum^{d/n} \mu(\frac{n}{d}) \ P^d \tag{3.18}$$

where μ is the Mobius function. Using the above equation, the total number of irreducible polynomial equations with degree 8 over GF(2) (including the AES standard irreducible polynomial) are found to be 30. Table 3.4 show the observed values using the cryptographic properties on isomorphic fields generated by each of these polynomials. The level of security is verified using cryptographic properties for AES algorithm by varying the affine matrix. There exist a total number of 255 affine matrices for 8×8 S-Box dimensions. From the pool of 255 matrices, only 190 matrices are invertible. Using these invertible matrices, 190 AES like S-boxes can be generated by replacing affine matrices. In [74–76], it is reported that there are fix points and repetition of entries in the S-box by replacing affine matrices. Furthermore, Waquas et al. found that a total of 47 S-Boxes can be generated by replacing affine matrices [74]. We have considered these 47 nonsingular binary affine matrices of size 8×8 for the construction of S-Boxes. After performing the simulations on 47 affine matrices, we have found that the following nonsingular binary affine matrix has achieved the best value.

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Table 3.3: Value using Cryptographic Properties for affine matrix of S-Box

NL	Entropy	CIB	SAC
106	0.99	14	12

The following two matrices also result in comparable values of cryptographic properties.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Table 3.4: Values using Cryptographic Properties for Different affine matrix of S-Box

NL	Entropy	CIB	SAC
108	0.97	14	14
104	0.98	14	15

The security provided against cryptanalysis by the generated S-Boxes using the affine matrices are examined using cryptographic properties. It is observed from Table 3.3 and Table 3.4, that the values of cryptographic properties vary with the corresponding change in the affine matrix.

3.6 Construction of S-Box using Composite Field Arithmetic

Mathematically, the classical S-Box used in the AES algorithm can be expressed as,

$$S^1 = MS^{-1} + C \tag{3.19}$$

where M is an 8×8 binary matrix and C is an 8-bit binary vector. The computation of multiplicative inverse over GF(2^8) is a complex task and consumes enormous hardware. In order to reduce the hardware complexity, S-Box is designed using CFA. In CFA, the GF(2^8) field is decomposed into lower order fields using irreducible polynomials of degree 2 which drastically reduces the gate count and thus reduces power consumption as well. An element in the composite field GF(2^4)^2) can be expressed as $bx + c$ and its multiplicative inverse using the extended Euclidean Algorithm is given by the following equation

$$(bx + c)^{-1} = b(b^2\lambda + c(b + c))^{-1}x + (c + b)(b^2\lambda + c(b + c))^{-1} \tag{3.20}$$

where b is the most significant nibble and c is the least significant nibble. The irreducible polynomials used in the CFA are as follows:

$$GF(2) \rightarrow GF(2^2) \qquad Q_0(x) = x^2 + x + 1 \tag{3.21}$$

$$GF(2^2) \rightarrow GF((2^2)^2) \qquad Q_1(x) = x^2 + x + \phi \tag{3.22}$$

$$GF((2^2)^2) \rightarrow GF(((2^2)^2)^2) \qquad Q_2(x) = x^2 + x + \lambda \tag{3.23}$$

The values of the constants ϕ and λ must be chosen carefully to ensure that the polynomials $Q_1(x)$ and $Q_2(x)$ remain irreducible, respectively. For example, $q = \{1100\}_2$ can be split as $q_H x + q_L$, where $q_H = \{11\}_2$ and $q_L = \{00\}_2$. Furthermore, $q_H = \{11\}_2$ can also be split as $\{1\}_2 x + \{1\}_2$ and similarly for q_L. The above decompositions from 4 bits to 1 bit are done using the irreducible polynomials presented in [77]. There exists two values of ϕ and eight values of λ that make the respective polynomials irreducible [78]. Overall, we have 16 combinations of $\{\phi, \lambda\}$ with which we can construct the composite field $GF(((2^2)^2)^2)$ for S-Box. The values of ϕ and λ are given below:

$$\phi = \{1,0\}_2 \, \lambda = \{1000\}_2 \, , \{1100\}_2 \, \lambda = \{1001\}_2 \, , \{1101\}_2 \tag{3.24}$$

$$\phi = \{1,1\}_2 \, \lambda = \{1010\}_2 \, , \{1110\}_2 \, \lambda = \{1011\}_2 \, , \{1111\}_2 \tag{3.25}$$

Out of these 16 possible ways of construction, we have selected 4 optimum values of ϕ and λ which lead to the minimum number of gates. The optimum pairs of $\{\phi, \lambda\}$ are as follows:

$$\phi = \{1,0\}_2 \, \lambda = \{1100\}_2 \, , \phi = \{1,1\}_2 \, \lambda = \{1000\}_2 \tag{3.26}$$

$$\phi = \{1,0\}_2 \, \lambda = \{1111\}_2 \, , \phi = \{1,1\}_2 \, \lambda = \{1010\}_2 \tag{3.27}$$

For computing the multiplicative inverse, CFA cannot be applied directly to a polynomial in GF(2^8). First we have to map each element of GF(2^8) to its composite field with an isomorphic mapping function $f(q) = \delta \times q$. The matrix δ is derived from the irreducible polynomials of GF(2^8) and its composite field. The δ matrix is given below.

$$\delta = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

3.7 Hardware Construction of CFA in Galois Field for S-Box

The overall architecture for calculating the multiplicative inverse in GF($(2^4)^2$) is shown in Figure 3.1. The multipliers in GF(2^4) can be decomposed into multipliers in GF(2^2) and then to GF(2), in which a multiplication is simply an AND operation.

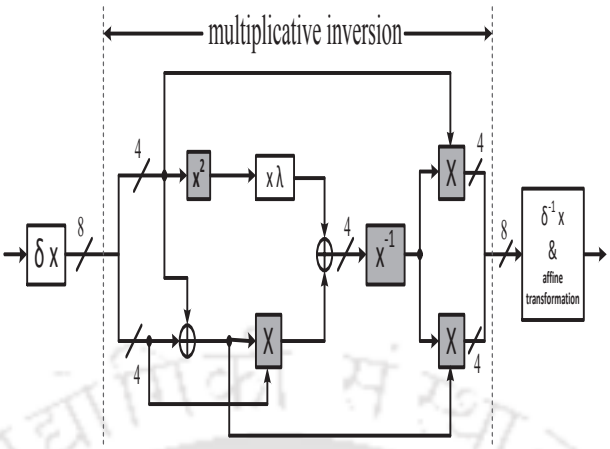

Figure 3.1: Composite Field Arithmetic

3.7.1 Squarer in GF(2^4) Block

The squarer in GF(2^4) block is shown in Figure 3.2, the four input bits $k \in$ of GF$((2^2)^2)$ are represented as $\{k_3, k_2, k_1, k_0\}$ and $\phi = \{1, 0\}_2$ and mathematically expressed as

$$\begin{aligned} k_3^{'} &= k_3, \\ k_2^{'} &= k_3 \oplus k_2, \\ k_1^{'} &= k_2 \oplus k_1, \\ k_0^{'} &= k_3 \oplus k_1 \oplus k_0 \end{aligned} \tag{3.28}$$

The bit expressions by considering $\phi = \{1, 1\}_2$ is given by

$$\begin{aligned} k_3^{'} &= k_3, \\ k_2^{'} &= k_3 \oplus k_2, \\ k_1^{'} &= k_3 \oplus k_2 \oplus k_1, \\ k_0^{'} &= k_1 \oplus k_0 \oplus k_2 \end{aligned} \tag{3.29}$$

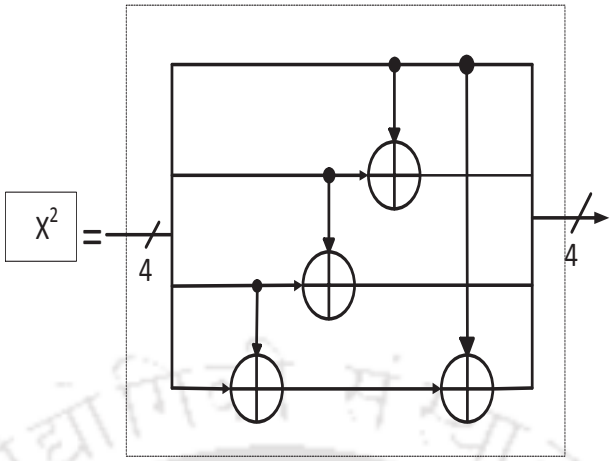

Figure 3.2: Squarer in GF(2^4)

3.7.2 Multiplier in GF(2^4) Block

It is clear from Figure 3.3, the GF(2^4) block itself contains two blocks namely, Xϕ and X. The GF $((2^2)^2)$ multiplier block is simplified by decomposing the field into GF$((2^2)^2)$ which implies two bit multiplications instead of four bits, as shown in Figure 3.4. Furthermore, it is simplified by decomposing the field to GF(2^2), where one bit multiplication is performed.

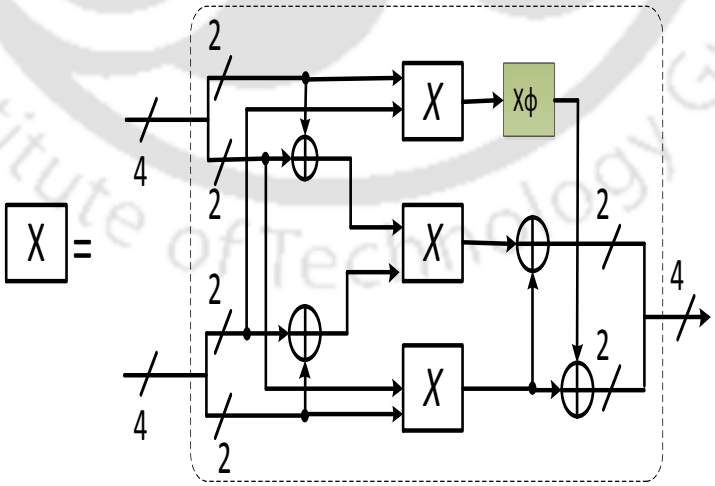

Figure 3.3: Multiplier in GF$((2)^4)$

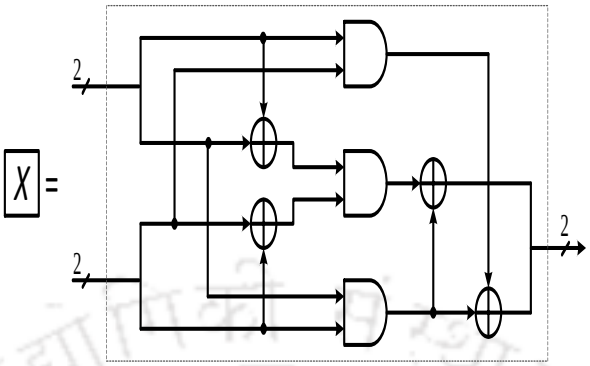

Figure 3.4: Multiplier in GF$((2)^2)$

3.7.3 Xλ Block in GF(2^4)

The Xλ block in GF(2^4) depends on both the values of ϕ and λ. This block can also be computed by mapping GF$(2^2)^2)$ to GF(2^2) as depicted in Figure 3.5. The bit expressions for the optimum pairs of ϕ and λ are illustrated as follows.

$$\begin{aligned} \phi = \{1,0\}_2 \, \lambda = \{1100\}_2 \,, \\ k_3^{'} = k_2 \oplus k_0, \\ k_2^{'} = k_3 \oplus k_2 \oplus k_1 \oplus k_0, \\ k_1^{'} = k_3, \\ k_0^{'} = k_2 \end{aligned} \tag{3.30}$$

$$\begin{aligned} \phi = \{1,0\}_2 \, \lambda = \{1111\}_2 \,, \\ k_3^{'} = k_2 \oplus k_0, \\ k_2^{'} = k_3 \oplus k_2 \oplus k_1 \oplus k_0, \\ k_1^{'} = k_3 \oplus k_0, \\ k_0^{'} = k_2 \oplus k_1 \oplus k_0 \end{aligned} \tag{3.31}$$

$$\begin{aligned} \phi = \{1,1\}_2 \, \lambda = \{1000\}_2 \,, \\ k_3^{'} = k_2 \oplus k_0, \\ k_2^{'} = k_3 \oplus k_2 \oplus k_1 \oplus k_0, \\ k_1^{'} = k_3, \\ k_0^{'} = k_2 \end{aligned} \tag{3.32}$$

$$\begin{aligned} \phi = \{1,1\}_2 \, \lambda = \{1010\}_2 \,, \\ k_3^{'} = k_2 \oplus k_0, \\ k_2^{'} = k_3 \oplus k_2 \oplus k_1 \oplus k_0, \\ k_1^{'} = k_3 \oplus k_1 \oplus k_0, \\ k_0^{'} = k_2 \oplus k_1 \end{aligned} \tag{3.33}$$

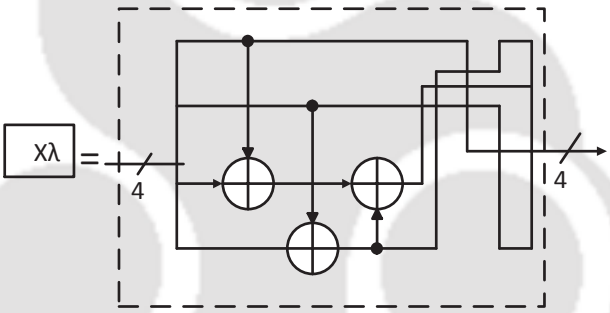

Figure 3.5: Constant multiplier (xλ)

3.7.4 Xϕ Block

The Xϕ block multiplies the 2 bits input to a constant ϕ as shown in Figure 3.6. Taking the input $k \in \mathrm{GF}(2^2)$ as $\{k_1, k_0\}$. The bit expressions of $\left\{k_1^{'}, k_0^{'}\right\}$ for $\phi = \{1,1\}_2 \,, \phi = \{1,0\}_2$ are given as

$$\begin{aligned} \phi = \{1,0\}_2 \\ k_1^{'} = k_1 \oplus k_0, \\ k_0^{'} = k_1 \end{aligned} \tag{3.34}$$

$$\begin{aligned} \phi &= \{1,1\}_2 \\ k_1^{'} &= k_0, \\ k_0^{'} &= k_2 \oplus k_1 \end{aligned} \tag{3.35}$$

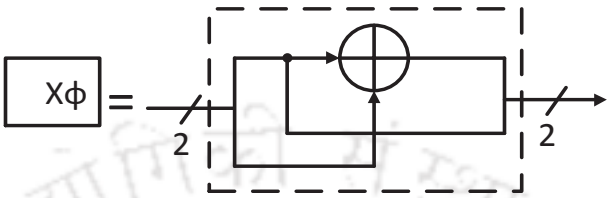

Figure 3.6: Constant multiplier (xϕ)

3.7.5 Inversion in GF(2^4) Block

There are several methods for the implementation of inversion in GF(2^4) block. The direct computation approach is used for the construction of inversion in GF(2^4) block. The bit expressions of this approach for input $k \in$ GF(2^4) as $\{k_3, k_2, k_1, k_0\}$ are given as follows.

$$\begin{aligned} \phi &= \{1,0\}_2 \\ k_3^{-1} &= k_3 \oplus k_3k_2k_1 \oplus k_3k_0 \oplus k_0, \\ k_2^{-1} &= k_3k_2k_1 \oplus k_3k_2k_0 \oplus k_3k_0 \oplus k_2 \oplus k_2k_1, \\ k_1^{-1} &= k_3 \oplus k_3k_2k_1 \oplus k_3k_1k_0 \oplus k_2 \oplus k_2k_0 \oplus k_1, \\ k_0^{-1} &= k_3k_2k_1 \oplus k_3k_2k_0 \oplus k_3k_1 \oplus k_3k_1k_0 \oplus k_3k_0 \\ &\quad \oplus k_2 \oplus k_2k_1 \oplus k_2k_1k_0 \oplus k_0 \oplus k_1 \end{aligned} \tag{3.36}$$

$$
\begin{aligned}
\phi &= \{1,1\}_2 \\
k_3^{-1} &= k_2 \oplus k_0k_3 \oplus k_1k_2k_3, \\
k_2^{-1} &= k_3 \oplus k_0k_3 \oplus k_1k_2 \oplus k_0k_2k_3 \oplus k_1k_2k_3, \\
k_1^{-1} &= k_1 \oplus k_2 \oplus k_0k_2 \oplus k_0k_3 \oplus k_1k_2 \oplus k_1k_3 \\
&\quad \oplus k_1k_2k_3 \oplus k_0k_1k_3, \\
k_0^{-1} &= k_0 \oplus k_1 \oplus k_3 \oplus k_0k_2 \oplus k_0k_3 \oplus k_1k_2 \\
&\quad \oplus k_0k_1k_2 \oplus k_0k_1k_3 \oplus k_0k_2k_3 \oplus k_1k_2k_3
\end{aligned} \tag{3.37}
$$

Once the multiplicative inverse is calculated, it is mapped back to its respective element in GF(2^8) by an inverse isomorphic function $f(q) = \delta^{-1} \times q$ and the corresponding δ^{-1} matrix is given below.

$$
\delta^{-1} = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}
$$

3.8 Hardware Implementation of Composite Field Arithmetic

The theoretical CFA based S-Box has been implemented and verified with number of test vectors on FPGA (XC2VP30) Virtex-II pro board using Xilinx ISE tool, as shown in Table 3.5.

The proposed architectural design of CFA based S-Box has been implemented using Verilog for four optimum cases of $\{\phi, \lambda\}$ values shown in Table 3.5 and Table 3.6. The CFA design implementation consists of multiplication units, inversion units, isomorphic mapping (δ), squarer, inverse isomorphic mapping (δ^{-1}) and affine transformation as

Table 3.5: FPGA Implementation of CFA for AES algorithm

FPGA	Architecture using LUT S-Box	Slices/ LUT	Power (μW)
Cyclone II EP2C5T144C6	Wong et al. [79]	95	NA
XV2VP30	Shantini et al. [80]	85	NP
	Proposed Architecture using CFA	Slices/ LUT	
XV2VP30(Ours)	$\phi = \{1,0\}_2\, \lambda = \{1100\}_2$	42	0.660
XV2VP30(Ours)	$\phi = \{1,1\}_2\, \lambda = \{1000\}_2$	47	0.750
XV2VP30(Ours)	$\phi = \{1,1\}_2\, \lambda = \{1010\}_2$	45	0.560
XV2VP30(Ours)	$\phi = \{1,0\}_2\, \lambda = \{1111\}_2$	52	0.852

* NA means not applicable

* NP means not provided

Table 3.6: ASIC Implementation of CFA for AES algorithm

	Architecture using LUT S-Box	Area (Gates)	Power (mW)
StandardCells UMCL18G212	Wong et al. [81]	174	NP
0.11 μm	Satoh et al. [82]	294	NP
ASIC	D.Canright [83]	255	NP
TSMC 0.18 μm	Proposed Architecture using CFA	Gates	
	$\phi = \{1,0\}_2\, \lambda = \{1100\}_2$	164	0.42
	$\phi = \{1,1\}_2\, \lambda = \{1000\}_2$	169	0.51
	$\phi = \{1,1\}_2\, \lambda = \{1010\}_2$	158	0.39
	$\phi = \{1,0\}_2\, \lambda = \{1111\}_2$	172	0.54

* NP means not provided

shown in Figure 3.1.

It is clear from Table 3.5, that there is reduction in number of LUT/Slices compared with the existing designs [79, 80]. It has also been observed that there is decrease in the area of proposed design by 50% in comparison with the existing design [80]. The ASIC results of CFA based S-Box using TSMC 0.18μm shown in Table 3.6. As compared with various architectural implementations of S-Box reported in [81–83], the proposed CFA S-Box shows reduction in number of gate count as shown in Table 3.6. The optimum values of $\phi = \{1, 0\}_2 \lambda = \{1100\}_2$ achieved the least number of gate count and low power consumption compared with existing works as shown in Table 3.5 and Table 3.6.

3.9 Conclusion

In this chapter, the algebraic construction of S-Box has been achieved using different polynomial equations. The change in irreducible polynomial equation of S-Box attribute to the level of security provided against cryptanalysis. However, the standard irreducible polynomial equation for the construction of AES S-Box has also been taken into consideration . We have observed that changing the underlying field to isomorphic field and changing the affine matrix provide better performance compared with that of standard irreducible polynomial equation. Moreover, it has also been found that the translation vector does not contribute to the security of S-Box, since no change has been observed on changing the translation vector. The S-Box has been implemented on FPGA using CFA and simulations are carried out using Xilinx ISE Design Suite. For implementation of CFA S-Box, different values of ϕ and λ are taken into consideration. It has been found that there exist 16 ways for construction of S-Box using CFA. Moreover, the optimum values of ϕ and λ are considered for construction of S-Box, achieved low hardware utilization for $\phi = \{1, 0\}_2, \lambda = \{1100\}_2$ and low power consumption. We have also observed that the coefficients of irreducible polynomial influence the isomorphic mappings and sub field operations. Moreover, we have also found that the construction of S-Box by CFA using standard irreducible polynomial requires less the number of gates as compared to the classical S-Box of AES algorithm.. There is reduction in area of proposed CFA based

S-Box compared to conventional S-Box by 50% and hence there is also decrease in power consumption.

4

S-Box realization using Linear Cellular Automata and second order Reversible Cellular Automata

Contents

Objective This chapter focuses on low power dissipating, less energy consuming architectures for S-Box using programmable cellular automata (PCA) and programmable second order reversible cellular automata (RCA^2). The architectures entail low power implementation with minimal delay overhead. Performance of the proposed S-Box architectures has been examined in terms of security using the cryptographic properties such as NL, CIB, SAC and entropy. It has been found that the proposed architectures are secure enough against cryptanalysis. It has been observed that the proposed S-Box modules using PCA and RCA^2 are more flexible, dynamic in nature and consume less power compared to the traditional LUT based S-Box. As the proposed architectures consume less energy, they are suitable for WBAN applications.

4.1 Introduction

The architecture of AES algorithm has been revisited in Section 1.3. In order to meet the requirements of WBAN, in this chapter, we have proposed a CA based architecture for realization of S-Box. The basic function of S-Box is to transform one byte of input data to another byte of secret data using LUT. The truth table of S-Box is basically a map function $f : B^n \rightarrow B^m$. The LUT based S-Box architecture requires more area and high energy consumption. Hence, LUT based S-Box architecture is not suitable for IEEE Standard 802.15.6 for WBAN applications. Moreover, the standard also demands a highly secure, cryptographic algorithm with less area and energy consumption.

4.2 Formulation of S-Box using Cellular Automata

The basic structure of CA, shown in Figure 4.1, consists of a group of cells with a finite size of length 8, which evolve at discrete time steps using deterministic rule. Each cell can store one of the two states 0 and 1. If the rightmost and the leftmost (extreme) cells of this finite size CA are considered to be adjacent to each other, then the CA is called a circular boundary CA. The one dimensional circular boundary CA evolves with different neighborhood configurations of elementary CA. Each elementary CA consists of central cell i surrounded by neighborhood cells of a defined radius r. Therefore the total

number neighbor cells in elementary CA is given as $n_i = 2r + 1$, including the central cell. We have considered $r = 1$. The total number of elementary CA given as $L = 2^{n_i}$. The next state of central cell R_i^{t+1} at time step $(t + 1)$ depends on the current state of central cell R_i^t and also the neighborhood cells R_{i-1}^t, R_{i+1}^t, at time t with deterministic rule of function f_p. Mathematically, R_i^{t+1} can be expressed as

$$R_i^{t+1} = f_p(R_{i-1}^t, R_i^t, R_{i+1}^t) \tag{4.1}$$

Table 4.1: Truth table for Rule 90 and 75

	111	110	101	100	011	010	001	000	
Decimal 90	0	1	0	1	1	0	1	1	Rule 90
Decimal 150	0	1	0	0	1	0	1	1	Rule 75

The representation of deterministic rules f_p in decimal form shown in Table 4.1 and the total number of CA rules considered are $2^L = 256$, where L is the array of cells. If the rules in a CA can be expressed using EXOR logic and/or EXNOR logic only, it is called an additive CA. The additive CA has been used in VLSI testing, bit-error correcting code and data encryption. If all the cells in CA evolve using the same deterministic rule, then the CA is called uniform CA. The dynamic nature of one dimensional periodic uniform CA depends on deterministic rule f_p and the number of iterations. We have considered a Programmable Cellular Automata (PCA) which is a modified version of one dimensional periodic uniform CA structure in which all the cells in the lattice obey the same rule [58].

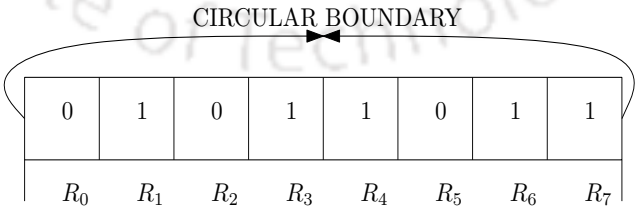

Figure 4.1: A Cellular Automata array of size $(R_0 - R_7)$ with a circular boundary condition.

4.3 Proposed PCA based S-Box

In order to overcome the limitations of classical S-Box, we have proposed a PCA based architecture for S-Box which requires low energy consumption and is also dynamic in nature. Moreover, the secret information from the existing AES algorithm architecture can be revealed using power analysis attacks [15]. Unlike the conventional LUT based S-Box, the proposed S-Box is dynamic in nature because of the fact that output of S-Box is a function of input rule which can be programmed. There is a total number of 256 rules that can be used to program onto the registers at discrete time steps. The output R_i^{t+1} depends on the input control signals $R_{i-1}^t, R_i^t, R_{i+1}^t$ as shown in Figure 4.2. For example, from Table 4.1, if the input rule is 90 and input data is 110, the output R_i^{t+1} should be 1. The output of proposed basic PCA structure for a given 8 bit input rule is one bit, as

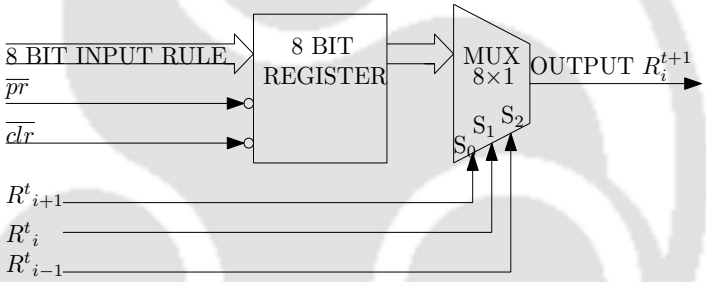

Figure 4.2: PCA Based basic Cell Structure

shown in Figure 4.2. In order to implement the S-Box which operates on 8 bits, eight such basic cells, shown in Figure 4.2 need to be interconnected. The proposed architectural design of 8×8 array PCA based S-Box implemented using logic gates, multiplexers and registers has been shown in Figure 4.3 . The initial 8 bits of CA array will be loaded into register R_1 using preset and clear signals. The bits in the register R_1 will be applied as control signals to 8:1 MUX (M_1-M_8) in circular fashion whose input is an 8 bit rule. First 3 bits R_7, R_0 and R_1 will act as a control signals to M_1, the rotated bits R_0, R_1 and R_2 to M_2 and the last MUX M_8 the control signal are R_6, R_7 and R_0. The MUXs produce output according to Table 4.1. The multiplexer outputs, so produced, will be used as CA array bits in subsequent iterations.

The control logic has a 6 bit up counter and a comparator. If the count value of counter is equal to the number of iterations in time step, then the output of control logic circuit goes high to enable the register (R_2). The latency incurred in computing the S-Box depends upon the number of iterations considered for PCA S-Box. However, on the other side, the ASIC implementation of PCA based S-Box architecture shown in Figure 4.3 utilizes few logic elements compared to that of LUT based S-Box. As a result, CA based S-Box architecture consumes less power and require small chip area and hence this hardware realization is much suitable for WBAN applications.

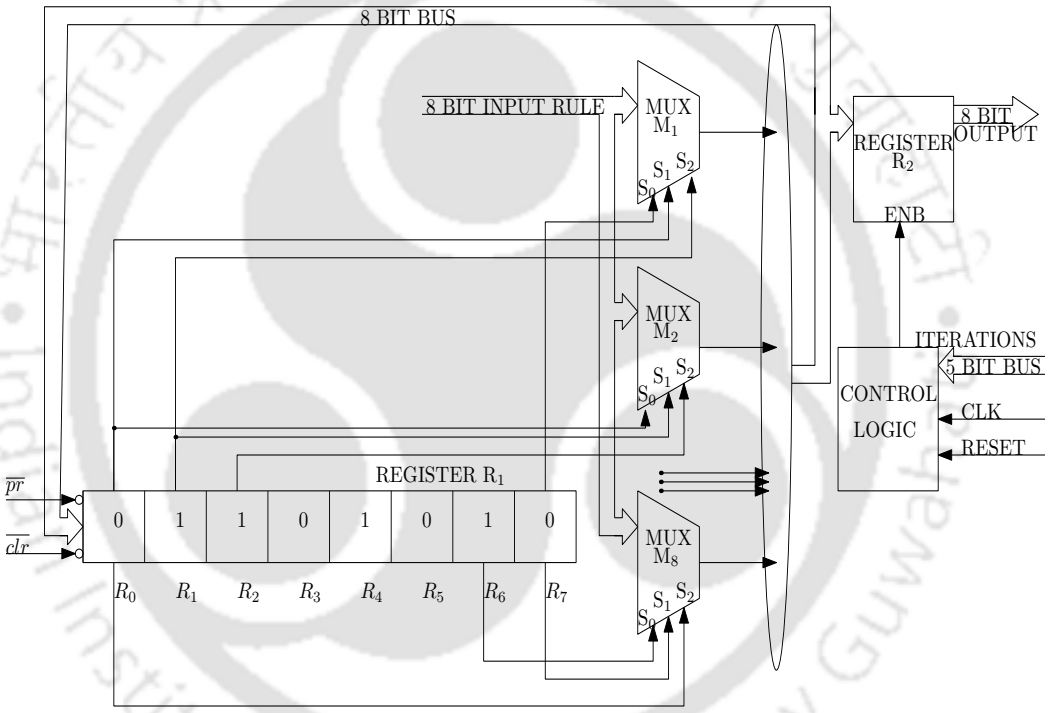

Figure 4.3: Proposed PCA based S-Box

4.4 Performance comparison between conventional LUT S-Box and Dynamic PCA S-Box

In order to validate the proposed S-Box realization, the cryptographic properties such as SAC, CIB, NL and CIB, as discussed in Section 3.4, are obtained.

The values of SAC for the proposed S-Box have been plotted in Figure 4.4. If the value

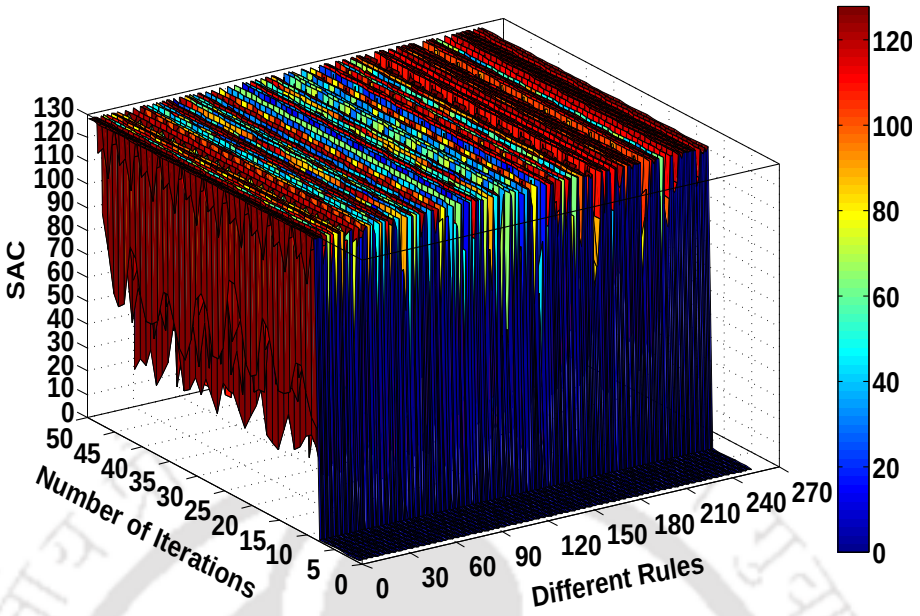

Figure 4.4: Value of SAC with Different Rules

of SAC is less for the observed ciphers, the cipher is secure against unauthorized attack. The achieved value of SAC ranges between [0, 128] and the best value observed is 14 for more than 26% of rules, which is clear from Figure 4.4. The PCA S-Box in terms of SAC gives better performance than that of classical S-Box.

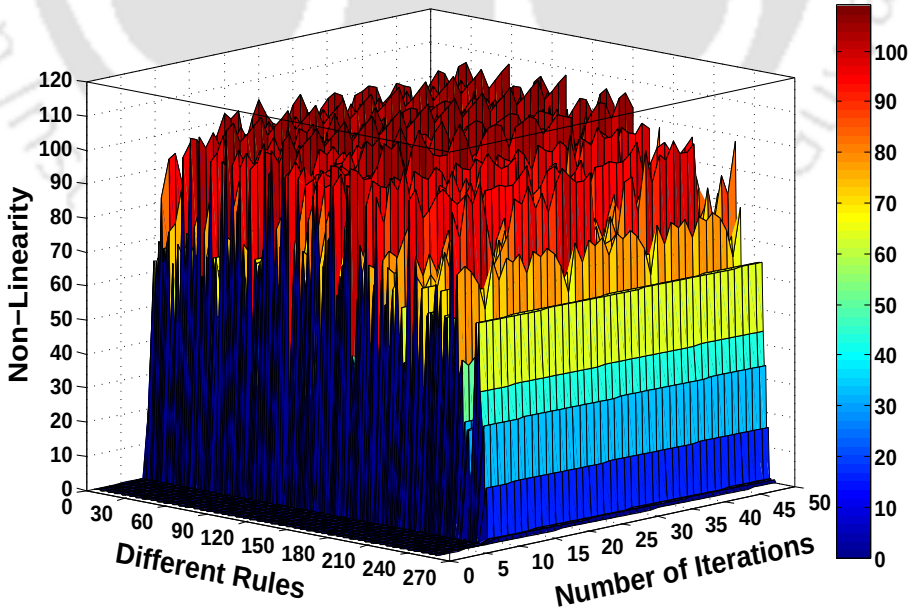

Figure 4.5: Non-Linearity with 256 Rules

The achieved values of NL for the proposed S-Box have been plotted in Figure 4.5. If the achieved value of NL for the observed cipher is significantly high, then the cipher is secure against cryptanalysis. It has been observed that the value of NL varies from [0, 109]. We have found that the achieved value of NL is more than 100 for 15% of 256 CA rules as shown in Figure 4.5, which indicates that the performance of PCA S-Box is comparable to that of classical LUT based S-Box.

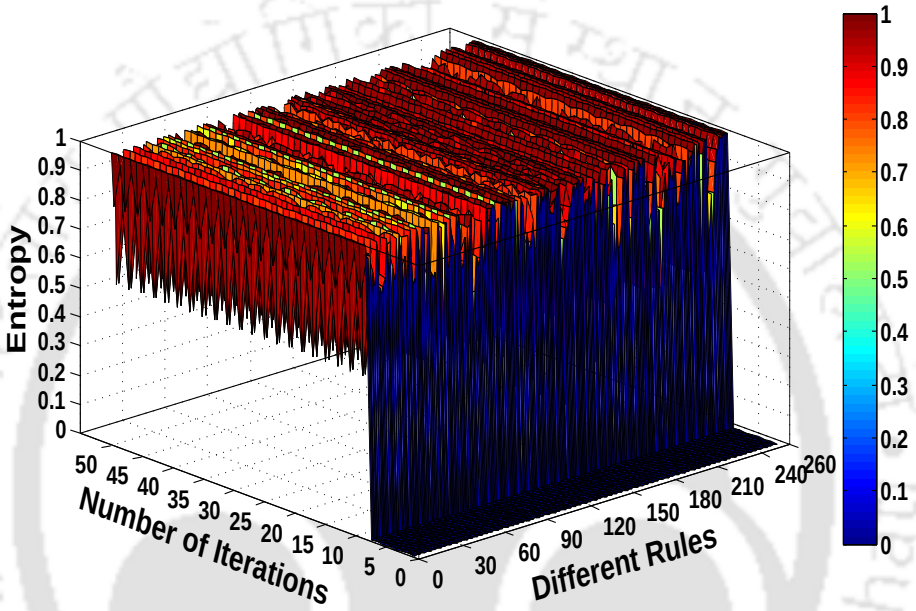

Figure 4.6: Entropy with Different Rules

The observed values of entropy for the proposed S-Box have been plotted in Figure 4.6. If the entropy value of cipher is high, then the cipher is difficult for cryptanalysis. The entropy value observed for PCA S-Box ranges from [0, 1], the best value attained is 0.99 and the values for most of the CA rule ranges between 0.95 and 0.99, shown in Figure 4.6. The achieved values for conventional S-Box are also been presented in Table 4.2. The performance of PCA based S-Box with respect to entropy is better than that of classical S-Box. If the value of CIB for cipher is less, then the cipher is more secure against cryptanalysis. The observed values of CIB for PCA S-Box varies between 0 to 128, the best value achieved is 0 and the values less than 14 are for 23% of 256 CA rules, as shown in Figure 4.7. The noticed value of classical S-Box is 14 as indicated in the Table 4.2. From the above observation, the PCA based S-Box provides much improved performance

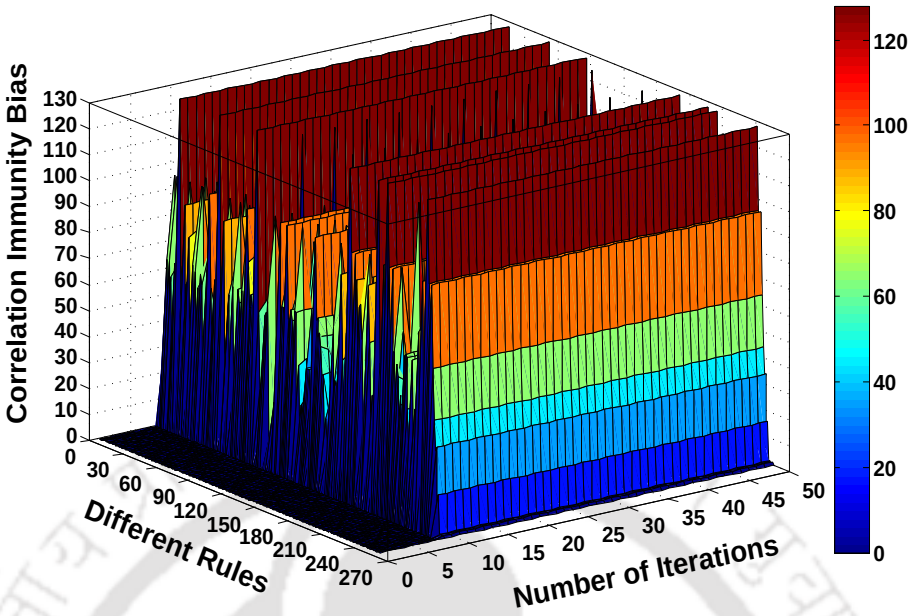

Figure 4.7: CIB with Different Rules

compared to the classical S-Box. The comparative performance of proposed PCA based S-Box and conventional LUT based S-Box in terms of level of security has been shown in Table 4.2. The PCA based S-Box is dynamic in nature and has been found to provide enough level of security, compared to LUT based S-Box and existing works [84–87].

4.4.1 Architectural Design

In order to validate the proposed architecture, AES algorithm with PCA based S-Box has been implemented using verilog, verified on FPGA board and synthesized with Cadence RTL compiler. The proposed architecture has been implemented on hardware using TSMC 0.18-μm technology (core voltage of 1.62 V) and UMC 0.13-μm technology (core voltage of 1.08 V) under worst-case conditions. The total time consumed to encrypt 128 bits of plain text is calculated by $Latency = Clockcycles \times Timeperiod$. The performance comparison of AES with PCA based S-Box and AES with LUT based S-Box are presented in Table 4.3, in terms of area, power dissipation, energy consumption and operating frequency. It can be noted that in our proposed PCA based S-Box realization, the number of iterations considered are 20 clock cycles. The total time taken to encrypt 128 bits of plain text using AES algorithm with PCA based S-Box is 244 clock cycles. The

Table 4.2: CIB, SAC, NL, Entropy values for PCA based S-Box and Standard AES S-Box using cryptographic properties

Rule No.	No of time Steps	Non-Linearity	Entropy	CIB	SAC
30	9	102	0.98	16	16
30	15	105	0.99	13	14
30	25	108	0.98	20	16
30	48	106	0.99	8	20
45	14	102	0.99	10	20
45	14	102	0.99	10	20
57	16	104	0.98	20	12
57	34	108	0.98	18	20
57	**21**	**106**	**0.99**	**14**	**12**
75	14	102	0.99	10	20
86	**25**	**108**	**0.98**	**20**	**16**
86	38	102	0.99	8	20
86	48	106	0.99	8	20
89	14	102	0.99	10	20
99	**22**	**109**	**0.98**	**18**	**20**
101	14	102	0.99	10	20
135	25	108	0.98	20	16
135	48	106	0.99	8	20
149	24	105	0.99	13	14
149	38	102	0.99	8	20
Hussain et.al [84]	NA	105 96	NP NP	NP NP	16 10
Clark et.al [85]	NA	90 100	NP NP	19 24	44 48
Millan et.al [86]	NA	80	NP NP	NP NP	16 18
Nedjah et.al [87]	NA	70 102	NP NP	NP NP	NP NP
Standard AES S-Box	Polynomial $x^8+x^4+x^3+x+1$	112	0.98	16	14

* NA means not applicable

* NP means not provided

number of gates utilized for LUT based S-Box and CFA based S-Box realization are 696, 294 respectively with 0.11-μm shown in Table 4.4, whereas in the case of proposed PCA based S-Box realization, the number of gates utilized are 113, 116 using TSMC 0.18-μm and UMC 0.13-μm technology libraries respectively.

Table 4.3: Hardware results of the Proposed AES algorithm with PCA based S-Box

AES	Tech	Area (mm^2)	Gates	Power (mW)	Frequency (MHz)	Clock cycles	Energy (nJ)
Kim [88]	0.25μm	NP	4000	0.02	0.1	870	174
Eslami [89]	0.18μm	2.25	NP	7.55	13.56	248	138
Manoj [90]	0.18μm	NP	NP	0.0512	1	500	25.60
Kaps [91]	0.13μm	NP	4070	0.0238	0.5	534	24.56
Proposed AES algorithm	0.18μm	0.184	4547	3.259	13.69	244	58.702
Proposed AES algorithm	0.13μm	0.069	3971	1.02	13.69	244	18.275

* NP means not provided

Table 4.4: Hardware results of the Proposed AES algorithm with PCA based S-Box

	Tech	Area (mm^2)	Gates	Power (μW)	Frequency (MHz)	Clock cycles	Energy (nJ)
A. Satoh [92] LUT based S-Box	0.11μm	NP	696	NP	NP	NP	NP
Sumio [93] LUT based S-Box	0.13μm	NP	712	29	10	NP	NP
A. Satoh [92] CFA S-Box	0.11μm	NP	294	NP	NP	NP	NP
Proposed PCA S-Box	0.13μm 0.18μm	NA	116 113	10	10	20	0.020

* NA means not applicable

* NP means not provided

Sumio et.al. [93], presented optimized low power S-Box architecture for AES which consumes power of 29 μW at 10 MHz using 130 μm CMOS technology, whereas the proposed PCA based S-Box at 10 MHz using 130 μm CMOS technology consumes power of 10 μW.

It can be easily seen that the proposed PCA based S-Box consumes 65% less power than the existing work [93]. The architecture of AES algorithm implemented using 0.18-μm technology consumes a power consumption of of 7.55 mW [89], whereas the proposed AES with PCA based S-Box at 13.69 MHz clock frequency requires power of 3.259 mW. The ASIC implementation of AES algorithm with CFA based S-Box using 0.18-μm technology takes 500 clock cycles to complete encryption of 128 bits plain text, when operated at 1 MHz frequency the power consumption is 51.20 μW [90]. The power dissipation, energy consumption of AES algorithm with proposed PCA based S-Box if operated at 1 MHz frequency using 0.18-μm technology consumes power, energy of 94.07 μW, 22.95 nJ respectively. Manoj et.al. reported energy consumption of 25.60 nJ, there is decrease in energy consumption of proposed AES algorithm with PCA based S-Box by 10% in comparison with the existing work [90]. It is clear from Table 4.3, that the proposed PCA based S-Box out performs in terms of power dissipation and energy consumption compared with the existing works [88–93].

4.5 Formulation of S-Box using 2^{nd} order reversible one dimensional cellular automata (RCA^2)

The basic structure of PCA and its functioning has been explained in Section 4.2. If the function is invertible then the rule is reversible which is a desired property in cryptography. We have observed that in PCA only 6 rules are reversible. In order to overcome this limitation of PCA, we have proposed a RCA^2 based architecture for S-Box which consumes low energy, can have 64 reversible rules and the mapping of Boolean functions is one on one. The structure of RCA^2 is slightly different as that of PCA, the results obtained with PCA is XORed with the previous value of the central cell i at time step $t-1$ in order to achieve the new content of RCA^2 at time step $t+1$. The next configuration of central cell C_i^{t+1} for RCA^2 at $t+1$ depends not only on present content of cell C_i^t but also its previous content C_i^{t-1}, as shown in Figure 4.8. Mathematically, the RCA^2 is represented by

$$C_i^{t+1} = (C_i^t \oplus C_i^{t-1}) \tag{4.2}$$

where $(C_i^t, C_i^{t-1}) = (K_i^{t+1}, K_i^{t-1})$ respectively at discrete time step $t+1$ and $t-1$.

4.6 Proposed RCA^2 based S-Box

The hardware realization of the proposed RCA^2 based architecture for S-Box consumes low energy. The traditional LUT based S-Boxes are rigid in nature, whereas, the proposed RCA^2 S-Box is dynamic in nature. The output of S-Box is a function of input rule which is programmable. The basic function of RCA^2 based S-Box is to transform 8 bits of input data into another secret data, achieved using a combinational logic, as shown in Figure 4.8. The initial 3 bits are loaded into the register R_1, R_2 using preset and clear signals.

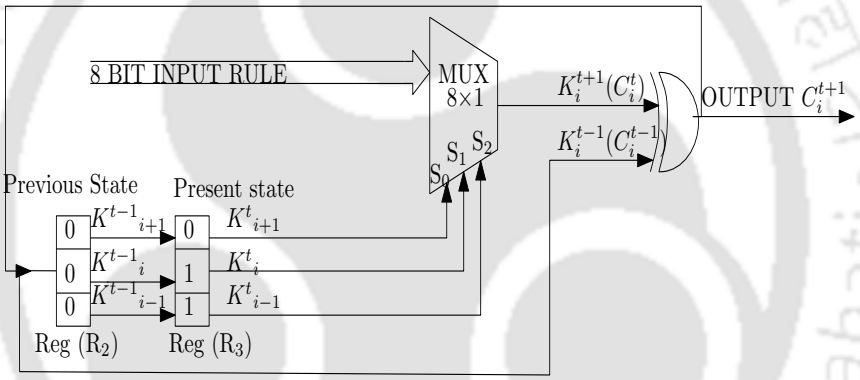

Figure 4.8: Basic cell structure of RCA^2

The output of C_i^{t+1} depends on the current C_i^t and the previous state C_i^{t-1} of cell C_i^t, where as $K_i^{t+1} = C_i^t$ and $K_i^{t-1} = C_i^{t-1}$, as depicted in Figure 4.8. The select lines of the multiplexer are activated and deactivated according to the control signals $K_{i-1}^t, K_i^t, K_{i+1}^t$ and the output of the multiplexer is mapped according to the stored 8-bit rule in the register.

It is clear from Figure 4.8 that the proposed basic RCA^2 structure produces one bit output for a given 8 bit input rule. As the S-Box operates on 8-bit, eight basic RCA^2 structures, shown in Figure 4.8, need to be interconnected to obtain the final architecture. This interconnection can be implemented using logic gates, multiplexers and registers, as

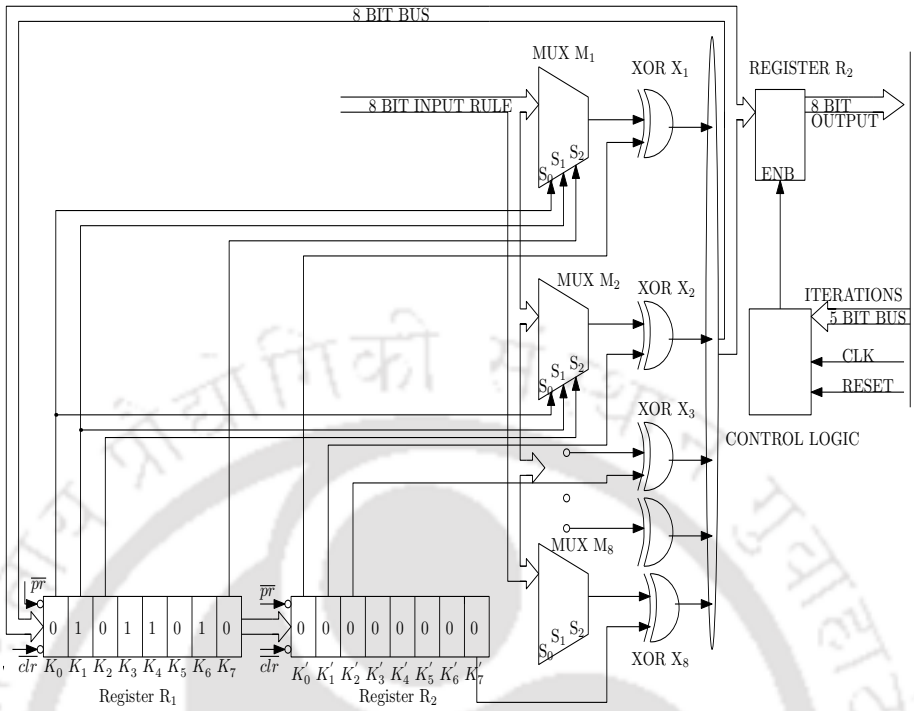

Figure 4.9: Proposed RCA^2 Based architecture

shown in Figure 4.9. Initially, the 8 bits are loaded into the RCA^2 array register R_1 using preset and clear signals. The input bits of register R_1 are connected as control signals to the 8:1 MUXes (M_1-M_8) in circular fashion whose input is an 8 bit rule. The 3 bits K_7, K_0 and K_1 are connected as control signals to the multiplexer M_1. For M_2 the control signals are K_0, K_1 and K_2. In a similar manner, the bits K_6, K_7 and K_0 are connected as control signals for M_8.

The previous value of bits K_0 to K_7 are stored in register R_2 as K_0' to K_7'. The output of MUX M_1 is XORed with previous bit K_0', output of MUX M_2 is XORed with K_1' and the output of MUX M_8 is XORed with last bit K_7'. The multiplexers produce outputs in accordance with Table 4.1. The multiplexer outputs are XORed with the contents of R_1 and R_2 to produce RCA^2 array bits which are used in subsequent iterations. The control logic comprises of a 6-bit up counter and a comparator. If the number of iterations in time step is equal to the count value of the counter, the control logic circuit output goes high to enable the register (R_3). The number of iterations defined in the control logic of RCA^2

determines the latency incurred in computing the S-Box. However, the RCA^2 based S-Box architecture shown in Figure 4.9 utilizes few logic gates in comparison with that of LUT based S-Box, shown in Table 4.6 and Table 4.7. It also consumes less power. The overhead incurred in the computation (number of time steps) using proposed RCA^2 based S-Box depends upon the specified number of iterations. However, as WBAN applications deal with low frequency biomedical signals, the process overhead incurred will not affect the overall performance of the system.

4.7 Security analysis of LUT and RCA^2 based S-Boxes

The level of security provided against cryptanalysis for the proposed RCA^2 S-Box has been observed using the cryptographic properties, namely, CIB, NL, SAC and entropy. If an S-Box satisfies these cryptographic properties, the S-Box is said to be secure against cryptanalysis, as discussed in Section 3.4. The observed values of SAC for the proposed

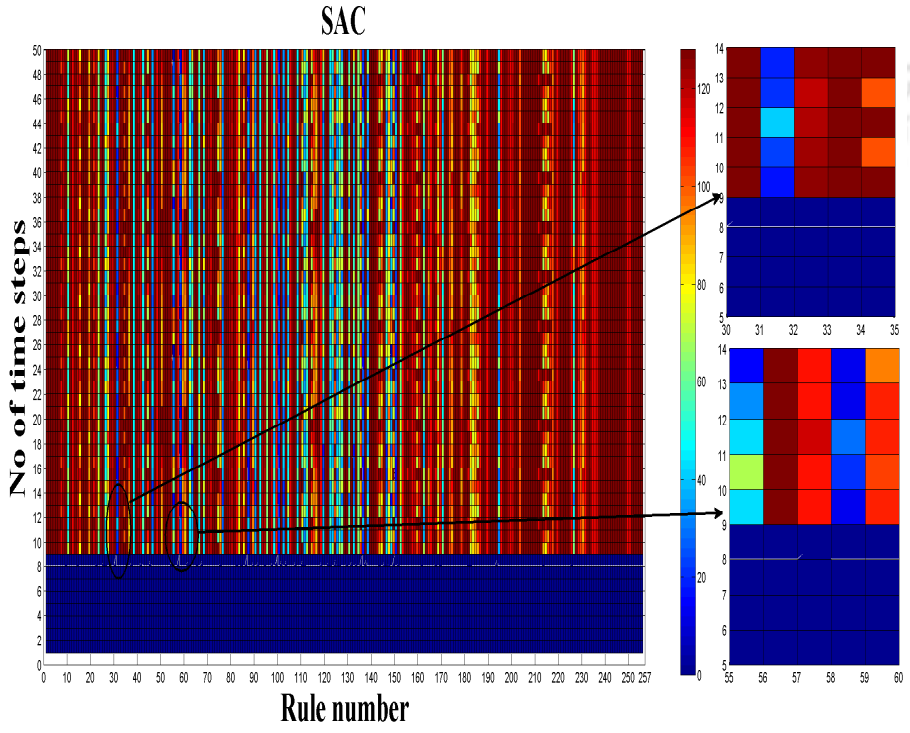

Figure 4.10: Values for SAC of RCA^2 based S-Box

RCA^2 S-Box have been plotted in Figure 4.10. It can be inferred that the value of SAC for

RCA^2 based S-Box is 16 for rule numbers 30, 57, 26, 99 and 135, as shown in Figure 4.10 and Table 4.5. It can be noted that the value is comparable with that of standard LUT based S-Box as shown in Table 4.5. Moreover, we have also found that 31.0323% out of 256 CA rules has SAC value of 16.

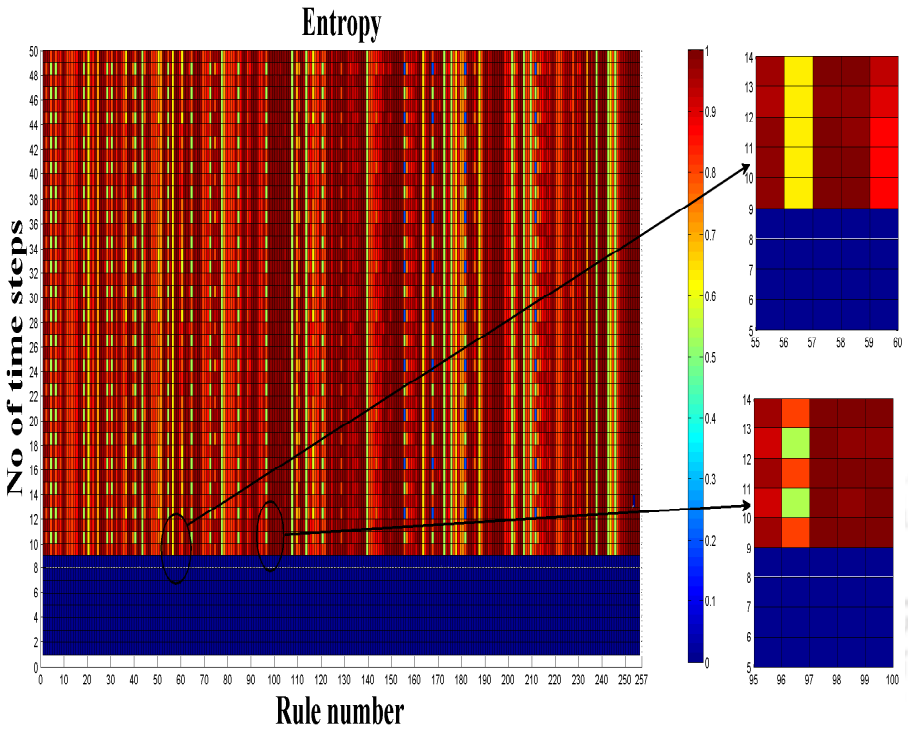

Figure 4.11: Values for Entropy of RCA^2 based S-Box

The values of entropy for the proposed RCA^2 S-Box have been plotted in Figure 4.11. The best values are achieved at rules 30, 57, 86, 99, 135 and 149, presented in Table 4.5. We have also found that in the proposed RCA^2 S-Box, 26.0323% out of 256 CA rules have better entropy values in comparison with the LUT based S-Box. The best value observed for entropy is 0.9914 at rule 57 as shown in Figure 4.11.

The values of NL for the proposed RCA^2 S-Box have been plotted in Figure 4.12. We have observed that the value of NL is high for rules 57, 99. It can be noted that 6.098% out of 256 CA rules have higher value of NL as shown in Figure 4.12. The maximum value of NL attained is 106 in case of RCA^2 based S-Box, as shown in Table 4.5.

The values of CIB for the proposed RCA^2 S-Box have been plotted in Figure 4.13.

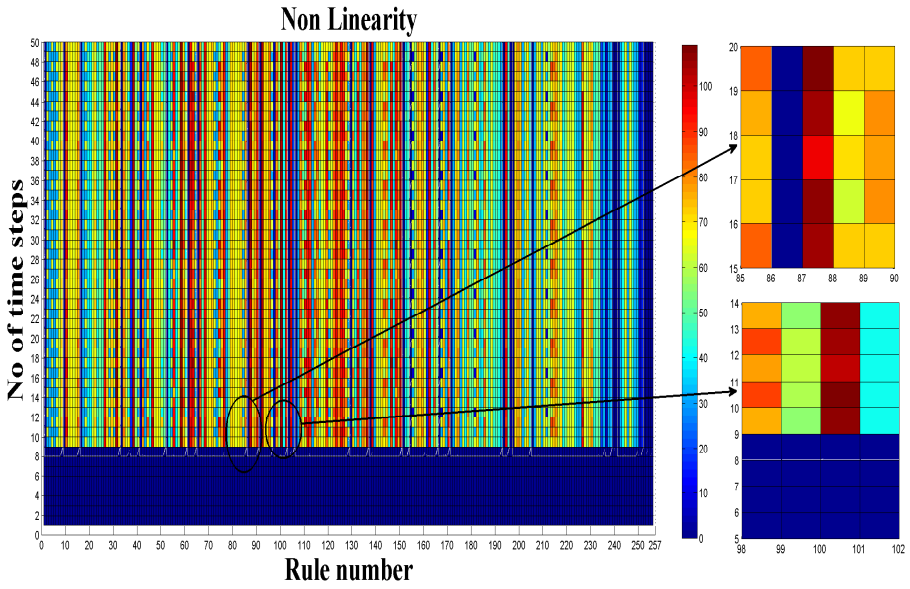

Figure 4.12: Values for Non Linearity of RCA^2 based S-Box

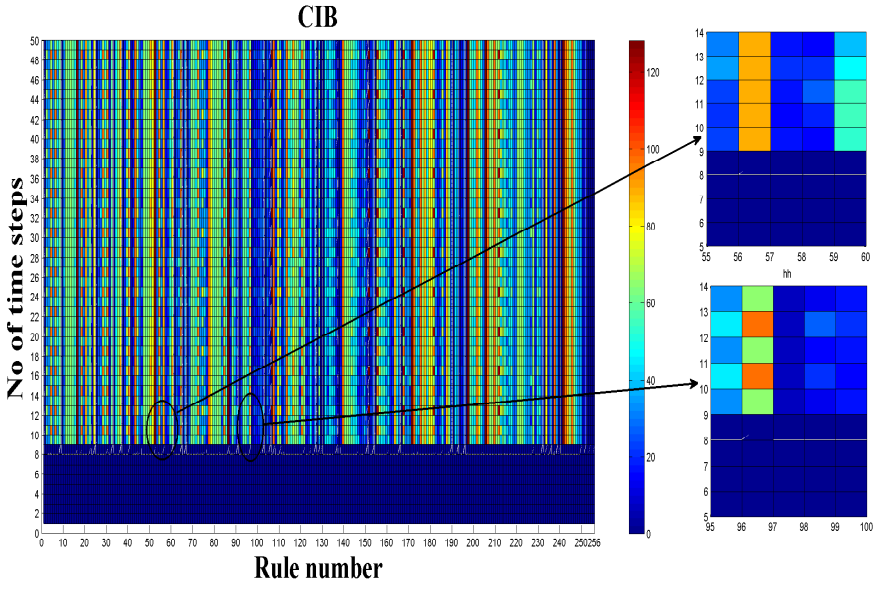

Figure 4.13: Values for Correlation Immunity Bias of RCA^2 based S-Box

It can be observed that the best values of CIB could be obtained at rules 57 and 99, presented in Table 4.5. The best observed value is 14 at rule 99. About 36.3548% out of 256 RCA^2 rules have better values of CIB, as shown in Figure 4.13.

Moreover, we have observed the results for RCA^2 based S-Box using 256 different rules. The value of SAC, CIB, NL and entropy of RCA^2 along with few reversible rules are shown in Table 4.5. The RCA^2 S-Boxes are dynamic in nature and more resistant to differential cryptanalysis as these provide enough level of security, compared to that of LUT based S-Box. A comparison of cryptographic properties for standard LUT based S-Box and proposed RCA^2 S-Box with existing works [84–87] has been shown in Table 4.5 .

Table 4.5: Cryptographic Properties values for RCA^2 based S-Box

Rule No	Time Step	NL	Entropy	CIB	SAC
30	15	100	0.98	16	16
57	**9**	**106**	**0.99**	**14**	**16**
86	10	100	0.98	16	16
99	**9**	**106**	**0.99**	**14**	**16**
135	13	100	0.98	18	16
149	46	100	0.98	16	16
169	29	100	0.99	15	16
225	12	101	0.98	19	16
Hussain et.al [84]	NA	105 96	NP NP	NP NP	16 10
Clark et.al [85]	NA	90 100	NP NP	19 24	44 48
Millan et.al [86]	NA	80	NP NP	NP NP	16 18
Nedjah et.al [87]	NA	70 102	NP NP	NP NP	NP NP
Standard AES S-Box	**Polynomial** $x^8+x^4+x^3+x+1$	112	0.98	16	14

* NA means not applicable

* NP means not provided

Table 4.6: Hardware results of Proposed AES algorithm with RCA^2 based S-Box

AES	Tech	Gates	Power (mW)	Frequency (MHz)	Clock cycles	Energy (nJ)
Kim [88]	0.25μm	4000	0.02	0.1	870	174
Eslami [89]	0.18μm	NP	7.55	13.56	248	138
Manoj [90]	0.18μm	NP	0.0512	1	500	25.60
Kaps [91]	0.13μm	4070	0.0238	0.5	534	24.56
Proposed AES algorithm	0.18μm	4830	3.856	13.69	244	68.726
Proposed AES algorithm	0.13μm	4120	1.65	13.69	244	29.408

* NP means not provided

Table 4.7: Hardware results of the Proposed AES algorithm with RCA^2 based S-Box

	Tech	Area (mm^2)	Gates	Power (μW)	Frequency (MHz)	Clock cycles	Energy (nJ)
A. Satoh [92] LUT based S-Box	0.11μm	NP	696	NP	NP	NP	NP
Sumio [93] LUT based S-Box	0.13μm	NP	712	29	10	NP	NP
A. Satoh [92] CFA S-Box	0.11μm	NP	294	NP	NP	NP	NP
Proposed RCA^2 S-Box	0.13μm 0.18μm	NA	136 124	14 15	10 10	20 20	0.028 0.030

* NA means not applicable

* NP means not provided

4.7.1 Architectural Design

The proposed RCA^2 based S-Box architecture with AES algorithm has been implemented using verilog and verified on FPGA board. The proposed architecture has been synthesized with Cadence RTL compiler at different operating clock frequencies using 0.18-μm technology (core voltage of 1.62 V) and 0.13-μm technology (core voltage of 1.08 V) under worst-case conditions. The performance of AES with RCA^2 based S-Box have been reported in Table 4.6 in terms of gate count, power dissipation, operating frequency and energy consumption. In the proposed RCA^2 based S-Box realization, the number of iterations considered to compute the RCA^2 S-Box is 20 cycles, a total of 244 clock cycles are required to compute proposed AES algorithm with RCA^2 S-Box. The proposed AES algorithm with RCA^2 based S-Box, when operated at 13.69 MHz clock frequency consumes 3.856 mW of power for encryption. The corresponding energy consumption is 68.726 nJ, which is 50% less compared to Eslami et.al. [89]. The traditional AES algorithm using CFA based S-Box on hardware consumes 500 clock cycles with power dissipation and energy consumption of 51.20 μW and 25.60 nJ respectively using 180-μm CMOS technology library [90]. Manoj et.al [90] presented AES algorithm with an energy consumption of 25.60 nJ, whereas, proposed RCA^2 S-Box architecture with AES algorithm requires power dissipation of 104.08 μW and energy consumption of 25.396 nJ at 1 MHz clock frequency. The proposed RCA^2 S-Box with AES architectures if operated at 1 MHz frequency, shows decrease in energy consumption, compared to Kaps et.al [91]. It is clear from Table 4.7, that the LUT based S-Box and the CFA based S-Box realizations on hardware consume 696, 294 gates respectively with 0.11-μm, whereas, the proposed RCA^2 S-Box realization require 136, 124 number of gates using 0.13-μm, 0.18-μm CMOS technology libraries respectively. The proposed RCA^2 S-Box when operated at 10 MHz using 0.13μm CMOS technology consumes power of 14 μW, whereas Sumio et.al. [93] presented a low power S-Box architecture with power consumption of 29 μW. It is clear from Table 4.6 that the proposed RCA^2 based S-Box for AES algorithm exhibits reduction in power dissipation and energy consumption in comparison with the existing works [88–93].

4.8 Conclusion

We have proposed an ultra low power, less area architecture for AES using PCA based S-Box and RCA^2 based S-Box for WBAN applications. In order to validate the proposed architectures, we have carried out both simulations and synthesis. Unlike [88,89,91,93,94], the proposed design require fewer logic elements, hence there has been reduction in power, energy and chip area compared to the conventional AES with LUT based S-Box. We have achieved comparable performance in terms of security for PCA based S-Box and RCA^2 based S-Box as that of classical LUT based S-Box using cryptographic properties. The PCA based S-Box and RCA^2 based S-Box for AES algorithm has been synthesized using Cadence RTL compiler to evaluate area, power and frequency of operation. In order to validate the simulation studies of proposed architectures with the existing works, the operating frequency taken into consideration is 13.69 MHz. However, WBAN applications operate at low frequency. The power dissipation, energy consumption of AES algorithm with proposed PCA based S-Box, when operated at 1 MHz frequency using 0.18-μm and 0.13-μm technology requires 94.07 μW, 89.06 μW respectively. It is clear, that the proposed PCA based S-Box performs much better in terms of power dissipation and energy consumption, compared to the existing works. The proposed RCA^2 based S-Box architecture with AES algorithm using 0.18-μm and 0.13-μm technology, if operated at 1 MHz consumes power of 104.08 μW and 96.87 μW respectively. Therefore, it has been observed that AES with PCA based S-Box and RCA^2 based S-Box are ultra low power and low energy consumption encryption algorithms and hence suitable for WBAN applications.

5

Encryption Algorithm using Hybrid Linear Cellular Automata and Hybrid Second order Reversible Cellular Automata

Contents

Objective In this chapter, we have presented encryption architectures using hybrid linear cellular automata (HLCA) and hybrid second order reversible cellular automata (HRCA^2). The proposed encryption architectures have been evaluated on hardware, in terms of power dissipation, energy consumption and frequency of operation. Moreover, the performance of proposed architectures concerning security, has been examined with cryptographic properties. The proposed architectures of HLCA and HRCA^2 algorithms have been designed and implemented on ASIC using UMC 0.13-μm and TSMC 0.18-μm CMOS technology libraries. The proposed HLCA and HRCA^2 architectures give better performance compared to that of traditional AES algorithm in terms of security. The proposed architectures have achieved low power dissipation and less energy consumption, hence suitable for WBAN applications.

5.1 Introduction

This chapter proposes low power encryption architectures using HLCA and HRCA^2. Like AES discussed in Section 3.2, the proposed HLCA and HRCA^2 algorithms also operate on symmetric key. However, the dynamic nature and ease of implementation of HLCA and HRCA^2 algorithms makes an efficient hardware realization in comparison with conventional AES algorithm. We have also examined the security provided by cipher text of proposed HLCA and HRCA^2 algorithms using cryptographic properties. The simulation results show that proposed hardware implementation could be achieved with low device utilization, thus consuming low power dissipation compared to the existing works. It has been observed that proposed HLCA and HRCA^2 algorithms are cryptographically secure against cryptanalysis compared to the conventional AES algorithm.

5.2 Proposed Hybrid Linear Cellular Automata (HLCA) based Encryption Algorithm architecture

The standard AES architecture inevitably expends more power and energy. In order to overcome this limitation of AES algorithm, we have developed an encryption algorithm using HLCA which consumes less energy. The output cipher text of HLCA encryption

algorithm is a function of input rule vector and hence the HLCA algorithm is dynamic in nature.

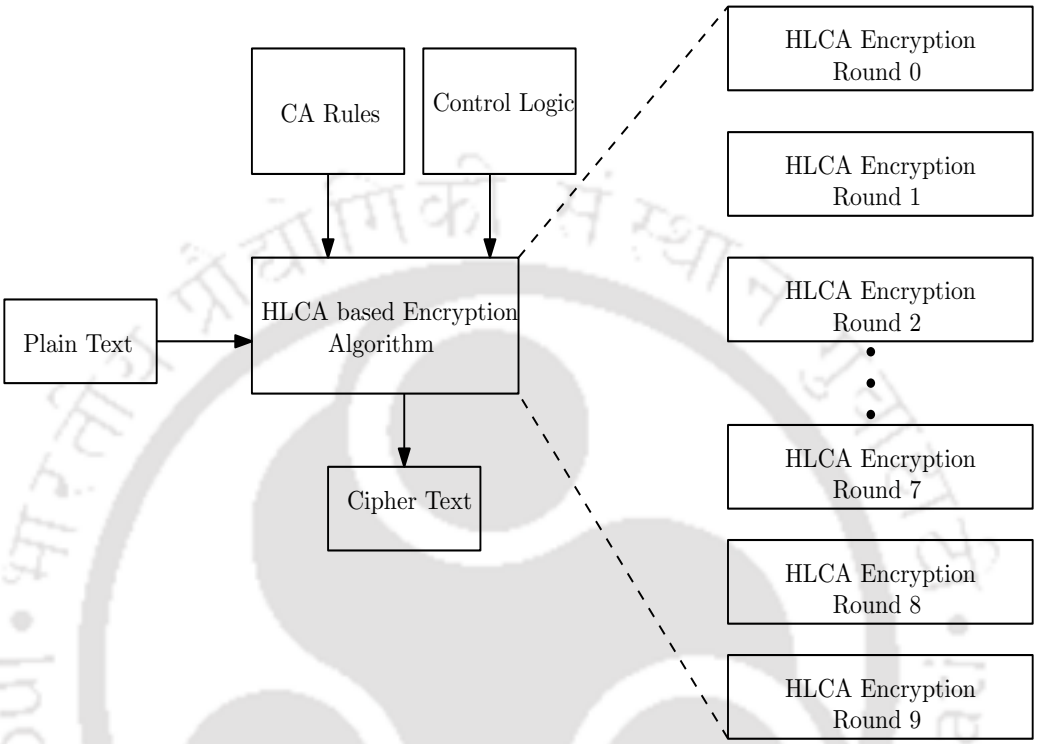

Figure 5.1: Block Diagram of proposed HLCA algorithm

As discussed in Section 3.2, the standard AES algorithm has 10 rounds of transformations for 128 bits secret key. The block diagram of proposed HLCA based algorithm shown in Figure 5.1 consists of 10 rounds of transformations similar to the conventional AES algorithm. Each round of transformation is iterated for 44 clock cycles with a different rule vector.

As described in Section 4.2, a lattice structure consists of 8 cells. The next state of central cell at discrete time step $t+1$ rely on the central cell i and neighboring cells $i+1, i-1$ with a CA rule at time t. Mathematically, T_i^{t+1} can be written as

$$T_i^{t+1} = K_p(T_{i-1}^t, T_i^t, T_{i+1}^t) \tag{5.1}$$

The basic purpose of the proposed HLCA algorithm is to transform 128 bits input plain

text to 128 bits output cipher text. The basic architecture of HLCA algorithm shown in Figure 5.2, has one bit output for 8-bit input rule. The final HLCA architecture has been achieved by interconnecting 128 basic HLCA structures, as shown in Figure 5.2. It produces 128 bits output cipher text. Using clear and preset signals, the 128 bits input plain text is loaded into the register K_1. The control logic block of HLCA algorithm consists of comparator and 6-bit up counter, which are used to regulate the number of iterations in each round of encryption process, shown in Figure 5.3. The output of T_i^{t+1} depends on 8 bits rule vector, previous state of cell T_i^{t-1} and the current state of cell T_i^t as depicted in Figure 5.2. The multiplexer select lines get activated or deactivated depending upon the input plain text $T_{i-1}^t, T_i^t, T_{i+1}^t$. The 128-bit input plain text of the register K_1 are utilized as control signals to 8:1 MUX (M_0-M_{127}) in a circular manner, the input to the 8:1 MUX is an 8-bit CA rule. The 3-bit T_{127}, T_0 and T_1 perform as control signals to M_0. For M_1, the control signals are T_0, T_1 and T_2. For the last MUX M_{127}, the control signal are T_{126}, T_{127} and T_0. As a result, the latency acquired to compute (number of time

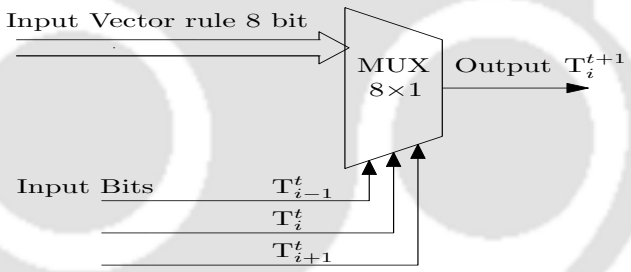

Figure 5.2: Basic HLCA cell structure

steps) each round of encryption process using HLCA algorithm depends upon the number of iterations defined in the control logic. Moreover, the number of iterations considered for each round of encryption process for HLCA algorithm is 44 clock cycles. Therefore, a total number of 440 clock cycles are used to compute 128 bits cipher text. However, the ASIC implementation of HLCA architecture depicted in Figure 5.3 utilizes fewer gates and lesser logic elements. The proposed architecture in comparison with AES algorithm consumes low power and hence the realization of proposed architecture on hardware is suitable for WBAN applications.

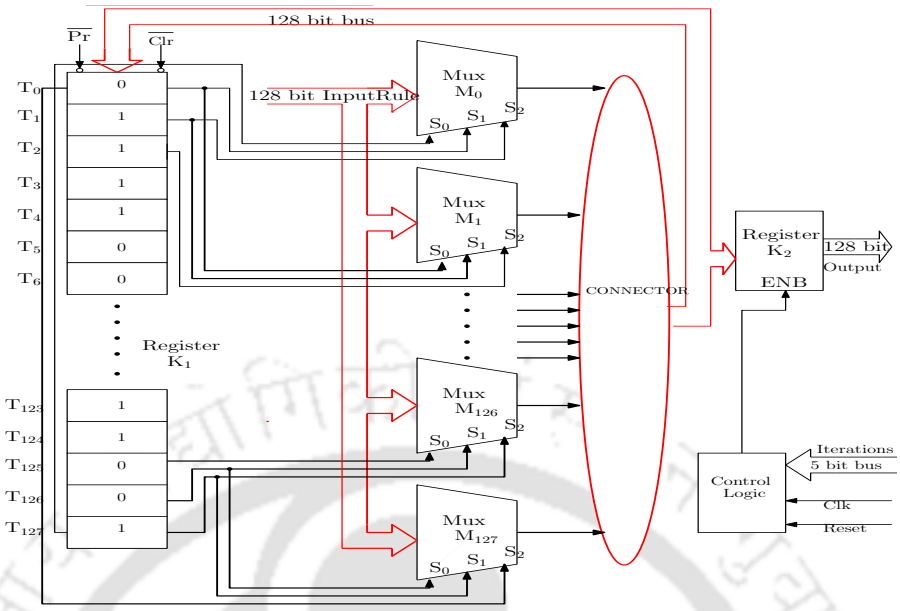

Figure 5.3: Architecture of proposed HLCA algorithm

5.3 Comparison of hardware architecture and cryptographic properties

The security provided by the proposed HLCA algorithm has been validated using cryptographic properties as discussed in Section 3.4. Moreover, to examine the level of security provided by the HLCA algorithm using cryptographic properties, the 128-bit input plain text with 256 different rule vectors are given to the FPGA board. The achieved output cipher text of HLCA algorithm from FPGA board is fed as input to MATLAB.

The values of SAC for the proposed HLCA based encryption algorithm have been plotted in Figure 5.4. The observed values of SAC vary between 0 and 128 and the best value is 14 for more than 26% of rules, as shown in Figure 5.4. The best values of SAC for few HLCA rules are shown in Table 5.1. The entropy values for proposed HLCA algorithm have been plotted in Figure 5.5. The best achieved values are at rules 55, 56, 57, 95 and 96, as shown in Figure 5.5. It can be noted that 36.54% of total 256 CA rules have better entropy values compared to the existing works. The best value of entropy obtained at rule 56 is 0.982 and for rule 95 is 0.991 as indicated in Figure 5.5 and Table 5.1.

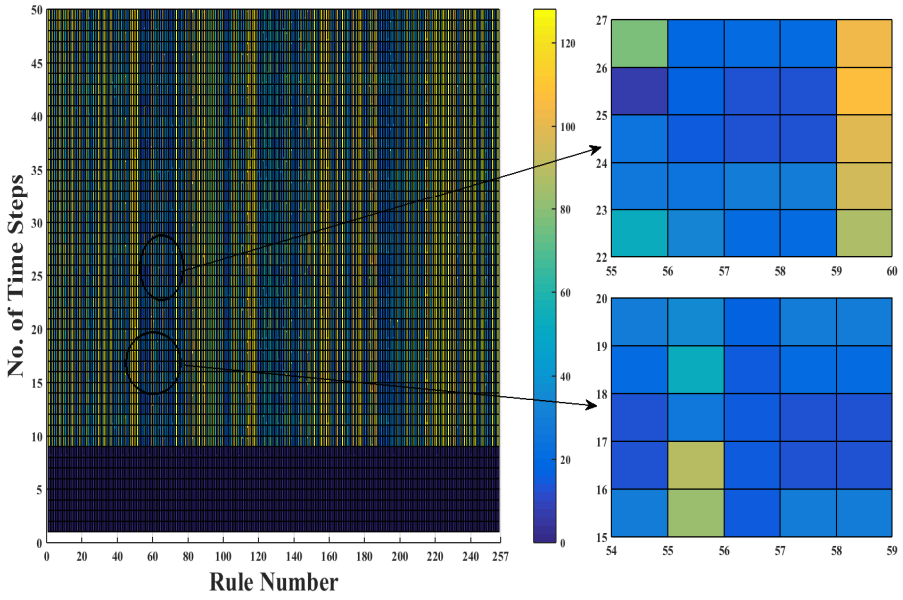

Figure 5.4: Results of SAC for HLCA based Encryption Algorithm

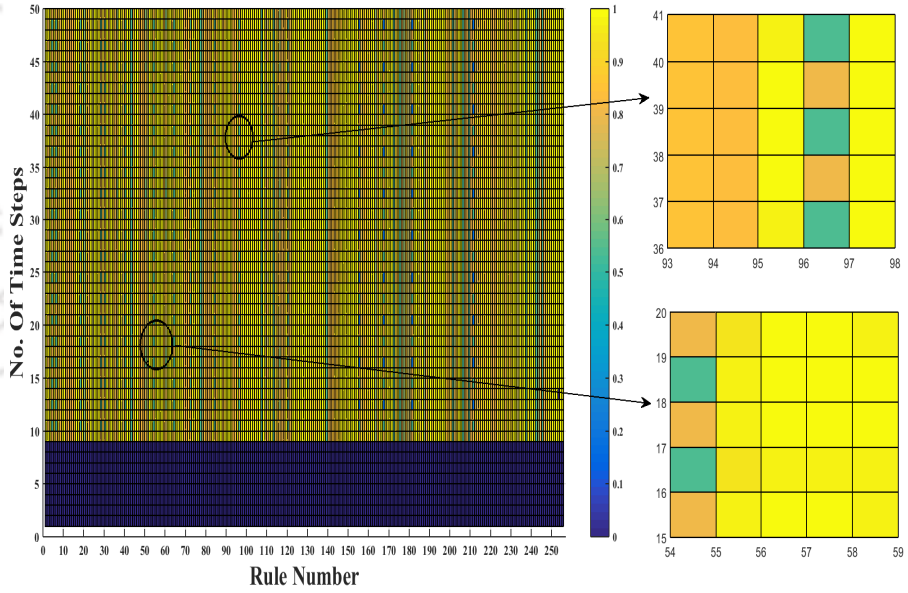

Figure 5.5: Values of Entropy for HLCA Algorithm

The NL values for the proposed HLCA algorithm have been plotted in Figure 5.6. It has been observed that the NL value is high for rules 57 and 95. It has been noticed, that the value of NL is high for 56.59% of total 256 CA rules, as shown in Figure 5.6. The best attained NL values are highlighted in Figure 5.6 and also shown in Table 5.1. The CIB

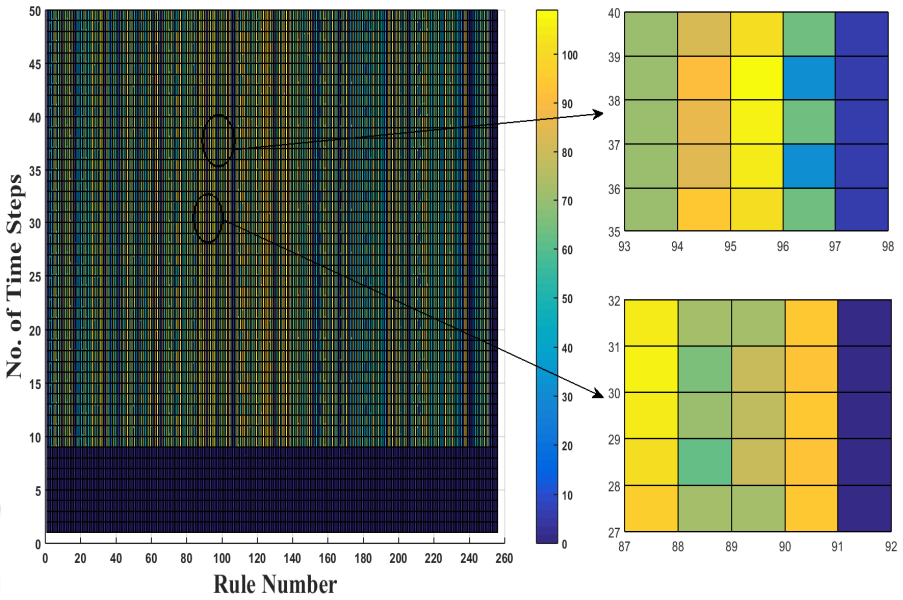

Figure 5.6: Value of NL for HLCA based Encryption Algorithm

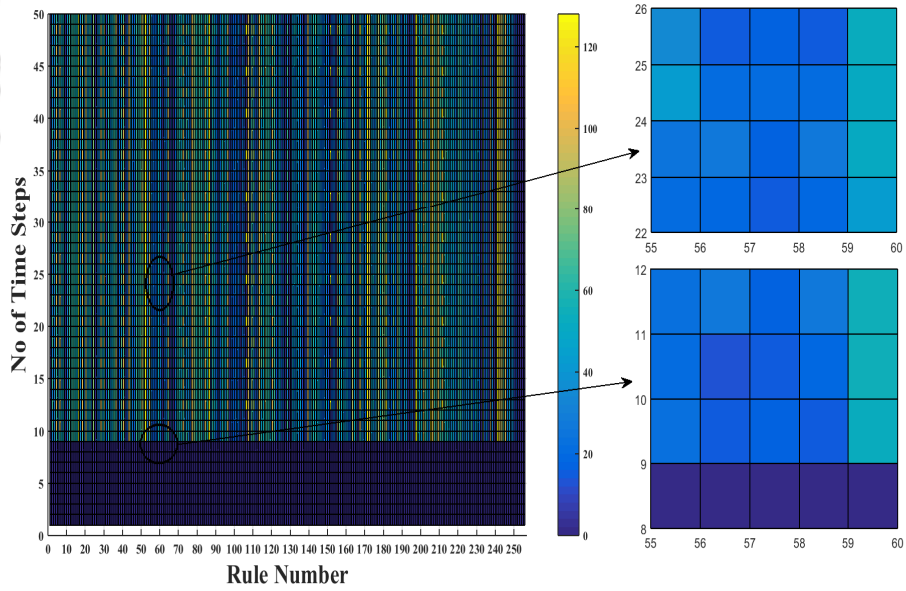

Figure 5.7: Values of CIB for HLCA based Encryption Algorithm

values for the proposed HLCA algorithm have been plotted in Figure 5.7. The achieved CIB values are highlighted in Figure 5.7 for rules 56 and 57. Out of 256 rules, the best values at rules 56, 57 and 96 are shown in Table 5.1. It has been observed that 66.45% of 256 rules have better CIB values, as depicted in Figure 5.7. It is clear from Table 5.1, that the value attained using cryptographic properties for HLCA 128 bits cipher text offer better security against cryptanalysis.

Table 5.1: Values achieved with Cryptographic Properties for cipher text of HLCA

Rule Vector	Proposed HLCA at Discrete Time Steps	NL	Entropy	SAC	CIB
55	10	102	0.91	12	18
56	18	104	0.98	16	10
57	24	**108**	**0.93**	**14**	**14**
57	25	**109**	**0.95**	**12**	**14**
90	30	**109**	**0.94**	**18**	**16**
90	34	105	0.97	16	18
95	38	**112**	**0.99**	**18**	**14**
95	40	98	0.95	14	18
96	45	101	0.98	12	14
Iqtadar [84]	NA	105 96	NP NP	NP NP	16 10
Clark [85]	NA	90 100	NP NP	19 24	44 48
Millan [86]	NA	80	NP NP	NP NP	16 18
Nedjah [87]	NA	70 102	NP NP	NP NP	NP NP
Standard AES algorithm	NA	112	0.98	16	14

* NA means not applicable

* NP means not provided

5.3.1 Comparison of architectures

The proposed HLCA architecture has been designed using Verilog and synthesis has been carried out using Xilinx tool. It has been implemented on Xilinx XC5VLX50-2FF676 FPGA. The device utilization and power consumption of proposed HLCA have been noted

in Table 5.2. It can be observed from Table 5.2, that the proposed HLCA algorithm exhibits better performance than the reported designs [95–98]. The throughput of the proposed HLCA algorithm can be calculated by

$$Throughtput = \frac{128\ bits \times Number\ of\ Clock\ cycles}{Clock\ Period} \tag{5.2}$$

Table 5.2: Comparison of FPGA Synthesis Results

	Proposed HLCA	[95]	[96]	[97]	[98]
FPGA	Xilinx	Xilinx	Xilinx	Xilinx	Altera
Device	XC5V LX50	XC5V LX50	XC5V LX50	XCV 600	APEX 20KC
Slices	148	303	399	1890	895
Flip Flops	396	922	NA	512	NA
Slice LUTs	412	564	1338	3645	NA
Throughput (Gbps)	1.60	1.33	4.34	0.352	1.188
Frequency (MHz)	498	425.46	339.1	140.39	120.65
Latency (ns)	50	96.35	29.5	363.3	107.7
Efficiency (Mbps/slice)	10.81	4.389	10.87	0.19	1.327
Power (mW)	0.660	NP	NP	NP	NP

* NP means not provided

In comparison to Deshpande et al. [95], the hardware utilization of the proposed HLCA algorithm on FPGA is reduced by 49%. There is 60% reduction in terms of slice utilization, in comparison to the work [96]. The proposed HLCA algorithm could achieve a throughput of 1.60 Gbps with 148 slices. The efficiency achieved by the proposed HLCA architecture is 10.81 Mbps/slice. Moreover, as the biomedical applications deal with low frequency of operation, the proposed HLCA algorithm if operated at low frequency there is enormous reduction in power consumption. Hence, the proposed HLCA algorithm is suitable for WBAN applications.

5.4 Architecture of proposed HRCA^2 based encryption algorithm

It has been noticed that HLCA is similar to PCA in which only 6 rules are reversible which makes decryption inefficient. To overcome this limitation, we have pro-

posed HRCA^2 encryption algorithm that is similar to RCA^2 architecture, which has 64 reversible rules. As explained in Section 4.5, the classification of CA is based on next state of the cell which relies on the present state and previous states of cells in an array. The HRCA^2 algorithm has been used to transform 128 bits plain text to cipher text. The block diagram of HRCA^2 based encryption algorithm has been shown in Figure 5.8. The next state of the central cell C_i^{t+1} of HRCA^2 at $t+1$ relies not only on the present cell C_i^t, but also on the previous cell C_i^{t-1}, shown in Figure 5.9. Mathematically, the HRCA^2 is expressed by

$$C_i^{t+1} = (C_i^t \oplus C_i^{t-1}) \tag{5.3}$$

where $(C_i^t, C_i^{t-1}) = (R_i^{t+1}, R_i^{t-1})$ at discrete time steps $t+1$ and $t-1$. The proposed HRCA^2 architecture has 10 rounds of transformations similar to standard AES algorithm as discussed in Section 3.2. The proposed HRCA^2 algorithm has been iterated for 44 clock cycles with different CA rules at each round of transformation.

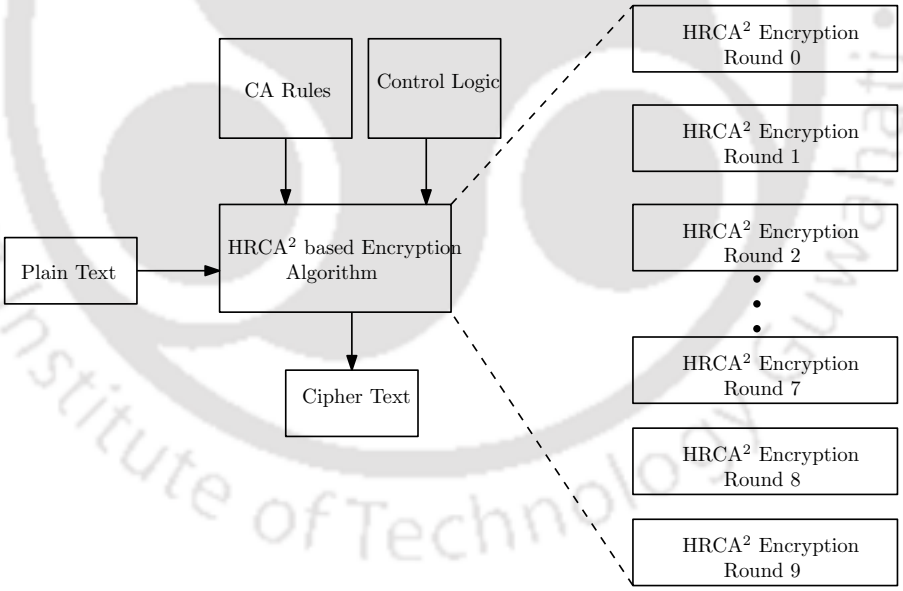

Figure 5.8: Block Diagram of proposed HRCA^2 algorithm

The proposed HRCA^2 basic structure for 8 bits input CA rule vector shown in Figure 5.9. In order to achieve 128-bit output cipher text, 128 basic HRCA^2 modules, depicted in Figure 5.9 are interconnected. The content of cell C_i^{t+1} depends on the current

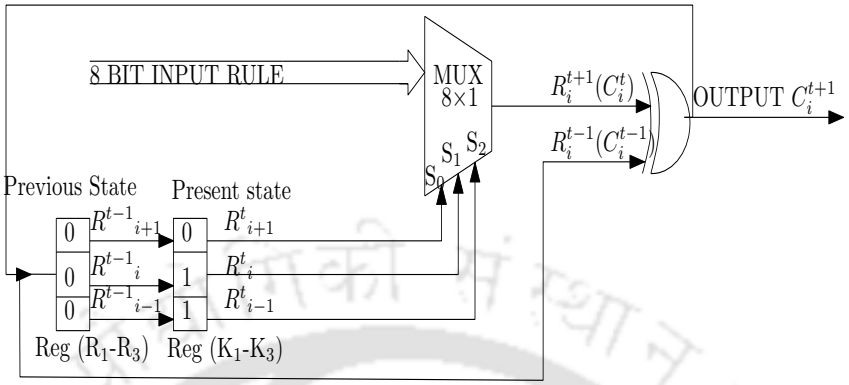

Figure 5.9: HRCA^2 basic cell structure

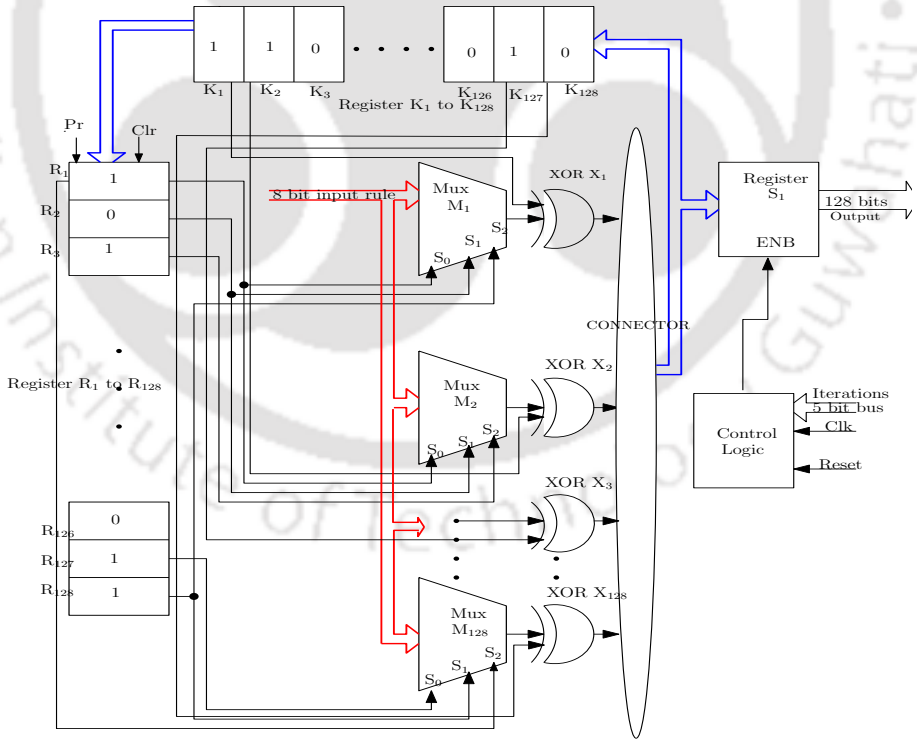

Figure 5.10: Architecture of proposed HRCA^2 algorithm

cell C_i^t and previous state of C_i^{t-1}, where $R_i^{t+1} = C_i^t$ and $R_i^{t-1} = C_i^{t-1}$, as shown in Figure 5.9. The proposed HRCA^2 encryption hardware requires a few logic gates, registers and multiplexers, as shown in Figure 5.10. Initially, the 128-bit plain text is loaded into register R_1 to R_{128}. The bits in register R_1 to R_{128} will function as control signals to 8:1 MUX (M_1-M_{128}). The bits in registers R_{128}, R_1 and R_2 act as control signals to M_1, R_1, R_2 and R_3 to M_2 and so on. For the last MUX M_{128}, the control signal are R_{126}, R_{127} and R_1. The output bits from X_1 to X_{128} are stored in registers K_1 to K_{128}. The previous bit of K_1 is XORed with the output of MUX M_1, K_2 is XORed with output of MUX M_2 etc, while the last bit K_{128} is XORed with output of MUX M_{128}. The XOR gates produce output bits which are used in subsequent iterations. The control logic of the HRCA^2 enables the register S_1, if the value of counter is equal to the number of time steps. The latency incurred in the encryption process is defined in the control logic by the number of iterations. Each round of transformation of HRCA^2 algorithm uses 44 clock cycles. A total of 440 clock cycles are required to compute 128 bits cipher text. The achieved output cipher using HRCA^2 varies with input rule vector and hence shows that the HRCA^2 algorithm output is dynamic in nature. However, the proposed HRCA^2 architecture simulation results show less hardware utilization and low power consumption. Apart from this, HRCA^2 architecture offers better level of security compared to the AES algorithm. Hence this proposed architecture is suitable for WBAN applications.

5.5 Comparison of Hardware Architecture and Security Analysis

The level of security provided by the proposed HRCA^2 algorithm against cryptanalysis has been examined using cryptographic properties as discussed in Section 3.4. The SAC values for the proposed HRCA^2 algorithm have been plotted in Figure 5.11. The SAC achieved values ranging from [0,128] and the best value observed is 12 for more than 21.50% of rules, which is clear from Figure 5.11. The best values of SAC for few reversible rules have been shown in Table 5.3. The entropy values for the proposed HRCA^2 based encryption algorithm have been plotted in Figure 5.12. It can be noticed that for proposed

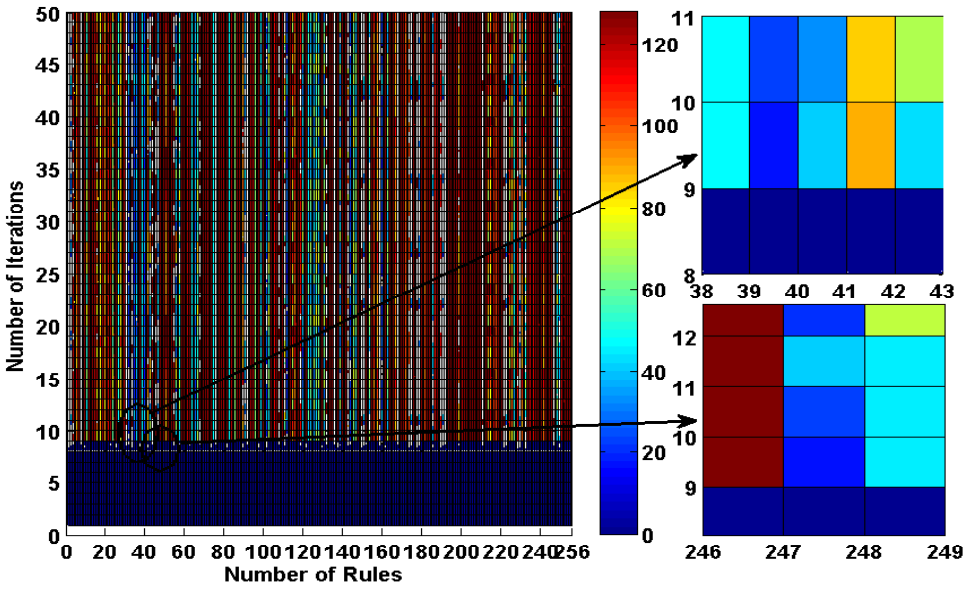

Figure 5.11: Results of SAC for HRCA^2 based Encryption Algorithm

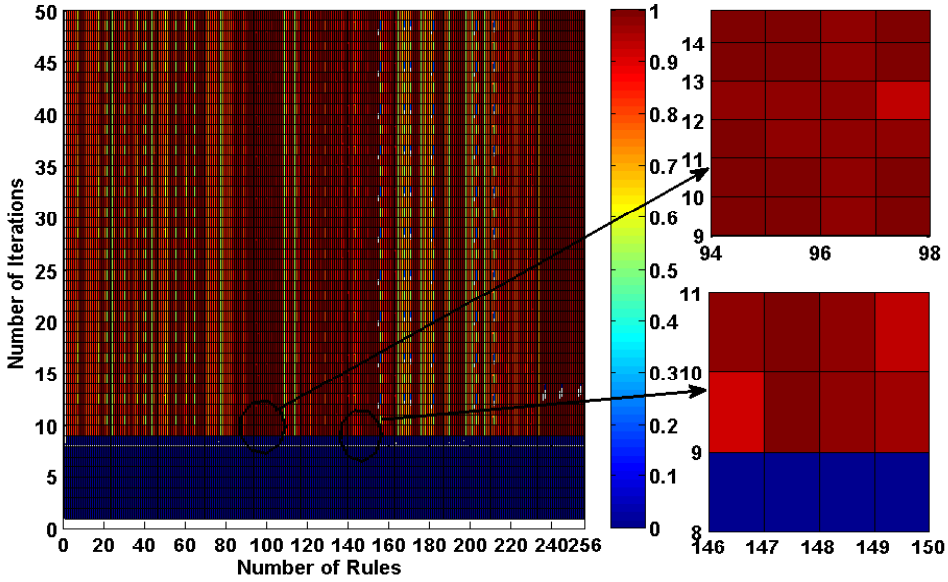

Figure 5.12: Values of Entropy for HRCA^2 Algorithm

HLCA^2 algorithm, 51.20% of total 256 CA rules have better entropy values in comparison with the entropy value of existing works. The best value of entropy obtained at rule 94 is 0.982 and at rule 146 is 0.991 as indicated in Figure 5.12 and Table 5.3. The NL values for

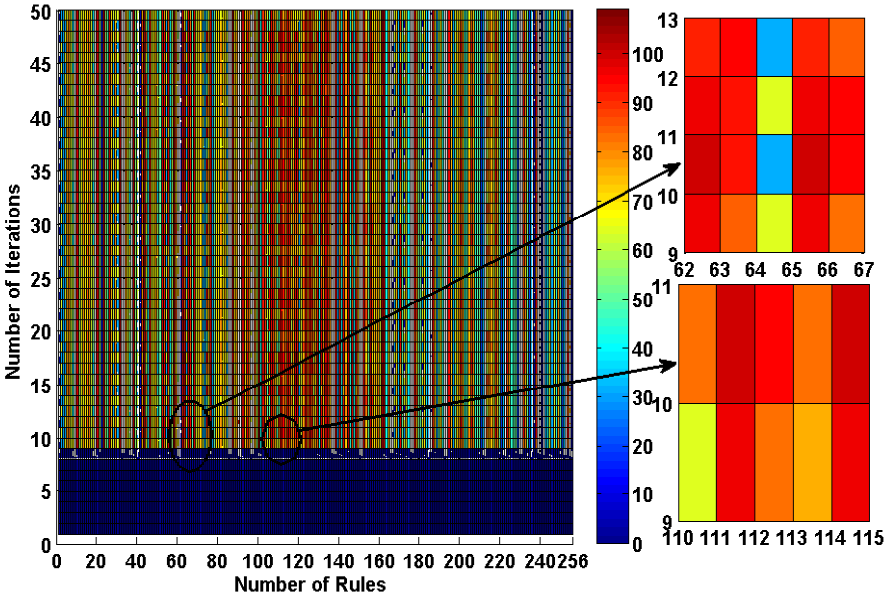

Figure 5.13: Value of NL for HRCA^2 based Encryption Algorithm

the proposed HRCA^2 based encryption algorithm have been plotted in Figure 5.13. We have observed that the NL value is high for rules 62, 66, 111 and 114. It has been noticed that the value of NL is high for 24.80% of total 256 CA rules, as shown in Figure 5.13. The attained NL values have been highlighted in Figure 5.13 and also shown in Table 5.3. The CIB values for the proposed HRCA^2 algorithm have been plotted in Figure 5.14. The achieved CIB values have been highlighted in Figure 5.14 at rules 73, 74, 75 and 99. Out of 256 rules, the best value is 12 at rule 62 and 65, as shown in Table 5.3. It has been observed that 34.56% of 256 rules have better CIB values as shown in Figure 5.14. The cipher text of HRCA^2 algorithm have been examined using cryptographic properties and the values have been presented in Table 5.3. The value of SAC, CIB, NL and entropy show that the cipher text generated by the HRCA^2 algorithm is secure against cryptographic attacks.

Table 5.3: Values achieved with Cryptographic Properties for cipher text of HRCA^2

Rule Vector	Proposed HRCA^2 at Discrete Time Steps	NL	Entropy	SAC	CIB
62	10	**111**	**0.91**	**12**	**12**
65	11	**110**	**0.98**	**16**	**12**
73	12	108	0.93	14	12
74	14	109	0.95	12	13
100	38	**112**	**0.99**	**18**	**14**
94	20	107	0.99	16	16
111	10	**110**	**0.94**	**18**	**16**
114	34	**110**	**0.97**	**16**	**18**
121	40	108	0.99	14	18
146	24	102	0.99	14	16
245	45	101	0.98	12	14
247	38	**112**	**0.99**	**14**	**14**
Iqtadar [84]	NA	105 96	NP NP	NP NP	16 10
Clark [85]	NA	90 100	NP NP	19 24	44 48
Millan [86]	NA	80	NP NP	NP NP	16 18
Nedjah [87]	NA	70 102	NP NP	NP NP	NP NP
Standard AES algorithm	NA	112	0.98	16	14

* NA means not applicable

* NP means not provided

Table 5.4: Hardware results of Proposed HRCA^2 based Encryption Algorithm

AES	Tech	Gates	Power (mW)	Frequency (MHz)	Clock cycles	Energy (nJ)
Kim [88]	0.25μm	4000	0.02	0.1	870	174
Eslami [89]	0.18μm	NP	7.55	13.56	248	138
Manoj [90]	0.18μm	NP	0.0512	1	500	25.60
Kaps [91]	0.13μm	4070	0.0238	0.5	534	24.56
Proposed HRCA^2 algorithm	0.18μm	3218	0.612	13.69	440	16.455
Proposed HRCA^2 algorithm	0.13μm	2342	0.280	13.69	440	8.999

* NP means not provided

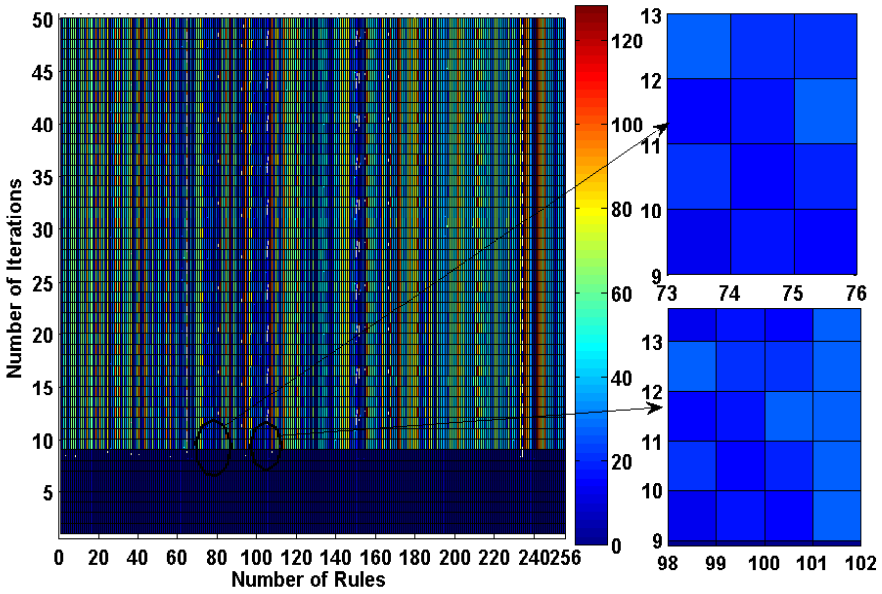

Figure 5.14: Values of CIB for HRCA^2 based Encryption Algorithm

Table 5.5: Comparison of FPGA Results

	Proposed HRCA^2	Proposed HLCA	[95]	[96]	[97]	[98]
FPGA	Xilinx	Xilinx	Xilinx	Xilinx	Xilinx	Altera
Device	XC5V LX50	XC5V LX50	XC5V LX50	XC5V LX50	XCV 600	APEX 20KC
Slices	172	148	303	399	1890	895
Flip Flops	412	396	922	NP	512	NP
Slice LUTs	425	412	564	1338	3645	NP
Throughput (Gbps)	1.34	1.60	1.33	4.34	0.352	1.188
Frequency (MHz)	454	498	425.46	339.1	140.39	120.65
Latency (ns)	50	50	96.35	29.5	363.3	107.7
Efficiency (Mbps/slice)	8.356	10.81	4.389	10.87	0.19	1.327
Power (mW)	0.824	0.660	NP	NP	NP	NP

* NP means not provided

5.5.1 Comparison of architectures

The proposed HRCA^2 algorithm architecture has been verified on FPGA using Xilinx ISE tool. Synthesis results of FPGA have been compared with the existing works shown in Table 5.5. The HRCA^2 has been designed using verilog and synthesized using TSMC 0.18-μm and UMC 0.13-μm technology libraries. The ASIC implementation results of HRCA^2 have been compared with the standard AES algorithm shown in Table 5.4. It has been observed from Table 5.4 and Table 5.5 that the proposed HRCA^2 architecture shows very low power dissipation and less energy consumption in comparison with the existing works [88–91]. It can be noted that the proposed HRCA^2 algorithm consumes 50% less energy consumption compared to that of Kaps et al. [91]. It is clear from Table 5.4 that the proposed HRCA^2 consumes 50% less energy, compared to the work presented in [91]. Eslami et al. in [89] has reported an energy consumption of 138 nJ, whereas the proposed architecture consumes energy of 16.455 nJ, which is very less compared to the reported work. It has been observed that the proposed HRCA^2 architecture consumes 50% less hardware, compared to works presented in [95,96]. The latency incurred in computing the cipher text using HRCA^2 algorithm is less in comparison to the existing designs [95,97,98].

5.6 Summary and conclusions

In this chapter, we have proposed HLCA and HRCA^2 encryption algorithms for cryptographic applications. It has been observed that the HLCA and HRCA^2 based encryption algorithms require less power consumption and attain high throughput, compared to the standard AES encryption algorithm. The proposed HLCA and HRCA^{2}algorithms have been simulated, implemented on FPGA XC5VL50 using verilog with Xilinx ISE tool and synthesized with Cadence RTL compiler. The cryptographic properties such as SAC, CIB, entropy and NL have been used to examine the security aspects of the cipher text generated using HLCA and HRCA^2 encryption algorithms. The security provided by the proposed HLCA and HRCA^2 have been compared with that of the conventional AES algorithm. The proposed HLCA algorithm when operated at 498 MHz frequency on FPGA achieved a high throughput of 1.60 Gbps and low power consumption of 0.660 mW. It can be noted

that in WBAN applications, the biomedical signals operate at low frequency with data rates of 10 bps to 10 Mbps. The proposed HRCA^2 encryption algorithm architecture if operated at 1 MHz using UMC 0.13-μm technology (core voltage of 1.08 V) and TSMC 0.18-μm technology (core voltage of 1.62 V) under worst-case conditions consumes power of 0.25 μW, 0.22 μW respectively. The proposed HRCA^2 algorithm architecture results show huge reduction in power dissipation and decrease in device utilization by 78% in comparison to the conventional AES algorithm. Hence, the proposed HLCA and HRCA^2 encryption algorithms are more suitable for WBAN applications.

6

Low power F function Architecture for Camellia Encryption Algorithm

Contents

Objective This chapter focuses on novel a approach to design low power architecture for F function, used in Camellia algorithm using RCA^2 and LPCA. The architecture results low energy consumption and low area utilization with minimum delay overhead. The security provided by the proposed RCA^2 based F function and LPCA based F function for Camellia algorithm have been been examined using cryptographic properties. The proposed F function architectures are evaluated in terms of power dissipation, energy consumption using TSMC 0.18-μm and UMC 0.13-μm technology libraries. Apart from that, it has been shown that proposed F function architectures are dynamic in nature, invertible, have low power dissipation compared with that of LUT based S-Box F function and hence applicable for WBAN applications.

6.1 Introduction

The Camellia algorithm is cryptographically more secure against differential cryptanalysis compared to AES algorithm [99]. The feistel network of Camellia makes the algorithm more resistant against unauthorized attacks. Unlike AES, the computation involved in S-Box of F function for Camellia algorithm are more complex and hence developing low power architecture to S-Box of F function in Camellia algorithm has been a challenging task. In this chapter, we have focused on RCA^2 and LPCA based low power architecture for F function in Camellia algorithm. Simulation studies show that the proposed architecture provides reduction in power dissipation and energy consumption, compared to the existing architectures. In order to check the level of security provided by the proposed architecture, we have obtained cryptographic properties, such as, NL, entropy, CIB and SAC. It has been observed that the proposed RCA^2 and LPCA F functions give similar performance in terms of security compared to LUT based S-Box of F function for Camellia algorithm. The reason for not using HLCA/HRCA^2 is that 440 clock cycles are required for HLCA/HRCA algorithm to compute 128-bit cipher text. If HLCA/HRCA is used to realize the F function for Camellia, the number of clock cycles required to compute the Camellia algorithm will increase enormously. This has the potential to increase the power consumption to the order of 0.1 W and hence not suitable for WBAN applications.

6.2 Architecture of Camellia algorithm

Camellia is a feistel network based block cipher algorithm which uses symmetric secret key for encryption and decryption of data. One of the important features of Camellia algorithm is that the same hardware architecture can be utilized for encryption as well as decryption.

6.2.1 Notations and symbols

$\oplus \cup \cap \rightarrow$ Bitwise exclusive-OR (XOR), AND and OR operation, respectively.

$\| \rightarrow$ Concatenation of two operands.

$\ggg_n$ Rotation to the right by n bits.

$\lll_n \rightarrow$ Rotation to the left by n bits.

$kw_{a(64)}$, $K_{b(64)}$, $kl_{c(64)}, \rightarrow$ Sub keys of 64 bits are used in the feistel rounds, where a = (1, 2, 3, 4), b = (1, 2, 3, 4, 5, . . 17, 18), c = (1, 2, 3, 4), the detailed discussion is presented in subsequent Section.

The secret key of Camellia encryption algorithm determines the number of feistel rounds (r). The Camellia encryption process with secret key size of 128, 192, 256 bits has 18, 24 and 24 feistel rounds respectively. The Camellia algorithm with secret key size of 128 bits is recommended by the latest IEEE 802.15.6 standard for WBAN applications [3, 7], which results in 18 feistel rounds of transformation. On the other hand, key lengths of 128 bits has 18 feistel rounds with 2 extra layers FL and FL^{-1} after every 6 feistel rounds as shown in Figure 6.1. The key lengths of 192 and 256 bits has $r = 24$ feistel rounds with 3 extra FL and FL^{-1} layers. For 128 bit secret key, the mathematical expression for $r =$

1 to 18 feistel rounds of transformation except for $r = 6$ and 8 can be expressed as.

$$L_r = R_{r-1} \oplus F(L_{r-1}, k_r), R_r = L_{r-1}. \tag{6.1}$$

For $r = 6$ and 8, the expression is as follows.

$$L_r^{'} = R_{r-1} \oplus F(L_{r-1}, k_r), R_r^{'} = L_{r-1}, \tag{6.2}$$

$$L_r = FL(L^r, kl_{\frac{r}{3-1}}), R_r = FL^{-1}(R_r^{'}, kl_{\frac{r}{3}}). \tag{6.3}$$

Initially, the plain text of 128 bits is divided into two sets L_0 and R_0, each of 64 bits. The two sets of 64 bit plain texts undergo XOR operation with subkeys $kw_{1(64)}, kw_{2(64)}, kw_{3(64)}, kw_{4(64)}$. These subkeys are generated in the key scheduling process using 128 bit secret key, as shown in Figure 6.3. The process of XOR operation with the sub keys before and after the round functions is called pre-whitening and post-whitening, respectively. After the pre-whitening phase, the bits are allowed to pass through the block containing 6 feistel rounds and followed by FL and FL^{-1} layer. The last 18^{th} feistel round of transformation has two sets R_{18} and L_{18} which are XORed with keys $kw_{3(64)}$, $kw_{4(64)}$ and concatenated to form cipher text (C_{128}) of 128 bits.

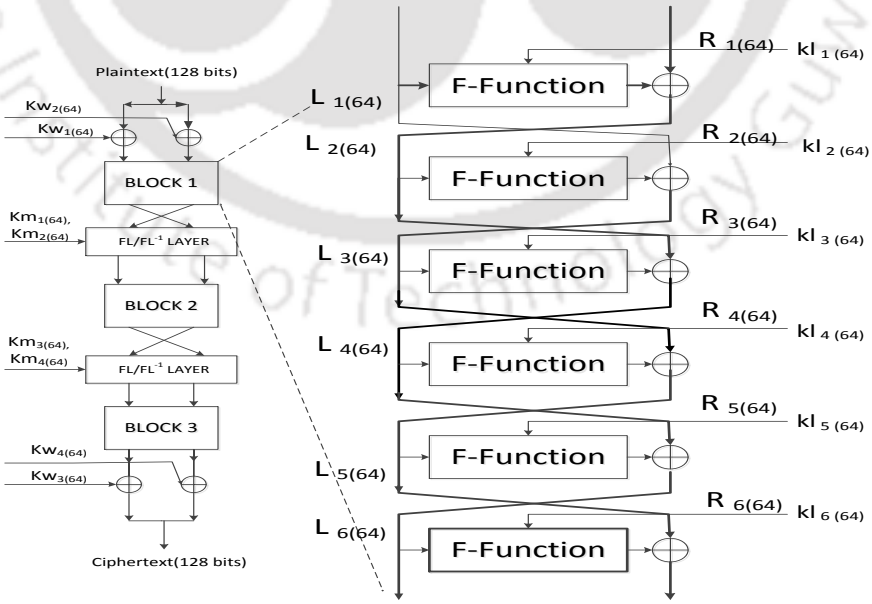

Figure 6.1: Camellia Encryption Process

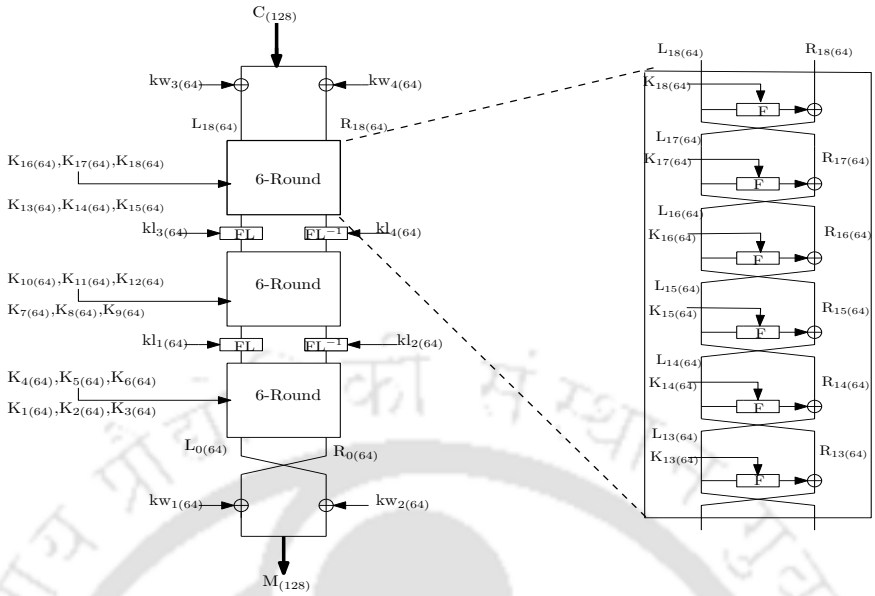

Figure 6.2: Camellia Decryption Process

In the decryption process of Camellia algorithm, initially, the cipher text is divided into two parts R_{18} and L_{18}, each of 64 bits. The 64 bit cipher text is XORed with subkeys kw_3, kw_4 and the output is passed through three blocks of feistel rounds of transformations as shown in Figure 6.2. The subkeys (kl_3, kl_4), (kl_1, kl_2) are used at 14 and 7 feistel rounds, respectively. Finally, the data is XORed with subkeys kw_2, and kw_1 as in post-whitening of encryption process and concatenated to form 128 bit plain text.

6.2.2 Key Scheduling Process

The subkeys $kw_{1(64)}, kw_{2(64)}, kw_{3(64)}, kw_{4(64)}$ are generated in the key scheduling process as shown in Figure 6.3. For a secret key of 128 bits, $K_{L(128)}$ is 128 bits secret key and $K_{R(128)}$ is 0. The keys $K_A(128)$, $K_B(128)$ are generated using $K_{L(128)}$ and $K_{R(128)}$ and the constants $\sum_{i(64)}$, where $i = (1, 2, 3, 4, 5, 6)$ are used as constants in key scheduling process as shown in Figure 6.3. These 64 bit constant values in hexadecimal representation are shown in Table 6.1. The constant hexadecimal value is calculated from second hexadecimal place to the seventeenth hexadecimal place of the hexadecimal representation of the square root of the i^{th} prime. The sub keys for encryption and decryption are obtained by shifting K_L and K_A by n bits. The shifting of subkeys varies in each round of encryption

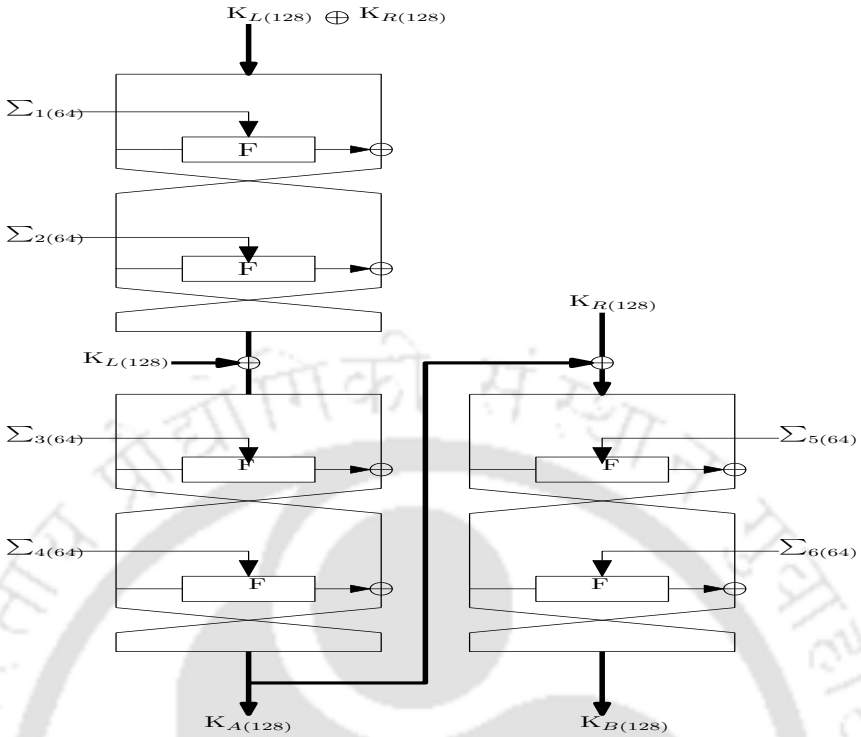

Figure 6.3: Key Scheduling Process

is shown in Table 6.2.

Table 6.1: Constant values of $\sum_{i(64)}$ for Key Scheduling

$\sum_1$	$A09E667F3BCC908B$H
$\sum_2$	$B67AE8584CAA73B2$H
$\sum_3$	$C6EF372FE94F82BE$H
$\sum_4$	$54FF53A5F1D36F1C$H
$\sum_5$	$10E527FADE682D1D$H
$\sum_6$	$B05688C2B3E6C1FD$H

6.2.3 FL and FL^{-1} Function

The function of FL and FL^{-1} layers is to keep the data more secure against attacks by lavishing arbitrariness to data as depicted in Figure 6.4. The FL and FL^{-1} layers are

Table 6.2: Value of SubKeys for 128 bits secret key

Process	Subkeys	Value
Prewhitening	kw_1	$(K_L \lll_0)_{L(64)}$
Prewhitening	kw_2	$(K_L \lll_0)_{R(64)}$
F (round 1)	K_1	$(K_A \lll_0)_{L(64)}$
F (round 2)	K_2	$(K_A \lll_0)_{R(64)}$
F (round 3)	K_3	$(K_L \lll_{15})_{L(64)}$
F (round 4)	K_4	$(K_L \lll_{15})_{R(64)}$
F (round 5)	K_3	$(K_A \lll_{15})_{L(64)}$
F (round 6)	K_4	$(K_A \lll_{15})_{R(64)}$
FL	$kl_{1(64)}$	$(K_A \lll_{30})_{L(64)}$
FL^{-1}	$kl_{2(64)}$	$(K_A \lll_{30})_{R(64)}$
F (round 7)	K_7	$(K_L \lll_{45})_{L(64)}$
F (round 8)	K_8	$(K_L \lll_{45})_{R(64)}$
F (round 9)	K_9	$(K_A \lll_{45})_{L(64)}$
F (round 10)	K_{10}	$(K_L \lll_{60})_{R(64)}$
F (round 11)	K_{11}	$(K_A \lll_{60})_{L(64)}$
F (round 12)	K_{12}	$(K_A \lll_{60})_{R(64)}$
FL	$kl_{3(64)}$	$(K_L \lll_{77})_{L(64)}$
FL^{-1}	$kl_{4(64)}$	$(K_L \lll_{77})_{R(64)}$
F (round 13)	K_{13}	$(K_L \lll_{94})_{L(64)}$
F (round 14)	K_{14}	$(K_L \lll_{94})_{R(64)}$
F (round 15)	K_{15}	$(K_A \lll_{94})_{L(64)}$
F (round 16)	K_{16}	$(K_A \lll_{94})_{R(64)}$
F (round 17)	K_{17}	$(K_L \lll_{111})_{L(64)}$
F (round 18)	K_{18}	$(K_L \lll_{111})_{R(64)}$
Postwhitening	kw_3	$(K_A \lll_{111})_{L(64)}$
Postwhitening	kw_4	$(K_A \lll_{111})_{R(64)}$

inserted at 6 and 12 feistel rounds of transformation as shown in Figure 6.1. The subkeys and data of 64 bits is subdivided into 32 bits then calculation is carried out as follows.

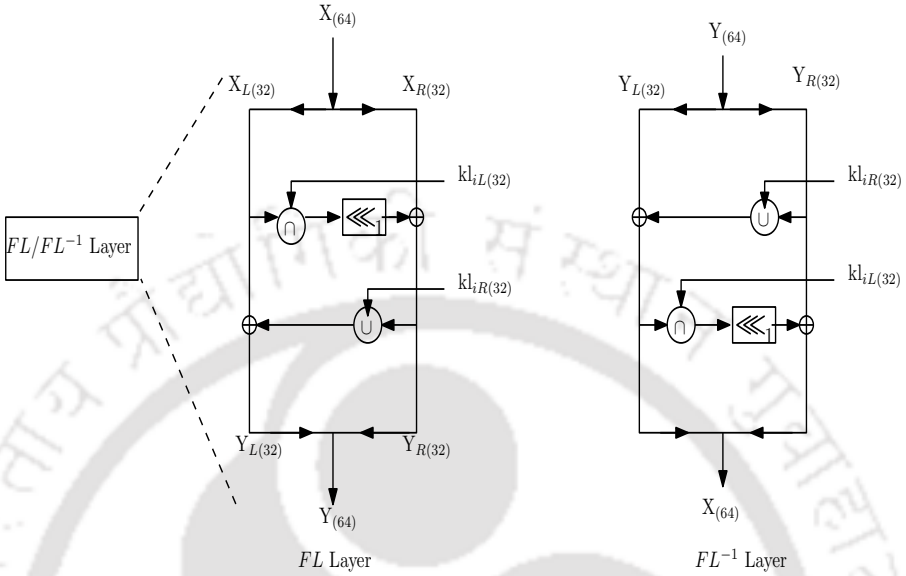

Figure 6.4: FL and FL^{-1} Layer

$$FL : N_{32} \times N_{32} \rightarrow N_{32} \tag{6.4}$$

$$(X_{L(32)} \| X_{R(32)}, K_{L(32)} \| K_{R(32)}) \rightarrow Y_{L(32)} \| Y_{R(32)} \tag{6.5}$$

Where $Y_{R(32)} = ((X_{L(32)} \cap K_{L(32)}) \lll 1) \oplus X_{R(32)}$ and $Y_{L(32)} = (Y_{R(32)} \cup K_{R(32)}) \oplus X_{L(32)}$ The FL^{-1} function is realized using following expressions.

$$FL^{-1} : N_{32} \times N_{32} \rightarrow N_{32} \tag{6.6}$$

$$(Y_{L(32)} \| Y_{R(32)}, K_{L(32)} \| K_{R(32)}) \rightarrow X_{L(32)} \| X_{R(32)} \tag{6.7}$$

Where $X_{L(32)} = (Y_{R(32)} \cup K_{R(32)}) \oplus Y_{L(32)}$ and $X_{R(32)} = ((X_{L(32)} \cap K_{L(32)}) \lll 1) \oplus Y_{R(32)}$.

6.2.4 F Function

The F function is the pivotal part of the Camellia algorithm which is used to cipher, decipher and key scheduling process. The security provided by Camellia algorithm depends

on the F function. The F function consists of P function and S function shown in Figure 6.5. The data after completion of prewhitening process is passed through S function

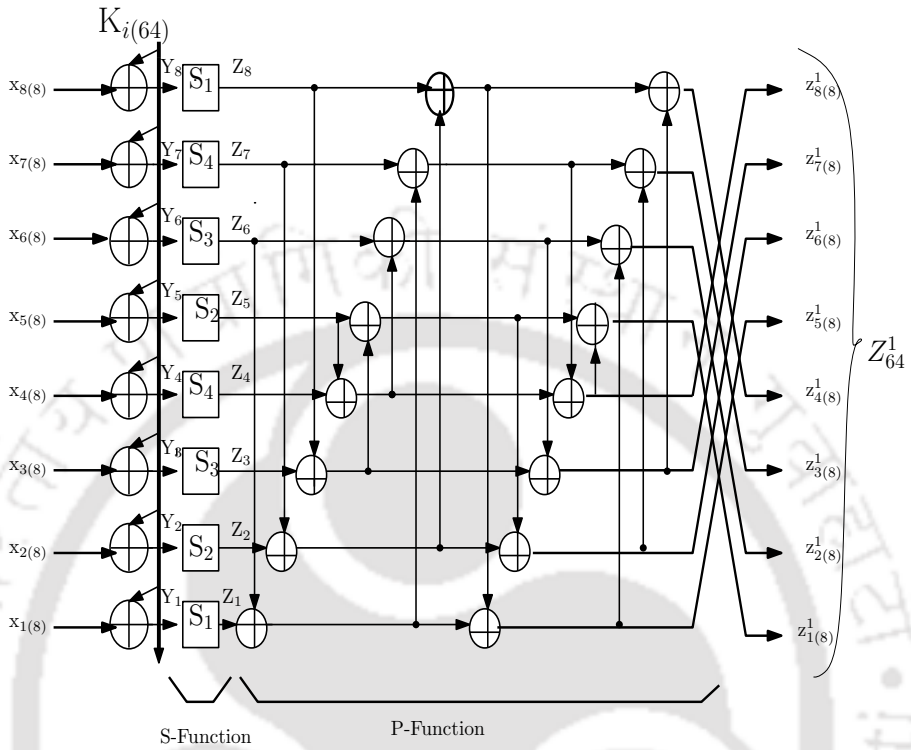

Figure 6.5: F function process

(S-Box) and P function to achieve random 64 bit output, shown in Figure 6.4. The S-Box is a nonlinear transformation of data using predefined LUT while P function is a linear transformation of data. The 64 bit Z^1_{64} output is achieved using the following expression.

$$P(S(X_{64} \oplus K_{L(64)})) \tag{6.8}$$

6.2.4.1 P Function

It can be clearly seen from the Figure 6.5 that F function is a linear transformation of the output of S function. Moreover, the linear transformation also comprehends the permutation for the output of S function and thus bestow the diffusion effect in the achieved cipher text. The structure of Camellia algorithm makes it more resistible toward linear and differential cryptanalysis [99].

$$Z^1_{8(8)} = Z_{7(8)} \oplus Z_{6(8)} \oplus Z_{5(8)} \oplus Z_{4(8)} \oplus Z_{1(8)} \tag{6.9}$$

$$Z^1_{7(8)} = Z_{8(8)} \oplus Z_{6(8)} \oplus Z_{5(8)} \oplus Z_{4(8)} \oplus Z_{3(8)} \tag{6.10}$$

$$Z^1_{6(8)} = Z_{8(8)} \oplus Z_{7(8)} \oplus Z_{5(8)} \oplus Z_{3(8)} \oplus Z_{2(8)} \tag{6.11}$$

$$Z^1_{5(8)} = Z_{8(8)} \oplus Z_{7(8)} \oplus Z_{6(8)} \oplus Z_{2(8)} \oplus Z_{1(8)} \tag{6.12}$$

$$Z^1_{4(8)} = Z_{7(8)} \oplus Z_{6(8)} \oplus Z_{5(8)} \oplus Z_{4(8)} \oplus Z_{3(8)} \oplus Z_{1(8)} \tag{6.13}$$

$$Z^1_{3(8)} = Z_{8(8)} \oplus Z_{6(8)} \oplus Z_{5(8)} \oplus Z_{3(8)} \oplus Z_{2(8)} \oplus Z_{1(8)} \tag{6.14}$$

$$Z^1_{2(8)} = Z_{8(8)} \oplus Z_{7(8)} \oplus Z_{5(8)} \oplus Z_{4(8)} \oplus Z_{2(8)} \oplus Z_{1(8)} \tag{6.15}$$

$$Z^1_{1(8)} = Z_{7(8)} \oplus Z_{7(8)} \oplus Z_{6(8)} \oplus Z_{4(8)} \oplus Z_{3(8)} \oplus Z_{1(8)} \tag{6.16}$$

In matrix form, the P function can be shown as follows.

$$\begin{bmatrix} Z^1_{8(8)} \\ Z^1_{7(8)} \\ Z^1_{6(8)} \\ Z^1_{5(8)} \\ Z^1_{4(8)} \\ Z^1_{3(8)} \\ Z^1_{2(8)} \\ Z^1_{1(8)} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} Z_{8(8)} \\ Z_{7(8)} \\ Z_{6(8)} \\ Z_{5(8)} \\ Z_{4(8)} \\ Z_{3(8)} \\ Z_{2(8)} \\ Z_{1(8)} \end{bmatrix} \tag{6.17}$$

6.2.4.2 S Function (S-Box)

The function of S-Box is to create confusion in data by using non linear transformation of input data. There are four types of S-Box such as S_1, S_2, S_3, S_4, used in Camellia algorithm. These S-Boxes are implemented on hardware using predefined LUT.

$$S : N_{32} \rightarrow N_{32} \tag{6.18}$$

$$S_1 : GF(2)^8 \rightarrow GF(2)^8 : n_{(8)} \rightarrow h(g(f(c5H \oplus n_{(8)}))) \oplus 6eH \tag{6.19}$$

$$S_2 : GF(2)^8 \rightarrow GF(2)^8 : n_{(8)} \rightarrow S_1(n_{(8)}) \lll 1 \tag{6.20}$$

$$S_3 : GF(2)^8 \rightarrow GF(2)^8 : n_{(8)} \rightarrow S_1(n_{(8)}) \ggg 1 \tag{6.21}$$

$$S_4 : GF(2)^8 \rightarrow GF(2)^8 : n_{(8)} \rightarrow S_1(n_{(8)}) \lll 1 \tag{6.22}$$

6.3 Proposed LPCA based F function

The conventional LUT based S-Box of F function realization on hardware utilize large number of memory cells which eventually consume more power. The overall F function realization on hardware require 16 LUT based S-Boxes. In order to overcome the limitation of traditional F function, we have proposed a LPCA based architecture for F function with low energy consumption and dynamic in nature. The proposed basic LPCA structure output is one bit with 8 bits input CA rule, shown in Figure 6.6. In order to implement the F function architecture of 64 bits, 64 such basic LPCA cell structure shown in Figure 6.6 are interconnected. Using logic gates, multiplexers and registers, the proposed LPCA

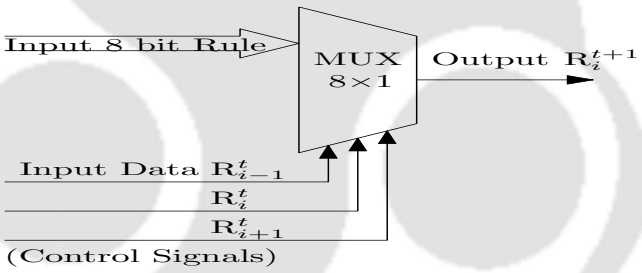

Figure 6.6: Basic LPCA cell structure

based F function design of 64×64 array has been implemented on hardware shown in Figure 6.7. First, the 64 bits input data X_{64} XORed with $K_{L(64)}$ and the output is loaded into register R_1 to R_{64}. The bits in the register R_1 to R_{64} act as control signals to 8:1 MUX (M_1-M_{64}) in a circular manner. Initially, the 3 bits R_{64}, R_1 and R_2 are applied as a control signals to M_1, the next bits R_1, R_2 and R_3 are applied to M_2 and the last MUX M_{64} the applied control signal are R_{63}, R_{64} and R_1. The control logic circuit is used enable the register (R_2) which comprises of comparator and 6 bit up counter. If the counter value and the number of defined iterations are equal, then the control logic circuit output goes high. The latency incurred in computing the F function is 20 clock cycles. The total

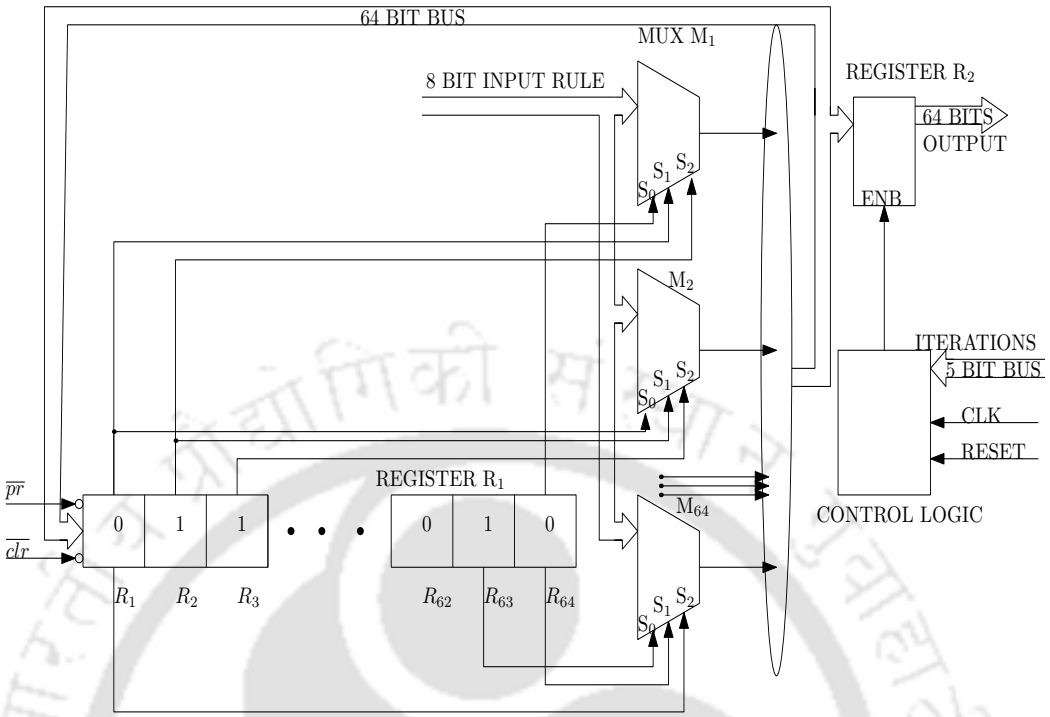

Figure 6.7: Proposed LPCA based F function

computational time of the Camellia algorithm with LPCA based F function is 84 clock cycles. Moreover, the LPCA based F function architecture has been implemented on ASIC shown in Figure 6.7 utilize few logic elements compared to that of LUT based S-Box. As a result, LPCA based F function architecture consumes less power and require small chip area and hence this hardware realization is much suitable for WBAN applications.

6.4 Performance Comparison between LUT based S-Box of F function, LPCA based F function

The cryptographic properties as discussed in Section 3.4, are taken into consideration in order to examine security aspects of the proposed F function realization. The SAC values achieved for the proposed F function realization have been plotted in Figure 6.8. The best value observed for SAC is 14 for more than 18% of rules, which is clear from Figure 6.8. The best observed value of SAC are at rules 62 and 125, shown in Figure 6.8. The values NL of the proposed F function realization have been plotted in Figure 6.9.

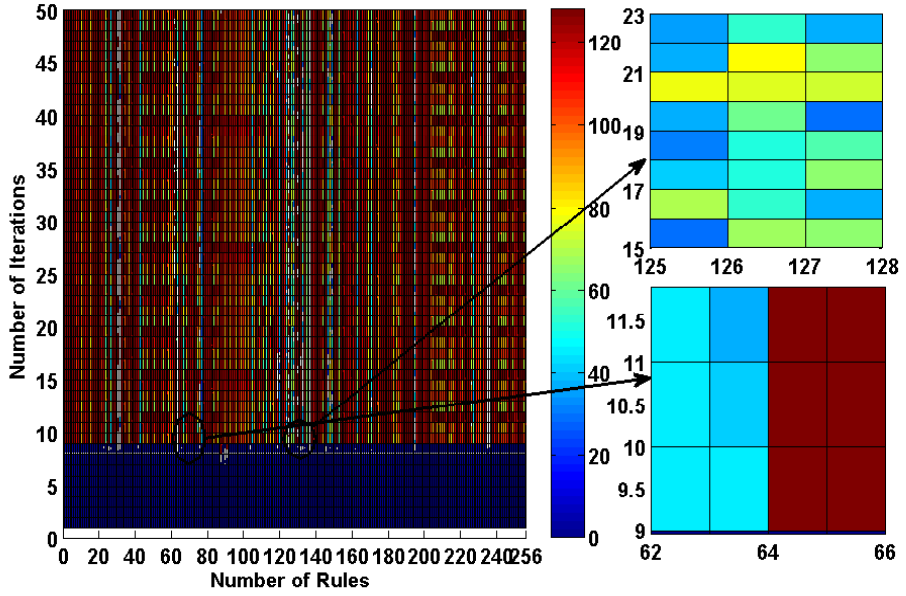

Figure 6.8: Values of SAC with Different Rules

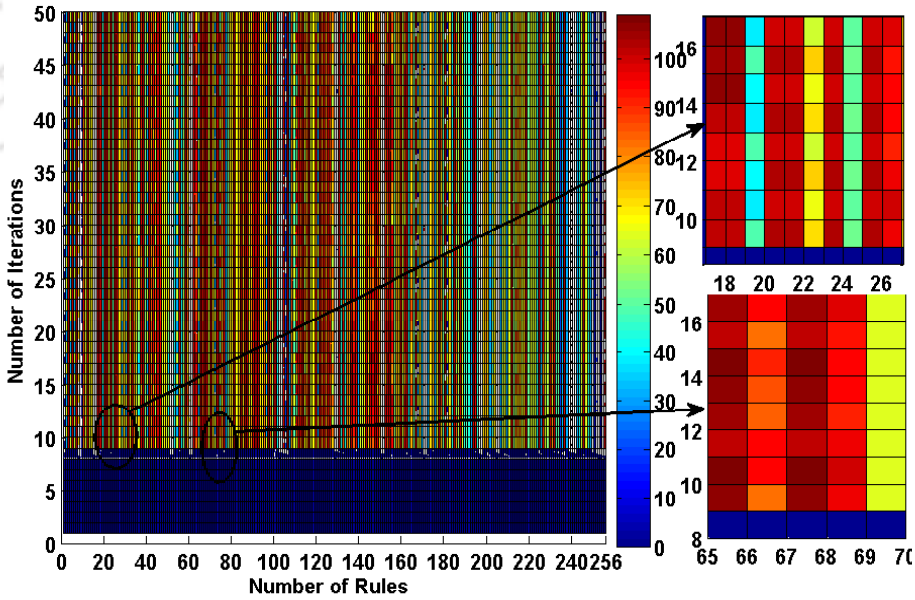

Figure 6.9: Values of NL with 256 Rules

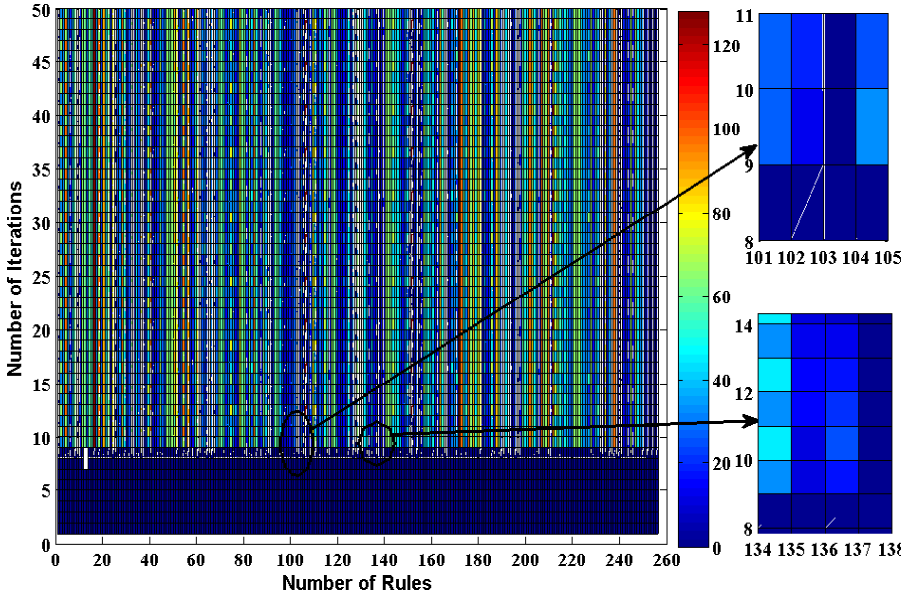

Figure 6.10: Values CIB with Different Rules

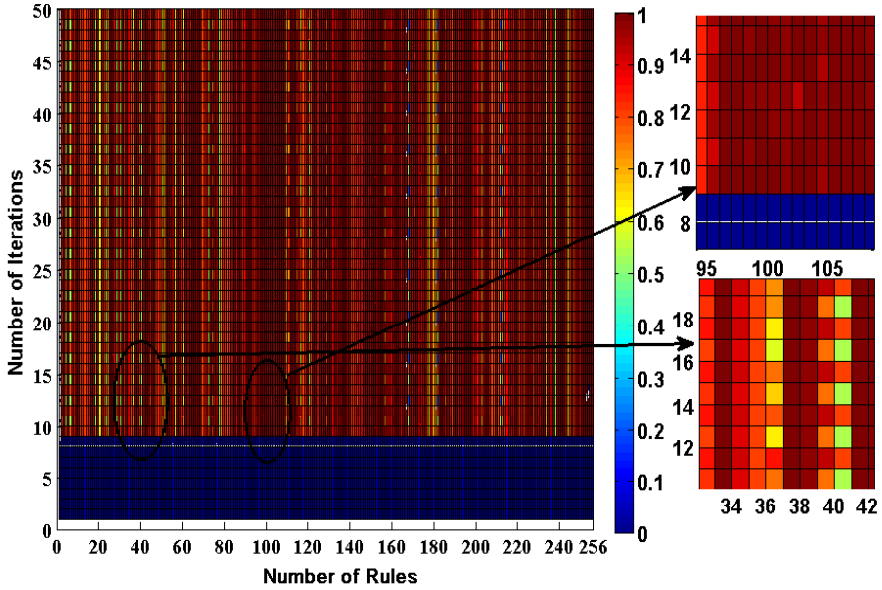

Figure 6.11: Values of Entropy with Different Rules

However, the value of NL varies from [0,109], it is observed that 15% of 256 LPCA has high NL values shown in Figure 6.9. The best observed values of NL are at rules 21, 22, 65 and 67 shown in Figure 6.9. The observed values of entropy for the proposed F function realization have been plotted in Figure 6.11. The entropy of LPCA F function varies from [0,1], the best value observed is 0.99. The achieved value for LUT based S-Boxes of F function has been depicted in the Table 6.3. The LPCA based F function in terms of security with respect to entropy has been found better than that of LUT based S-Box of F function. The best observed values of entropy are at rules 102, 103, 36 and 38 as shown in Figure 6.11. The observed values of entropy for the proposed F function realization are plotted in Figure 6.10. The CIB values of LPCA F function varies from [0,128]. The CIB values less than 12 are for 33% of 256 LPCA rules as shown in Figure 6.10. The best observed value of CIB are at rule 101, 103 and 135 shown in Figure 6.10. The performance of LPCA F function in terms of security is comparable with that of traditional F function.

The comparative performance of proposed LPCA based F function and conventional LUT based S-Box of F function in terms of level of security are shown in Table 6.3. The LPCA based F function is flexible, dynamic in nature and also found that LPCA F function provides enough level of security compared to LUT based S-Box of F function.

6.5 Proposed RCA^2 based F function

In order to overcome the limitation discussed in Section 4.5, we have focused on RCA^2 based architecture for F function which consumes low power dissipation. The basic function of RCA^2 based F function in Camellia algorithm is to transform 64 bits input data to another secret data. The output of proposed fundamental RCA^2 structure is one bit for a given input rule, shown in Figure 6.12. The F function operates on 64 bits, hence 64 basic cells shown in Figure 6.12 are interconnected.

The bits in registers R_1 to R_{64} will function as control signals to the multiplexer M_1-M_{64}. The input 64 bits are XORed with $K_{L(64)}$ and attained output bits are loaded into register R_1 to R_{64} using preset and clear signals. The 3 bits R_{64}, R_1 and R_2 will act as a control signals to M_1. The bits R_1, R_2 and R_3 control signals to multiplexer M_2 and the

Table 6.3: CIB, SAC, NL, Entropy values for LPCA based F Funtion and Standard Camellia LUT S-Box using cryptographic properties

Rule No.	No of time Steps	Non-Linearity	Entropy	CIB	SAC
21	12	**110**	**0.95**	**16**	**16**
23	14	**109**	**0.97**	**13**	**14**
30	15	**108**	**0.98**	**20**	**16**
35	14	106	0.98	8	20
42	14	102	0.99	10	20
62	14	102	0.99	10	14
65	14	107	0.98	20	12
67	10	**108**	**0.98**	**18**	**20**
102	12	106	0.99	14	10
103	14	102	0.99	18	16
104	12	108	0.98	14	14
125	7	101	0.99	6	14
135	12	102	0.99	6	18
138	12	106	0.99	10	22
Hussain et.al [84]	NA	105 96	NP NP	NP NP	16 10
Clark et.al [85]	NA	90 100	NP NP	19 24	44 48
Millan et.al [86]	NA	80	NP NP	NP NP	16 18
Nedjah et.al [87]	NA	70 102	NP NP	NP NP	NP NP
Standard Camellia LUT S-Box of F function					
S_1-Box	NA	105	0.99	15	16
S_2-Box	NA	109	0.99	13	12
S_3-Box	NA	110	0.99	15	16
S_4-Box	NA	111	0.99	14	14

* NA means not applicable

* NP means not provided

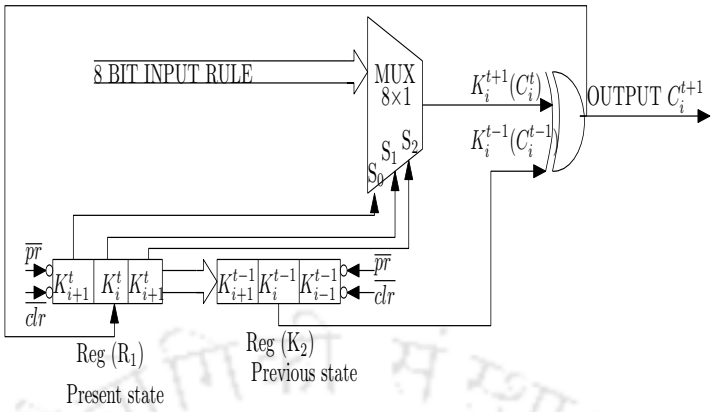

Figure 6.12: Baisc Structure of RCA^2 based F function

MUX M_{64} the control signal are R_{63}, R_{64} and R_1. The register K_1 to K_{64} are used to store the previous value of bits R_1 to R_{64} as K_1 to K_{64}. The previous bit K_1 is XORed with the output of MUX M_1, K_2 is XORed with output of MUX M_2 and similarly, the bits K_{64} is XORed with output of MUX M_{64}. The control logic comprises of 6 bit up counter and a comparator. If the output value of counter is equal to the number of iteration in time step, then the output of the control logic circuit goes high to enable the register (R_3). The total time taken for computing the F function depends upon the number of iterations defined in the RCA^2. The RCA^2 based F function architecture shown in Figure 6.13. The simulation results show that the RCA^2 based F function architecture consume less power and hence the proposed hardware architectures are applicable for WBAN applications.

6.6 Security analysis of LUT based S-Box of F function and RCA^2 based F function

The level of security provided by the proposed F function has been validated using the cryptographic properties, such as, CIB, NL, SAC and entropy as discussed in Section 3.4. It has been found that the value of SAC for RCA^2 F function is 18 for rules 53, 55, 101, 102 and 103 shown in Figure 6.14, which is comparable with that of traditional LUT based S-Box of F function. Moreover, we have also found that 20.45% out of 256 CA rules has SAC value of 14.

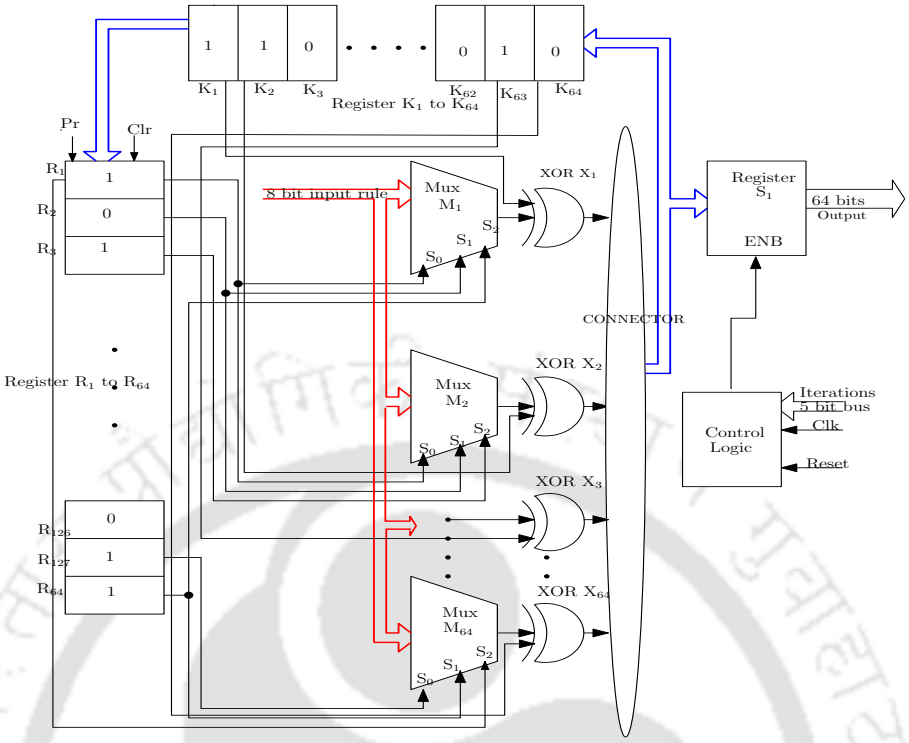

Figure 6.13: Proposed RCA^2 based F function

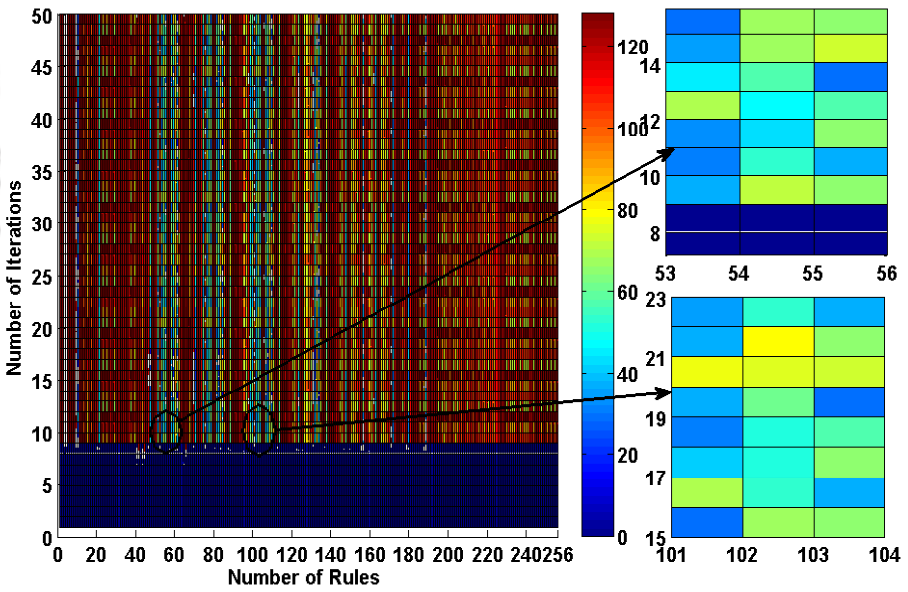

Figure 6.14: Values for SAC of RCA^2 based F function

The values of SAC achieved at rules 30, 57, 75, 86 and 98 have been presented in Table 6.4. It has been observed that 58.782% out of 256 CA rules better entropy value

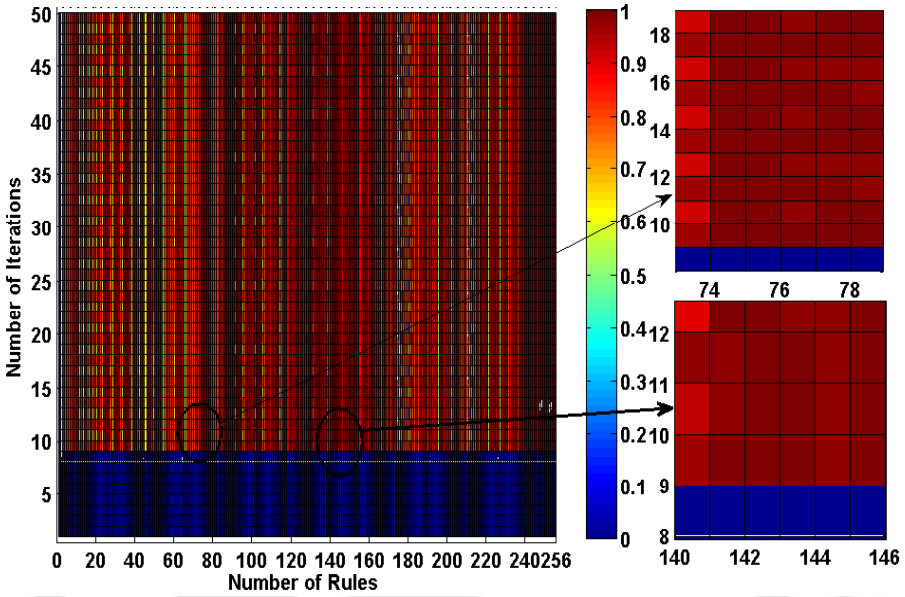

Figure 6.15: Values for Entropy of RCA^2 based F function

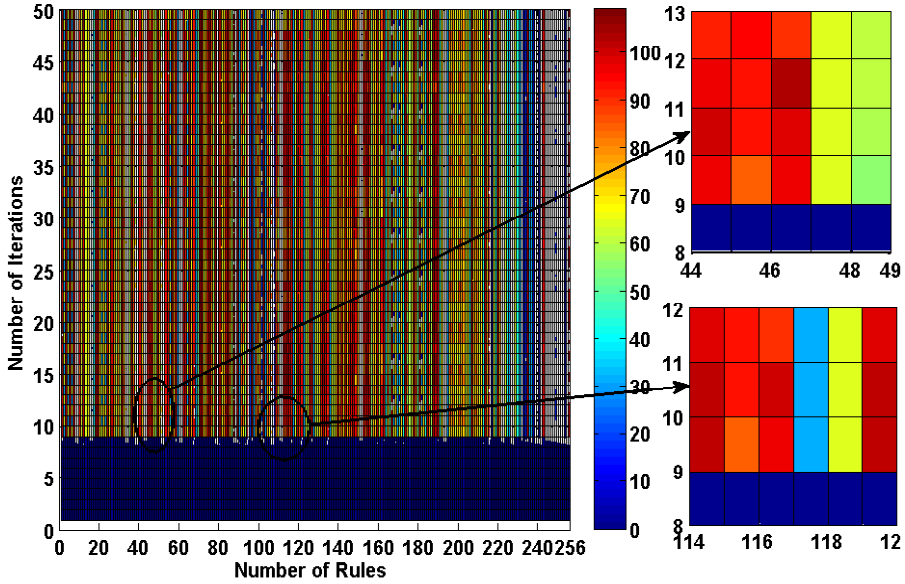

Figure 6.16: Values for NL of RCA^2 based F function

for RCA^2 based F function than standard LUT based F function entropy values. The observed values of entropy is 0.9914 at rule 57 and 99 as shown in Figure 6.15. It has been observed that the value of NL is high for rules 45, 47, 116, 120 and 21.023% out of 256 CA rules have high value of NL, shown in Figure 6.16. The maximum value of NL attained is 106 for RCA^2 based F function, shown in Table 6.4.

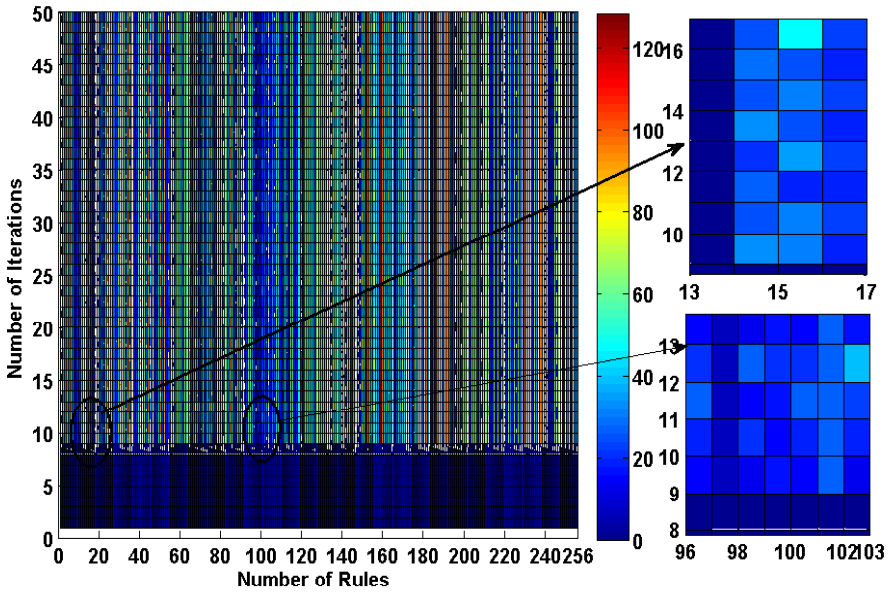

Figure 6.17: Value for CIB of RCA^2 based F function

It has been observed that the best values of CIB are at rules 30, 57 and 99 reported in Table 6.4. For 58.458% out of 256 RCA^2 rules have better value of CIB as shown in Figure 6.17. The value of SAC, CIB, NL and entropy of RCA^2 based F function along with few reversible rules have been shown in Table 6.4. The RCA^2 based F function is dynamic in nature and more resistant to differential cryptanalysis so as to provide enough level of security. The observed value using cryptographic properties for LUT based S-Box of F function and proposed RCA^2 F function to the existing works shown in Table 6.4 [84–87].

6.6.1 Architectural Design Comparision

The proposed architecture of Camellia algorithm with RCA^2 based F function have been implemented using Verilog, verified on FPGA board and simulation results are reported in Table 6.5. The proposed RCA^2 based F function for Camellia algorithm im-

Table 6.4: Cryptographic Properties values for RCA^2 based F function

Rule No	Time Step	NL	Entropy	CIB	SAC
30	15	100	0.98	14	16
53	8	98	0.96	14	18
57	10	**106**	**0.99**	**12**	**16**
75	6	101	0.99	18	16
86	10	100	0.98	12	14
98	14	101	0.98	14	10
99	12	**106**	**0.99**	**16**	**14**
102	18	102	0.97	15	18
135	13	100	0.98	18	28
142	10	99	0.99	16	18
149	14	100	0.98	18	20
169	14	100	0.99	16	24
225	12	101	0.98	20	18
Hussain et.al [84]	NA	105 96	NP NP	NP NP	16 10
Clark et.al [85]	NA	90 100	NP NP	19 24	44 48
Millan et.al [86]	NA	80	NP NP	NP NP	16 18
Nedjah et.al [87]	NA	70 102	NP NP	NP NP	NP NP
Standard Camellia LUT S-Box of F function					
S_1-Box	NA	105	0.99	15	16
S_2-Box	NA	109	0.99	13	12
S_3-Box	NA	110	0.99	15	16
S_4-Box	NA	111	0.99	14	14

NA means not applicable

NP means not provided

plemented on FPGA consumes less hardware and low energy consumption compared with reported designs [40, 41, 100]. The proposed RCA^2 F function and LPCA F function architectures with Camellia algorithm have been implemented using TSMC 0.18-μm technology (core voltage of 1.62 V) and UMC 0.13-μm technology (core voltage of 1.08 V) under worst-case conditions at different clock frequencies using Cadence RTL Compiler. The total time consumed to encrypt 128 bits plain text by Camellia algorithm is calculated by $Latency = Clockcycles \times Timeperiod$. However, the proposed RCA^2 based F function realization and LPCA F function, the number of iterations considered are 20 clock cycles, therefore, the total time taken to encrypt 128 bits plain text by Camellia algorithm with RCA^2 F function and LPCA F function is 84 clock cycles respectively.

Table 6.5: FPGA Simulation results of Proposed Camellia Algorithm with RCA^2 based F function

Camellia	FPGA	Slices	Power (mW)	Frequency (MHz)	Clock cycles	Energy (nJ)
Kavun [100]	XC3S50-5	321	NP	103	400	174
Yalla [40]	XC3s50-5	318	NP	NP	875	NP
Satoh [41]	Virtex-E xcv1000E-8	2833	NP	55.30	18	NP
Proposed Camellia algorithm	Virtex-E xcv1000E-8	1758	1.59	500	84	31.80

* NP means not provided

Table 6.6: Hardware results of the Proposed Camellia algorithm with LPCA based F function

Camellia	Tech	Gates	Power (mW)	Frequency (MHz)	Clock cycles	Energy (nJ)
Kavun [100]	0.13μm	4313	NP	253	400	NP
Akoi [42]	0.035μm	272,819	NP	10	NP	NP
Satoh [41]	0.13μm	20,788	NP	327.87	22	NP
	0.18μm	20,911	NP	222.22	22	NP
Wang [101]	0.065μm	4.6M	NP	300	NP	23.90
Proposed Camellia algorithm	0.18μm	3128	1.24	10	84	10.416
	0.13μm	2783	0.76	10	84	6.384

* NP means not provided

Table 6.7: Synthesis results of Proposed Camellia Algorithm with RCA^2 based F function

Camellia	Tech	Gates	Power (mW)	Frequency (MHz)	Clock cycles	Energy (nJ)
Kavun [100]	0.13μm	4313	NP	253	400	NP
Akoi [42]	0.035μm	272,819	NP	10	NP	NP
Satoh [41]	0.13μm	20,788	NP	327.87	22	NP
	0.18μm	20,911	NP	222.22	22	NP
Wang [101]	0.065μm	4.6M	NP	300	NP	23.90
Proposed Camellia algorithm	0.18μm	3328	1.96	10	84	16.464
	0.13μm	2986	0.85	10	84	7.14

* NP means not provided

It is clear from Table 6.6, that the proposed Camellia algorithm with RCA^2 based F function and LPCA based F function out performs in terms of power dissipation and energy consumption compared with existing works. Although the power consumption is not mentioned in [41], the gate count is much larger than the gate count required by the proposed architecture. Hence, the proposed algorithm can be considered better than Satoh et.al. algorithm [41]. The CFA based S-Box implementation on hardware is 2 to 5 times faster than the LUT based S-Box. However, the CFA have gate count of 2 to 9 times more than classical S-Box. Akash Satoh et.al. [41], presented LUT based S-Box for Camellia where S_1, S_2, S_3, S_4 utilize 540, 560, 557, 562 gates respectively and CFA based S-Box use 256 gates with 0.18-μm CMOS technology. The proposed LPCA based F function consume 144, 168 gates using 0.18-μm and 0.13-μm technology libraries respectively. As a result, the LUT based S-Box and CFA based S-Box architecture consume more power and area [41]. In [92], the gates used for realization of CFA based S-Box and LUT based S-Box are 696, 294 respectively with 0.11-μm technology, whereas, the proposed RCA^2 based F function utilize 184, 236 gates using 0.18-μm and 0.13-μm technology libraries respectively. Simulation results show that the proposed LPCA based F function and RCA^2 F function architecture with Camellia algorithm when operated at 10 MHz clock consume power of 0.76mW, 0.85 mW and energy consumption of 6.384 nJ, 7.14 nJ respectively shown in Table 6.6 and Table 6.7, it is clear that there is decrease in energy consumption for

proposed architectures compared with Wang et.al [101].

6.7 Summary

In order to overcome the limitation of traditional S-Box of F function used in Camellia algorithm, we have proposed RCA^2 based F function and LPCA based F function architectures. Unlike the designs reported in [40–42, 100, 101], the proposed architecture requires fewer logical components, hence results in low power dissipation. The security issues of the classical LUT based S-Box of F function, LPCA based F function and RCA^2 based F function have been examined using cryptographic properties and also found that the proposed F functions give better performance than that of traditional LUT based S-Box F function realization. The proposed architecture has been synthesized using Cadence RTL Compiler to evaluate area, power and frequency of operation. The Camellia algorithm with proposed LPCA F function and RCA^2 F function architectures, operated at 10 MHZ clock using 180-μm CMOS technology, consume power of 1.24 μW, 1.96 μW and energy consumption of 10.416 nJ, 16.464 nJ respectively. Therefore, it has been observed that Camellia with RCA^2 based F function and LPCA based F function are an ultra low power architectures, which are suitable for WBAN applications.

7

Conclusions

Contents

Objective

In this chapter, the summary of contributions made by this thesis towards the development of encryption algorithms for WBAN applications has been discussed targeting Low power consumption and less energy dissipation. Future possibilities of research in this field are also outlined.

7.1 Summary

The objective of this thesis work is to develop low energy consuming cryptographic algorithms for WBAN applications. To achieve this, block cipher cryptographic algorithms such as AES, Camellia have been taken into consideration, adopted in latest IEEE Standard 802.15.6 for WBAN applications [3, 6, 7] .

(i) **Evaluation of S-Box and Implementation using Composite Field Arthimetic**: In Chapter 3, the LUT based S-Box has been constructed using 30 different irreducible polynomials including the AES standard polynomial. The study reveals that the LUT based S-Box can be constructed using other polynomial equations. The level of security provided by the construction of S-Box using different irreducible polynomials has been verified with cryptographic properties. Moreover, the S-Box has been constructed by CFA technique for standard irreducible polynomial in order to reduce the gate count. The hardware realization of proposed CFA S-Box achieved less hardware utilization and low power consumption compared to the classical S-Box of AES algorithm.

(ii) **Low Energy Architectures of S-Box for AES algorithm**: In Chapter 4, the theory of CA has been applied for the construction of S-Box. The S-Box architecture for AES using PCA and RCA^2 are discussed in this Chapter. The level of security provided by the proposed S-Box has been verified using cryptographic properties, namely, NL, CIB, SAC and entropy. Simulation studies also show that the proposed S-Box architecture using PCA and RCA^2 utilizes less hardware, enjoys low power dissipation and less energy consumption.

(iii) **CA based encryption algorithms**: In Chapter 5, low power encryption algorithms have been developed using HLCA and HRCA^2. The security provided by the proposed encryption algorithms has been examined using cryptographic properties.

(iv) **Low Power Architecture of F function for Camellia algorithm**: In Chapter 6, the F function architecture for Camellia encryption algorithm has been designed using LPCA and RCA^2. The F function using LPCA and RCA^2 are more flexible and dynamic in nature with low power dissipation capability compared to that of traditional LUT based S-Box. Due to less energy consumption, the proposed architectures are suitable for WBAN application. The security provided by the proposed F function against cryptanalysis has been verified using cryptographic properties and it has been found that the proposed F function is secure enough against cyber-attacks.

7.2 Contributions

The major contributions of the research work reported in this thesis include:

(i) The construction of S-Box is achieved using different polynomial equations and S-Box has been implemented using CFA technique.

(ii) The architectures of S-Box have been implemented on hardware using PCA and RCA^2.

(iii) The low power encryption algorithms have been designed using HLCA, HRCA^2 and verified on hardware.

(iv) The F function architecture for Camellia algorithm has been designed using HRCA^2, LPCA and implemented on hardware.

(v) The hardware performance of the proposed architectures have been verified on FPGA and ASIC using TSMC 0.18-μm, UMC 0.13-μm CMOS technology libraries.

(vi) The security provided by the proposed architectures has been examined using cryptographic properties such as NL, CIB, SAC and entropy.

7.3 Directions for future work

Based on the outcome of this thesis work, this section provides the possible future directions for research.

(i) The CFA based S-Box has been implemented on FPGA. Further, the CFA based S-Box architecture can be implemented with different irreducible polynomial equations in order to examine hardware utilization and power consumption.

(ii) The concept of CA has been used for the construction of S-Box for AES, Camellia. Further, the proposed S-Box architecture can be investigated for implementation with folding and pipelining techniques where high throughput is essential.

(iii) The proposed encryption algorithms using HLCA and HRCA^2 can be realized using folding, sub-pipelining technique for better performance. A thorough analysis is required in order to implement these techniques.

(iv) The F function architecture of Camellia with RCA^2 and PCA using folding and unfolding techniques can be implemented for better performance .

(v) In this work, a set of cryptographic properties have been considered in order to examine the security level of the achieved cipher text of encryption algorithm. A better analysis can be developed in order to check the robustness of cipher text against unauthorized attacks such as power analysis attack and linear cryptanalysis.

Bibliography

[1] M. Al Ameen, J. Liu, and K. Kwak, "Security and Privacy Issues in Wireless Sensor Networks for Healthcare Applications," *Journal of Medical Systems*, vol. 36, no. 1, pp. 93–101, 2012.

[2] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao, "Body area network security: A fuzzy attribute-based signcryption scheme," *IEEE Journal on Selected Areas in Communications,*, vol. 31, no. 9, pp. 37–46, September 2013.

[3] *IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks*, Std., Feb 2012.

[4] S. Ullah, H. Higgins, B. Braem, B. Latre, C. Blondia, I. Moerman, S. Saleem, Z. Rahman, and K. Kwak, "A Comprehensive Survey of Wireless Body Area Networks," *Journal of Medical Systems*, vol. 36, no. 3, pp. 1065–1094, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10916-010-9571-3

[5] J. Choi and C. Lee, "Maximum a posteriori (MAP)-based tag estimation method for dynamic framed-slotted ALOHA (DFSA) in RFID systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, 2012. [Online]. Available: http://dx.doi.org/10.1186/1687-1499-2012-268

[6] *Advanced Encryption Standard (AES)*, Federal Information Processing Standards Publication 197 Std., November 26 2001.

[7] *Specifications of Camellia - a128-bit block cipher*, Nippon Telegraph and Telephone Corporation, Mitsibishi Electric Corporation Std.

[8] National Institute of Standards and Technology, *FIPS PUB 46-3: Data Encryption Standard (DES)*, Oct. 1999, supersedes FIPS 46-2.

[9] W. Burr, "Selecting the Advanced Encryption Standard," *IEEE Security Privacy,*, vol. 1, no. 2, pp. 43–52, Mar 2003.

[10] E. Biham, A. Biryukov, and A. Shamir, *"Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials"*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 12–23.

[11] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, "Wireless Body Area Networks: A Survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1658–1686, Third 2014.

[12] Hanlen. L ,D. Smith, A. Boulis, B. Gilbert, V. Chaganti, L. Craven, D. Fang, T. Lamahewa, D. Lewis, D. Miniutti, O. Nagy, D. Rodda, K. Sithamparanathan, Y. Tselishchev, A. Zhang, "Wireless body-area-networks: toward a wearable intranet." in National ICT Australia, 2011.

[13] P. K. Manchi and R. Paily and A. K. Gogoi, "Low-Power Digital Baseband Transceiver Design for UWB Physical Layer of IEEE 802.15.6 Standard," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2474–2483, Oct 2017.

[14] S. Nandi, B. K. Kar, and P. P. Chaudhuri, "Theory and Applications of Cellular Automata in Cryptography," *IEEE Trans. Comput.*, vol. 43, no. 12, pp. 1346–1357, Dec. 1994.

[15] A. Bechtsoudis and N. Sklavos, "Side Channel Attacks Cryptanalysis against Block Ciphers Based on FPGA Devices," in *2010 IEEE Computer Society Annual Symposium on VLSI (ISVLSI),*, July 2010, pp. 460–461.

[16] Xinmiao Zhang and Parhi, K.K., "On the Optimum Constructions of Composite Field for the AES Algorithm," *IEEE Transactions on Circuits and Systems II: Express Briefs,*, vol. 53, no. 10, pp. 1153–1157, Oct 2006.

[17] M. M. Wong and M. L. D. Wong and I. Hijazin and A. K. Nandi, "Composite field GF(((2²)²)²) AES S-Box with direct computation in $GF(2^4)$ inversion," in *2011 7th International Conference on Information Technology in Asia*, July 2011, pp. 1–6.

[18] M. M. Wong and M. L. D. Wong, "t," in *2nd Asia Symposium on Quality Electronic Design (ASQED)*, Aug 2010, pp. 318–323.

[19] S. Morioka and A. Satoh, "A 10-Gbps full-AES crypto design with a twisted BDD S-Box architecture," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 7, pp. 686–691, July 2004.

[20] P. Shastry, N. Somani, A. Gadre, B. Vispute, and M. Sutaone, "Rolled architecture based implementation of AES using T-Box," in *IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), 2012*, Aug 2012, pp. 626–630.

[21] I. Hammad and K. El-Sankary and E. El-Masry, "High-Speed AES Encryptor With Efficient Merging Techniques," *IEEE Embedded Systems Letters*, vol. 2, no. 3, pp. 67–71, Sept 2010.

[22] Desai, A and Ankalgi, K. and Yamanur, H. and Navalgund, S.S., "Parallelization of AES algorithm for disk encryption using CBC and ICBC modes," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT),*, July 2013, pp. 1–7.

[23] Liakot Ali and Ishak Aris and Fakir Sharif Hossain and Niranjan Roy, "Design of an ultra high speed AES processor for next generation IT security ," *Computers and Electrical Engineering*, vol. 37, no. 6, pp. 1160 – 1170, 2011.

[24] Arora, A and Parameswaran, S. and Ragel, R. and Jayasinghe, D., "A Hardware/Software Countermeasure and a Testing Framework for Cache Based Side Channel Attacks," in *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom),*, Nov 2011, pp. 1005–1014.

[25] S. U. Jonwal and P. P. Shingare, "Advanced Encryption Standard (AES) implementation on FPGA with hardware in loop," in *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, May 2017, pp. 64–67.

[26] I. Verbauwhede and P. Schaumont and H. Kuo, "Design and performance testing of a 2.29-GB/s Rijndael processor," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 3, pp. 569–572, Mar 2003.

[27] Xinmiao Zhang and Parhi, K.K., "High-speed VLSI architectures for the AES algorithm," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,*, vol. 12, no. 9, pp. 957–967, Sept 2004.

[28] P. Hamalainen and T. Alho and M. Hannikainen and T. D. Hamalainen, "Design and implementation of low-area and low-power aes encryption hardware core," in *9th EUROMICRO Conference on Digital System Design (DSD'06)*, 2006, pp. 577–583.

[29] A. Bouhraoua, "Design Feasibility Study For A 500 Gbits/s AES Cypher Decypher Engine," in *2006 International Conference on Microelectronics*, Dec 2006, pp. 190–193.

[30] A. Hodjat and I. Verbauwhede, "Area-throughput trade-offs for fully pipelined 30 to 70 Gbits/s AES processors," *IEEE Transactions on Computers*, vol. 55, no. 4, pp. 366–372, April 2006.

[31] D. Feng and L. Chen and L. Zeng and Z. Niu, "FPGA/ASIC based Cryptographic Object Store System," in *Third International Symposium on Information Assurance and Security*, Aug 2007, pp. 267–272.

[32] Manoj Kumar, Thanikodi and Karthigaikumar, Palanivel, "FPGA implementation of an optimized key expansion module of AES algorithm for secure transmission of personal ECG signals," *Design Automation for Embedded Systems*, Oct 2017.

[33] V. Shende and M. Kulkarni, "FPGA based hardware implementation of hybrid cryptographic algorithm for encryption and decryption," in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, Dec 2017, pp. 416–419.

[34] Tian, Xingyu and Fan, Chunlei and Liu, Jia and Ding, Qun, *Design and Implementation of Network Video Encryption System Based on STM32 and AES Algorithm*. Cham: Springer International Publishing, 2018, pp. 51–58.

[35] Sayed, Mostafa Ahmed Mohamed and Rongke, Liu and Ling, Zhao, *FPGA Design and Implementation of High Secure Channel Coding Based AES*. Cham: Springer International Publishing, 2018, pp. 355–366. [Online]. Available: https://doi.org/10.1007/978-3-319-66628-0_34

[36] X. Gao and E. Lu and L. Li and K. Lang, "LUT-based FPGA Implementation of SMS4/AES/Camellia," in *Fifth IEEE International Symposium on Embedded Computing 2008*, Oct 2008, pp. 73–76.

[37] X. Tang, B. Sun, and C. Li.

[38] Denning, Daniel and Irvine, James and Devlin, Malachy, "A key agile 17.4 Gbit/sec Camellia implementation," *Field Programmable Logic and Application*, pp. 546–554, 2004.

[39] Y. Deng and T. Xie and H. Shi and J. Gong, "Research on the F-function of Camellia," in *2011 International Conference on Electrical and Control Engineering*, Sept 2011, pp. 1232–1235.

[40] P. Yalla and J. P. Kaps, "Compact FPGA implementation of Camellia," in *2009 International Conference on Field Programmable Logic and Applications*, Aug 2009, pp. 658–661.

[41] Satoh, Akashi and Morioka, Sumio, "Hardware-focused performance comparison for the standard block ciphers aes, camellia, and triple-des," in *International Conference on Information Security*. Springer, 2003, pp. 252–266.

[42] Aoki, Kazumaro and Ichikawa, Tetsuya and Kanda, Masayuki and Matsui, Mitsuru and Moriai, Shiho and Nakajima, Junko and Tokita, Toshio, *Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms — Design andAnalysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 39–56.

[43] Y. Huo and D. Liu, "High-Throughput Area-Efficient Processor for Cryptography," *Chinese Journal of Electronics*, vol. 26, no. 3, pp. 514–521, 2017.

[44] Z. Cica, "Pipelined implementation of Camellia encryption algorithm," in *24th Telecommunications Forum (TELFOR) 2016*, Nov 2016, pp. 1–4.

[45] D. Denning and J. Irvine and M. Devlin, "A high throughput FPGA Camellia implementation," in *Research in Microelectronics and Electronics, 2005*, vol. 1, July 2005, pp. 137–140 vol.1.

[46] Y. Hori, T. Katashita, and K. Kobara, "Energy and area saving effect of dynamic partial reconfiguration on a 28-nm process fpga," in *IEEE 2nd Global Conference on Consumer Electronics (GCCE) 2013*, Oct 2013, pp. 217–218.

[47] M. S. Elpeltagy and M. M. Abdelwahab and M. S. Sayed, "Image encryption using camellia and chaotic maps," in *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Dec 2015, pp. 209–214.

[48] Smith, Alvy Ray, "Simple computation-universal cellular spaces and self-reproduction," in *IEEE Conference Record of 9th Annual Symposium on Switching and Automata Theory, 1968.,*, Oct 1968, pp. 269–277.

[49] M. Szaban, J. Nowacki, A. Drabik, F. Seredynski, and P. Bouvry, "Application of Cellular Automata in Symmetric Key Cryptography," in *Advances in Information Technology*, ser. Communications in Computer and Information Science, B. Papasratorn, K. Lavangnananda, W. Chutimaskul, and V. Vanijja, Eds. Springer Berlin Heidelberg, 2010, vol. 114, pp. 154–163.

[50] *A New Kind of Science.* Champaign, Ilinois, US, United States: Wolfram Media Inc., 2002.

[51] O. Lafe, *Cellular Automata Transforms.* Boston, MA: Springer US, 2000, pp. 23–44.

[52] Lafe, "Data compression and encryption using cellular automata transforms," in *IEEE International Joint Symposia on Intelligence and Systems, 1996.,*, Nov 1996, pp. 234–241.

[53] R. Shiba, S. Kang, and Y. Aoki, "An image watermarking technique using cellular automata transform," in *2004 IEEE Region 10 Conference TENCON 2004.*, vol. A, Nov 2004, pp. 303–306 Vol. 1.

[54] L. Kotoulas, D. Tsarouchis, G. C. Sirakoulis, and I. Andreadis, "1-d cellular automaton for pseudorandom number generation and its reconfigurable hardware implementation," in *IEEE International Symposium on Circuits and Systems 2006*, May 2006, pp. 4 pp.–.

[55] S. Nandi and S. Roy and S. Nath and S. Chakraborty and W. Ben Abdessalem Karaa and N. Dey, "1-D group cellular automata based image encryption technique," in *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, July 2014, pp. 521–526.

[56] Jarkko Kari, "Theory of cellular automata: A survey," *Theoretical Computer Science*, vol. 334, no. 13, pp. 3 – 33, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030439750500054X

[57] J. Chen, M. Lai, Y. Huang, and G. Zhou, "The automata model for cloud storage," in *2012 International Conference on Information Science and Technology (ICIST),*, March 2012, pp. 583–590.

[58] Nandi, S. and Pal Chaudhuri, P., "Theory And Application Of Cellular Automata In Cryptography," *IEEE Transactions on Computers,*, vol. 46, no. 5, pp. 639–639, May 1997.

[59] Z. Chai, Z. Cao, and Y. Zhou, "Encryption based on reversible second-order cellular automata," in *Parallel and Distributed Processing and Applications - ISPA 2005 Workshops*, G. Chen, Y. Pan, M. Guo, and J. Lu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 350–358.

[60] X. Xia and Y. Li and Z. Xia and R. Wang, "Multi-Granularity Reversible Cellular Automata Applied in Data Encryption," in *2009 International Conference on Computational Intelligence and Software Engineering*, Dec 2009, pp. 1–4.

[61] S. A. Hosseini and I. Mohammadi and S. R. Kamel, "A parallel image encryption based on elementary cellular automata using two processors," in *2014 International Congress on Technology, Communication and Knowledge (ICTCK)*, Nov 2014, pp. 1–5.

[62] Xingyuan Wang and Dapeng Luan, "A novel image encryption algorithm using chaos and reversible cellular automata," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 11, pp. 3075 – 3085, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1007570413001524

[63] A. Y. Niyat and R. M. H. Hei and M. V. Jahan, "Chaos-based image encryption using a hybrid cellular automata and a DNA sequence," in *2015 International Congress on Technology, Communication and Knowledge (ICTCK)*, Nov 2015, pp. 247–252.

[64] M. T. Rodriguez-Sahagun and J. B. Mercado-Sanchez and D. Lopez-Mancilla and R. Jaimes-Reategui and J. H. Garcia-Lopez, "Image Encryption Based on Logistic Chaotic Map for Secure Communications," in *2010 IEEE Electronics, Robotics and Automotive Mechanics Conference*, Sept 2010, pp. 319–324.

[65] S. Rajagopalan and S. Rethinam and S. Janakiraman and H. N. Upadhyay and R. Amirtharajan, "Cellular automata synthetic image: A trio approach to image encryption," in *2017 International Conference on Computer Communication and Informatics (ICCCI)*, Jan 2017, pp. 1–6.

[66] *Advanced Encryption Standard (AES)*, Federal Information Processing Standards Publication 197 Std., November 26 2001.

[67] K.-T. Cheng and V. Agrawal, "An entropy measure for the complexity of multi-output boolean functions," in *27th ACM/IEEE Design Automation Conference, 1990. Proceedings.*,, Jun 1990, pp. 302–305.

[68] O. Rothaus, "On bent functions." *Journal of Combinatorial Theory, Series A*, vol. 20, no. 3, pp. 300 – 305, 1976. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0097316576900248

[69] A. Webster and S. Tavares, "On the design of s-boxes," in *Advances in Cryptology CRYPTO 85 Proceedings*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1986, vol. 218, pp. 523–534. [Online]. Available: http://dx.doi.org/10.1007/3-540-39799-X_41

[70] C. Adams and S. Tavares, "Good s-boxes are easy to find," in *Advances in Cryptology 'CRYPTO' Proceedings*, ser. Lecture Notes in Computer Science, G. Brassard, Ed. Springer New York, 1990, vol. 435, pp. 612–615. [Online]. Available: http://dx.doi.org/10.1007/0-387-34805-056

[71] T. W. Cusick and P. Stnic, *Cryptographic Boolean Functions and Applications*. Elsevier Inc, 2009.

[72] J. Clark, J. Jacob, and S. Stepney, "The design of s-boxes by simulated annealing," *New Generation Computing*, vol. 23, no. 3, pp. 219–231, 2005. [Online]. Available: http://dx.doi.org/10.1007/BF03037656

[73] Murphy, Sean and Robshaw, Matthew J.B., "Essential Algebraic Structure within the AES," in *Advances in Cryptology — CRYPTO 2002*, M. Yung, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 1–16.

[74] U. Waqas and S. Afzal and M. A. Mir and M. Yousaf, "Generation of AES-Like S-Boxes by Replacing Affine Matrix," in *2014 12th International Conference on Frontiers of Information Technology*, Dec 2014, pp. 159–164.

[75] Jie Cui, Liusheng Huang, Hong Zhong, Chinchen Chang and Wei Yang, "AN IMPROVED AES S-BOX AND ITS PERFORMANCE ANALYSIS," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 5(A), pp. 2291–2203, May 2011.

[76] Das, S., "Generation of AES-like 8-bit Random S-Box and Comparative Study on Randomness of Corresponding Ciphertexts with Other 8-bit AES S-Boxes," in *Intelligent Computing, Networking, and Informatics*, D. P. Mohapatra and S. Patnaik, Eds. New Delhi: Springer India, 2014, pp. 303–318.

[77] Z. H. Xian and S. L. Sun, "Study on Test for Structure of S-boxes in Rijndael," in *2010 Second International Workshop on Education Technology and Computer Science*, vol. 3, March 2010, pp. 84–86.

[78] Xinmiao Zhang and Parhi, K.K., "On the Optimum Constructions of Composite Field for the AES Algorithm," *IEEE Transactions on Circuits and Systems II: Express Briefs,*, vol. 53, no. 10, pp. 1153–1157, Oct 2006.

[79] M. M. Wong and M. L. D. Wong, "A high throughput low power compact AES S-box implementation using composite field arithmetic and Algebraic Normal Form representation," in *2nd Asia Symposium on Quality Electronic Design (ASQED)*, Aug 2010, pp. 318–323.

[80] N. Shanthini, P. Rajasekar and H. Mangalam, "Design of low power S-Box in Architecture Level using GF," *International Journal of Engineering Research and General Science (IJERGS)*, vol. 2,Issue. 3, pp. 268–276, 2014.

[81] M. Wong and M. Wong, "New lightweight AES S-box using LFSR," in *2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS),*, Dec 2014, pp. 115–120.

[82] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," in *Advances in Cryptology ASIACRYPT 2001*, ser. Lecture Notes in Computer Science, C. Boyd, Ed. Springer Berlin Heidelberg, 2001, vol. 2248, pp. 239–254.

[83] D. Canright, "A Very Compact S-Box for AES," in *Cryptographic Hardware and Embedded Systems CHES 2005*, ser. Lecture Notes in Computer Science, J. Rao and B. Sunar, Eds. Springer Berlin Heidelberg, 2005, vol. 3659, pp. 441–455.

[84] I. Hussain, T. Shah, M. A. Gondal, and W. A. Khan, "Construction of Cryptographically Strong 8x8 S-boxes 1," *World Applied Sciences Journal*, vol. 13 (11), pp. 2389–2395, 2011.

[85] J. A. Clark, J. L. Jacob, and S. Stepney, "The design of S-boxes by simulated annealing," *New Generation Computing*, vol. 23, no. 3, pp. 219–231, 2005. [Online]. Available: http://dx.doi.org/10.1007/BF03037656

[86] W. Millan, "How to Improve the Nonlinearity of Bijective S-Boxes," in *Proceedings of the Third Australasian Conference on Information Security and Privacy*, ser. ACISP '98. London, UK, UK: Springer-Verlag, 1998, pp. 181–192.

[87] N. Nedjah and L. d. M. Mourelle, "Designing Substitution Boxes for Secure Ciphers," *Int. J. Innov. Comput. Appl.*, vol. 1, no. 1, pp. 86–91, Apr. 2007.

[88] M. Kim, J. Ryou, Y. Choi, and S. Jun, "Low Power AES Hardware Architecture for Radio Frequency Identification," in *Advances in Information and Computer Security*, ser. Lecture Notes in Computer Science, H. Yoshiura, K. Sakurai, K. Rannenberg, Y. Murayama, and S. Kawamura, Eds. Springer Berlin Heidelberg, 2006, vol. 4266, pp. 353–363.

[89] Y. Eslami, A. Sheikholeslami, P. 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, no. 1, pp. 43–56, Jan 2006.

[90] T. Sharma and R. Thilagavathy, "Performance analysis of advanced encryption standard for low power and area applications," in *2013 IEEE Conference on Information Communication Technologies (ICT)*, April 2013, pp. 967–972.

[91] J.-P. Kaps and B. Sunar, "Energy Comparison of AES and SHA-1 for Ubiquitous Computing," in *Emerging Directions in Embedded and Ubiquitous Computing*, ser. Lecture Notes in Computer Science, X. Zhou, O. Sokolsky, L. Yan, E.-S. Jung, Z. Shao, Y. Mu, D. Lee, D. Kim, Y.-S. Jeong, and C.-Z. Xu, Eds. Springer Berlin Heidelberg, 2006, vol. 4097, pp. 372–381.

[92] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," in *Advances in Cryptology ASIACRYPT 2001*, ser. Lecture Notes in Computer Science, C. Boyd, Ed. Springer Berlin Heidelberg, 2001, vol. 2248, pp. 239–254.

[93] Morioka, Sumio and Satoh, Akashi, "An Optimized S-Box Circuit Architecture for Low Power AES Design," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, ser. Lecture Notes in Computer Science, B. Kaliski, e. Ko, and C. Paar, Eds. Springer Berlin Heidelberg, 2003, vol. 2523, pp. 172–186.

[94] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong Authentication for RFID Systems Using the AES Algorithm," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, ser. Lecture Notes in Computer Science, M. Joye and J.-J. Quisquater, Eds. Springer Berlin Heidelberg, 2004, vol. 3156, pp. 357–370.

[95] H. S. Deshpande, K. J. Karande, and A. O. Mulani, "Efficient implementation of AES algorithm on FPGA," in *2014 International Conference on Communications and Signal Processing (ICCSP),*, April 2014, pp. 1895–1899.

[96] M. H. Rais and S. M. Qasim, "Efficient hardware realization of advanced encryption standard algorithm using Virtex-5 FPGA," *IJCSNS International Journal of Computer Science and Network Security*, vol. 9, no. 9, pp. 59–63, 2009.

[97] T. Hoang and V. L. Nguyen, "An Efficient FPGA Implementation of the Advanced Encryption Standard Algorithm," in *2012 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF),*, Feb 2012, pp. 1–4.

[98] P. B. Ghewari, J. Patil, and A. Chougule, "Efficient hardware design and implementation of AES cryptosystem," *International Journal of Engineering Science and Technology*, vol. 2, no. 3, pp. 213–219, 2010.

[99] D. Bai and L. Li, *New Impossible Differential Attacks on Camellia.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 80–96. [Online]. Available: https://doi.org/10.1007/978-3-642-29101-2_6

[100] E. B. Kavun and T. Yalcin, "A pipelined camellia architecture for compact hardware implementation," in *21st IEEE International Conference on Application-specific Systems, Architectures and Processors ASAP 2010*, July 2010, pp. 305–308.

[101] B. Wang and L. Liu and C. Deng and M. Zhu and S. Yin and S. Wei, "Against Double Fault Attacks: Injection Effort Model, Space and Time Randomization Based Countermeasures for Reconfigurable Array Architecture," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1151–1164, June 2016.

List of Publications

Journal Publications

- Published Papers:

1. Bhoopal Rao Gangadari, Shaik Rafi Ahamed *"Programmable Cellular Automata based low power Architecture to S-Box : An Application to WBAN"*, in Springer, Circuits, Systems, and Signal Processing, vol. 37, pp.1116-1133, 2017.
2. Bhoopal Rao Gangadari, Shaik Rafi Ahamed *"Low Power S-Box Architecture for AES Algorithm using Programmable Second Order Reversible Cellular Automata: An Application to WBAN"*, in Springer, Journal of Medical Systems, vol. 40, no.12, pp.257-269, Dec. 2016.
3. Bhoopal Rao Gangadari, Shaik Rafi Ahamed *"Design of cryptographically secure AES like S-Box using second-order reversible cellular automata for wireless body area network applications"*, in IET, Healthcare Technology Letters, 3, pp. 177-183, 2016..
4. Hemangee K, Rao, G. Bhoopal and and Arshi, Sharique and Trivedi, Gaurav *"A Security Framework for NoC Using Authenticated Encryption and Session Keys"*, in Springer, Circuits, Systems, and Signal Processing, vol. 32, no 06, pp.2605-2622, 2013..

Conference and Workshop Publications

1. Bhoopal Rao Gangadari, Shaik Rafi Ahamed *"FPGA implementation of compact S-Box for AES algorithm using composite field arithmetic"*, 2015 Annual IEEE India Conference (INDICON), New Delhi, 2015, pp. 1-5.
2. Bhoopal Rao Gangadari, Shaik Rafi Ahamed *"Analysis and algebraic construction of S-Box for AES algorithm using irreducible polynomials"*, 2015 Eighth International Conference on Contemporary Computing (IC3), Noida, 2015, pp. 526-530.

3. Bhoopal Rao Gangadari, Shaik Rafi Ahamed3, R. Mahapatra and R. K. Sinha *"Design of cryptographically secure AES S-Box using cellular automata"*, 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), Visakhapatnam, 2015, pp. 1-6.
4. Bhoopal Rao Gangadari, Shaik Rafi Ahamed *"Low Hardware Complexity Encryption Algorithm using 1st Order 1-D programmable Linear Cellular Automata"*, in Proc. International Conference on Signal Processing and Integrated Networks, SPIN 2017. pp. 385-389.
5. Bhoopal Rao Gangadari, Shaik Rafi Ahamed *"FPGA Implementation of Hybrid Linear Cellular Automata based Encryption Algorithm"*, in Proc. International Conference on Circuits and Signal Processing, ICCSP 2017. pp.281-285.

Manuscripts to be Communicated

1. Bhoopal Rao Gangadari , Shaik Rafi Ahamed *"Ultra Low power Consumption Hybrid Second Order Cellular Automata based Encryption Algorithm"*, to be submitted in IEEE Trans on Information Forensic and Security.
2. Bhoopal Rao Gangadari , Shaik Rafi Ahamed *"Low Power F function Architecture for Camellia Algorithm using Second Order Cellular Automata"*, to be submitted in IEEE Trans on Very Large Scale Integration .
3. Bhoopal Rao Gangadari , Shaik Rafi Ahamed *"Low Power F function Architecture for Camellia Algorithm using Linear Programmable Cellular Automata"*, to be submitted in IEEE Trans on Multi-Scale Computing Systems.