Role of Cryptographic Welch-Gong (WG-5) Stream Cipher in RFID Security

by

Rajesh Kumar Mota

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2012

©Rajesh Kumar Mota 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The purpose of this thesis is to design a secure and optimized cryptographic stream cipher for passive type Radio Frequency Identification (RFID) tags.

RFID technology is a wireless automatic tracking and identification device. It has become an integral part of our daily life and it is used in many applications such as electronic passports, contactless payment systems, supply chain management and so on. But the information carried on RFID tags are vulnerable to unauthorized access (or various threats) which raises the security and privacy concern over RFID devices. One of the possible solutions to protect the confidentiality, integrity and to provide authentication is, to use a cryptographic stream cipher which encrypts the original information with a pseudo-random bit sequence. Besides that RFID tags require a resource constrained environment such as efficient area, power and high performance cryptographic systems with large security margins. Therefore, the architecture of stream cipher provides the best trade-off between the cryptographic security and the hardware efficiency.

In this thesis, we first described the RFID technology and explain the design requirements for passive type RFID tags. The hardware design for passive tags is more challenging due to its stringent requirements like power consumption and the silicon area. We presented different design measures and some of the optimization techniques required to achieve low-resource cryptographic hardware implementation for passive tags.

Secondly, we propose and implement a lightweight WG-5 stream cipher, which has good proven cryptographic mathematical properties. Based on these properties we measured the security analysis of WG-5 and showed that the WG-5 is immune to different types of attacks such as algebraic attack, correlation attack, cube attack, differential attack, Discrete Fourier Transform attack (DFT), Time-Memory-Data trade-off attack. The implementation of WG-5 was carried out using 65 nm and 130 nm CMOS technologies. We achieved promising results of WG-5 implementation in terms of area, power, speed and optimality. Our results outperforms most of the other stream ciphers which are selected in eSTREAM project.

Finally, we proposed RFID mutual authentication protocol based on WG-5. The security and privacy analysis of the proposed protocol showed that it is resistant to various RFID attacks such as replay attacks, Denial-of-service (DoS) attack, ensures forward privacy and impersonation attack.

Acknowledgements

I would like to thank my MASc supervisor, Dr. Mark Aagaard for his support during this work. I would also like to thank my thesis readers Dr. Guang Gong and Dr. Anwar Hasan for their insightful comments that helped in improving the quality of this thesis. I would like to thank my fellow researchers Kalikinkar Mandal and Hazem Shehata for their valuable discussions during this work. Finally, I would like to thank Ontario Graduate Scholarship (OGS) program for their generous support in this research work.

My family has played a significant role and very supportive during my entire work. No words to express my gratitude to my wonderful wife, Swayam. She has been understanding and supported me unconditionally.

Dedicated to my mom, dad, sister, brother and my wife.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Cutting edge technologies have made recent advances in microelectronics. One of the most advanced and widely used technologies is wireless, non-contact automatic system or Radio Frequency Identification (RFID) [2]. It was first introduced during World War II, since then, it has become an integral part of our daily lives. Nowadays, RFID tags are used in various applications such as, item tracking, contactless smart cards, access control, e-passports, supply chain management and so on [3]. The use of this technology by some of the major companies like Wal-Mart Stores Inc. and the US military forces agency have greatly increased the popularity and demand of RFID systems.

RFID systems consist of mainly three components; *tags*, which can store information or data; *readers*, which is used for reading the data from the tags; *back-end database* system which is connected to readers via network for the processing of tag's data. Generally, the information stored in RFID tags are in the form of unique code called as Electronic product code (EPC) and is available in various lengths (96, 108 bits) [4] [5] depending upon the application.

Based on their power sources RFID tags are differentiated into *Passive*, *active* and *semi passive*. In the current work we focus on the passive type RFID tags. Passive tags do not contain batteries unlike semi passive and active tags. Passive tags extract power from the reader via electromagnetic waves which makes it small and less expensive. The size of the tag varies depending on applications. Due to the technological advancements in semiconductor manufacturing, researchers are shrinking the size of RFID tag which further reduces its cost. With the increasing applications and high demands, RFID systems must full fill some of the design goals such as low cost, small size and higher data rates.

The aim of the present work is to propose a lightweight stream cipher called WG-5 for the low cost passive RFID tags. WG-5 is a variant of WG stream ciphers [6], which was submitted to eSTREAM project in 2007. Further, it was focused as a multi-output WG ciphers (MOWG) [7] wherein, its trade offs between the security and hardware matrices were discussed. In [8], WG-7 has been proposed as a lightweight stream cipher and its software implementation was carried out on different micro controller units. In the present work, the design goal of WG-5 stream cipher is to ensure cryptographically secure and to achieve hardware implementation efficiently in terms of less gate count, low power and better performance using the current CMOS technologies. The next goal is to propose a mutual authentication protocol based on the proposed lightweight stream cipher. Further, the proposed protocol must be secure against some of the attacks such as man-in-the-middle, denial of service, tampering of tags and also to ensure RFID privacy.

1.1 Motivation

The major concern in RFID systems is security and privacy. Usually, RFID tags communicate data in the form of unique code, to the readers, over an unsecured communication channel. The unique code can be related to manufacturer details of any product or it can be personal information. The security and privacy issues arise in RFID systems when the unique code are accessed by an unauthorized readers by eavesdropping or rogue monitoring. This leads to manipulation of tags data, tracking the objects or persons. To overcome the aforementioned concerns, cryptographic solutions or alternatively symmetric cipher cryptosystems are introduced in low cost passive RFID tags.

The silicon area utilization of the passive tags is directly proportional to the cost of the tag and according to one of the previous studies, the cost of the passive tags should not exceed five cents [9]. Based on current advanced CMOS technologies, the design of cryptographic circuitry for RFID tags should not be more than 5000 GE (gate equivalence) [10] [11]. The symmetric stream ciphers can be implemented in RFID tags with less than 5000 GE [12].

Since passive RFID tags receive their power from the reader, the power consumption is an important factor to be considered. The power consumption usually vary depending on the tag operations performed such as; transmission rate, response time and writing the data into tags memory [10]. Generally, the power available for cryptographic design in passive tag should be in the range of 5 to 15 μA [10] [13]. Therefore, passive tags have very limited power availability to perform the aforementioned operations. Thus, one can envisaged that, the implementation of symmetric cryptosystem in low cost passive tags would be a challenging task due to its constrained environment. The cryptographic designs which achieve these RFID passive design goals such as less area, low power and low cost are called *lightweight cryptography*.

In addition, *single authentication protocols* are one of the solutions to ensure security and privacy in RFID systems which ensures that, the data is communicated between the two authorized persons. If the tag authenticates a reader it means that the data is sent only to the authorized reader. Similarly, if the reader authenticates a tag, it suggests that the tag is not forged by any unauthorized entities. Moreover, compared to *single authentication protocols*, the *mutual authentication protocol* is more secure way of communication in which both the reader and the tag authenticates each other before exchanging the data. Most of the authentication protocols published are based on block ciphers or other cryptographic systems such as hash functions [14] [15] [16] [17] [18] and very few protocols are based on stream ciphers [19] [20].

1.2 Organization

In chapter 2, we discuss the background information on various cryptographic systems and their properties. In addition, we explain different types of attacks that an adversary can perform over a cryptographic system. In chapter 3, we describe the overview of mathematical concepts like finite fields, polynomials, normal, polynomial and optimal normal basis which play an important role in designing a symmetric stream ciphers. Since, the design of stream cipher depends on the selection of basis we explain various options that can be selected depending on the hardware requirements in section 3.4. Later in section 3.5, we discuss the LFSR properties and its hardware representation for any given polynomials over a finite field.

In chapter 4, we discuss the overview of stream ciphers and different types of attacks on them. In section 4.5, we describe the WG stream cipher [6] and its proven cryptographic properties. WG stream cipher is based on the mathematical definitions and the keystream is generated based on its WG transformation function which are explained in section 4.5.2 and 4.5.3 respectively. Since, WG transformation function can be implemented in different architectures, the procedure for selecting the parameters for its implementation were explained in section 4.5.5. The important parameters that are involved in design of WG stream cipher are number of bits over the finite field, primitive feedback polynomial, generating polynomial, length of the LFSR, representation of field elements either in normal or polynomial basis or ONB, multiplier architectures like serial or parallel based depending on the selection of basis. These parameters can affect the security level of WG cipher and utilization of hardware resources of the cipher.

In chapter 5, we gave an overview of RFID systems, its applications, different frequency bands, EPC structure and standards. The main focus of this chapter is to explain the security hardware design requirements for passive type RFID tags. Section 5.5, explains the role of cryptographic solutions, RFID optimality metrics, RFID hardware design considerations and different optimization techniques for passive type RFID tags. Finally, section 5.6 and 5.7 describes some of the RFID privacy goals and definitions and explain different types of attacks on RFID systems respectively.

In chapter 6, we propose lightweight stream cipher WG-5 for low cost passive RFID tags.

After selecting the design parameters for WG-5, we describe the architecture of WG-5 keystream generator and calculated the related properties of WG-5 in section 6.1 and 6.2 respectively. Based on the measured properties we carried out the security and privacy analysis of WG-5 stream cipher in section 6.3. Later, we carried out the implementation details of WG-5 cipher starting with the datapath i.e. WG-5 core and then the control units; LFSR and FSM in section 6.4. Furthermore, we measured the implementation details of WG-5 cipher using 65 and 130 nm CMOS technology in section 6.5. In section 6.6, we proposed RFID mutual authentication protocol based on WG-5 stream cipher and its security and privacy analysis has been carried out in section 6.6.1. Atlast, we compared the WG-5 results with other existing stream and block ciphers in section 6.7.

Finally, in chapter 7 the conclusion of present work is described along with the future work.

Chapter 2

Overview of Cryptography

The communication over an insecure channel can be accessed by unauthorized persons. In order to prevent from unauthorized accesses, cryptography is widely used in different applications. Cryptography provides few of the security objectives which is discussed in section 2.2. In section 2.3, the types of cryptosystems such as symmetric and asymmetric key systems are discussed. In cryptography, hash functions and digital signatures play an important role in achieving some of the security objectives, which is explained in section 2.4. In order to achieve high level of security few of the cryptographic protocols has been described in section 2.5. While, in section 2.6 the concept of pseudo-randomness is discussed. Atlast in section 2.7 different types of possible attacks on the cryptographic systems are discussed.

2.1 Keywords

Keywords: These are the following keywords listed below which are used throughout this material.

• ***Information***: It can be a data or a message which is communicated between two-parties. The parties can be either a person or a computer terminal.

• ***Sender***: Sender is one of the two-parties, who transmits the information.

• ***Receiver***: Reciever is also one of the two-parties, who receives the information.

• ***Attacker/adversary/eavesdropper***: An unauthorized party who tries to get access to the secure communication channel used by the sender and receiver.

• ***Channel***: It is a means of communication between the sender and receiver. Channels can be secure and unsecured. Secure channel does not allow the attacker to modify, delete, insert or read the information, compared to unsecured channel.

- ***Confidentiality or Secrecy***: Assurance for the data or information that is communicated and cannot be accessed by any unauthorized parties.

- ***Cryptography***: Study of mathematical techniques which deal with the information security. It enables the confidentiality, when information is communicated via unsecured channel. Cryptographic algorithms are designed by mathematical techniques to introduce security.

- ***Cipher***: Any encryption technique referred as a cipher. Formally, its a technique of concealing the readability and meaning of the original information.

- ***Cryptographic system***: Its a collections of various cryptographic algorithms which includes ciphers and cryptographic protocols.

- ***Cryptanalysis***: A technique used for deciphering the information, without any knowledge of enciphering details. Persons who work on cryptanalysis are called as attackers or cryptanalysts.

- ***Entity***: An entity can be referred to as person or computer terminal.

2.2 Objectives of cryptography

The main goal of cryptography is to prevent and detect the fraud from the malicious activities. Cryptography not only provide better security over the communication channels but also provides following services.

1. *Confidentiality:* It ensures that when the information is transmitted over a channel and not accessed by any unauthorized person.

2. *Data Integrity:* It ensures the ability to address the manipulation of original data. Manipulation can be either insertion, modification or deletion.

3. *Authentication:* It ensures the ability to identify or to verify the communicating parties so that, no one should be able to pretend as Alice and send an information to Bob (*data origin authentication*). Alice and Bob should be able to identify each other (here Alice is a sender and Bob is a receiver) (*entity authentication*) [21]

4. *Non-repudiation:* It prevents from denying the previous commitments or actions done by the communicating parties.

2.3 Overview of Cryptosystem

It is a collection of various cryptographic algorithms which includes ciphers and various cryptographic protocols. The model of the cryptosystem is shown in Figure 2.1. There are three parties involved in the cryptosystem called as transmitter, receiver and adversary. Initially the transmitter chooses a *plaintext* message before it transmits over a channel. Generally communication takes place over an insecure channel before transmitting to the legitimate receiver. In order to avoid malicious activities by the adversary about the plaintext; the transmitter transforms the plaintext into unknown format known as *ciphertext* with the help of a *secret key* (sequence of bits). The process of transformation is known as *encryption*. Before transmitting the information, the sender must share the secret key with the receiver by means of some secure channel. Once receiver knows the secret key and recovers the plaintext message by applying the transformation called as *decryption* over the ciphertext.

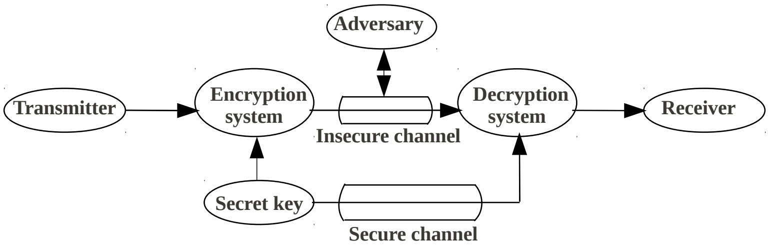


Figure 2.1: Cryptosystem

Encryption algorithm is a transformation or substitution of steps while, *decryption algorithm* is just a reverse run of it. Usually, the secret key and the plaintext are inputs for the encryption algorithm. Secret key values are independent of the plaintext and the encryption algorithm produces various outputs for different secret key values. For secure cryptosystem one should design a strong encryption algorithm and choose the secret key wisely. For example, the secret key must be long and possess randomness properties so that, the adversary should not decrypt the ciphertext and discover the secret key.

There are two types of cryptosystem which are as follows. If the sender and receiver uses the same secret key for encryption and decryption operation, then the system is called as the *symmetric key* or single-key or conventional cryptosystem [22]. If both the sender and receiver uses two different keys thats is *public-key* (non-secret) and *private-key*(secret) for encryption and decryption then the system is called as *asymmetric key* ,two-key or *public-key cryptosystem*.

2.3.1 Symmetric key cryptosystems

Symmetric-key cryptosystems provide secure communication between a pair of communication parties and the adversary who tries to intercepts the message (m) cannot get any significant information about the message contents. In this system both the sender and receiver share the same secret key (k) for both encryption (E) and decryption (D) operations.

Example: If Alice wants to send a message m to Bob through a secure communication channel. Using encryption algorithm, Alice first generates ciphertext (C) as $E(k, m) = C$ and sends it to Bob. Bob on the other side, using decryption algorithm he restores the original message as $D(k, C) = m$.

Typically, symmetric key cryptosystems are used for ensuring the *confidentiality* and *integrity* of the data. One way to achieve data integrity is by using *message authentication code* (MAC) algorithm [23]. In this algorithm, first the sender generates the MAC code which is a simple block of data that is generated depending on the message length using secret key by running the MAC algorithm. The message along with the MAC code are sent to the receiver and the receiver would check the *integrity* of the incoming message by running the same MAC algorithm and comparing the transmitted MAC code. If the code is identical then the receiver can assure that there is no modifications done to the message.

Example: If Alice wants to send a message m to Bob, then Alice generates MAC code as $X = MAC(k, m)$ using MAC algorithm. Usually the MAC code is protected by a secret key k. Now Alice sends X along with m to Bob. At the receiver side Bob verifies the data integrity of the message by checking the MAC code.

Symmetric key cryptosystems are of two types; *Stream ciphers* and *Block ciphers* and explained below.

Stream ciphers:

In stream cipher the plaintext is converted to ciphertext by one bit at a time. It generates arbitrarily long stream of key material (bits) known as *keystream*. The generation of *keystream* output is based up on the *internal state* which is usually hidden inside the cipher and changes frequently as cipher operates. During encryption the *keystream* is XORed (exclusive-or operation) with each plaintext one bit at a time. Some of the examples of stream ciphers are Welch-Gong *(WG)* cipher, *RC4*, *grain*, *trivium*, *A5/1* and so on.

Block ciphers:

Block cipher operates on the fixed length blocks (i.e. group of bits) of plaintext or ciphertext. The encryption operation is an unvarying transformation, which is controlled by using the secret key. For example, a block cipher might take 128-bit block of plaintext as an input and generate 128-bit block of ciphertext. Examples of block ciphers are Data Encryption Standard *(DES)*, Advanced Encryption Standard *(AES)* and so on.

2.3.2 Asymmetric key cryptosystems

Public key cryptosystem is a well known example of asymmetric key cryptosystem. The concept of public key cryptosystem was mainly introduced to distribute the secret keys securely, which is also known as *key agreement scheme* [21]. The basic idea behind the public key cryptosystem is that, each person contains two different keys one is called the *public key* (pk) and the other is called the *private key* or *secret key* (sk). The public key is available to everybody where as, the secret key is kept secret and it is known only to the owner. The public key is used for encrypting the message while the private key is used for decryption.

Example: In public key cryptography Alice uses Bob's public key for encryption operation as $E(pk, m) = C$ and sends ciphertext to Bob. Bob on the other hand recovers the message from ciphertext, using his secret key by decryption operation as $D(sk, C) = m$.

From the above example we need to know that only Bob knows the secret key and nobody can decrypt the message, even Alice cannot decrypt the message back even if she has lost the message. This system provides better *confidentiality* of the data.

The well known examples of public key algorithms are *RSA* (named after its inventors: Rivest, Shamir, Adleman) and *Diffie–Hellman* algorithms.

2.3.3 Symmetric vs asymmetric key cryptography

1 . Symmetric key systems provide secure communication channel whereas asymmetric-key systems are good at secure key exchange management.

2 . Design of symmetric key systems are easy to analyze and implement in hardware implementation compared to asymmetric. Because, symmetric key systems consists of simple mathematical operations involving addition, subtraction and multiplication operations. and these operations can be implemented by using simple XOR and AND gates in hardware. Where as, public key systems consists of hard complex mathematical operations which involves addition, multiplication and division or inversion, which utilizes more number of gates for implementation, for example elliptic curve crypto systems.

3 . Length of key for symmetric-key ciphers are relatively short compared to asymmetric-key systems.

4 . To ensure better security various cryptographic mechanisms like pseudo-random number generators, digital signatures use the concepts of both symmetric-key and asymmetric-key system.

2.4 Other cryptographic primitives

In this section we discuss about two algorithms; hash functions and digital signatures. In cryptography these are used to ensure data integrity and authentication. Hash functions are considered as one of the encryption algorithms like secret key and public key algorithms and known as one-way encryption systems where no secret keys are used. Whereas, digital signatures use concepts of public key cryptosystems and ensure non-repudiation by using hash functions.

2.4.1 Hash functions

Cryptographic hash functions takes a block of message as an input and returns random bit string of fixed length known as *hash value*. The hash value will change if the input message is altered intentionally or accidentally.

Cryptographic hash function properties

1. For any given message, it is easy to compute hash value.
2. For any given hash value, it is infeasible to find its corresponding message.
3. For any two messages, their hash values need not be the same.

In cryptography, hash functions are used to provide the *integrity* of a message (m). if the hash value of m is stored in a secure place then the modification of m can be detected by calculating the hash value and comparing it with the stored value. Hence, the hash functions are also known as *modification detection codes* (MDCs) [23].

In addition, hash functions provide message *authentication* and it authenticate the origin of the message. If hash functions are used in message authentication it is called as message authentication code (MAC).

2.4.2 Digital signatures

The process of *signing* to impose the transformation of message and the secret key into a tag is called as *digital signature*. The purpose of *Digital signatures* is to provide a reason to the receiver to trust or identify the message which is sent by the known sender only.

Digital signatures uses the concept of *public-key cryptosystem*. Usually, digital signatures depend on the secret key of the signer and can be generated only by the signer. In general, digital signature consists of two algorithms they are [23]:

1. *Sign algorithm*: It takes a message and a secret key as an input and generates the signature.
2. *Verify algorithm*: For the given message, public-key and signature, it verifies whether the message is authentic or not.

Digital signatures are intended to provide both *data integrity*, *authentication* and *non-repudiation* of the message.

Example: If Alice wants to sign a message (m), by using the *Sign* algorithm with secret key (sk) and generates the signature as $s = Sign(sk, m)$. When Bob receives the signature (s) for the message m, he checks the signature is $valid$ or not by applying the *verify* algorithm by using Alice's public key (pk) as $Verify(pk, s, m) = valid$.

From the above example, we can say that Bob can verify the Alice signature and the message. It is possible only when Bob uses the correct Alice's public key and the message.

On the other hand, there are possibilities that adversary can inject the message into the communication channel and pretend as Alice or Bob. It is called as a *forgery* or possible type of *attacks* (see section 2.7) on the transit information. In that case nobody can identify the forgery of the message, even Bob too. To overcome these type of forgeries or attacks, digital signatures uses cryptographic *hash functions*. Here, Alice first applies the hash function to the message to get the hash value and than she encrypts the hash value using her private key. This encrypted hash value is used as Alice signature and sends the digital signed message to the Bob. Now, Bob knows the message and the digital signature. Now, Bob applies the hash function on the message to get the hash value and decrypts the hash value using Alice public-key. Finally, Bob verifies whether the two hash values are *valid* or *not*; if they are valid then Bob confirms that the digital signature message is sent by Alice.

Thus, by using the hash functions in digital signatures the *non-repudiation* can be ensured. It suggests that now Alice cannot state that the message is not signed by her.

2.5 Cryptographic protocols

A *Cryptographic protocol* can be defined as an algorithm with well defined sequence of steps which must be followed by two or more communicating parties, to solve the issues involving confidentiality, authenticity and data-integrity.

In modern cryptography, the development of well defined protocols lead us to provide high level of security. Cryptographic protocols are developed based on *cryptographic primitives* such as hash functions, symmetric and asymmetric algorithms. Cryptographic protocols are called as *multi-party algorithms*. Meaning, it must be communicated between at least two parties.

2.5.1 Key-exchange and entity authentication

The major draw back of symmetric-key cryptosystem is the key exchange or key agreement. This issue becomes more evident especially when, used in a cluster of network systems where more than two entities communicate. It is know that, in symmetric-key systems both Alice and bob agrees on a secret key. To share the secret key securely symmetric-key system uses the concept of public-key cryptosystems. The first solution available for the key exchange problem is the *Diffie-Hellman key agreement* protocol [21] which is based on public-key cryptosystems.

The draw back of continuous key exchange using Diffie-Hellman key agreement is that, there is no authentication between the communicating parties. Whereas, entity authentication ensures the identity of the communicating parties there by avoiding impersonation. In entity authentication, Alice can prove her identity by signing her signature on the messages. There are possibilities that adversary can intercepts Alice signed message and later on adversary can authenticate as Alice. This type of attack is called as *replay attack*. One solution to prevent from replay attack is to vary the Alice signature in [23] .

One of the methods used to prevent replay attack is the *challenge-response protocol*. This protocol is based on public-key digital signature scheme. In challenge-response protocol, first Bob sends any random number to Alice and she uses the same random number in the message and signs it with her signature before sending back to Bob. Now Bob verifies the random number which he has sent earlier to Alice and later on Bob validates her signature. Here, the random number is viewed as a challenge. Other than this protocol, there are other protocols like *two-way* and *three-way* authentication protocols [23]. These authentication protocols play a major role in some of the technologies for instance in RFID technology, where the RFID reader and tag must authenticate each other for communications. How cryptographic authentication protocols are used in RFID will be discussed in chapter 6.

2.5.2 Identification

Identification is defined as a technique which provides the identities of both the entities to verify each other which are involved in a communication.

One of the techniques used earlier is personal identification number (PIN) and password method. If Alice wants to do electronic transactions with the bank, first she has to enter her PIN and a password into the system to view her account details. But the passwords and PIN are not secure. Since there are chances that if anyone come to know her password or PIN during her transaction, it might impersonate as Alice and extract all her banking details.

There are protocols designed to eradicate these type of issues. For example, fiat-shamir identification protocol, zero-knowledge [23]. Thus, cryptographic protocols play a major role in ensuring the confidentiality, data integrity, authentication and non-repudiation. The key-agreement generally should take place between two parties (sender and receiver) in the field of communication. The establishment of keying methods between the parties ensures the confidentiality of secret keys so that, once the secret keys are agreed, any messages can be encrypted securely. Once the keys are agreed, it also ensures the authenticity by using public or private key agreement, this can be achieved using the concepts of digital signatures or hash functions. Now the receiver can verify the senders identity and ensures the non-repudiation. By using public keying methods one can provide the data-integrity, that is if the senders key is signed by any trusted third party then the receiver can consider that key with confidence.

2.6 Pseudo-randomness

The concept of randomness is used for the better encryption purpose. Most of the encryption algorithms prefer the random key selection so that, keys cannot be predicted by an adversary. In general, pseudo-random sequence bits are generated by an algorithm known as *pseudo-random bit generators*. The output of these generators is a long sequence of pseudo-random sequence of bits for any given small random input which is known as the initial state. The basic building blocks of pseudo-random generators are the Linear feedback shift register (LFSR) [24].

2.7 Attacks

The main objective of cryptography is to provide security for the transmitted message from the adversary, who tries to extract the information from the transmitted message. Before designing any cryptographic system one should assume that, the adversary knows everything about cryptographic system implementation and the algorithms used. This principle is commonly stated as the *kerkhoff's principle*.

Adversaries try to extract information about the secret key so that, ciphertext can be broken to recover the original message. Attacks by the adversaries can be classified into two types, *passive* and *active*.

In *passive* type of attacks, the adversary observes only the communication channel and it is a threat to the confidentiality of the data.

In *active* type of attacks, the adversary can modify, delete or inject the data into the communication channel and it is threat to the data integrity, authenticity and confidentiality.

Attacks can be divided into following types based on the resources used by the adversary. The objective of these types of attacks is to recover the plaintext from the ciphertext or to extract the information about the secret key.

1. *Ciphertext-only attack:* In this type of attack the adversary has the knowledge of ciphertext and tries to extract the plaintext or the decryption key from the known ciphertext. Any encryption scheme is considered insecure if it cannot resist this type of attack.

2. *Known-plaintext attack:* Here the adversary uses knowledge of plaintext and its corresponding ciphertext and tries to extract the secret key and recover the encrypted message.

3. *Chosen-ciphertext attack:* The adversary selects the ciphertext and tries to decrypt its corresponding plaintext. Assuming that the attacker has access to the decryption system.

4. *Chosen-plaintext attack:* In this type of attack the adversary first chooses the plaintext and obtain its ciphertext. After few analysis on how to obtain the plaintext from the unknown ciphertext, the adversary tries to analyze the cryptographic pattern outputs that are being used and try to recover the secret key.

5. *Man-in-the-middle attack:* Here the adversary gets access to the communication channel used by the sender and receiver and tries to send some non-specific key exchange informations from the middle of communication.

6. *Replay attack:* The adversary retains the capability to store or record the communication of the sender and receiver. Later on the adversary replays the same communication after some point of time.

7. *Relay attacks:* It is similar to man-in-the-attack and replay attack. In this the adversary relays the message from the sender and sends it to the receiver. Relay attacks are commonly occur in wireless devices such as RFID devices.

8. *Side-channel attacks:* In side channel attacks, the attacker tries to get secret information of the cryptographic systems which are implemented (physically) hardware.

    (a) *Timing analysis attacks:* In timing attacks, the attacker tries to analyze the execution time taken by the cipher for encryption or decryption operations. Basically, it is based on measuring the time taken by a unit to perform its operation and the measuring information can reveal the secret key.

(b) *Power analysis attacks:* In this type of attacks, the attacker tries to get information from the power consumptions of the devices.

Chapter 3

Mathematical Background

3.1 Introduction

Finite Fields play a major role in some of the most interesting applications of modern algebra to the real world. In particular, the applications related to the data communication is a vital concern in our information friendly society. In today's technological advancements in the areas of space and satellite communications, protecting the privacy of information involve the use of finite fields in one way or the another.

This chapter begins with a brief overview of concepts of field and group. Next, we will explain the concepts and types of finite fields of the form GF(p), where p is a prime number. Before going to the details of finite field extension of the form GF (p^n), where n is a positive integer, it needs to be discussed some of the elementary background in polynomial arithmetic operations. Finally, we briefly discuss about the normal basis and types of optimal normal basis.

3.1.1 Modular Arithmetic

Modular arithmetic has gained importance in the area of cryptography. In Public Key Cryptosystem algorithms such as RSA and Diffie-Hellman algorithm uses the theory of modular arithmetic including, symmetric key algorithms such as AES, IDEA and RC4. The major advantage of using modular arithmetic is that it allows us to do faster multiplication operations. For example, in any complex operations such as polynomial greatest common divisor calculation where we come across large number of integers to perform number of multiplication operations. The use of modular arithmetic reduces the computing times of these large operations. In one of the applications like error correcting codes each digit of the code is related to the elements of the finite field by using the modular arithmetic theory.

Note that, the modular operator (mod n) maps all the integers within the confined set of integers $\{0, 1, 2, \ ... \ (n-1)\}$ and all the arithmetic operations are performed within this set. This techniques is called as **modular arithmetic**.

The set of integers and nonzero integers of mod n are denoted by $\mathbb{Z}_n$ and $\mathbb{Z}_n^*$ respectively.

Example 1: Modular addition and multiplication over modulo 23.

Suppose, $12 + 20 = (12 + 20) \ mod \ 23 = 32 \ mod \ 23 = 9$

since the remainder is 9 when 32 is divided by 23.

Similarly, in multiplication operation; $8 \ \times \ 9 = 72 \ mod \ 23 = 3$, since the remainder is 3 when 72 is divided by 23.

3.2 Groups and Fields

Fields and groups are the well known algebraic structures of the abstract or modern algebra. In abstract algebra, we work with sets on which elements can be operated algebraically. For instance, we can say that by combining two elements of a set in several different ways the third element of the set can be obtained. All these operations will follow certain specific rules which will define the nature of the set. The notation followed for operations on set of elements is usually same as the notation for ordinary addition and multiplication.

Groups :

Definition 1 *A **group** (G) is defined as a pair $(S, \ \bullet)$, where S is set of elements with binary operation $\bullet$, such that, it obeys axioms from A1 - A4.*

The binary operator $\bullet$ is generic and can be referred to addition, multiplication and other mathematical operation.

Note: Here the set S is the representation of group G. From now on we represent group as G.

Axioms	***Meaning***
A1. Closure	For all a, b in G, $a \bullet b$ will be in G
A2. Associative	For all a, b, c in G, $a \bullet (b \bullet c)$ = $(a \bullet b) \bullet c$.
A3. Identity	There exists an element e in G, for all a in G, such that $a \ \bullet \ e$ = $e \ \bullet \ a$ = a. More formally, $\exists \ e \in G$, $\forall \ a \in G, \ a \ \bullet \ e = e \ \bullet \ a = a$

Note: We denote identity element in G as **i**

A4. Inverse	For every a in G, there exists an element x in G, such that $a \bullet x = x \bullet a = \mathbf{i}$. More formally, $\forall\, a \in G, \exists\, x \in G, a \bullet x = x \bullet a = \mathbf{i}$

In order to satisfy the A4 axiom the operation must have an identity element.

Example 2: For any positive integer n ($\mathbb{Z}_n$, **+**) is a group.

The set $\mathbb{Z}_5$ = $\{0, 1, 2, 3, 4\}$, where $n = 5$, with addition module 5 forms a group, for example $2 + 4 = 1\ (mod\ 5)$.

Example 3: If p is prime then $(\mathbb{Z}_p^*,\ \times)$ is a group.

The set $\mathbb{Z}_5^*$ = $\{1, 2, 3, 4\}$, where $p = 5$, with multiplication module 5 forms a group, for example $2\ \times\ 4 = 3\ (mod\ 5)$.

Note: If a group has finite number of elements then it is called as ***finite group***, otherwise, it is an ***infinite group***. The ***order*** of the group is the number of elements in the group.

Definition 2 *A group is called an* ***abelian group*** *if it satisfies the following axiom.*

Axioms	***Meaning***
A5. Commutativity	For all a, b, in G, $a \bullet b = b \bullet a$

Definition 3 *A group is called* ***cyclic*** *if there are one or more members that can be used to generate all members by raising the* ***generator*** *to a power. More formally:* $\exists\ g \in G,\ \forall\ a \in G,\ \exists\ k,\ a = g^k$

A cyclic group is always abelian.

Example 4: If p is prime then $(\mathbb{Z}_p^*, \times)$ is a cyclic group.

The Cyclic group $(\mathbb{Z}_7^*,\ \times)$, where $p = 7$, the order of the finite group is 6. And the elements 3 and 5 are the generators of this cyclic group $\mathbb{Z}_7^* = \{1, 2, 3, 4, 5, 6\}$, then the powers of 3 module 7 are

$1 = 3^6$, $2 = 3^2$, $3 = 3^1$, $4 = 3^4$, $5 = 3^5$, $6 = 3^3$

Fields :

Definition 4 *A* ***field*** *F, is defined as a set of elements with two binary operations* $+,\ \times$*, which is represented as* $(F,\ +,\ \times)$*. Such that it obeys the following axioms.*

Axioms	***Meaning***
(A1 - A5)	F forms a abilian group with respect to addition
(A1 - A3) and A5	F with respect to multiplication satisfies the axioms.

Note: The additive and multiplicative identity elements in F is denoted by 0 and 1 respectively.

A6. Multiplicative inverse	For every a in F, except $a = 0$, there exists an element x in F such that $a \times x$ = $x \times a$ = 1. More formally, $\forall\ a \neq 0 \in F$, $\exists\ x \in F$, $a \times x = x \times a = 1$

Example 3: For any field $(F,\ +,\ \times)$. $(F,\ +)$ forms an abilian group.

Example 4: For any field $(F,\ +,\ \times)$. $(F^*,\ \times)$ forms an abilian group, where F^* is the elements of F excluding additive identity element 0.

3.3 Finite Fields:

Finite fields play a major role in the area of cryptography. Most of the cryptographic algorithms such as the Digital Signature Standard (DSS), the El Gamal public key encryption, elliptic curve public key cryptography are heavily depend on the properties of finite fields and it is also used in Advanced Encryption Standard (AES) cryptography.

The order of the finite field must be a power of prime p^n, where n is a positive integer. Here two cases exits; for $n = 1$, the finite field is of the form GF(p) where GF stands for Galois Field and for $n > 1$, the finite field is of the form GF(p^n). The finite field GF(p) has different structure compared to the finite field GF(p^n).

Types of Finite Fields:

(i) ***Prime field:*** It is defined as a field of the form GF(p) of order p, where p is prime. All the elements in this field and arithmetic operations (+, $\times$) perform with respect to the modulo p.

(ii) ***Binary field:*** It is defined as a field of the form GF(p^n) of order p^n, where n is a positive integer. Usually binary field is constructed using the prime field.

3.3.1 Finite Field of the form GF(p) :

For any prime p, finite field of order p, the elements of GF(p) is defined as the set $\{0, 1, 2, ...(p-1)\}$, along with the arithmetic operations modulo p. GF(p) can also be denoted by the set of integers $\mathbb{Z}_p$.

Example 4: The arithmetic operations in the simplest finite field of the form GF(p).

The simplest finite field is GF(2), where $p = 2$, and its elements are $\{0, 1\}$, this is a special case where the arithmetic operations $+$ and $\times$ is just equivalent to the XOR and AND operations respectively.

The additive inverse elements for 0 and 1 is 0 and 1 respectively.

The multiplicative inverse elements for 0 and 1 is **-**(do not exist) and 1 respectively.

Example 5: The arithmetic operations in the finite field of the form GF(3).

The finite field of GF(3), where $p = 3$, and its elements are $\{0, 1, 2\}$, the arithmetic operations $+$ and $\times$ is just as simple operations on integers followed by a reduction modulo p. For instance, in GF(3), $2+2 = 4$ which is reduced to 1 modulo 3. Similarly, $2 \times 2 = 4$ reduced to 1 module 3.

The additive inverse elements for 0, 1 and 2 is 0, 2 and 1 respectively.

The multiplicative inverse elements for 0, 1 and 2 is **-**(do not exist), 1 and 2 respectively.

Note : Before going to the finite field of the form GF(p^n), we need to discuss about the concepts of polynomials. Since they are extensively used in finite fields GF(p^n).

3.3.2 Polynomial Arithmetic:

The elements of GF(p^n), when $n > 1$, can be represented as polynomials, whose cofficients belong to GF(p) and degree should be less than n. When p is 2, the elements of GF(p^n) is represented as binary numbers $\{0, 1\}$. This means that each term in a polynomial expression is represented by one bit in the corresponding binary expression.

We are interested in polynomials over fields. From now on the field will be denote as $(F, +, \times)$ as F.

Definition 5 *A* ***polynomial*** *is defined as a mathematical expression involving sum of powers in one or more variables multiplied by their cofficients (constants). A polynomial with one variable and their constant cofficients is represented by*

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_2 x^2 + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i$$

where, n (integer $n \geq 0$) is called the degree of the polynomial, where the cofficients a_i, $0 \leq i \leq n$, are elements of F and x is the symbol which is not belonging to F referred to as indeterminate (symbol which does not stand for anything). F is also called cofficient set, when $a_n \neq 0$ and such polynomials are known to be defined over F. It is possible to have same powers of x when we compare two polynomials $f(x)$ and $g(x)$ over F.

$$\text{Let } f(x) \ = \sum_{i=0}^{n} a_i \ x^i \text{ and } g(x) \ = \sum_{i=0}^{m} b_i \ x^i$$

A. Addition of $f(x)$ and $g(x)$ is defined by:

$$f(x) + g(x) = \sum_{i=0}^{n} (a_i + b_i) \ x^i, \text{ for n = m}$$

B. Multiplication of $f(x)$ and $g(x)$ is defined by:

$$f(x) \times g(x) = \left(\sum_{i=0}^{n} a_i \ x^i \right) \times \left(\sum_{i=0}^{m} b_i \ x^i \right) = \sum_{k=0}^{n+m} c_k \ x^k$$

where $c_k = \sum_{0 \leq i < k} a_i \ b_{k-i} = a_0 b_k + a_1 b_{k-1} + ... + a_{k-1} b_1 + a_k b_0$.

Note that the degree of the product is the sum of the degrees of the two polynomials.

Example 6: Let $f(x) = x^3 + x^2 + 1$ and $g(x) = x^3 + x$, calculate $f(x) \times g(x)$

$$\begin{aligned} x^3 \times (x^3 + x^2 + 1) &= x^6 + x^5 \qquad\quad + x^3 \\ x \times (x^3 + x^2 + 1) &= \qquad\qquad\quad x^4 + x^3 + x \\ \hline &= \ x^6 + x^5 + x^4 \qquad + x \end{aligned}$$

C. Division of $m(x)$ and $p(x)$ is defined by:

Polynomial division over Galois field can be computed using the rules of multiplication and addition. The division of two polynomials $m(x) \div p(x)$ is defined as $m(x) = q(x) \times p(x) + r(x)$, where quotient $q(x)$ and reminder $r(x)$ are results of the division operation.

Example 7: Let $m(x) = x^5 + x^3 + x^2$ and $p(x) = x^3 + x$, calculate $m(x) \div p(x)$.

$$
\begin{array}{ll}
 & \qquad\qquad x^2 \\
\cline{2-2}
x^3 + x \;\; | & x^5 + 0x^4 + x^3 + x^2 + 0x^1 + 0x^0 \\
 & x^5 \qquad\quad + x^3 \\
\cline{2-2}
 & \qquad\qquad\quad x^2
\end{array}
$$

Therefore $q(x)$ is x^2 and $r(x)$ is x^2.

Verify the above results using the definition :

$$
\begin{aligned}
m(x) &= q(x) \times p(x) + r(x) \\
&= (x^2) \times (x^3 + x) + x^2 \\
&= x^5 + x^3 + x^2
\end{aligned}
$$

Note: Polynomials over F are important in constructing the structure of linear feedback shift register sequences (See Section 3.7).

3.3.3 Finite Field of the form GF(p^n) :

In this section we are interested in constructing the finite field of the form GF(p^n). In the previous section we saw finite field of the form GF(p) of order p, where p is prime. The elements of GF(p) $= \mathbb{Z}_p = \{0, 1, 2,\ ...\ (p-1)\}$, under arithmetic operations $(+, \times)$ are performed with respect to modulo p operation. We use the similar concept to construct the finite field of the form GF(p^n), containing $q - 1$ elements, where $(q = p^n)$ under the reduced module $p^n - 1$ operation.

Fact: The set GF(p^n) along with two arithmetic operations $(+, \times)$ forms a finite field and the order of the field is p^n.

Note: We denote non zero elements of GF(p^n) as GF(p^n)* .

Irreducible Polynomial over Field :

Definition 6 *A polynomial $i(x)$ is known as an **irreducible polynomial** over a field F, if and only if $i(x)$ cannot be derived as a product of two or more polynomials over field F. With analogy to integers, an irreducible polynomial is also called a **prime polynomial**.*

Fact: For every prime p and for every degree $n > 1$, there exists at least one irreducible polynomial of degree n over GF(p).

Example 8: Consider the irreducible polynomial $i(x) = x^3 + x + 1$ over the field $GF(2)$.

The possible factors of $i(x)$ with degree less than $i(x)$ are x and $x + 1$. The product of these two possibilities will not be equal to $x^3 + x + 1$. Therefore, the given $i(x)$ is irreducible polynomial over $GF(2)$.

Primitive Element and Primitive Polynomial :

Definition 7 *A **primitive element** of GF(p^n) is an element which is a generator of a cyclic group GF(p^n)*. An irreducible polynomial over GF(p) having zero as a primitive element in GF(p^n) is called a **primitive polynomial** over GF(p).*

Note: Not all irreducible polynomials are primitive.

Root of Polynomial :

Definition 8 *If $p(x)$ is a polynomial over F, then the element $\alpha \in F$ such that $p(\alpha) = 0$ is called a root (or zero) of the polynomial $p(x)$.*

Subfield :

Definition 9 *If a subset S whose elements are from field F which satisfies the field axioms along with the arithmetic operations of F, then S is called a **subfield** of F.*

GF(p^m) is a subfield of GF(p^n), if and only if m is a positive divisor of n, i.e., GF(p^m) $\subset$ GF(p^n).

Example 9: GF(2^2) $\subset$ GF(2^4) and GF (2^4) $\subset$ GF(2^{12})

Extension Field :

Definition 10 *A field F is known as an **extension field**, if S is a subset lying under the set F with respect to the field operations then by definition S is a subfield of F and F is an extension field of S. Which is denoted by F/S and read as " F over S ".*

If F is an extension field of field L, which in turn an extension of K, then L is an **intermediate field** of the field extension F/S.

To construct (extension field) a finite field of the form (GF(p^n), +, ×), of order p^n, where n is a positive integer, by selecting an irreducible polynomial as $f(x)$ of degree n over GF(p). Let α be a root of the $f(x)$, which satisfies $f(x)$ as $f(\alpha) = 0$. Then GF(p^n) is defined as [25]

GF(p^n) $= \{a_0 \ + \ a_1 \ \alpha \ + ... + \ a_{n-1} \ \alpha^{n-1} \mid a_i \in GF(p)\}$

Note: Here one should know that the addition operation is done by module GF(p) whereas, the multiplication is done by modulo of irreducible polynomial.

Example 11: To construct a finite field of the form GF(2^3) with irreducible polynomial $f(x) \ = \ x^3 + x + 1$ over the field GF(2).

Let α be a root of $f(x)$, i.e. $f(\alpha) = \ 0$. The finite field GF(2^3) is defined as $\{a_0 \ + \ a_1 \ \alpha \ + \ a_2 \ \alpha^2 \mid a_i \in GF(2)\}$

All the elements of GF(2^3) can be derived from the given equation $f(x) \ = \ f(\alpha) \ = \ \alpha^3 + \alpha + 1 \ = \ 0$, with respect to the modular 7 operation.

The elements of $GF(2^3)$ are $0, \ 1, \ \alpha, \ \alpha^2$ and from α^3 they can be derived as $\alpha^3 = \alpha \ + \ 1$, and α^4 can be written as $= (\alpha^3)(\alpha) = (\alpha + 1)(\alpha) = \alpha^2 + \alpha$. Similarly, we can derive rest of the elements.

For the given irreducible polynomial of degree 3, table given below shows 3-tuple binary, polynomial and exponential notations.

Binary Notation	**Polynomial Notation**	**Exponential notation**
000	0	$0 = \alpha^\infty$
100	1	$1 = \alpha^0$
010	α	α
001	α^2	α^2
110	$1 + \alpha$	α^3
011	$\alpha + \alpha^2$	α^4
111	$1 + \alpha + \alpha^2$	α^5
101	$1 + \alpha^2$	α^6
	$\alpha^7 = 1$	

Here the elements of GF(2^3) are represented in polynomial notation (basis and is explained in section 3.4) and exponential notation which represents the elements in GF(2^3). Also note that, GF(2^3)* i.e. the non zero elements of GF(2^3) form a cyclic group of order 7 with generator α, where $\alpha^7 = 1$.

The polynomial $f(x)$ and α is called the defining polynomial and defining element respectively. We can say that GF(2^3) is the ***extension field*** of GF(2).

For instance, addition of two elements $(1 + \alpha + \alpha^2)$ and $(1 + \alpha)$ is $(1 + \alpha + \alpha^2) + (1 + \alpha^2)$ $= \alpha$.

For multiplication, it can be written in a simpler way as $(1 + \alpha + \alpha^2) = \alpha^5$ and $(1 + \alpha^2)$ $= \alpha^6$. Therefore $(1 + \alpha + \alpha^2)$ $(1 + \alpha^2) = \alpha^5\alpha^6 = \alpha^{11} = \alpha^4 = \alpha + \alpha^2$

Trace Function :

Definition 11 *Suppose, F = GF(p^n) and K = GF(p), then the* ***trace function*** $Tr_{F/K}(x)$ *is defined by*

$$Tr(x) = Tr_{F/K}(x) = x + x^p + ... + x^{p^{n-1}} = \sum_{i=0}^{n-1} x^{p^{i+1}},\ x \in F$$

The trace function $Tr(x)$ converts GF(p^n) → GF(p)

Example 11: For the finite field GF(2^3) which is defined by $\alpha^3 + \alpha + 1$. Compute Tr(α) and Tr(α^3).

From the trace function $Tr(x)$, when $p = 2$ and $n = 3$, we can extract Tr(α) $= \alpha + \alpha^2 + \alpha^4$ $= \alpha + \alpha^2 + \alpha + \alpha^2 = 0$, Similarly for
Tr(α^3) $= \alpha^3 + \alpha^6 + \alpha^5 = (1 + \alpha) + (1 + \alpha^2) + (1 + \alpha + \alpha^2)$ $= 1$

3.4 Basis

Basis can be defined based on the circumstances being used. For instance, an element in a finite field can be represented in the form of a basis. Since, we are interested in finite fields where two types of basis exists; ***polynomial basis*** and ***normal basis***. Thus an element in a finite field can be represented either in polynomial [26] [27] or normal basis [28] [29] [30].

Polynomial Basis :

Definition 12 *Consider the finite field as GF (p^n) and let $\alpha \in$ GF (p^n) be the root of an irreducible polynomial of degree n over GF(p). Then the* ***Polynomial basis*** *is represented as* $\{1, \alpha, \alpha^2 ... \alpha^{n-1}\}$ *of GF (p^n) over GF (p).*

Where α is called a primitive element of GF(p^n).

Example 12: If p = 3 and n=2. then GF(3^2) is a simple extension field of of GF(3) of degree 2. Let $\alpha \in$ GF(3^2), be a root of the irreducible polynomial x^2+1 over GF(3), then the polynomial basis is $\{1, \alpha\}$ of GF(3^2) over GF(3).

Normal Basis :

Definition 13 *For any positive integer n in GF (p^n), there will be always a normal basis for the finite field GF(p^n) over GF(p). If $\gamma \in$ GF (p^n) be a normal element, then the* ***normal basis*** *is represented as $\{\gamma, \gamma^{2^1}, \alpha^{2^2}$ ... $\gamma^{2^{n-1}}\}$.[31].*

where γ is called a generator or normal element of GF(p^n) over GF(p). Which is is represented in the form of $n \times m$ matrix and denoted by M.

Example 13: If p = 2 and n=3. then GF(2^3) is a simple extension field of GF(2) of degree 3. Let $\alpha \in$ GF(2^3), be a root of the irreducible polynomial $x^3 + x^2 + 1$ over GF(2), then the normal basis is $\{\alpha,\ \alpha^2,\ 1 + \alpha + \alpha^2\}$ of GF(2^3) over GF(2).

3.4.1 Selection of basis for hardware implementation:

In this section we briefly discuss about the selection of basis for the finite field arithmetic operations especially for hardware implementation. We know that finite fields play a vital role in cryptography and especially in symmetric and asymmetric key cryptosystems which involve finite field arithmetic operations.

Usually arithmetic operations over GF(2^n) are performed under reduced modulo of the irreducible polynomial $f(x)$ over GF(2). Arithmetic addition and subtraction operations are performed under modulo 2. Addition of two polynomials is nothing but the bit-wise exclusive-or (XOR) operation of their binary representation whereas multiplication operation is the complex and time consuming over the field GF(2^n). Complexity is based on the selection of irreducible polynomial and the basis used to represent the finite field elements. Besides that there are past publications [32] [33], where they talk about the efficient implementation of finite field arithmetic.

Polynomial basis is considered for hardware optimizations, because in polynomial basis the the multiplication operation can be implemented using simple shift and XOR operations [34]. In normal basis hardware implementation for squaring an element is simply a right cyclic shift of its coordinates [31]. Therefore, squaring in normal basis is simple and cost effective. But, multiplication in normal basis is more complex. Complex in terms of utilization of hardware resources such as number of logic gates, area and so on. To overcome this in normal basis, in [35] they introduced the optimal normal basis.

Optimal normal basis holds an upper hand in efficient hardware resources utilization [28] [29], such as number of logic gates, XOR and AND gates compared to normal basis. In section 5, we have seen two types of optimal normal basis, type I and type II. Type I basis are efficient in hardware implementation over GF(2^n), but the drawback is that they are not suitable to some of the cryptographic areas since n is even [31] whereas type II is suitable since n can be odd. Since then number of researchers started to introduce efficient implementation of multipliers using type II [31] [33] [28] [29].

3.5 LFSR and mathematical description:

Linear feedback shift register (LFSR) have been widely used in keystream generators in stream ciphers, random number generators in most of the cryptographic algorithms. In this section we discuss about LFSR definitions and their sequence representation. Each of the square blocks in figure 3.1, is a 2 state (0 or 1) storage units. The n binary storage units are called as the stages of the shift register and their contents are in the form of n bits in length, which is called as the internal state of the shift register.

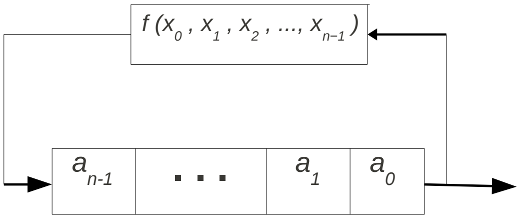


Figure 3.1: Block diagram of LFSR

Let $(a_0, a_1, a_2, ..., a_{n-1}) \in GF(2^n)$ be the initial state of the LFSR and $f(x_0, x_1, x_2, ..., x_{n-1})$ be the feedback function or feedback polynomial, as shown in figure 3.1. If the feedback function is a linear function then it can be expressed as

$$f(x_0, x_1, x_2, ..., x_{n-1}) = c_0x_0 + c_1x_1 + c_2x_2 + ... + c_{n-1}x_{n-1},\ c_i \in GF(2) \tag{3.1}$$

After each consecutive clock pluses the LFSR will generate a output binary sequence $\underline{b}$ of the form $\underline{b} =\ a_0, a_1, ...$

The output sequence of the LFSR satisfies the following recursive relation [24]

$$a_{k+n} \ = \sum_{i=0}^{n-1} c_i \ a_{k+i}, \ \ k = 0, 1, ... \tag{3.2}$$

The output of the LFSR is considered as a linear recursive sequence. If the feedback function is linear then the output sequence is called LFSR sequence. Otherwise it is called as the *nonlinear feedback shift register* (NLFSR) sequence.

Notes : Let $\underline{b}$ is the binary sequence. In general the linear feedback function equation (3.2) is represented in polynomial of the form $f(x) = x_n + c_{n-1}x_{n-1} + ... + c_1x + c_0$, which is known as the *characteristic polynomial* of the LFSR.

Example 1: Consider a 3-stage LFSR as shown in figure 3.2 with the linear feedback function $f(x_0, x_1, x_2) \ = \ x_0 + x_1$ with initial state as $(1, \ 0, \ 0)$ equivalent to $(a_0, \ a_1, \ a_2)$. The output sequence would be $10010111001011.....$ which is repeated periodically with a period of 7.

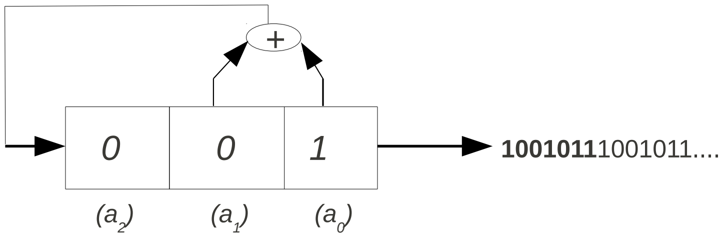


Figure 3.2: 3-stage LFSR

3.5.1 Different types of sequences:

1. ***Binary Sequence:*** $\underline{b} = b_i, \ b_i \in GF(2)$, is a binary sequence over GF(2).

2. ***M-Sequence:*** The output sequence generated by an n-stage LFSR with non zero initial values and has maximal period $2^n - 1$ is know as the maximal length sequence or in short called as m – sequence [25].

3. ***De-Bruijn sequence:*** De-Bruijn sequence is the output of an n-stage NLFSR having period 2^n and satisfies the n-tuple occurrence exactly once in each period. From any m - sequence with period as $2^n - 1$ we can obtain the de-Bruijin sequence by inserting a 0 into the run of $n - 1$ consecutive zeros of the m - sequence [25].

Example 2: Consider a 4-stage LFSR as shown in figure 3.3 with the characteristic polynomial $f(x) = x^4 + x + 1$ and with the initial state as $(0,0,0,1)$ equivalent to $(a_0, a_1, a_2, a_3, a_4)$. The output is a m - sequence is 0001001101011110001 00..... which is repeated periodically with a period of 15.

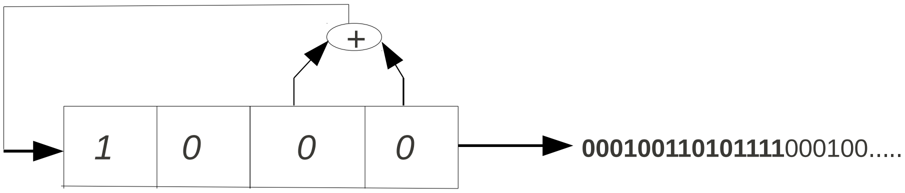


Figure 3.3: 4-stage LFSR

3.5.2 Advantages of LFSR Properties:

Since LFSR's are widely used in keystream generators. To consider LFSR based keystream generators as cryptographically secure, the design of LFSR should have the following desirable properties [21].

1. Large Period.
2. Large Linear complexity.
3. Good statistical properties.

Notes: If $\underline{b}$ is a binary sequence then the linear complexity of $\underline{b}$ is the shortest length of the LFSR that generates $\underline{b}$. It is denoted by LS($\underline{b}$). For any given $\underline{b}$ of length N, one can compute the linear span of the sequence using the *Berlekamp-Massey algorithm*.

3.5.3 Hardware implementation of LFSR over Galios fields:

In hardware implementation the LFSR contains N registers connected together to form a shift register. Generally, shift register is a sequence of flip flops in which the output of the last flip flop is connected (feedback) to the earlier flip flops by an XOR gate as shown in figure 4. Suppose the length of the LFSR is N than it consists of N – stages of flip-flops and the stored bits are controlled by a single clock. At each clock pulse, the bits in the storage elements is shifted by one position (right position) to the next stage, that is, there is a transition from one state to next.

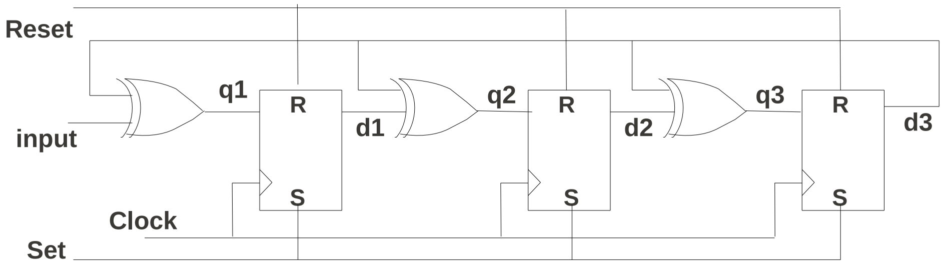


Figure 3.4: 3-bit LFSR circuit

Some of the design parameters when designing LFSR are the number of flip flops, external or internal XOR gates, feedback taps (inputs fed to the XOR) and reset signal. On reset the register is set to all 1's and for analysis purpose here we use LFSRs with internal XOR gates since their circuitry are matched with the polynomials over the Galois fields.

The LFSR generates a maximum bit sequence of length $2^n - 1$ combinations, where n is the size of the finite field. The feedback taps on the LSFR is selected based on the chosen polynomial over the finite field. Polynomial arithmetic operations are carried out with respect to the $mod\ 2$ operations i.e. the coefficients of the polynomial must be either 1's or 0's. These polynomials are called feedback or characteristic polynomials. The LFSR bit sequence can be represented by the characteristic polynomial. If the bit sequence is 110011 then the characteristic polynomial is denoted as $x^5 + x^4 + x^1 + 1$.

Example 3: Represent the LFSR for the characteristic polynomial $p(x) = x^5 + x^4 + x^2 + x + 1$.

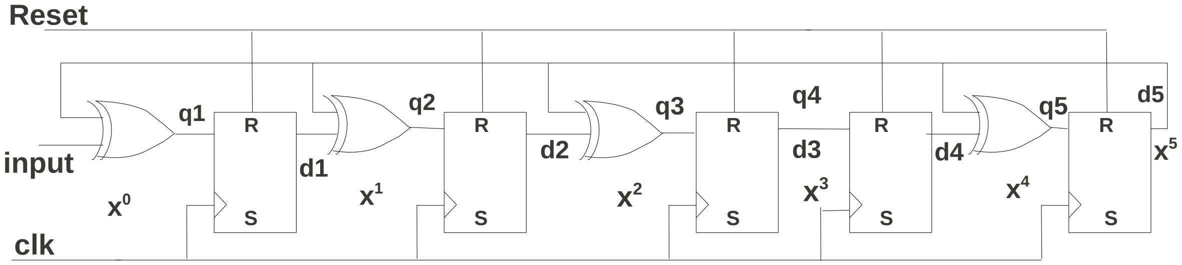


Figure 3.5: LFSR implementing characteristic polynomial

From the above example exponents of polynomial are represented as: x^0 is the input to the LFSR, x^1 is the output of the first flip flop and x^2 is the output of the second and so on. Moreover the maximum exponent of the polynomial represents the number of flip flops used in the LFSR.

other exponents denote the flip flops with or without tap connections to the feedback line from the last flip flop.

3.5.4 Polynomial multiplication and division in LFSR:

Using the example 6 from section 3.3.2, the multiplication $f(x) \times g(x) = (x^3+x^2+1) \times (x^3+x)$ circuit using LFSR is shown in figure below.

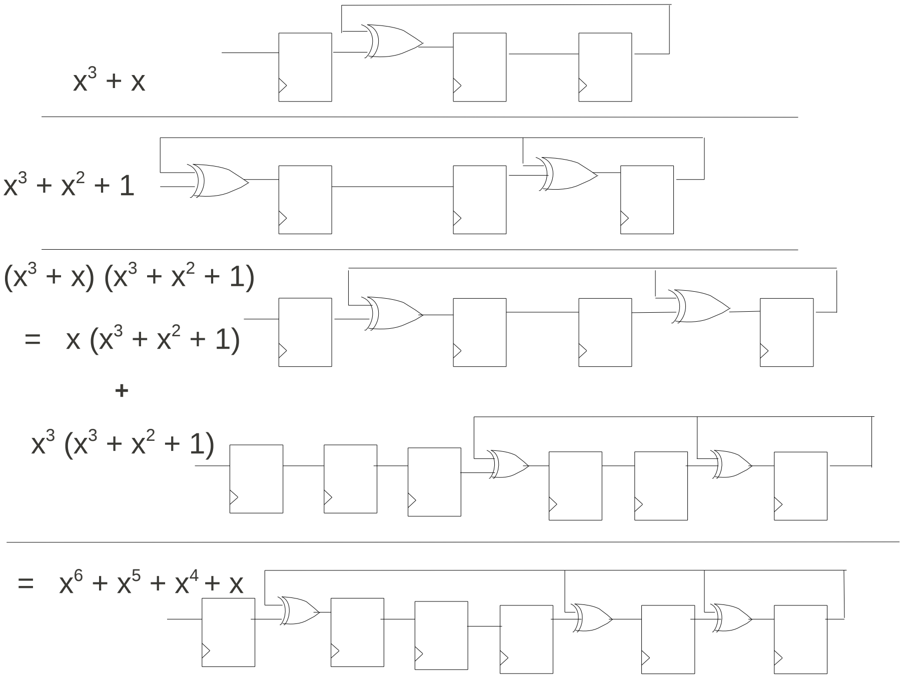


Figure 3.6: LFSR implementing multiplication

Similarly, by reusing the example 7, the division of $m(x) = x^5 + x^3 + x^2$ and $p(x) = x^3 + x$ can be implemented using LFSR is shown in the figure below.

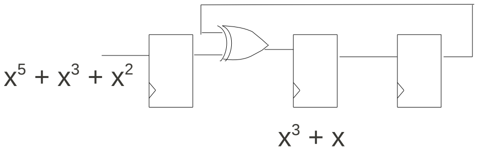


Figure 3.7: LFSR implementing division

The input to the LFSR is 101100 which is the bit vector representation of $m(x)$ and fed sequentially in the higher order terms first into the LFSR circuit. At the end of clock cycle 6, the remainder $r(x) = 100$ (x^2) is stored in the flip flops.

Chapter 4

Welch-Gong stream cipher

The motivation of this chapter is to discuss about the WG stream cipher it was first proposed by Guang Gong and Youssef [36]. The organization of this chapter is as follows, discussed about the general stream cipher design in section 4.1. In section 4.2 classified different types of stream ciphers. In section 4.3 discussed the design of stream ciphers based on LFSR. The design of stream ciphers can be vulnerable to different types of attacks which have discussed in section 4.4. Finally, in section 4.5, explained about the WG stream cipher, of its cryptographic properties and discussed the parameter selections for WG in order to achieve an efficient cryptographic stream cipher design.

4.1 Stream Ciphers

Stream cipher generates a cryptographically secure random sequence of bits known as *keystream*. For cryptographic operations keystream is XORed with either the plaintext or ciphertext at the bit level. The architecture of stream cipher consists of a shift register to store the secret key and initialization vector in combination with the feedback function which updates with respect to clock. The non linear filtering function is used to combine the m bits nonlinearly to generate a single bit of keystream. The basic architecture of stream cipher is as shown in figure 4.1. Stream cipher is also known as the *state cipher* because the process of encryption not only depends on the plaintext bits and the secret key, but also it depends on the internal state of the inputs.

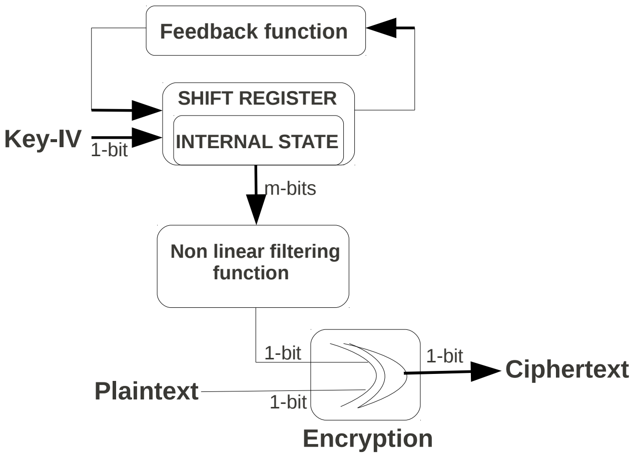


Figure 4.1: Architecture of stream cipher

Stream and block ciphers comes under the family of symmetric key systems. Some of the differences between them is that stream ciphers encrypt the plaintext message one bit at a time. Whereas block ciphers encrypt the plaintext in large blocks of bits at a time. Stream ciphers can execute faster, particularly in hardware, one of the reasons is that, operate on a single bit or a byte at a time, and don't have to store all the plaintext bits in buffer when compared to block ciphers. Apart from that, stream ciphers are built using simple devices which are easy to implement in hardware and execute efficiently, example Linear feedback shift registers (LFSR). Due to these factors stream ciphers have less hardware complexity. Stream ciphers have no error propagation because in stream ciphers during the transmission if a ciphertext bit is modified, then the decryption of other ciphertext bits does not effect. On the other hand keystream generation is independent to the encryption or decryption operation.

4.2 Classification of stream ciphers

Stream ciphers are classified into three types; One-time pad ciphers, synchronous stream ciphers and self synchronous stream ciphers.

In *one-time pad ciphers*, the keystream bits are used only once and the That is different keystream bits are generated for different plaintext messages. The disadvantage of one-time pad ciphers is that, the keystream must be of same length as plaintext. There by making insecure in practice [21].

The drawback of one-time pad ciphers motivates the design of stream ciphers whose keystream is pseudo-randomly generated from a smaller secret key [21]. In general, stream ciphers generate their keystream based on their internal state. If the keystream is generated independently to the plaintext and ciphertext then it is called *synchronous stream cipher* where as, if the keystream generated based on some previous ciphertext bits then it is *self-synchronous stream cipher*.

In *synchronous stream ciphers* the keystream is generated independently to the plaintext and ciphertext messages. Later on keystream is combined with the plaintext for encryption or with ciphertext for decryption operations. During communication both sender and receiver must be synchronized by using the same key and the same internal state. Sometimes synchronization might fail due to insertion or deletion of some of the ciphertext bits which further fails decryption. To restore synchronization again the process of *re-synchronization* is used [21]. Synchronous stream ciphers have better advantages in hardware applications because of less resources like limited gate count or low power consumption. Synchronous stream cipher is also know as *additive stream cipher*.

One of the drawback of additive stream cipher are, they uses the same secret key, resulting the same keystream generation always. From the security point of view re-use of same secret key is not a good idea.

To overcome this problem, the concept of *initialization vector (IV)* is used usually, stream ciphers IV is combined with the secret key. Initialization vector is a random block of bits which changes with every instance of the cipher. It is used to introduce randomness to the output of the cipher. This makes the output of the stream cipher unique and random compared to the outputs produced by the same key.

In *self synchronous stream ciphers* or asynchronous stream ciphers, the keystream is generated based on the secret key and also on the previously generated ciphertext bits.

4.3 Stream Ciphers based on LFSRs

We have already discussed about the concepts of LFSR in chapter 3. The rationale behind use of LFSRs in stream cipher design is to achieve few of the properties such as long period, large linear complexity and statistical properties. These properties play a major role in cryptography. Most of the stream cipher designs are constructed based on LFSRs are well suited in implementing in hardware and also are easy to analyze mathematically. Since LFSRs are the basic building blocks of stream cipher it is important to note that the output sequence of the LFSR does not inherit the property of linearity. If the stream ciphers exhibit the linearity property then its output would be vulnerable to known or chosen-plaintext attacks (see chapter 2). To avoid these linearity properties of the LFSR, there are three methodologies proposed to increase the security of stream ciphers based on LFSR's [21], they are

1. Nonlinear combination generators
2. Nonlinear filter generators
3. Clock-controlled generators

4.4 Classification of stream cipher attacks

This section briefly discusses different types of attacks on stream ciphers. Generally, cryptanalyst analyzes the keystream generator of the stream cipher. Thus most of the stream ciphers based on LFSRs are vulnerable to different types of attacks. These are as follows

1. *Exhaustive key search:* It is a common type of attack applicable to any stream cipher. For a given keystream, the attacker tries all the possibilities of keys, generates the keystream and compares them with the given keystream to find out the actual secret key.
2. *Periodic attacks:* The period (length) of the keystream must be large. If the period is small then it enables easy prediction.
3. *Correlation attacks:* Its a general type of attack applicable to all stream ciphers based on LFSR. In this type of attack, the attacker can correlate the keystream with the output of a similar device such as LFSR. This leads to extraction of the secret key.
4. *Algebraic attacks:* Stream ciphers are based on the system of mathematical structure usually, algebraic equations. By solving the algebraic equations one can recover the secret key.

Note: The linear complexity (see Chapter 3) of a binary sequence is length of the shortest LFSR which can generate that sequence. Linear complexity can be easily measured by using Berlekamp-Massey algorithm [21]. The attacker can produce the same binary sequence of the LFSR if the linear complexity is too small.

In order to be secure against these type of attacks, stream ciphers based on LFSR designs must posses a good cryptographic properties. How to achieve cryptographic properties by selecting various parameters is discussed in the next section by explaining a concrete example of a stream cipher called WG stream cipher.

4.5 Welch-Gong (WG) Stream Cipher:

WG cipher is an example of LFSR based stream cipher. In this section we briefly describe the WG cipher, its cryptographic properties. The parameter selections to ensure better cryptographic properties and for efficient hardware implementation are also explained.

4.5.1 Introduction:

For small and efficient hardware implementations, most of the hardware based stream ciphers use *Linear feedback shift registers* (LFSRs) and Boolean functions with compact *Algebraic Normal Forms* (ANF) [6] as their basic building blocks. Following the discovery of algebraic attacks on the Boolean functions with compact ANFs there are no longer secure. To overcome these algebraic attacks, *non linear feedback shift registers* (NLFSR) are used, Which basically updates the internal state of the stream cipher nonlinearly.

Usually in stream cipher design the complexity comes while analyzing the design. Most of the stream cipher designers invest more time in analyzing the design itself. Moreover LFSR based stream ciphers have well defined theoretical results for the purpose of analyzing compared to NLFSR based designs. WG cipher is designed such a way that its design is easy to analyze. Which allow the designers to evaluate or prove various security properties of the design.

The drawback of LFSR based stream cipher designs; they are vulnerable to algebraic attacks. To overcome these algebraic attacks, WG is designed based on nonlinear Boolean functions with large number of inputs and high degree [6]. In order to avoid complexity of hardware implementation, Boolean functions are designed using the polynomial form instead of ANFs. Since, the polynomial forms can be implemented using small finite field multipliers.

WG cipher consists of a keystream generator which generates long pseudo-random binary sequence called *keystream*. The basic idea here is to keep the WG keystream as random as possible. So, that encryption and decryption operations will be cryptographically more secure. Therefore, to ensure randomness properties WG keystream generator uses transformations called as *WG transformations*. Which acts like a *nonlinear filter function* [6] discussed in section 5.3. WG transformation can be implemented in polynomial form using the finite field arithmetic. and generates a sequence called *WG transformation sequence*. The existence of several cryptographic properties for the WG transformation sequence, which makes WG cipher suitable for cryptographic applications are already known [6].

4.5.2 Mathematical description of WG :

WG cipher considered as a *nonlinear filtering function*. The basic building block of WG keystream generator consists of a l stage linear feedback shift register followed by a *WG transformation* as shown in figure 4.2. To achieve few the cryptographic properties, the selection of LFSR is based on the selection of primitive polynomial (see chapter 2) of degree l over the finite field $GF(2^m)$ which is of the form $p(x) = x^l + x^{l-1} + ... + x + \beta$, where $\beta \in GF(2^m)$. The primitive polynomial acts as a feedback polynomial for the l stage LFSR.

The LFSR generates a maximal length WG transformation sequence known as *m-sequence* (see Chapter 2) over the field $GF(2^m)$ This in turn filters the output of the LFSR by the nonlinear WG transformation. Using the trace function denoted by $Tr(x)$ (see Chapter 2) the sequence is converted into binary keystream (b_i). In figure 4.2 the feedback signal "initial" is used only once during the initialization phase. Only the feedback which is in the LFSR keeps running when the cipher is operating. The output of the cipher would be 1 bit at a time.

WG design generates the binary keystream of period $2^n - 1$, where $n = ml$, m is the bit width of the LFSR and l is the degree of the primitive polynomial.

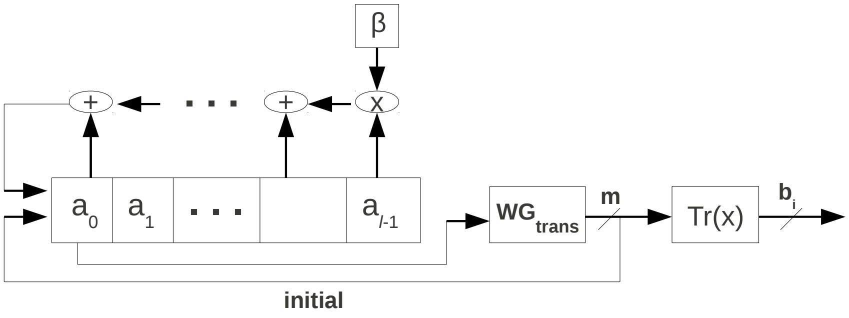


Figure 4.2: WG Keystream generator

Note: The feedback polynomial of the LFSR and the parameters of WG transformation should be selected properly to achieve better cryptographic properties and for efficient cipher design.

4.5.3 WG transformation

In this section, the formal definition of WG transformation is discussed [36]. WG transformation $f(x)$ is defined mathematically as

$$f(x) = Tr\ (g(x+1)+1), \quad where\ \ x \in GF(2^m) \tag{4.1}$$

Note, that the trace function Tr is used to convert the $f(x)$ from $GF(2^m) \rightarrow GF(2)$ and is defined as:

$$Tr_1^m(x) = \sum_{i=0}^{m-1} x^{2^{mi}},\ x \in GF(2^m) \tag{4.2}$$

The function $g(x)$ exists only when $m(mod3) \neq 0$ [36], where $g(x) = x + x^{q_1} + x^{q_2} + x^{q_3} + x^{q_4}$, $x \in GF(2^m)$, where q_i is defined as follows [36] [6].

For $n = 3k - 1$,

$$\begin{aligned} q_1 =& \ 2^k + 1, \\ q_2 =& \ 2^{2k-1} + 2^{k-1} + 1, \\ q_3 =& \ 2^{2k-1} - 2^{k-1} + 1, \\ q_4 =& \ 2^{2k-1} + 2^k - 1 \end{aligned} \tag{4.3}$$

and For $n = 3k - 2$,

$$\begin{aligned} q_1 =& \ 2^{k-1} + 1, \\ q_2 =& \ 2^{2k-2} + 2^{k-1} + 1, \\ q_3 =& \ 2^{2k-2} - 2^{k-1} + 1, \\ q_4 =& \ 2^{2k-1} - 2^{k-1} + 1. \end{aligned} \tag{4.4}$$

If α is a primitive element (see Chapter 2) of $GF(2^m)$, then the WG transformation sequence $\underline{b}$ can be derived as $\underline{b}$ = b_i , as shown in figure 4.2 and where $b_i = f(\alpha^i) = Tr\ (g(\alpha^i + 1) + 1), i = 0, 1, 2, 3$ ...

Apart from that from equation (1) we can say that the WG transformation involves computations such as multiplications over the finite field $GF(2^m)$. Whereas, elements of the finite field can be represented either in polynomial, normal, or in optimal normal basis and the selection of these basis are discussed in section 5.5.1

4.5.4 Properties of WG :

WG transformation sequence has widely investigated in the area of sequence design and found the existence of various properties both on randomness and cryptographic. Since the security of any cryptosystem relies heavily on randomness or unpredictability, various cryptographic properties related to the WG sequence $\underline{b}$, [36] .

Some of the Cryptographic properties are :

1. Long period of $2^n - 1$.
2. Balance property.
3. Ideal 2 – level autocorrelation.
4. n-tuple distribution .
5. The linear complexity of WG sequence increases exponentially with n.
6. 1 – order resilient.

WG sequence is proved for all the above mentioned properties, therefore it is well suited for the cryptography applications. Since, WG transformation involves various parameters such as finite field width m, length of the LFSR l, selection of basis for representing finite field elements and selection of primitive polynomial or feedback polynomial. Therefore it is worth to discuss about the selection of these parameters. in order to achieve hardware efficiency and cryptographic properties which is discussed in next section.

4.5.5 Selection of parameters for WG hardware implementation

For the purpose of our research we will concentrate only on smaller finite fields for example $GF(2^4)$ or $GF(2^5)$, because our focus is to design an efficient WG cipher for an RFID applications. and RFID needs less hardware resource utilization, low power and speed.

1. WG transformation exists only if $m(\ mod \ 3) \neq 0$ [36]. The value of m is selected in such a way that $m(mod3) = 2$, because the value of m results in smaller values of k and which in turn reduces the number of multiplications (section 5.3).

2. For the smaller values of m i.e. $m \leq 11$, table based design can be used for storing all the $GF(2^m)$ values. Later on the table based design can be implemented either in random logic or in ROM. Here *optimal normal basis* (ONB) can be used and exists only for values $m \geq 11$ [6].

3. WG cipher can be implemented either in normal or polynomial basis, and the selected basis must be at least 1-order resilient.

4. The complexity of computing WG transformation involves in multiplication operations over $GF(2^m)$. And the hardware implementation relies on choosing the basis which is used to represent the field elements.

5. If the WG transformation is implemented using normal basis of the form x^{2^i}, in hardware it is achieved by shifting of bits of the element x by cyclically to right by i position which computes x^{2^i} [37]. And optimal normal basis multipliers are used because they are smaller than other multipliers.

6. If the WG transformation is implemented using the Galois field multipliers, then the ONB is preferable. Since, ONB multipliers are smaller and occupies less area and low cost. But ONB does not exist for every m.

7. If the WG transformation is implemented using the random logic or a ROM then the polynomial basis is preferable. This allows the implementation of multiplications in the LFSR by wired shifts and with less XOR gates [24]. The utilization of hardware resources depends on the selection of basis and in turn it correlates with the clock speed [24].

8. After selection of basis, the LFSR feedback polynomial $p(x)$ (which must be a primitive polynomial) and a generating polynomial $g(x)$ is selected. $g(x)$ is generally used to generate all the elements of $GF(2^m)$ and the basis are used to represent these elements. The primitive polynomial is of the form of $p(x) = x^l + x^{l-1} + ... + x + \beta$. To implement, β polynomial basis is preferred because of less hardware utilization compared to normal basis.

   Note: If a polynomial basis is used then β is of the form α^k, where α is a root of $g(x)$ and values of k in equation 2 varies such that multiplication by α^k can be implemented using wired shifts and few XOR gates. Where as if we use normal basis then multiplication by β needs to be a constant which uses more area.

9. The degree of the primitive polynomial $p(x)$ must be large, so that, the LFSR generates a longer period and to achieve this the size of the internal state (LFSR in bits) should be twice as the secret key.

10. The size of the internal state must be twice as the secret key, in order to prevent from the time/memory/data trade off attacks. While the algebraic degree of the primitive polynomial must be high, to prevent from the algebraic attacks [6].

11. To ensure the large linear complexity of the keystream, parameters such as n and m must be selected such that the linear complexity is higher. Where m is the finite field bit width and n is the size of the internal state (LFSR in bits). Linear complexity can be increased by increasing the number of stages in LFSR and vice versa [6].

12. The keystream generated by the WG keystream generator ensures the property of periodic two-level auto correlation property [6].

    Note: To achieve this property, the feedback polynomial of the LFSR over the finite field $GF(2^m)$ and the WG transformation should use the same basis in the WG keystream generator.

13. We are interested in smaller finite fields ($m \leq 11$), table based design is used for computing all the elements of $GF(2^m)$ and in turn table based design can be implemented using random logic as well as to achieve higher clock speed. On the other hand, polynomial basis is used to represent the field elements.

14. To avoid the slowing down of the clock speed of the cipher which occurs because of the multipliers in the LFSR feedback which dominant the critical path. The number of multiplications (i.e. efficient implementation of β) and number of taps on the feedback LFSR can be reduced as much as possible.

From the above discussion it is evident that, the design parameters such as, the feedback polynomial of the LFSR, the number of bits m used for the WG transformation, and the basis used to represent the field elements affect the hardware implementation of the WG cipher and its security.

Chapter 5

Radio Frequency Identification (RFID)

The motivation of the chapter is to understand the RFID technology. In section 5.1 we briefly discuss various reasons why RFID has been the successor of optical bar code system. Section 5.2, explains the components involved in the RFID system and different types of RFID tags and their characteristics. Section 5.3 describes the important concepts such as inductive coupling which is involved during the communication between the tag and the reader. RFID system functionalities, frequency bands and requirements are developed by the standard organization bodies such as ISO and IEC, where as the format of RFID tags data is developed by the EPC global. Different standards and classes of EPC are described in section 5.4. Section 5.5 explains the security and the role of cryptographic solutions to overcome the security concerns in RFID systems. We also explained various hardware metrics and their constraint requirements for passive type RFID tags. Finally, section 5.6 and 5.7 describe some of the RFID privacy goals and definitions and explain different type of attacks on RFID systems respectively.

5.1 RFID technology

RFID technology has become an integral part of our daily life. RFID is considered as a wireless automatic identification and data capture (AIDC) technology [2]. Any remote object or person that has an RFID device attached can be identified automatically. The communication in RFID system generally takes place between the three components, RFID tag or transponders, RFID reader or transceivers and back-end database system (computer). RFID tags are classified into three types; passive, active and semi-passive tags which can operate on different frequency bands; Low frequency (LF), high frequency (HF), ultra high frequency and microwave frequency. For our research purpose we are interested in passive tags which communicate with the reader and operate on high frequency.

On the other hand, optical bar codes were used on consumer objects for identifying over twenty years. Universal Product Code (UPC) or bar code is one of the widely used optical barcodes, designed in 1974 [38]. RFID technology stands as a successor before UPC bar code technology for past several years. Definitely RFID tags has benefits over barcodes. Some of the advantages of RFID tags over barcodes are:

1. RFID tag do not have to be in line of sight for operation or without any precise position whereas, optical bar code require line of sight.
2. RFID readers can read (scan) hundreds of tags per second whereas, optical bar code has to be read manually one at a time.
3. RFID tags store their identification number, store specific application data, execute and respond the data information for any particular queries from the reader.
4. Since RFID tags are silicon based various functionalities can be implemented like integrated sensors, read/write storage, capable of supporting cryptographic encryption systems and access control [39].

RFID stands as “the first important technology of the 21st century” [40]. Due to the technological advancements in the semiconductor manufacturing, researchers are shrinking the size of RFID tags. In recent years, Hitachi came up with the worlds smallest new RFID chips called Hitachi μ-type chip which is just 0.4×0.4 millimeters and it uses uses ROM for storing a unique 128 bit identification number [41]. Due to cutting edge technologies the cost of RFID tags has drastically dropped and the size of tag varies depending on applications. The cost of RFID tags at present can go up to U.S. $\$1.50$ for smaller quantities, it is expected that this cost would drastically fall in the future years to $\$0.10$ cents or less [42]. Where as RFID readers can cost several thousand dollars each at present, since they are in demand too, it is likely that their cost will soon drop significantly [42].

The caliber of RFID devices basically depends on how well the RFID standard definitions are implemented. Some of the standardization bodies like International Organization for Standardization (ISO) and International Electro-technical Commission (IEC) play a crucial role in regulating the use of RFID. We will discuss briefly about different standards of RFID in the section 6.5.

The broadening of RFID standards diminishing cost of RFID further, many well known large organizations like Wal-Mart and the U.S. Department of Defense (DoD) gave a good start in deploying RFID tags by using them for their suppliers. RFID technology anticipate to play a vital role in many different applications in the near future. Now a days RFID is used in hundreds, if not thousands of applications [43] like proximity cards for access control into the buildings, supply chain tracking, airports, tracking books in libraries, traffic management, monitoring the patients in hospitals and preventing theft of automobiles.

RFID system

In general, RFID device is called an RFID tag. RFID tag is a tiny silicon chip which is designed for contact less (or wireless) data communications using radio frequency (RF) waves. The silicon chip stores the unique identification information of the object or human and transmits the identity in the form of unique serial number (ID) known as Electronic Product Code (EPC) to the RFID reader which in turn communicates with the back end database system. RFID devices are also known as EPC tags.

Example of RFID system: In a hospital over 300 patients can be tagged with the RFID tags. Now the doctors and nurses would be able to identify the patients name, gender and age with their RFID reader. With this information now the hospital staff can retrieve the patients file from the back end database system.

5.2 Components of RFID system

RFID system consists of mainly three components and the interaction between these components are given in Fig. 5.1 [44].

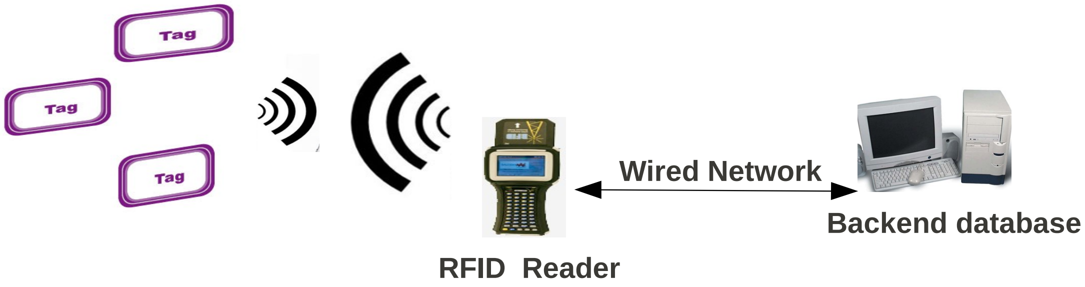


Figure 5.1: RFID System

1. **Tags:** RFID tags are also called as transponders it consists of a tiny integrated circuit (IC) which is integrated with a small non-volatile memory used for storing the identification number (ID) of an object (e.g., the laptop) and with an antenna connected to it. The antenna

permits the tag to couple with the electromagnetic (EM) field which is generated by the reader antenna to obtain the power or to communicate with the reader or to do both. The tag sends the ID to the reader under request. Tags can also be called RFID labels which are attached to the objects. Transponders have wide range of functionalities implemented in them like read/write memories, encryption algorithms, sensors and access control.

2. **Readers:** RFID readers are also called interrogators or transceivers. These are capable of generating electromagnetic (EM) field for transmitting and receiving the responses from the tag. RFID readers can scan hundreds of tags per second. At the same time, reading distance might be hundreds of meters which depends on the type of tag being used, frequency bands, reader power limitations and interference by other systems [45]. Generally, RFID readers are placed in a fixed locations to communicate with the tags and also with back end database systems. Readers will be in charge for queries to the tags.

3. **Backend database systems:** Since tags have small memory space, and it cannot store all the information related to the tagged objects. Back end database system is used for storing the rest of the tagged object informations. For verification purpose the RFID readers will check the database to identify a tagged object and to obtain further information.

Moreover, that memories in RFID tags can be either read-only or read-write accesses. In fact, nowdays the standard one are read-write memory tags. Read-only tags are programmed by the manufactures with their ID number and they cannot be altered. Whereas, read-write memory tags are divided into two parts for the user purpose, first part is for secure read-only in which user can write the unique ID number and the second part is for any free rewritable data [46].

5.2.1 Types of tags

RFID tags can be classified into three types; passive, semi-passive and active tags based on their power source.

Passive tags: These tags are small in size with no batteries and less expensive. Thus the main source of power in these tags is from the signals coming from the RFID reader. Passive tags cannot operate in the absence of RFID reader.

Semi-passive tags: Tags do have batteries, but they require an external source to activate them by means of interrogated signals from the reader.

Active tags: Tags have batteries to power up their circuitry for transmissions. Active tags can initiate a communication [44] and can operate even in the absence of RFID reader. Active tags are expensive.

Passive and active tags are widely used in most of the RFID systems and also in many industries. Semi-passive tags are the combination of both passive and active tags characteristics. Selection of tags are mainly based on their frequency, range, memory, and other characteristics. Some of the major characteristics of active and passive tags are tabulated below in the table 6.1 based on [47].

Characteristics	Passive tags	Active tags
Strength of signal required	very high	very low
Availability of power	only in the presence of reader	anytime
Read range	upto $3-5$ meters or even less	upto 100 meters
Multiple tag scanning	few numbered within 3 meters of reader	1000 of tags scanned, upto 100 meters
Data-storage	128 bytes in read-write memory [3]	upto 128 kb in read-write memory [3]
Life time	upto 20 years	upto 5 - 10 years

Table 5.1: characteristics of passive and active tags

5.3 Inductive coupling communication

In this section we discuss some of the concepts used during the communication between the passive type tag and the reader.

Passive tags receive their power to enable their integrated circuits from the RFID readers via electromagnetic field. It uses the concept of inductive coupling or far field (defined as the area from the antenna to the point where the electromagnetic field forms) for communicating the data with the readers. Inductive coupling is used only for the frequencies of 13.56 MHz and below [48]. The concept behind inductive coupling is that, the reader's antenna generates electromagnetic field and induces it in the form of current into the tag's coupling elements such as coiled antenna and a capacitor. The current induced in the coupling element is used to charge the on-tag capacitor, which provides the operating voltage and the power to the tag circuitry. Inductive coupling works only in the near field communications (near field is defined as the area after the point at which electromagnetic wave is fully formed and separated from the antenna).

5.4 RFID Standards and frequency bands

Since RFID systems are accepted by many different applications, RFID should provide some functionalities other than just basic object identifier [49]. These functionalities should be developed on a proper standardized system. Major organizations like ISO and EPC global which play a crucial role in developing RFID standards for different frequencies and applications.

5.4.1 Standards

Till today, many RFID standards have already been in use and many are being proposed. Standards mainly observe how the tag and the reader communicate to each other, how the data content is edited or organized and know whether the applications are following the standards, for instance, in companies while shipping their RFID labels. ISO collaboratively works with the International Electro-technical Commission (IEC) they deal with the general type of standards covering with the problems related to the air interface, data-content, conformance and performance [1]. ISO standards are mainly towards the technology side not particularly into application-oriented [1]. Some of the ISO standards and their respective applications are explained below:

ISO 18000 is a multi part standard specifies protocols for LF, HF and UHF frequency bands. Whereas, in HF tags ISO 14443 and 15693 standards are used for applications such as proximity cards and vicinity cards respectively. In addition to these standards ISO 18000-3 standards are used for item management, air interference communications [5] [3] [1] [50].

5.4.2 EPC structure and classes

EPC global was developed from the other organization called the Global Standard 1 or GS 1 and it deals with barcode standardization. EPC global developed the EPC, which is most widely used and recognized. EPC has a unique code for identification of objects or items which is similar to the bar code numbering system i.e. UPC (Universal Product Code). Like the bar codes, EPC contains information about the manufacturer the product details and the most crucial part is the unique serial number, which is not shared by any other item or object on this earth. The current form of the EPC code is shown below in figure.5.2

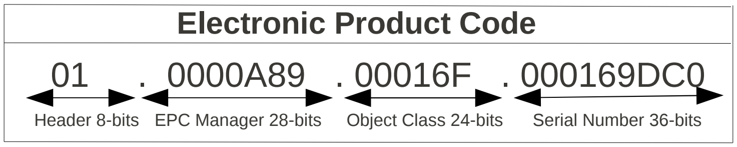


Figure 5.2: EPC structure

There are different EPC structure exists but for our research purpose we are interested only in 96 bits [51]. The EPC code present on the tag generally is a 96 bit length and used for encoding and reading the data from the 96 bit RFID tags. As shown in figure 5.2, 96 bit identifier structure is divided into four partitions which are explained below:

- Header (8-bits): Defines which version of EPC is being used and also defines the number, type and length of all subsequent data partitions. This 8 bits provide 256 possible partitioning schemes [52]. First scheme is EPC type I, proposed for object identification number.
- EPC Manager (28-bits): Defines about the manufacturer or entity and responsible for tracking and maintaining the object class type codes and serial numbers in their domain [52]. Thus, 28 bit provide maximum of 2^{28} million possible manufacturers [52].
- Object class (24-bits): This is equivalent to the product number. Additionally considered as a tracking mechanism for particular groups like stock keeping unit (SKU) or can be used for lot number.
- Serial number (36-bits): The final partition encodes the unique object identification number. For all similar object types, EPC serial number provides 2^{36} unique identifiers.

EPC global has classified RFID tags into five classes (1–5) in order to improve their design, develop different capabilities and usage. The classification has become well popular in manufacturing and supply chain applications due to its specifications. Higher the class levels, higher the functionalities and more sophisticated capabilities. The classification of RFID tag classes are defined as follows:

- Class 0 or 1: Simple passive identity tags with minimum functionalities like read-only memory.
- Class 2: Passive tags with higher functionalities like tag identity, read - write memory, with enhanced security features [45].

- Class 3: Semi-passive tags enhanced with additional built in power source such as battery which enables the tag even in absence of RFID reader. Therefore, tags can provide better read range compared to the class 1 and 2 tags.
- Class 4: Active tags with an in built battery to operate the tag by itself. It has same functionalities as class 3 tags. In addition, they can communicate with other class 5 tags using both active and passive communication [49].
- Class 5: RFID readers which encompass the functionality of class 4 active tags and in addition has capabilities to power up and communicate with pure passive class 1, class 2 and class 3 tags using passive communication.

Due to wide area of applications and demands, EPC global is developing the functionalities and standards in the same classes, for example, Class 1 is an updated version of class 0, which has capability to work on both frequencies UHF and HF. The new version of Class 1 Generation has better features like larger memory size, user memory option and kill command option, which allows the tag to permanent defunction [1].

5.4.3 Frequency bands and characteristics

In order to utilize the RFID communication without any interference with other radio service, RFID frequencies are divided into different frequency bands. In general, frequency bands which are used by cell phones, broadcast, astronomy etc, are called primary user bands. In addition to these bands, there are different bands which are used for industrial, scientific and medical (ISM) purposes. RFID system is allowed to operate on ISM bands mostly [53] and these are 125/134 KHz (LF), 13.56 MHz (HF), 830-930 MHz (UHF), 2.45 GHz (MW), [3] [1].

Different types of RFID tags uses different frequency bands and are selected on the basis of application requirement. The operating range of the RFID tags mainly depends on its frequency used. Some of the characteristics and applications of RFID is shown below in table 5.2.

5.5 RFID security by hardware design

RFID technology is widely used in many applications but the major concern in RFID technology is security and privacy. Since, the communication between the tag and the reader takes place via wireless, i.e. the information exchanged between them is over an insecure channel. Generally, insecure communication channels are vulnerable to different types of attacks. Therefore, security is needed for the data which is written on the tag and privacy is required when the tag is carried

Frequency bands	Low frequency(LF) [125, 134-135 KHz]	High frequency(HF) [13.56 MHz]	Ultra high frequency (UHF) [830-930 MHz]	Microwave frequency (MW) [2.4 GHz]
Range	Short range($<$ 1m)	Higher range($<$ 1.5m)	Long range(6-8 m in US and $>$100m for active tag)	Long range($>$10m and $>$100m for active tags)
Data transmission rate	Low data rate ($<$1Kbits/s)	Reasonable data rate(25 Kbits/s)	High data rate (30 Kbits/s)	High data rate ($>$100 Kbits/s)
Type of tags	Passive	Passive	Active	Active
Tag size	Large	Medium	Small	Small
Applications	Access control, vehicle identification, animal tracking etc.	Electronic ticketing, smart cards, access control	Baggage tagging, electronic toll, collection etc	Finished products tracking, electronic toll collection
ISO/IEC standards	14223, 11784/5, 18000 - 2	15693, 14443, 18000 - 3	10374, 18000 - 6	10374, 18000 - 4, 18000 - 5
EPC standards	–	Class 1 (Gen 1) HF	Class 0 (Gen 1), class 1 (Gen 1), UHF class 2 (Gen 2)	–

Table 5.2: RFID tag frequency bands and characteristics (Based on [1])

by a person or an object. To overcome these concerns in RFID tags, one of the solution is to implement cryptographic solutions in RFID tags. Recent publications have surveyed more elaborately about the security and privacy issues in RFID systems [54] [55].

5.5.1 Role of cryptography

In chapter 2 we have seen various cryptographic primitives such as symmetric-key, asymmetric key and hash functions. Cryptographic primitives are built in RFID systems to achieve some of the security objectives which are described below:

Confidentiality normally, the data is communicated between the RFID tag and the reader over the insecure communication channel. To keep the data confidential or secret from the attackers, the data is encrypted using a secret key in the RFID system. It is important to protect the data either from tag-to-reader or reader-to-tag because of various factors [11]. Confidentiality can be achieved either by using symmetric or asymmetric systems.

Authentication is another important objective of cryptography. Here the objective is to prove their identities either by the RFID tag or the reader during their communication. There exists number of cryptographic protocols for the purpose of authentication, one of the commonly used protocol is challenge-response protocol. Authentication protocols can be either *single-side* or *mutual authentication*. In single-side authentication only the reader or tag will authenticate their identities whereas in mutual authentication both the tag and the reader should prove their identities to each other. In RFID system, authentication plays a major role in terms of security feature such as preventing cloning of tag and unauthorized access to the tags [11]. Data authentication can be achieved by MACs (message authentication codes) in symmetric-key systems.

Data integrity Provides security from the data manipulation threats between the tag and the reader. Usually, data manipulation can be found by comparing it with the transmitted data. In cryptography, hash functions are used to provide integrity of the data which ensures from man-in-the-middle attack.

Non-repudiation is not important for RFID applications [11].

In general, the role of cryptographic objectives required for constrained devices like RFID system depends on the type of application. For low-cost RFID applications the level of security requirement is moderate because the amount of data required for encryption and decryption is limited [13]. Therefore the role of cryptography in RFID tags depend on factors such as physical space required for the implementation, cost of the tag and power consumption. In next sections we discussed the various factors required while designing a cryptosystem for a low cost RFID tag.

5.5.2 Lightweight cryptography

Since, both security and privacy issues in RFID systems needs to be addressed by means of cryptographic solutions. It is a challenging task mainly because of strong constraint environment for computing and communication resources, which leads to higher cost. Due to these factors, the design of cryptographic solutions in RFID systems is not an easy task. Therefore, the design of efficient cryptographic algorithms and protocols has been proposed which is known as *lightweight cryptography*.

To take into account, the strong stringent requirements in RFID tags such as area, power and cost, the design of dedicated lightweight symmetric and asymmetric cryptosystems must be within the range of RFID constraint limitations. The advantage of symmetric-key over the asymmetric system is that, it has limited mathematical computations. Which allows the hardware design of symmetric-key systems efficient in terms of area and power to implement them on any hardware platforms. Most of the recent publications have proposed efficient lightweight cryptography based on block ciphers [18] [56] [57] and stream ciphers [58] [59]. Recent studies suggest that stream ciphers require fewest computational resources in terms of area, power and performance compared to block ciphers and hash functions [60]. Whereas, asymmetric or public-key systems involve complex arithmetic operations over higher finite fields. These operations involve hundreds of bits thus need large amount of memory size to store them. Due to this, the design utilizes more silicon area and power compared to symmetric systems. Some of these factors makes them more cost expensive in terms of hardware implementations. Therefore, public-key systems are not preferred to use in RFID systems in past couple of years [11].

5.5.3 Factors for good constraint implementation

In this section we briefly discuss some of the factors needs to be considered while designing a hardware cryptographic stream cipher for a specific application like RFID. Here, we are interested in passive type RFID tags which require highly efficient cryptographic ciphers, in terms of hardware resource utilization and low power.

Since, design of passive RFID tags need stringent requirements in terms of less silicon area and limited power budget, the design of stream cipher must be constrained with these factors. Technically, the silicon area and the power affects the cost of RFID tags. Thus if the design of the cipher utilizes less silicon area then it is possible that, the design may utilize limited power. The power available for passive RFID tags are limited because, passive RFID tags requires power to enable its circuit operation via air interface [11]. Therefore cryptographic circuitry must consume limited power.

In addition to above requirements security is one of the important factor that needs to be considered while designing the cipher for RFID tags. One of the examples of passive RFID tags where security plays a major role is the supply chain management where the products are transfered from one place to another using the concept of EPC. Here, each product has its own unique code or EPC for each passive RFID tag. The ability to track or identify these products raises the security and privacy concerns, in other words the data transmitted or exchanged between the tag and the reader can be monitored or read by some unauthorized persons. Therefore, building up a cryptographic cipher functionalities is a good solution to overcome these security concerns.

Besides above mentioned factors, the design of cryptographic cipher must have a good data transmission rate when it is used in RFID applications.

In the later sections of this chapter, we will discuss some of the important measurements that needs to be calculated while designing a hardware circuitry for a RFID tag. Few of the optimization techniques commonly used to achieve limited silicon area and power under specific Complimentary metallic oxide semiconductor (CMOS) technologies are also discussed.

5.5.4 RFID optimality metrics

This section discuss about the cryptographic hardware circuitry design metrics measured for the purpose of RFID tags. By measuring the following metrics, specifies us how efficient the design is in terms of various performances. These metrics can be measured during any digital design flow (for example; ASIC flow) with the help of simulation tools. The results of the metrics can justify whether the design is suitable for an RFID application or not. CMOS technology is used for the design of cryptographic circuitry.

Area: Area can be defined as the amount of space utilized by the digital core design over the silicon, i.e. the design excluding the power resources and the input-output (I/O) ports. Usually, area is denoted as μm^2 but it can also be measured in terms of Gate Equivalence (GE). The measure of GEs gives a technology independent measure of area. It can be measured by dividing the total area of the design with the lowest power two-input nand gate's area. Further, the *pre* and the *post* place-and-route analysis are carried out in ASIC design flow. In pre-place-and-route analysis, the area is measured before the place-and-route and the measure is based on just the area of the cells, or possibly with an estimation of area of wires. Whereas, in post-place-and-route analysis the area is measured after routing wires between the cells. In recent technologies, the area of wires can be a limiting factor in the area of circuit. The latter analysis is more accurate than the pre-place-and-route analysis.

Maximum clock frequency: It is defined as the design at which the highest rate of the input clock is required for its operations. It is measured by calculating the critical path (longest timing path in the design) and setting the upper bound of the clock frequency.

Bits per cycle: It is defined as the number of output bits generated per clock cycle from a cryptographic design. For example, it is measurement of the number of output keystream bits per clock cycle from a stream cipher design. In other words, it can be measured as the number of output bits from all subsequent blocks of keystream divided by the number of clock cycles per block. It is also called output rate of the design.

Throughput: It is defined as the rate at which the circuit generate its output with respect to time. Throughput is measured by multiplying bits per cycle with the clock frequency. It is expressed in terms of bits-per-second. Maximum throughput attain at the time of maximum clock frequency.

Latency: It is a performance measure like throughput, which measures the time to compute one cryptographic encryption operation (including the total time taken during the initialization, loading key and IV). Usually, it is measured (units in milliseconds or microseconds) between the start of the encryption operation and its completion.

Power consumption: It is a measurement to calculate the power consumed by a cryptographic CMOS circuit. The total power consumption by a CMOS circuit is the sum of two factors; static and dynamic power consumptions which are discussed in detail in section 5.5.5.2.

Energy per bit: It is measured as the total power consumed divided by the throughput. Here, both (power and throughput) are measured at the same clock frequency.

Area-time product: Its a product of the time taken by the design to generate new output each time and the total design area. Whereas, its reciprocal will give the *throughput-to-area* ratio which is used for measuring the design efficiency .

Power-area-time product: It is measured as a product of area-time product and the power consumption.

Power-time product: It is a product of time and the power consumed. This measurement is useful in RFID applications, where both power and time play a major role during communication between RFID reader and tag.

5.5.5 RFID design considerations and process

The following discussion would be on considering various factors while designing a cryptographic cipher for a passive type RFID tags. Designing a cryptographic cipher using CMOS technology for an RFID tag is a challenging task because of its various stringent requirements such as, the CMOS silicon area, power consumption and the number of clock cycles required for a single encryption operation.

5.5.5.1 CMOS technology and silicon area limitations

The CMOS technology is generally used for designing and fabricating any digital integrated circuitry. Among various CMOS technologies known till date the most commonly used CMOS technology is 130–250 nm for the production of high frequency (HF) RFID tags. But most recently fabricated RFID tags were based on 180 and 130 nm CMOS process technologies [11]. The other newer technologies like 90, 60 and 45 nm were not used for RFID tags yet because of their drawbacks. They are not featured with the non-volatile memories like EEPROM or flash to store the unique identity or EPC on the tags [11]. The RFID readers supply voltage in the form of electro magnetic field to the tags to enable their circuitry which might exceed there voltage limits in the newer technologies.

There are millions of RFID tags available in the market therefore, the challenge is to design and fabricate a cost effective passive RFID tag. This is possible, only if the CMOS silicon area is very less that is, the total size of RFID tag must be less including the cryptographic cipher in it. The size of RFID tags increases as the cryptographic functionalities increases. In other words, we can say that the size of the RFID tag depends on the complexity of the cryptographic functionalities and the area of cryptographic cipher is directly proportional to cost of RFID tag. By using the advanced CMOS technologies the size of a single GE will be smaller which makes the silicon area lower. Besides that, the cryptographic circuitry which contains 1000 GE can be implemented on a RFID tag without any additional cost of production [11]. Whereas symmetric cryptographic cipher can be designed upto 5000 GE [11]. If the symmetric cryptographic cipher uses advanced CMOS technologies such as 180 nm or less then the size of the cryptographic circuitry would be comparatively smaller compared to RFID tag of sizes $0.1 - 0.25\ mm^2$ [11].

5.5.5.2 Power and energy limitations

In CMOS technology the total power consumed by the circuit is the sum of *static* and *dynamic* power consumption. Static power consumption is caused by the leakage current of each transistor gate in the circuit and therefore it is proportional to circuit size. In most of the designs, static power is often small and ignored based on the selection of CMOS process technology (for advanced CMOS technology static power increases with the increase in subthreshold leakage current which increases due to decrease in threshold voltage) [61]. However, the major concern is the dynamic power consumption which occurs due to the switching of gate output from 1 to 0 and vice versa. Additionally, dynamic power consumption is proportional to the clock frequency of the circuit and the switching activity of gates because it concentrates around clock edges [11]. To minimize the power consumption of any CMOS design, it needs to minimize the factors in equation given below.

$$P_{dynamic} = \quad C_L \cdot V_{DD}^2 \cdot f_{clk} \cdot p_{sw} \tag{5.1}$$

Where C_L is the load capacitance of the circuit design, f_{clk} is the effective clock frequency, V_{DD} is the supply voltage and p_{sw} is the switching activity of the gates in the circuitry. We will discuss in detail various methods to reduce the power consumption later in section 5.5.6.

Power consumed by the cryptographic circuitry in passive tags is also an important factor to look into. While implementing cryptographic circuitry on an RFID tag it should not limit the operating range of the RFID tag. Because, the power available for the RFID tag from the RFID reader is very less, if more power is utilized by the cryptographic circuitry itself then the remaining power would be insufficient which may affects the tags operating range. Generally, in digital circuits power consumption can be measured indirectly by measuring the average current $Iavg$ because $Iavg$ can be calculated directly from simulators. In RFID passive tags, the power received from the readers is denoted as average power which is proportional to the average current; $Pavg = Iavg.Vdd$, and $Vdd = 1.5v$ is constant [11]. Therefore, the average power consumption by the cryptographic circuitry in passive tags must be low. Usually, in RFID tags the average power consumption is calculated in terms of μA. The upper bound for the average current consumption in RFID tags available is 15μA (in $350nm$), beyond this limit the operating range would be decreased [12]. It has been shown previously that the passive tags power must be in range of 5 to 10 μA [10].

On the other hand, the energy consumption by the cryptographic circuitry depends on the average power $Pavg$ along with the duration t of the cryptographic computation i.e $E = Pavg \cdot t$. But, energy efficiency plays an important role only for the devices which are powered by batteries. In passive tags average power is transmitted via electromagnetic waves from the reader. Thus, energy consumption in passive tags does not play an important role unless there is long enough time for cryptographic computation [11].

Most of the digital circuits designed for RFID tags have clock frequency 100kHz [62] [11]. The average current depends approximately linearly on the clock frequency and the supply voltage [12]. Therefore, during computation of the cryptographic circuitry power consumption per clock cycle must be equal i.e, no clock cycle must consume excessive power.

5.5.5.3 Throughput and latency requirements

In RFID systems, throughput and latency are also important parameters to consider. Usually, in RFID systems the data transmission rate is low ranges from 6 Kbps to 106 Kbps [11]. In terms of cryptographic hardware throughput which are clocked at 100KHz must be able to compute an encryption in 2500 clock cycles in the slowest case and 40 clock cycles in the fastest case [11].

Latency measures the time to compute one encryption it means that the number of clock cycles required for computation one cryptographic encryption. The total time taken to complete the cryptographic computation might also influence the response time of the tag. This can happen in stream ciphers, especially during initialization phase where frequent key updates are needed, depending on protocol being used that might lead to long latency [12]. For example, in RFID HF tags the response time is 300 μs [12] [11].

5.5.6 Low Power Design techniques

Due to low power budget requirements in RFID systems, the design of cryptographic primitive must ensure low power utilization. In this section, we discuss some of the low power design techniques which have been used widely in recent years [12] [11] [63] [64].

To minimize the dynamic power consumption ($P_{dynamic}$) mentioned in equation (5.1) it is clear that, C_L increases as the size of the chip increases and size of the chip increases due to more number of gates present in it. The power consumption can be reduced by minimizing the chip size and suppling the minimum supply voltage V_{DD}.

The switching activity p_{sw} can be minimized by using the method called sleep logic [12]. By using this method, unnecessary switching activity can be ignored and it is implemented by inserting AND gates at the input of the combinational logic. Thus, if there is better supply voltage and the minimum switching activity then the better choice to reduce the power consumption is by minimizing the effective clock frequency f_{clk} of the circuit. Effective clock frequency can be minimized by using the concept of clock gating [12]. Clock gating is used to switch off some of the logic blocks in the circuit when they are not in use. Previous study [11] suggests that by using clock gating concept to datapath registers and also to some of the control logic registers can minimize power consumption significantly. In other words these registers are used only when there is a potential signal change. In addition, gated clock can be used in the memory cells in a design to minimize the unwanted switching activities [65].

In passive RFID systems, the energy consumption $E = Pavg \cdot t$ and duration of the computation is of minor priority as explained in section 5.3.2. One of the effective approach for lowering the average power $Pavg$ is to increase the computation with respect to time. Thus, by stretching out the same computation can lower the $Pavg$. Since, most of the RFID circuits are not clocked higher than 100 KHz another approach is to serialize the computation by stretching it out over more number of clock cycles. Both approaches are based on increasing the time t, it is defined as $t = N_c \cdot C_t$, where N_c is number of clock cycles and C_t is the cycle time $C_t = 1/f_{clk}$, i.e. C_t is reciprocal of clock frequency f_{clk} [11].

5.6 RFID privacy goals and definitions

Privacy is also an important concern in RFID systems. Tags start responding automatically and start transmitting their information if any unauthorized readers are in the reading range. Therefore, an attacker can easily read the tags confidential information from a long distance without the knowledge of the person who is carrying the tags. Hence protecting the tags data is important.

Information privacy In a ideal situation the backend database system or server is assumed to be secure and reader communicates with server over secure channel. Only a legitimate reader can query the server to look up tag keys.

Strong privacy A system is said to have strong privacy if the adversary cannot distinguish between the outputs of any given tag. But an adversary is capable of obtaining the tags output at a given time and break the privacy [66]

Forward untraceable It means that, if an adversary is able to access the internal state of the tag at time T, the adversary cannot determine whether the tag has involved in a transaction after certain amount of time T+α (for $\alpha > 0$), provided only when adversary has not continuously eavesdropping the tag after time T [67].

Backward untraceable If an adversary is able to deduce the internal state of the tag at time T, the adversary cannot tell whether the tag is involved in a transaction before time T [67].

Challenge response A tag can reply with information only if, it receives the legitimate challenge from the reader. The paradox here is that, a reader cannot know which challenge has to transmit to a tag unless it knows the tag's identity.

Random nonce To further strengthen the challenge-response mode, it is encouraged to use the random nonce. This makes sure that every exchange is a unique. Thus, an adversary observing the tag and the reader will have a growing set of challenge and responses. To make the security symmetric, it is important for both parties to use random nonce for mutual authentication.

Key update The tag should update its secret key after every successful authentication. This will further strengthen the strong privacy of the system. Most protocols exchange new keys. Based on the current secret shared by both parties, the keys should be updated. Moreover, the new keys should still maintain the correctness of the system.

Once a tag has been successfully authenticated by the system, its stored key/secret will be freshly randomized so that any kind of tracing can be prevented.

5.7 Types of attacks on RFID tags

Denial-of-service: Tags which are specially designed by the unauthorized persons (attackers) to create a confusion to the interrogator from identifying the individual tags.

Spoofing: It is a kind of technique in which a attacker duplicates the tags data and transmits it to the reader.

Counterfeiting or Cloning: It can be defined as duplicating of one tags data to another tag and later on the duplicated tag is used for communicating with the reader.

Data tampering: In data tampering the attacker tries to erase the data on the tag and make it useless for communication or even he tries to modify the tags data.

Clandestine tracking: Generally RFID tags emit their unique serial number in respond to the readers query. The readers can scan the tags when they are in readable range. Clandestine scanning is a possible threat because, the tag starts responding without even informing to the reader. Therefore, people who are carrying the objects with tags attached can be tracked with there unique numbers even though if the tag does not contain any personal information. The problem of privacy increases when the unique number is combined with the personal information.

Clandestine Inventorying: Clandestine inventorying occurs in EPC tags which contain the information about the manufacturer of the object, object code etc. So, any person carrying reader can say about what the other person is carrying and even try to get the personal information.

5.8 summary

Passive type RFID tags operate on high frequency (HF) bands (13.56 MHz) and supports the RFID standards ISO 14443/15693 protocols. The major applications of this type of tags are contactless smart cards, animal identification and so on.

The design of stream cipher must ensure some of the cryptographic objectives in-order to provide efficient and secure communication between the RFID tag and the reader. In HF RFID systems the data transmission rate is around 6 Kbps and 106 Kbps. Due to various stringent requirements for RFID tags, the design of cipher must contain less than 5000 GE. CMOS technologies like 130 and 180 nm can be used for fabricating the RFID tag and for the production of high frequency tags 130-250 nm are used. For a passive type tags the power consumption must be in the range 5 to 10 μA. Where as the upper bound of average current consumption of a cryptographic primitive is 15μA under CMOS 350 nm technology for an RFID.

Chapter 6

WG-5 Stream Cipher

In this chapter we discuss the implementation of WG-5 stream cipher. Section 6.1 describes the structure of WG-5 keystream generator, and Section 6.2 and 6.3 explain the various mathematical properties and the security analysis of WG-5 respectively. In Section 6.4, we describe the detailed architecture of WG-5 and in section 6.5, we measure the area and performace of WG-5 in 65 nm and 130 nm. In Section 6.6 RFID mutual authentication protocol based on WG-5 is proposed and its security and privacy analysis is discussed. Finally, in section 6.7 we compare the WG-5 implementation results with other stream and block ciphers.

6.1 WG-5 keystream generator

WG-5 keystream generator consists of 32 stages of LFSR and the elements of LFSR are over the field $GF(2^5)$ as shown in figure 6.1. The output of LFSR is connected to a nonlinear WG-5 transformation function ($WG5_{trans}$). The feedback polynomial of the LFSR is primitive over the field $GF(2^5)$ and is represented as $p(x) = x^{32} + x^{23} + x^{13} + \beta$, where β is root of the generating polynomial $g(x) = x^5 + x^4 + x^2 + x + 1$.

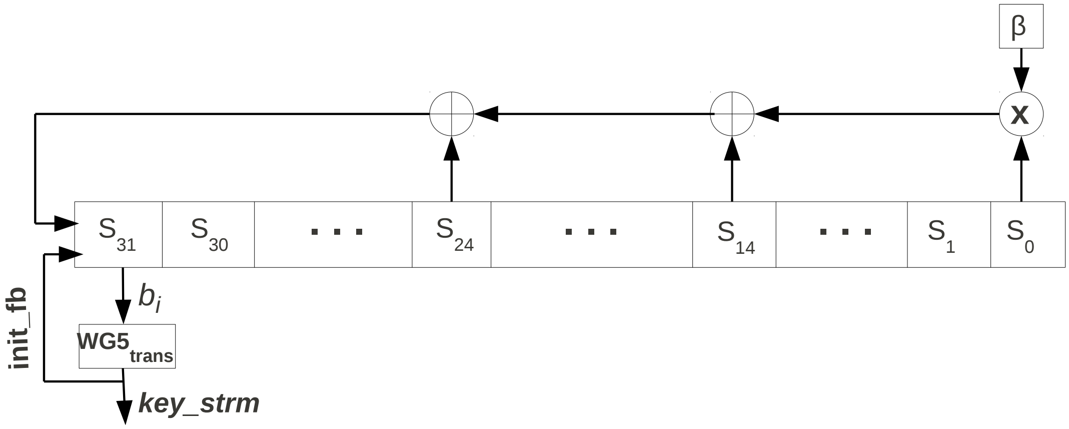


Figure 6.1: WG-5 Keystream generator

The nonlinear WG-5 transformation, $GF(2^5) \to GF(2)$, is applied to the output of LFSR to generate the keystream. The WG-5 transformation is calculated according to the definition discussed in section 4.5.3. Therefore, $f(x) = Tr(t(x)),$ where $t(x) = h(x+1)+1$ and $h(x) = x + x^5 + x^7,\ x \in GF(2^5)$. After ignoring the common coset leaders and common terms, we get $f(x) = Tr(x^7),\ x \in GF(2^5)$. If we consider the decimation of 11 for WG-5 transformation, then it is defined as $WG5_{trans}(x) = f(x^{11}) = f(x^{15}) = Tr(x^{15})\ x \in GF(2^5)$.

WG-5 keystream generator consists of key/IV initialization and keystream generation steps. During initialization step the cipher is initialized by loading the key and initial vector bits into the LFSR. It runs for 32 clock cycles with the initial feedback signal $(init\_fb)$ from WG-5 transformation. Each stage of the LFSR is denoted as S_i, where $0 \le i \le 31$. More precisely, let $S_0, S_1, ..., S_{31} \in GF(2^5)$ be the internal states of WG-5, then the output of LFSR is denoted by $b_i = S_i,$ where $i = 0, 1, ..., 31$. The nonlinear recursive relation of b_i can be represented as $b_i = b_{i-14} + b_{i-24} + \beta b_{i-32}$.

After key is initialized, the initial feedback signal (init_fb) is disconnected from LFSR to produce the keystream bits. To generate keystream bits, the contents of LFSR first stage S_{31} is given to $WG5_{trans}$ which gives outputs of 1-bit after each clock cycle. More precisely, at each clock cycle the contents of LFSR is shifted right and the updated value of stage S_{31} is fed to the $WG5_{trans}$ to generate the running keystream bits (key_strm). Thus, the running keystream bits is bitwise XORed with the plaintext to generate the ciphertext.

6.2 Cryptographic properties of WG-5

The keystream generated by WG-5 has period of $2^{160} - 1$ and it possess balanced property. It is proven that the boolean functions used in construction of WG transformation has 1-order resiliency property [36] which can be immune to correlation attacks. In addition, it is also proven that the WG transformation has an orthogonal transformation [36] thus, when $WG5_{trans}$ is combined with the output of LFSR the resultant sequence is the ideal two level autocorrelation sequence. The keystream generated by WG-5 generator is a GMW sequence. The internal state of LFSR in WG-5 is 160 bits in length which is twice the size of the secret key. WG-5 has acceptable linear complexity to consider it as a lightweight cryptosystem. Given, $WG5_{trans}$ (x)= $Tr(x^7), x \in \mathbb{F}_{2^5}$, the linear complexity can be computed as *LC* $= n \times \sum\limits_{i \in I} l^{w(i)} = 5 \times \sum\limits_{i \in I} 32^{w(i)} \approx 2^{17.32}$ where, index set $I = \{7\}$ and l is the number of internal states of the LFSR. If we consider the decimation of 11 then the LC can be increased to *LC* $\approx 2^{22.32}$ with index set $I = \{15\}$. But the 11-decimation does not provide 1-order resiliency property.

6.3 Security analysis of WG-5

Security level is an important parameter that needs to be considered while designing the stream cipher. In this section, we present the extensive security analysis of WG-5 cipher based on the measured linear complexity values given in the previous section. Our results demonstrate that, in order to attack WG-5 cipher an adversary requires $2^{17.\ 32}$ or $2^{22.32}$ consecutive keystream bits. Here we show that, WG-5 cipher is secure against various attacks such as algebraic, correlation, cube, differential, discrete Fourier Transform and time-memory-data trade-off.

A. *Algebraic attacks:* It is shown previously that the stream ciphers based on LFSR are potentially vulnerable to algebraic attacks [68]. We calculated the algebraic immunity of WG-5(**x**) as 3 and according to [68], the time and data complexity of algebraic attack that recovers the internal state of the WG-5 generator were found to be $7/64$ **.** $\binom{160}{3}^{\log_2 7} \approx 2^{51.13}$ and $\binom{160}{3} \approx 2^{19.35}$ respectively which suggests that WG-5 is resistance to algebraic attacks. One more type of algebraic attack is the fast algebraic attack in which the adversary needs even more keystream bits than the previous algebraic attack [69].

B. *Correlation attacks:* This attack on synchronous stream ciphers are based on the correlation between the keystream bits and the LFSR output bits. To analyze correlation attacks on WG-5 cipher, the nonlinearity and resiliency property needs to be considered. We calculated the non linearity of WG-5 transformation as 12 based on [36]. The output of WG-5 transformation or keystream bits in WG-5 has 1-order resiliency property, which means that the keystream bits are not correlated to any single bit of the LFSR output. This explains that correlation attacks on WG-5 is not feasible.

Fast correlation attack [70] is a type of correlation attack in which the decoding problem can be efficiently solved by Maximum Likelihood (ML) decoding algorithm to retrieve the internal state of LFSR. The complexity of this attack on WG-5 cipher was estimated on the basis of the theoretical bounds given in [70]. Let h be a linear function such that hamming weight of h(x) XOR WG5(x) is minimum. Then, the probability of producing the same output for a given input x is $P[WG5(x) = h(x)] = \frac{2^5 - N_{WG5}}{2^5} = 0.625$. For the successful attack, the amount of keystream required on the basis of results given in [70] and with parameter $t = 3$ is $N \approx 1/4 \cdot (k \cdot 12 \cdot ln2)^{1/3} \cdot \epsilon^{-2} \cdot 2^{\frac{l-k}{3}}$ and the decoding complexity is given by $C_{dec} = 2^k \cdot k \cdot \frac{2ln2}{2\epsilon^6}$, where $l = 160$ is the size of the internal state of the LFSR in bits, $\epsilon = P[WG5(x) = h(x)] - 0.5 = 0.125$ and k is the number of internal state bits recovered. For lower $(k = 3)$ and higher $(k = 80)$ values of k, the amount of keystream required is approximately $2^{57.85}$ (not achievable) and $2^{35.73}$ respectively and the decoding complexity is approximately $2^{98.76}$, which surpass the complexity of exhaustive key search. The above analysis suggests that the WG-5 is secure against fast correlation attack.

On the other side, if WG-5 transformation is consider by 11-decimation then, the keystream generated by it does not provide 1-order resiliency property. But the above calculation results show that the required keystream bits N and the decoding complexity C_{dec} of correlation attacks required to attack WG-5 are much higher and infeasible to achieve. Therefore we can say that using 11-decimation function WG-5 is secure against fast correlation attacks.

C. *Cube attack:* It was first proposed in [71] and it is based on the degree of the polynomials used for the design of stream ciphers. Since, in WG-5 stream cipher, the degree increases very fast after $2l$ clock cycles (l is the length of the LFSR). Thus, after 64 clock cycles the degree is high and to successfully to launch the cube attack the degree should be low. Therefore, we suggest that cube attack is not possible on WG-5.

D. *Differential attack:* Study of differential cryptanalysis in [72] is that, any differences at the input (key or the plaintext) side will lead to a predicted difference at the output (keystream), which can help to analyze the internal state of the stream cipher. Differential cryptanalysis is applicable only when the cipher has linear behavior. In WG-5, during the key/IV initialization step the LFSR runs for 64 clock cycles and the output of LFSR is filtered by a nonlinear WG transformation. For $32 \leq i \geq 95$, the output of the LFSR is $b_i = WG(b_{i-1}) + (b_{i-14}) + (b_{i-24}) + \beta b_{i-32}$, if any bit change at the input side of LFSR, then after 32 clock cycles S_{31}

will be affected which will lead to subsequent change in the internal state of the LFSR. Single bit of the LFSR will influence any cell in the LFSR internal state. Therefore, differential attacks are infeasible in WG 5 due to its nonlinear transformations.

E. *Discrete Fourier Transform attack (DFT):*

   In general, the idea of DFT attack is that the adversary tries to recover the internal state of the stream cipher by using DFT sequences. Initially in [73], the authors proposed an efficient algorithm to recover the internal state of the cipher without using the DFT sequence in specific scenarios. For any keystream generator with m-sequence of period $2^n - 1$ based on its boolean function , the complexity of attack is $O\,(D)$ after observing D keystream bits. While its pre-computation complexity is $O(D(\log_2 D)^3)$. Furthermore, in [74] they extended this attack to the finite field $GF(2^m)$ with similar complexity values. For WG-5, the adversary requires to achieve $2^{17\cdot\,32}$ keystream bits with a complexity of $O\,(2^{17.32})$ and pre-computation complexity of $O\,(2^{23\,\cdot\,44})$. Additionally, with increased linear complexity of WG-5 the complexity of this attack would be $O\,(2^{22.32})$ and pre-computation complexity of $O\,(2^{35\,\cdot\,76})$ which is impossible to achieve. This suggests that DFT attacks are infeasible in WG-5.

F. *Time-Memory-Data Trade-off attack:* The attack was discussed in [75]. The adversary implements this attack in two phases. In the first phase the adversary analyses the design architecture of the stream cipher and stores the analysis results in the form of hard disks or tables. In the second phase the adversary collects the stream cipher data (keystream) for an unknown key. Now, by knowing the size of the tables as well as data and time required, the adversary tries to recover the internal state of the stream cipher. In WG-5 the internal state is 160 bits which is twice the size of the secret key and the complexity of this attack is $O\,(2^{80})$ which is impractical to attain. This showed that, the WG-5 is secure against time-memory-data trade off attacks.

6.4 WG-5 in hardware

In this section, we discuss detailed architecture of WG-5 as shown in figure 6.2. The hardware implementation of WG-5 consists of datapath and control circuitry. Further, the datapath implementation consists of WG-5 transformation and the control circuitry implementation consists of FSM and LFSR. The FSM is used to generate the control signals for changing the WG-5 cipher configuration that is loading key into the registers, initialization phase and run or keystream generation phase. Note that, in addition to 32-stages of LFSR, we divided WG-5 transformation block into two; WG-5 core and trace function because during initialization phase WG-core signal $(init\_fb)$ is fed as one of the inputs to the LFSR.

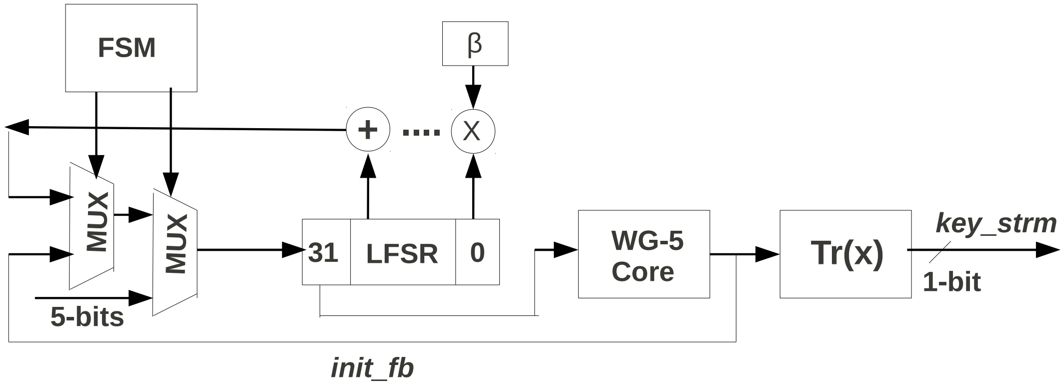


Figure 6.2: Architecture of WG-5

6.4.1 WG-5 Core and trace function

In this section, we discuss the implementation of datapath circuitry; WG-5 core and trace function. It is a combinational circuitry which consists of AND gates, XOR gates and bitwise shifting (re-wiring). The important building components in WG-5 core are, multiplier and squaring.

As we have already discussed in chapter 4, the elements of WG-5 transformation can be represented either in polynomial or normal basis. We used normal basis representation for implementing WG-5 core. Based on normal basis, different multiplier architectures exists as we have discussed in literature survey in section 3.4. For the purpose of WG-5 implementation, we used bit parallel normal basis multiplier architecture which utilizes 40 XOR gates and 25 AND gates. While squaring implementation is normal basis is just a simple bit wise shifting.

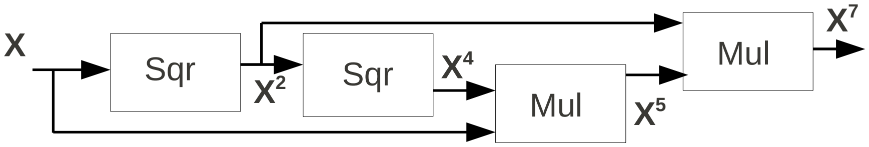


Figure 6.3: WG-5 Normal basis circuit

The design goal of WG-5 core block is to generate the term x^7 which we calculated according to $WG5_{trans}$ definition explained in section 6.1. The architecture of WG-5 core in normal basis is shown in figure 6.3. So to generate x^7 we used squaring and multiplication operations and it requires two squaring and 2 multiplication operations.

Thus, we completed the datapath implementation by connecting the output of WG-5 core signal $(init\_fb)$ which is 5-bits to the trace function and also fed the same signal to the LFSR input. The trace function implementation is a sequence of XOR gates which adds all 5 bits and outputs 1-bit. Current section focused only on datapath implementation of WG-5 cipher. In the next section we explain the control circuitry implementation details.

init	load	WG-5 phase	input at register S_{31}
0	0	keystream generation	$lfsr\_fb$
0	1	loading registers	5-bit input
1	1	loading registers	5-bit input
1	0	initialization	$lfsr\_fb \oplus init\_fb$

Table 6.1: Configuration of WG-5 based on $init$ and $load$ signals

6.4.2 Linear feedback shift register

In this section, we discuss the design of control circuitry of WG-5. First, we introduce the implementation of LFSR and than we explain its operational behavior based on control signals.

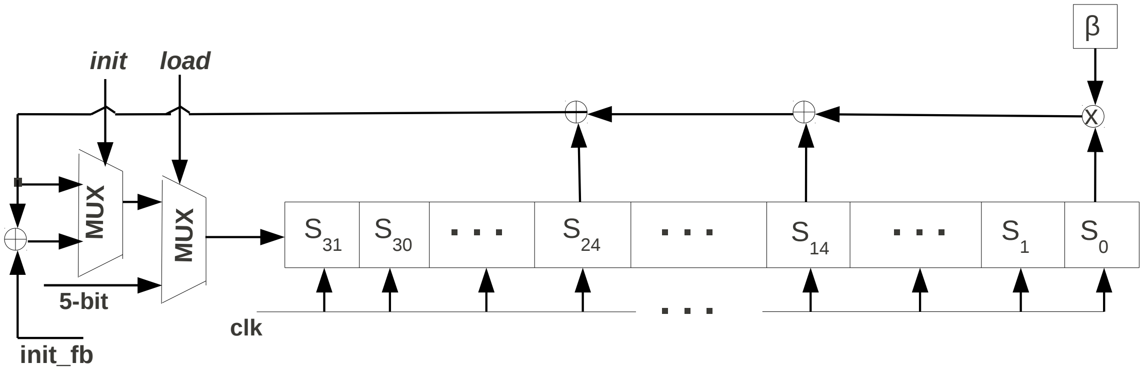


Figure 6.4: WG-5 LFSR

We implemented 32-stages of LFSR in WG-5 as shown in figure 6.4. In addition, we introduced two sets of multiplexers at the input side of the LFSR. The multiplexers are used to change the configuration of the LFSR with the help of two control signals $init$ and $load$. The configuration of two control signals are shown in table 6.1. The β element at the feedback of the LFSR is a constant element which is multiplied to the output of LFSR stage S_0 and later XORed with the taps as shown in figure 6.4. Thus, the total number of inputs to the LFSR are five; they are clock (clk), 5-bit input for loading the registers, two control signals $init$, $load$ to change the configuration of LFSR and the initial feedback signal ($init\_fb$) coming from the WG-5 core. Based on values of $init$ and $load$ control signals at the input side of LFSR, the phase of WG-5 cipher is executed.

Furthermore, for loading key and initial vector bits into registers from S_{31} to S_0, serial or parallel based design can be adopted. In WG-5, we used serial based design for loading all 32 stages of LFSR because, it utilizes 5 multiplexers at the input side of LFSR and it loads every clock cycle. So in total it takes 32 clock cycles to fully complete loading. Whereas, parallel based loading requires 160 (5 bits × 32 LFSR stages) multiplexers, because each stage requires 5 multiplexers and it takes one clock cycle to load all 32 stages. Since, loading phase occurs only once at the beginning of WG-5 cipher operation, the utilization of 32 clock cycles by serial based loading can be an acceptable trade off.

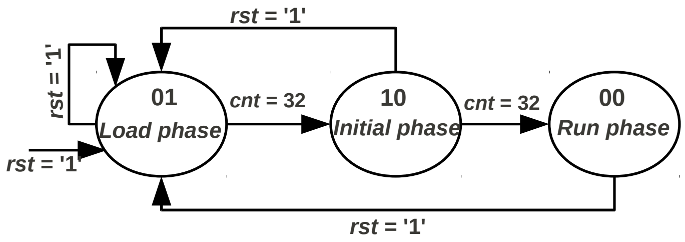


Figure 6.5: WG-5 Finite State Machine Implementation

6.4.3 Finite state machine (FSM)

By designing the finite state machine we completed the design of control circuitry. The purpose of FSM is to generate two control signals $init$ and $load$ for the LFSR which changes the configuration of WG-5 cipher. It consists of 3-states and its transition diagram is shown in figure 6.5. The FSM design contains a counter to count the number of clock cycles being used at each phase of WG-5 cipher and encoded the 3-states (LOAD_PHASE, INITIAL_PHASE and RUN_PHASE) with 2-bit vector, where, the left bit indicates the $init$ signal and right bit indicates the $load$ signal as shown in table 6.1.

When reset signal rst is set to 1, FSM starts its operation as shown in figure 6.5. When rst is 1 the state machine enters the state LOAD_PHASE and the counter signal cnt resets to 0. As shown in table 6.1, during this phase the $init$ is 0 and $load$ is 1, which loads the first 5-bits into the stage S_{31} of LFSR. Moreover, in LOAD_PHASE, the cnt starts incrementing by 1 at every clock cycle and when it reaches the value 32 the state machine makes transition to next state INITIAL_PHASE because loading the bits into all 32-stages of LFSR registers takes 0 to 31 clock cycles.

Once, the state machine reads INITIAL_PHASE state the $init$ and $load$ values as 1 and 0 respectively as shown in table 6.1. During this phase, the input to the first stage S_{31} of LFSR is the addition of LFSR feedback signal $lfsr\_fd$ and the WG-5 core initial feedback signal $init\_fb$. In this phase, the counter counts 32 clock cycles and after 32 clock cycles, the state machine makes transition to next state RUN_PHASE.

In RUN_PHASE state the $init$ and $load$ values are 0 and the state is called as keystream generation. During this phase the input to first stage S_{31} of LFSR will be the LFSR feedback signal $lfsr\_fd$. In this state, the counter becomes idle and the finite state machine stay in this state until the reset signal rst is set to 1. Moreover, the keystream is generated during this state and other states LOAD_PHASE and INITIAL_PHASE are used for key initialization of the WG-5 cipher. Note that in FSM, if rst is set to 1 then the state machine makes transition to LOAD_PHASE irrespective of the current state and the counter resets to value 0.

6.5 Area and performances results

This section summarizes the implementation results of WG-5 cipher over the field $GF(2^5)$. The length of cryptographic secret key and initial vector (IV) are 80-bits. The implementation of WG-5 cipher was carried out using CMOS 65 nm and 130 nm technologies.

The results of area, performance and power tabulated in table 6.2 which were obtained for ST microelectronics 65 nm and 130 nm cell library using Synopsys Design Compiler for logic synthesis. The results obtained are after the place and route. The area is calculate in terms of gate equivalents (GE), in 65 nm one GE is equivalent to a 2-input NAND gate value of 1.5600 μm^2 whereas in 130 nm one GE is 9.9792 μm^2. The power is measured in terms of Milli Watts (mW) with respect to the clock frequency of 1 GHz. The performance is calculated as speed × bits over clock cycles. Optimality is ratio of throughput to Area×power.

	65 nm				**130 nm**			
	Area (GE)	Power (mW)	Perf	Opt	Area (GE)	Power (mW)	Perf	Opt
WG-5	1493	1.76	6.25	0.38	2219	0.86	6.25	0.52
WG-5(Deci)	1666	2 .198	6.25	0.27	2492	0.92	6.25	0.43

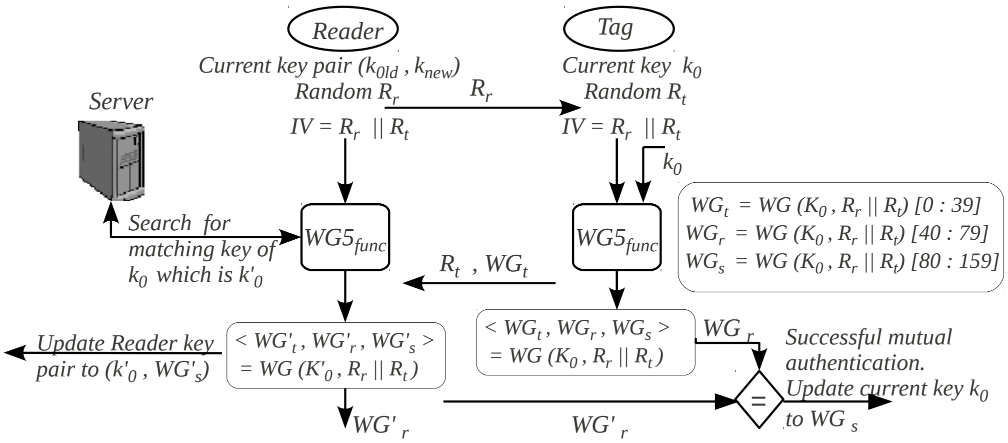
Table 6.2: Area and performance of WG-5

From the above results we conclude that WG-5 cipher is a lightweight stream cipher and it can be used for RFID applications. In the next section, we propose RFID mutual authentication protocol based on WG-5 cipher design.

6.6 WG-5 in RFID mutual authentication protocol

In this section, we discuss mutual-authentication protocol based on WG-5 stream cipher. The proposed protocol is in fact an instance of the private and efficient protocol discussed in [19]. To avoid the DoS attacks, the server stores and updates for each and every tag, a pair of potential current key pair (K_{old}, K_{new}) for the tag. The protocol acts as follows; First we assume that the RFID system consists of RFID reader, tag and the server. Tag carries 80-bit secret key k_0 and a unique ID_i.

The communication of the protocol works as follows and is shown in figure 6.6.



Reader
Tag
Current key pair (k0ld , knew)
Random Rr
Rr
Current key k0
Random Rt
Server
IV = Rr || Rt
IV = Rr || Rt
k0
WG5func
WG5func
WGt = WG (K0 , Rr || Rt) [0 : 39]
WGr = WG (K0 , Rr || Rt) [40 : 79]
WGs = WG (K0 , Rr || Rt) [80 : 159]
Search for matching key of k0 which is k'0
Rt , WGt
Update Reader key pair to (k'0 , WG's)
< WG't , WG'r , WG's > = WG (K'0 , Rr || Rt)
< WGt , WGr , WGs > = WG (K0 , Rr || Rt)
WG r
=
Successful mutual authentication.
Update current key k0 to WG s
WG' r
WG' r

Figure 6.6: RFID Mutual authentication protocol using WG-5

Step1: The reader sends request along with an 40-bit random nonce R_r to the tag.

Step2: Upon receipt of R_r the tag generates random nonce R_t of length 40-bit, computes the initial vector $IV = R_r \parallel R_t$. Now tag uses IV and 80 bit secret key k_0 as inputs to the $WG5_{func}$ and generates the keystream bits of length 160 bits. The tag outputs WG_t of length 40 bits and sends (WG_t, R_t) to the reader.

Step3: The reader tries all tag ID_i and key k_i stored at the server end until it finds $WG\,'_t = WG_t$. Generally, the reader checks the MSB bit of the generated keystream sequence, if it matches, then it checks the further keystream bits of sequence. Suppose MSB bit is not matching then it stops generating further keystream bit sequence and selects the new key.

Step4: Upon authentication of the tag, the reader updates the current pair associated with tag to $(k_0', WG\,'_s)$. In order to authenticate reader, it generates $WG\,'_r$ of length 40-bits and sends it to tag.

Step 5: Once, the tag receives $WG\,'_r$, the tag checks whether $WG_r = WG\,'_r$, if it is then, the reader is authenticated successfully. After reader is authenticated, the tag updates its current key value k_0 to WG_s. Now the reader and tag are said to be mutually authenticated.

6.6.1 Security and privacy analysis of the current protocol

In this section we discuss various security and privacy analysis of our protocol.

6.6.1.1 Man in the middle attack

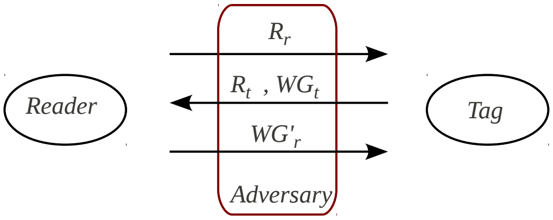


Figure 6.7: Man-in-the-middle attack

An adversary can listen and modify the communication between a genuine tag and the reader as shown in figure 6.7. Thus, it has access to the R_r, R_t (random challenges which form the IV) and $WG_r(K, IV)$, $WG_t(K, IV)$ (the responses from the reader and the tag respectively). But the ID of the tag and the secret part of the WG-5 stream cipher, i.e. WG_s are never exchanged, not even in the encrypted form. We cannot stop the adversary from knowing whether tag authentication was successful or not. If the reader sends $WG\,'_r$, then the adversary knows that the tag has been authenticated.

6.6.1.2 Replay attack

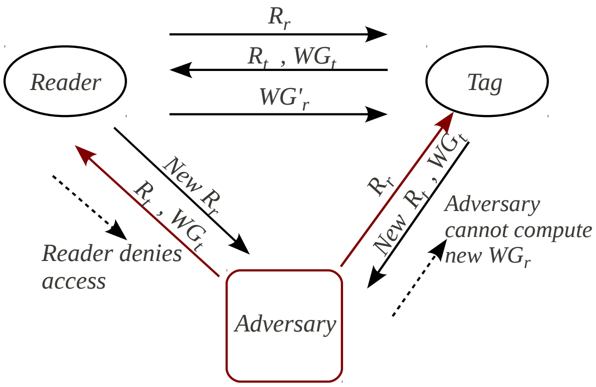


Figure 6.8: Replay attack

In previous section, it is clear that, how the adversary may launch a man in the middle attack and observe the exchange of bits between a genuine tag and the reader. Even it can also determine whether the authentication was successful and key was updated or not. Since, the server stores a pair of old and new key values, the adversary may try to attack the previously authenticated values i.e. R_t and WG_t and try to use them for future communications. To overcome this issue, the reader will provide the random nonce R_r and $IV = R_r \parallel R_t$ so that the IV will change. As a result of that the value of WG_t stored with the adversary will be rendered useless. The communication of this attack is shown in figure 6.8.

6.6.1.3 Impersonation

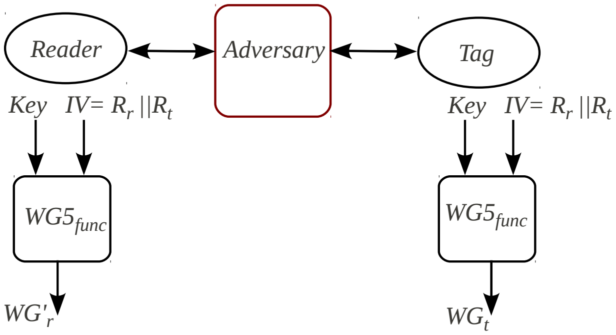


Figure 6.9: Impersonation attack

An adversary will try to impersonate a tag or a reader while communicating with a genuine reader or a genuine tag as shown in figure 6.9. To do so, it has to compute the tag response WG_t or the reader response $WG\,'_r$ which are required to confirm the authentication. Thus, without knowing the key, it is impossible for the adversary to compute these values.

Since direct attacks will not be effective, the adversary will try to run it in two phases. In the first phase it will collect some values of WG_t and $WG\,'_r$, which will be used to construct a distinguisher D. The queries and responses will be stored by the distinguisher D. Then the final authentication attempt will be done by the adversary and for this it will use the distinguisher D to predict the required WG-5 response.

Most of the distinguisher's are differential or algebraic in nature. A differential attack would require at least $2^{17.32}$ or $2^{22.32}$ continuous bits from the stream cipher output. Our protocol doesn't exchange the last 80 bits of every output hence, we believe that the differential distinguisher will not work against our protocol. Algebraic distinguisher's rely on the initialization outputs of the stream cipher and in practical applications, initialization output will be suppressed. Thus, the algebraic attacks will be rendered ineffective against our protocol.

Overall, WG-5 function has the randomness property i.e. no distinguisher can distinguish it from a random stream of bits. Thus, a two phase impersonation attack is unlikely to succeed.

6.6.1.4 Tampering

An adversary can tamper with the tag and know its internal state which is the key of the WG cipher function. In the worst case scenario, even if the adversary knows this key, it cannot predict any transaction it observed in the past (and its WG-5 function). This is because of the fact that, after every successful authentication the key gets updated thus, for any authentication which occured in the past, the adversary has no means of computing the key.

6.6.1.5 Forward privacy

We observe that, an adversary will be able to link the tags internal state with the immediate failed authentication. This means that, an adversary may be able to track the tag after tampering, as long as it doesn't have a successful authentication with a genuine reader. Once that is done, the key will be updated and adversary will once again lose the track of the tag.

An adversary may observe two tags of the system at a given time, saving the random challenges and computed responses. If the tags go out of range and then one of the tags interacts with the adversary, the adversary cannot distinguish between the two. Nor, it can even predict (or confirm) if the tag is actually one of the two. This is supported by the pseudo-randomness property of the WG-5 function.

6.6.1.6 Denial of service(DoS)

An adversary acting like a man in the middle can block communications from either the tag or the reader to simulate a denial of service. But we believe that, the exposure time is bounded by T, that is after some time the adversary will be unable to interfere and the tag will have a successful authentication with a genuine reader. Rather than temporary DoS, the adversary would try to remove the tags key from the pairs stored on the server's side.

6.6.1.7 De-synchronization

A man in the middle adversary will wait for the reader to authenticate the tag and send its response i.e. $WG\,'_r$. The tag will update its key from K_o to $K_1 = WG_s$ only if the reader sends the right response. The adversary may drop $WG\,'_r$ and send some dummy response as shown in figure 6.10 (a). The tag will not be able to verify the reader and update its key. Thus, the key pair with server reads (K_0, K_1) but the tag still stores ID K_0 in its internal state.

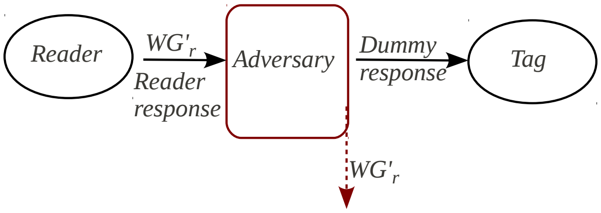


Figure 6.10: De-synchronization attack

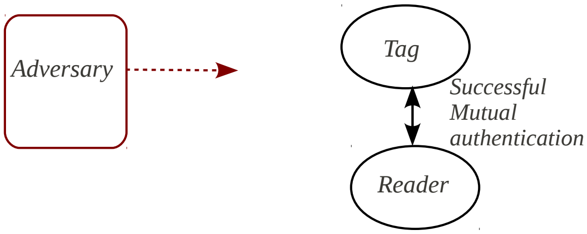


Figure 6.11: Tag recovery from attack

This still doesn't desynchronize the tag from the system. If the tag has an authentication exchange with a genuine reader, it will be able to identify the tag, thanks to the old key K_0 being stored in the server key pair (K_0, K_1). Once the exchange is successful as shown in figure 6.11, the tag key will be updated to K_2 and the server key pair will be updated to (K_0, K_2). Notice that the server realizes K_1 was never used by tag and discards it.

If the second exchange is again overtaken by an adversary and reader's response is corrupted, the tag's key is kept at K_0 and the server key pair becomes (K_0, K_2). Hence the tag is still operable and can make the server discard the key which was never used. The only way to de-synchronize the tag is to have two successful impersonation attacks, which we have seen before, is hard to achieve with WG protocol.

Protocol	Forward privacy	DoS	Impersonation	Replay attack	Mutual Authentication
S-Protocol [76]	No	Yes	Yes	No	Yes
EMAP [77]	Yes	No	Yes	No	Yes
OSK [78]	No	No	Yes	Yes	Yes
PEP [79]	No	No	Yes	Yes	Yes
PEPS (WG-5)	Yes	Yes	Yes	Yes	Yes

Table 6.3: WG-5 Protocol comparison

Therefore, in this section we proposed that, the WG-5 stream cipher based mutual authentication protocol ensures both security and privacy in RFID tags. The protocol is resistant to DoS, impersonation, replay attacks and also ensures forward privacy. Table 6.3 shows the comparison between WG-5 protocol with other existing protocols.

6.7 Comparison of results

In this section, we summarize and compare the ASIC performance of WG-5 implementation, with other stream and block cipher design results. Most of the stream ciphers that are mentioned in tables 6.4 and 6.5 were submitted to state-of-the-art stream cipher conferences (SASC) [80] [60] [81] and eSTREAM project.

Table 6.4 shows the comparison of WG-5 implementation results and its comparison with other hardware stream and block cipher designs, which were proposed for RFID tags. For the comparison purpose, we analyzed the cipher design which has the highest level of security of 80 bit key size and below. In addition to the standard design algorithm AES-128, in which the security is well examined, the new stream cipher design of Grain and Trivium were also selected for comparison. The other block ciphers KATAN and KTANTAN both consists of block sizes 32, 48 and 64 bits and 80 bit key size were also considered for comparison. Whereas, the mCrypton cipher has 64 block size bits with 80 bit key size. Furthermore, we compared other parameters like the chip area, power consumption, bits per clock cycle, clock frequency, throughput and optimality. The results of WG-5 chip area is based on synthesis and are denoted in gate equivalents (GE). For 130 nm CMOS technology, one gate equivalent is compared to a 2-input NAND gate of 9.9792 μm^2. The power consumption mentioned in the fourth column is denoted in μW which is calculated at a clock frequency of 100 kHz and a supply voltage of 1.5 V. The throughput is calculated for 100 khz and it is measured as bits per cycle multiplied with clock frequency. The optimality of WG-5 and other cipher is measured as the ratio of throughput / Area × power.

The comparison table shows that most of the selected stream ciphers have achieved their design goals requirement for passive type RFID tags. Grain-80 has less chip area compared to WG-5. Thus, reduces the overall chip area in terms of GE in Grain-80 whereas, in WG-5 we have chosen 80-bits for both secret key and IV bits which increases the number of gate equivalents for the same reason as above. The comparison of other stream ciphers Trivium-80, F-FCSR-H-80, Decim-80, Edon80, Pomaranch80, Mickey2(80) with WG-5 shows that they require more number of registers for storing the internal state of the cipher. Although, the power consumption of WG-5 is higher than the Grain-80, Trivium-80 and Decim-80 but, it is well below the required goal power consumption for passive type RFID tags. Compared to WG-5, block ciphers KATAN and KTANTAN have less chip area because in KTANTAN the key is burnt on device permanently and it is unchangeable. Moreover, KTANTAN is cryptographically not secure because it is vulnerable to man-in-the-middle attack [82]. The optimality of WG-5 is comparatively higher than Trivium-80 and lower than Grain-80 this is due to the selection of 80-bit secret key and 64-bit IV. This suggests that Grain-80 has achieved higher optimality than WG-5 by compromising its security level.

Table 6.5 shows the other related stream and block cipher results over 90, 180 and 350 nm CMOS and FPGA technology.

Cipher	Chip area(GE)	Power(μW) @100kHz	Throughput(Kbps) @100kHz	Optimality
Grain-80 [60]	1294	3.3	100	2.34
Trivium-80 [60]	2599	5.6	100	0.68
AES-128 [62]	5398	–	237	–
AES-128 [62]	3400	–	1	–
F-FCSR-H-80 [62]	4760	10.58	800	1.58
Decim-80 [62]	2603	5.43	25	0.17
Edon80×4 [62]	4969	10.49	5	0.01
Edon80pl [62]	13010	25.05	100	0.03
Pomaranch80 [62]	5357	16.13	100	0.11
Mickey2(80) [62]	3188	7.10	100	0.44
KATAN-32 [83]	802	381 (nW)	12.5	4.10
KATAN-48 [83]	927	439 (nW)	18.8	4.54
KATAN-64 [83]	1054	555 (nW)	25.1	4.29
KTANTAN-32 [83]	462	146 (nW)	12.5	18.53
KTANTAN-48 [83]	588	234 (nW)	18.8	13.66
KTANTAN-64 [83]	688	292 (nW)	25.1	12.49
mCrypton-64 [84]	2420	–	–	–
WG-5	1729	6.03	100	1.00
WG-5 (Deci)	1838	6.18	100	0.9

Table 6.4: Comparison of WG-5 with other Hardware ciphers in 130 nm

Cipher	Technology	Chip area (GE)	Current(μA), Power(μW) @100kHz	Throughput
STREAM CIPHERS				
Grain-80 [12]	0.35 μm	3360	$0.80\mu A$	–
Trivium-80 [12]	0.35 μm	3090	$0.68\mu A$ –	–
Edon-80 [85]	0.35 μm	2922	–	2.18 Mbps
Quad-128 [86]	90 nm (Virtex 4 Xilinx FPGA)	2961GE (85 slices)	–	–
E0 [87]	Xilinx FPGA	1902	0.77 mW	93 Mbps
BLOCK CIPHERS				
AES-128 [64]	0.35 μm	3400	$4.5\mu W$	9.9 Mbps
AES-128 [65]	0.25 μm	3900	$1.94\mu A$	–
Present [88]	0.35 μm	999.52	$3.39\mu A$	11.4kbps
Present [89]	0.18 μm	1570	$5\mu A$	200 kbps
DES-64 [56]	0.18 μm	2309	1.19 μA	5.55 Kbps
DESL-64 [90]	0.18 μm	1848	0.89 μA	5.55 Kbps

Table 6.5: Results of other block and stream ciphers

Chapter 7

Conclusion

In this thesis, we proposed the design of lightweight stream cipher WG-5 which is cryptographically secure and can be used for a resource-constrained applications such as low-cost RFID tags. As it was mentioned earlier that, passive type RFID tags have stringent requirements in terms of chip area and available power supply. The constrained design goals need more attention towards the tailored security solutions for passive type RFID tags. Thus, the lightweight cryptography systems are in high demand.

In chapter 6, we implemented the lightweight stream cipher WG-5 which is a variant of the WG stream cipher. Since WG cipher contains cryptographic properties, we measured few of the values related to WG-5 such as period and linear complexity (LC). The LC of WG-5 was $2^{17.32}$ and it was increased to $2^{22.32}$ by using 11- decimation value in oder to achieve higher cipher complexity. Based on both LC values we carried out the security analysis of WG-5 in detail. The analysis showed that, WG-5 is secure against algebraic, correlation, cube, differential, discrete Fourier Transform and time-memory-data trade-off attacks.

Further, the implementation of WG-5 was carried out using 65 and 130 nm CMOS technology. For the purpose of WG-5 core implementation, normal basis representation was used to make squaring operation simpler because in normal basis it is just a wire shifting. For the multiplication operation parallel based multiplier which is purely combination circuitry consists of AND gates and XOR gates was used. The number of parameter such as chip area, power, throughput and optimality were calculated for WG-5 using with and without decimation functions at $100KHz$ clock frequency in 130 nm CMOS technology. The chip area of WG-5 was found to be 1838 GE with decimation and 1729 GE without decimation function whereas, the power consumption was 6.03 and 6.18 μW respectively. The optimality of WG-5 was calculated as 1.00 and 0.9 respectively. Compared to the WG-5 core and FSM, the 32-stage fibonacci feedback style LFSR utilized more number of hardware resources in terms of area and power. The implementation results of WG-5 in 130 nm showed that, it meets the design goals for the passive type RFID tags. Further, we compared 130 nm results with other existing stream and block ciphers which were submitted to eSTREAM 2007 and SASC. This comparison shows that, the optimality of WG-5 is more than Trivium-80, F-FCSR-H-80, Decim-80, Edon80pl, Edon80×4, Pomaranch80, Mickey2(80) and the security level of WG-5 is more than Grain-80, Katan and Ktantan. This clearly suggests that taken together both the optimality and security level WG-5 outperforms many of the ciphers.

Next we proposed, the RFID mutual authentication protocol based on WG-5 stream cipher in section 6.6. The security and privacy analysis of the protocol showed that, it is resistant to DoS, impersonation and replay attacks. The comparison of our protocol with other existing protocols suggested that, it offers mutual authentication and ensures forward privacy. Putting together, implementation of WG-5, its verified security analysis and the proposed RFID protocol suggest that, WG-5 is a promising candidate for the passive type RFID tags.

7.1 Future work

It was noticed that, the power consumption of WG-5 is higher compared to the top stream cipher contenders Grain and Trivium. Future work can concentrate on reducing the power consumption of WG-5 by using one of the optimization techniques such as clock gating. Furthermore, for the proposed mutual authentication protocol using WG-5 we can measure the tag computation such as transmission rate, response time and writing the data into tags memory which influences the cost of the tag. Therefore, computations of WG-5 must be minimized in order to reduce the tag computation.

Bibliography

[1] Adhiarna, N. and Jae-Jeung Rho, “Standardization and global adoption of radio frequency identification (RFID): Strategic issues for developing countries,” in *Computer Sciences and Convergence Information Technology, 2009. ICCIT ’09. Fourth International Conference on*, pp. 1461–1468, nov. 2009.

[2] S. F. Wamba, L. A. Lefebvre, Y. Bendavid, and E. Lefebvre, “Exploring the impact of RFID technology and the EPC network on mobile B2B ecommerce: A case study in the retail industry,” *International Journal of Production Economics*, vol. 112, no. 2, pp. 614–629, 2008. Special Section on RFID: Technology, Applications, and Impact on Business Operations.

[3] Hagl, Andreas and Aslanidis, Konstantin, “RFID: Fundamentals and applications,” in *RFID Security*, pp. 3–26, Springer US, 2009. 10.1007/978-0-387-76481-8_1.

[4] EPC global Inc., “EPC generation 1 tag data standards version 1.1 rev.1.27.” `http://www.gs1.org/gsmp/kc/epcglobal/tds/tds_1_1_rev_1_27-standard-20050510.pdf`, May 2005.

[5] Juels, A., “RFID security and privacy: a research survey,” *Selected Areas in Communications, IEEE Journal on*, vol. 24, pp. 381–394, feb. 2006.

[6] Yassir Nawaz and Guang Gong, “WG: A family of stream ciphers with designed randomness properties,” *Information Sciences*, vol. 178, no. 7, pp. 1903–1916, 2008.

[7] C. Lam, M. Aagaard, and G. Gong, “Hardware implementations of multi-output welch-gong ciphers,” tech. rep., Department of Electrical and Computer Engineering University of Waterloo, Waterloo, Ontario N2L 3G1, CANADA, 2009.

[8] Y. Luo, Q. Chai, G. Gong, and X. Lai, “A lightweight stream cipher WG-7 for rfid encryption and authentication,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–6, dec. 2010.

[9] S. Sarma, “Towards the 5 cent tag,” *White paper, Auto ID center*, November 2001.

[10] Garcia-Alfaro, Joaquin, Barbeau, Michel, and E. Kranakis, “Security threat mitigation trends in low-cost RFID systems,” in *Data Privacy Management and Autonomous Spontaneous Security*, vol. 5939 of *Lecture Notes in Computer Science*, pp. 193–207, Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-11207-2_15.

[11] Feldhofer, Martin and Wolkerstorfer, Johannes, “Hardware implementation of symmetric algorithms for RFID security,” in *RFID Security*, pp. 373–415, Springer US, 2009. 10.1007/978-0-387-76481-8_15.

[12] Feldhofer, Martin, “Comparison of low-power implementation of trivium and grain,” In Work- shop on The State of the Art of Stream Ciphers (SASC 2007), January 31–February 1, 2007, Bochum, Germany, pp. 236– 246, ECRYPT, February 2007, 2007.

[13] C. Paar, A. Poschmann, and M. J. B. Robshaw, “New designs in lightweight symmetric encryption,” in *RFID Security*, pp. 349–371, Springer US, 2009. 10.1007/978-0-387-76481-8_14.

[14] K. Rhee, J. Kwak, S. Kim, and D. Won, “Challenge-response based RFID authentication protocol for distributed database environment,” in *Security in Pervasive Computing*, vol. 3450 of *Lecture Notes in Computer Science*, pp. 70–84, Springer Berlin / Heidelberg, 2005.

[15] Gene Tsudik, “YA-TRAP: Yet another trivial RFID authentication protocol,” *Pervasive Computing and Communications Workshops, IEEE International Conference on*, vol. 0, pp. 640–643, 2006.

[16] T. Dimitriou, “A lightweight RFID protocol to protect against traceability and cloning attacks,” *Security and Privacy for Emerging Areas in Communications Networks, International Conference on*, vol. 0, 2005.

[17] Avoine, G. and Oechslin, P., “A scalable and provably secure hash-based RFID protocol,” in *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pp. 110–114, march 2005.

[18] Feldhofer, Martin and Dominikus, Sandra and Wolkerstorfer, Johannes, “Strong authentication for RFID systems using the AES algorithm,” in *Cryptographic Hardware and Embedded Systems - CHES 2004*, vol. 3156 of *Lecture Notes in Computer Science*, pp. 85–140, Springer Berlin / Heidelberg, 2004.

[19] Billet, Olivier and Etrog, Jonathan and Gilbert, Henri, “Lightweight privacy preserving authentication for RFID using a stream cipher,” in *Fast Software Encryption*, vol. 6147 of *Lecture Notes in Computer Science*, pp. 55–74, Springer Berlin / Heidelberg, 2010.

[20] H.-W. Kim, S.-Y. Lim, and H.-J. Lee, “Symmetric encryption in RFID authentication protocol for strong location privacy and forward-security,” in *Proceedings of the 2006 International Conference on Hybrid Information Technology - Volume 02*, ICHIT ’06, pp. 718–723, IEEE Computer Society, 2006.

[21] A. J. Menezes, P. C. V. Oorschot, and Scott A. Wilson, *Handbook of applied cryptography*. CRC Press, first ed., 1996.

[22] W. Diffie and M. Hellman, “New directions in cryptography,” *Information Theory, IEEE Transactions on*, vol. 22, pp. 644–654, nov 1976.

[23] H. Delfs and H. Knebl, *Introduction to cryptography: principles and applications*. Springer, second ed., 2007.

[24] L. Chen and G. Gong, *Communication systems security*. Draft, 2010.

[25] S. W. Golomb and G. Gong, *Signal design with good correlation: for wireless communications, cryptography and radar applications*. Cambridge University Press, 2005.

[26] C. Koc and B. Sunar, “Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields,” in *Information Theory, 1998. Proceedings. 1998 IEEE International Symposium on*, p. 294, aug 1998.

[27] H. Wu and M. Hasan, “Low complexity bit-parallel multipliers for a class of finite fields,” *Computers, IEEE Transactions on*, vol. 47, pp. 883–887, aug 1998.

[28] B. Sunar and C. Koc, “An efficient optimal normal basis type II multiplier,” *Computers, IEEE Transactions on*, vol. 50, pp. 83–87, jan 2001.

[29] A. Reyhani-Masoleh and M. Hasan, “A new construction of Massey-Omura parallel multiplier over GF(2m),” *Computers, IEEE Transactions on*, vol. 51, pp. 511–520, may 2002.

[30] C. Wang, T. Troung, H. Shao, L. Deutsch, J. Omura, and I. Reed, “VLSI architectures for computing multiplications and inverses in GF(2m),” *Computers, IEEE Transactions on*, vol. C-34, pp. 709–717, aug. 1985.

[31] C. Kim, Y. Kim, S. Ji, and I. Park, “A new parallel multiplier for Type II optimal normal basis,” in *Computational Intelligence and Security* (Y. Wang, Y. ming Cheung, and H. Liu, eds.), vol. 4456 of *Lecture Notes in Computer Science*, pp. 460–469, Springer Berlin / Heidelberg, 2007.

[32] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*. Cambridge Univ. Press, 2002.

[33] A. J. Menezes and I. F. Blake, *Applications of finite fields*. Kluwer Academic. Press, 1993.

[34] J.-P. Deschamps, J. L. Imaña, and G. D. Sutter, *Hardware implementation of finite-field arithmetic*. McGraw-Hill, 2009.

[35] R. C. Mullin, I. M. Onyszchuk, S. A. Vanstone, and R. M. Wilson, "Optimal normal bases in GF(pn)," *Discrete Applied Mathematics*, vol. 22, no. 2, pp. 149–161, 1988-1989.

[36] Guang Gong and Youssef, A.M., "Cryptographic properties of the Welch-Gong transformation sequence generators," *Information Theory, IEEE Transactions on*, vol. 48, pp. 2837–2846, nov 2002.

[37] C. H. Lam, "Verification of pipelined ciphers," *Electrical and Computer Engineering, University of Waterloo*, 2008.

[38] "Uniform Code Council." `http://www.gs1us.org/about_us/history/the_universal_product_code`, 2011.

[39] Weis, Stephen and Sarma, Sanjay and Rivest, Ronald and Engels, Daniel, "Security and privacy aspects of low-cost radio frequency identification systems," in *Security in Pervasive Computing*, vol. 2802 of *Lecture Notes in Computer Science*, pp. 50–59, Springer Berlin / Heidelberg, 2004. 10.1007/978-3-540-39881-3_18.

[40] S. Garfinkel and B. Rosenberg, *RFID: Applications, Security and Privacy*. Addison-Wesley Professional, first ed., 2005.

[41] K. Takaragi, M. Usami, R. Imura, R. Itsuki, and T. Satoh, "An ultra small individual recognition security chip," *Micro, IEEE*, vol. 21, pp. 43–49, nov/dec 2001.

[42] "RSA laboratories." `http://www.rsa.com/rsalabs/node.asp?id=2120#3`, 2011.

[43] Jeremy Landt, "History of RFID. IEEE potentials 0278-6648/05." `http://autoid.mit.edu/pickup/RFID_Papers/008.pdf`.

[44] S. Martínez, M. Valls, C. Roig, J. Miret, and F. Giné, "A secure elliptic curve-based RFID protocol," vol. 24, pp. 309–318, Springer Boston, 2009. 10.1007/s11390-009-9226-3.

[45] T. López, D. Ranasinghe, B. Patkai, and D. McFarlane, "Taxonomy, technology and applications of smart objects," vol. 13, pp. 281–300, Springer Netherlands, 2011. 10.1007/s10796-009-9218-4.

[46] Intermec, “ABCs of RFID: Understanding and using radio frequency identification .” www.intbarcode.com/pdf/ABCsofRFID_wp_web.pdf.

[47] Kamran Ahsan and Hanifa Shah and Paul Kingston, “RFID applications: An introductory and exploratory study,” *CoRR*, vol. abs/1002.1179, 2010.

[48] Klaus Finkenzeller, *RFID handbook: fundamentals and applications in contactless smart cards and identification*. John Wiley and Sons, second ed., 2003.

[49] Daniel W. Engels, Sanjay E. Sarma, “Standardization requirements within the RFID class structure framework,” Auto-ID White paper, MIT, September 2005, one ed.

[50] INTERNATIONAL STANDARD ISO/IEC 18000-4 , “Information technology — Radio frequency identification for item management — Part 4: Parameters for air interface communications at 2.45 GHz, .” http://webstore.iec.ch/preview/info_isoiec18000-4%7Bed2.0%7Den.pdf.

[51] AutoID Center, “Draft protocol specification for a 900 MHz Class 0 radio frequency identification tag.” http://www.gs1.org/docs/epcglobal/standards/specs/900_MHz_Class_0_RFIDTag_Specification.pdf.

[52] David L. Brock, “The virtual electronic product code,” Auto-ID center White paper, MIT, February 1, 2002.

[53] Sarma, Sanjay and Weis, Stephen and Engels, Daniel, “RFID systems and security and privacy implications,” in *Cryptographic Hardware and Embedded Systems - CHES 2002*, vol. 2523 of *Lecture Notes in Computer Science*, pp. 1–19, Springer Berlin / Heidelberg, 2003. 10.1007/3-540-36400-5_33.

[54] M. R. Rieback and B. Crispo, “The evolution of rfid security,” *IEEE Pervasive Computing*, vol. 5, 2006.

[55] S. Garfinkel, A. Juels, and R. Pappu, “RFID privacy: an overview of problems and proposed solutions,” *Security Privacy, IEEE*, vol. 3, pp. 34–43, may-june 2005.

[56] A. Poschmann, G. Le, K. Schramm, and C. Paar, “A family of light-weight block ciphers based on des suited for rfid,” in *Proceedings of FSE 2007, LNCS*, Springer-Verlag, 2006.

[57] “KATAN and KTANTAN - a family of small and efficient hardware-oriented block ciphers.,” in *CHES*, pp. 272–288, 2009.

[58] M. Hell, T. Johansson, and W. Meier, “Grain - a stream cipher for constrained environments,” *Int. J. Wire. Mob. Comput.*, vol. 2, pp. 86–93, May 2007.

[59] De Canniere, C and Prennel, B, “New stream cipher designs: The estream finalists.,” Lecture Notes in Computer Science, Vol. 4986, ISBN 978-3-540-68350-6, 2008.

[60] T. Good and M. Benaissa, “Hardware results for selected stream cipher candidates,” in *of Stream Ciphers 2007 (SASC 2007), Workshop Record*, pp. 191–204, 2007.

[61] Neil H.E. Weste and David Harris , “CMOS VLSI design,” in *A Circuits and Systems perspective*, Pearson Addison Wesley, 2005.

[62] Good, Tim and Benaissa, Mohammed, “ASIC hardware performance,” in *New Stream Cipher Designs*, vol. 4986 of *Lecture Notes in Computer Science*, pp. 267–293, Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-68351-3_19.

[63] Sandeep kumar and Kerstin lemke and Christof paar, “Some thoughts about implementation properties of stream ciphers,” The State of the Art of Stream Ciphers, Citeseer, 2004.

[64] M. Feldhofer and J. Wolkerstorfer and V. Rijmen, “AES implementation on a grain of sand,” *IEE Proceedings - Information Security*, vol. 152, no. 1, pp. 13–20, 2005.

[65] M. Kim, J. Ryou, Y. Choi, and S. Jun, “Low power AES hardware architecture for radio frequency identification,” in *Advances in Information and Computer Security*, vol. 4266 of *Lecture Notes in Computer Science*, pp. 353–363, Springer Berlin / Heidelberg, 2006. 10.1007/11908739_25.

[66] A. Juels and S. A. Weis, “Defining strong privacy for RFID,” tech. rep., 2006.

[67] R. Phan, J. Wu, K. Ouafi, and D. Stinson, “Privacy analysis of forward and backward untraceable rfid authentication schemes,” *Wireless Personal Communications*, vol. 61, pp. 69–81, 2011. 10.1007/s11277-010-0001-0.

[68] N. T. Courtois and W. Meier, “Algebraic attacks on stream ciphers with linear feedback,” pp. 345–359, Springer-Verlag, 2003.

[69] N. Courtois, “Fast algebraic attacks on stream ciphers with linear feedback.,” in *CRYPTO*, pp. 176–194, 2003.

[70] V. V. Chepyzhov, T. Johansson, and B. Smeets, “A simple algorithm for fast correlation attacks on stream ciphers,” 2001.

[71] I. Dinur and A. Shamir, “Cube attacks on tweakable black box polynomials,” in *Proceedings of the 28th Annual International Conference on Advances in Cryptology: the Theory and Applications of Cryptographic Techniques*, EUROCRYPT ’09, (Berlin, Heidelberg), pp. 278–299, Springer-Verlag, 2009.

[72] E. Biham and O. Dunkelman, "Differential cryptanalysis in stream ciphers," 2007.

[73] S. Ronjom and T. Helleseth, "A new attack on the filter generator," *Information Theory, IEEE Transactions on*, vol. 53, pp. 1752–1758, may 2007.

[74] S. Ronjom and T. Helleseth, "Attacking the filter generator over GF(2m)," in *Proceedings of the 1st international workshop on Arithmetic of Finite Fields*, WAIFI '07, (Berlin, Heidelberg), pp. 264–275, Springer-Verlag, 2007.

[75] A. Biryukov and A. Shamir, "Cryptanalytic time/memory/data tradeoffs for stream ciphers," in *Advances in Cryptology — ASIACRYPT 2000* (T. Okamoto, ed.), vol. 1976 of *Lecture Notes in Computer Science*, pp. 1–13, Springer Berlin / Heidelberg, 2000. 10.1007/3-540-44448-3_1.

[76] J. Lee and Y. Yeom, "Efficient RFID authentication protocols based on pseudorandom sequence generators," *Designs, Codes and Cryptography*, vol. 51, pp. 195–210, 2009. 10.1007/s10623-008-9255-x.

[77] P. Peris-lopez, J. C. Hern, J. M. Estevez-tapiador, and A. Ribagorda, "EMAP: An efficient mutual authentication protocol for low-cost RFID tags," in *In: OTM Federated Conferences and Workshop: IS Workshop*, pp. 352–361, Springer-Verlag, 2006.

[78] M. O. Koutarou, K. Suzuki, and S. Kinoshita, "Cryptographic approach to "privacy-friendly" tags," in *In RFID Privacy Workshop*, 2003.

[79] C. Berbain, O. Billet, J. Etrog, and H. Gilbert, "An efficient forward private RFID protocol," in *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, (New York, NY, USA), pp. 43–53, ACM, 2009.

[80] T. Good, W. Chelton, and M. Benaissa, "Review of stream cipher candidates from a low resource hardware perspective," SASC (2006), 2006.

[81] T. Good and M. Benaissa, "Hardware performance of phase-III stream cipher candidates," SASC (2008), 2008.

[82] A. Bogdanov and C. Rechberger, "A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN," in *Proceedings of the 17th international conference on Selected areas in cryptography*, SAC'10, (Berlin, Heidelberg), pp. 229–240, Springer-Verlag, 2011.

[83] C. De Cannière, O. Dunkelman, and M. Knežević, "KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers," in *Cryptographic Hardware and*

Embedded Systems - CHES 2009, vol. 5747 of *Lecture Notes in Computer Science*, pp. 272–288, Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-04138-9_20.

[84] L. Chae and K. Tymur, "mCrypton – a lightweight block cipher for security of low-cost RFID tags and sensors," in *Information Security Applications* (J.-S. Song, T. Kwon, and M. Yung, eds.), vol. 3786 of *Lecture Notes in Computer Science*, pp. 243–258, Springer Berlin / Heidelberg, 2006. 10.1007/11604938_19.

[85] M. Kasper, S. Kumar, K. Lemke-Rust, and C. Paar, "A compact implementation of Edon80," in *eSTREAM, ECRYPT Stream Cipher Project*, 2006.

[86] D. Arditti, C. Berbain, O. Billet, and H. Gilbert, "Compact FPGA implementations of quad," in *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, ASIACCS '07, (New York, NY, USA), pp. 347–349, ACM, 2007.

[87] L. Batina, J. Lano, N. Mentens, S. B. Örs, B. Preneel, and I. Verbauwhede, "Energy, performance, area versus security trade-offs for stream ciphers," in *In The State of the Art of Stream Ciphers, Workshop Record (2004), ECRYPT*, pp. 302–310, 2004.

[88] C. Rolfes, A. Poschmann, G. Leander, and C. Paar, "Security for 1000 gate equivalents," SECSI Workshop Berlin, March 2008, 2008.

[89] A. Bogdanov, L. R. Knudsen, G. Le, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "Present: An ultra-lightweight block cipher," in *the proceedings of CHES 2007*, Springer, 2007.

[90] A. Poschmann, G. Leander, K. Schramm, and C. Paar, "New light-weight crypto algorithms for RFID," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pp. 1843–1846, may 2007.