Date of Acceptance:

Grade:

Instructor:

Symmetric Encryption based Privacy using Lightweight Cryptography for RFID tags

Raviteja Sudulaganti

Master Thesis
UNIVERSITY OF HELSINKI
Department of Computer Science

Helsinki, March 25, 2015

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Laitos — Institution — Department Department of Computer Science

Tekijä — Författare — Author
Raviteja Sudulaganti

Työn nimi — Arbetets titel — Title
Symmetric Encryption based Privacy using Lightweight Cryptography for RFID tags

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level Master Thesis	Aika — Datum — Month and year March 25, 2015	Sivumäärä — Sidoantal — Number of pages 46

Tiivistelmä — Referat — Abstract

RFID technology emerged as the promising technology for its ease of use and implementation in the ubiquitous computing world. RFID is deployed widely in various applications that use automatic identification and processing for information retrieval. The primary components of an RFID system are the RFID tag (active and passive), the reader and the back-end server (database). Cost is the main factor that drove RFID tags to its immense utilization in which passive tags dominate in today's widely deployed RFID practice. Passive tags are low cost RFID tags conjoined to several consumer products (like clothes, smart cards and devices, courier, container, etc) for the purpose of unique identification. Readers on the other hand act as a source to track and record the passive RFID tag's activities (like modifications, updates and authentication). Due to the rapid growth of RFID practice in the past few years, measures for consumer privacy and security has been researched. The uncertainties that arise with the passive RFID tags are handling of user's private information (like name, ID, house address, credit card number, health statement, etc) which are posed to considerable threat from the adversary. Passive tags are inexpensive and contain less overhead and are considered good performers and consequently lack in providing security and privacy.
Lightweight cryptography is an area of cryptography developed for low cost resourced environment. Mutual authentication is defined as the process of verifying an authorized tag and a reader (reader and server respectively) by an agreed algorithm to mutually prove their legitimacy with each other. Adversary is a third party who tries to hear the ongoing communication between the tag and the reader (reader and server respectively) anonymously. In this thesis, symmetric lightweight ciphers like Present and Grain are introduced as mutual authentication protocols to rescue the privacy aspects and properties of the RFID tags. These ciphers are simple, faster and suitable to implement within the passive RFID network and reasonably lay a foundation for the preservation of privacy and security of the RFID system. Lightweight ciphers use hash functions, pseudo random generators, SP networks and linear feedback shift registers to randomize data while mutual authentication scheme uses lightweight ciphers to manage authorize the legitimacy of every device in the RFID network.

ACM Computing Classification System (CSS):
Security and Privacy ⇀ Cryptography ⇀ Security services ⇀ Human and social aspects of security and privacy

Avainsanat — Nyckelord — Keywords
identification, RFID, tag, lightweight, cryptography, authentication, adversary, cipher, privacy

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Introduction

1.1 Motivation

Smart cards and devices play an enormous role in our day to day life. Even though these devices are not readily attached to us, we often come into contact with them (tags, readers and servers) in our daily life unknowingly (e.g. ATM money withdrawal, credit card payments, local transportation service, toll systems, speed ticket systems, etc.). Smart cards act as secure carriers, encrypting and computational devices. RFID tags are available as smart cards, java cards and dotnet cards in serving various consumers near toll systems, payment systems, identification systems and tracking systems for a safer and convenient communication. Java cards and dotnet cards are used mostly as vicinity cards by several firms. These tag (cards or labels) carry information about user's profile data, bank transactions and their personal information and have the ability to store the past traces of the user and can be retrieved later when commanded by a reader or the server. Some of the applications using smart cards include users gym and cosmetic payments, offline transactions, online transactions which draws attention concerning their security and privacy issues. These smart cards hold threats to famous people. Queries and feedback of various users perspectives are drawn and evaluated which concluded the uncertainties that exist in using the RFID technology. Cryptographic techniques have been implemented in the RFID systems for safeguarding the information while processing and transferring them. Immense research had been undergone in the field of lightweight cryptography which is developed to support the security of the smart devices like RFID tags, sensor devices, etc. RFID technology has been in play since 1950's while it rapidly grew and gained attention in the mid 1990's. The commercial use of RFID worldwide started with theft detection and anti-counterfeiting technology called European Article Surveillance (EAS) in the shopping malls. Later the technology turned to provide different services like access control in parking lots, inventory supply chain management and toll systems by issuing tokens (RFID tags) and querying the tokens by the authorized to retrieve the required information, with the help of the readers. Cryptography has been a major tool serving security issues since decades but cryptography in the form of lightweight cryptography is a rescue tool for constrained devices like RFID tags.

1.2 Problem Statement

Due to the reduction of the costs and ease of implementation of the RFID tags caused it's rapid increase in the distribution from the supply chain management to several other commercial applications. The sudden demand or usage had created certain insecurities and privacy issues that are to be absolved. Adversaries have the opportunity to collect and track one's information from his RFID tag. Adversaries are third persons who try to capture private information without prior permission. This enables the eavesdropper to promote the personal identity information of an user by executing an illegal activity. The most important entities that maintain user privacy are user data and location. Forward and Backward privacy are two forms of privacy that hold user's privacy by measuring the eavesdroppers capabilities to a certain extent. It is said that if a system supposedly achieves forward and backward privacy in the trial of an user's privacy is notably considered as almost achieved privacy in that particular system. Forward and Backward privacy can be easily broken if any of the illegitimate tags or readers participate in the unauthorized communication process. Cryptography plays an important role in transforming information allowing a limited possibility for an eavesdropper to gain any portion of the exchangeable information during his interaction between devices through the various possible channels. Lightweight cryptography is one of the area of cryptography which utilize the cryptographic algorithms that are primarily dedicated to the low cost devices. The primary measures that are considered in developing new cryptographic ciphers are gate equivalence (GE), chip size, number of registers, time measurement for delay-response, power consumption, computation power, etc. Since RFID tags need lightweight resources and faster techniques in implementation, maintaining privacy and security in the system is a primary challenge. In this thesis, we present a method for authentication and preservation of user privacy by using various lightweight symmetric ciphers.

1.3 Thesis Organization

The thesis report is divided into four chapters. The first chapter introduces the basic theme of the thesis, the problem statement and the thesis limitations. The second and the third chapters introduce RFID technology and lightweight cryptography in a structured and detailed manner. The last chapter present the privacy and security issues of the contemporary RFID systems and how the lightweight cryptographic primitives play a vital role in maintaining and preserving the privacy of the end users. This thesis studies certain lightweight symmetric ciphers which when used along with the authentication protocols find better solutions for preserving privacy. This report presents an overall idea about how the RFID system is maintained, their threats (privacy and security issues) and the use of cryptography in maintaining these threat prone low-resource devices (RFID devices) in reaching a goal of privacy successiveness and user confidentiality.

1.4 Thesis Delimitation

The thesis is arranged in such a way to demonstrate the overall RFID system in a cryptographic environment. The objective of the thesis is to present a privacy preserving analysis by showing the least possible probability in gaining information by an adversary. This thesis does not evaluate the cryptanalysis of the lightweight stream ciphers as the ciphers are presumed to be secure. Since the thesis is much concerned about the privacy preserving aspects, the security aspects of the cryptographic ciphers are not explained. There is a hypothesis to present the ability of the ciphers as ciphers are believed to work and resist complications created by the adversary for those issues considered in this thesis.

2 Overview of RFID

2.1 History of RFID

The roots of the RFID show it's use way back in the Second World War in 1940 to differentiate the good from the bad warplanes. The first official patent of RFID was released in 1973 in the United States received for a working active RFID tag with a rewritable memory. The same year a patent from a California entrepreneur was received for a passive RFID tag. The first RFID test application was an automatic opening of a door when the reader was brought near door where the tag was located. Later, RFID was used by the US Agricultural department to maintain the dosage system of the cows in the cattle farms. In 1960's, a commercial RFID application called Electronic article surveillance (EAS) was introduced to control thefts in shopping centers. RFID research was started and funded by the US defense government for proper surveillance of nuclear materials spread all over. In 1970's, the free flow toll systems were tested with the passive RFID transponders which are implanted in the vehicles and read by the readers located under the surface of the highway. Sooner, RFID technology grabbed attention by various industrial sectors for its ease of use and performance. In 1980's, RFID was introduced in many mainstream applications. Some of its applications include electronic toll systems worldwide, personal access systems, animal tagging, and many other industrial applications. In 1990's, there was an emergence to develop open standards for RFID systems as a security measure due of its wide deployment. Auto ID center was an organization founded in 1999 by MIT university with a group of four other universities to develop standards for RFID. The MIT Auto ID center's main aim was also to develop the electronic product code (EPC) as a global identification system using RFID tags and to replace the conventional bar code systems which use universal product code (UPC). Later in 2003, MIT AutoID center was named Auto ID Labs which was run by the EPC Global and the group of partners of the Auto ID labs. RFID was widely deployed for solutions related to various applications used in schools to the US defense department. There was an assumption that RFID will be deployed in all service sectors in the near future.

2.2 RFID today

Today RFID is everywhere. As RFID has been rapidly proliferating globally, there is a need in obtaining legal and reliable solutions to the global privacy and security issues [11]. At present, Americans are the highly sensitive people towards these issues as they abundantly use RFID technology and the Europeans on the other hand are towards accepting the RFID technology quickly and the Asians are about to introduce the technology in many places [11]. Hence, it adds a duty to the researchers and developers to act upon the growing privacy issues and to develop better algorithms and techniques that avoid major threats to the privacy of the daily consumers as well as the retailers and the producers. In summary, today RFID technology is used in Schools, Libraries, Hospitals, Agriculture, Infrastructure, Payment systems, Identification systems, Detection systems and Alarms, Tracking devices and Defense.

2.3 Fundamentals of RFID

RFID stands for radio frequency identification. The main components of the RFID system are the tag, the reader and the back-end server. RFID technology originated from the time of invention of the transmitter and the receiver. The RFID system uses radio waves in transmitting and receiving signals for the data communication. The basic advantage of RFID tags to that of the conventional bar codes is that the former have read and write capabilities and the information on the RFID tags can be modified, updated and locked. RFID devices utilize less power for short range communications in handling the data activity. RFID devices operate in three different frequency bands namely low frequency (LF), high frequency (HF) and the ultra high frequency (UHF). The low frequency bands range up to one meter which use inductive coupling technique for the data transfer while the latter range up to ten meters which use the back scattering technique. Some major applications of RFID devices include Supply chain and retail management, defense systems, health care systems, access control systems, transportation systems and so on.

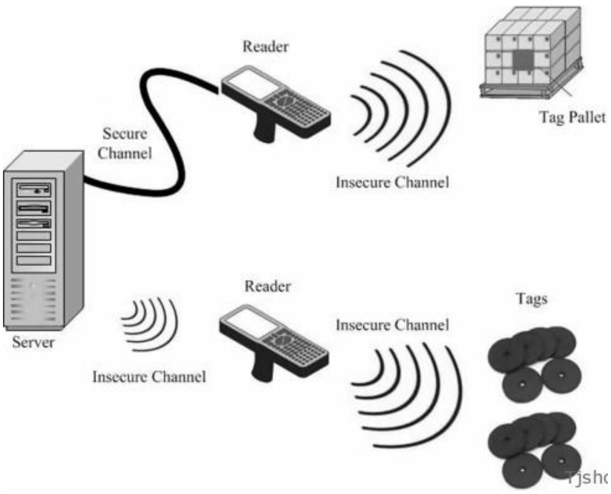

Figure 1: An RFID system

2.3.1 Tags

Tag is the basic component of the RFID system. Tags are also called as the transponders. Each tag contain an unique code and a limited memory to store information. There are three types of tags defined according to the power and range capabilities namely active, semipassive and passive tags. Usually, passive tags have no power source and they retrieve the signal energy from the reader while semi-passive and active tags have a battery source to generate a signal by their own and can exchange information with the reader. The main characteristics of a tag include singulation, anti tag collision interference and its unique identity. A tag header contains three elements namely a CRC, EPC and a password. Cyclic redundancy check (CRC) is the checksum of the tag to check if any errors occurred during data transmission or storage. Electronic product code (EPC) is the unique serial number used to identify the tag. Password is the block of data used for authenticating a legitimate reader. RFID tags currently operate in three different frequency bands namely low frequency module (LF), high frequency module (HF) and ultra high frequency module (UHF). Generally, low frequency bands operate in lower read range and data rates and high frequency bands operate in higher read ranges and data rates. Each frequency band work on their own mechanism and no one frequency band is ideal for all applications[2].

	Active tag	*Semi-active tag*	*Passive tag*
Power Source	Yes	Yes	No
Storage	Yes	Yes	No
Performance	High	High	Low
Cost	High	High	Low

Table 1: Types of tags

	Low cost tags	*High cost tags*
Power Source	Passive	Active
Storage	32 bytes - 1KB	32 - 70KB
Security	250 - 2000GE	3000 - 5000GE
Distance	3 meters	10 cm
Price	5 - 50cents	several euros

Table 2: Tag specifications

2.3.2 Readers

Reader is the powerful component of the RFID system. Readers are also called as the transceivers. Each reader has the capacity to encompass certain number of tags within the read range. Reader can change, update and modify the information in the tags by proving its legitimacy through an authentication process. There are two types of readers namely simple and complex readers. Simple readers are mostly used for verification in supply chain management, tracking goods which require less computation. Complex readers are meant for applications which needs computation and evaluation of the legitimate RFID tags[2].

	Low Frequency	*High Frequency*	*Ultra High Frequency*
Bandwidth	125-135 KHz	6 - 27 MHz	400 - 930 MHz
Range	1.0 meter	10 meters	100 meters and more

Table 3: Tag frequencies

2.3.3 Back-end server

Back-end server or back-end database is the third key component of the RFID system. Each server maintain a database which contain the information of all the legitimate readers and tags. Back end servers are connected to the readers directly or wireless to retrieve information and verify the tag. The tag related information is processed by the server for authentication and the approval is sent back to the reader for further transactions. The reader can be queried by the server for the detailed information of the tags and the reader respectively.

2.3.4 Data Communication

The communication in the RFID system can occur between the tag, the reader and the server while it is initiated either by the reader or the server(in case of passive tags). The tag and the reader communicate through insecure channels (in most cases) where there is a possibility of eavesdropping while in case of reader and server (when connected through wire) reduces the means of eavesdropping. There are two channels that carry the information to both the ends namely the forward channel and the backward channel in which the former is responsible as the information barrier from the reader to the tag while the latter is responsible for the exchange of information from the tag to the reader. Digital and analog signaling techniques like Manchester encoding, non-return-zero (NRZ), pulse position modulation (PPM), amplitude shift keying (ASK), pulse shift keying (PSK) and frequency shift keying (FSK) are used for encoding and modulating the communication signals while transmission. RFID systems mostly use these techniques to detect any noise (error) on the channel and prompt the reader to wait or delay the argument process [1].

2.3.5 EPC and EPC Gen2 Standard

EPC is abbreviated as Electronic product code. It is considered as a standard code for RFID tags. All the EPC codes are administered by the EPCglobal standard system. EPC is a 96 bit string which is capable of identifying 16 million object classes producing 68 million serial numbers for each class serving for 268 companies all over the world. The EPC was soon being extended to 128 bit for future applications. An EPC consists of four parts namely EPC header, EPC manager number, EPC product number and a unique serial number. The EPC header specifies the version number, format, and the length (64, 96,128-bit) of the tag , the EPC manager refers to the manufacturer identification number, the EPC product number refers to the type of product and the EPC serial number refers to the unique identification of the desired item. EPC was firstly created by the MIT Auto-ID center organization and then licensed to EPCGlobal in October 2008.

EPC Gen2 standard was introduced to develop the second generation EPC tags which are being used in today's commercial applications where these codes play a vital role in the unique identification of the smart devices. These codes are assigned to low-cost tags that are used for smart applications with limited resources and fast processing capabilities. The purpose behind the enhancement was to utilize the first generation Class 1 tags for the second generation smart device applications. The detailed information about the tag classes is showed in the Table 2.4.

2.3.6 EPCglobal and ISO standards

EPCglobal is the universal standard for maintaining and improvising EPC tags and products. It is formed in October 2008 replacing the MIT Auto-ID center. EPCglobal is responsible for all the EPC enabled data sharing worldwide. EPCglobal defines several classes of tags which embrace all the types of tags discovered and applied so far [12]. ISO standards on the other hand introduce certain policies and rules in maintaining and applying RFID devices in different environments. Some of the general ISO standards are listed in the Table 2.5.

2.4 Need for Cryptography

Many cryptographic techniques have been designed to overcome the problems in using RFID technology. The main problems that are evident by using

	Tag classification	Description
Class 0	Read only and passive	Programmed by the manufacturer
Class 1	"Write once, read many" and passive	Programmed by customer not reprogrammable
Class 2	rewritable and passive Gen2	Passive tag with 65KB rewritable memory, reprogrammable
Class 3	Semipassive, Gen2	Built-in battery with increased range, reprogrammable
Class 4	Active, Gen2	Runs its own circuitry with use of its battery, reprogrammable
Class 5	Active, Gen2	Communicates with all other Class 5 tags, reprogrammable

Table 4: Classes of tags [12]

ISO standard number	*Description*
ISO 11784/11785	RFID for animals
ISO 14443	Proximity and Vicinity cards, e.g credit cards.
ISO 15961/15963	Information technology and unique identification
ISO 18000	Interface communications, e.g Air, Frequency, etc
ISO 18185	Electronic seals in Foreign containers
ISO 15459	Unique identification in Transportation

Table 5: ISO standards for RFID tags

RFID are providing security and privacy to the end users. The main aim of using cryptography is to randomize the tag results over time when sending it to the reader so as to prevent any privacy threats. Several solutions have been designed to fit the lightweight crypto-algorithms in terms of execution and low computation capabilities. Low cost tags (passive RFID tags) certainly cannot perform all the standard cryptographic operations necessary to offer security due to the high overhead [16]. Hence, achieving a considerable security and privacy in passive tags is an upcoming challenge. Many researchers have come up with solutions that offer cryptography to the low cost tags in which the foremost two techniques were proposed by the MIT Auto-ID center and the RSA laboratories for the Eavesdropping problem. Researcher's from the MIT center proposed a technique called silent tree walking in which the primary concern was that no reader can broadcast tag's information. The second technique was proposed by the RSA laboratories by using pseudonyms where the tags carry multiple identifiers each time and are completely random and can only be recognized by the legitimate readers and not eavesdroppers [16]. At present, various lightweight implementations are proposed to maintain privacy and security in the RFID systems. Lightweight cryptography is defined as "As light as a feather, and as hard as dragon-scales"[17]. It is also defined as 'lightweight" and "cryptography" together leads to think about lack of security, but there exist combinations that allow limited constrained devices to persist by achieving performance and scalability while maintaining privacy and security[18].

2.5 RFID applications

RFID technology has been in use since World War II. Today RFID technology is used almost everywhere. RFID technology is used by many firms in many applications for improving their internal and external efficiency. In general, most common applications that utilize RFID technology presently are:

Transportation

RFID technology is used in various local transport systems in different countries, like Switzerland, UK, Finland, etc; by issuing unique RFID tags to all its customers to provide access to distinct bus services. RFID tags are also used to distinguish various foreign containers and automatically identify a unique product among various products inside one container and to pop-up a distinct categorized product among the whole.

Agriculture

Agriculture is the one of the first areas that used RFID technology. RFID tags are used in recording movements of animals like cattle, pigs, etc. By using the record information, modern farmers can monitor the health status of the animals from time to time. The same concept can be used for ensuring correct feed to specific category of animals by maintaining a database of information. All the information related to each RFID tag (unique identification of every animal) is stored in the central database and is updated from time to time.

Retail and Supply Management

RFID technology is a major advancement in the Retail and Supply management used in keep track of all the products from manufacturing to the assembly phases and maintaining them. It is easy to monitor the movement of the finished products throughout the inventory. RFID is also used to evaluate and count the frequent hot selling products and the non-frequent selling products in the inventory. This scheme possibly reduce the costs of the working units and surveillance. RFID is also used in theft detection in protecting the selling products from theft and counterfeiting. RFID is also used in preserving brand protection in identifying the fake products (fake tags).

Smart devices

RFID tags act as sim cards in smart phones which implicates that the smart phone inturn acts as a RFID device. Smart phones act as readers in active mode and tags in passive mode. RFID tags are used in credit cards and other smart cards to uniquely verify a user to control accessibility under certain limitations. RFID tags are also used in ePassports and National ID cards in many countries. All these cards are implanted with RFID tags inside and are used for identification and access controlling purposes in different applications.

Navigation Systems and Defense

RFID devices are incorporated with GPS to provide more accurate information to the mobile users by installing RFID tags (to store location and other necessary information) in various parts of the roads. All the RFID tags (tagged smart phones) refer to the readers (GPS systems) which provide the users with accurate information to reach the destination. By using RFID reader modules incorporated with GPS and gyroscope (device which rotates for different dimensional views) together provide highly accurate information to the end users. RFID is also used in PDA's assisted with bluetooth and Internet which provide audio information about the route to the destination to the visually impaired persons. RFID is also used in defense to trace the nuclear weapons (weapons installed with RFID tags) and to improvise the management between the manufacturers and the suppliers.

Health Care Systems

RFID is used in the Health care management to eliminate drug counterfeiting and theft control. It is also used for misuse of medications and mis-identification of the drugs. RFID is used in patient tracking and medical assistance by tagging RFID tags to the patient. This process helps in avoiding improper drug usage to the patient by advance installing the drug information to the tag. The doctors can study the patient medical status and profile by just reading the tag information. RFID is also used to identify and count the drugs in the shelves by placing readers (installed with the drug database) in the shelves.

Access Control

RFID is used in airport baggage management for maintaining the right information about the luggage of a specified airplane. RFID is used in various toll road systems for automatic identification of vehicles by verifying the RFID tags installed in the cars. RFID is used in door access systems by installing RFID tags (in the form of plastic cards) permitted to different access points. RFID is also used in parking lots to control traffic and display the vacant parking slots to the users. RFID provides real time security in various applications.

3 Lightweight Cryptography

Since traditional cryptography is not suitable for low constrained devices like RFID devices and sensor (smart) devices, there was a need to develop new cryptographic models that suits the environment of low constrained devices. This type of cryptography is defined as lightweight cryptography. Lightweight Cryptography is a part of modern cryptography developed for low-constrained devices and is considered as a prominent state-of-art technique. Lightweight cryptography aims to utilize all the available limited resources of the system (in terms of hardware and software) and provide better efficiency and overall performance. Most of the factors that are considered while implementing lightweight cryptography are gate equivalence, computational power of microprocessors, power consumption, random access memory (RAM) and read only memory (ROM) in terms of hardware and key-size in terms of software. Other factors include the design of the cryptographic algorithms, computational methods, lightweight cryptographic tools and distinct RFID protocols. Cryptography is all about masking a message using a key to produce a cipher and unmasking the cipher when required using the same key to get the original message. This cipher uses keys, random generators, logic gates, etc. to perform encoding and decoding of the confidential information. Ciphers are of two types, namely symmetric and public key ciphers. Symmetric ciphers are those which is use same key for encoding and decoding the message while public key ciphers use different keys at both the ends. There are several symmetric lightweight ciphers like Present, Grain, etc; that are prominently used in today's smart devices (like RFID tags, sensor devices) to mask the signals using cryptography to safeguard information from the third parties (adversaries).

3.1 Preliminaries

3.1.1 Linear feedback shift register (LFSR)

Linear feedback shift register is a n-bit shift register that generates a periodic sequence ranging from 0 to 2^n-1. They are known to produce binary sequences

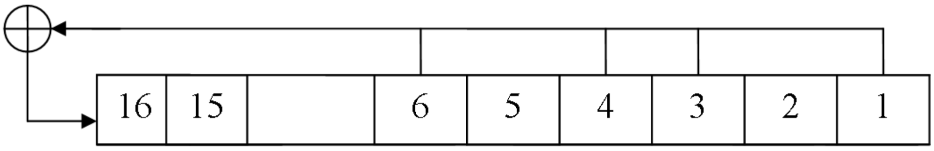

Figure 2: Linear feedback shift register

with good pseudo random properties, but probable to be predicted and hence are mostly combined with a non-linear function. LFSR's are used often to construct stream ciphers and pseudo random number generators (PRNG). For example: Consider 16-bit LFSR shown in Fig 4.1 whose polynomial equation can be derived as $X^5 + X^3 + X^2 + 1$. As the LFSR has 16 bits (0 $\leq$ i $\leq$ 15) in which 0th, 3rd, 5th and 7th bits are selected for feedback and the output register (Z_{15}) of the 16-bit LFSR would be as follows:

$$Z_i = (Z_{i-8} + Z_{i-10} + Z_{i-12} + Z_{i-15}) \bmod 2 \ldots \tag{1}$$

LFSR's are easy to implement as it requires minimal hardware logic. There are two types of linear feedback shift registers namely internal LFSR and external LFSR. In external LFSR's, the feedback to various XOR gates in the circuit is fed from the output of the previous LFSR. In internal LFSR's, the feedback to various XOR gates are within the LFSR. LFSR's are used in the construction of counters, PRNG's, etc.

3.1.2 Non-linear feedback shift register (NFSR)

NFSR is one of the prominent component in the construction of the modern stream ciphers for RFID and smart devices. They can be viewed as the finite state automata. Finite automata is a theory of state machines which accept specific nature of strings or bits and produce strings of another state. They are easy to implement and faster than the LFSR's. The output sequence of the NFSRs have good statistical properties and hard to predict. NFSRs have been proposed as an alternative to the LFSRs for generating key streams for stream ciphers. NFSRs are shift registers whose current state is a non-linear function of the previous state. The output sequences of the NFSRs are hard

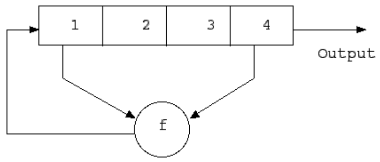

Figure 3: Non-linear feedback shift register

to predict and are more resistant to the algebraic attacks. NFSRs can be implemented in Fibonacci and Galois configurations. The last bit of the state is considered as the feedback bit and the feedback can be applied to every bit. Galois configured NFSRs are faster and hence became popular for its use in many stream ciphers like Grain and Trivium (eStream ciphers).

3.1.3 SP and Feistal networks

A substitution-permutation network consists of a series of S-boxes and P-boxes interconnected virtually to diffuse randomness of the input data. Advanced encryption standard (AES) is designed using the SP network. S-box (substitution box) is defined as a logical look-up table with a series of mathematical operations processed linearly in order to retrieve the original information. The inputs to the S-box are a series of bits of length say 'm' which are transformed into a series of bits of a length say 'n' which can be retrieved back using a reverse operation. These inputs are again fed into another S-box for a more strategic preservation of the primary input information. P-box is defined as a permutation box which shuffles the given set of bits across the inputs of S-boxes. In a SP network, S-boxes and P-boxes are chained together as to create confusion and diffusion across the various inputs and outputs in building a ciphertext. On the other hand, Feistel networks are slight similar to the SP networks not only in the implementation but in the design. Feistel networks use internal functions called round functions which are iterated in schedule consisting of initial vectors (IV) and the key schedule. Round functions play a major role in iterating the sequence of instructions embedded as a procedure. Feistel networks also use a procedure called sub-mixing which combines several functions so as to create randomness in the whole feistel network. SP and feistel networks are constructed in terms of both hardware and software to get the optimized implementation of the algorithm.

3.1.4 Pseudo Random Generators

A Pseudo random generator is an algorithm that accept a string of say 'm' bits as input and produce an output string of say 'n' random bits larger than size of the input string, n>>m. Pseudo random generators are used in the cryptographic encryption algorithms to provide randomization in the key stream generation. Key generation is the fundamental process that injects randomness to the entire process eventually. Hence PRNG's are implemented primarily for the key stream generation and authentication and encryption schemes in the conventional and lightweight cryptography. Conventionally, PRNG's have two characteristics that they bound to, in which firstly is they do not allow detection of any of the original (initial) inputs so as to prove its randomness statistically. Secondly, PRNG's does not allow any eavesdropper to predict the next bit based on a given set of bits or probability. The latter statement is assumed to be passed of a PRNG if and only if it passes the former statement [24].

3.1.5 Cryptographic Hash Functions

Hash function is an algorithm that takes a binary string of some message of any length as input and transforms into a fixed length output called a hash value. A hash function is a function f:D$\rightarrow$R , where the domain D = [0,1]* consists of binary string of variable length and output R = $[0,1]^n$ consists of binary string of fixed length. Hence f is a function which takes a message M and transforms it into a hash value h of size n. Usually every hash function compresses a finite string into a much shorter fixed length output [15,26].

Properties of a hash function:
[i] A hash function f should accept a message of any size.
[ii] A hash function should produce a fixed length output.
[iii] Given any given message M, it should be easy to determine a hash value h.

[iv] Given a hash value h, it is computationally difficult to find M such that H(M)= h.

[v] Given a message M_1, it is computationally unfeasible to find another message M_2, with H(M_1) $\neq$ H(M_2)

[vi] It is computationally unfeasible to find any pair of distinct messages (M_1, M_2) such that H(M_1) = H(M_2).[41]

3.2 Symmetric Lightweight protocols

Symmetric encryption is the oldest and fastest methods of cryptography used in today's applications. Symmetric encryption uses a single secret key to encrypt and decrypt the information. This methodology of using the same secret key to retrieve the information is mostly used in private surroundings. Symmetric protocols are divided into stream and block ciphers which are categorized based on their encryption mechanisms. Stream ciphers are symmetric ciphers which encrypt the message bit by bit and the encryption of each bit is related to it's previous bit. They are fast in processing, but consume much memory. Stream ciphers are further divided into synchronous and self-synchronous ciphers. Synchronous ciphers are a type of symmetric ciphers which tend to remain erroneous in case of transmission errors as the key totally depends upon the previous sequences of the keystream. Self-synchronous stream ciphers are self adjustable ciphers as the key is a sequence of the previous plaintext and ciphertext. Block ciphers encrypt the message as blocks of data in which each block is separately encrypted with a single key. Stream ciphers and block ciphers are significant in their approaches and applied in their places of interest. There are several lightweight stream and block ciphers that are developed for applications presently used today. A hybrid cipher called Humming bird-2 (mixture of block and stream cipher) which is also developed for authentication of the low constrained devices. Similarly, Grain being a stream cipher (belongs to e-stream family) is also dedicated to deal with security and privacy preserving aspects while implemented in low cost environments. In this section, a brief implementation and working of two ciphers (Present and Grain) are presented. Various ciphers have been developed for smart devices and their tools to improve security and privacy complexities. Some of the important cryptographic protocols that are useful in authentication of RFID devices are explained in this section and are compared to analyze the privacy aspects in the next chapter.

- Rekeying: Since symmetric ciphers use a shared key directly to cryptographically process data, using it as a master key and deriving subkeys and using those subkeys for processing is called re-keying. This process is mostly used in block ciphers and is expected to bring good security results.

3.2.1 A lightweight block cipher: PRESENT

PRESENT is a symmetric block cipher which can be implemented in both software and hardware. It has moderate security levels and uses a 80 bit keystream. PRESENT uses an extremely simple SP network, an 80-bit key and 64-bit block size. It contains 16 S-boxes of dimensions 4x4 and uses two rounds of simple bit permutations. The final cipher is then extracted using 31 rounds of key scheduling. This cipher uses the smallest s-boxes (4x4) in hardware implementations. The key is designed to eliminate the symmetry in the process and to prevent side channel attacks. The main intention in using S-box and permutation layers is to introduce non-linearity and diffusion in the overall input process. Rekeying is not possible using present cipher. The whole cipher is constructed using four functions namely *addroundkey*, *S-box layer, P-layer* and *key scheduling.* The working of the PRESENT cipher can be explained using a single stage called *addroundkey* with 32 iterations and three sub-stages where the key and state are updated and modified using *S-box layer, P-layer* and *key scheduling*[5].

Terms:
K: 80 bits , user supplied key
K_i: 64 bits, roundkey where 0≤i≤63 (leftmost 64 bits of K)
b_j: 64 bits, block size where 0≤j≤63
w_i: 16 (*4 bit) words where 0≤i≤15
S[w_i]: current state in the S-box
P[i]: Position of every i^{th} bit of S-box in the P-box.

Addroundkey

Addroundkey is a key which is modified every round with the scheduled key to produce the version of the current state. Given the user supplied key of 80 bits and the roundkey (initially the LSB 64 bits of the supplied key) is 64 bits, and a block of 64 bits, the acquired current state (64 bits) for each round (0 to 31) can be expressed as in the equation 3.2.

$$b_j = b_j \oplus {K^i}_j \text{ where } 0 \leq \text{j} \leq 63 \tag{2}$$

S-box layer

The s-boxes substitute the 64-bit current state as sixteen 4-bit words and arrange them in rows and columns. After every round of arranging of s-boxes, p-boxes take the i^{th} bit in the 4x4 s-box array and arrange it in the position P[i] in the box. In each S-box layer, the current state of 64 bits, b_{63} . . . . b_0 are conceived as sixteen 4-bit words.The 16 words are represented as w_{15} . . . .w_0 where each w_i can be expressed as in the equation 3.3. The S-box updates for each of the values of w_i as S[w_i] accordingly.

$$w_i = b_{4*i+3} \mid\mid b_{4*i+2} \mid\mid b_{4*i+1} \mid\mid b_{4*i} \text{ where } 0 \leq \text{i} \leq 15 \tag{3}$$

P-box layer

In the permutation-layer, every i^{th} bit of the state in the s-box is moved to position P[i] in the p-box. We can say that the bits in the s-box array are shuffled within the p-box of the same dimension of an s-box.. The main aim of using the p-boxes is to diffuse the elements of the s-box in a certain manner so that there exists confusion in understanding the pattern of the arrangement. The p-boxes are always a key component in the construction of an SP network.

Key Scheduling

The main aim of this function is to update the key in each round of the total 32 rounds. The user supplied key being 80 bits (K : k_{79} . . . k_0) in which the leftmost 64 bits (K_i : k_{79} . . . k_{16}) are conceived as the round key bits, in each round of the Addroundkey stage. The 80 bits of the user key K is updated in an unusual fashion for every round to maintain randomization. The updation is as shown below:

At round i (0<i <31), the 64-bit round key is updated as in the equation 3.4.

$$k_{63}. \; . \; . \; k_0 = k_{79} \; . \; . \; . \; k_{16} \tag{4}$$

After extracting the round key in every i_{th} round, the 80-bit user key K is updated as in the equation 3.5. The 80-bit original key is updated every round using the equation 3.5. The updation takes place in three steps. The updated key is always 80-bits. The round key is chosen by taking the leftmost 64 bits from the updated original key after every round. The current state after every round is calculated using the equation 3.4. The current state is updated every round using XOR operation. The current state at the 32^{nd} round is the ciphertext[5].

$$\begin{aligned} k_{79}k_{78} \; . \; . \; .k_0 &= k_{18}k_{17} \; . \; . \; . \; . \; k_{20}k_{19} \\ k_{79}k_{78}k_{77}k_{76} &= \mathrm{S}[k_{79}k_{78}k_{77}k_{76}] \\ k_{19}k_{18}k_{17}k_{16}k_{15} &= k_{19}k_{18}k_{17}k_{16}k_{15} \oplus i_{th}.\mathrm{round} \end{aligned} \tag{5}$$

The cipher is extracted after 32 rounds of the above process. Since it uses an SP network for construction, the plaintext given by the user can be extracted by using a reverse process using the addroundkey function. Present is a new lightweight block cipher developed by the Orange labs in France, Ruhr University Bochum (Germany) and the Technical University of Denmark. The cipher was designed by eight people. It is one of the most compact encryption methods ever designed and is 2.5 times smaller than Advanced

Encryption Standard (AES). It is mostly focused on hardware implementations and suitable for situations where low power consumption and high chip efficiency is desired. PRESENT is included as a new international standard by International Organization for Standardization and the International Electro-technical Commission [29,30].

3.2.2 A lightweight stream cipher: Grain

Grain is a lightweight stream cipher developed for constrained environments. The cipher is simple in design for software implementation but complex to construct in hardware. Considering AES (advanced encryption standard) which being a successful stream cipher for many applications despite consuming large number of gates is not feasible to fit in the RFID environment. RFID environment needs smaller and faster working ciphers. The primary properties of lightweight cryptographic ciphers are gate area, speed, security and simplicity. Every cipher has its own significance in terms of simplicity and design where partial differences in hardware and software implementations exist. The design of GRAIN [4] is based on two shift registers namely linear feedback shift register (LFSR) and the non-linear feedback shift register (NFSR). The advantage of using the LFSR is that it guarantees a minimal period in using the key stream and producing a balanced output. On the other hand, NFSR introduces a non-linearity to the cipher as the LFSR when used solely is compromised to attacks. Many stream ciphers in history are built on using linear feedback shift registers alone which is easy and simple to build and to produce good statistical properties. Stream ciphers are bit (process bit by bit) and word (process word by word) oriented.Word oriented stream ciphers are complex to implement but produce better throughput. Speed of the cipher depends upon the amount of the hardware. Grain is a bit-oriented synchronous stream cipher. In synchronous cipher the keystream is generated separately from the plaintext. The memory requirements for implementing this protocol are LFSR as an 80 bit register, NFSR holds 80 bits, an 80-bit keystream and a 64-bit initialization vector (IV).

Terms:

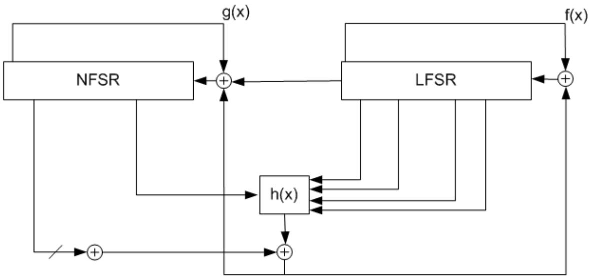

Figure 4: Grain Cipher

f(x): feedback function of LFSR

S_i: state of LFSR (0< i<79)

g(x): feedback function of NFSR

b_i: state of NFSR (0<i<79)

h(x): filter function of both LFSR and NFSR

K_i: Key bits(0<i<63)

IV_i: Initialization vector (0<i<79)

The construction of the cipher can be explained in two phases. In the first phase, the cipher is initialized with the key and the IV bits and NFSR is loaded with the key bits ($b_i = K_i$). The first 64 bits of the 80-bit LFSR are loaded with the 64 least significant bits of the initial vector IV and the remaining bits are filled with one's so as to avoid the zero state of an LFSR as its working is not possible. The linear feedback register (LFSR) is fed by the feedback polynomial f(x) and the cipher to generate the new state of LFSR. The feedback polynomial f(x) is given in the equation 3.7. Similarly the non-linear feedback register is fed by the feedback polynomial g(x) and the cipher to get the new state of NFSR. The non-linear feedback polynomial is given in the equation 3.8. The cipher is clocked 160 times without producing any running keystream.

$$f(x) = 1 + x^{18} + x^{29} + x^{42} + x^{57} + x^{67} + x^{80} \tag{6}$$

$$g(x) = 1+x^{18}++x^{20}+x^{28}+x^{35}+x^{43}+x^{47}+x^{52}+x^{59}+x^{66}+x^{71}+x^{80}+x^{17}.x^{20}+x^{43}.x^{47}$$

$$+x^{45}.x^{71}+x^{20}.x^{28}.x^{35}+x^{47}.x^{52}\ x^{59}+x^{17}.x^{35}.x^{52}.x^{71}+x^{20}.x^{28}.x^{43}.x^{47}+x^{17}.x^{20}.x^{59}.x^{15}$$

$$+x^{17}.x^{20}.x^{28}.x^{35}.x^{43}+x^{47}.x^{52}.x^{59}.x^{65}.x^{71}+x^{28}.x^{35}.x^{43}.x^{47}.x^{52}.x^{59} \tag{7}$$

The filter function **h(x)** which uses five variables from LFSR and NFSR together is again xored (masked) with the state of the NFSR to produce the output (keystream). The output of the filter function is fed to both the LFSR and NFSR using a XOR operation. The filter function is expressed as in equation 3.8. The feedback polynomial of the linear feedback shift register and the non-linear feedback shift register are shown in the equation 3.6 and 3.7. In the second phase, the updates are progressed in background in every register. The update functions of the linear feedback shift register and the non-linear feedback shift register are shown in the equation 3.9 and 3.10.

$$\mathrm{h(x)}=x_1+x_4+x_0x_3+x_2x_3+x_3x_4+x_0x_1x_2+x_0x_2x_3+x_0x_2x_4+x_1x_2x_4+x_2x_3x_4 \tag{8}$$

$$S_{i+80} = S_{i+62} + S_{i+51} + S_{i+38} + S_{i+23} + S_{i+13} + S_i \tag{9}$$

$$\begin{aligned}
&b_{i+80} = s_i + b_{i+62} + b_{i+60} + b_{i+52} + b_{i+45} + b_{i+37} + b_{i+33} + b_{i+28} + b_{i+21} + \\
&b_{i+14} + b_{i+9} + b_i + b_{i+63}\ b_{i+60} + b_{i+37}\ b_{i+33} + b_{i+15}\ b_{i+9} + b_{i+60}\ b_{i+52}\ b_{i+45} \\
&+b_{i+33}\ b_{i+28}\ b_{i+21} + b_{i+63}\ b_{i+45}\ b_{i+28}\ b_{i+9} + b_{i+60}\ b_{i+52}\ b_{i+37}\ b_{i+33} + b_{i+63} \\
&b_{i+60}\ b_{i+21}\ b_{i+15} + b_{i+63}\ b_{i+60}\ b_{i+52}\ b_{i+45}\ b_{i+37} + b_{i+33}\ b_{i+28}\ b_{i+21}\ b_{i+15}\ b_{i+9} \\
&+ b_{i+52}\ b_{i+45}\ b_{i+37}\ b_{i+33}\ b_{i+28}\ b_{i+21}
\end{aligned} \tag{10}$$

Grain cipher is mostly suitable for hardware implementations. The security requirements correspond to 2^{80} complexity. Grain is implemented in a 160-bit memory. The period of the output functions depend upon all the functions as input of the NFSR is masked with the output of the LFSR and then the key and IV. The criteria behind the masking is in order to balance the NFSR state. In the key initialization phase, both the contents are combined and clocked 160 times in the filter function before producing the running key which shows its randomness property. The last 16 bits of the LFSR are not used in the feedback function or filter function as to use that space to increase the speed by the hardware. Grain can be implemented at rates 1 bit /clock cycle to 1 word/clock cycle, but preferred to be used at 1 bit/clock cycle due to high focus on small hardware complexity [4].

3.2.3 Summary

In symmetric cryptography, there are two types of implementing lightweight ciphers. The first one is picking a state of art technique like AES and building a lightweight cipher based on its model. The second one is building a domain ad-hoc specific new lightweight protocol. The absence of decryption is a factor that can reduce the overall hardware of the cipher but cannot avoided in many applications. Several domain specific ciphers EPCBC and PRint are developed based on Present cipher which are used for electronic product code (EPC) encryption applications. Stream ciphers are a type of symmetric ciphers and are well suited for constrained devices. Despite their evolution and efforts of concern, block ciphers are considered superior to them. Stream ciphers are simple and speeder when in implemented in hardware but takes lengthy initialization. Hash functions are another research filed of lightweight cryptography. They contain too large hardware for constrained devices (i.e more than 3000GE). After release of Present cipher, there are many efforts to build lightweight hash functions. The advantages of hashes are efficiency with low cost devices. Symmetric ciphers and hash functions are the latest methods of ciphering devices for protecting privacy and security of systems (esp. RFID devices).[31]

4 Security and Privacy in RFID

RFID tags operate in an inherently insecure and noisy environment. This type of communication environment calls for measures like security and privacy. Security of a system is the ability to keep its information secure from the adversary. The security of the RFID system wholly depends upon the software and hardware implementation used in the system. Common security attacks and problems of RFID networks are physical attacks, forward security, backward security, passive and active eavesdropping. Privacy of a system is the ability of the RFID system to keep the meaning of the information secure from the adversary. Privacy of the system wholly depends upon the mechanism of the data preservation while authenticating the legitimate tags and readers. Some of the common properties to maintain privacy in the system are backward untraceability, forward untraceability, tag anonymity, tag indistinguishability and tag unlinkability. Other attacks related to privacy and security are denial of service (DoS) attack, tag impersonation attack, server impersonation attack, man in the middle attack and the replay attack[9,14].

4.1 Privacy properties

Tag anonymity

Tag anonymity is a property which guarantees to secure users identity information from any adversary. A tag carries users private information such as name, age, location, etc, which are meant to be protected from other sources. Gaining the information of the tag's attributes lead to vulnerability of the users privacy and security.

Tag unlinkability

Tag unlinkability can be defined as a property that guarantees to secure all linkable information of any user based on the record of his past behavior. Tags are mostly reused by many manufacturers in which by physically tampering the tag and gaining its information allows the adversary to access the information of the previous owner. Cryptographic techniques play a vital role in securing the private information which disallow the adversary to predict any of tag's information [21].

Forward untraceability

Forward untraceability is a property where an adversary can succeed in gaining the internal state of the tag at time τ but cannot guess or predict the future state of the tag after time τ. Forward untraceability and Backward untraceability are the two strong privacy notions that are proposed to ensure forward and backward privacy in the offline and online communication [33]. Robust pseudo random bit generators (PRBG) are used to ensure forward untraceability [15]. Robust PRBG's are rather considered stronger than the standard PRBG's in achieving forward and backward privacy. Standard PRBG's use keyed hash function in contrast to robust PRBGs [7,21].

Backward untraceability

Backward untraceability is a property where adversary can succeed in gaining the internal state of the tag at time τ but fail to guess any of the previous states before time τ. The adversaries execute the forward and backward tracing with the help of the CORRUPT query.

Tag impersonation attack

Tag cloning refers to gather the identifying information of the tag to create a duplicate tag. These tags are used purposefully in several instances where the legitimate tag has a role to play. All the legitimate readers authenticate these tags by confirming their fake identity. This leads to the impersonation of an authorized user which is a severe privacy issue that has to be avoided. The solution to this issue to implement a mutual authentication scheme that authenticates a tag and the reader (respectively tag and the back-end server) mutually by means of lightweight cryptographic schemes [3].

Server impersonation attack

Server impersonation attack occurs when an adversary impersonates a valid server to a tag or a reader. In this case, desynchronize can occur where the tag updates the data but the server does not. Hence the tag and the server are said to be incapable of successful communication. The tag is said to

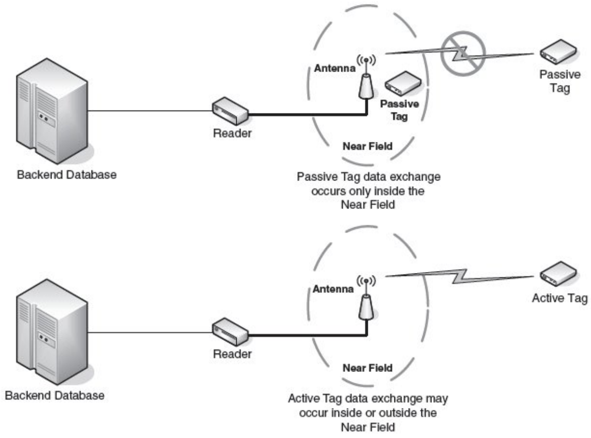
Antenna
Passive Tag
Passive Tag
Reader
Backend Database
Near Field
Passive Tag data exchange occurs only inside the Near Field
Antenna
Active Tag
Reader
Backend Database
Near Field
Active Tag data exchange may occur inside or outside the Near Field

Figure 5: Passive Eavesdropping and Active scanning

release all its stored secrets and the genuine server is still not update any information in its DB entry.

4.2 Affects of Adversaries

Eavesdropping is a spying technique to retrieve secret information without the permission of the legitimate party involved in the conversation. Eavesdropping can take place in any type of communication in which the exchangeable information between the legitimate sender and the receiver is being read anonymously. Adversary (eavesdropper) is a third party who spies on two legitimate parties without their permission. Eavesdropping can take place in wired, wireless communication, half and full handshake communication and multiplexed communication. The main aim of the eavesdropper (adversary) is to acquire partial information from all the possible channels within the near field to acquire the remote functioning of the devices without their consent [13].

4.2.1 Passive Eavesdropping

Passive eavesdropping is a problem of unauthorized reading of tag's information by illegitimate authentication through forward communication channel. A forward channel is the channel through which the information between reader and tag is transmitted whereas the backward channel is the channel through which the information between the tag and the reader is transmitted. Passive eavesdropping takes place within the near field region. A near field region is an area at which the tag is visible to the reader. The main aim of an passive adversary is to acquire the EPC of the tag and gain the information of the customer's personal attributes and traces. This type of eavesdropping requires a low range communication, like the adversary with the fake reader needs to be within 10m distance of the tag to trace or scan the information. [10].

4.2.2 Active Scanning

Active Scanning or Active eavesdropping is defined as unauthorized reading of both the backward and forward communication channels. The main aim of an active adversary is unauthorized access of legitimate tags and readers and totally modify the contents of the tag and the reader behavior. It can also be possible to manipulate the back-end server if an active adversary successfully impersonates a tag [10].

4.3 Mutual Authentication

In an RFID system, the reader is considered as a single powerful device and a secure participant as it evaluates information of the tag from the server through a wired channel in most cases. On the other side, tags are inexpensive and insecure devices which are prone to attacks and are communicated through wireless channels. Privacy in a system makes sure the adversary cannot link relationships of the tag based on its previous or present information. Clearly, the purpose of RFID tag is to identify itself just to the reader and nobody else. The primary aim of an adversary in an RFID system to basic knowledge is to physically tamper a tag and retrieve possible information and apply several algorithms to break the logic of the system. The main concern of the RFID system is (i) not to allow the tag to become corrupted and release end users information to the third parties who later take the benefit of the private information. (ii) not to allow a reader to become corrupted as it can expose the tag and server information at a broader view. A malicious reader can trace the tag's information and can manipulate the data in the back end server and vice versa. Hence there is a necessity for authentication of all the three devices (namely tag, reader and back-end server) to prove their legitimacy [28]. Several cryptographic protocols (symmetric and asymmetric) have been proposed to help devices in mutually authenticating each other in the RFID system. Mutual authentication schemes are designed with the assistance of time-stamps, pseudo-random numbers, hash functions and cryptographic protocols. In this thesis, mutual authentication is proved using symmetric ciphers like PRESENT, GRAIN and Humming Bird-2. The main aim of a cipher is to randomize the state information on one end which could be retrieved by decryption at the other end only. The main aim of the mutual authenticating protocol is to achieve user freedom from privacy threats from any third party. Mutual authentication helps in curbing the threats such as tag impersonation, server impersonation, etc; [8].

4.3.1 Mutual Authentication using Cryptographic Hash-functions

Terms

Hash: hash function combines multiple strings into an unpredictable string
CRC: cyclic redundancy check
SK: shared key
MK: main key
KD: key decryption

The mutual authentication protocol proposed in [27] is a novel hash based mutual authentication scheme executed between the tag and the reader assuming the communication between the reader and back-end database is secure. The hash function uses a pseudo random generator to execute the hash process. According to the protocol, the back-end database stores the following values for every tag T_i: (tag ID (ID), shared key (SK), the main key (MK)). After initialization of the process, the tag and the database execute the authentication procedure via the reader [27]. The authentication procedure is initialized by the reader. Upon receiving a request from a reader, the tag generates a random number N_1 by using a pseudo random generator. Eventually the tag calculates two values namely H_1 and CRC_1. They are defined as:

$$\begin{gathered} H_1 = \text{Hash}(N_1 \mid\mid \text{ID} \mid\mid \text{SK}) \\ CRC_1 = \text{CRC}(N_1 \mid\mid H_1) \end{gathered} \tag{11}$$

The tag sends the values N_1, H_1 and CRC_1 to the reader subsequently. Upon receiving the data values (N_1, H_1, CRC_1) from the tag, the reader recalculates $CRC^{'}$ to verify the accuracy of the received values.

$$CRC^{'} = \text{CRC}(N_1 \mid\mid H_1) \tag{12}$$

If the equation $CRC^{'} = CRC_1$ tends to be correct, the reader sends the values (N_1, H_1) to the back-end database. Else the reader just abandons the authentication process. Upon receiving the values from the reader, the back-end database checks the database if there exists an $ID^{'}$ that satisfies the following equations:

$$\text{Hash}(N_1 \mid\mid ID \mid\mid \text{KD}(\text{MK} \mid\mid ID^{'})) = \text{Hash}(N_1 \mid\mid \text{ID} \mid\mid \text{SK}) \text{ where KD}(\text{MK} \mid\mid ID^{'}) = \text{SK} \tag{13}$$

In case of the above equation satisfying both sides confirms that the authentication is successful. The server calculates shared key using the key decryption function and therefore verifies the equation. The authentication succeeds or else the process get terminated by the server. Upon a successful completion, the server sends $ID^{'}$ to check the conformity with the tag and the reader. Upon receiving the value ($ID^{'}$) from the back-end database, the reader computes two new values:

$$H_2 = \text{Hash}(N_1 \mid\mid ID^{'})$$
$$CRC_2 = \text{CRC}(N_1 \mid\mid H_2)$$

(14)

The reader sends the values (N_1, H_2, CRC_2) to the tag. Upon receiving the values from the reader, the tag verifies the value of N_1 whether it is generated by itself and if not, it abandons the authentication process. Else, it recalculates $CRC_2^{'}$ and checks if it is equal to the received value CRC_2. If $CRC_2 \neq CRC_2^{'}$, it abandons the authentication process. Else, it recalculates $H_2^{'}$ and checks if it equal to H_2. If $H_2^{'} \neq H_2$, then it abandons the authentication process. The tag-to-reader authentication is completed if the above two cases are successful. Hence the mutual authentication process is said to be successful.

Hash functions have been producing good results in cryptographic applications. Hashes are used in several security systems where the information is of utmost importance[19]. Cryptographic hash functions produce great results in lenience to a comparatively high gate size relative to the symmetric ciphers.

4.4 Symmetric Encryption based Privacy

Privacy is a basic requirement to be fulfilled when RFID tags are carried by the users personally from place to place. These tags are supposed to be free without any possibility of spying. Mutual authentication is a procedure which allows systematic verification of a legitimate tag and a reader participating in the communication by agreeing to a certain protocol. Symmetric ciphers are strong, flexible and contain less overhead compared to the public key ciphers in the context of lightweight cryptography. In this section, it is assumed that the symmetric ciphers are appropriate in helping the RFID system to solve the privacy issues.

4.4.1 Block cipher Authentication using GCM mode

Block ciphers have a special set of operations known as "block cipher modes of operation" that are used externally to the encryption process to provide services such as confidentiality and authenticity. There are several block cipher modes that are developed till today in which the general four modes used are Electronic code book (ECB), Cipher-block-chaining mode (CBC), Cipher feedback mode (CFB) and the output feedback mode (OFB). Several other modes of operation are Galois Counter mode (GCM), Counter Chainblocking mode (CCM), etc. The modes are not explained in the report as they are not required in detail. All the block cipher modes rendezvous with each other when used in similar contexts. In this thesis, we show the authentication procedure of PRESENT block cipher using the GCM mode.

Terms
P_i : Plaintext of n blocks.
A_i : Additional authenticated data
IV : Initialization vector, 64 bits (known to tag, reader and the server).
S : Output string generated by encrypting both ciphertexts.
MSB_T : Most significant bit of the authenticated tag T, used for authentication.
$GCTR$: Galois counter function
$GHASH$: Galois keyed hash function

The mutual authentication procedure contains three main functions namely Present, GCTR and GHASH. Present is a block cipher which encrypts the certain information to produce the first ciphertext. Similarly, GCTR produces the second ciphertext. Both the ciphertexts are then encrypted using the keyed hash function GHASH to produce a temporary output. The output obtained from the GHASH is used in producing an authentication tag. The primary ciphertext C_1 is produced by the block cipher PRESENT by encrypting the key and 64 bit 0's as shown in the equation 4.5. While the encryption proceeds, GCTR function is initialized parallely using initialization vector

(IV) and the plaintext (P_i) to produce the second ciphertext block (C_2) as shown in the equation 4.5. The three variables namely ciphertext from block cipher (C_1), authenticated data (A_i) and ciphertext from GCTR (C_2) are key hashed by GHASH function to produce the temporary output S.

$$\begin{aligned}
&C_1 = \text{PRESENT}(0^*, \text{key})\\
&C_2 = \text{GCTR(IV, P)}\\
&\text{S} = \text{GHASH } (C_1, \text{ A}, C_2)\\
&\text{T} = \text{GCTR(IV, S)}
\end{aligned} \tag{15}$$

The temporary output S along with the initialization vector are used by GCTR function to generate the authentication tag T in which the most significant bit t of the authentication tag T is used as the authentication code. Both the ciphertext C_1,C_2 and T are send to the reader which inturn forwards them to the back end server. The server decrypts the second ciphertext and the initialisation vector by GCTR(IV, C_2) to retrieve the plaintext P. Secondly, the server calculates the authentication tag $T^{'}$ as shown in equation 4.6 and checks if t = $t^{'}$. Once the server realises the match, the mutual authentication is successful. In case there is no match, the server discontinues the authentication procedure.

$$\begin{aligned}
&\text{T} = \text{GCTR(IV, S)}\\
&t = MSB_T \text{ [GCTR (IV, S)]}\\
&t^{'} = MSB_T \text{ [GCTR (IV, S)]}
\end{aligned} \tag{16}$$

The server then calculates a new authentication bit $t^{'}$ as shown in the equation 4.6 and sends it to the reader. The reader recalculates the $t^{'}$ using the formula shown in the equation 4.6 and confirms the successful authentication of the legitimate tag. The whole authentication process works on three key functions which are Present, GCTR and GHASH. While the block cipher tries to strengthen the key string by encrypting the key with an other string, the other two functions supposedly did the routine work like encrypting the plaintext and further protection of the resulted plaintext as shown in the equation 4.5. The Present cipher is solely maintained to encrypt the key with the zero string. The block cipher's 31 round iterative process lay a strong ground in which the adversary finds hard to break the cipher even by using reverse engineering. The cipher which combines encrypted data with the cipher produced by the Galois counter is a reliable effort for safeguarding the privacy of the information.

4.4.2 Stream cipher authentication using Challenge-Response approach.

The estream ciphers like Grain, Trivium are proposed to be used for low-constrained authentication which are the only lightweight stream ciphers in present use today. The estream project is mainly dedicated to develop lightweight stream ciphers for low-constrained devices for security applications. Grain and Trivium are awarded as the best among the few estream protocols. In this subsection, we consider Grain to mutually authenticate the devices in the RFID system.

Terms

K: key used by the tag

(K^i,K^{new}): key pair (current and new) stored in the key array of the server

a: random value computed by the reader

b: random value computed by the tag

c: Encryption of IV and key K computed by the tag for authentication as challenge

d: Encryption of IV and key K computed by the reader for authentication as response)

G_t: Grain cipher encryption and decryption at the tag
G_r: Grain cipher encryption and decryption at the reader
G_s: Grain cipher encryption and decryption at the server

The communication is initialized by the reader as the passive tag has no equipment to produce a signal. Grain is solely maintained to produce the keystream which is later conjoined with the plaintext using a binary arithmetic operation (like XOR). The reader sends a random value a to the tag. On receiving the value a from the reader, the tag computes a random value b and concatenates both a and b to produce the initialization vector as shown in the equation 4.7. After computing the IV, the tag also computes a new value c produced by Grain by encrypting IV and the key as shown in the equation 4.7. The tag returns c to the reader. On receiving the value c from the tag, the reader decrypts the tag using G_t(IV, K) and then searches the value K in the key pair of the server. If the K match among the key pair of the server, the authentication is said to be successful.

$$\begin{aligned}
&\mathrm{IV} = a \mid\mid b \\
&c = G_t(\mathrm{IV{,}K}) \\
&d = G_r(\mathrm{IV{,}K})
\end{aligned} \tag{17}$$

The reader then computes value d using the Grain cipher by encrypting IV and its key as shown in the equation 4.7. The reader returns the value d to the tag. The tag decrypts the value d using the Grain cipher. After retrieving the key the tag checks the key K^i with the key K in the tag. If there is a match, the authentication is successful. The tag updates its new key to G_s(K,IV).K_i is one of the key pair stored in the server. The server keeps track of the current keys of the tag and server. The server calculates a new key K^{new} after every authentication process and sends it to the tag.If any of the tags are not matching the keypair in the server, the authentication is cancelled.

The strength of the stream ciphers lie on generating the keystream. Grain being a rigorous stream cipher produces a 160 times clocked keystream. This allows the cipher to perform despite the authentication process is simple. The mutual authentication process being simple is not an obstacle for the stream ciphers as they preserve the message better until the key is secure because the encryption between the key and the plaintext is done by a simple operation. All three devices can apply the Grain cipher function to retrieve and send the private information in order to prove their legitimacy.

4.5 Analogy

Security and privacy are evaluated based on the amount of information the adversary claims to be genuine. Privacy is said to be under threat if an adversary gains non-negligible information of a legitimate tag (or reader respectively). The main aim of the symmetric protocols is to let the RFID system to run and control the devices and maintain anonymity, untraceability, forward and backward privacy, tag and server impersonation [6]. Hash functions are mostly used to conserve a whole file as it encrypts the given file name or number into a hash value which is hard to predict. Cryptographic hash functions are used in many security applications for authentication as they are practically hard to break. Tags contain permanent identifiers and some tags are re-writable as well. Tag anonymity and untraceability are conserved by hash functions rather than the symmetric ciphers in case of re-writable tags as the cipher functions are combined with the mutual authentication functions and work online. Hash functions carry the tag identifiers along with the message and the hashes are hard to break as no two hashes are the same. The symmetric ciphers are not bounded to safeguard the information of the tag identifiers offline as physically tampering the tags do violate the authentication protocol. The ciphers can safeguard the tag's identity if and only if the tag works online and in passive mode.

Hash functions provide a better unlinkability than compared to the symmetric ciphers as the hashes carry all the folder information as a single message and no two hashes are the same to predict. Symmetric ciphers can

	key size	block size	gate size
Hash function	-	128	7000
Present(block cipher)	80 bits	64 bits	1500
Grain (stream cipher)	64 bits	1 bit	1200

Table 6: Attributes of the protocols

guarantee the anonymity only if the channel is secure because the adversary can physically tamper the tag and get the previous owner information of the tag which is unethical. The ciphers guarantee that no malicious reader can act as the original one as the symmetric authentication using the ciphers are hard to predict as they need to correspond to the challenge response approach. Forward untraceability is a situation that has to be safeguarded after the state had been predicted and therefore do not let the adversary gain any future state based on the obtained information. Hash functions need large hardware because of their complex algorithms and their gate equivalent is approximately around 7000 gates. When the adversary gains the internal state of the cipher by tampering the tag or randomly responding to the reader can only be allowed to contain or maintain the state for that particular instance while the challenge from the reader comes into play, the illegitimate tag cannot predict the key and other attributes of the authentication process to complete the mutual authentication process. The block cipher having 31 rounds of added round key iterative process consumes lot of time for the adversary to gain the key and respond to the reader instantly. Hence it is not possible for the adversary to gain the future state of the cipher.

Grain being a stream cipher plays a prominent role in the challenge response authentication scheme as it encrypts the key and the initialization vector to produce the secondary cipher which is encrypted along with the message bit to produce the primary cipher. Grain having a mixed linearity feature as it uses both linear and non-linear feedback registers produce a vigorous random cipher which is hard to predict even the current state is revealed. Therefore the symmetric ciphers always hold the adversary in gaining the future state with the help of the authentication scheme. In terms of backward untraceabil-

	Hash function	PRESENT	GRAIN
Tag anonymity	G	G	G
Tag Unlinkability	G	G	G
Forward Untraceability	PG	G	G
Backward Untraceability	PG	PG	PG
Tag Impersonation attacks	PG	PG	PG
Server Impersonation attacks	PG	G	G

Table 7: Properties for privacy preservation, G - Guaranteed, PG - Partially guaranteed, NG- Not guaranteed

ity, physically tampering the tag and gain the internal state of the cipher and predicting the previous cipher and the key is possible as the adversary has enough time to draw it. Hence the backward privacy is partially guaranteed given that the physical tampering of the tag is a possibility. Controlling the reader using an illegitimate tag is one way the adversary approach to the RFID system to retain the key for the later stage but it seems neither the symmetric ciphers nor the hash function does allow the adversary to totally gain the key and the only possibility at that moment is to gain hold of the reader for a single transaction where the identifiers in the tag and the server are different and are found out in the later transaction. The ultimate aim of an adversary in impersonating a tag is to gain little information of the tag from the reader (or the server respectively). The accomplishment of obtaining information in a single transaction cannot help the adversary completely as the reader comes into contact with the server at some point (challenge response time out). Similarly the adversary tries alternatively by impersonating the server (or the reader respectively) but the legitimate tag can contact the reader (or the server respectively) but awaits to the response from the server where the adversary fails in synchronization on the wired channel between the reader and the server. As the database details do not match after a timeout occurs, either of the reader and the server process discontinues due to the termination of the authentication procedure.This shows that the adversary cannot prevail in the current scenario even if he succeeds to entertain either of the reader or the server with little key input information (or by using a malicious device). Symmetric lightweight ciphers save time in recognizing the threat to the RFID device from an adversary in compared to the hash functions as hashes need more hardware and processing for maintain and run the cyclic redundancy check method (CRC) at every end.

5 Conclusion

Today RFID systems have reached several destinations from a wired-home application to a retail supply chain management systems. As the RFID technology provides cheap source of maintenance, there is a rapid increase in its utilization and hence new security challenges had arose for the designers as well as the consumers. Cryptographic tools such as hash functions and symmetric ciphers are developed for the RFID systems to secure its environment from the third party interference and spying of the potential information that are exchanged. Cryptographic research community had undergone several challenges to conserve the security and privacy of the end-users and some are yet to be achieved. Lightweight cryptography was a new development and approach tool for the real security applications which has been proven reasonably an efficient solution in privacy for the end users. Several privacy models that are proposed by Avoine[11] and Juels, Weis [12] are the best methods till date but are practically hard to implement on the low-constrained platform. In this thesis, the importance of the lightweight protocols has been analysed to greater limits while preserving privacy and security of the RFID system using mutual authentication schemes such as challenge-response scheme and GCM mode scheme.

The hard reconciling issues like anonymity, untraceabilty and impersonation of RFID devices are safeguarded by the authentication schemes while the data is preserved by the symmetric ciphers. With the large use of RFID tags everywhere, especially the low cost tags demanded the need for symmetric ciphers and hardware efficient hash functions. The symmetric ciphers are selected and organized based on the circuit size or the gate size which are considered optimal around 2000-3000 gates or less, passive devices not consuming much energy (which limits the range of the device). The use of the initialization vector (IV) allows rapid re-initialization of the cipher without re-keying the cipher. The IV's are not repeated when used in the GCM mode as the cipher becomes insecure as counter functions are applied more than once using the same IV. The authentication schemes provide an organization for the devices that lays the foundation for a secure communication with a mutual agreement of providing genuine information using challenge response behavior and avoiding the third party interference.

The primitive use of symmetric ciphers like block ciphers in the construction of the hash functions started when the well known block cipher DES (Data encryption standard) was developed. The introduction to lightweight cryptography had separated the two protocols (symmetric ciphers and hash functions) as the cipher size and hardware size had to be reduced in order to prevail in the low cost arena. The new style of compact hashing using 64 bit and 128 bit hash outputs had driven the hash functions into the lightweight cryptography. The hashes have proven themselves as a strong model of cryptography as they use much larger range than other ciphers in the encryption but at the same time they consume large hardware area. The range of the hardware requirements is directly proportional to cost effectiveness and it is left to the application that demand the requirements.

The main design goal of a cipher is to reduce the total number of logic gates required to materialize the cipher. This metric is called Gate Equivalent (GE).A small GE predisposes that the circuit is cheap and consumes less power. An optimum circuit to protect and maintain efficiency is 1000-3000GE because passive tags run on the host power.. Other design goals are memory, processing power, throughput and power savings.

6 References

[01] Ng, Ching Yu, Contributions to RFID security, Doctor of Philisophy Thesis, School of Computer Science and Engineering, University of Wollong, 2012.

[02] Boyeon Song - "RFID Authentication Protocols using Symmetric Cryptography", Phd Thesis, Royal Holloway, University of London, 2009.

[03] Mike Burmester and Breno de Medeiros - "RFID Security: Attacks, Countermeasures and Challenges"

[04] Martin Hell, Thomas Johansson and Willi Meier. "Grain - A Stream Cipher for Constrained Environments".

[05] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe. - "PRESENT: An Ultra-Lightweight Block Cipher".

[06] Olivier Billet, Jonathan Etrog and Henri Gilbert - "Lightweight Privacy Preserving Authentication for RFID Using a Stream Cipher"

[07] Raphael C.-W. Phan, Jiang Wu, Khaled Ouafi and Douglas R. Stinson - "Privacy Analysis of Forward and Backward Untraceable RFID Authentication Schemes"

[08] Boyeon Song, Chris J Mitchell - " RFID Authentication Protocol for Low-cost Tags"

[09] Aikaterini Mitrokotsa, Melanie R. Rieback and Andrew S. Tanenbaum - "Classification of RFID Attacks"

[10] Maxim Kharlamov - "An Overview of RFID Security and Privacy threats"

[11] Lynn A. Fish, Wayne C. Forrest - "The State of RFID Privacy and Security 2010: Issues and Solutions"

[12] Juan Ignacio Aguirre, "EPCglobal: A Universal Standard", February 2007.

[13] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels - "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems"

[14] J. Aragones-Vilella, A. Martnez-Balleste and A. Solanas - "A Brief Survey on RFID Privacy and Security"

[15] Jihwan Lim, Heekuck Oh, and Sangjin Kim - "A New Hash-Based RFID

Mutual Authentication Protocol Providing Enhanced User Privacy Protection"

[16] RFID Privacy and Security, Securing RFID tags from Eavesdropping, RSA Laboratories. "http://www.rsa.com/rsalabs/node.asp?id=2118"

[17] Axel York Poschmann - "LIGHTWEIGHT CRYPTOGRAPHY: Cryptographic Engineering for a Pervasive World.", Phd Dissertation, Bochum, February 2009. http://eprint.iacr.org/2009/516.pdf

[18] Diana Maimut, Khaled Ouafi - "Lightweight Cryptography for RFID tags"

[19] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, and Y. Seurin - "Hash Functions and RFID Tags: Mind the Gap"

[20] J. Wu and D.R. Stinson - "How To Ensure Forward and Backward Untraceability of RFID Identification Schemes By Using A Robust PRBG"

[21] Miyako Ohkubo, Koutarou Suzuki and Shingo Kinoshita - "Cryptographic Approach to 'Privacy-Friendly' Tags".

[22] Xinxin Fan and Guang Gong, Daniel W. Engels and Eric M. Smith - "A Lightweight Privacy-Preserving Mutual Authentication Protocol for RFID Systems"

[23] SIMSON L.GARFINKEL, ARI JUELS, RAVI PAPPU "RFID Privacy: An Overview of Problems and Proposed Solutions"

[24] Jan Krhovjak, Dissertation thesis, Brno 2009 - Cryptographic random and pseudorandom data generators

[25] MIKE BURMESTER, JORGE MUNILLA - Lightweight RFID authentication with forward and backward security

[26] Joseph Sterling Grah, Master of Science thesis - "Hash functions in Cryptography"

[27] Li huixian, Yin Ping, Wang Xuan, Pang Liaojun - "A Novel Hash based RFID Mutual Authentication Protocol".

[28]Radu-Ioan Paise, Serge Vaudenay - "Mutual Authentication in RFID - Security and Privacy."

[29] Katholieke Universiteit Leuven - "Ultra-lightweight encryption method becomes international standard". Retrieved 2012-02-28.

[30] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw and Y. Seurin: "Hash functions and RFID tags- Mind the Gap".

[31] Joaquin Garcia-Alfaro, Georgios Lioudakis, Nora Cuppens-Boulahia, Simon Foley and William M. Fitzgeral; "Data Privacy Management and Autonomous Spontaneous Security".