A Thesis Report

On

Hardware Implementation of Hybrid Approach for Data Encryption and Authentication in Smart Cards Security

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

MASTER OF TECHNOLOGY

In

VLSI DESIGN

Submitted By

Jaskaranbeer Kaur

Roll No. 601562012

Under Supervision of

Mrs. Manu Bansal

(Assistant Professor, ECED)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY, PATIALA, PUNJAB

JULY, 2017

DECLARATION

I, Jaskaranbeer Kaur hereby declare that the work presented in this thesis entitled "**Hardware Implementation of Hybrid Approach for Data Encryption and Authentication in Smart Cards Security**" 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.

Jaskaranbeer Kaur

Date: Aug 10, 2017

JASKARANBEER KAUR

Roll No. 601562012

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

Manu Bansal

Date: 10/8/17

Mrs. Manu Bansal

Assistant Professor

ECED, TU, Patiala

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to Mrs. Manu Bansal, Assistant Professor, Electronics and Communication Engineering Department, Thapar University, Patiala for her patient guidance and support throughout this report. I am truly very fortunate to have the opportunity to work with her. I found this guidance to be extremely valuable.

I am also thankful to our Head of the Department, Dr. Alpana Aggarwal for providing us adequate environment in carrying the work.

I would like to extend my gratitude to all those people who directly or indirectly helped me in the process and contributed towards this work.

Jaskaranbeer Kaur

601562012

ABSTRACT

With the increase in online applications, remote user verification and secure authentication of digital identities has become a concerned issue. Smart cards are considered as best solution for providing security and authentication to highly classified data such as financial records, defence credentials, health care records, ATM PIN information etc. in many real life applications. The successful use of smart cards as security token led to realization of much potential to make smart cards more secure but some security threats as man-in-the-middle attack, impersonation attacks, replay attack, insider attack etc. may reduce the efficiency of smart cards. To resist the impact of these threats, various cryptography algorithms are applied to smart cards applications. Due to resource constrained environment in smart cards, implementation of existing cryptography algorithms in these devices identified a major research gap in terms of area consumption.

This thesis work attempts to resolve these resource constrained issues and provides efficient data security and authentication for smart card applications. In proposed work, a hybrid algorithm of encryption and authentication is implemented. For providing encryption, optimized lightweight cryptography algorithm PRESENT is used and authentication of data is achieved by data hiding with redundant bits based redundant bit security algorithm. The proposed technique is implemented with two PRESENT S-Box approaches of look up table and Boolean logic. Resultant technique is analysed in terms of performance parameters of cryptography security and resources utilization. This analysis reflects that proposed technique is efficient to ensure security and authentication in smart cards with less area consumption.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF SMART CARDS

With the evolution in technology, digitalization of real life applications leads to various security concerns. To resolve these security issues smart cards are preferred. Smart cards are built in microprocessor tokens intended to perform diverse range of applications. Smart Cards are portable devices capable of storing sensitive information, user identity validation, authentication, data processing and provide accuracy, integrity, customization to ensure security [1]. Smart cards are considered as temper resistant device which means that information stored in these cards can't be modified easily. Smart cards contain a computer chip embedded in it so, these are referred as chip card or integrated chip cards. Smart cards send or receive signals to applications via smart cards reader.

1.1.1 History of Smart Cards:

Initially in 1968, patent was filed for using plastic as a carrier of microchips. In 1970 the concept of smart cards was proposed for the first time. Smart cards were first patented in 1978. After testing of smart cards using microprocessor in 1982, in 1990 smart cards were used in GSM phone equipment for the first time. A prepaid card project was started in 1992 in Denmark. Then Federal employee smart cards for the identification purposes were used in 1999. After that various types of smart cards were developed for wide range of applications [2].

1.2 SMART CARDS ARCHITECTURES

1.2.1 Smart Cards 3 Layered Architecture:

The 3 layered architecture for smart card is shown in figure 1.1. Bottom layer of the architecture is hardware abstraction layer containing basic hardware components such as I/O, Coprocessor and memory components such as RAM, ROM, and EEPROM. This layer is the interface of smart cards to physical world. Middle layer is the operating system layer which acts as interface between hardware and application layer. Operating system layer performs the basic functions of hardware resource management, memory management, I/O management in smart cards. Operating system is placed in ROM. Topmost layer is application layer which is inaccessible to user. This layer contains all the applications of the smart cards systems in EEPROM. Communication protocols for data in smart cards are governed by this application layer.

Any command message considered as data packet containing the complete instructions and data is sent by the application layer and response message is returned to the application

layer from the card which is called application protocol data units, following some communication protocol. Communication of the card and user is provided by sending 5 byte including one class byte, three parameters byte and one instruction byte transmission header. Error checking is done by parity bit of header byte. If the 5 byte data is transmitted correctly, one byte acknowledgement is transmitted.

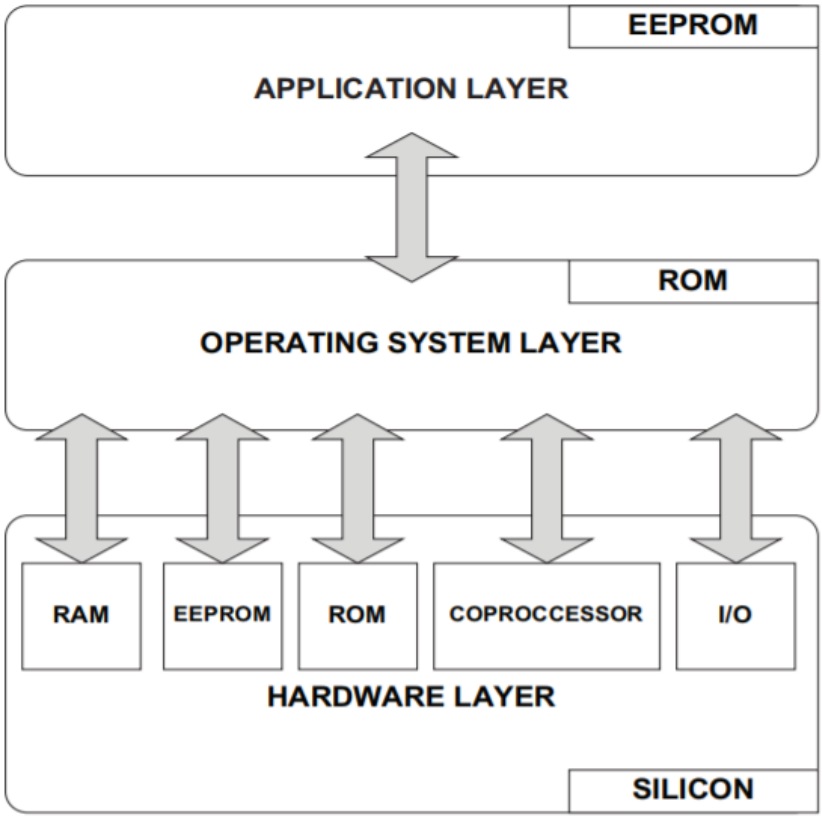

Figure 1.1 Smart Cards 3 Layered Architecture [2]

If some additional data is transmitted then a number of bytes are sent to application layer. When whole instruction is transmitted then a response message is delivered to application layer.

The 3 layered smart cards architecture provides security to less extent as the data security and error checking depends on some additional 5 byte instruction header. With the advancement of various architecture technologies such as mixed technology, semiconductor technology and power consumption techniques a new 5 layered architecture for smart cards has been proposed.

1.2.2 Smart Cards 5 Layered Architecture:

This 5 layered architecture contained additional crypto component in the system which apply various cryptography protocols for the security of data. Figure 1.2 represents new 5

layered smart cards architecture which includes CPU as basic component along with embedded hardware, IP blocks, Memory management and security components.

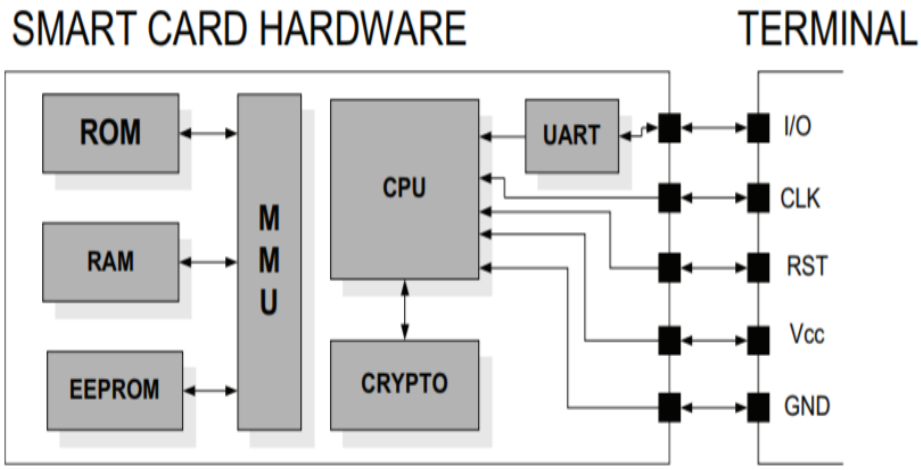

Figure 1.2 Smart Cards 5 Layered Architecture [2]

ROM of the card contains operating system. It is used for storage of codes, lookup tables and is efficient in terms of area and power requirements. RAM is used for the temporary storage of codes and data routines. RAM has restricted small size and it is divided into regions for stack, variables, cryptography algorithms, registers and I/O buffers. EEPROM is non-volatile memory. It occupies almost 50% of the smart cards area. For storage it is divided into two regions, in the first regions manufacturing data is stored and in the second region, pointers and tables are loaded. ROM is operating system storage memory, RAM is CPU working memory and EEPROM is for the storage and retrieval of data. MMU handles all the memory requests to the CPU. I/O lines are used for the communication to outside world. UART handles the request and response messages. Crypto processor is basically for providing security to data by operating cryptographic tasks. Power is supplied to the chip by VCC and GND contacts. CLK signal is provided for synchronization and RST signal takes the card to initial state.

Whenever smart card has to make data secure provided at the I/O lines, it is powered up by VCC. UART receives the request command from terminal. Further CPU requests to MMU. Data at the smart card is stored in the ROM and RAM and sometimes on EEPROM conditionally. This data is one way provided to the crypto component where encryption and decryption is performed. Then again data is provided to CPU.

1.3 TYPES OF SMART CARDS

Various types of smart cards are used by individuals for identification, authentication and data storage elements. Depending on the card's logical structure and the application for which it is used, smart cards are of five types as described below:

- **Contact Cards:** Contact smart cards have contact pads of approximately 1 square centimetre area. When smart card is inserted into smart card reader these pads provide physical and electrical connectivity to communicate between smart cards and user [3].

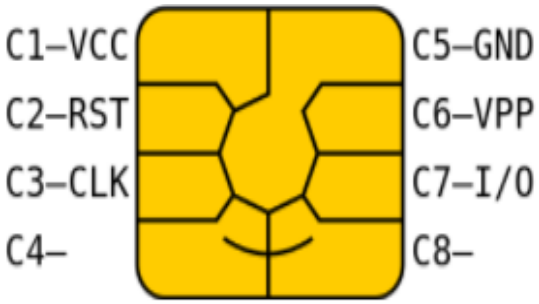

Figure 1.3 Contact Based Smart Cards [3]

VCC and GND provide power to smart cards, RST is for reset card communication, CLK is used for provision of CLK timing signal, I /O pins are for the communication to host. VPP pin is for programming voltage input (optional). Pin C4 and C8 are for USB interfaces.

- **Contactless Smart Cards:** In contactless smart cards, communication is achieved by radio frequency field. These cards do not have extra contact pads but also required an IC antenna in smart cards. When the card is inserted into smart card reader, RF field is generated by the smart card reader which induces electric current in smart cards IC antenna and energy is transferred by the reader. This energy works as a power source for smart cards [4].

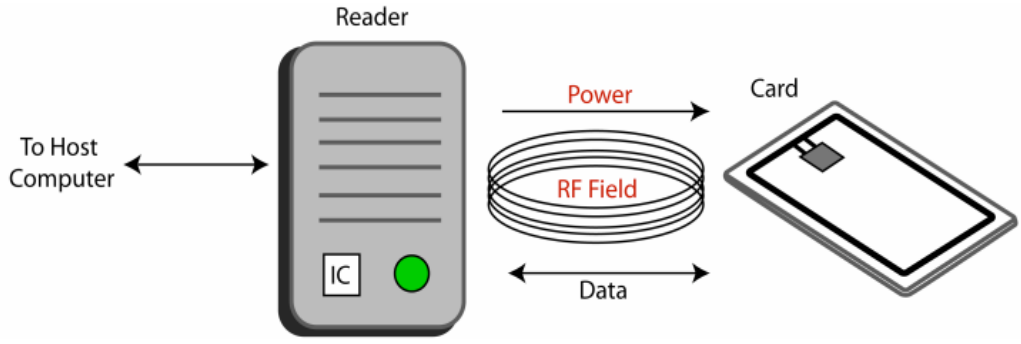

Figure 1.4 Contactless Smart Cards [4]

- **Dual Interface Smart Cards:** Dual interface smart cards imply features of both the contact cards and contactless cards. These cards have some pads for communication and also contain antenna for providing power source to IC. Dual interface smart cards are designed to

support multiple applications. For example, these cards provide contact payments and contactless e-commerce transactions [4].

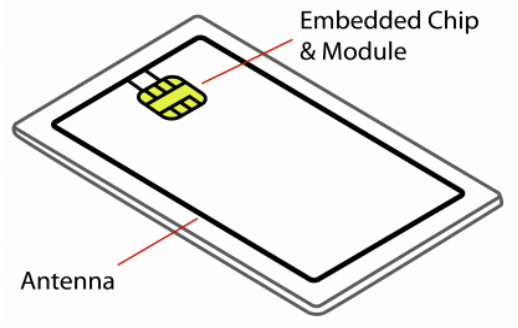

Figure 1.5 Dual Interface Smart Cards [4]

- **Memory Cards:** Memory circuits are embedded in memory cards which perform the operation of read, write and store information to any particular location to prevent it from unauthorized access. Any data manipulation or modification can't be done on these memory cards. Hence, memory cards have limited functionality [3].

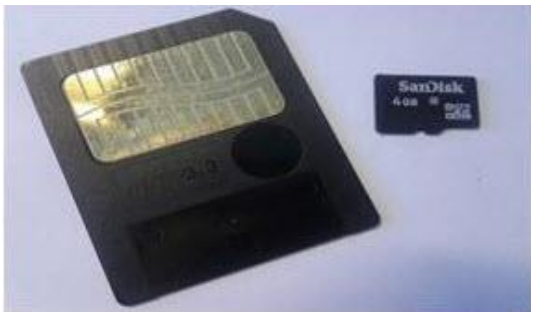

Figure 1.6 Memory Smart Card [3]

- **Microprocessor Based Smart Cards:** These smart cards have a microprocessor chip along with memory blocks. Data manipulation and processing can be performed on these cards. The operation of microcontroller chip is controlled by either dynamic or fixed operating systems [3]. Operating system also allows the user to issue its own commands, data structure etc. All these user defined concerns are stored in non-volatile EEPROM memory [4].

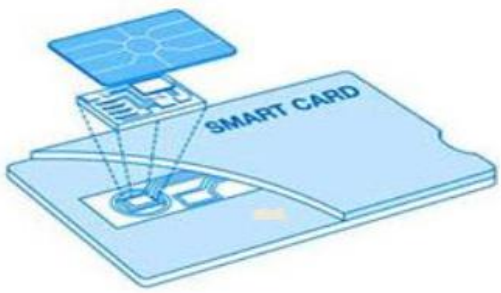

Figure 1.7 Micro-processor Based Smart Cards [4]

1.4 APPLICATIONS OF SMART CARDS:

Due to small size, portable and large device storage capacity, smart cards are used in variety of applications for identification, authentication and data storage.

- **Finance:** Contactless smart cards are used for secure and fast debit and credit cards payments. Any debit and credit cards in the banks are should be smart cards with proper issue date. Smart cards provide electronic purse for the replacement of coins in small transactions at counter. In e-commerce applications for secure transactions across the internet [1].
- **Government Resources:** Indian government use smart cards technology for identification by having their personal information linked to these resources. For example, Aadhar Card. Also the Department of Defence Common Access Card imply smart card technology for the military credentials. The Federal government issued smart cards ID to employees to meet Homeland Security Presidential Directive [3].
- **Health Care:** In health care services, smart cards are used for the identification of patients, maintaining patient's records, to reduce medical frauds. Smart cards based ID cards are also carried by the doctors and nurses so as to ensure secure access control to emergency information [5]. Smart cards based consumer health cards are helpful to ensure medical insurance eligibility [1].
- **Educational Systems:** Smart cards are applied in schools and colleges for student identification. Memory smart cards are used for maintaining attendance, financial and academic records. Also the electronic purse feature of smart cards are helpful in paying institutional fee, canteen bills etc [3].
- **Mobile Communication area:** Smart cards provide valid identification for GSM digital mobile phones. Also smart cards work as data storage element to store necessary information about new communication device and billing orders. SIM (Subscriber Identity Module) used in smart phones are also a type of smart card [5].
- **Computer Security:** In case of computer security, smart cards are used as security tokens. Smart cards can be applied for single login to any computer or browser. Mozilla Firefox Web Browser saves digital certificates using smart cards [1].
- **Transportation:** Smart cards provide E-ticketing for transit emergency situations. Smart cards are also used for transit fee payment, EMV and parking [1].

1.5 SECURITY ISSUES IN SMART CARDS

Now days smart cards very popular to control sensitive information. So, at the same time the security of information infrastructure from eavesdroppers is becoming major requirement. Although smart cards are temper resistant devices and also offer a variety of security mechanisms, yet there are some attacks possible on the smart cards. In case of various attacks on smart cards any third party may be able to decode secure information from the smart cards. Also by using various technological techniques an attacker may read the smart cards memory bus data.

Some of the possible attacks on smart cards are: [6]

- **Insider Attack:** Insider attacks are the malicious attacks incurred by the user that has authorized access to the smart cards. Authorized party can attack smart cards more easily as they have knowledge about smart cards operating system, communication nodes etc. In authentication schemes, a user has password to access the server. If same password is used by any another server then the particular user can access data of another server. This leads to insider attack. Countermeasure for this attack includes password transmission by applying some operations on it and for each user this operation must be random.
- **Man-In-The-Middle Attack:** Man-in-the-middle attack, a form of active eavesdropping, occurs when an attacker sits in between the user and the server which will have access to the smart cards information without the knowledge of server and user. This will lead to leakage of private information such as PIN, patient history, user identity etc. and it will exploit processing of various transactions. This attack can be prevented by applying the two level authentication by authentication centre. Authentication server will send the message containing secret identity of user and server. This will lead to failure of access by any third adversary.
- **Replay Attack:** In replay attack, adversary access the login or authentication information varies it and transmit to the server to impersonate the authorized user. When a user requests to server for login by transmitting login information over channel then the adversary access that information, change some contents of it and make the information suitable for adversary login to server and retransmit it. Hence genuine user is not able to login. Countermeasure for replay attack is use of timestamps along with hash mechanism for sending login information. Data is sent in the form of (hash-original data-private key) with timestamps. Both the hash mechanism and private keys are required to extract information and if timestamp threshold is small, particular user can access the request only once, so adversary fails to extract information in one attempt.

- **Impersonation Attack:** In this attack the adversary attempts to impersonate the access information to masquerade genuine user. Attacker sends ID to the server to act as legal user. On the other hand the server issues password and smart card to the user. On getting the password attacker verifies a suitable pair of ID and password and hence made it possible to access information.
- **Denial of Service Attack:** Denial of service attack makes the network resources and communication facilities unavailable for smart cards authentication. One way to attempt this attack is that the adversary sends wrong login requests continuously to make the server busy and prevents authorized user to login at the server. These attacks can be prevented by applying some filter protocol at the server to block the extra traffic.
- **Stolen Verifier Attack:** An adversary steals the verification and authentication data from the server. From this stolen data, attacker generates the communication data between the user and server and sends it to the server. It the adversary sends correct communication data, then it impersonate the legal user for next authentication session. This attack can be resisted by using hash functions and ECC based authentication attack which eliminates the use of any verification tables at the server.

To make smart cards resistant against these attacks, various cryptographic algorithms are used [3].

1.6 OVERVIEW OF CRYPTOGRAPHY FOR SMART CARDS SECURITY

Cryptography originates from Greek word "Kryptos" which refers to "Hidden Secrets". Cryptography is concerned with hiding of secret information. Cryptography is the study of techniques to make data indistinguishable to prevent it from unauthorized access. Cryptography encodes highly confidential data to an unreadable format to protect it from malicious third parties and after secure transmission of data again decode it into readable format**.** Thus, cryptography is the art and science of making highly classified data secure from intruders. Figure 1.8 represents brief overview of cryptography.

1.6.1 Terminology:

Main terms related with cryptography are given below:

- **Plaintext:** Original data that has to be transmitted is known as plaintext.
- **Cipher text:** The encoded data which is in intelligible form is called cipher text.
- **Encryption:** Encryption is the process of converting the plain data into highly confidential form at transmitter side.

- **Decryption:** Decryption is the process of converting data back to its original form at receiver side.

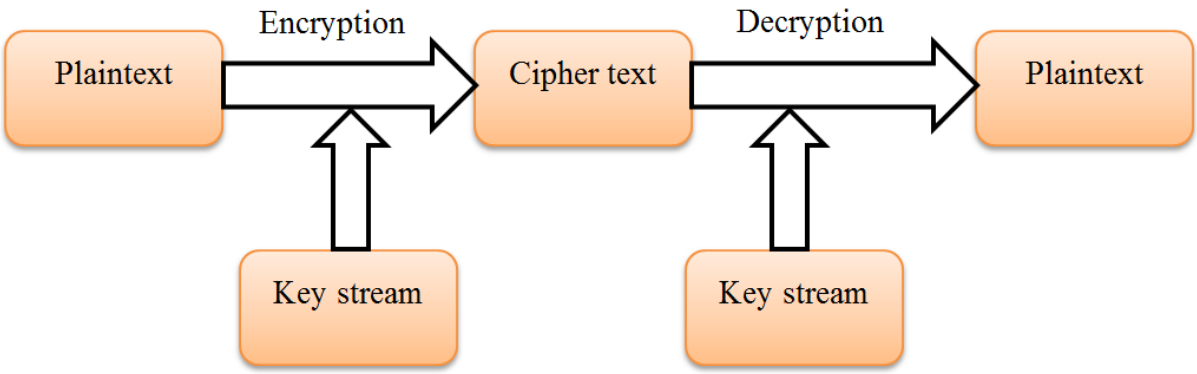

Figure 1.8 Overview of Cryptography

1.6.2 Goals of Cryptography [7]

Cryptography is concerned with provision of the given goals to counter security issues:

- **Authentication:** Authentication refers to verification of user's identity. Receiver checks the identity whether the data transmitted by an authorized user or any wrong identity. It ensures that any kind of intruder doesn't have access to sensitive data.
- **Confidentiality or Privacy:** Confidentiality assures that transmitted data is not disclosed to any unauthorized party. Confidentiality is achieved by symmetric and asymmetric encryption. Data is encoded and decoded only by authorized users.
- **Integrity:** Integrity ensures that the data at receiver end is same as that was sent from the transmitter end. Alteration of message is done only by authorized party. In cryptography, hashing is used to create a digest message from the message sent along with transmitted data. At the receiver side, any another digest message is created to match with the original one. By doing this, integrity is ensured in data.
- **Non Repudiation:** This is a method to ensure that the sender has actually sent the data and receiver has received the data. It means that the data has been sent and received by the parties that are claiming to send and receive data. By doing this users can't deny about sending and receiving the data and their authenticity on the data [7].
- **Access Control:** This property ensures that only the authorized users can access the data information. It is the selective restriction to the data access and operations performed on the data [4].

1.6.3 Types of Cryptography: In cryptography along with the transmitted data, a secret key is also sent for encryption and decryption. Depending on the secret key used, cryptography is of two types. Figure 1.9 shows hierarchy for types of cryptography.

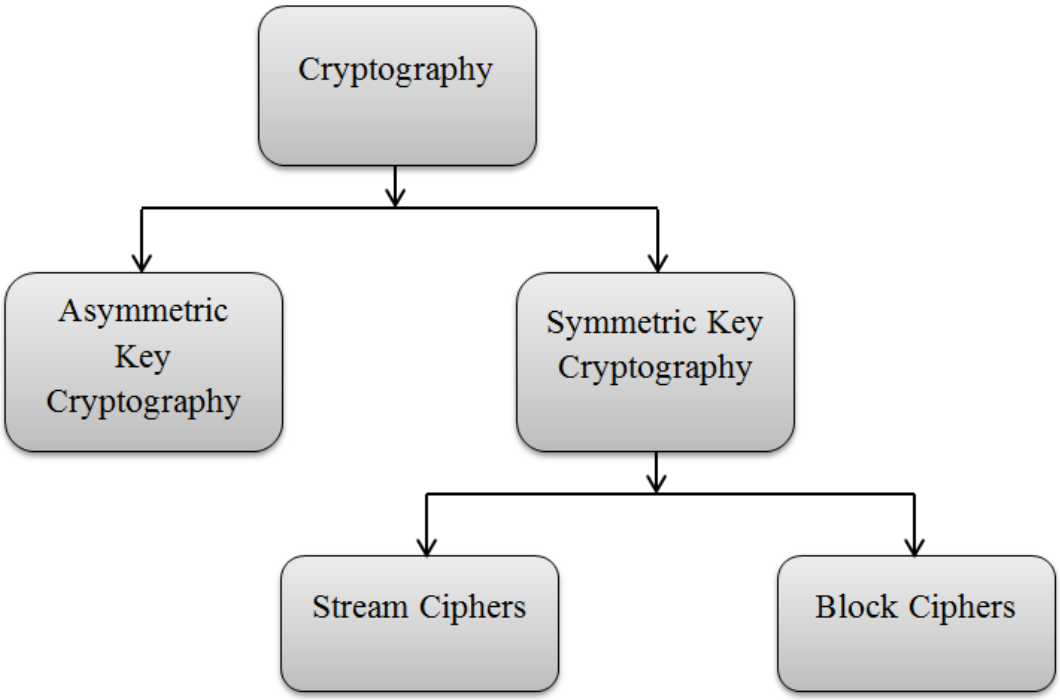

Figure 1.9 Types of Cryptography

1.6.3.1 Symmetric Key Cryptography: This is known as private key cryptography. Symmetric key cryptography uses the same key to encrypt and decrypt the data as described in figure 1.10. Encryption and decryption of data are just reverse of each other. Main thing of concern in symmetric key cryptography is mechanism by which key is shared for encryption and decryption. Key used in cryptography can be random bits or string of bits. Applied key tend to change the content of the original applied data to make it undistinguishable. By knowing the secret key data can be encrypted and decrypted easily [8].

❖ **Advantages of Symmetric Key Cryptography: [9]**

- Relatively fast because operations implemented in algorithms are simple.
- As single key is required for encryption and decryption, hence symmetric cryptography is simple to implement and needs less resources.
- No widespread security compromise because for each pair of sender and receiver different key is used, hence on extracting one key only the data associated with that particular pair is affected.

- Use password authentication schemes to prove identity of a particular user.

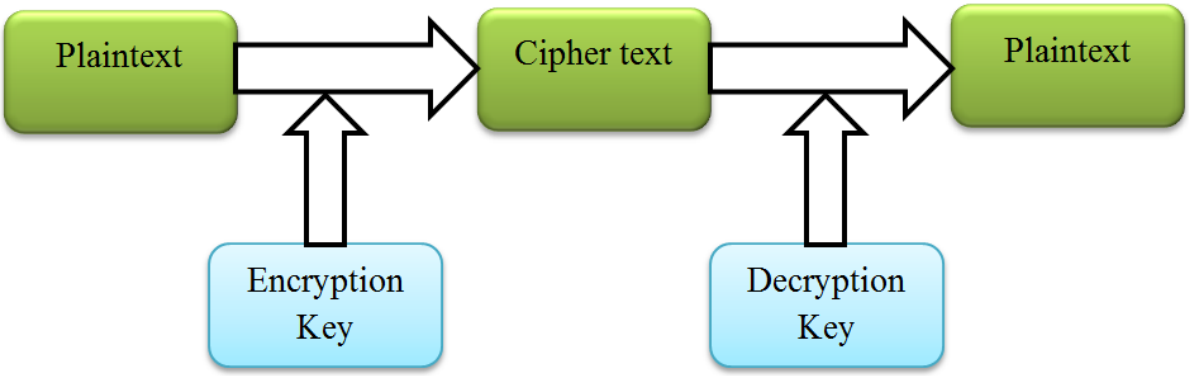

Figure 1.10 Symmetric Key Cryptography

❖ **Disadvantages of Symmetric Key Cryptography: [9]**

- No provisions of non-repudiation as digital signatures are not used in symmetric cryptography.
- Less secure due to shared key for encryption and decryption. If the key is known the data can be easily extracted.
- Key sharing problem. Key has to be transmitted to receiver for decryption before the transmission of original data.
- For the communication between sender and receiver, each time new key is generated. So too many keys are there in symmetric cryptography.

Some symmetric key algorithms are Advanced Encryption Standard (AES), PRESENT, Data Encryption Standard (DES), Blowfish, International Data Encryption Algorithm (IDEA) etc.

Symmetric key cryptography algorithms are further divided into two categories:

➢ **Stream Ciphers:** Stream ciphers one bit at a time. Stream ciphers imply infinite length pseudorandom sequence of bits as key to encrypt data. Cryptographic algorithm is applied to each bit of data with corresponding bit of key stream to produce cipher-text bit. Stream ciphers generally imply two operations: Mixing and XOR operation [10].

❖ **Advantages of Stream Ciphers:**

- Less susceptible to chosen plaintext attacks as each time different bits are produced.
- Best for the cases of unknown data because encryption of one bit does not depend on surrounding bits.
- Easy to analyse mathematically.

❖ **Disadvantages of Stream Ciphers: [11]**

- In case of synchronous stream cipher if one cipher text bit is lost, resynchronization of key generators is required at transmitter and receiver side to produce the same key to encrypt that particular bit.
- Periodicity of the bits in case of short length keys, hence less resistant to some attacks.

➢ **Block Ciphers:** Block ciphers encrypt N bits at one time known as block of data. The data is firstly partitioned into fixed length blocks and each block is encrypted separately. Block cipher are used for processing of bulk data [11].

❖ **Advantages of Block Ciphers:**

- Block ciphers are used for encryption and authentication at the same time.
- Immunity to tempering as the attacker has to modify each random bit.

❖ **Disadvantages of Block Ciphers:**

- More susceptible to chosen plaintext attacks as identical plaintext blocks produce identical cipher text.
- High memory requirement for processing of blocks of data.
- Any fault in one bit may affect the whole block.

As depicted by comparative analysis of block and stream ciphers in Table 1.1, it is observed that block cipher is used for large data and stream ciphers are used for small data.

Table 1.1 Comparison of Block Ciphers and Stream Ciphers

Block Ciphers	Stream Ciphers
Encryption of a particular block depends on the previous blocks.	Encryption of a particular bit is independent of its surrounded bits.
Since blocks of bits are processed at one time, so it is relatively complex and slower.	Processing of one bit at a time is faster and less complex.
Block ciphers are designed from the feistel networks. Hence these ciphers have particular design to be followed.	Stream cipher designs include no repetition for long periods of time. Hence these are random in nature.
Key used to encrypt block of data is same.	Different key is used to encrypt each bit data.
Padding is done in some cases of short length blocks.	Bits are processed continuously as a chain.
Examples: IDEA, AES, DES, SERPENT, PRESENT etc.	Examples: SALSA, GRAIN, RABBIT, FISH, RC4 etc.

Block ciphers follow a certain pattern for the encryption of data. Thus, depending on the pattern followed block ciphers are of two types:

- **Feistel Network:**

Fesitel network is for construction on any symmetric block ciphers. It is a structure where an internal function called round function is iterated repeatedly. Figure 1.11 depictes feistel network for cryptographic algorithms. The type of feistel structure used depends on desired security of system. It is considered that 3 rounds of feistel network are sufficient to generate pseudorandom value and 4 rounds are considered for generation of strong pseudorandom numbers. Encryption and decryption operations are identical except reversing the key generation. So, less circuitry is required which results in less area requirement. Also high the number of rounds, the system will be more secure, but it also leads to slow processing time [11].

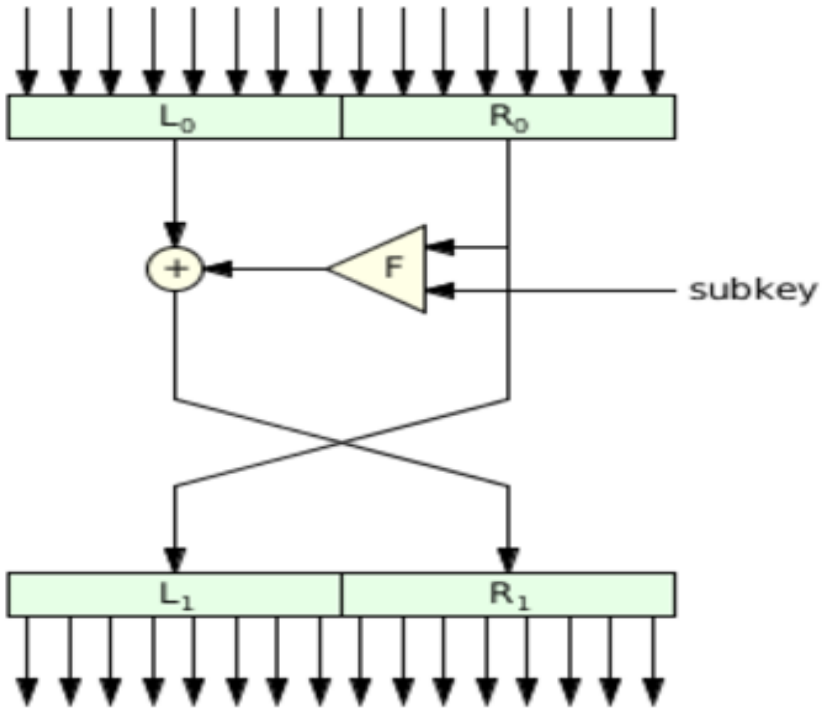

Figure 1.11 Feistel Network [11]

- **Substitution-Permutation Network (SPN)**

SPN is another type of block cipher which makes use of some mathematical operations to generate random values. As presented in figure 1.12, it consists of passing the plaintext and key into various layers of Substitution boxes and Permutation boxes to make cipher text blocks with high degree of confusion. Substitution boxes replace a fix value with another

fix value but the size of both the input and output of S-boxes should be same. Permutation boxes just shuffle the bits. Output of S-Box of one round is taken and shuffled to different positions bitwise. Efficient P-Box distributes data bits to as many S-Boxes as possible. SPN network also perform operations generally XOR and shifting, on applied key and produces round keys. These rounds keys are also operated with the P-Box output by various mathematical operations [11].

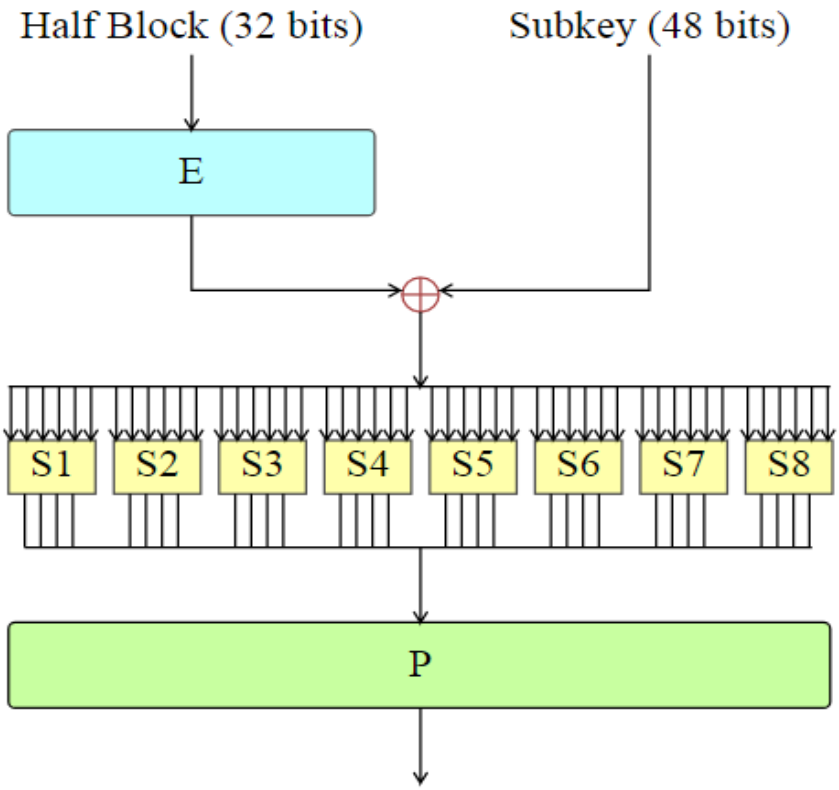

Figure 1.12 Substitution Permutation Network [11]

1.6.3.2 Asymmetric Key Cryptography: This is known as public key cryptography. This technique uses a pair of keys for encryption and decryption. Public key is used for encryption and private key is used for decryption. Keys used for encryption are disseminated publicly but private keys are known only to the receiver [1]. Figure 1.13 gives idea about asymmetric key cryptography.

❖ **Advantages of Asymmetric Key Cryptography: [12]**

- Eliminates the need of key sharing protocols as different keys are used for encryption and decryption.
- Provision of authentication. Allows the use of digital signatures, ECC, RSA which verifies that the data is originally sent from the particular sender.

- Highly secure because to extract the original information attacker has to determine two different keys.

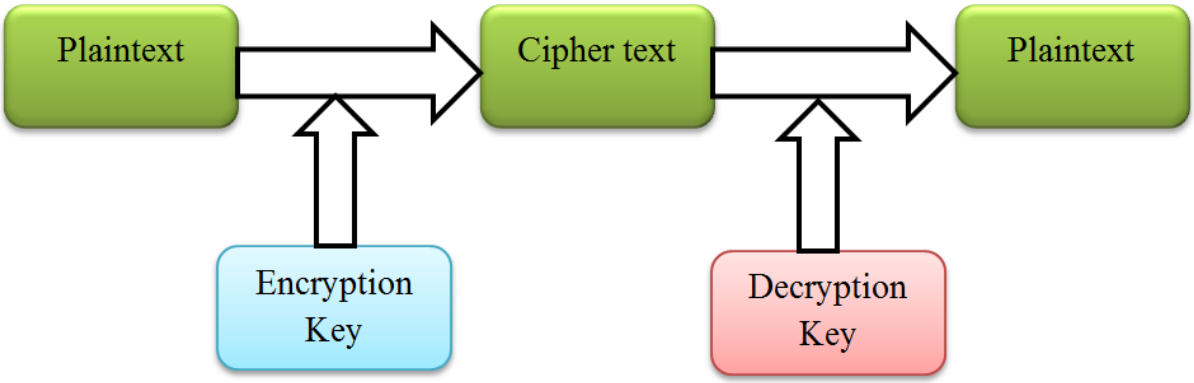

Figure 1.13 Asymmetric Key Cryptography

- As digital signature are able to detect if the data is altered in transit. Hence it provides non repudiation to data.

❖ **Disadvantages of Asymmetric Key Cryptography: [12]**

- Large key size to reduce exhaustive search, brute force attacks on public key.
- More complex as require non-linear circuitry to make public key secure.
- Slower than the symmetric key cryptography due to large key size and more complexity.
- Security compromise as data can be extracted if attacker determine private key.

Some asymmetric key algorithms are Rivest Shamir Algorithm (RSA), Diffie Hellman key exchange algorithm, Digital Signature algorithms, ECC etc.

1.7 THESIS OUTLINE

- **Chapter 2** provides literature survey, in which various authentication techniques and data encryption techniques in the field of smart cards are surveyed.
- **Chapter 3** describes problem formulation in the previous works and proposed methodology to improve previous schemes.
- **Chapter 4** presents detailed description of algorithms used in the proposed technique. In this chapter various steps related to encryption and authentication are discussed.
- **Chapter 5** provides implementation results of various approaches followed and their comparison in terms of area, power, timing summary etc.
- **Chapter 6** concludes the work done and future directions in which further work is possible.

CHAPTER 2

LITERATURE SURVEY

Smart cards have become important part of daily life applications for providing data security and authentication. Various techniques using cryptography algorithms in the field of smart cards authentication and encryption have been proposed. This section provides brief review of smart cards related techniques.

2.1 SURVEY FOR SMART CARDS AUTHENTICATION TECHNIQUES

A survey of various authentication schemes for smart cards is as following:

Atul Singhal, *et al.* [13], proposed authentication scheme for smart cards based on hash functions. This scheme consist of four phases, registration phase for the registration of user with server, login phase for secure login of user with server, authentication phase for the mutual authentication of server and user and password change phase for updating password without assistance of any server. It eliminates the need of verification tables; provide mutual authentication and early detection of wrong passwords.

Himanshu Mittal [14], introduced a new smart cards authentication scheme comprised of hash functions, XOR operations, Diffie-Hellman algorithm. This multi-server scheme resists many of the possible attacks such as man in middle attack, insider attack etc. and establish secure communication between user and server without using any kind of encryption techniques, verification table. This scheme consists of four phases: Server registration phase where each server has to register at authentication server with a unique ID. User registrations phase in which each user has to register at authentication server with their Id and password. Authentication of remote user and server in which any of the users registered with authentication server can access any of the registered servers with a unique login. Mutual authentication and session key generation phase, where the target server and user generate session keys with mutual authentication to communicate with each other. This scheme use less number of hash functions than existing Xie and Chen scheme.

S.Nivetha, *et al.* [15], proposed a new authentication scheme to make smart cards secure against various malicious operations such as hacking of system, creation of duplicates. This is a two level scheme, first level uses PIN for authentication where the server generates an ID and PIN for a particular smart card at time of registration. Then this PIN is obtained by mobile sensing and sent to the server for smart card authentication. The second level uses

OTP for authentication which is generated by using hash function SHA-256. This OTP is sent by the server to the user via SMS when the sensed PIN is matched with PIN created by server. Received OTP is entered in card reader. Entered OTP is sent back to the server for authentication. In this way smart cards are made authenticated at two levels.

Ruhul Amin, *et al.* [16], designed a new ECC based authentication scheme and a secure medical architecture to ensure security of sensitive data in Telecare Medical Information systems. This technique consist of phases system setup phase, user registration phase, login phase and mutual authentication and key agreement phase, password update phase and password recovery phase. By using this scheme a single patient can login to the internet and hence there is secure exchange of information between patient and doctor. The proposed scheme is verified by the simulation through automated validation security schemes and then compared with the existing multi- server internet security techniques. This authentication is comparable to cost of other schemes but it is very efficient in login, authentication, password update and detection.

Chun Ta Li [17], introduced an updated password authentication scheme based on ECC. Password authentication leads to secure data transmission over internet. This proposed scheme to resolve the attacks such as offline password guessing attack, stolen verifier attacks on the scheme proposed by Islam and Biswas. This six phase scheme allows the remote users and servers to communicate through a privacy preserving authentication.

Ravi Singh Pippal, *et al.* [18], presented an image encryption based secure smart card authentication scheme to provide efficiency and security to user. This scheme encrypts the information by using a key image. proposed scheme is resistive to various attacks such as insider attack, password guessing attack, stolen verifier attack, replay attack etc. and also provide some key features such as mutual authentication, session key agreement, change of password without any notification to server.

Norbert Drum, *et al.* [19], introduced a flexible ECC based scheme for resource constrained environment where ECC intense calculations are shifted to more power authentication terminal and physical implementation of such scheme on smart card systems. One way authentication between user and smart card is carried out in 25 ms. Thus provide flexible and resource constrained scheme for smart cards authentication.

Cheng Cai, *et al.* [20], proposed new scheme to prevent information leakage. To verify the server in traditional authentication schemes, smart card transmit its card number which

results in information leakage and threat of being attacked. This mutual authentication scheme is based on verification by both the authentication server and the smart cards. In this scheme, card number being transmitted is sent in the form of random number. Also a card specific key in coded form stored in header file is specified as card access condition. Authentication server has to pass the security test of smart card. After that authentication server verify smart card identity. Only then it can access the contents of card. This symmetric mutual authentication scheme leads to enhanced security.

Hong-Bin Tang, *et al.* [21], introduced a timestamp based authentication scheme for smart cards using ECC. This scheme overcomes the vulnerabilities of previous defined authentication schemes by Yang et al., Shen et al. and Awasthi et al. and improve the efficiency by making the cards resistant against various attacks such as Man-in-middle attack, smart card lost attack, replay, Denial of service, modification attack etc. User has access to change password freely without maintaining any records at the server. This scheme is also optimized from area perspective as it requires 3 ECC multiplications and only 6 hash functions calculations.

2.2 SURVEY FOR SMART CARDS ENCRYPTION TECHNIQUES

Survey of proposed encryption techniques using cryptographic algorithms is gives as:

Johan Borst, *et al.* [22], compared various cryptography primitives to provide security and authentication to smart cards. Symmetric algorithms such as DES, AES, IDEA, RC5 are used for data encryption, Message authentication codes (MAC) are applied as symmetric authentication primitive, ECC, RSA, Hash functions are considered for asymmetric authentication. Various attacks related to smart cards such as timing attacks, fault based analysis, side channel analysis and their countermeasures are discussed. General countermeasure to avoid all system based attacks is to provide randomization using specific hardware.

Karima Dichou, *et al.* [23], described the best FPGA implementation for DES. Smart cards require secure hardware architecture with some constrained area and memory conditions. DES algorithm is implemented on various FPGA devices, in pipelined and non-pipelined architectures. 16 stage pipelined architecture consumes least area slices, hence best from area perspective. As far as throughput is considered, the best one is implementation of single chip DES on Xilinx XC4020E which provides throughput of 26.7 Mbpsec. To achieve both the area and throughput requirements, DES implementation on the Virtex XCV150-6 is

preferred. To reduce area the sub-key generation and key scheduling circuits are removed. Also this provides throughput of 10 Gbps when pipelined.

Carsten Rolfes, *et al.* [24], represented implementation of round based, serialized and parallel architectures of lightweight algorithm PRESENT to use in active and passive smart cards. Parallel architecture has high area to throughput ratio, but at the disadvantage of high area consumption, hence suitable for high end and back end smart devices. Serial architecture is suitable for low end passive smart devices. It consumes least area of 1000GE but high processing time for 563 clock cycles. Round based architecture has high throughput rate and affordable energy consumption per encryption, so preferred for low cost smart devices.

Yadollah Eslami, *et al.* [25], introduced hardware implementation of both public and private key algorithms (AES, DES, ECC) in a single universal processor for use in smart card applications with required power and performance specifications. DES is for providing best past compatibility, ECC has best encryption efficiency among public key algorithms and AES is used for its high throughput and security. The resulting cryptographic processor of three algorithms occupies 2.25mm^2 area in 0.18µm technology which is 9% of the available area of smart cards. Area overhead of the crypto processor is reduced by using FeRAM instead of non-volatile SRAM memory.

Suraj K. Dhanuka, *et al.* [26], implemented ultra-lightweight Hummingbird for smart cards security requirements. Hummingbird algorithm is hybrid of block cipher and stream cipher. By virtue of this paper, authors identified design space, optimization techniques and comparative survey of various hardware implementations with various models of cipher blocks, s boxes for the target algorithm. From area and power perspective, Hummingbird asynchronous design is better than synchronous design. Use of 4 substitution boxes than 1 substitution box leads to less power consumption and processing time

Zhang Peng, *et al.* [27], compared two algorithms RSA and ECC for data encryption in smart cards. ECC use the small number of keys than RSA hence it is most widely used than RSA. ECC leads to the security parameters of the smart cards at less hardware resource use and less bandwidth. The only advantage provided by the RSA is the simple to implement but at the same time it compromise with the security. Hence ECC is best chosen out of the two algorithms for smart cards encryption.

E. Surya, *et al.* [28], compared the various symmetric and asymmetric key encryption algorithms for use in smart cards applications. Algorithms such as AES, Triple DES, Blowfish have less memory and area requirement and also more compatible in security aspects than RSA, ECC. Out of symmetric key algorithms Blowfish, AES, Triple DES, the Blowfish algorithms has key size varying from 32-448 bits, so brute force attack and exhaustive search attack on key can easily be countered. Also Blowfish algorithm encrypts data 16 times, hence it is best secure from any hacker's attack than AES and Triple DES.

Baris Ege, *et al.* [29], provides two lightweight algorithms AES and PRESENT in counter and XOR-Encrypt-XOR (XEX) modes for use in non-volatile memory smart cards. To increase the lifetime of the CTR mode, both algorithms are implemented with a special write counter mode and address scrambling scheme based on block cipher modes. For the CTR mode, AES is implemented in 10K gates along with power consumption of 600μW/MHz but at the same time PRESENT algorithm can be implemented in less than 6K gates and also it does not add up any additional latency. Also in case of address scrambling scheme with XEX mode, PRESENT provides processing speed of 48.7MHz and its security for non-volatile memory smart cards can be increased by replacing S-Box with secret proprietary S Boxes.

Qihui Jhang, *et al.* [30], represented the asynchronous AES implementation comprising of AES algorithm features and properties of asynchronous circuits to protect smart cards from side channel attacks in financial and security applications. Low cost and high throughput asynchronous AES architecture is proposed and HDL techniques are applied on it to reduce area and power. The proposed crypto processor is implemented in VLSI design using 130nm CMOS technology. The resulting design comprises only 7.3% of the area consumed by Standard AES. Also the latency is 32.3% and throughput is 60% high than other AES implementation.

Maryam Savari, *et al.*[31], provide a detailed survey of various encryption algorithms such as ECC, DES, Triple DES, RSA to ensure fulfilment of security requirements in smart card devices. Out of asymmetric key algorithms ECC and RSA, ECC is considered as more efficient for smart cards. ECC consist of difficult mathematical computations, hence resist to brute force attacks. Also ECC provides equal security level to that of RSA with smaller key length, less power consumption. In case of symmetric key cryptography, AES algorithm is preferred where the security is prime requirement than area, speed, power. On the other hand

DES and Triple DES is chosen where area, power, memory constraints are priority then security.

Yoso Adi Setyoko, *et al.* [32], represents a new multi-purpose smart card design to make secure financial transactions and for identity based applications. This design implementation provides a new technique to store smart cards data in compact size and also how to manage the key for each user identity. Data is stored in the encrypted form and completed with SHA-1 signature. The existing algorithms as DES, RSA require high computation effort, so the Authentication Encryption (AE) is implemented in the proposed design architecture which has a fast verification process than the previous DES-RSA verification.

Maryam Savari, *et al.*[33], proposed a new way of combining various smart cards in a single multipurpose card and implementing a common cryptography algorithm in it to make data secure. Smart card is used in many applications as health care ID card, banking debit card, government identity card, payphone card etc. where the security is prime concern. The encryption method for the smart card must be strong enough to prevent any third party access to card. This paper represents combinations of various algorithms as AES, DES, and ECC for different applications. For use in health services, large data has to be processed, so combination algorithm of ECC and DES is used. The security is the prime concern in banking card applications. Hence combinational algorithm of AES and ECC is preferred.

2.3 Comparative Analysis of Various Existing Encryption and Authentication Techniques for Smart Cards Security

In the survey presented above, authors proposed various schemes for data authentication and security in smart cards. Though individually each scheme is effective in providing security requirements, yet it has some advantages and limitations as compared to other schemes. The studied authentication and encryption techniques are analysed in brief in Table 2.1.

Table 2.1 Comparison of Various Existing Authentication and Encryption Techniques for Smart Cards Security

Authors	Proposed Work	Advantages	Limitations
Atul Singhal, *et al*. [13] S. Nivetha, *et al*. [15]	Authors proposed smart cards authentication technique based on one way hash functions. [13] A two level authentication scheme is designed by using hash function SHA-256. [15]	Proposed scheme eliminates use of verification tables; hence prevents stolen verifier attacks. [13]	The scheme using SHA-256 will have large block size of 256 bits; hence this scheme is not so efficient due to area and memory constraints in smart cards. The scheme is susceptible to man-in-the-middle attack as OTP sent to mobile at second level has no secure communication channel. [15]
Himanshu Mittal [14]	Diffie Hellman based multi-server authentication scheme is introduced for secure smart cards.	Scheme is highly secure because in Diffie Hellman algorithm the g must be primitive root of p, and it is very difficult to calculate such root. So, proposed scheme is less susceptible to man-in-the-middle, impersonation and replay attacks.	Computationally high area consuming. Each multiplication varies with the square of n which must be large. Susceptible to man-in-middle attack as both the parties involved are not authenticated to each other.
Ruhul Amin, *et al*. [16] Norbert Drum, *et al*.[19]	A medical system architecture based on ECC is designed for secure login of single patient to medical database. [16]	Mathematical modelling of ECC consists of ECC curves generation, finding suitable points on the curve satisfying a particular equation and	Although mathematical modelling of ECC makes it secure against various attacks, but these computations result in

	An ECC based cryptography processor is introduced for authentication purposes in resource constrained devices. [19]	then secret key generation. This modelling is difficult to implement which makes it secure.	high power consumption.
Carsten Rolfes, *et al*. [24]	Three architectures of lightweight algorithm PRESENT are implemented for smart cards devices security in 1000GE.	Pipelined architecture of PRESENT can be used at back end devices as this produce very large current. The application where area is prime concern serial architecture is used as it is implemented in less than 1000GE. If high processing speed is required, round based architecture is preferred whose processing speed is 32 clock cycles.	----
E. Surya, *et al*. [28]	Authors compare symmetric key encryption algorithms DES, Triple DES, AES and Blowfish for smart cards data encryption.	AES is considered best for providing security in both hardware and software implementations because of non-linear shift rows and mix columns operations. For Blowfish algorithm, the key size is variable ranging from 32-448 bits, so it is impossible for attacker to extract the original keys. Hence Blowfish is secure against exhaustive search and brute force attack.	Due to large key size in AES the time taken for encryption and decryption adds up a constraint to speed of smart cards devices.
Baris Ege, *et al*. [29]	For smart cards non-volatile memory, AES and PRESENT are implemented in Counter (CTR) and XOR-Encrypt-XOR (XEX) mode.	Memory encryption for smart cards is best suited by using PRESENT algorithm as it is implemented in less than 6K gates without any additional latency.	----

CHAPTER 3

GAPS IN STUDY AND OBJECTIVE

3.1 PROBLEM FORMULATION

From literature survey it is observed that for providing security to highly classified data, AES and DES are two preferred symmetric key algorithms [31]. Authors **Karima Dichou, *et al.*** [23] worked on DES and reported that hardware implementation of DES requires 4245 LUT to ensure efficient security. According to authors **Baris Ege, *et al.*** [29] and **Qihui Jhang, *et al.*** [30] second commonly used algorithm for smart cards encryption is AES but AES also consumes 4401 LUT for hardware [34]. Both these algorithms require huge area and memory for providing security. Hence these algorithms are not considered so efficient for smart cards.

Along with security, authentication is equally important in smart cards. The asymmetric cipher RSA and ECC preferred for smart card authentication [27]. **Ruhul Amin, *et al.*** [16] and **Norbert Drum, *et al.*** [19] proposed various authentication schemes based on ECC algorithm. ECC is dependent on mathematical calculations which make it difficult to extract the information, hence making system more secure but at the same time these mathematical calculations require large computational memory. Hardware implementation of ECC consumes 36727 LUT [35] which is very high making it inefficient for use in smart cards applications. RSA is also used for authentication in smart cards [27]. Though it is simpler than ECC and requires less memory than ECC [31] but it also have some mathematical calculations which makes it more area consuming of approximately 19213 LUT [36]. Hence both the RSA and ECC require large area and memory for data authentication.

3.2 OBJECTIVE

Gaps related to area and memory constraints make the above mentioned algorithms inefficient for smart cards security purposes. So, the main research direction is to design a technique within the limit of area and memory constraints to fulfil security requirements for smart cards. The objectives of defined work are:

- To analyse the need of security and authentication in smart cards applications.
- Hardware implementation of PRESENT algorithm and its optimization.
- To design a system consisting of hybrid encryption and authentication algorithm for providing both security and authentication without any additional area overhead.

- Performance analysis of optimized algorithm with existing on the basis of LUT, Flip flops, throughput, efficiency.

3.3 PROPOSED METHODOLOGY

The main focus of the work is to design a system for provision of both the security and authentication to smart cards applications.

To resolve area issue lightweight cryptography algorithm is implemented for data encryption. According to the NIST report for data security, PRESENT is considered as highly secure algorithm and it is also accepted as standard for lightweight cryptography algorithms under ISO/IEC 29192-2. [37] So for encryption in smart cards PRESENT can be used in place of AES and DES.

In case of authentication, a new scheme redundant bit security (RBS) can be implemented. This scheme results in original data hiding with redundant bits followed by certain pattern. Redundant bits are also generated by lightweight block cipher ANU. RBS technique provides authentication at sender and receiver side as well as security because it is difficult to find out whether a particular bit is redundant bit or transmitted data bit. Hence this technique is resistant to various attacks.

To achieve both security and authentication, PRESENT algorithm is integrated with the redundant bit security scheme. PRESENT provides optimized results in data encryption and this result is combined with redundant bits based on particular pattern to ensure authentication.

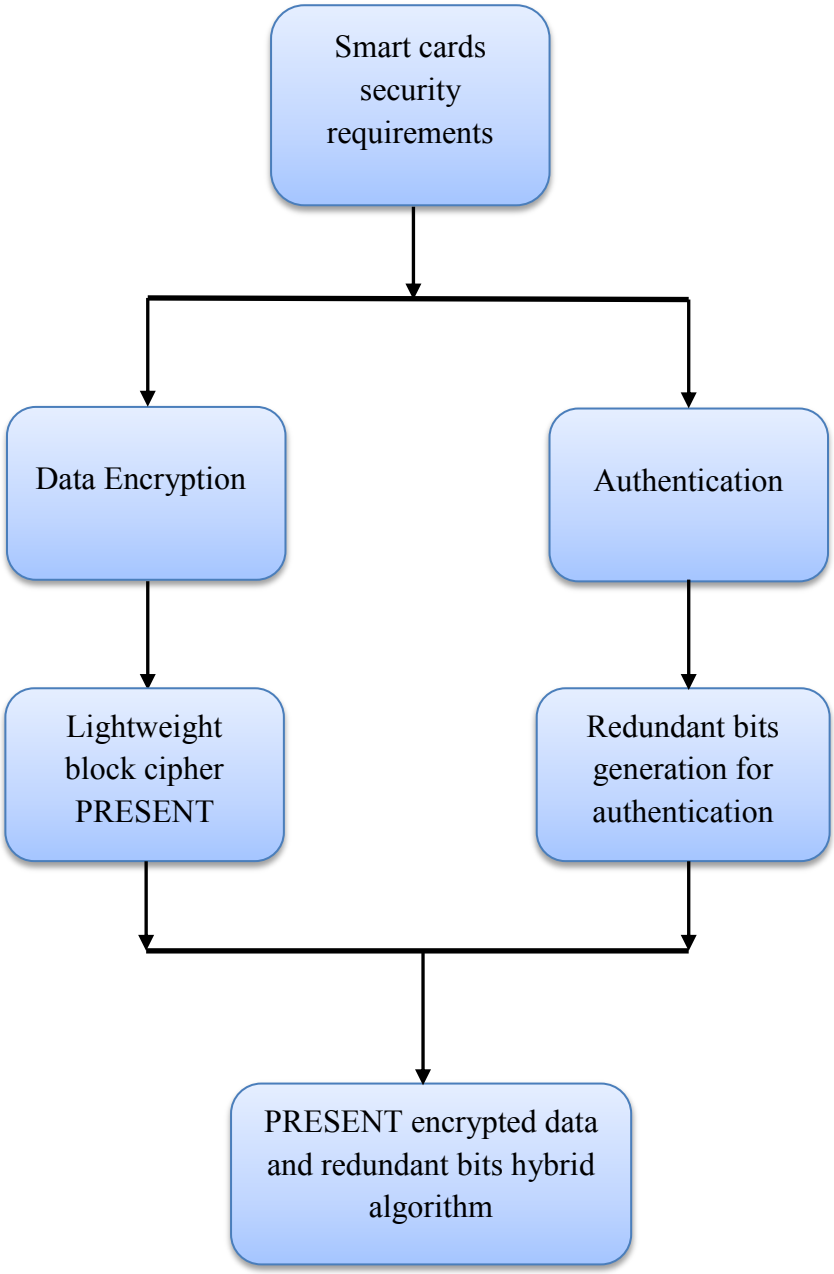

Figure 3.1 Brief Overview of Proposed Methodology

CHAPTER 4

OVERVIEW OF ALGORITHMS

4.1 OVERVIEW OF PRESENT BLOCK CIPHER

4.1.1 PRESENT Encryption

PRESENT is symmetric block cipher proposed by Bagdonav et al. [38] in 2007. PRESENT is substitution permutation network which supports block size of 64 bits and key size of 80 bits or 128 bits. It consists of 31 rounds and a last 32 round only for the key mixings step. One round of PRESENT consists of given operations:

- Add round key
- S Box layer
- P layer
- Key scheduling

PRESENT has optimized architecture; hence requires less area and memory as compared to AES, DES etc. making it suitable for resource constrained environments such as RFID, smart cards, wireless sensors, IOT etc. [39].

Depending on the design techniques, PRESENT algorithm has three architectures:

- Serial architecture
- Parallel architecture
- Round based architecture

Serial architecture consumes require less than 1000GE for hardware implementation but at the same time it requires high processing time of approximately 563 clock cycles. So, it is suitable for low end passive devices. Parallel architecture has high throughput to area ratio but it consumes high area making it suitable for back end devices. Round based architecture has high throughput rate and affordable energy consumption per encryption, so preferred for low cost smart devices [24].

4.1.1.1 Algorithmic Description of Round Based Architecture (PRESENT):

At algorithmic level, PRESENT is summed up as shown in Figure 4.1:

```
Generate round keys ( )
for n= 1 to 31 do
add round key (state, Kn)
S Box layer (state)
P layer (state)
end for
add round key (state, K32 )
```

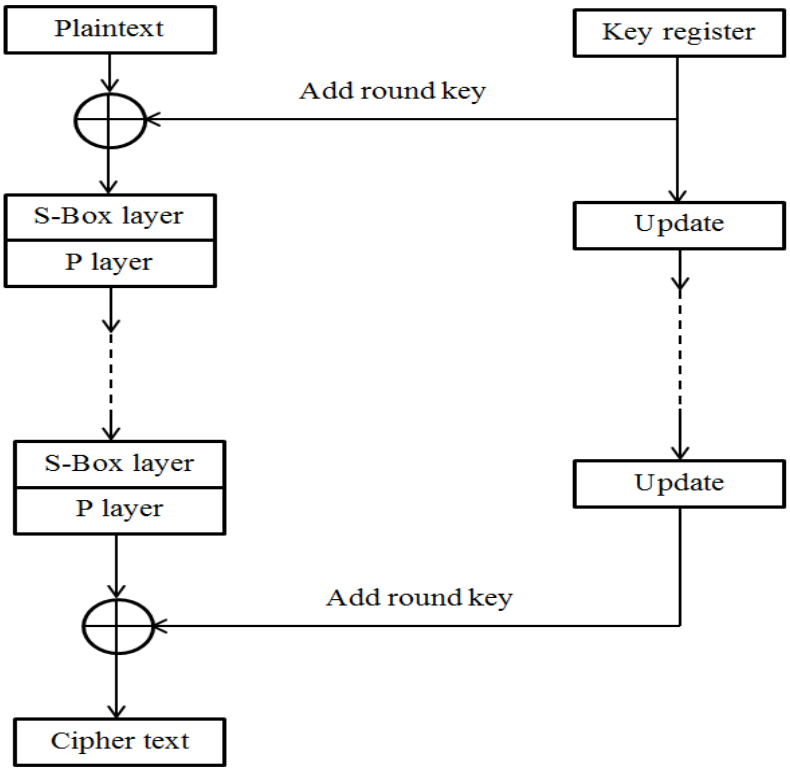

Figure 4.1 Algorithmic Approach of PRESENT [38]

- **Add Round Key:** Each round in PRESENT includes XOR operation to generate round key K_n for n=1 to 32, where K_{32} is used to XOR the last data.

Let the applied plaintext is $b_{63}.....b_0$, round key $K_n = k^n_{63}\ldots\ldots k^n_0$ for n= 1 to 32. Add round key operation consists of

$$b_j = b_j \text{ XOR } k^n_j \qquad \text{(for j= 0 to 63)}$$

- **S Box Layer:** PRESENT uses single 4 bit non-linear substitution box which is applied 16 times in parallel for each round.

For this operation the add round key output $b_{63}\ldots.b_0$ is taken as sixteen 4 bit words $s_{15}\ldots.s_0$ which is provided as

$$s_m = s_{(4*m)+3} \,||\, s_{(4*m)+2} \,||\, s_{(4*m)+1} \,||\, s_{(4*m)} \qquad \text{(for m= 0 to 15)}$$

Each s_m is replaced by another fix value provided by the following table

Table 4.1 PRESENT S Box Layer

s_m	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[s_m]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

- **P Layer:** P layer provides linear bit wise permutation to create confusion of data. In this layer bit n of S Box output is moved to bit position P(n)

Table 4.2 PRESENT P Layer

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P(n)	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
n	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P(n)	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
n	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
P(n)	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
n	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
P(n)	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

- **Key Scheduling:** PRESENT supports 80 bits and 128 bits keys which are stored in key register K. At each round left most 64 bits of key register contents are considered as round key.

➢ **80 Bits Key:** Key register is updated as follows:

1) $[k_{79}\ k_{78}\ k_{77}\ldots\ldots k_2\ k_1\ k_0] = [k_{18}\ k_{17}\ k_{16}\ldots\ldots k_{21}\ k_{20}\ k_{19}]$ (61 bit left rotate)

2) $[k_{79}\ k_{78}\ k_{77}\ k_{76}]$ = S Box$[k_{79}\ k_{78}\ k_{77}\ k_{76}]$ (S Box layer)

3) $[k_{19}\ k_{18}\ k_{17}\ k_{16}\ k_{15}]$ = $[k_{19}\ k_{18}\ k_{17}\ k_{16}\ k_{15}]$ XOR round counter

Round counter value is incremented at each round.

- **128 Bits Key:** Key register is updated as follows: [40]

1) $[k_{127}\ k_{126}\ k_{125}.....k_2\ k_1\ k_0]$ = $[k_{66}\ k_{65}\ k_{64}.....k_{69}\ k_{68}\ k_{67}]$

2) $[k_{127}\ k_{126}\ k_{125}\ k_{124}]$ = S Box $[k_{127}\ k_{126}\ k_{125}\ k_{124}]$

3) $[k_{123}\ k_{122}\ k_{121}\ k_{120}]$ = S Box $[k_{123}\ k_{122}\ k_{121}\ k_{120}]$

4) $[k_{66}\ k_{65}\ k_{64}\ k_{63}\ k_{62}]$ = $[k_{66}\ k_{65}\ k_{64}\ k_{63}\ k_{62}]$ XOR round counter

Data path of round based architecture of PRESENT is depicted in Figure 4.2. Round based architecture consumes 1 clock cycle for each round. Round based architecture uses one S-layer and P layer which leads to data path of one 64 bit XOR, one P-layer, and 16 parallel S-boxes. Key scheduling consists of 64 bit state register to store internal state and 80 bit key register to store updated key. Key scheduling components includes 80 bit key register, 5 bit XOR, one S-box, 61 bit shifter and key is generated on the fly at each round. Loop based approach is preferred for area saving.

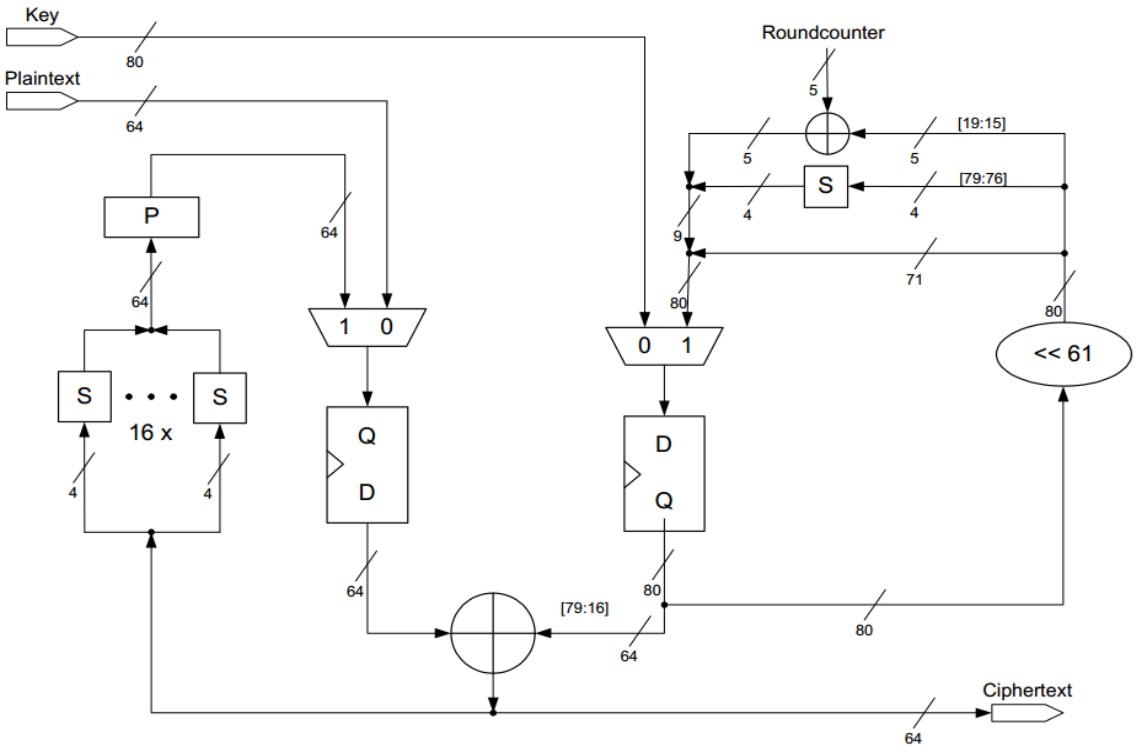

Figure 4.2 Data Path of Round Based Architecture [24]

To reduce power consumption clock gating is provided at control signals which leads to provision of best results in terms of area, power and energy consumption.

4.1.1.2 Various Approaches for PRESENT S-Box

Substitution boxes in PRESENT add to non-linearity in the data. S-Box follows two approaches:

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

Table 4.3 PRESENT S-Box Look Up Table Based Approach

S_m	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[S_m]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

- **Boolean S-Box:** [41] This new approach implements S-Box in the form of Boolean function. Each bit of the data is represented by a different Boolean function. Whole S-Box is derived from 4 SOP expressions. In case of hardware implementation of this approach includes 45 AND gates and 17 OR gates.

Consider the input and output data

Din: (a) & (b) & (c) & (d)

Dout: (W) & (X) & (Y) & (Z)

Table 4.4 PRESENT S-Box Boolean Logic Based Approach

Output Bit	Boolean Expression
W	(a'c'd') + (a'c d) + (a b'd) + (a b'c) + (a'b c)
X	(b'c d') + (a b c') + (b'c'd) + (a'b'c') + (a'b c d)
Y	(a b d) + (a b'c') + (a'c d') + (a'b'c) + (a b'd')
Z	(a'c d) + (a c d') + (a b'd') + (a'b'd) + (a'b c'd') + (a b c'd)

Number of AND gates and OR gates can be factorization is applied on above Boolean expression.

- **Factorization Approach for S-Box Boolean Logic**

Table 4.5 Factors for PRESENT S-Box Expressions

Factors	Expressions
f1	a'c'd
f2	ab'
f3	a'c
f4	cd'
f5	b'c'
f6	ab
f7	df3
f8	c'f6
f9	d'f2

Table 4.6 Reduced PRESENT S-Box Logic with Factorization

Output bits	Boolean expression
W	(f1) + (f7) + (df2) + (cf2) + (bf3)
X	(b'f4) + (f8) + (df5) + (a'f5) + (bf7)
Y	(df6) + (c'f2) + (d'f3) + (b'f3) + (f9)
Z	(f7) + (af4) + (f9) + (a'b'd) + (bf1) + (df8)

4.1.2 PRESENT Decryption

PRESENT decryption algorithm is reverse of the PRESENT encryption [42]. The modelling of decryption S-Box and P- layer is not defined yet but is also considered as inverse of decryption S-Box and P- layer.

4.1.2.1 Key Design Approaches for PRESENT Decryption

PRESENT decryption is designed by applying two approaches depending on applied input keys:

- **Using Last Key as Input Key Stream:** In this approach, last key is obtained from PRESENT encryption and this is applied as input key stream to the decryption module.

Then, reverse of key scheduling and PRESENT encryption is implemented which obtain original plaintext data from the applied cipher text and key stream. This approach has a limitation that in decryption, for each time encryption module is required to obtain last key. This leads to extra clock cycles and delay.

- **Using Initial Key as Input:** In this case, initial key is applied as input key stream. The key is processed before data and all the required keys are generated. These key values are stored in an array and each key is taken from the array whenever it is required for processing of data. This eliminates the need of encryption each time but it leads to area and memory overhead due to extra resources requirement by key array.

4.1.3 Hybrid Algorithm for PRESENT Encryption and Decryption:

Further a mux based hybrid algorithm for PRESENT encryption and decryption is designed. This algorithm uses an array based approach for the key scheduling and key required is taken from the array. A mux based control pin is added to the design which decides the occurring of encryption and decryption processes.

4.2 SMART CARDS AUTHENTICATION TECHNIQUE

In smart cards for secure login to system, for secure data transmission through contact less cards, data authentication is required. RBS is one main technique used for authentication. Redundant bit security is a method of adding redundant bits to the original data bits depending on secret key [43]. RBS consists of two phases:

- Generation of random redundant bits
- Integration of transmitted data with redundant bits.

Redundant bits are generated by using linear feedback shift registers (LFSR) or Pseudo random number generators (PRNG). Then, these random values are used to hide the original data. In this way by using RBS technique, encryption and authentication of data is achieved. RBS is considered as highly secure technique as it is resistant against brute force, known plaintext, chosen plaintext attacks. But it has condition that for hiding of data depending on key, the key value must have certain probability of 0 and 1. So some additional operations are required to maintain this probability which leads to high computational memory.

To reduce this flaw, in implemented technique, RBS is used for hiding data but not dependent on secret key value. Integration of redundant bits into original transmitted data depends on a particular pattern without any interference of key. This reduces memory consumption to some extent.

4.2.1 Redundant Bits Generation for smart cards authentication

Redundant bits applied for data hiding must be random to ensure security. In proposed technique redundant bits are generated by using lightweight cryptography block cipher, ANU.

ANU is feistel cipher consisting of 25 rounds. It supports block length of 64 bits and key of 128 bits. The 64 bit plaintext is split into two parts of 32 bits as shown in Figure 4.3. Round function is applied to these 32 bits parts. Round function includes circular shift and substitution operation. Upper 32 bits of plaintext are circular rotated left by 3, pass through substitution box and then XOR with lower 32 bits and the result v1 is placed in lower 32 bits. In the next step, again upper bits are circular rotated right by 8, pass through substitution box and XOR with round key and v1. This result is also placed in lower 32 bits position. Both the lower 32 bits and upper 32 bits are passed through P-layer separately. At the end the results of 32 bits are swapped [44].

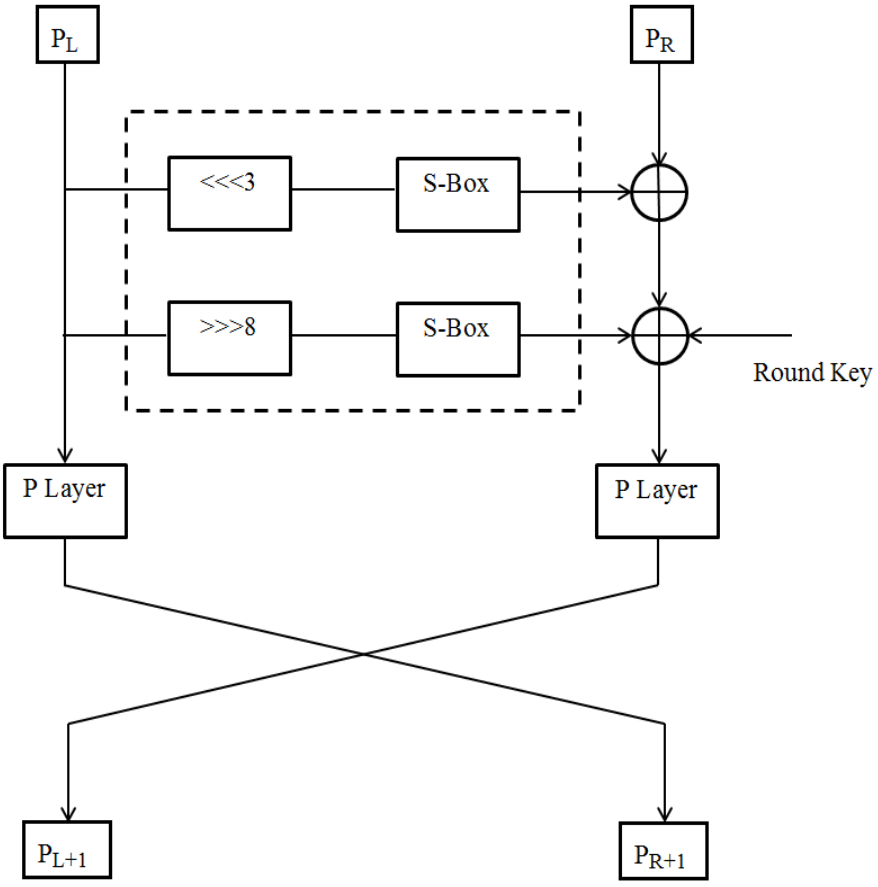

Figure 4.3 Lightweight Block Cipher ANU

CHAPTER 5

IMPLEMENTATION RESULTS

This chapter presents the implementation results for various approaches of PRESENT algorithm and Redundant bit generation algorithm. Synthesis is done in XILINX VIVADO Design Suite 2016 using VHDL language. Synthesizable code is generated for Digilent XC7A35T-1CPG236C BASYS-3 Board of Artix-7 family. Provided board has 5200 slices (each slice with 6 input LUT and 8 flip flops), ,internal clock of exceeding 400MHz, on chip analog to digital converter (XADC).16 input switches, 16 output LEDs, W5 clock pin, 5 user push buttons, three PMOD ports, 12 bit VGA output, USB-UART bridge.

5.1 PERFORMANCE PARAMETERS FOR CRYPTOGRAPHY ALGORITHMS

The efficiency of a cryptographic algorithm is described in terms of its security parameters. Security performance analysis parameters of cryptographic algorithms are:

- **Throughput:** Throughput is the rate of generation of new outputs such as cipher text and authentication tags etc. Throughput is measured by the formula:

Throughput= (Block size* Frequency) / Clock Cycles

In case of lightweight cryptography implementations, high throughput is not a prime requirement but some applications demands moderate throughput value [45].

- **Efficiency:** Efficiency is defined ratio of throughput and area at a fixed clock frequency. It is basically the measurement of area consumed and time taken to process a single output bit [45].

Efficiency= Throughput / area

- **Avalanche Effect:** Avalanche effect describes the change in the cipher text with a small variation either in plaintext bits or key bits. For encryption algorithms to be secured it is desirable that even a single bit change in the plaintext or key stream bits should vary as many bits in the output data [46].
- **Correlation Factor Between Input and Output Data:** Correlation factor is the measurement of relation/dependence between input and output values. This describes the degree to which output data is associated with the input data. In cryptography algorithms less correlation factor is desirable [46].
- **Brute Force Attack:** In brute force attack, the attacker all possible combinations of the plaintext to obtain the resulted cipher text. It is a type of hit and trial attack by which the intruder can extract the information such as personal identification information, login

passwords, authentication data etc. These attacks are also known as exhaustive search attacks [47].

- **Linear Cryptanalysis:** In linear cryptanalysis, the intruder tries to extract information by using high probability linear occurrences of plaintext and cipher text. Linear attacks derive the equations from the relation of input and output and then use these equations to obtain the remaining information [48].
- **Differential Cryptanalysis:** Differential cryptanalysis describes the effect of differences in input at the resultant differences on output [48].

5.2 PRESENT BLOCK CIPHER

5.2.1 PRESENT Encryption Simulation and Synthesis Results

PRESENT encryption algorithm is implemented with block size of 64 bits and key size 80 of bits. Further PRESENT S-box is implemented using two approaches of look up table and Boolean logic.

Plaintext: 4c 0a 52 3b 19 47 8e 60

Key stream: 25 09 1d 47 38 5e 9f 3c 2b 0a

Cipher text: 0c bc a5 3f 3b 66 b7 98

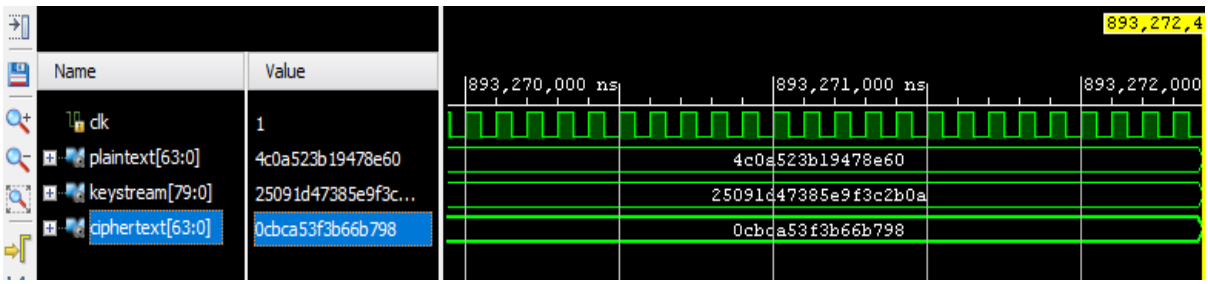

Figure 5.1 Simulation of PRESENT Encryption

Synthesis results describe hardware performance reports in terms of area, power, throughput, clock cycles etc. Table 5.1 compares results of our implemented PRESENT by both S-Box approaches with the existing PRESENT algorithm.

Table 5.1 Comparison of Various S-Box Approaches of PRESENT with Existing Results

Parameters	Existing Results		Proposed Techniques Results	
	Boolean logic based S-Box [41,42]	Look up table based S-Box [42]	Boolean logic based S-Box	Look up table based S-Box
LUT	300	350	281	233
Flip Flops	152	154	143	150
Clock cycles	32	32	32	32
Frequency (MHz)	258	240	258	240
Throughput (Mbps)	516	480	516	480
Efficiency (Mbps /LUT)	1.72	1.371	1.83	2.06

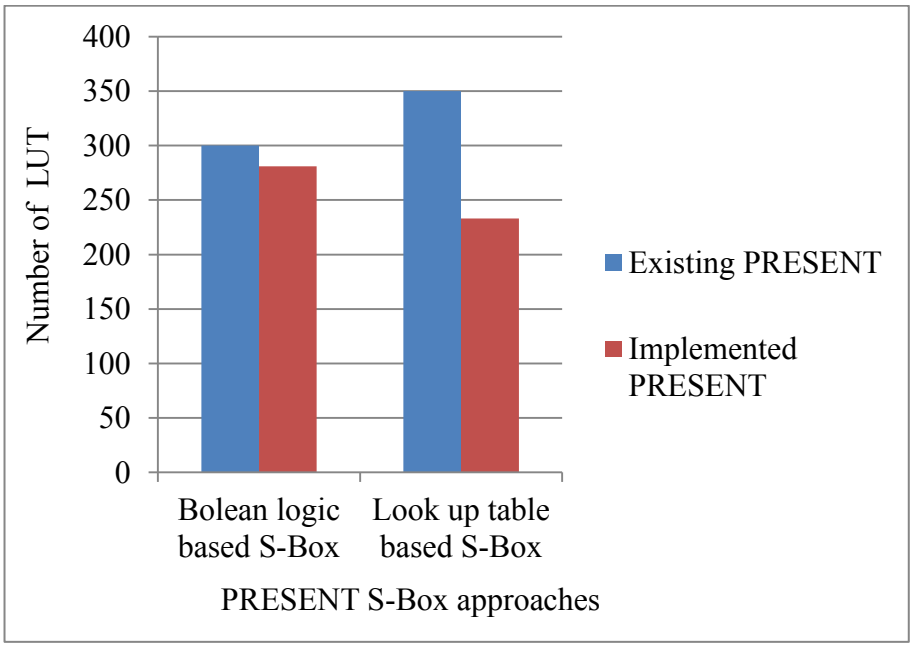

Figure 5.2 Number of LUT for Various S-Box Approaches of Existing PRESENT and Implemented PRESENT

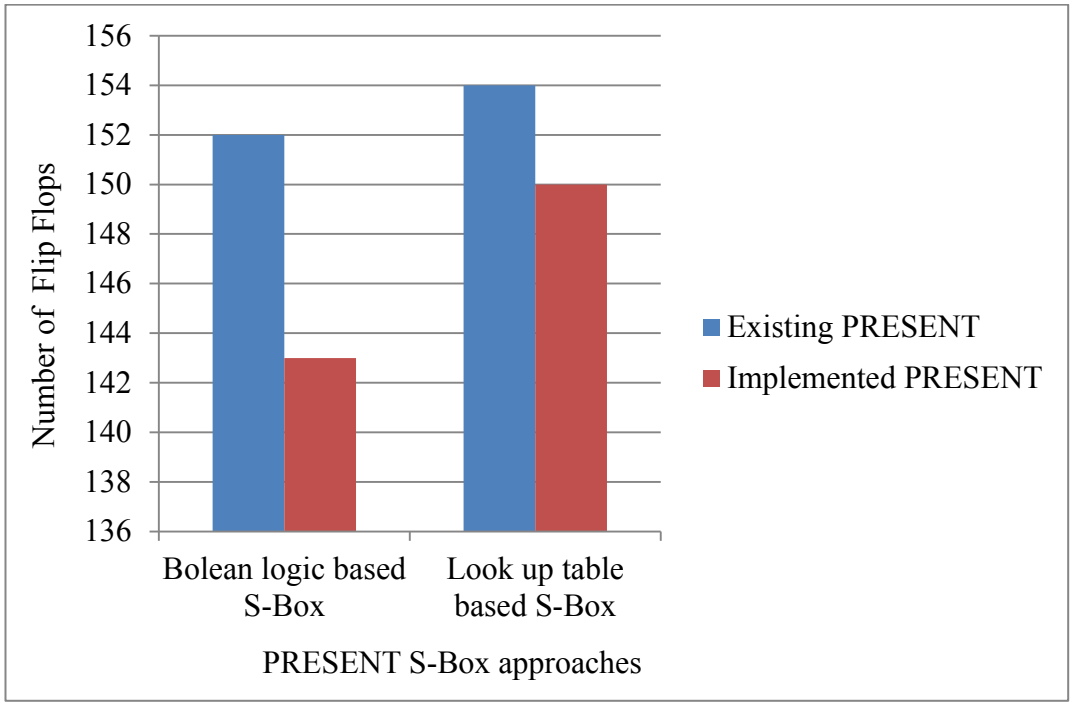

Figure 5.3 Number of Flip Flops for Various S-Box Approaches of Existing PRESENT and Implemented PRESENT

From the Figures 5.2 and 5.3, it is clear that implemented PRESENT algorithm consumes less number of LUT and flip flops than the existing one. The reason of this less area consumption is:

- In case of Boolean logic based S-Box, S-box can be further reduced by using factorization. This factorization actually reduces the number of AND, OR and NOT gates used in logic which further results in less area consumption.
- In case of look up table, LUT consumed are less than the existing PRESENT. In existing results of PRESENT an array is used for S-Box operation which requires more memory elements for data processing. On the other hand in implemented PRESENT, look up table is used for S-Box which consumes less memory than array. So, number of LUT and flip flops is less in implemented PRESENT.

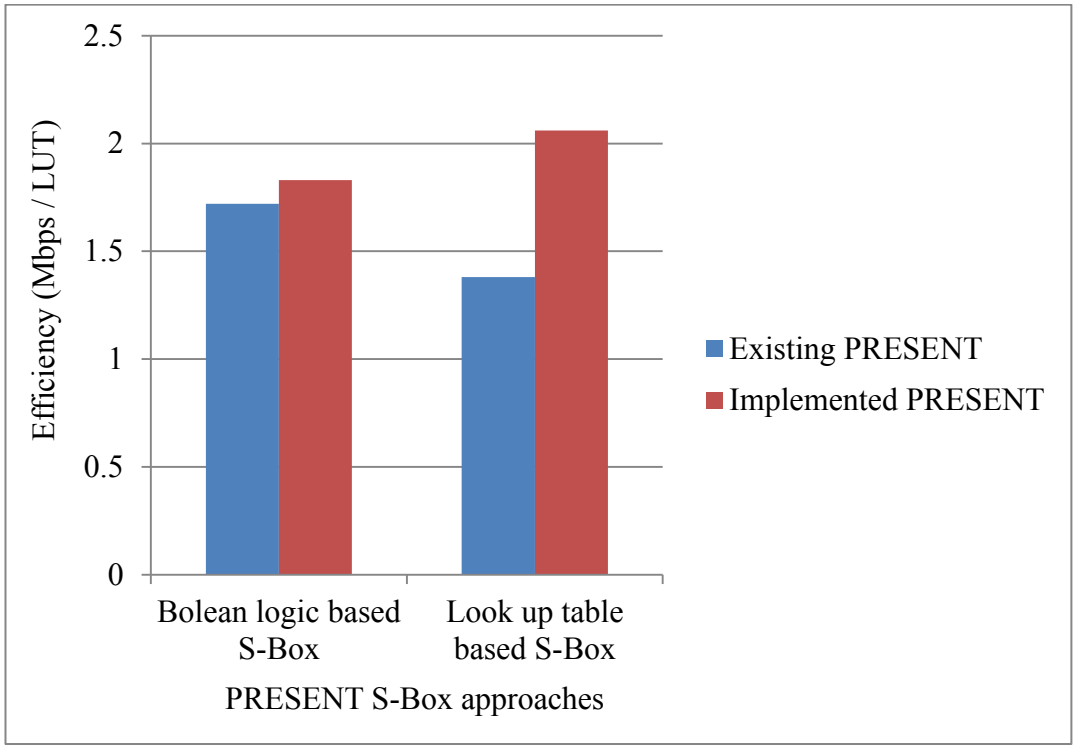

Figure 5.4 Efficiency for Various S-Box Approaches of Existing PRESENT and Implemented PRESENT

Figure 5.4 presents the efficiency for implemented PRESENT and existing PRESENT. The implemented design is more efficient as it is capable of providing same security with less area requirement.

5.2.2 PRESENT Decryption Simulation and Synthesis Results

Decryption module of PRESENT is designed by using two approaches:

- By Using Last Key as Input Key Stream

Cipher text: 0c bc a5 3f 3b 66 b7 98

Key stream: 48 75 45 a1 1a b0 1a 74 f4 70

Plaintext dec: 4c 0a 52 3b 19 47 8e 60

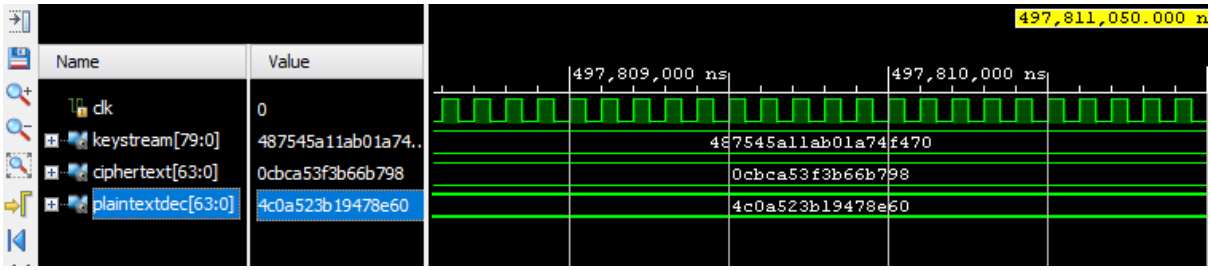

Figure 5.5 Simulation Results of PRESENT Decryption Using Last Key as Input

- Using Initial Key as Input Key Stream

Cipher text: 0c bc a5 3f 3b 66 b7 98

Key stream: 25 09 1d 47 38 5e 9f 3c 2b 0a

Plaintext dec: 4c 0a 52 3b 19 47 8e 60

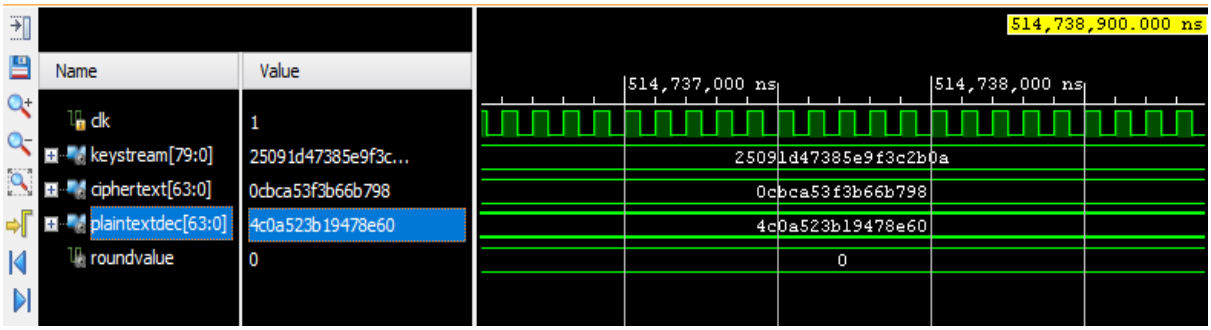

Figure 5.6 Simulation Results of PRESENT Decryption Using Initial Key as Input

In synthesis results of decryption comparison of both approaches in terms of LUT, flip flops, power consumption, and efficiency are presented in Table 5.2

Table 5.2 Comparison of Key Generation Approaches of PRESENT Decryption

Parameters	Decryption Using Last Key	Decryption Using Initial Key
LUT	215	3033
Flip Flops	149	2089
Power consumption (W)	0.063	0.093
Frequency (MHz)	100	100
Throughput (Mbps)	200	200
Efficiency (Mbps/ LUT)	0.083	0.065

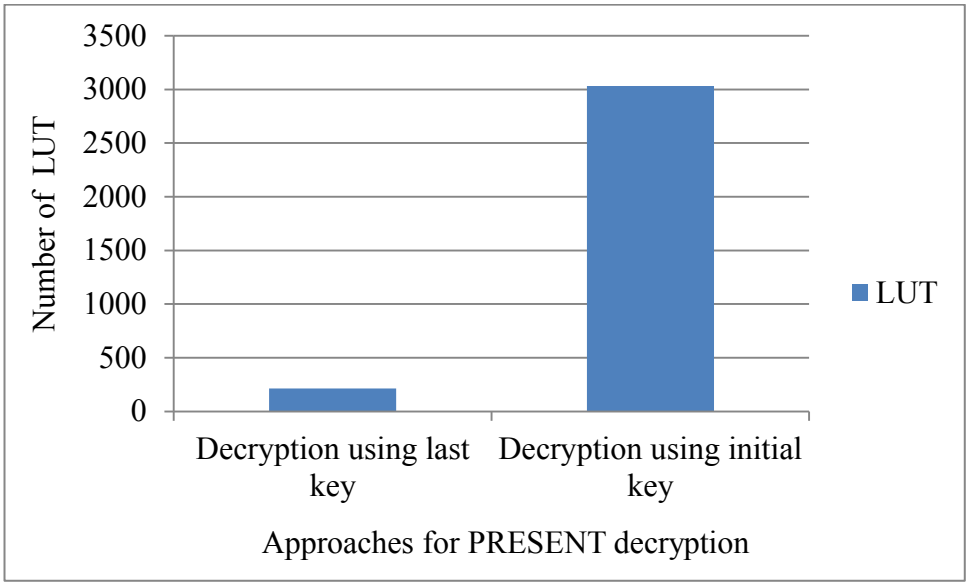

Figure 5.7 Number of LUT for PRESENT Decryption Using Last Key as Input and Initial Key as Input

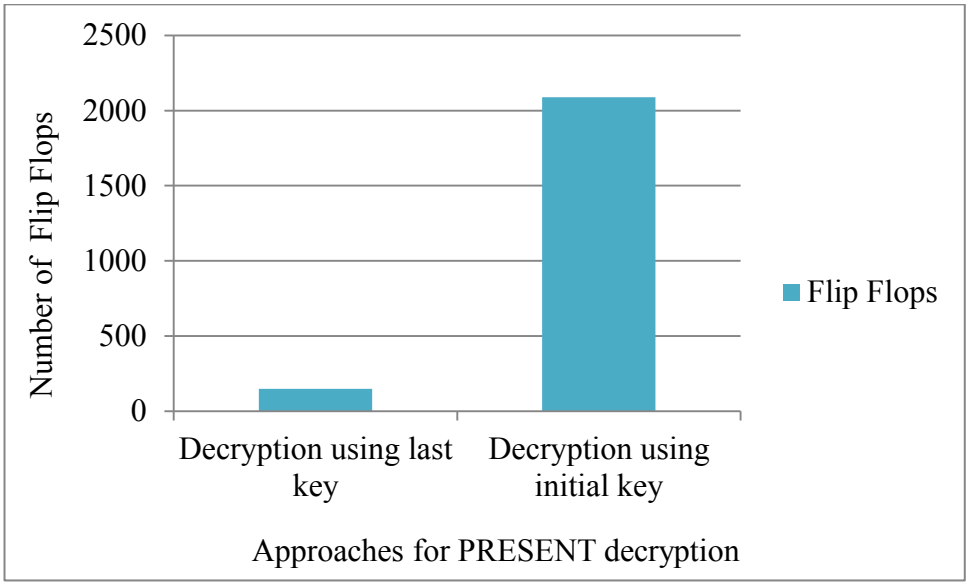

Figure 5.8 Number of Flip Flops for PRESENT Decryption Using Last Key as Input and Initial Key as Input

From Figures 5.7 and 5.8, it is depicted that decryption using last key as input consumes less number of LUT and flip flops. It is because for initial key as input, the key is processed first and stored in an array. This array increases the area requirement. On the other hand, in last key as input, key is processed along with the data, so same key register is updated again and again, hence it consumes less area.

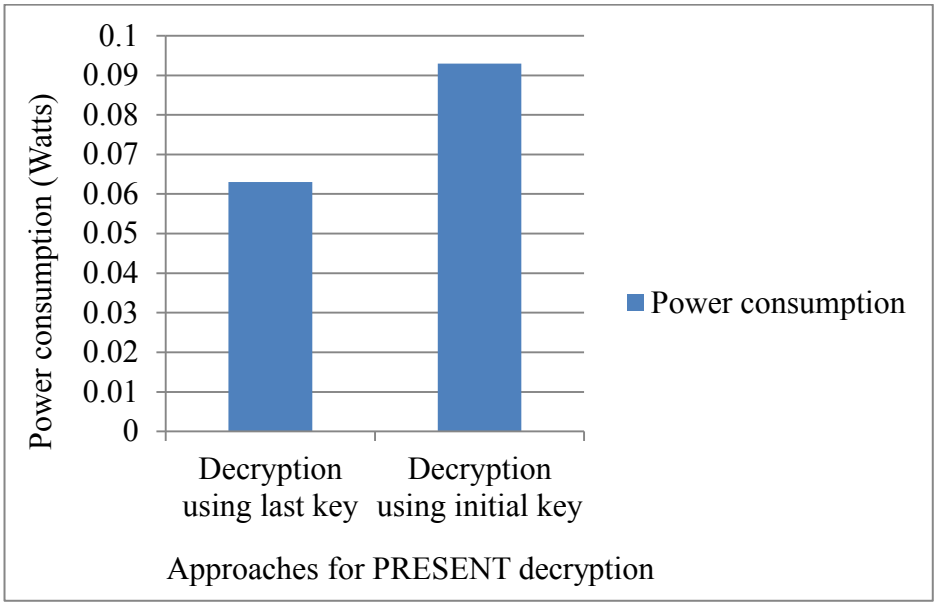

Figure 5.9 Power Consumption for PRESENT Decryption Using Last Key as Input and Initial Key as Input

Figure 5.9 shows that power consumption of using initial key as input is high. This is due to the fact that key processing in loop and then storing it in array requires high computational memory. Some extra computational power is required to process these memory elements data. So, power consumption is high in case of using initial key as input.

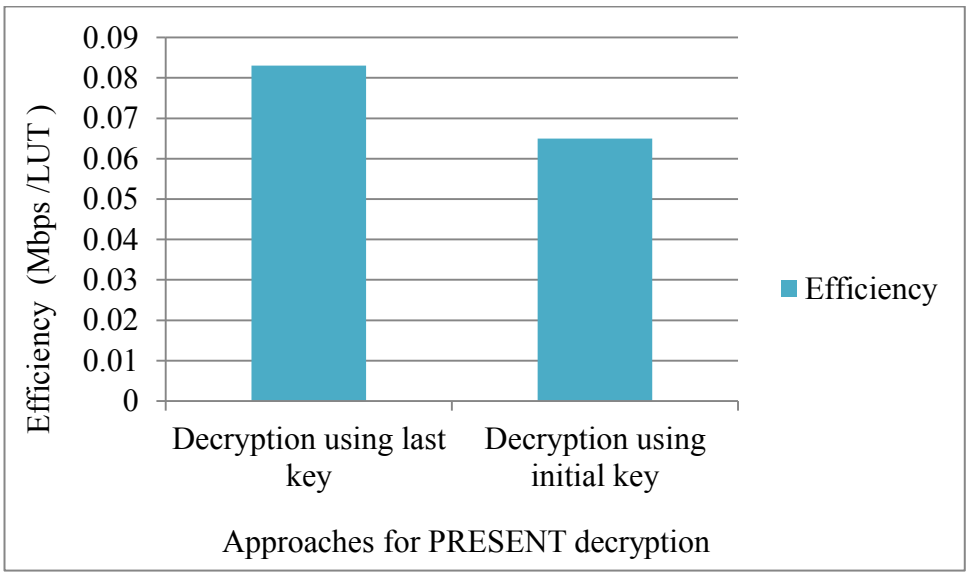

Figure 5.10 Efficiency for PRESENT Decryption Using Last Key as Input and Initial Key as Input

From Figure 5.10, it is depicted that last key as input is more efficient as it produces the same result and security level with less area and power consumption.

5.2.3 Hybrid Algorithm for PRESENT Encryption and Decryption Simulation and Synthesis Results

In case of PRESENT encryption and decryption algorithm an additional control pin is provided at the input which decides whether encryption will occur or decryption will occur. If control input is '1', encryption will take place and if control is '0', the hybrid algorithm will operate as decryption module.

- Control: '1' (Encryption)

Plaintext: 4c 0a 52 3b 19 47 8e 60

Key stream: 25 09 1d 47 38 5e 9f 3c 2b 0a

Cipher text: 0c bc a5 3f 3b 66 b7 98

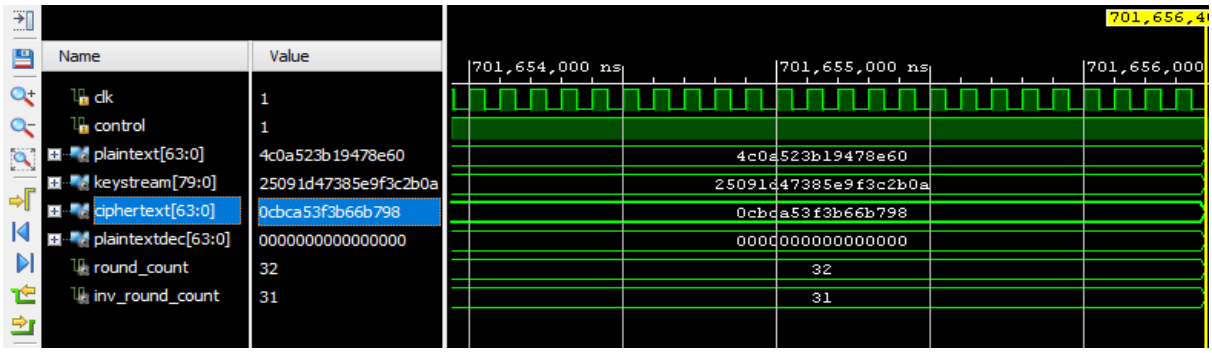

Figure 5.11 Simulation Result for Control='1'

- Control: 0 (Decryption)

Cipher text: 0c bc a5 3f 3b 66 b7 98

Key stream: 25 09 1d 47 38 5e 9f 3c 2b 0a

Plaintext dec: 4c 0a 52 3b 19 47 8e 60

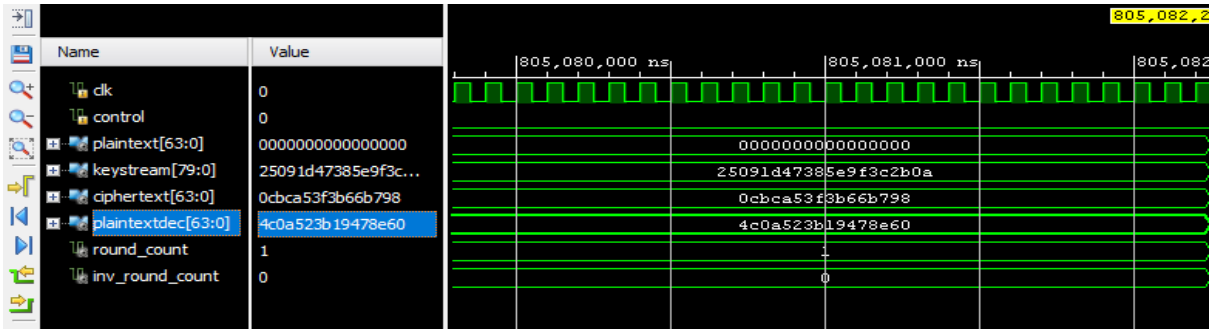

Figure 5.12 Simulation Result for Control='0'

Synthesis results provides resulting values for performance parameters for PRESENT hybrid implementation which are shown in table 5.3

Table 5.3 Performance Parameters Report for PRESENT Encryption and Decryption

Parameters	Resulting value
LUT	4009
Flip Flops	2204
Power consumption (W)	0.098
Frequency (MHz)	100
Throughput (Mbps)	200
Efficiency (Mbps/ LUT)	0.049

5.3 AUTHENTICATION TECHNIQUE SIMULATION AND SYNTHESIS RESULTS

For smart cards authentication, data is transmitted in hidden form along with redundant bits. This algorithm is described from redundant bit security algorithm for RFID. Redundant bits are generated using lightweight ANU block cipher.

Key stream: 25 09 1d 47 38 5e 9f 3c 2b 0a

Cipher temp: 4c 0a 52 3b 19 47 8e 60

Cipher text: 10 50 00 44 11 04 05 45 01 41 10 15 40 54 14 00

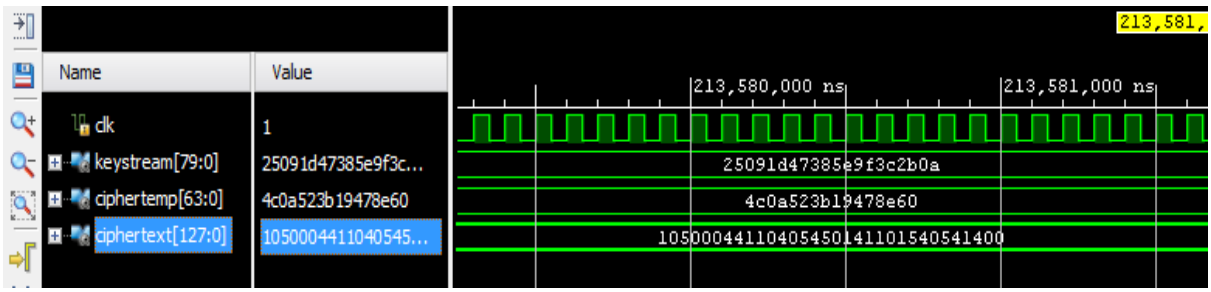

Figure 5.13 Simulation Result for Authentication Algorithm Along with Redundant Bit Generation

Various synthesis results for authentication cipher is shown in Table 5.4

Table 5.4 Synthesis Results of Performance Parameters for Authentication Algorithm

Parameters	Resulting value
LUT	5272
Flip Flops	1289
Power consumption (W)	0.068
Frequency (MHz)	100
Throughput (Mbps)	400
Efficiency (Mbps/ LUT)	0.075

From Table 5.4 it is concluded that proposed authentication scheme provides high efficiency with moderate area and power consumption.

Area consumption for suggested asymmetric algorithms and our implemented RBS algorithm for authentication is presented in table 5.5.

Table 5.5 LUT Based Comparison of RSA, ECC and Implemented RBS

Parameters	RSA [36]	ECC [35]	RBS
LUT	19213	36727	5272

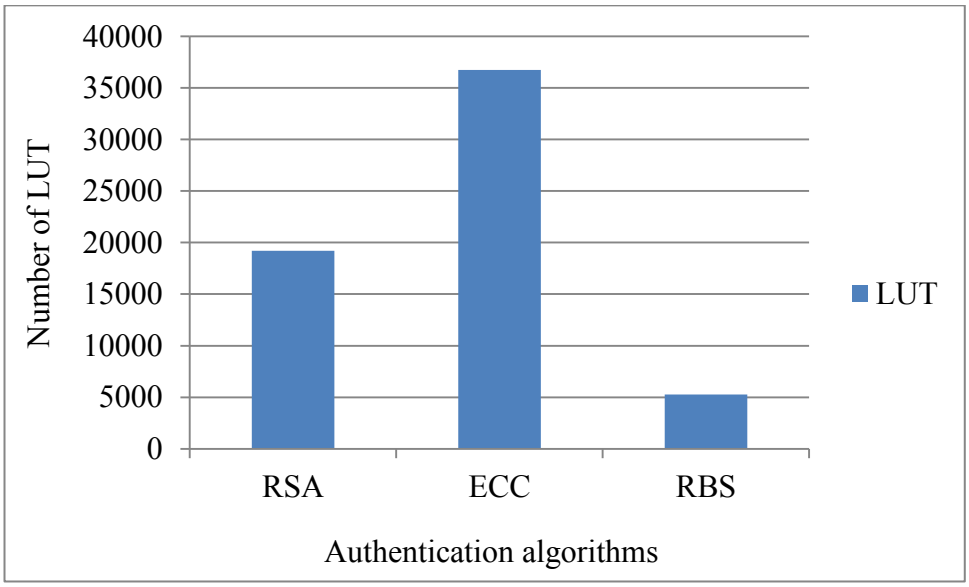

Figure 5.14 LUT Consumed by RSA, ECC and RBS Implementation

From the Figure 5.14 it is clear that RBS consumes least number of LUT which are approximately 4 times lesser than RSA and 7 times lesser than ECC. Hence, RBS is preferred for resource constrained smart cards authentication.

5.4 SMART CARDS ENCRYPTION AND AUTHENTICATION HYBRID ALGORITHM SIMULATION AND SYNTHESIS RESULTS

To fulfil this need of smart cards authentication and encryption, we designed a hybrid approach in which lightweight algorithm PRESENT is used for encryption and Redundant bit security is used for data hiding to provide authentication.

Plaintext: 4c 0a 52 3b 19 47 8e 60

Key stream: 25 09 1d 47 38 5e 9f 3c 2b 0a

Cipher text: 80 5a 67 50 6c 39 25 df 25 47 1c b4 cf b7 69 e8

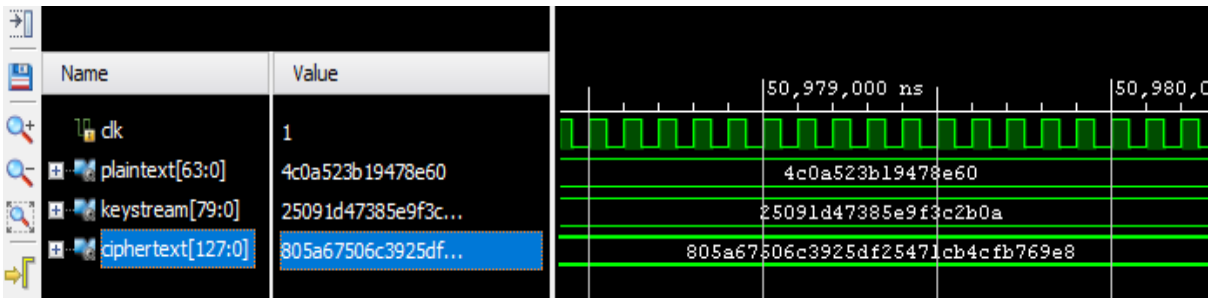

Figure 5.15 Simulation Results of Encryption and Authentication Hybrid Algorithm

Hybrid algorithm of encryption and authentication is further designed by varying the approach of PRESENT S-Box i.e. by using look up table and by using Boolean logic. Table 5.6 depicts various performance parameters for both the approaches

Table 5.6 Comparison of Security Parameters for Authentication and Encryption for Various Approaches of PRESENT S-Box

Parameters	Look up table S-Box for encryption and authentication	Boolean S-Box for encryption and authentication
LUT	9925	9173
Flip Flops	520	502
Power consumption (W)	0.091	0.081
Frequency (MHz)	100	100
Throughput (Mbps)	400	400
Efficiency (Mbps/LUT)	0.037	0.040
Avalanche effect	49.2%	49.2%
Correlation factor	0.1506	0.1506
Brute force attack	2^{128} Combinations Less susceptible	2^{128} Combinations Less susceptible
Linear and differential attack	Not possible	Not possible

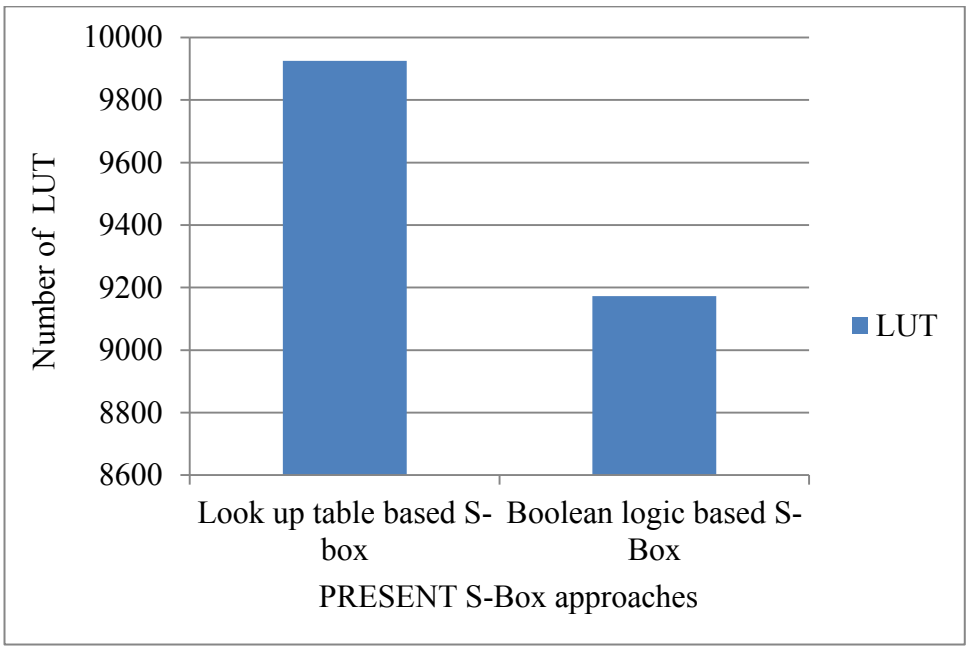

Figure 5.16 Number of LUT for Encryption and Authentication for Various PRESENT S-Box Approaches

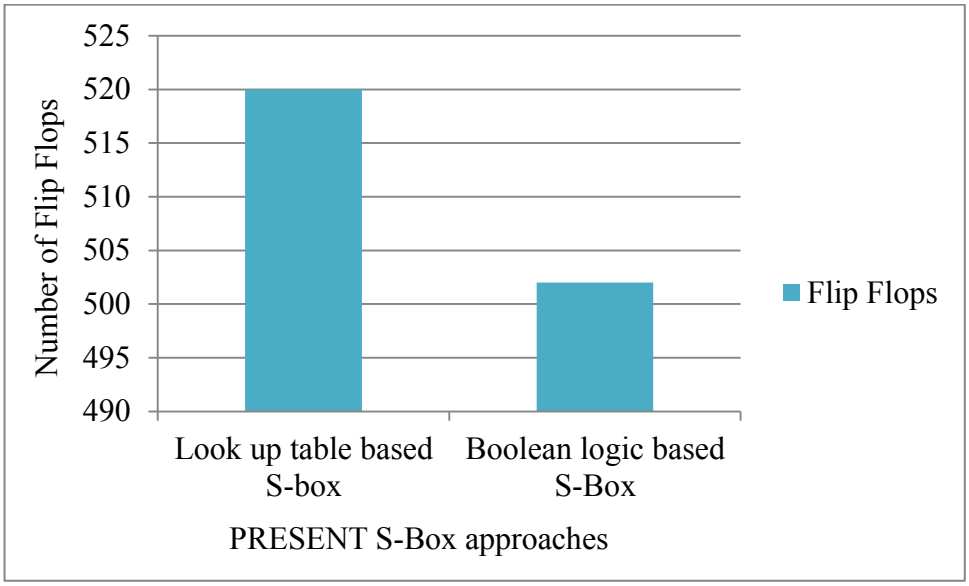

Figure 5.17 Number of Flip Flops for Encryption and Authentication for Various PRESENT S-Box Approaches

Figure 5.15 and 5.16 describe that look up table based approach requires high area. The reason is that processing data in look up table requires extra memory elements than Boolean logic expression.

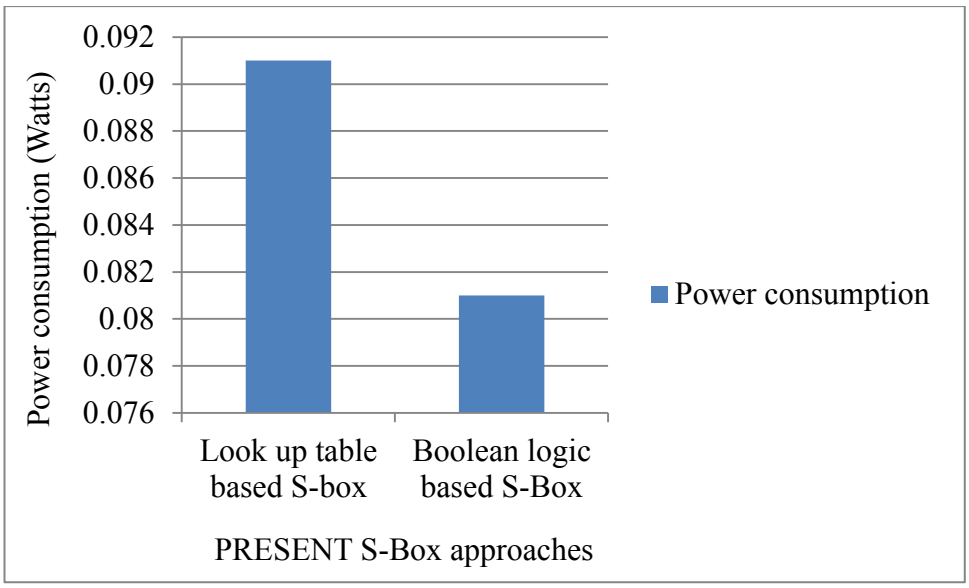

Figure 5.18 Power consumption for Encryption and Authentication for Various PRESENT S-Box Approaches

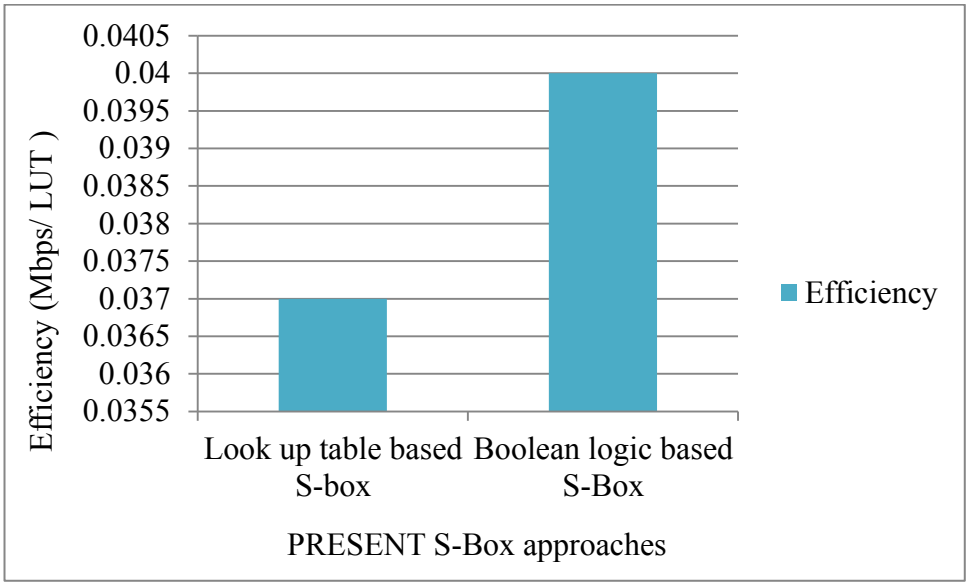

Figure 5.19 Efficiency for Encryption and Authentication for Various PRESENT S-Box Approaches

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

Smart cards play vital role as storage element, identification entity and authentication tool for various real life applications. Smart cards are considered as secure carrier of highly classified data as financial user records, healthcare emergency information, personal identification details etc. But there are some security threats related to smart cards. To counter these threats various cryptography algorithms are applied on smart cards. From literature survey it is observed that symmetric key algorithms AES, DES are used for data encryption and asymmetric key cryptography algorithms RSA, ECC are preferred for authentication purposes. But AES and DES require large area for implementation. RSA and ECC are highly secure but these also require high computational power and memory. So, all these algorithms are considered less efficient for use in smart cards. To resolve these constrained performance issues, a new hybrid technique of encryption and authentication is proposed for use in smart cards.

In this work, optimized PRESENT algorithm is implemented for data security. PRESENT is designed using look up table based and Boolean logic based S-Box approaches and compared with the existing PRESENT implementation. The implemented PRESENT consumes 33.42% less area in look up table based S-Box approach due to use of look up table in place of array for substituting values. Boolean logic based approach uses 6.33% less as factorization is applied in Boolean logic based S-Box. Decryption module of PRESENT is designed just by reversing the encryption module with two approaches of key design i.e. using last key and using initial key as input key stream. Decryption using last key is 6 times more efficient than using initial key as input. Then, a hybrid approach for PRESENT encryption and decryption is implemented.

For authentication of smart cards, redundant bits are generated using lightweight block cipher ANU and authentication is achieved by hiding of data bits with generated redundant bits. Proposed authentication scheme is 7 times area efficient than existing algorithms for authentication.

A hybrid technique using optimized PRESENT for encryption and authentication algorithm based on redundant bits security is implemented using two approaches of PRESENT S-Box. From the performance analysis, it is concluded that it is very efficient in terms of LUT, Flip

flops, throughput and power consumption. Also the implemented scheme is resistant to attacks such as brute force attack, linear and differential cryptanalysis.

6.2 FUTURE SCOPE

Smart cards are chosen best for the applications where data security and authentication is prime concern. Various authentication schemes and cryptographic algorithms are implemented for gaining efficient level of security. From literature survey some research gaps are observed. Following are some research directions to resolve these gaps:

- Two level authentication scheme includes sending OTP to user via SMS. This leads to the man-in-middle and impersonation attacks on smart card. To resist such attacks the encrypted format of OTP should be send.
- In case of block cipher, if data bits are less than block size, zero padding is required which leads to extra operation required for random padding. To reduce this overhead stream cipher must be applied on smart cards.
- ECC based authentication schemes are highly secure against various attacks but at the same time these require high computational power for mathematical calculations hence some power optimization techniques such as clock gating, wire length reduction, hierarchical design etc. can be implemented on the designed scheme.

REFERENCES

[1] Sweta, "Smart Card and Its Application," *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 5, Issue 7, pp. 1158-1160, July 2015.

[2] G. Selimis, A. Fournaris, G. Kostopoulos and O. Koufopavlou, "Software and Hardware Issues in Smart Card Technology," in *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 143-152, 2009.

[3] K. Markantonakis and K. Mayes, *Smart Cards, Tokens, Security and Applications*, Springer, 2008.

[4] W. Rankl and W. Effing, *Smart card handbook*. Chichester: Wiley, 2010.

[5] L. A Mohammed, Abdul Rahman Ramli, V. Prakash, and Mohamed B. Daud, "Smart Card Technology: Past, Present, and Future," *International Journal of The Computer, the Internet and Management,* Vol. 12, pp. 12 – 22, 2004.

[6] R.S. Pippal, C. D. Jaidhar, and S. Tapaswi,"Security Issues in Smart Card Authentication Scheme," *International Journal of Computer Theory and Engineering*, vol. 4, no. 2, p.p. 206-215, 2012.

[7] Ritu Tripathi and Sanjay Agrawal, "Comparative Study of Symmetric and Asymmetric Cryptography Techniques," in *International Journal of Advance Foundation and Research in Computer (IJAFRC),* Volume 1, Issue 6, p.p. 2348 – 4853, June 2014.

[8] E. Surya and C. Diviya, "A Survey on Symmetric Key Encryption Algorithms," *International Journal of Computer Science & Communication Networks*, pp.475-477, 2012.

[9] M. Agrawal and P. Mishra, "A Comparative Survey on Symmetric Key Encryption Techniques," in *International Journal on Computer Science and Engineering (IJCSE)*, vol. 4, no. 4, May 2012.

[10] S.O. Sharif and S.P. Mansoor, "Performance analysis of Stream and Block cipher algorithms," *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE),* vol. 1, pp. 522-526, 2010.

[11] H. Dhiman, Md Asif Mushtaqe and S. Hussain, " An analytical analysis of stream ciphers and block cipher algorithms," in *International Journal for Research in Applied Sciences and Engineering Technology*, vol. 2, no. 5, May 2104.

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

[13] A. Singhal and M. Ramayia, "A novel safe and efficient smart card authentication scheme using hash function," in *Engineering Universe for Scientific Research and Management*, vol. 7, no. 1, 2015.

[14] H. Mittal, "Diffie-Hellman Based Smart-Card Multi-server Authentication Scheme," *2014 International Conference on Computational Intelligence and Communication Networks*, Bhopal, pp. 808-812, 2014.

[15] S. Nivetha, N. E. Elizabeth, T. P. Padmasha, and I. Gohulalakshmi, "Secure authentication process in smart cards," *10th International Conference on Intelligent Systems and Control (ISCO)*, 2016.

[16] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, "An Efficient and Practical Smart Card Based Anonymity Preserving User Authentication Scheme for TMIS using Elliptic Curve Cryptography," *Journal of Medical Systems*, vol. 39, no. 11, Mar. 2015.

[17] C. T. Li, "A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card," in *IET Information Security*, vol. 7, no. 1, pp. 3-10, March 2013.

[18] Ravi Singh Pippal, Pradeep Gupta and Rakesh Singh, "A Novel Smart Card Authentication Scheme using Image Encryption," *International Journal of Computer Applications*, vol. 72, no. 72, pp. 8-14,2013.

[19] N. Druml , M. Menghin , A. Kuleta, C. Steger, R. Weiss, H. Bock, J. Haid "A Flexible and Lightweight ECC-Based Authentication Solution for Resource Constrained Systems," *17th Euromicro Conference on Digital System Design*, pp.372-378,2014.

[20] C. Cai, Y. Zhu and B. Wang, "A Novel Mutual Authentication Scheme for Smart Card without Information Leakage," *IEEE 17th International Conference on Computational Science and Engineering*, pp. 599-604, 2014.

[21] H.Tang, X. Liu and L. Jiang, "A Robust and Efficient Timestamp-based Remote User Authentication Scheme with Smart Card Lost Attack Resistance," International Journal of Network Security, vol.15, no.6, pp.446-454, Nov. 2013.

[22] J. Borst, B. Preneel, and V. Rijmen, "Cryptography on smart cards," *Computer Networks*, vol. 36, no. 4, pp. 423–435, 2001

[23] K. Dichou, V. Tourtchine, and F. Rahmoune, "Finding the best FPGA implementation of the DES algorithm to secure smart cards," *2015 4th International Conference on Electrical Engineering (ICEE)*, pp. 1-4, 2015.

[24] C. Rolfes, A. Poschmann, G. Leander, and C. Paar, "Ultra-Lightweight Implementations for Smart Devices – Security for 1000 Gate Equivalents," *Smart Card Research and Advanced Applications Lecture Notes in Computer Science*, pp. 89–103, 2008.

[25] Y. Eslami, A. Sheikholeslami, P. G. Gulak, S. Masui, and K. Mukaida, "An area-efficient universal cryptography processor for smart cards," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 1, pp. 43-56, 2006.

[26] S. K. Dhanuka, P. Sachdeva, and S. S. Shaikh, "Cryptographic algorithm optimisation,"*IEEE International Advance Computing Conference (IACC)*, pp. 1111-1115. 2015.

[27] Z. Peng and J. J. Fang, "Comparing and implementation of public key cryptography algorithms on smart card," *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, pp. 508-510, 2010.

[28] E. Surya and C. Diviya. "A Survey on Symmetric Key Encryption Algorithms." *International Journal of Computer Science & Communication Networks*, pp.475-477, 2012.

[29] B. Ege, E. B. Kavun, and T. Yalçın, "Memory Encryption for Smart Cards," *Smart Card Research and Advanced Applications Lecture Notes in Computer Science*, pp. 199–216, 2011.

[30] Q. Zhang, J. Cao, D. Yu, X. Cao, X. Zhang, Y. Ye and B. Chen, "A low-energy high-throughput asynchronous AES for secure smart cards," *IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, pp. 487-490, 2015.

[31] M. Savari and M. Montazerolzohour, "All about encryption in smart card," *Proceedings Title: International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, pp. 54-59, 2012.

[32] Y. A. Setyoko and I. G. B. Baskara Nugraha, "Multipurpose Smart Card System," *International Conference on ICT For Smart Society (ICISS)*, pp. 264-268, 2014.

[33] M. Savari, M. Montazerolzohour and Y. E. Thiam, "Combining encryption methods in multipurpose smart card," *Proceedings Title: International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, pp. 43-48, 2102.

[34] Y. Zhang and X. Wang, "Pipelined implementation of AES encryption based on FPGA," *2011 IEEE International Conference on Information Theory and Information Security*, pp. 170-173, 2010.

[35] M. Masoumi and H. Mahdizadeh, "Efficient Hardware Implementation of an Elliptic Curve Cryptographic Processor over GF (2^{163})," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 6, no.5, 2012.

[36] E. A. Michalski and D. A. Buell, "A Scalable Architecture for RSA Cryptography on Large FPGAs," *2006 International Conference on Field Programmable Logic and Applications*, pp.1-8, 2006.

[37] "Information technology – Security techniques –Lightweight cryptography – Part 2: Block ciphers", ISO/IEC 29192-2:2012, ISO, 2012.

[38] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," *Cryptographic Hardware and Embedded Systems - CHES Lecture Notes in Computer Science*, pp. 450–466,2007.

[39] P. Yalla and J. P. Kaps, "Lightweight Cryptography for FPGAs," *2009 International Conference on Reconfigurable Computing and FPGAs*, pp. 225-230, 2009.

[40] J. C. H. Castro, P. P. Lopez and J. P. Ausmasson, "On the key schedule strength of PRESENT," Lecture Notes in Computer Science, vol. 7122, 2012.

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

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

[43] Z. Jeddi, E. Amini and M. Bayoumi, "RBS: Redundant Bit Security Algorithm for RFID Systems," *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, pp. 1-5, 2012.

[44] G. Bansod, A. Patil, S. Sutar and N. Pisharoty, "An Ultra Lightweight Encryption Design for

Security in Pervasive Computing," *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, pp. 79-84, 2016.

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

[46] M. Rani and Dr. S. Kumar, "Analysis on Different Parameters of Encryption Algorithms for Information Security," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 8, p.p. 104-107, 2015.

[47] J. Pospisil and M. Novotny, "Lightweight cipher resistivity against brute-force attack: Analysis of PRESENT," *IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, pp. 197-198, 2012.

[48] A. Arora, Priyanka and S.K. Pal. "A Survey of Cryptanalytic Attacks on Lightweight Block Ciphers," *IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS)*, vol. 2, no.2, April 2012.

PUBLICATIONS

- Jaskaranbeer Kaur, Ajay Kumar and Manu Bansal, "Lightweight Cipher Algorithms for Smart Cards Security: A Survey and Open Challenges," 4th International Conference on Signal Processing, Computing and Control(ISPCC-2017), sponsored by IEEE.(Accepted)

601562012_jaskaran

ORIGINALITY REPORT

%5	%3	%4	%0
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Tay, J. J., M. L. D. Wong, M. M. Wong, C. Zhang, and I. Hijazin. "Compact FPGA implementation of PRESENT with Boolean S-Box", 2015 6th Asia Symposium on Quality Electronic Design (ASQED), 2015. Publication	%1
2	www.crypto.ruhr-uni-bochum.de Internet Source	%1
3	www.ci2ma.udec.cl Internet Source	<%1
4	eprint.iacr.org Internet Source	<%1
5	pdfs.semanticscholar.org Internet Source	<%1
6	tavanaonline.com Internet Source	<%1
7	www.enggjournals.com Internet Source	<%1

"Evolutionary Computation and Cryptology",

8	Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion - GECCO 16 Companion, 2016. Publication	<%1
9	www.ukessays.com Internet Source	<%1
10	www.knowingtheworld.com Internet Source	<%1
11	www.crypto.rub.de Internet Source	<%1
12	usbeta.ru Internet Source	<%1
13	Understanding Cryptography, 2010. Publication	<%1
14	Sodagudi, Suhasini and Kurra, Rajasekhara Rao. "Towards Security In Cognitive Radio Networks", International Journal of Applied Engineering Research, 2015. Publication	<%1
15	components.alldatasheet.fr Internet Source	<%1
16	www.faqs.org Internet Source	<%1
17	Zhang, Yuanyuan Khan, Muhammad Khurram C. "Provable secure and efficient digital rights	<%1