A Thesis Report

on

Optimization of CLEFIA Algorithm for Information Security in E-Healthcare Devices

Dissertation Submitted in Partial Fulfillment of the Requirement for the Award of the Degree of

MASTER OF TECHNOLOGY

in

VLSI DESIGN

Submitted By

Isha Bhardwaj

Roll No. 601562011

Under Supervision of

Mrs. Manu Bansal

(Assistant Professor, ECED)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY, PATIALA, PUNJAB

JULY, 2017.

DECLARATION

I, Isha Bhardwaj hereby declare that the work presented in this thesis entitled "Optimization of CLEFIA Algorithm for Information Security in E-Healthcare Devices" in partial fulfillment of the requirement for the award of degree of Master of Technology submitted at Electronics and Communication Engineering Department, Thapar University, Patiala is an authentic record of work carried out under supervision of Mrs. Manu Bansal (Assistant Professor, ECED, Thapar University, Patiala). The matter presented in this this has not been submitted either in part or full to any other university or institute for the award of any other degree.

Isha Bhardwaj

Date: 10-08-2017

ISHA BHARDWAJ

Roll No. 601562011

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

Manu Bansal

Date: 10/08/17

Mrs. Manu Bansal

Assistant professor

ECED, TU, Patiala

ACKNOWLEDGEMENT

"The successful completion of any task would be incomplete without accomplishing the people who made it possible and whose constant guidance and encouragement secured the success", I take this opportunity to express my gratitude to **Mrs Manu Bansal, Assistant Professor**, Electronics and Communication Engineering Department, Thapar University, Patiala for her positive and excellent guidance, constant encouragement, keen interest, invaluable co-operation, generous attitude and support throughout this work. I am truly very fortunate to have the opportunity to work with her.

I am also thankful to our **Head of the Department, Dr. Alpana Aggarwal** as well as entire faculty and staff of Electronics and Communication Engineering Department, for providing us adequate environment in carrying the work.

I would like to thank my parents for their constant support and all those people who directly or indirectly helped me in the process and contributed towards this work.

Isha Bhardwaj

601562011

ABSTRACT

The advancement in technology has led to the emergence one of the best innovation named Internet of Things (IoT). It enables objects with RFID, smart sensors, communication technologies, and Internet protocols to allow them to communicate with each other and take smart decisions. The IoT has a variety of application domains, including E-Healthcare. E-Healthcare aims in providing instant health solutions to patients' by connecting them directly to the medical professionals. Electronic sensors are used to collect medical data from the patient's body and transmit it to the healthcare system. One of the major concerns regarding smooth application of E-Healthcare is information security. It is essential to ensure trust and data secrecy from the starting throughout the medical treatment to prevent any unauthorized access or unnecessary interruption as this may lead to wrong medical treatment of patients'. Therefore, data encryption is necessary but due to the limitations in device area, computing complexity and power consumption of smart devices, the performance and efficiency of conventional algorithms is not up to mark. In this work study of NIST recommended lightweight encryption algorithm CLEFIA is done. The flow of algorithm is thoroughly studied along with the techniques that can be used to optimize it. The main aim is to optimize the algorithm in terms of area utilisation and efficiency by modelling S-Boxes, Diffusion Matrices and use of Lookup Tables. The work is done at both software and hardware level. The compilation of CLEFIA algorithm is done on GCC compiler and simulation on TSIM simulator. The optimization in terms of cycle count and cycles per instruction at software level is achieved. There is 75% reduction in cycle count and 5.8% improvement in cycles per instructions. At hardware level optimized synthesized code is generated for xc5vlx50t-3 board of Virtex-5 Family and reduction of 29% in area is seen with increase of 33% in efficiency.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABREVIATIONS AND SYMBOLS

Abbr.	Name
IoT	Internet of Things
RFID	Radio Frequency Identification
AES	Advance Encryption Standard
DES	Data Encryption Standard
RSA	Rivest, Adi Shamir, and Adleman
ECC	Elliptic Curve Cryptography
GFN	General Feistel Network
SPN	Substitution-Permutation Network
ECB	Electronic Code Book Mode
CBC	Cipher Block Chaining Mode
CFB	Cipher Feedback Mode
OFB	Output Feedback Mode
NIST	National Institute of Standards and Technology
DSM	Diffusion Switching Mechanism
GF	Galois Field
GCC	GNU Compiler Collection
FPGA	Field Programmable Gate Array
LUT	Look Up Table
RTL	Register Transfer Logic

Symbols	Meaning
$0\times ab$	Hexadecimal notation of ab
a| b	Concatenation of a and b
a⊕b	Bitwise exclusive OR of a and b, Addition in GF(2^m)
a·b	Multiplication in GF(2^m)
a≪b	b-bit left shift operation
a≫b	b-bit right shift operation

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW OF INTERNET OF THINGS

The notion of Internet of Things has revolutionized technology. It has become more important to the world than any other technology in history. It started with the idea of linking radio frequency identification and sensor networks to Internet leading new opportunities and visions. This framework has allowed direct machine to machine communication over the internet, allowing them to participate in the web as a vast network of independent, self-establishing devices. The basic notion of IoT is to connect things together so that they can communicate with each other and communicate with people too. The core concept of IoT is that devices used in our day to life be equipped with identification, sensing, processing and networking possibilities which enables them to connect with other devices over internet share information and accomplish some valuable goals [1].

IoTs have applications in almost every aspect of our daily life. It can be incorporated in various domains like Intelligent Transport System design where such devices can monitor traffic jams, vehicle rules violation, report accidents, *etc.* In design of Smart Cities where these devices monitor air quality, efficient lighting up of the city, monitor environmental changes. Provide Smart Security, where surveillance of city, infrastructure, alarming, tracking of people can be done. Designing of Smart Homes is also one of the major application of IoTs, home safety and security of belongings, energy consumption management, communication with appliances be the few features. In medical sector, it can be used to monitor patient's health parameters, keep track of their medicine intake, activities, provide support for independent living and improve quality of living [2], Figure 1.1 gives more detail into the applications.

1.2 HEALTHCARE AND IOT

In health Industry, IoT provides vast options to take care of people. Objects or People can be equipped with RIFD Tags, actuators, wireless medical sensors, etc. These devices give access to the patient's health as they keep monitoring it continuously. They have capabilities to connect to other machines, connect doctors to machine and vice-versa. This system provides an intelligent communication between machines, smart devices, humans in order to assure an effective healthcare system. Few of the applications of IoTs in Healthcare are Ambient Assistant Living, Glucose Level Sensing, Electrocardiogram Monitoring, Blood Pressure Monitoring, Medication Management, Body Temperature Management, Drugs Monitoring, Healthcare solutions using Smartphones, [3] etc. Figure 1.2 shows some of the IoT healthcare

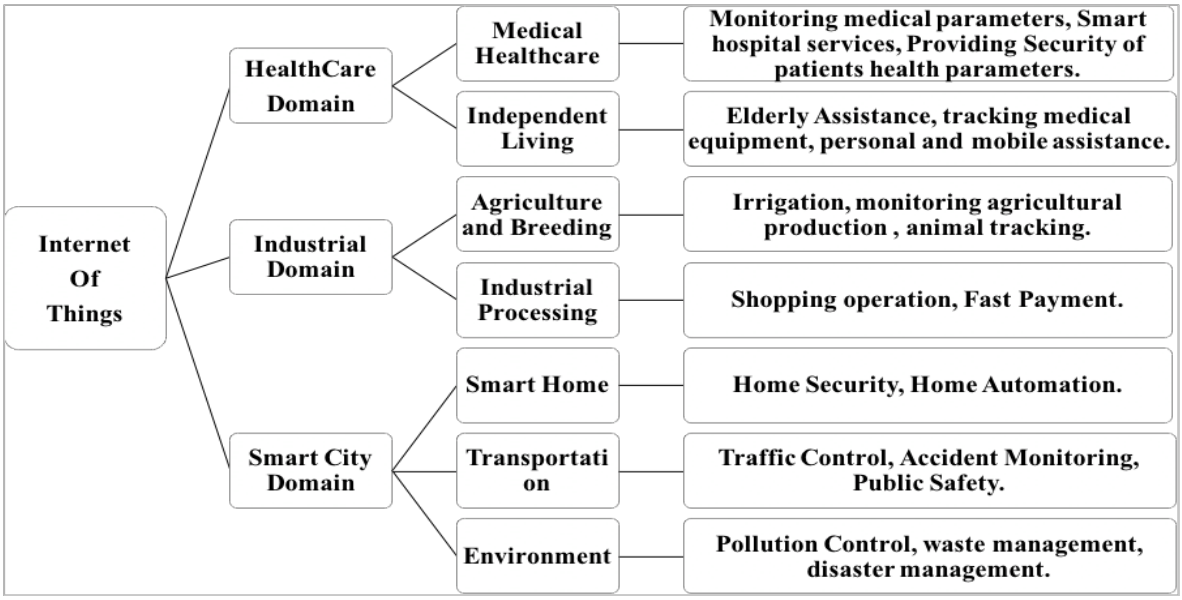

Figure 1.1: Internet of Things: Applications

products. The dependence of Healthcare on IoT is rapidly increasing. It is not only improving the quality of care but also reducing the cost. Based on individuals unique behavioral, social, cultural and biological characteristics, they can be provided with personalized healthcare i.e. the right care for the right person at the right time. IoT promises to manage a digital identity for each individual. The classifications of IoT based Healthcare systems are clinical Care and Remote Monitoring [4].

Figure 1.2: IoT Healthcare products [3]

1.2.1 Clinical Care

Here monitoring of hospitalized patients is done, who require constant close attention. These monitoring systems collect information from sensors, which is then analyzed and stored. After this the information is sent to the doctors/caregivers wirelessly for further analysis. A constant check can be kept on the patient and any irregularity can be taken care of immediately. This way the quality of care is enhanced through constant attention which eliminates the need for a caregiver to actively engage in taking care and collecting data, and also reduces the cost of care.

1.2.2 Remote Monitoring

The lack of immediate health monitoring leads to so many health risks and this problem is faced all over the world. But IoTs have made it easier to tackle such problems. Small yet powerful wireless solutions along with internet make it possible to provide health solutions and constant monitoring of patients. The patient's health data can be collected through Devices and then transmitted securely over the internet to the medical facilities where the medical professionals can immediately recommend health solutions.

1.3 IOT HEALTHCARE ARCHITECTURE

The IoT healthcare architecture explains the arrangement of different technologies that are used in the smooth functioning of it. The work here is not just to transfer information between different parties, but perform multiple operation and that too in real time. The architecture is divided into layers which perform different functions. The first component is the Physical Layer. It is equipped with sensors and RFID tags which interact with the physical environment and collect data. The Data is acquitted by various sensors that measure temperature, ECG, etc. The next layer is Network Layers which is responsible for establishing network and communicating the information. Network Layer supports communication technologies like WIFI, Bluetooth, Zigbee, etc. Next is the Middleware Layer which establishes connection between the heterogeneous components and related applications, and manage them. They provide access to the information and manage data. Last layer is the Application Layer. It is responsible to provide application specific services to the users, in this case Healthcare [5]. Figure 1.3 explain the basic IoT Healthcare architecture for patient monitoring. The medical wireless sensors collect data of the patient. This data is then sent it through the various layers of the IoT architecture. Each layer performs its respective work and the data reaches the Medical service providers.

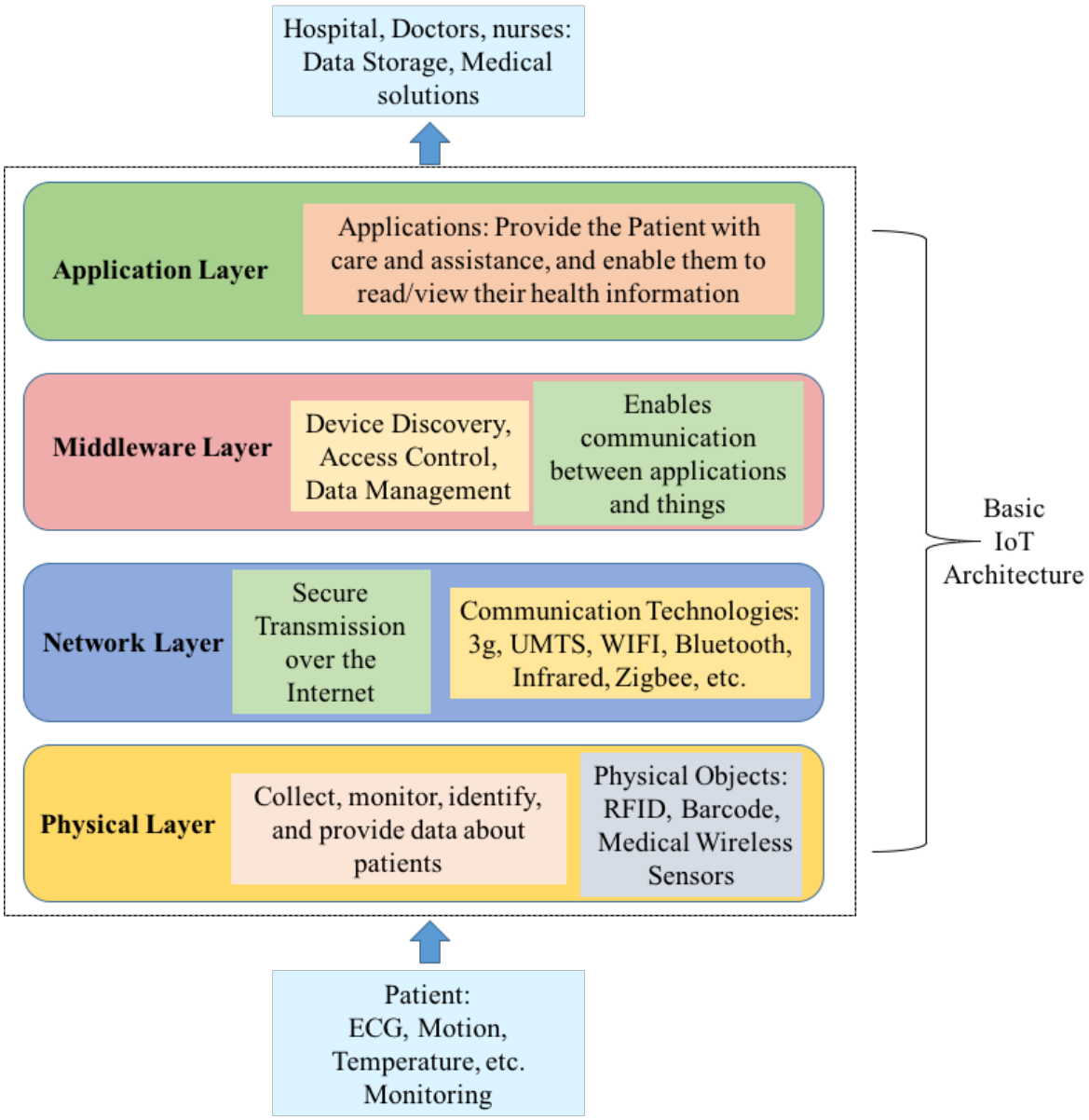

Figure 1.3 Healthcare architecture for patient monitoring

1.4 IOT HEALTHCARE SECURITY

The IoT healthcare industry is rapidly growing. In coming years' medical sector is expected to adopt various solutions of IoTs and provide services through it. The new E-Healthcare IoT devices and applications will make it easier to provide medical health anywhere and anytime. Since these devices and applications are going collect a large amount of vital information such as personal health data and transfer on public networks may lead to many security issues. These smart devices being connected to global information networks put themselves under the threat of attackers. To enable the full implementation of IoT in healthcare domain, it is very important to first analyze, identify the different features of IoT security and privacy such as security requirements, threats models, vulnerabilities and countermeasures that can be taken for smooth working of the system. Table 1.1 highlights some features of IoT Healthcare security [3].

Table 1.1: Healthcare Security Features

Features of IoT Healthcare Security		Parameters	Description
Security Requirement	To achieve secure services for information sharing	Confidentiality	Ensures information is inaccessible to unauthorized users
		Integrity	Ensures medical data is not altered
		Authentication	Ensures identity of the source
		Data Freshness	Data collected and sent is new
		Authorization	Ensures that only authorized nodes are accessible for services
		Resiliency	Ensures even if a device is compromised, data on it is still protected
Security Challenges	Providing security solution that meet the technology parameters	Computational Limitations	Since the devices are resource constrained, it becomes mandatory to use security solution that require less memory, consume less power, and are fast
		Memory Limitations	
		Energy Limitations	
		Mobility	Since devices are mobile, they may enter different networks, therefore developing a security protocol which is acceptable everywhere is a challenge.
Threats and Attacks	Attacks based on information theft and modification	Eavesdropping	Attacker intercepts the information threatening data privacy and confidentiality
		Alteration	Attacker gains unauthorized access to health data and tampers with them to create confusion
		Fabrication	Attacker forges messages by injecting false information to threaten message authenticity
		Message Replay	Attacker replays existing data and threatens data freshness
		Man-in-the-middle	Attacker alters communication between two parties who are unaware of it.

Healthcare data are very critical to understand and study a patient's condition so that he can be given best possible treatment. The quality of data is important as it will help the medical professionals to provide ideal treatment. But electronic processing of data encourages misuse of information. Data Breach and Data abuse are few types of security issues which need to be taken care of. In Data Breach the Sensitive and confidential information can be copied, stolen, viewed, transmitted, used by unauthorized individuals. Data abuse can lead to various frauds such as health insurance frauds, medical fraud, drug fraud. In health insurance fraud one can deceive, conceal or misinterpret data that may bring health care benefits to the individual. Selling of illegal drugs or costly drugs at lower prices can be done in Drug Fraud. Medical Frauds can lead to compromise of crucial information of patient, his medical parameters can be modified, which will result to wrong treatment and loss of life. To Prevent this thorough

study of IoT architecture is needed as to where we can provide security to our data so that it is not misused [5].

1.5 OVERVIEW OF CRYPTOGRAPHY

The basic service provided by cryptography is to share information between two participants over an unsecured channel in a way that the information is not readable by other parties. Cryptography is a science of making a system that provides Information Security. It deals with securing of digital data. The Communication Model consists of two parties, who are communicating over a public channel. For example, A and B are two parties who are sharing private information with each other and E is eavesdropping on them. Now A will manipulate his information in a way that no one will be able to decode it and it reach B safely. Even If E gets hold of the data he will not be able to decipher it and the information will be useless for him [6]. Figure 1.4 shows the basic communication model.

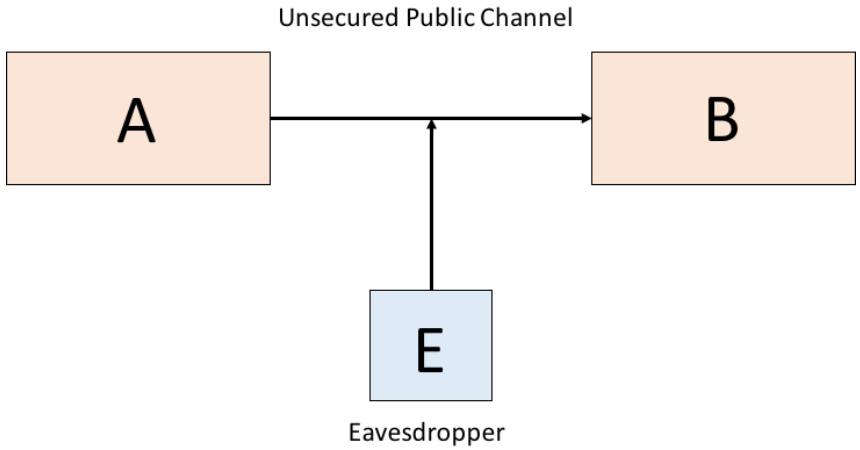

Figure 1.4: Basic Communication Model

The Objective of cryptography is to provide following fundamental services that keep the information hidden from any unknown source [6].

- Authentication: it provides the identity of the person from whom the information has originated. It confirms the receiving party that the data received by them is from an identified and verified sender.
- Confidentiality: It is the fundamental service provided by Cryptography. It ensures that the message sent by the originator cannot be read by any other party except the intended receiver. It is also referred to as secrecy or privacy. We can achieve it by using mathematical algorithms for data encryption.
- Integrity: it is the service which identifies any change in the received data. While transmitting the data may get altered, modified intentionally by unknown source or by

mistake. It cannot prevent the manipulation of data but it can detect whether the data has been modified.

- Non-repudiation: This service is the mechanism that proves whether the data is sent by the actual sender and not by any other party. It is a surety that the originator cannot deny the creation or transmission of the data to the recipient.

Cryptography can be used in many ways like data encryption, authentication, digital signature. The Figure 1.5 explains the basic Cryptographic Model. Some of the basic terms related to cryptography [7].

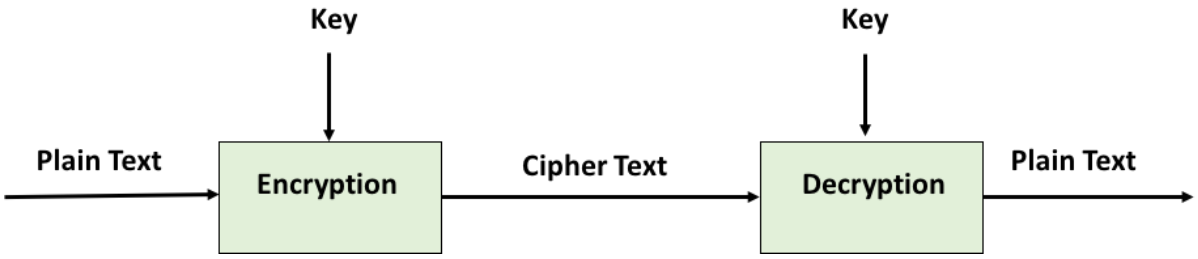

Figure 1.5: Basic Cryptographic Model

- Plain Text: It is the original message the sender wants to communicate to the other person.
- Cipher Text: This message is a scrambled version of plain text which is communicated to the other party. It is scrambled so that it is unreadable for unauthorized party. Cipher Text is shared over the unsecured channel safely.
- Key: It is a string of data that is used to encrypt and decrypt the message. They are kept secret. Their strength varies according to their length. String with more bits is considered strong and tough to break.
- Encryption: It is the technique in which we scramble the Plain Text into Cipher Text. It is used in cryptography to communicate confidential data over unsecure channel. This takes place at the side of sender. It requires one secret key and an algorithm to perform encryption over Plain Text.
- Decryption: It is opposite of Encryption. Here Cipher text is decoded back to Plain text in order to get the original message back. This is done at the receiver side.
- Interceptor: also called attacker, is the one who tries to find the Plain Text. He can use various techniques to do so, generally called Cryptographic attacks.
- Cryptanalysis: It is the technique of analysis and breaking of codes.

In todays world where large amount of confidential data is generated and shared among organizations, there is a need to protect it from attackers. Security of data has become crucial

nowadays. It helps to ensure the privacy of a user from attackers. Therefore, to protect data various algorithms have been designed who offer different level of security.

1.6 TYPES OF CRYPTOGRAPHY

There are three types of cryptographic schemes, Figure 1.6.

- Symmetric Key Cryptography (Secret key Cryptography): Technique where single Key is used for both encryption and decryption.
- Asymmetric Key Cryptography (Public Key Cryptography): Technique which uses two different keys, one for encryption and the other for decryption.
- Hash Function: This technique uses mathematical functions to encrypt data. It is irreversible technique.

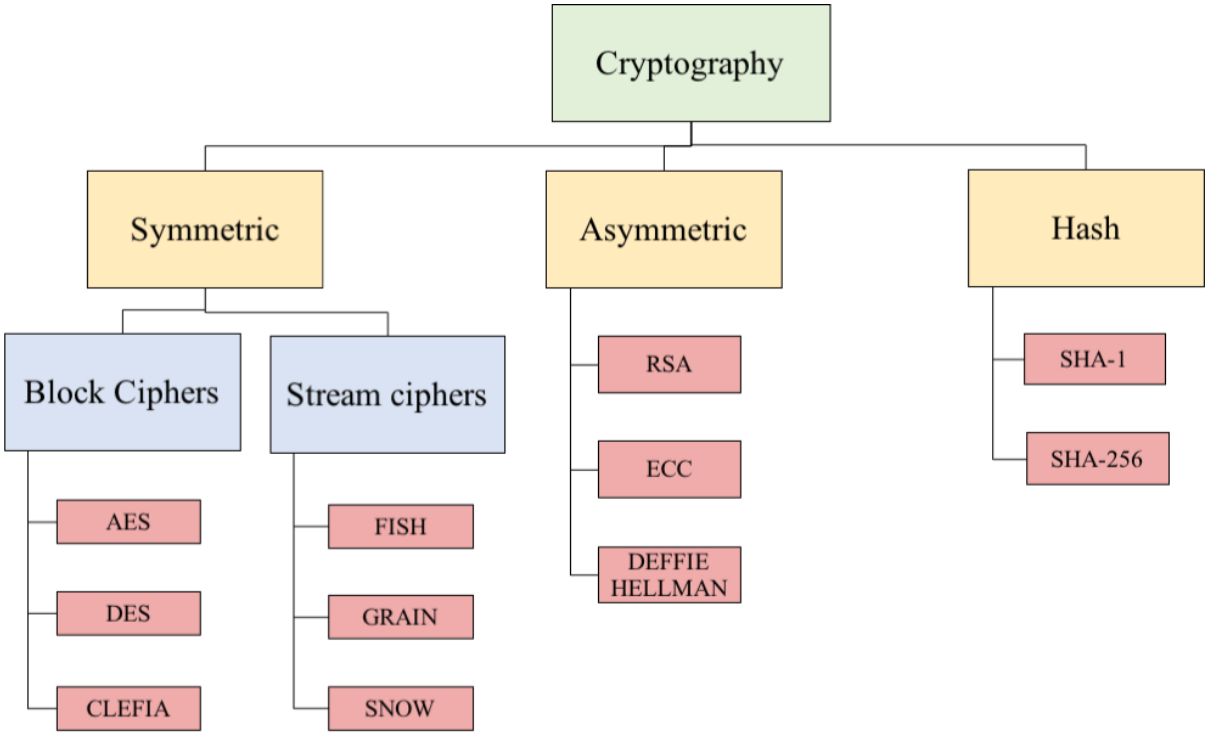

Figure 1.6: Classification of Cryptography

1.6.1 Symmetric key Cryptography

In symmetric key cryptography, the encryption and decryption is done using the same key. This is also called Secret key cryptography. The sender encrypts the plain text using the secret key and transmits the cipher text to the receiver. The receiver uses the same key to decrypt the cipher text to get plain text. Since single is being used it is very important for both the parties to have the key. Figure 1.7 explains the symmetric key encryption.

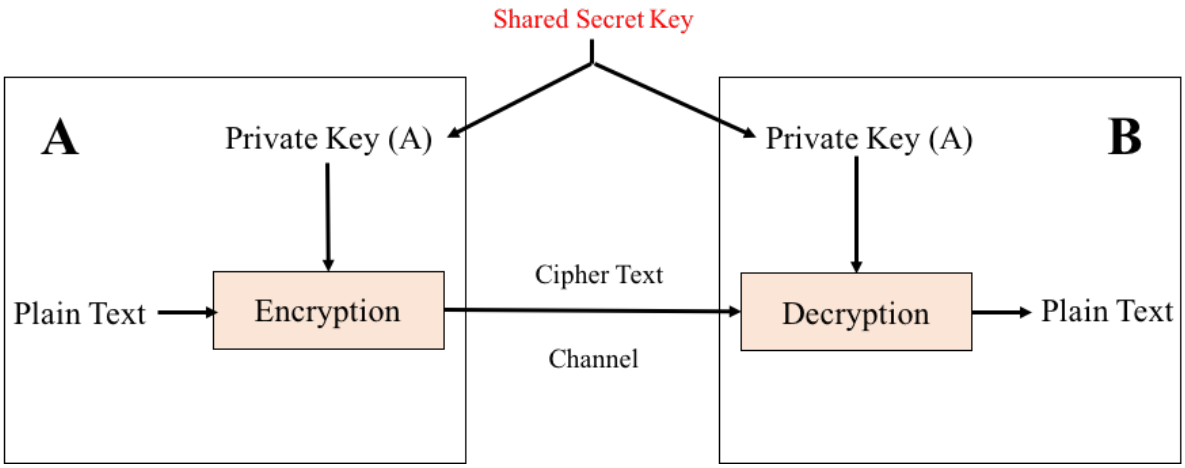

Figure 1.7: symmetric Key Cryptography Model

There are two types of ciphers used in this technique.

1.6.1.1 Stream Ciphers

Operations are done on a single bit (byte) of data at a time and some form of feedback function is implemented in order to keep the key changing. Here same plain text will encrypt to create a different cipher text.

1.6.1.2 Block Ciphers

Block ciphers divides plaintext (input) into blocks of bits of same length and encrypt that each block using same key. Here same block of plain text will create same block of cipher text when same key will be used on them. Modern Block Ciphers are of iterative nature. They perform a function number of times. This function has linear and non-linear layers which change the data on every iteration using round keys generated from main secret key. There are two types of structures of block ciphers:

- Feistel Network (FN): Encryption in this structure takes place in multiple rounds. These rounds consist of substitution step which is followed by permutation. The input plain text is divided into halves, where the left half is operated (XORed) with the result of a function that takes two inputs, secret key and right half of plain text. After this operation, the halves are swapped for the next round. This goes till the specified number of rounds, which depends on the respective algorithm. The decryption process is reversible of encryption in this structure, where cipher is taken as input and round keys are reversed.

  There are many types of Feistel structures, discussed in the Table 1.2. General Feistel Network is denoted by GFN(d,r), where “d” is number of branches (number of divisions of input) and “r” is the number of rounds. Figure 1.8 shows the basic illustration of the structure.

Table 1.2: Types of Feistel Networks

No.	FN	Description
1	Balanced FN	The input text is divided into equal number of bits for each branch.
2	Unbalanced FN	The input text is not divided equally.
3	Alternating FN	The rounds alternate the output of F-functions.
4	GFN(d, r) (d=4) Type-1 Type-2 Type-3	They are 4-branch structures. Type-1 uses one F-function, Type-2 uses two F-functions and type-3 uses three F-functions.

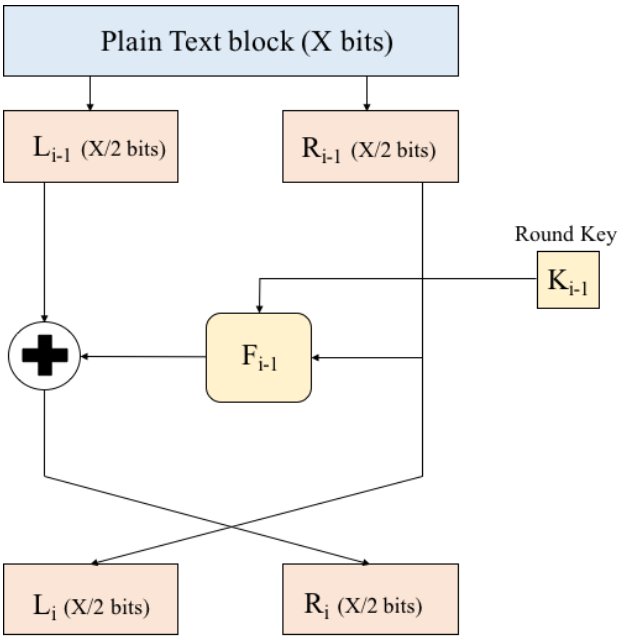

Figure 1.8: Feistel Structure for one round

- Substitution-Permutation Network (SPN): In this structure, the plaintext block and key are taken as input. They are passed through alternating layers of substitution boxes (S-boxes) and permutation boxes (P-boxes), these boxes provide confusion and diffusion respectively. In S-box the input bits are substituted by other bits and in P-box permutation of the output bits of S-box is done. The output of P-box is fed into the next round as input. Decryption is done using the inverse of S-boxes and P-boxes. Figure 1.9 shows the implementation of SPN structure for one round.

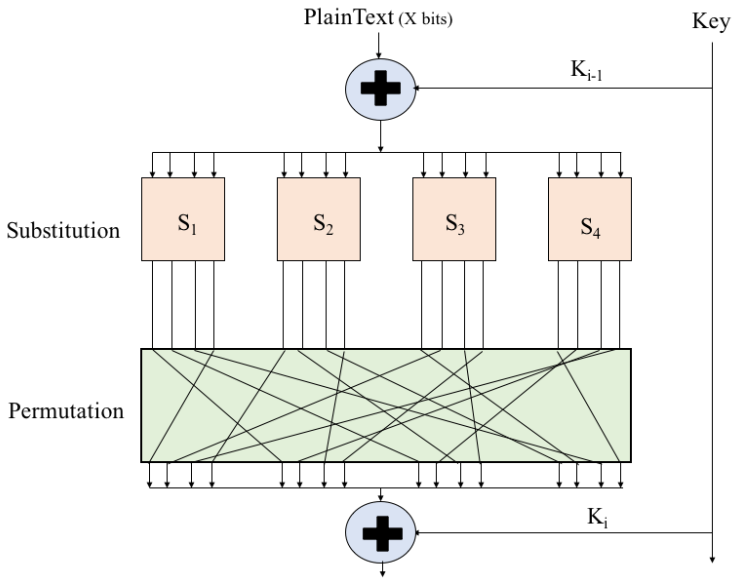

Figure 1.9: SPN structure for one round

Block ciphers have different modes of operation in which they can work. These modes result in achievement of different properties which enhance the security of the ciphers. The following modes are the most significant:

- Electronic Code Book Mode (ECB): In block cipher the data is divided into blocks of fixed size. Since plaintext is more than the block size, it is divided into many sequential data blocks and each block is operated one at a time. ECB is the simplest application, here if two plaintext blocks contain same data then their cipher text blocks will also be same provided same key is used on both of them.
- Cipher Block Chaining Mode (CBC): In this mode of operation feedback mechanism is used. The current plain text block is XORed to the previous cipher text block and the result is encrypted with the key. Advantage of CBC over ECB mode is that two similar plain text blocks will never result in same output.
- Cipher Feedback Mode (CFB): It is a block cipher implementation, which makes the cipher into a self-synchronising stream cipher. In this mode, we can encrypt data in small blocks i.e. less than the block size.
- Output Feedback Mode (OFB): The block cipher behaves as a synchronous stream cipher in OFB mode. A key stream block is generated which is XORed with plaintext to get cipher text. This key stream block is then fed back instead of cipher text block. For initial step an initialisation vector is needed.

Some examples of Symmetric key ciphers are AES (Advance Encryption Standard, SPN structure), DES (Data Encryption Standard, FN structure), CLEFIA, RIVIST Ciphers such as RC1, RC2, SNOW etc. There are many advantages of symmetric key cryptography like [8],

- The process of encryption is simple, fast and computational power is low.
- One is needed for both encryption and decryption and they are relatively short.
- Data throughput rates are high.

There are some disadvantages to this system like,

- Key establishment and distribution: Before starting the communication between two parties, both of them should agree on a key and share it via a secure mechanism.
- Scalability: Number of keys required as compared to the number of participants in the message exchange equals about the square of the number of participants.
- Authentication: Symmetric algorithms cannot be used for Digital Signatures.

These drawbacks create constraints on modern day communication. Today information is exchanged between many non-familiar and non-trusted parties therefore some highly secure schemes are needed. These limitations give rise to Asymmetric key cryptography.

1.6.2 Asymmetric Key Cryptography

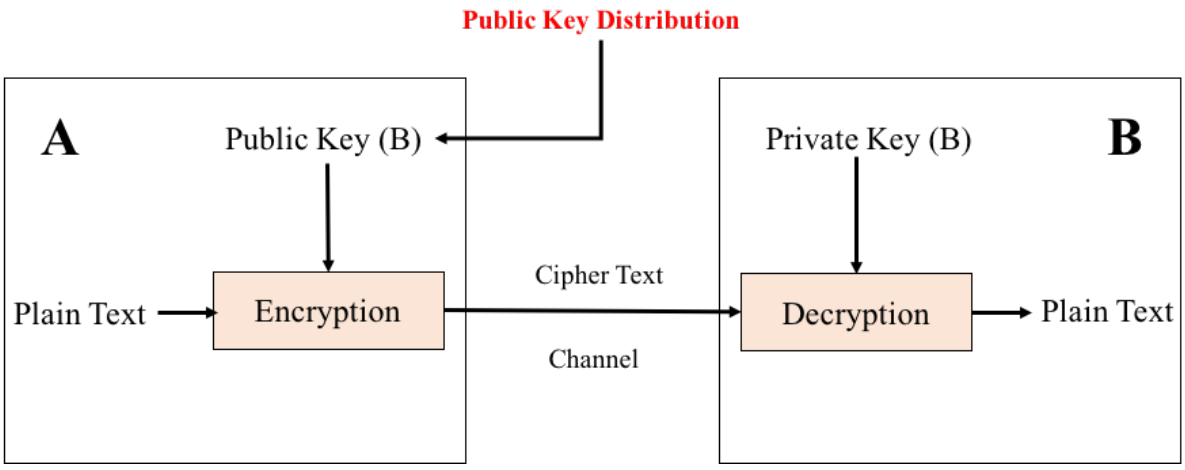

Figure 1.10: Asymmetric Key Cryptography Model

In this encryption process, different keys are used for encrypting and decrypting data, Figure 1.10. These keys are mathematically related but knowledge of one key will not lead to the other key easily. One key encrypts and other decrypts data and the order of the key used does not but both are needed to carry out the process. One key is called private key which is not shared with anyone and other is public key which is shared over the medium. It depends upon mathematical functions which are easy to calculate but their inverse function is comparatively hard to compute, example factorization and logarithms. In case of factorization, it is easy to calculate the product of two prime numbers but if only the number is given calculating its prime factors is relatively more difficult. This feature makes Asymmetric key cryptography more

secure. It also provides to solution to problems such as key distribution and management and provision of non-repudiation [6].

Advantages of asymmetric key cryptography are,

- Highly secure mechanism: Their complex calculation makes it difficult to break them.
- Secure Key distribution: Sharing of keys over the channel becomes secure as using private key is kept a secret and without it the cipher text can never be obtained.
- Authentication: Asymmetric ciphers provide authentication using Digital signatures.

Even after so many advantages there are few loop holes in asymmetric scheme. In order to achieve high levels of security large key sizes are used which make the process slow and use more area. Few asymmetric algorithms are RSA, ElGamal, Deffie Hellman, Elliptic Curve Cryptography (ECC). In order to overcome this problem ECC was introduced, it uses small key sizes but provides same level of security when compared to RSA [6]. RSA uses factorization and ECC uses logarithmic concept.

1.7 OVERVIEW OF LIGHTWEIGHT CRYPTOGRAPHY

Lightweight Cryptography was designed to be used for resource constrained devices, these devices have limited power and memory therefore they do not work well with conventional cryptography algorithms. Conventional ciphers work well with devices which can support large memory, complex calculations and where power is no issue for example servers and desktop computers, etc. These ciphers have large block size, key size and rounds which makes it becomes difficult to apply them on small devices as the performance of both device and cipher will get affected. Lightweight ciphers have properties which are suitable to be used in these devices. They have lesser key lengths, small block sizes and less number of rounds. NIST approved the use of Lightweight ciphers for resource constrained devices. Following are the aspects of Lightweight cryptography.

1.7.1 Target Devices

Lightweight cryptography aims at variety of devices like Wireless Sensors Networks, RFID, low power Embedded systems, etc. Some of the devices are mentioned in the table 1.3. It can be seen from the table that the devices do not provide much memory. In order to implement the ciphers on them these limitations should be kept in mind.

Table 1.3: Constrained Devices [9]

Type	CPU	RAM	Flash/ROM
Crossbow TelosB	16-Bit MSP430	10 Kb	48 Kb
RedBee EconoTAG	32-Bit MC13224v	96 Kb	128 Kb
Atmel AVR Raven	8-Bit ATMega1284P	16 Kb	128 Kb
Crossbow Mica2	8-Bit ATMega 128L	4 Kb	128 Kb

1.7.2 Performance Metrics

There is a tradeoff between the performance of ciphers and resources required for a given level of security. Performance is expressed in terms of energy consumption, throughput, latency, and power. The resources mandatory for hardware implementation are expressed as gate area, slices or gate equivalents, and in software implementation they are denoted by registers, RAM and ROM usage. Resources are also signified as cost of the device, more memory usage will lead to increase in the cost of device. Energy and power consumption are significant metrics because of constrained nature of devices. Some of the devices are battery operated, they have limited amount of stored energy. Latency in encryption is the measure of time from the moment plaintext is entered till the output cipher text is achieved. It important for real time applications where we need output as soon as possible therefore it should be high. Moderate Throughput levels are required for Lightweight ciphers. For hardware platforms area is main parameter. It is denoted as slices for FPGAs and GE for ASIC implementations. For a low-cost RFID tag gate count varies from 1000-10000 gates, out which only 200-2000 are available for security purposes. Area and power go hand in hand, more area more power, therefore less area is preferred [10]. In software implementations RAM is used to store intermediate values that are generated while computations and ROM is used to store the code, S-boxes or round keys. This leads to a tradeoff between using stored values in lookup tables or generate them on the fly.

1.7.3 Design Choices for Lightweight ciphers

Many lightweight ciphers are modified versions of conventional ciphers. A lot of ciphers have been proposed keeping Advance Encryption Standard (AES) as standard [11]. These ciphers have been designed by simplifying the conventional ciphers to improve their efficiency, reduce area and power consumption like DESL is a modified version of DES, it uses one S-box in place of eight and the initial and final permutations have been removed for better hardware results. PRESENT is a lightweight block cipher which was specially designed for constrained devices [12]. SIMON and SPECK family of ciphers were designed to be flexible, simple and have good software and hardware performance [13]. Few parameters for lightweight ciphers are discussed below:

- Block Size: Smaller block sizes in comparison to AES are used to save memory, preferably less than 128 bits but below 64 bits may cause plaintext recovery attacks.
- Key Size: Key size taken is also small example PRESENT uses 80-bit key. Small key size leads to key recovery attacks and related key attacks where as large key sizes increase security but also area.
- Rounds: selecting number of rounds is tricky, less rounds will increase computation but decrease security, whereas large number of rounds increase security but increase power consumption too. Complexity of operations in the rounds is simpler than the conventional. 4-bit S-boxes are used in place of 8-bit S-boxes. This results in significant area reduction and less power consumption.

Choosing best parameters for a lightweight cipher is the most tedious task. There is always a tradeoff between cost, security and performance. Large key sizes provide more security but increase cost, small number of rounds increase performance but security is affected, secure hardware performance can be achieved by pipelining architecture but it will increase gate area.

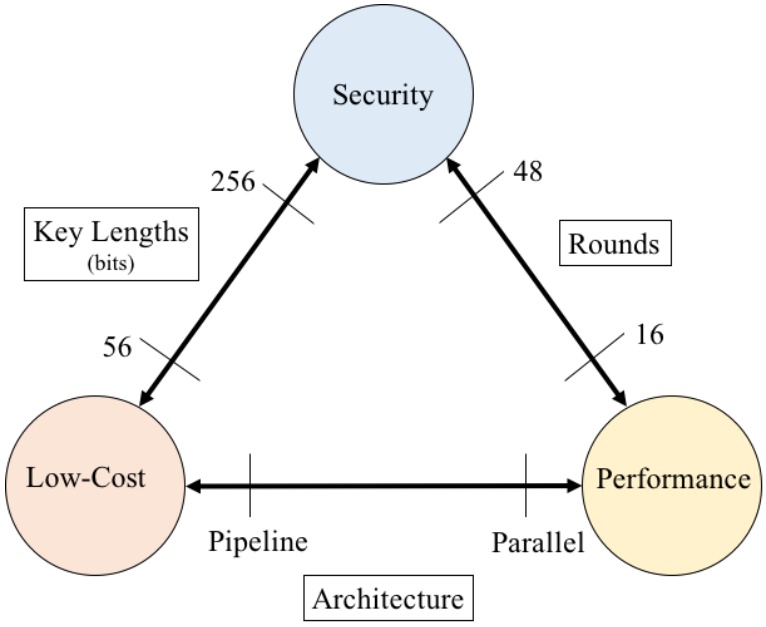

Figure 1.11: Design trade-offs for Lightweight Cryptography [10]

There will always be two out of three design goals which will be optimized and it will be difficult to optimize all three at the same time. Figure 1.11 shows the tradeoffs between the design. Examples of lightweight ciphers are

- Block Ciphers: Present, Clefia, Katan, Simon & Speck
- Stream Ciphers: Trivium, Grain V1, Hummingbird
- Public key ciphers: ECC.

1.8 ORGANIZATION OF THESIS

The thesis is divided into 6 chapters which include Introduction; Literature Review; Problem Formulation; Proposed work; Simulation and Results and Conclusion and Future Scope.

- Chapter 2 is the literature review of work related to e-Healthcare, information security concerns in wireless communication and cryptographic solution for resource constrained devices.
- Chapter 3 presents the research gaps, problem formulation and possible solutions.
- Chapter 4 discusses the proposed algorithm to countermeasure the problems discussed in the problem statement.
- Chapter 5 has the simulation results of the implemented algorithm. Comparison is made between the direct implementation and optimised implementation of the algorithm, evaluating performance based on area and power consumed.
- Chapter 6 summarizes the conclusions drawn in the thesis along with its future scope.

CHAPTER 2
LITERATURE SURVEY

In this chapter, the research done on internet of things and its healthcare application and the limitations faced by them has been discussed. Study of information security issues on the resource constrained devices used in the healthcare applications has been done. Cryptographic solutions that can be applied to overcome the issues have been reviewed.

2.1 SURVEY ON INTERNET OF THINGS AND E-HEALTHCARE

S. M. R. Islam *et al.* [14] discuss about the rise of internet of things (IoTs). As it is spreading widely in many domains such as home automation, transportation, environment *etc.*, it has a major role in health care also. Today everything is connected with internet. Information security has become a major concern nowadays. IoT based healthcare has promising technological, economic, and social prospects. This paper surveys the advancement made in IoT based healthcare domain and also reviews the up-to-date network architectures, applications, and industrial trends. The authors have analysed the security concerns, privacy issues and attack taxonomies from health care perspective. This paper also proposes an intelligent collaborative security model to minimize security risk and discusses various innovations such as ambient intelligence, big data and wearable's that can be used in healthcare domain. The various IoT policies and E-Health regulations are discussed which can benefit people assessing IoT based healthcare technologies. Technologies which can be used, network type, quality of service, business models and healthcare data protection which can facilitate the progress in E-Health have been discussed.

K. Niranjan Devi *et al.* [15] discuss about how how much vital the healthcare data is when it comes to a patient's proper care. It is very important that the data collected by the healthcare devices is accurate and reaches the medical facility for fine treatment. Nowadays with the help of IoTs we can access medical data anywhere and anytime. But this also leads to data abuse. Therefore, security of data is main concern. In this paper, to improve security of data, IoT based secure data routing is done and encryption algorithms are used to provide confidentiality. Comparison of RC5, XTEA, MD5 and SHA2 algorithms is done. Where XTEA turns out good for Encryption on resource constrained devices and MD5 Hashing algorithm for confidentiality.

Jinyuan Sun *et al.* [16] have proposed a secure Electronic Health Record (EHR) system which protects the patients the medical information and provide higher accessibility of patient's

protected health information in case of emergencies which is more effective than the traditional paper record systems. The authors work, Healthcare System for Patient Privacy (HCPP), provides privacy to patient's personal medical information using cryptographic algorithms and wireless network infrastructures which will retrieve information on time for immediate treatment of the patient in emergency condition. Searchable symmetric encryption is used when user's data is being stored on public servers. This system works with all the cryptography parameters, confidentiality, integrity, access control. This system is secure and efficient and has been demonstrated to robust to various attacks.

Lobna Yehia *et al.* [17] discuss about the various security issues in IoT healthcare applications. Some common techniques such as access control limits (logical and physical access), steganography, hashing, cryptography is mentioned which provide counter measures to these issues. They put focus on a hybrid cryptographic technique where multiple ciphers can be used, taking the benefit their strength for providing security to the patient's health information. These techniques can be applied to wireless medical sensors which collect and transmit large amounts of heath data which need to be protected from malicious attacks.

A Boonyarattaphan *et al.* [18] have proposed a security framework that applies effectively on data transmission and authentication for e-Health services. The framework includes efficient protocol architecture. Two risk adaptive authentication techniques, suitable to be used in various scenarios depending on different e-Health services. These services are cost effective solutions. For secure data transmission authors have discussed various cryptographic algorithms with recommended key lengths, which can be used for implementing the authentication mechanisms.

Kim *et al.* [19] discuss about the potential threats in the healthcare systems. they have described security requirements for the systems. They proposed a systematic architecture, which allows the patient to the access to information recorded by sensing nodes in a personal healthcare device.

Xuan Hung Le *et al.* [20] propose a security scheme based on lightweight public key, named Authentication and Access Control based on Elliptic curve cryptography (MAACE). This protocol is based on mutual authentication where the user (healthcare professional) can authenticate to an accessed medical sensor node. It ensures that the data does not goes in hand of unauthorised person and the data is not tampered or originated by a malicious node. MAACE is low memory and more scalable protocol in comparison to the existing public key based schemes.

Pradeep Kumar *et al.* [21] highlight the use of wireless sensor networks in healthcare. The smallest unit of this network is a wireless sensor which performs major operations. These sensors are integrated on various wireless communication motes like Mica2, Telos, etc. Mica2 mote supports a 7.3 Mhz Atmel ATmega128L CPU with RAM of 4 KB and ROM of 128 KB. These sensors are less memory, low power devices with limited computational power therefore come under the category of resource constrained devices. The authors discuss about the possible security threats and privacy issues which prevail on the sensors. Use of encryption mechanisms is highlighted to overcome the need of data security and authentication.

Wasim A *et al.* [22] examined various existing access control models in Sensors. Access control is a security concern faced in healthcare as it allows medical data to be exchanged in the network. These resources should only be accessible to authorised users. Cryptography based access control techniques are highly suitable for web based environments as it offers great level of security. The authors have proposed a new model for access control based on cryptography combined with role based access control. When there are more entities the security level among them is distributed and based on public key infrastructure.

Moshaddique Al Ameen *et al.* [23] discuss about how use of wireless sensor network applications in IoT healthcare is growing with time. They are used in IoT devices to gather and transmit information. Since these devices are wireless in nature their security and privacy are major concerns. Discussion on attacks made on information and their countermeasures is done. The use of security mechanisms like data encryption, data integrity and authentication are discussed.

Devesh C Jinwala *et al.* [24] have implemented ciphers which can be used in wireless sensor networks. WSNs are resource constraint devices therefor communication security is a challenge in them. The ciphers implemented are lightweight in nature i.e. designed especially for resource constrained devices. Optimised implementation and simulation of corrected block Tiny Encryption Algorithm (XXTEA 128-bit key size) is done and compared to algorithm SkipJack (80-bit key size) on MiCA2 motes platform. Overall requirements of SkipJack are higher and XXTEA proved a good alternative to it.

Yi Sheng Shiu *et.al* [25] worked on the security of information exchange in physical layer in wireless network. Attacks and numerous security approaches for physical layers are discussed. Cryptographic techniques are employed at the upper layers of physical layers. The physical layer security approaches are characterized into five major approaches, they are power, channel, code, theoretical secure capacity, and signal detection.

2.2 SURVEY ON CRYPTOGRAPHY ALGORITHMS

Bassam J. Mohd *et al.* [26] have studied lightweight ciphers for resource constrained devices. Comparison is done on the basis of hardware and software implementations of these ciphers. Throughput, area of code, performance efficiency, power and energy are taken into account for comparison. The comparative result shows that CLEFIA has high throughput with respect to others.

Mickael Cazorla *et al.* [27] have implemented lightweight ciphers on the platform MSP430 microcontroller with external clock of 8Mhz. The codes are written in C language. Comparison of ciphers is on the basis of CPU cycles and energy consumption. In total 17 ciphers have been implemented in which memory requirements of CLEFIA and AES are comparable.

Hugo Krawczyk *et al.* [28] show how to combine symmetric encryption and authentication before building a secure channel to protect the communication over the insecure channel. They show that the secure channel protocol is aimed to work with any combination of encryption must use the method of first encryption and then authentication is insecure and authenticate first and then encrypt method is more secure if CBC mode is used.

J. Lan *et al.* [29] implement a 32 bit RNG architecture for low power cryptographic applications. It is approved by NIST and has proved to make high quality random number. This random number generator is used generate keys for algorithms. The keys determine the security level of the algorithm. The RNG can be used to enhance the performance of communication devices and cryptographic application by improving flexibility and power consumption.

Jacob John [30] did a survey on cryptographic algorithms specially designed for constrained devices. Such as smart cards, RFID systems, and wireless sensor networks. Security features of the ciphers were analysed and their performance on hardware implementations was compared. AES had the highest area count and PRESENT and GRAIN the least.

Jia Hao Kong *et al.* [31] have done a survey on approximately 100 cryptographic algorithms that were used in past and new ones currently in use. They aim at both conventional and lightweight ciphers. They have discussed about the current contribution of these ciphers. Analysis on the basis of hardware and software performance is discussed.

Chih-pin Su *et al.* [32] designed an AES processor which has low cost and high throughput value. They proposed an efficient hardware implementation of the AES with the capability of key expansion. The hardware overhead S-box is reduced by 64% by using this transformation

technique.

P. Hamalainen *et al.* [33] have implemented AES for low power and memory devices. An 8-bit architecture is used for encryption supporting 128 bit keys. The optimised result for the algorithm in a 0.13μm technology leads to utilisation of 3.1K gates. The throughput at clock frequency of 153MHz is 121 Mbps.

Yaoping Liu *et al.* [34] implement AES algorithm by implementing S-box based on composite Field arithmetic. In this scheme the multiplicative inverse over the Galois field GF(2^8) is mapped over GF((2^4)^2) and then the multiplicative inverse of GF(2^4) is optimised by Genetic Algorithm . It forms a new architecture of S-box. When compared to the results of direct implementation of AES a reduction of 49.29% is seen.

Makoto Kotegawa *et al.* [35] perform optimisations on the AES algorithm in terms of speed, area size and clock frequency for authenticated encryption. AES-OTR is nonce based and has a Feistel structure which increased the speed of computation.

S. M. Soliman *et al.* [36] propose an optimised design for AES-128 bit encryption only algorithm. They apply the concept of partial loop unrolling, to optimize area, power and throughput iterations and multistage pipelining is used. This design achieves 2490 slices at a frequency of 266 Mhz.

Akishita T *et al.* [37] implemented 128-bit block cipher CLEFIA using very compact Hardware architecture which is based on 8-bit shift registers. The implementations were based on novel serialised architectures without using extra registers in the data processing block. The three types of architectures were implemented depending on the required cycles for one block process by efficiently applying clock gating. The synthesis was done using 0.13μm standard cell library. They obtained the smallest implementation i.e. the area required was 2,488 GE only, which in comparison to smallest implementation of AES-128 and CLEFIA-128 is 50% smaller. The area requirement for the new architectures supporting both encryption and decryption are 2,601 GE. This has achieved 23% reduction in area usage when compared to the smallest implementation of AES-128 bit, both encryption and decryption.

N. R. Potlapally *et al.* [38] show how the resource constrained devices get affected by the security mechanisms, their effect on the energy consumption of these devices. They analyse the energy requirement of various cryptographic algorithms used for security purposes in these devices. They also discuss various opportunities for achieving an energy efficient implementations of security protocols on the devices.

Charalampos Manifavas *et al.* [39] provide a comparative analysis of lightweight ciphers applicable to resource constrained devices. They present recent advancements made in the symmetric, asymmetric and hash algorithms. Features of lightweight algorithms are discussed, like size, cost, speed and power consumption, hardware and software requirements.

Table 2.1: Comparison of Standard AES Cipher with Lightweight Ciphers

Ref.	Algorithm	Tech.	Area	Power	Throughput (Mbps)	Remarks
[30]	AES-128	0.35µm	3595 GE	-	12.59	Comparison of AES with PRESENT
	PRESENT-80	0.18µm	1570 GE	-	200	
[33]	AES- 128	0.13µm	3.1 KGates	37 µW	121	Optimisation on the basis of Area, Power and Speed.
			3.2 KGates	30µW	104	
			3.9 KGates	62µW	232	
[34]	AES-128	0.18µm	2844.07(μm^2)	75 µW	-	S-Box Optimisation.
[35]	AES-128	Zynq-7000 board	2809 slices	-	278	Optimisation on the basis of Speed, Area and Clock
			2809 slices	-	278	
			3263 slices	-	215	
[31]	SIMON-128	0.13µm	1274 GE	-	12.9	-
	SPECK-128	0.13µm	1501 GE	-	21.6	
[37]	CLEFIA-128	0.13µm	2488 GE	-	39	-

Table 2.1 shows various algorithms which are compared with AES 128-bit cipher. These algorithms can be used as lightweight solutions for information security in resource constrained devices.

CHAPTER 3
PROBLEM FORMULATION

3.1 OBSERVATION

From the literature survey, it has been observed that the internet of things is growing at a fast pace but so are the security concerns regarding the data shared through this technology. In Healthcare application of IoTs, because of remote monitoring of patients' it becomes vital to send unaltered information to the healthcare providers in order to get optimum services. This information shared is subject to various attacks therefore, it becomes mandatory to safeguard the information from theft, tampering or misuse. Authors in [17, 19] discuss about security concerns and threats on the medical information and consider cryptography as a solution. Standard cryptography algorithms are not preferred for small devices applications since they are resource constrained. Many limitations are faced by these devices in terms of area, power, computation speed as discussed in [21]. Therefore, cryptographic algorithms which are area efficient and consume less power in computation are considered. Survey on such algorithms and existing techniques which are suitable for application in healthcare are discussed in [16, 24].

3.2 GAPS IN STUDY

From the table 2.1 It can be seen that AES, which is a standard algorithm for the formulation of lightweight ciphers does not fully satisfy the parameters of resource constrained devices. The standard memory allotment given for security in these devices is between 200-2000 gates, example RFID tag. In previous implementations of AES algorithm, the gate count of AES is above 2000 which is more than the required range therefore, NIST approved few algorithms like PRESENT and CLEFIA as Lightweight ciphers which can be used in resource constrained devices. Different approaches are taken to overcome the design trade off in terms of cost, efficiency and security level in these ciphers to get the best possible result. It becomes a tedious task to design ciphers which satisfies all parameters of devices and have security level comparable to AES.

3.3 OBJECTIVES

From the previous sections, objectives can be drawn.

- To study and analyze the privacy and data security issues in E-Healthcare.
- To study the various cryptography techniques for data security and analyze their performance on resource constrained devices.
- To analyse CLEFIA algorithm as a Lightweight Cipher and optimize it for application on resource constrained devices.

- Comparison of different approaches of optimisation with the basic implementation of the cipher.

CHAPTER 4
OVERVIEW OF CLEFIA ALGORITHM AND OPTIMIZATION

4.1 OVERVIEW OF CLEFIA

CLEFIA algorithm was designed by Sony Corporation in 2007 and standardized by NIST shortly. The name CLEFIA is derived from a French word “clef”, meaning “key”. It is a block cipher with 128-bit block and supports key sizes of 128, 192, 256-bits with rounds 18, 22 and 26 for respective key sizes. It was designed to have a good balance between Security, speed and computation cost. Several types of design technologies are used in it to achieve these goals such as General Feistel Network, S-Boxes, Diffusion Matrices, etc. Some of the features of CLEFIA are listed below [40],

- Security: It uses Diffusion Switching Mechanism (DSM), which enhances its immunity against linear and differential attacks. The Key scheduling part is designed to show strong immunity against related key attacks and provide security against differential attacks. Compact F functions in GFN and use of two S-box systems enhance its immunity against many attacks.
- Performance: The design goal was to achieve better performance and security in comparison to AES. Use of lightweight components to improve its hardware and software efficiency. Shared implementation of key scheduling part and data processing part also improves its performance.

Here CLEFIA Algorithm using 128-bit Key is discussed. The Algorithm is divided into two parts: Data processing part and Key scheduling part. First Structure of General Feistel Network is discussed [40].

4.1.1 General Feistel Network (GFN)

To perform encryption CLEFIA uses Type-II GFN(d,r) i.e. the 128 bit input plain text is divided into equally in blocks of 32-bit each (d=4). These 32-bit blocks are iterated through the GFN 18 (r) times to get the cipher text. The GFN structure is depicted in Figure for i^{th} round. Type-2 GFN(d,r) uses two different F-functions, F_0 and F_1 [41].

There is 4 (d) blocks of 32-bit input data X_i $(0 \leq i < d)$ and output Y_i $(0 \leq i < d)$. Each round in the GFN uses two Round Keys (RK) of 32-bit therefore, a total of (d*r) keys (2 keys/rounds × 18 rounds) are used. The working of one round of type-2 GFN is given in the Figure 4.1. Table 4.1 gives the flow of GFN(4,r) and the input is given as,

$$GFN(4,r):\begin{cases} \{\{0,1\}^{32}\}^{36} \times \{\{0,1\}^{32}\}^{4}\} \longrightarrow \{\{0,1\}^{32}\}^{4} \\ (RK_{0(32)}, \ldots., RK_{2r-1(32)}, X_{0(32)}, \ldots, X_{3(32)}) \longmapsto Y_{0(32)}, \ldots\ Y_{3(32)} \end{cases} \quad (4.1)$$

Table 4.1: Flow of GFN(4,r)

Step 1.	$P_0 \mid P_1 \mid P_2 \mid P_3 \leftarrow X_0 \mid X_1 \mid X_2 \mid X_3$
Step 2.	For i=0 to r-1 do,
Step 2.1	$P_1 \leftarrow P_1 \oplus F_0(RK_{2i}, P_0)$ $P_3 \leftarrow P_3 \oplus F_1(RK_{2i+1}, P_2)$
Step 2.2	$P_0 \mid P_1 \mid P_2 \mid P_3 \leftarrow P_1 \mid P_2 \mid P_3 \mid P_0$
Step 3.	$Y_0 \mid Y_1 \mid Y_2 \mid Y_3 \leftarrow P_3 \mid P_0 \mid P_1 \mid P_2$

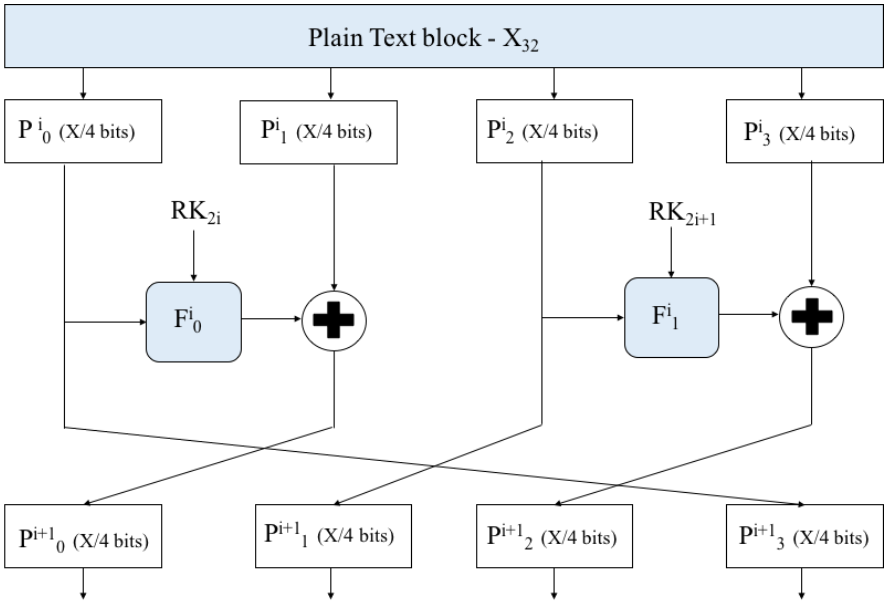

Figure 4.1: One Round of GFN(4, r)

In Feistel Network Decryption process reversible to encryption and is obtained by reversing the order of Round Keys and Cipher text becomes the input and Plain text is the output. The flow of inverse of $GFN^{-1}(4,r)$ is given in table 4.2 and input is given as [41],

$$GFN^{-1}(4,r):\begin{cases} \{\{0,1\}^{32}\}^{36} \times \{\{0,1\}^{32}\}^{4}\} \longrightarrow \{\{0,1\}^{32}\}^{4} \\ (RK_{2r-1(32)}, \ldots., RK_{0(32)}, Y_{0(32)}, \ldots, Y_{3(32)}) \longmapsto X_{0(32)}, \ldots\ X_{3(32)} \end{cases} \quad (4.2)$$

4.1.2 Data Processing Part

This part performs encryption and decryption. It is based on GFN(d, r), uses two F-functions, Whitening keys , S-boxes and Diffusion Matrices.

Table 4.2: Flow of Inverse GFN(4, r)

Step 1.	$P_0 \mid P_1 \mid P_2 \mid P_3 \leftarrow Y_0 \mid Y_1 \mid Y_2 \mid Y_3$
Step 2.	For i=0 to r-1 do,
Step 2.1	$P_1 \leftarrow P_1 \oplus F_0 (RK_{2(r-i)-2}, P_0)$ $P_3 \leftarrow P_3 \oplus F_1 (RK_{2(r-i)-1}, P_2)$
Step 2.2	$P_0 \mid P_1 \mid P_2 \mid P_3 \leftarrow P_1 \mid P_2 \mid P_3 \mid P_0$
Step 3.	$X_0 \mid X_1 \mid X_2 \mid X_3 \leftarrow P_3 \mid P_0 \mid P_1 \mid P_2$

4.1.2.1 F-Functions

Type-2 GFN uses two different F-Functions of 32-bit each. They perform Substitution using 8-bit S-boxes (S_0, S_1) and Diffusion using a 4×4 diffusion matrices (M_0, M_1). Table 4.3 and 4.4 give the flow of functions. Their Input/output are defined as follows [42].

$$F_0F_1 = \begin{cases} \{0,1\}^{32} \times \{0,1\}^{32} \longrightarrow \{0,1\}^{32} \\ (RK_{32},\ x_{32}) \longrightarrow y_{32} \end{cases} \tag{4.3}$$

where $F_0 : (\ RK_{32},\ x_{32}) \longrightarrow y_{32}$

Table 4.3: Flow of Function F_0

Step 1.	$P \leftarrow RK \oplus x$
Step 2.	Break $P = P_0 \mid P_1 \mid P_2 \mid P_3$, where $P_i \in \{0,1\}^8$
	$P_0 \leftarrow S_0 (P_0)$, $P_1 \leftarrow S_1 (P_1)$ $P_2 \leftarrow S_0 (P_2)$, $P_3 \leftarrow S_1 (P_3)$
Step 3.	Break $y = y_0 \mid y_1 \mid y_2 \mid y_3$, where $y_i \in \{0,1\}^8$ $(y_0, y_1, y_2, y_3) = M_0 (P_0, P_1, P_2, P_3)$

And $F_1 : (\ RK_{32},\ x_{32}) \longrightarrow y_{32}$

Table 4.4: Flow of Function F_1

Step 1.	$P \leftarrow RK \oplus x$
Step 2.	Break $P = P_0 \mid P_1 \mid P_2 \mid P_3$, where $P_i \in \{0,1\}^8$
	$P_0 \leftarrow S_1 (P_0)$, $P_1 \leftarrow S_0 (P_1)$ $P_2 \leftarrow S_1 (P_2)$, $P_3 \leftarrow S_0 (P_3)$
Step 3.	Break $y = y_0 \mid y_1 \mid y_2 \mid y_3$, where $y_i \in \{0,1\}^8$ $(y_0, y_1, y_2, y_3) = M_0 (P_0, P_1, P_2, P_3)$

The processing of F-functions can be done simultaneously which is good for hardware performance. The size of these functions is comparatively smaller than the ones used in traditional Feistel structure which improves both hardware and software performance [43]. Figure 4.2 shows the construction of F-Functions.

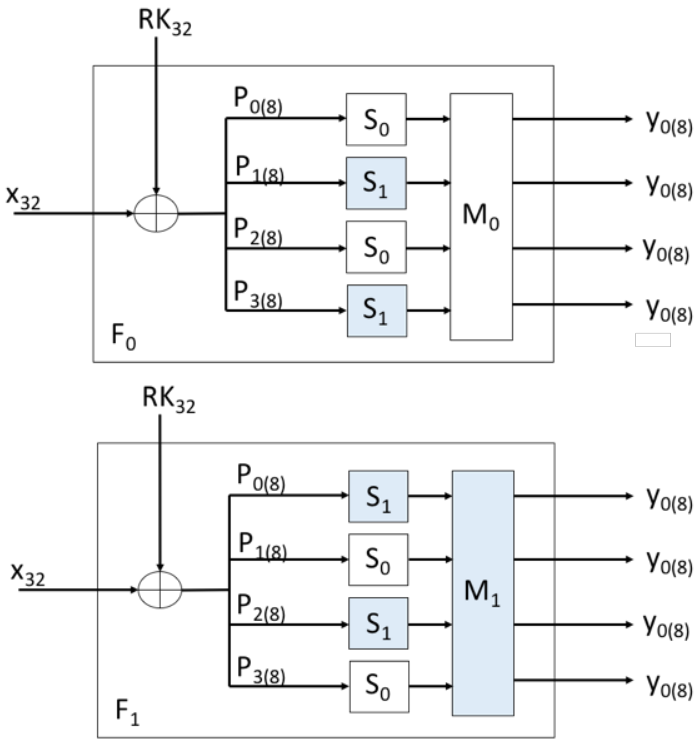

Figure 4.2: F-Functions: F_0, F_1 [42]

4.1.2.2 S-Boxes

In any algorithm S-boxes are chosen because they provide good immunity against known attacks and are suitable for efficient hardware implementation. CLEFIA employs two non-linear S-Boxes: S_0, $S_1$of 8-bit each [42].

$$S_0, S_1 = \begin{cases} \{0,1\}^8 \longrightarrow \{0,1\}^8 \\ x_8 \longmapsto y_8 \end{cases} \tag{4.4}$$

- S_0: It is based on 4-bit random S-Boxes SS_0, SS_1, SS_2 and SS_3. They are connected with 2×2 matrix, the multiplication is done over GF(2^4) (Galois Field) defined by the polynomial $z^4 + z + 1$. The table 4.5 explains S_0.

  Figure 4.3 shows the construction of S_0. Table 4.6 shows the values of S-boxes SS_0, SS_1, SS_2 and SS_3. These values are Random bit strings generated by AES in Counter mode.

Table 4.5: Flow of S-Box S_0

Step 1.	Break $x_8 = x_0 \mid x_1$, where $x_i \in \{0,1\}^4$
Step 2.	$t_0 \leftarrow SS_0(x_0)$, $t_1 \leftarrow SS_1(x_1)$
	$u_0 \leftarrow t_1 \oplus 0{\times}2 \cdot t_0$, $u_1 \leftarrow t_0 \oplus 0{\times}2 \cdot t_1$
Step 3.	$y_0 \leftarrow SS_2(u_1)$, $y_1 \leftarrow SS_3(u_0)$, where $y_8 = y_0 \mid y_1$, $y_i \in \{0,1\}^4$

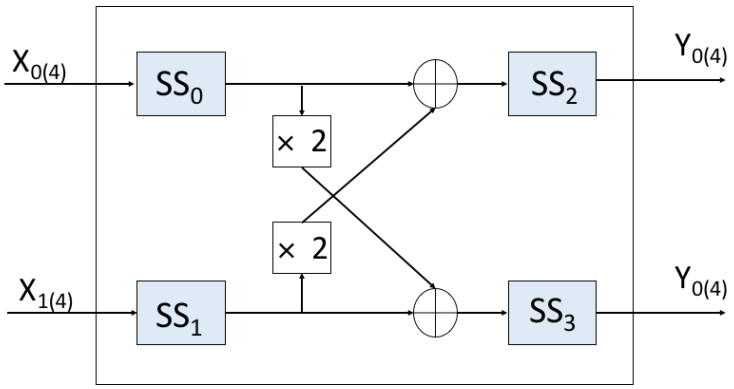

Figure 4.3: S_0 Construction [42]

Table 4.6: Values of SS_i $(0 \le i < 4)$

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$SS_0(x)$	e	6	c	a	8	7	2	f	b	1	4	0	5	9	d	3
$SS_1(x)$	6	4	0	d	2	b	a	3	9	c	e	f	8	7	5	1
$SS_2(x)$	b	8	5	e	a	6	4	c	f	7	2	3	1	0	d	9
$SS_3(x)$	a	2	6	d	3	4	5	e	0	7	8	9	b	f	c	1

- S_1: It is based on the inverse function over the Galois Field GF(2^8) defined by irreducible polynomial $z^8 + z^4 + z^3 + z^2 + 1$. S_1 is defined as

$$y = \begin{cases} g(f(x)^{-1} & if\ f(x) \neq 0 \\ g(0) & if\ f(x) = 0 \end{cases} \tag{4.5}$$

Figure 4.4 shows the construction of s_1. The equations *f(·)* and *g(·)* are affine transformations over the Galois Field GF(2), defined by

$$f, g = \begin{cases} \{0,1\}^8 \longrightarrow \{0,1\}^8 \\ x_8 \longmapsto y_8 \end{cases} \tag{4.6}$$

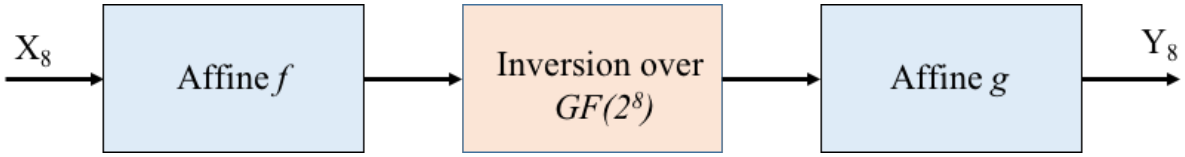

Figure 4.4: S_1 Construction [43]

$$f(\cdot)\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \quad (4.7)$$

$$g(\cdot)\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (4.8)$$

Where $x, y = \begin{cases} x_0 \mid x_1 \mid x_2 \mid x_3 \mid x_4 \mid x_5 \mid x_6 \mid x_7 \\ y_0 \mid x_1 \mid x_2 \mid x_3 \mid x_4 \mid x_5 \mid x_6 \mid x_7 \end{cases}$ $x_i, y_i \in \{0,1\}$ and constants in f and g are $0\times1e$ and 0×69 respectively.

S-Boxes in CLEFIA can also be used in Look Up Table form, Figure 4.5.

4.1.2.3 Diffusion Matrices

Two Diffusion Matrices are used in F-Functions, M_0 in F_0 and M_1 in F_1. They are 4×4 matrices with elements $h_{ij} = a_{i\oplus j}$ for certain set $\{a_0, a_1, a_2, a_3\}$. In CLEFIA they are defined as follows.

$$M_0 = \begin{pmatrix} 0\times01 & 0\times02 & 0\times04 & 0\times06 \\ 0\times02 & 0\times01 & 0\times06 & 0\times04 \\ 0\times04 & 0\times06 & 0\times01 & 0\times02 \\ 0\times06 & 0\times04 & 0\times02 & 0\times01 \end{pmatrix}$$

$$M_1 = \begin{pmatrix} 0\times01 & 0\times08 & 0\times02 & 0\times0A \\ 0\times08 & 0\times01 & 0\times0A & 0\times02 \\ 0\times02 & 0\times0A & 0\times01 & 0\times08 \\ 0\times0A & 0\times02 & 0\times08 & 0\times01 \end{pmatrix} \quad (4.9)$$

The multiplication of elements in these matrices and input vectors is done over the field GF(2^8), defined over the polynomial $z^8 + z^4 + z^3 + z^2 + 1$. The matrices were chosen because they have low hamming weights and can be efficiently implemented in hardware because the number of XOR gates used are low [43].

S_0

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

S_1

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

Figure 4.5: Lookup Tables for S_0 and S_1 [42]

4.1.3 Key Scheduling

This part generates Round Keys (RK) and Whitening Keys (WK) from the main secret key (K). CLEFIA supports key size of 128, 192, 256-bit length. For performing all this Constant value, Intermediate Key and DoubleSwap function are used. They are described below.

- Constant Values: CLEFIA-128 uses 60 constant values, out of which first twenty-four are used to generate L and rest are used to generate RK. The following Table 4.7 explains the method used to generate the constant values [41].

Table 4.7: Flow of generating Constant Values

Step 1.	$T \leftarrow IV^{(k)}$, k=128
Step 2.	For i = 0 to $l^{(k)}$-1 do
Step 2.1	$CON_{2i}^{(k)} \leftarrow (T \oplus P) \mid (T' \lll 1)$
Step 2.2	$CON_{2i+1}^{(k)} \leftarrow (T' \oplus Q) \mid (T \lll 8)$
Step 2.3	$T \leftarrow T \cdot 0x0002^{-1}$
Where $P_{(16)}$ = 0xb7e1, $Q_{(16)}$ = 0x243f For 128-bit Key IV=0x428a, l=30	

In step 2.3 the multiplication is done in field GF(2^{16}) defined by primitive polynomial $z^{16} + z^{15} + z^{13} + z^{11} + z^5 + z^4 + 1$. A Look up table can also be used in place of generating the values on the fly, but it will increase the cost of storing the values in Hardware.

- Intermediate Key (L)

  The intermediate key is generated by using GFN(4,12), where the input elements are the 32-bit Constant values $CON_i^{(128)}(0 \leq i < 24)$) and the main secret Key K ($K = K_0 \mid K_1 \mid K_2 \mid K_3$).

- Double Swap Function ($\sum$): The purpose of this function is to update the Intermediate Key by swapping its bits repeatedly in every two rounds. This is done to destroy any relation between all the Round Keys. It is more hardware efficient than the rotation operation**.** The Function is defined as following.

$$X_{(128)} \longmapsto Y_{(128)} \qquad (4.10)$$
$$Y = X[7 - 63] \mid X[121 - 127] \mid X[0 - 6] \mid X[64 - 120]$$

where X[a - b] denotes that the string is cut from a^{th} bit to b^{th} bit and then concatenated with the others, following Figure 4.6 shows the DoubleSwap Function.

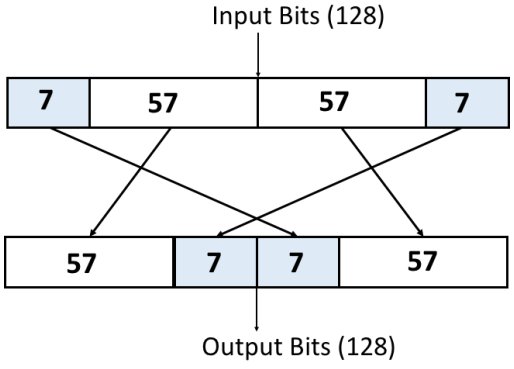

Figure 4.6: DoubleSwap Function ∑ [41]

4.1.3.1 Round Keys

The Round Keys are used in Encryption and Decryption process. After generating the Intermediate Key L, K and L are used to generate Whitening Keys and Round Keys. The process is described below in table 4.8.

Table 4.8: Flow of Key Scheduling Part

Step 1.	Generate Constant Values $\Rightarrow CON_i^{(128)} (0 \leq i < 59)$
	Generate L from K
Step 2.	$L \leftarrow GFN_{4,12}(CON_0^{(128)}, \ldots CON_{23}^{(128)}, K_0 \ldots K_3)$
	Expanding K and L
Step 3.	$WK_0 \mid WK_1 \mid WK_2 \mid WK_3 \leftarrow K$
Step 4.	For i = 0 to 8 do the following
Step 4.1	$T \leftarrow L \oplus \left(CON_{24+4i}^{(128)} \mid CON_{24+4i+1}^{(128)} \mid CON_{24+4i+2}^{(128)} \mid CON_{24+4i+3}^{(128)}\right)$
Step 4.2	$L \leftarrow \sum (L)$
Step 4.3	If i is odd: $T \leftarrow T \oplus K$
Step 4.4	$RK_{4i} \mid RK_{4i+1} \mid RK_{4i+2} \mid RK_{4i+3} \leftarrow T$

4.1.4 Encryption and Decryption

The encryption (ENC_r) and Decryption (DEC_r) in CLEFIA uses a GFN(4,18). The input and output are 128-bit blocks of data. The plaintext and cipher text is divided into four 32-bit blocks represented as $P_i, C_i \in \{0, 1\}^{32}$ $(0 \leq i < 4)$ where $P = P_0 \mid P_1 \mid P_2 \mid P_3$ and $C = C_0 \mid C_1 \mid C_2 \mid C_3$. The Whitening Keys defined as WK_0, WK_1, WK_2, $WK_3 \in \{0, 1\}^{32}$ and the 36 Round Keys $RKi \in \{0, 1\}^{32}$, $0 \leq i < 2r$ are generated from the Key Scheduling Part [41]. The process of Encryption and is defined in table 4.9.

Table 4.9: Flow of Encryption Process

Step 1.	$P_0 \mid P_1 \mid P_2 \mid P_3 \leftarrow P$
Step 2.	$T_0 \mid T_1 \mid T_2 \mid T_3 \leftarrow P_0 \mid (P_1 \oplus WK_0) \mid P_2 \mid (P_3 \oplus WK_1)$
Step 3.	$T_0 \mid T_1 \mid T_2 \mid T_3 \leftarrow GFN_{(4,r)}(RK_0, \ldots RK_{2r-1}, T_0, T_1, T_2, T_3)$
Step 4.	$C_0 \mid C_1 \mid C_2 \mid C_3 \leftarrow T_0 \mid (T_1 \oplus WK_2) \mid T_2 \mid (T_3 \oplus WK_4)$
Step 5.	$C \leftarrow C_0 \mid C_1 \mid C_2 \mid C_3$

Decryption process is inverse of encryption and is defined by $GFN_{d,r}^{-1}$ explained in table 4.10. The order of round keys is reversed and Cipher Text becomes the input as shown below. Figure 4.7 shows the Complete Working of CLEFIA [41].

Table 4.10: Flow of Decryption Process

Step 1.	$C_0 \mid C_1 \mid C_2 \mid C_3 \leftarrow P$
Step 2.	$T_0 \mid T_1 \mid T_2 \mid T_3 \leftarrow C_0 \mid (C_1 \oplus WK_2) \mid C_2 \mid (C_3 \oplus WK_3)$
Step 3.	$T_0 \mid T_1 \mid T_2 \mid T_3 \leftarrow GFN_{(4,r)}(RK_0, \ldots RK_{2r-1}, T_0, T_1, T_2, T_3)$
Step 4.	$P_0 \mid P_1 \mid P_2 \mid P_3 \leftarrow T_0 \mid (T_1 \oplus WK_0) \mid T_2 \mid (T_3 \oplus WK_2)$
Step 5.	$P \leftarrow P_0 \mid P_1 \mid P_2 \mid P_3$

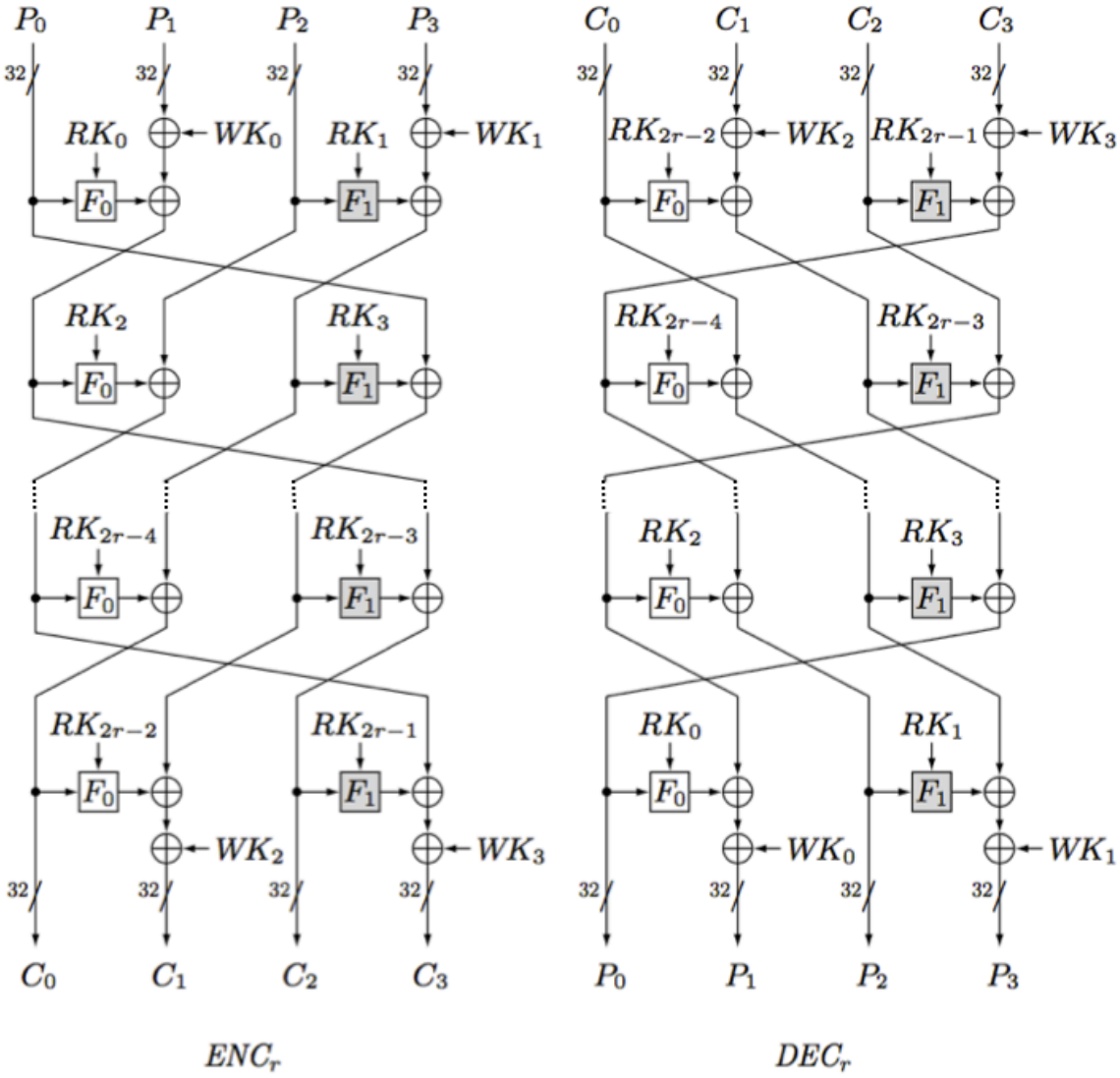

Figure 4.7: CLEFIA Encryption and Decryption Process [42]

4.2 OPTIMIZATION TECHNIQUES FOR CLEFIA

4.2.1 Software Simulation

In software simulation targeting for minimum execution cycles the following approaches are used.

- Look Up Tables: The use of lookup tables can reduce the time, as getting value from memory is faster than generating it through various operations. Any complex function

that has tedious calculations can be replaced by a lookup table when timing is an issue. The tables can be calculated first and then stored in the memory. The S-Boxes S_0 and S_1 are used as Look Up tables in the software simulation. The Constant Values that are generated on the fly can also be used as a Look Up Table, since they do not depend on any variable parameter like Key or Plain Text. The benefit of using Lookup tables is that it resolves timing attacks on the algorithm. The different operations take different time to execute thus affecting the timing of software. The attacker can measure the timing of these operations and discover the secret data. Use of Lookup table can resolve this problem as finding values takes same time every time rather than calculating them through logical operations.

- Inline Functions: Use of inline function can lead to optimisation of codes. Inline keyword can be used for small functions which are not recursively called. This will inline the function with the code execution. This approach reduces the extra overhead of calling the function.
- Diffusion Matrices: The compact implementation of Diffusion matrices can be achieved by breaking them in three parts using properties of matrices. The perform multiplication over GF(2^8) with irreducible polynomial $z^8 + z^4 + z^3 + z^2 + 1$, by elements $0\times02,\ 0\times04, 0\times06, 0\times08, 0\times0A$. By applying this method, computations are decreased as now only multiplication done is by elements $0\times02, 0\times04, 0\times08$. This reduces overall cycle count. Table 4.12 explains the multiplication of X with 0×02 in GF(2^8) [43].

M_0

$$\begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{pmatrix} 0\times01 & 0\times02 & 0\times04 & 0\times06 \\ 0\times02 & 0\times01 & 0\times06 & 0\times04 \\ 0\times04 & 0\times06 & 0\times01 & 0\times02 \\ 0\times06 & 0\times04 & 0\times02 & 0\times01 \end{pmatrix} \cdot \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} \tag{4.11}$$

$$= \begin{pmatrix} 0\times01 & 0\times00 & 0\times00 & 0\times00 \\ 0\times00 & 0\times01 & 0\times00 & 0\times00 \\ 0\times00 & 0\times00 & 0\times01 & 0\times00 \\ 0\times00 & 0\times00 & 0\times00 & 0\times01 \end{pmatrix} \cdot \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} + \begin{pmatrix} 0\times00 & 0\times02 & 0\times00 & 0\times02 \\ 0\times02 & 0\times00 & 0\times02 & 0\times00 \\ 0\times00 & 0\times02 & 0\times00 & 0\times02 \\ 0\times02 & 0\times00 & 0\times02 & 0\times00 \end{pmatrix} \cdot \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

$$+ \begin{pmatrix} 0\times00 & 0\times00 & 0\times04 & 0\times04 \\ 0\times00 & 0\times00 & 0\times04 & 0\times04 \\ 0\times04 & 0\times04 & 0\times00 & 0\times00 \\ 0\times04 & 0\times04 & 0\times00 & 0\times00 \end{pmatrix} \cdot \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

$$\begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} \oplus \begin{pmatrix} (X_1 \oplus X_3) \cdot \{0\times02\} \\ (X_0 \oplus X_2) \cdot \{0\times02\} \\ (X_1 \oplus X_3) \cdot \{0\times02\} \\ (X_0 \oplus X_2) \cdot \{0\times02\} \end{pmatrix} \oplus \begin{pmatrix} (X_2 \oplus X_3) \cdot \{0\times04\} \\ (X_2 \oplus X_3) \cdot \{0\times04\} \\ (X_0 \oplus X_1) \cdot \{0\times04\} \\ (X_0 \oplus X_1) \cdot \{0\times04\} \end{pmatrix}$$

M_1

$$\begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{pmatrix} 0\times01 & 0\times08 & 0\times02 & 0\times0A \\ 0\times08 & 0\times01 & 0\times0A & 0\times02 \\ 0\times02 & 0\times0A & 0\times01 & 0\times08 \\ 0\times0A & 0\times02 & 0\times08 & 0\times01 \end{pmatrix} \cdot \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} \tag{4.12}$$

$$= \begin{pmatrix} 0\times01 & 0\times00 & 0\times00 & 0\times00 \\ 0\times00 & 0\times01 & 0\times00 & 0\times00 \\ 0\times00 & 0\times00 & 0\times01 & 0\times00 \\ 0\times00 & 0\times00 & 0\times00 & 0\times01 \end{pmatrix} \cdot \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} + \begin{pmatrix} 0\times00 & 0\times00 & 0\times02 & 0\times02 \\ 0\times02 & 0\times00 & 0\times02 & 0\times02 \\ 0\times02 & 0\times02 & 0\times00 & 0\times02 \\ 0\times02 & 0\times02 & 0\times02 & 0\times00 \end{pmatrix} \cdot \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

$$+ \begin{pmatrix} 0\times00 & 0\times08 & 0\times00 & 0\times08 \\ 0\times08 & 0\times00 & 0\times08 & 0\times00 \\ 0\times00 & 0\times08 & 0\times00 & 0\times08 \\ 0\times08 & 0\times04 & 0\times08 & 0\times00 \end{pmatrix} \cdot \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

$$\begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix} \oplus \begin{pmatrix} (X_2 \oplus X_3) \cdot \{0\times02\} \\ (X_2 \oplus X_3) \cdot \{0\times02\} \\ (X_0 \oplus X_1) \cdot \{0\times02\} \\ (X_0 \oplus X_1) \cdot \{0\times02\} \end{pmatrix} \oplus \begin{pmatrix} (X_1 \oplus X_3) \cdot \{0\times08\} \\ (X_0 \oplus X_2) \cdot \{0\times08\} \\ (X_1 \oplus X_3) \cdot \{0\times08\} \\ (X_0 \oplus X_2) \cdot \{0\times08\} \end{pmatrix}$$

Table 4.11: Flow of Galois Field Multiplication in C Language [47]

Unsigned char Multiply_2(unsigned char)
{ // multiplication is done over GF(2^8) where polynomial p(x) = "0×11d"
If(x & 0×80U)
{ x = x ⊕ 0× 0EU; }
Return ((x ≪ 1) | (x ≫ 7));
}
#define Multiply_4(x) = (Multiply_2 (Multiply_2 (x))); #define Multiply_8(x) = (Multiply_2 (Multiply_4 (x)));

4.2.2 Galois Field

The multiplication in S-Box are based on Galois Field. We can represent the data in GF as vector. The field allows mathematical operations which scramble the data effectively. Elements in GF are represented as,

$$\begin{aligned} GF(p^n) = {} & (0, 1, 2, \ldots, (p-1)) \cup \\ & (p, (p+1), \ldots, (p+p-1)) \cup \\ & (2, (p^2+1), \ldots, (p^2+p+1)) \cup \end{aligned} \tag{4.13}$$

$$(p^{n-1}, (p^{n-1} + 1), \ldots, (p^{n-1} + p - 1))$$

Where p is called characteristic of the field, p^n is the order of the field, $p \in P$ and $n \in Z^+$ [44].

Example 1: GF (5) = (0, 1, 2, 3, 4), it has five elements and each of them is a polynomial of power zero.

Example 2: GF(2^4) = (0, 1, 2, (2^1 + 1), 2^2, (2^2 + 1), (2^2 + 2), (2^2 + 2^1 + 1), (2^3 + 2^2), (2^3 + 2^2 + 1), (2^3 + 2^2 + 1), (2^3 + 2^2 + 2 + 1)), which is = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 0, 11, 12, 13, 14, 15). It has 16 elements (2^4), and each is a polynomial of at most degree 3. In polynomial form, they can be represented as follows.

$$\begin{array}{cccc} 0 & z^2 & z^3 & z^3 + z^2 \\ 1 & z^2 + 1 & z^3 + 1 & z^3 + z^2 + 1 \\ Z & z^2 + z & z^3 + z & z^3 + z^2 + z \\ z + 1 & z^2 + z + 1 & z^3 + z + 1 & z^3 + z^2 + z + 1 \end{array} \quad (4.14)$$

Mathematical operations can be performed like addition, subtraction, multiplication and inversion. The examples of arithmetic operations in GF(2^4) with reduction polynomial $f(z) = z^4 + z + 1$ are given below [44].

- Multiplication: $(z^3 + z^2 + 1) \cdot (z^2 + z + 1) = (z^5 + z + 1)$ the degree of which is greater than reduction polynomial therefore, it is reduced by reduction polynomial. $(z^5 + z + 1)$ mod $(z^4 + z + 1) = z^2 + 1$.
- Inversion: $(z^3 + z^2 + 1)^{-1} = z^2$, since $(z^3 + z^2 + 1) \cdot z^2 \; mod \; (z^4 + z + 1) = 1$.

4.2.3 S-Box Optimization

The working of S-boxes is based on Galois Field. There are direct lookup tables which can be implemented instead of using multiplications in GF, but it consumes area to store all those 8-bit 256 values. To avoid area consumption, on the fly generation of these values can be done.

- S_0: It consists of three layers, Figure 4.8:
  1. Substitution Layer 1
  2. Linear Transformation
  3. Substitution Layer 2

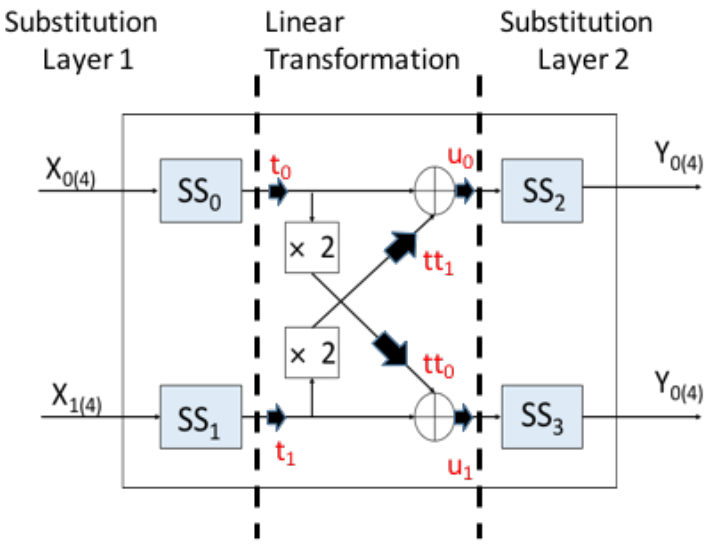

Figure 4.8: Layers in S_0

Substitution Layers 1 and 2 can be implemented using 4-bit S-boxes (SS_0, SS_1, SS_2, SS_3) with 16 values each. Linear transformation is done over the field GF(2^4) defined by primitive polynomial $x^4 + x + 1$. This can be obtained by bit-shifting and XORing. Table 4.11 explains the process.

Table 4.12: Flow of Optimized S_0

Step 1.	Break $x_8 = x_0 \mid x_1$, where $x_i \in \{0,1\}^4$
Step 2.	Let $t_0 \leftarrow SS_0(x_0)$, $t_1 \leftarrow SS_1(x_1)$
Step 3.1	If$((2 * t_0) < 0x10)$ then $tt_0 \leftarrow (2 * t_0)$ else $tt_0 \leftarrow (2 * t_0) \oplus (0x13)$, where $(0x10 \rightarrow x^4$ and $0x13 \rightarrow x^4 + x + 1)$
Step 3.2	If$((2 * t_1) < 0x10)$ then $tt_1 \leftarrow (2 * t_0)$ else $tt_1 \leftarrow (2 * t_1) \oplus (0x13)$
Step 4.	$u_0 \leftarrow t_0 \oplus tt_1$, $u_1 \leftarrow t_1 \oplus tt_0$
Step 5.	$y_0 \leftarrow SS_2(u_1)$, $y_1 \leftarrow SS_3(u_0)$
Step 6.	$y_8 = \{ y_0 \mid y_1 \}$, $y_i \in \{0,1\}^4$

CHAPTER 5
SIMULATION AND RESULTS

This Chapter presents the results of optimized CLEFIA Algorithm on Software and Hardware and their comparison with the existing values. Different Parameters are defined to quantify the obtained results.

5.1 SOFTWARE SIMULATION

The software Simulation of CLEFIA-128 is done using GCC compiler. GCC stands for "GNU Compiler Collection", it is an integrated collection of compilers for software languages like C, C++, Java, etc. When a code is run, it performs preprocessing, compiling, assembly and linking [45].

- Pre-processing: It expands Macros
- Compilation: Source Code to Assembly code
- Assembly: Assembly Language to Machine Language
- Linking: Create final executable file

Figure 5.1 gives brief overview of compilation in GCC. It provides with various options to control warnings, C Dialect, code optimizations, etc. To execute the code TSIM simulator is used.

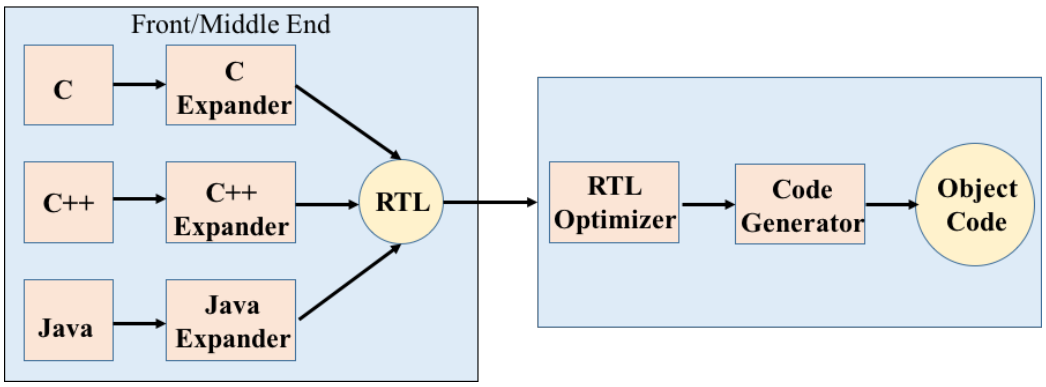

Figure 5.1: Existing Compilation process in GCC [46]

Few commands that are used while compiling the codes are listed in table 5.1

Table 5.1 GCC Commands [45]

-w	Prevent all warning messages
-Wall	Enable all warning messages
-g	To debug the code in native operating system format
-o	Controls level of optimization (-o2, -o3, etc.)
perf	This command displays various execution statistics

The simulation flow of execution of C code in GCC is shown in Figure 5.2.

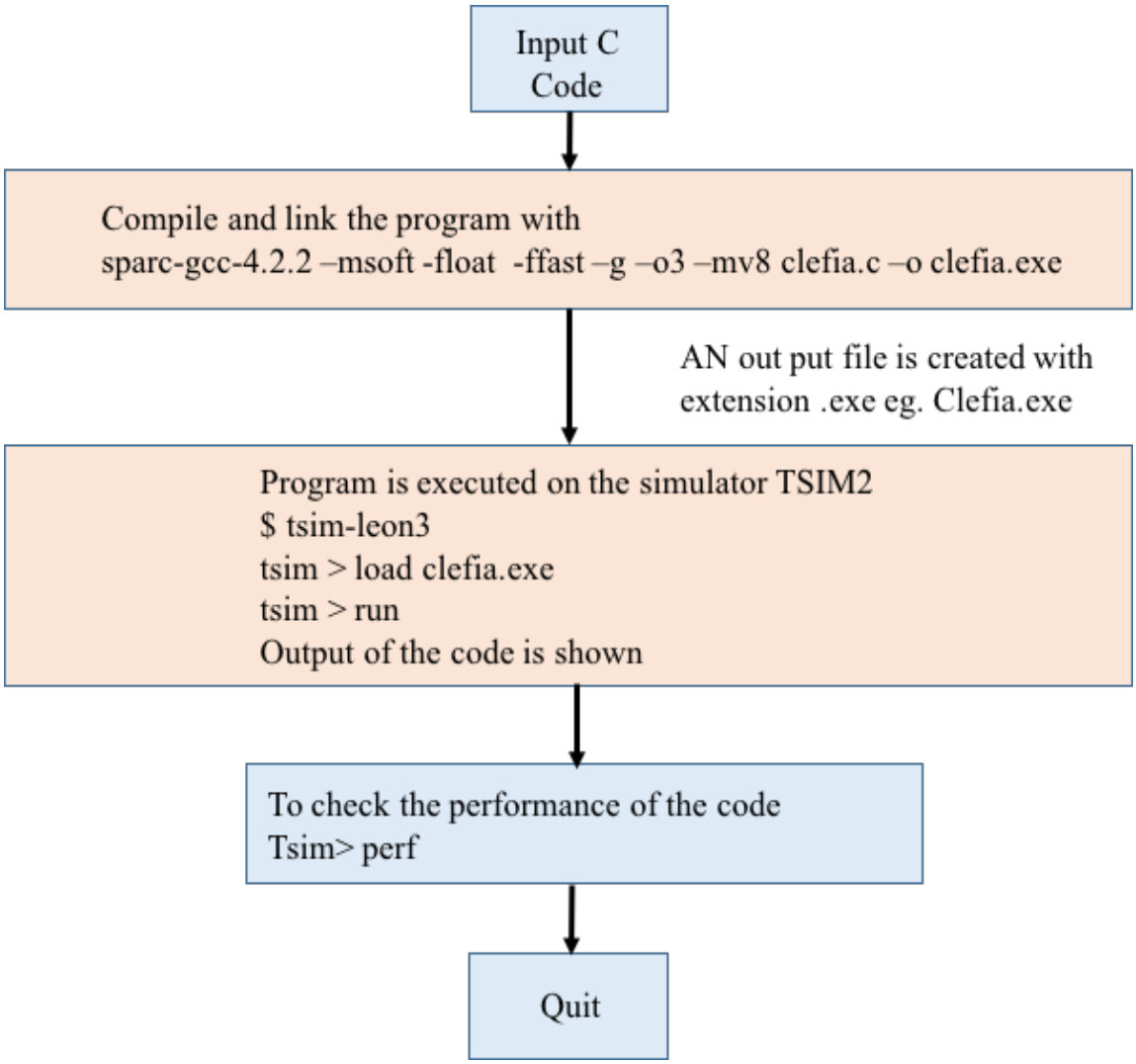

Figure 5.2: Simulation Flow for GCC

5.2 SOFTWARE PERFORMANCE

The Flow of C Code for complete execution of CLEFIA is shown in Figure 5.3 and the output is shown in Figure 5.4. The program performs complete Encryption and generates cipher text. Using that cipher text, it performs Decryption to get back the plain text. This process involves Key scheduling and Data Processing. For S- Box and Constant Values Look-Up Tables are used and Diffusion Matrices are optimized using the properties of matrices. The performance results are compared with the reference code provided by Sony Corporation [47]. In their Implementation S-Boxes are used in lookup table form and Constant Values used to generate intermediate key and round keys are generated during execution. The parameters used to compare the performance are Cycle count, number of instructions and Cycles per Instructions (average number of clock cycles used per instruction, CPI).

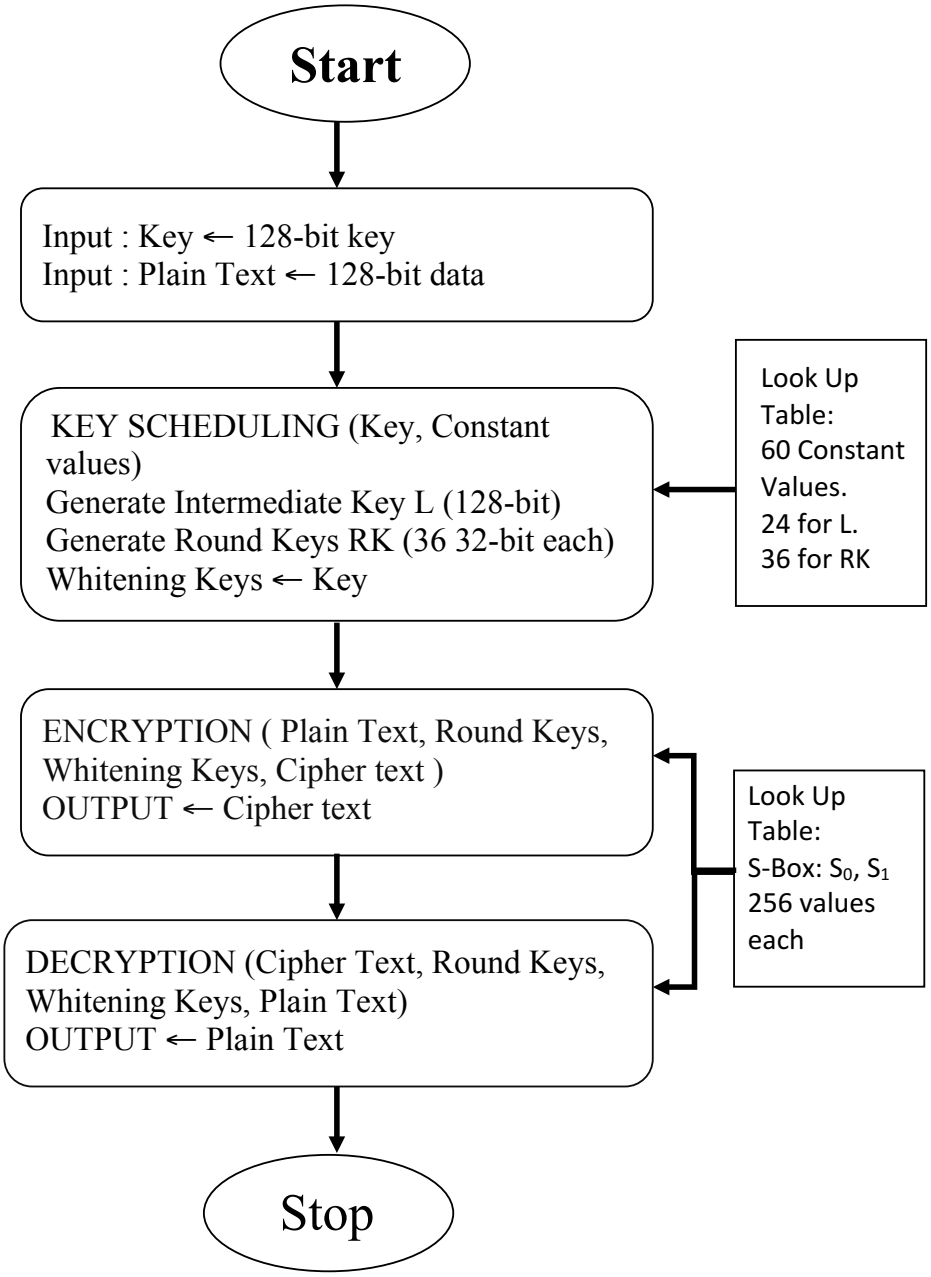

Figure 5.3: Flowchart for C Code

```
tsim> run
starting at 0x40000000
The Secret Key is = ffeeddcc     bbaa9988        77665544        33221100

The Plain text is = 112233       44556677        8899aabb        ccddeeff

Encryption Process:
 Input: Plain Text
 Output: Cipher Text
The Cipher text is = 915b1d9e    4c2cde28        3e9ff83a        2fbacb89

Decryption procrss:
 Input: Cipher Text
 OutPut: Plain Text
The Plain Text is= 112233        44556677        8899aabb        ccddeeff


Program exited normally.
```

Figure 5.4: Software Output of CLEFIA

- Table 5.2 shows the performance result of Key Generation and Encryption process for existing and optimized CLEFIA.

Table 5.2: Key Scheduling and Encryption

Key + Encryption		
Parameters	Existing CLEFIA [47]	Optimized CLEFIA
Cycles	189775	74120
Instructions	116366	43365
CPI	1.63	1.70
CPU performance = 50Mhz		

- Table 5.3 shows the performance result of Key Generation and Decryption process for existing and optimized CLEFIA

Table 5.3: Key Scheduling and Decryption

Key + Decryption		
Parameters	Existing CLEFIA [47]	Optimized CLEFIA
Cycles	191372	74745
Instructions	116483	43583
CPI	1.64	1.71
CPU performance = 50Mhz		

- Table 5.4 shows the complete performance of CLEFIA which performs Key Scheduling, Encryption and Decryption for existing and optimized CLEFIA.

Table 5.4: Complete execution for existing and optimized CLEFIA

Complete Execution (Key + Encryption + Decryption)		
Parameters	Existing CLEFIA [47]	Optimized CLEFIA
Cycles	422210	102683
Instructions	248554	64130
CPI	1.70	1.60
CPU performance = 50Mhz		

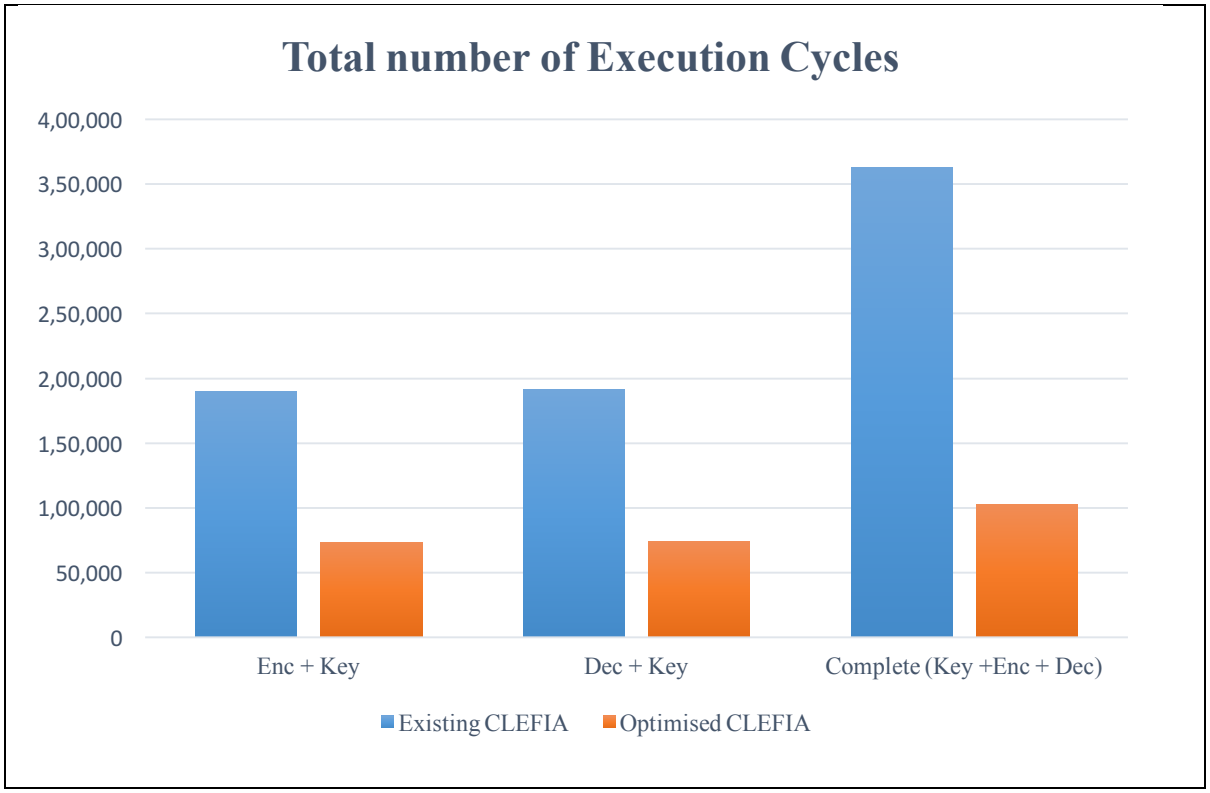

Figure 5.5: Comparison of Execution cycles for existing and optimized CLEFIA

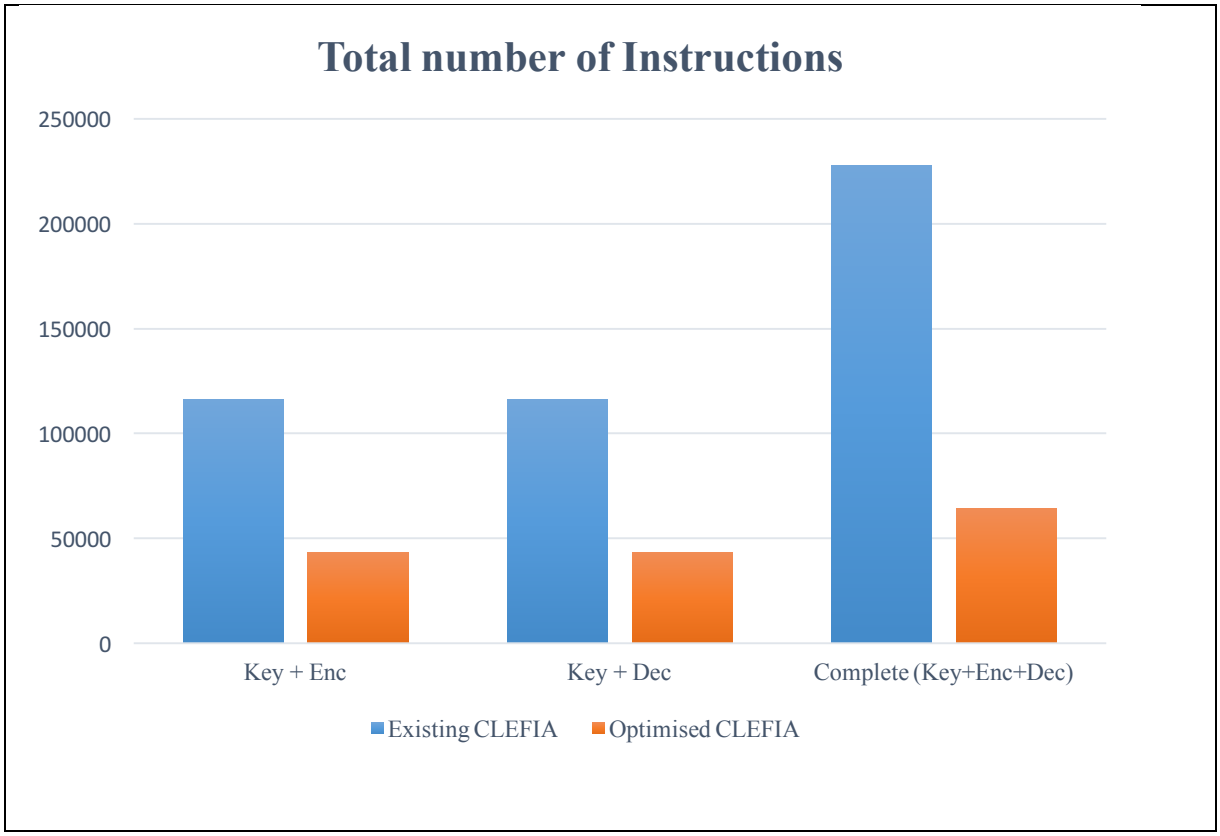

Figure 5.6: Comparison of Total Instructions for existing and optimized CLEFIA.

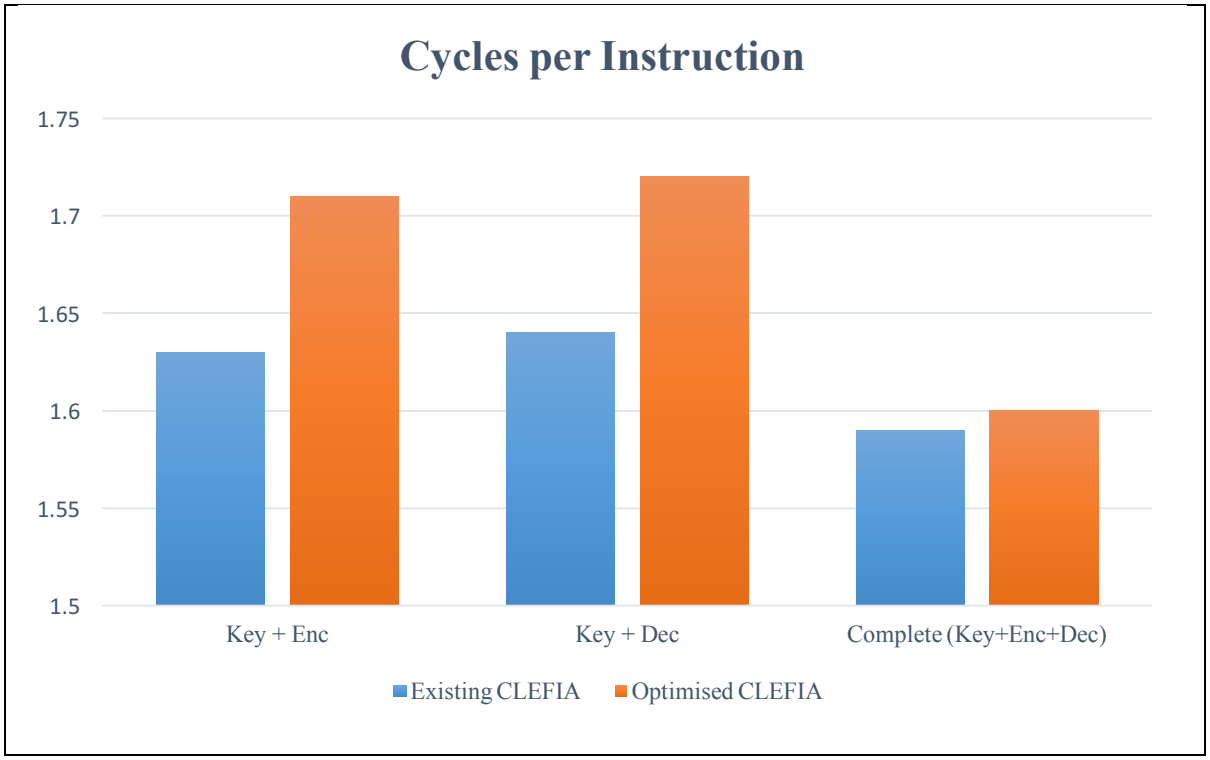

Figure 5.7: Comparison of CPI for existing and optimized CLEFIA

From the Figures 5.5 and 5.6 it can be concluded that there is reduction in execution cycles of optimized CLEFIA algorithm and in complete process the reduction of 75% is achieved. In Figure 5.7 the overall execution of the algorithm shows an achievement of 10% in cycles per instruction, which makes its performance better than the existing one.

5.3 HARDWARE IMPLEMENTATION

The hardware implementation is done using Xilinx Vivado Tool 2016 in Verilog Hardware Language. The synthesized code is generated for xc5vlx50t-3 board of Virtex-5 Family. It works on 65nm technology, has four LUTs per slice, 6-input LUTs and operates at maximum frequency of 540 Mhz. CLEFIA is implemented by optimizing S-Box S_0 and Diffusion matrices. The Flow of complete encryption and decryption process is shown in Figure 5.8 and Figure 5.9-5.10 shows behavioral simulation results of encryption and decryption process and the Figure 5.11 is the RTL of the synthesized code. The encryption process takes 18 clock cycles to generate cipher text after key scheduling.

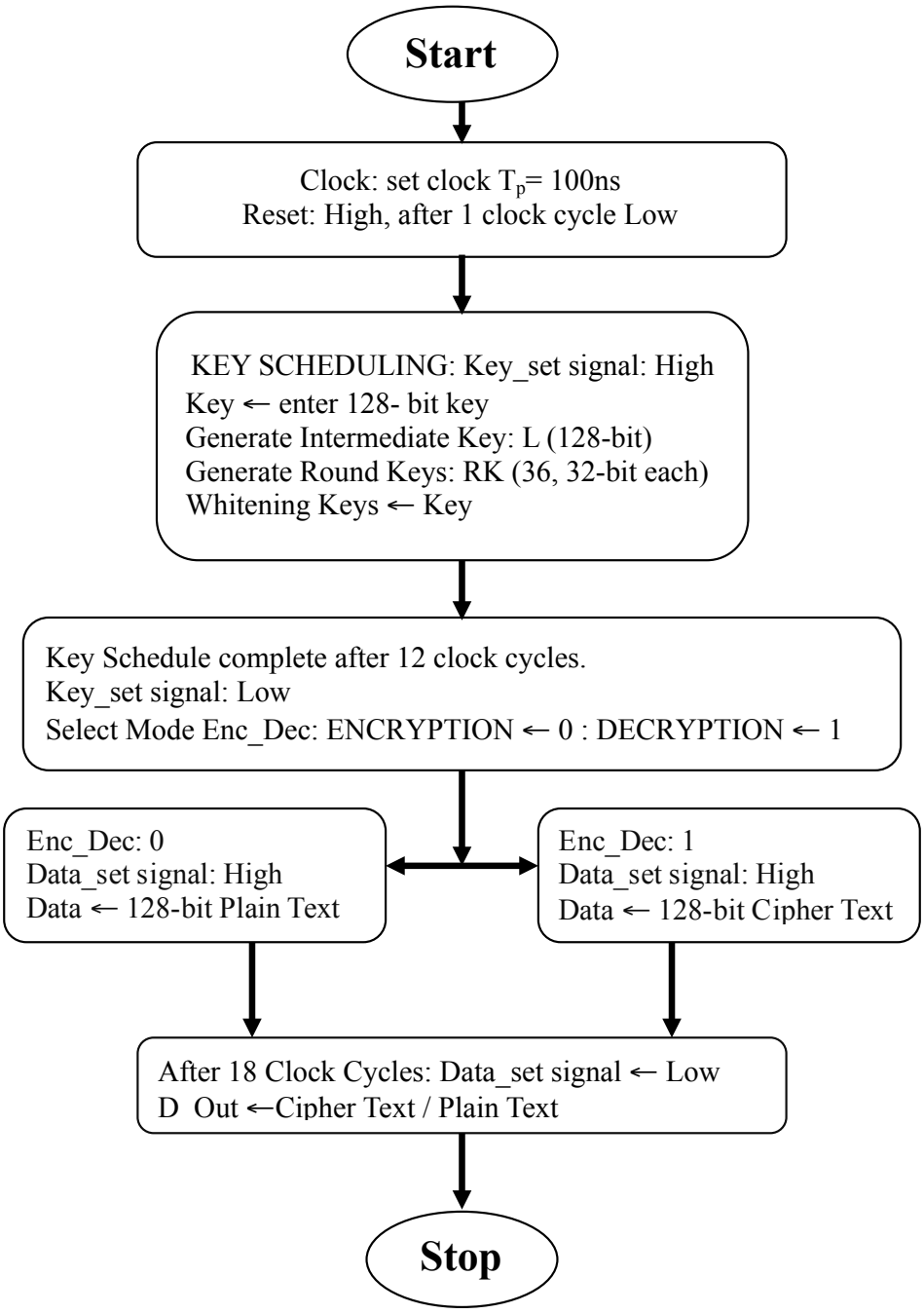

Figure 5.8: Flowchart for Hardware Code

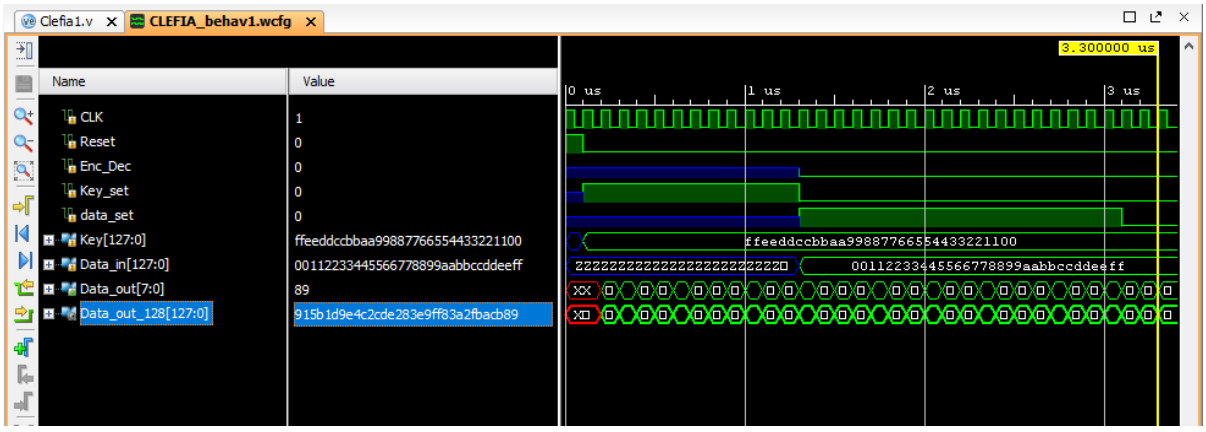

Figure 5.9: Output of Encryption Process

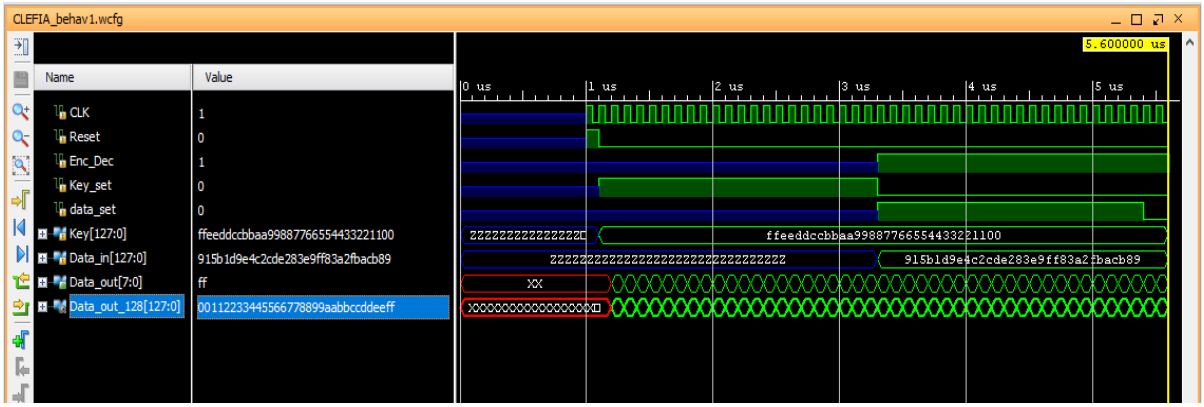

Figure 5.10: Output of Decryption Process

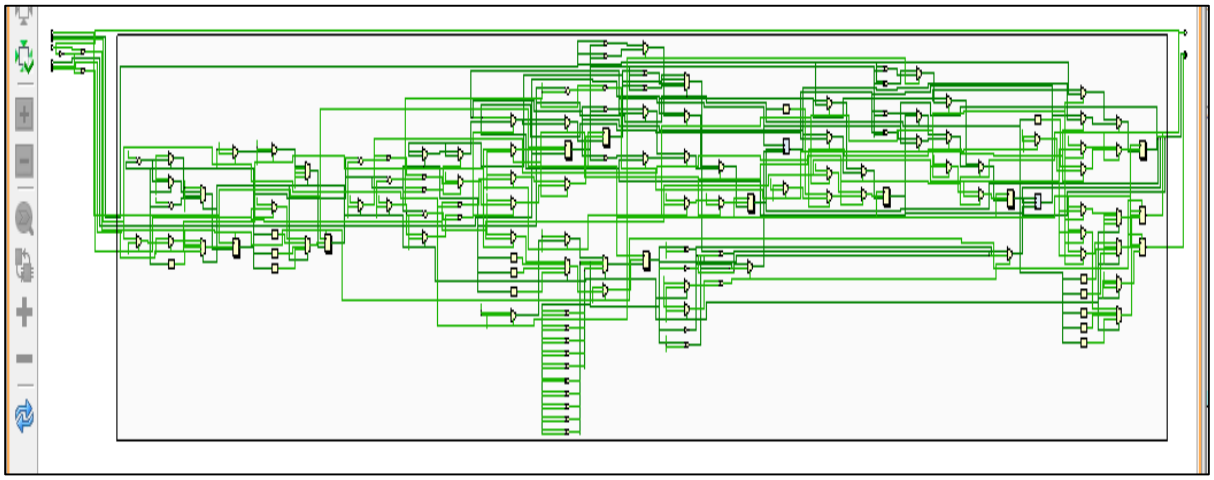

Figure 5.11: RTL Design

The hardware result parameters are obtained by synthesizing the code on the target board. Table 5.4 shows the performance result of Encryption process and Key Generation for both implementations and compared with the existing compact implementation of CLEFIA in [48]. The parameters taken for comparison are area consumed in terms of slices, throughput and efficiency of the algorithm. The aim is to obtain minimum area consumed by the algorithm and high efficiency on resource constrained devices used in E-Healthcare applications.

Table 5.5: Hardware Results of CLEFIA

Parameters	Existing Clefia [48]	Optimized (Using Boolean S-box)	Optimized (Using Look Up Tables for S-Boxes)
Key Size	128-bits		
Block Size	128-bits		
Slices	361	291	411
Throughput (Gbps)	1.28	1.28	1.28
Efficiency (Kbps/slices)	3546	4168	3114

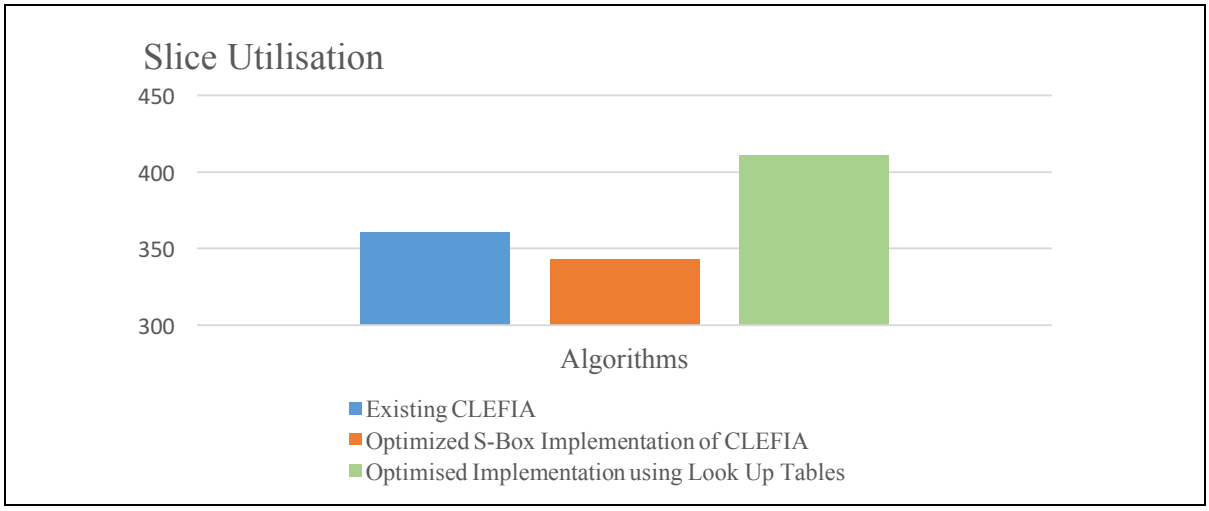

Figure 5.12: Comparative Analysis of slice Utilization Parameter for Existing and Optimized Clefia Algorithm

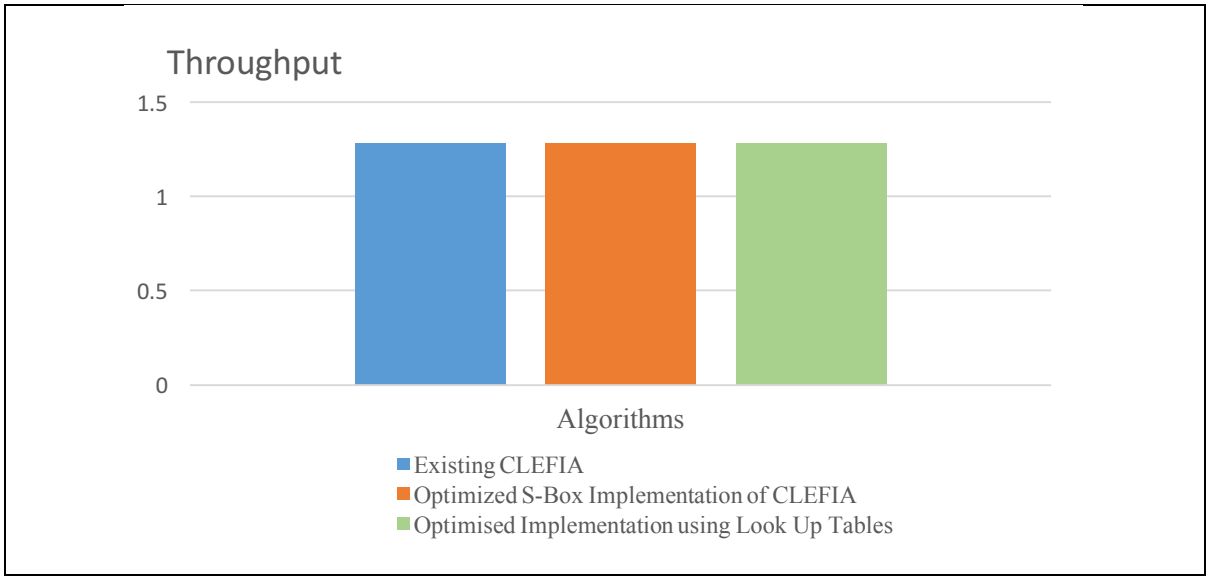

Figure 5.13: Comparative Analysis of Throughput Parameter for Existing and Optimized Clefia Algorithm

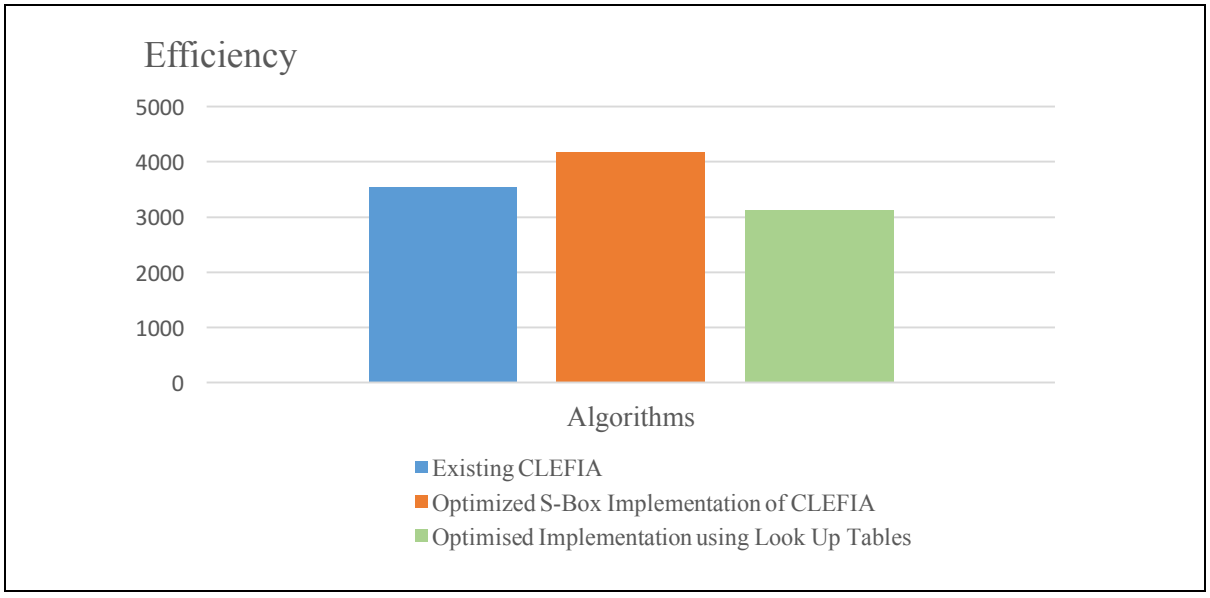

Figure 5.14: Comparative Analysis of Efficiency Parameter for Existing and Optimized Clefia Algorithm

The Figure 5.10 compares the area consumed in terms of Slices on the FPGA board. CLEFIA using Boolean S-box shows the least area utilized which is 19% less than the existing. Figure 5.11 shows the throughput which appears same as block size used, clock cycle count for encryption and operating frequency is same for all three implementations. Figure 5.12 compares the efficiency of the implementation, which is best for CLEFIA using Boolean S-Box in place of lookup tables. Figure 5.13 shows the post implementation design for CLEFIA algorithm, depicting slices used and zoomed view of one slice which has LUTs, MUXes and Registers.

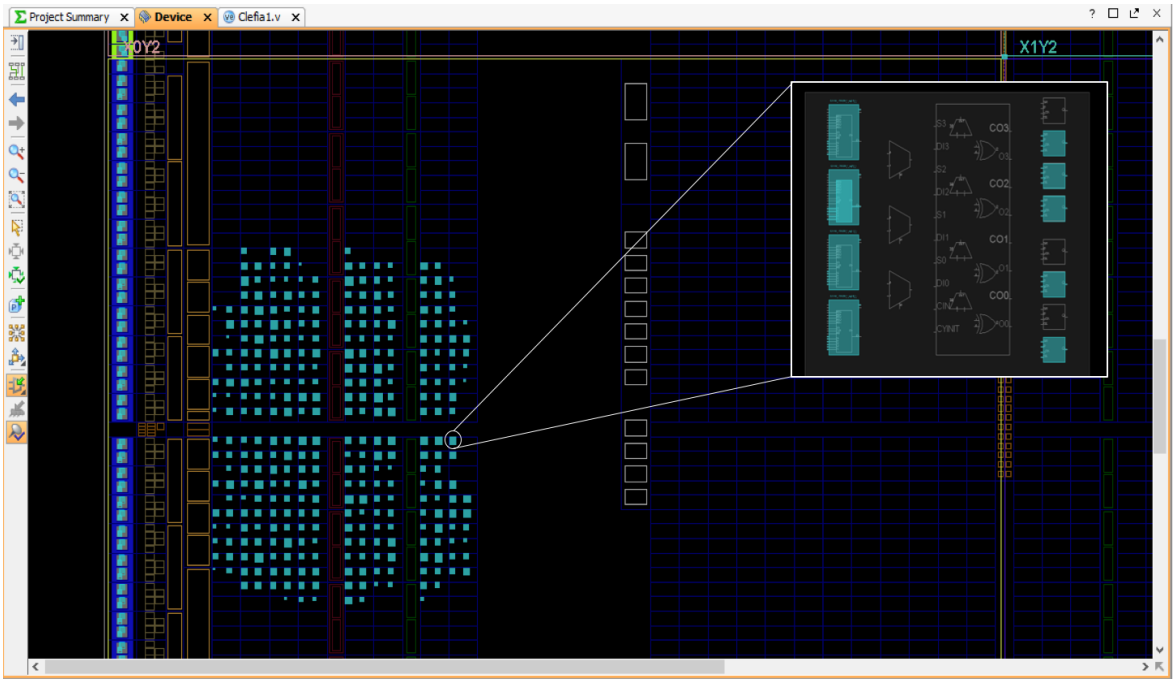

Figure 5.15: Post implementation Result

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

Internet of Things is unique technology which connects everything with the internet through physical objects. It provides various applications that can be incorporated in our daily lives for comfortable living. E-Healthcare is one of the major application of IoTs which aims in improving human health. Despite the benefits of this technology there are few challenges faced in terms of information threat and security as the medium to share the information is susceptible to many attacks and data misuse. A survey on E-Healthcare attacks and their security is done. From the survey, it's found that cryptography algorithms are used for data security and authentication purposes in E-Healthcare such as AES, RSA, ECC *etc.* The E-Healthcare devices have sensor nodes; RFID tags that have limited memory available so lightweight algorithms are used as they show good performance on small devices.

In this work, NIST recommended Lightweight cipher CLEFIA is studied and optimized at both software and hardware level. The software implementation of Clefia algorithm is done on GCC and simulated on T-sim simulator. The results reflect that the cycles count is reduced by 75% and CPI is improved by 5.8%. The hardware implementation of CLEFIA algorithm is done using Xilinx Vivado tool and synthesized code is generated for Virtex-5 family board. Performance analysis is done on the basis of area, throughput and efficiency. Area utilization in terms of Slices is given and optimized CLEFIA using Boolean S-Box uses 29% less area and is 33% more efficient than the CLEFIA implementation using Look Up Tables.

In the lightweight ciphers to provide the same level of security as in conventional ciphers, the number of rounds are increased. The large number of rounds degrades the performance of the cipher. Therefore, it required to design a lightweight cipher which provides fast confusion and diffusion in less number of rounds. Secure hybrid model can be designed using CLEFIA and any authentication algorithm which will provide both encryption and authentication of information shared on small devices.

References

1 A. Whitmore et al., “The Internet of Things—A survey of topics and trends,” in *Information Systems Frontiers,* Springer, April 2015, vol. 12(2), pp. 261-274.

2 P. Shah et al., “Applications and Challenges Faced by Internet of Things - A Survey,” in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering Future Intelligent Vehicular Technologies,* Springer, 2017, pp.182-188.

3 J. S. Jeong et al., “A Design Characteristics of Smart Healthcare System as the IoT Application,” in *Indian Journal of Science and Technology*, October 2016, vol. 9(37).

4 L. Catarinucci et al., "An IoT-Aware Architecture for Smart Healthcare Systems," in *IEEE Internet of Things Journal*, Dec. 2015, vol. 2(6), pp. 515-526.

5 Alok Kulkarni and Sampada Sathe, “Healthcare applications of the Internet of Things: A Review,” in *International Journal of Computer Science and Information Technologies*, 2014, vol.5, pp. 6229-6232.

6 D. Hankerson et al., “Guide to Elliptic Curve Cryptography,” in *Springer*, Verlag, 2003.

7 Kritika Acharya et al., “Analysis of Cryptographic Algorithms for Network Security,” in *International Journal of Computer Applications Technology and Research*, 2014, vol. 3(2), pp. 130-135.

8 Vishwa gupta et al., “Advance cryptography algorithm for improving data security,” in *International Journal of Advanced Research in Computer Science and Software Engineering*, 2012, vol. 2(1).

9 Anuj Sehgal et al., “Management of Resource Constrained Devices in the Internet of Things,” in *IEEE Communications Magazine*, 2012, vol. 50(12).

10 A. Juels and S.A. Weis, “Authenticating Pervasive Devices with Human Protocol,” in Advances in Cryptology-Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2005, vol. 3621.

11 A. Kerry et al., “Report On Lightweight Cryptography,” *NIST*, March 2017.

12 A. Bogdanov et al., “PRESENT: An Ultra-Lightweight Block Cipher,” in *Cryptographic Hardware and Embedded Systems - CHES 2007 Lecture Notes in Computer Science*, Springer, 2007, pp. 450-466.

13 R. Beaulieu et al., “The SIMON and SPECK lightweight block ciphers,” in *Proceedings of the 52nd Annual Design Automation Conference,* 2015, pp. 1-6.

14 S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain and K. S. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," in *IEEE Access*, 2015, vol. 3, pp. 678-708.

15 K. Niranjana Devi and R. Muthuselvi, “Secret Sharing of IoT Healthcare Data Using cryptographic algorithm,” in *International Journal of Engineering Research*, May 2016, vol. 5(4), pp. 790-991.

16 Jinyuan Sun et al., “HCPP: Cryptography Based Secure EHR System for Patient Privacy and Emergency Healthcare,” in *31st international conference on Distributed Computing System*, 2011, pp. 373-382.

17 L. Yehia et al., “Hybrid Security Techniques for Internet of Things Healthcare Applications,” in *Advances in Internet of Things*, July 2015, vol. 5, pp. 21-25.

18 A Boonyarattaphan et al., “A security framework for e-Health service authentication and e-Health data transmission,” in *9th International Symposium on Communications and Information Technology*, Icheon, 2009, pp. 1213-1218.

19 J. Kim et al., “Towards a Security Policy for Ubiquitous Healthcare Systems,” in Proc. *1st international Conference on Ubiquitous Convergence Technology*, 2006, pp. 263-272.

20 X.H. Le et al., “An efficient mutual authentication and access control scheme for wireless sensor network in healthcare,” in *Journal of Networks*, 2011, vol. 27, pp. 355–264.

21 Pardeep Kumar and and Hoon-Jae Lee, “Security Issues in Healthcare Applications Using Wireless Medical Sensor Networks: A Survey,” in *Sensors*, Basel, Switzerland, 2017, vol. 12.1, pp. 55–91.

22 W.A. Al-Hamdani, “Cryptography based access control in healthcare web systems,” in *Proceedings of 2010 Information Security Curriculum Development Conference*, USA, October 2010, pp. 66–79.

23 Moshaddique Al Ameen et al., “Security and Privacy Issues in Wireless Sensor Networks for Healthcare Applications,” in *Journal of Medical Systems*, February 2012, vol. 36(1), pp 93-101.

24 D.C. Jinwala et al., “Investigating and Analysing the Light-weight ciphers for Wireless Sensor Networks,” in *INFOCOMP Journal of Computer Science*, June 2009, vol. 8(2), pp. 39-50.

25 Y.S. Shiu et al., “Physical Layer Security in Wireless Networks,” *IEEE Wireless Communications*, 2011, vol. 1, pp. 66-74.

26 Bassam J. Mohd et al., “A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues,” in *Journal of Network and Computer Applications*, December 2015, vol. 58, pp. 73-93.

27 M. Cazorla et al., “Survey and benchmark of lightweight block ciphers for wireless sensor networks,” in *International Conference on Security and Cryptography (SECRYPT)*, Reykjavik, Iceland, 2013, pp. 1-6.

28 H. Krawczyk, “The order of encryption and authentication for protecting communications (or: How secure is SSL?),” in *Advances in Cryptology—CRYPTO 2001*, Springer, Berlin, 2001, pp. 310-331.

29 J. Lan et al., “A Random Number Generator for Low Power Cryptographic Application,” in SoC *Design Conference (ISOCC)*, 2010 International. IEEE, 2010, vol. 1, pp. 328-331.

30 J. John, “Cryptography for Resource Constrained Devices: A Survey,” in *International Journal on Computer Science and Engineering,* 2012, vol. 4(11), pp. 1766.

31 J. H. Kong et al., “A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments,” in *Journal of Network and Computer Applications*, 2015, vol. 49, pp. 15-50.

32 C. H. Kim, “Improved Differential Fault Analysis on AES Key Schedule”, in *IEEE Transactions on Information Forensics and Security*, 2012, vol. 7(1), pp. 41-50.

33 P. Hamalainen et al., “Design and implementation of low-area and low-power AES encryption hardware core,” in *Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference on. IEEE*, 2006, pp. 577-583.

34 M. Kotegawa et al., “A new compact hardware architecture of S-Box for block ciphers AES and SM4,” in *EICE Electronics Express*, May 2017, vol.14(11).

35 M. Kotegawa et al., “Optimization of Hardware Implementations with High-Level Synthesis of Authenticated Encryption,” in *Bulletin of Networking, Computing, Systems, and Software*, 2016, vol. 5(1), pp. 26–33.

36 S. M. Soliman et al., “Efficient implementation of the AES algorithm for security applications,” in *29th IEEE International System-on-Chip Conference*, Seattle, WA, 2016, pp. 206-210.

37 T. Akishita and H. Hiwatari, “Very Compact Hardware Implementations of the Block Cipher CLEFIA,” in *Selected Areas in Cryptography - SAC 2011 ser. LNCS A. Miri and S. Vaudenay Eds. LNCS*, Springer, Heidelberg, 2012, vol. 7118, pp. 278–292.

38 L. Yehia et al., “Hybrid Security Techniques for Internet of Things Healthcare Applications,” in *Advances in Internet of Things*, July 2015, vol. 5, pp. 21-25.

39 A. A. Moshaddique et al., “Security and Privacy Issues in Wireless Sensor Networks for Healthcare Applications,” in *Journal of Medical Systems*, February 2012, vol. 36(1), pp 93-101.

40 Sony Corporation. The 128-bit Block Cipher CLEFIA Design Rationale, (June 2007). Available at https://www.sony.net/Products/cryptography/clefia/download/index.html (Accessed on 17th August 2016)

41 T. Shirai et al., "The 128-Bit Blockcipher CLEFIA (Extended Abstract), " in A. Biryukov (Ed.) Fast Software Encryption - FSE 2007 Lecture Notes in Computer Science, Springer, 2007, vol. 4593, pp. 181-195.

42 Sony Corporation. The 128-bit Block Cipher CLEFIA Algorithm Specification, (June 2007). Available at https://www.sony.net/Products/cryptography/clefia/download/index.html (Accessed on 17th August 2016)

43 Sony Corporation. The 128-bit Block Cipher CLEFIA Security and Performance Evaluations, (June 2007). Available at https://www.sony.net/Products/cryptography/clefia/download/index.html (Accessed on 17th August 2016)

44 C. J. Benvenuto, “Galois field in cryptography,” University of Washington, Seattle, 2012.

45 B. J. Gough, “An Introduction to GCC-For the GNU compilers gcc and g++,” in *Software Design Lecture Notes the GCC Compiler*. Available: http://www.network-theory.co.uk/docs/gccintro/gccintro_3.html

46 Diego Novillo, “The Inner Workings of GCC,” in *Red Hat Magazine*, Issue. 2, December 2004.

47 Sony Global. Reference Code v1.0.1. Available at https://www.sony.net/Products/cryptography/clefia/download/index.html (Accessed on 17th August 2016)

48 M. M. Kermani and R. Azarderakhsh, “Efficient fault diagnosis schemes for reliable lightweight cryptographic ISO/IEC standard CLEFIA benchmarked on ASIC and FPGA,” in *IEEE Trans. Ind. Electron.*, Dec. 2013, vol. 60(12), pp. 5925–5932.

Publications

- Isha Bhardwaj, Ajay Kumar, Manu Bansal, "A Review on Lightweight Cryptography Algorithms for Data Security and Authentication in IoTs," in 4th International Conference on Signal Processing, Computing and Control(ISPCC-2017), sponsored by IEEE. (Accepted)

601562011_Isha_Bhardwaj

ORIGINALITY REPORT

%10	%7	%8	%0
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.iacr.org Internet Source	%2
2	Riazul Islam, S. M., Daehan Kwak, Md Humaun Kabir, Mahmud Hossain, and Kyung-Sup Kwak. "The Internet of Things for Health Care: A Comprehensive Survey", IEEE Access, 2015. Publication	%1
3	Lecture Notes in Computer Science, 2012. Publication	<%1
4	ftp.arcane-networks.fr Internet Source	<%1
5	www.ijser.org Internet Source	<%1
6	uclab.khu.ac.kr Internet Source	<%1
7	cryptrec.go.jp Internet Source	<%1
8	ijarcce.com Internet Source	<%1

9	Yehia, Lobna, Ayman Khedr, and Ashraf Darwish. "Hybrid Security Techniques for Internet of Things Healthcare Applications", Advances in Internet of Things, 2015. Publication	<%1
10	Panayiotis Kotzanikolaou. "Appendix A: Cryptography Primer: Introduction to Cryptographic Principles and Algorithms", Network Security, 06/06/2007 Publication	<%1
11	www.academypublisher.com Internet Source	<%1
12	Lecture Notes in Computer Science, 2015. Publication	<%1
13	Naveed, Islam, and William Puech. "Data Cryptography", Signal and Image Processing for Biometrics Naït-Ali/Signal and Image Processing for Biometrics, 2013. Publication	<%1
14	ieeexplore.ieee.org Internet Source	<%1
15	mdpi.com Internet Source	<%1
16	Whitmore, Andrew, Anurag Agarwal, and Li Da Xu. "The Internet of Things—A survey of topics	<%1