1-1-2006

Security protocol based on random key generation for an Rfid system

Karthik Raghavan
University of Nevada, Las Vegas

SECURITY PROTOCOL BASED ON RANDOM KEY GENERATION FOR AN RFID SYSTEM

by

Karthik Raghavan

Bachelor of Engineering, Computer Science
University of Madras, India
2003

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Computer Science
School of Computer Science
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
December 2006

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

Thesis Approval
The Graduate College
University of Nevada, Las Vegas

JANUARY 4TH, 20 07

The Thesis prepared by

KARTHIK RAGHAVAN

Entitled

SECURITY PROTOCOL BASED ON RANDOM KEY GENERATION FOR AN RFID SYSTEM

is approved in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Yoohwan Kim
Examination Committee Chair

Dean of the Graduate College

Ajoy Kumar Datta
Examination Committee Member

Examination Committee Member

Graduate College Faculty Representative

ABSTRACT

Security protocol based on random key generation for an RFID system

by

Karthik Raghavan

Dr. Yoohwan Kim, Examination Committee Chair
Assistant Professor of Computer Science
University of Nevada, Las Vegas

Radio Frequency Identification (RFID) is a technology, which describes the transmission of unique information by a wireless device, over Radio waves, when prompted or read by a compatible reader.

The basic components in implementing RFID are RFID tags which are small microchips attached to a radio antenna, mounted on a substrate, and a wireless transceiver/reader that queries the RFID tags.

This thesis deals with research issues related to security aspects in the communication between an RFID tag and its reader. More precisely, it deals with a new, simple and efficient security protocol based on an encryption that uses the concept of regular public key regeneration, which can be effortlessly adopted in an RFID application.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

This thesis would not have been possible without the support of many people. Many thanks to my Advisor, Dr. Yoohwan Kim, whose support and suggestions were invaluable in my research and in this taking shape. I am also very grateful to Dr. Ajoy K. Datta and Dr. Laxmi Gewali for having supported me, throughout my academic years. I would also like to thank Dr. Rama Venkat for being a part of my Advisory Committee.

Last but not the least, I would like to thank my parents, my sister and my friends who have always stood by me and were a major factor in me making me what I am.

CHAPTER 1

INTRODUCTION

Radio Frequency Identification (RFID) is an Automatic Identification method that works by storing and retrieving data remotely, using simple transponder devices [1]. A typical RFID system consists of a low-power transponder chip, called the RFID Tag that stores the identification data, a Radio Frequency Reader that can receive and transmit data using Radio Frequency waves, and a Backend application that stores all the data needed by the system which can validate the tag as being a legitimate one. RFID technology is finding wide-scale use these days, in various applications basically for the reason that it is easy to implement and is of relatively low cost.

An RFID tag is an object that can be attached to, or incorporated into a product, animal, or person for the purpose of identification using radio waves. Chip-based RFID tags contain silicon chips and antennas. Passive tags require no internal power source, whereas active tags require a power source [1]. Apart from being classified based on their power source, RFID tags are also classified into different classes, based on their functional characteristics. The RFID Class Structure classifies tags as belonging to one of five classes: Class 1 (Identity Tags), Class 2 (Higher Functionality Tags), Class 3 (Semi-Passive Tags), Class 4 (Active Ad Hoc Tags), or Class 5 (Reader Tags) [2]. The following figure shows the inlay of a passive RFID tag:

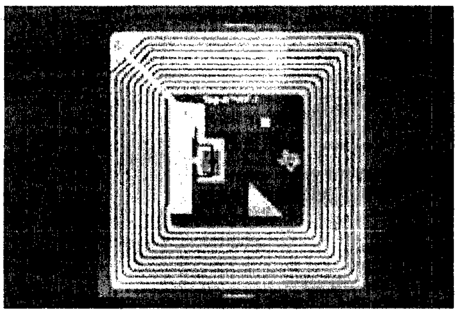


Figure 1 An RFID Tag that is being used in applications today. Inlay of Texas Instruments' 14mmX14mm HF Tag [3]

An RFID reader is a device that is used to interrogate an RFID tag. The reader has an antenna that emits radio waves; the tag responds by sending back its data. A number of factors can affect the distance at which a tag can be read (the read range). The frequency used for identification, the antenna gain, the orientation and polarization of the reader antenna and the transponder antenna, as well as the placement of the tag on the object to be identified will all have an impact on the RFID system's read range [4].

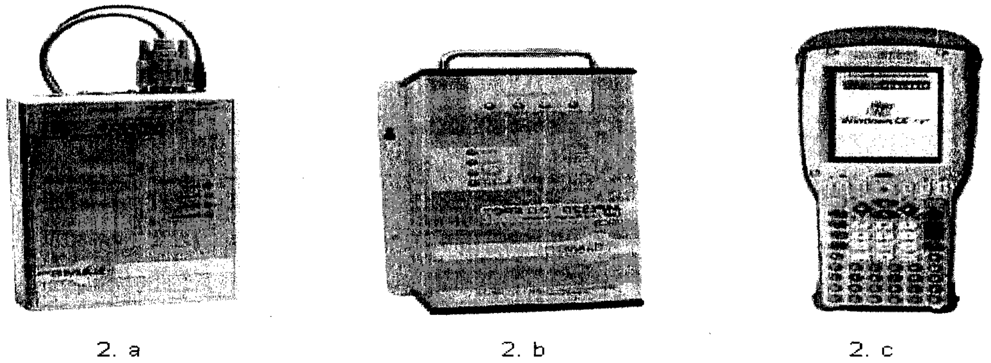


Figure 2.a High-Frequency Reader [5]; Figure 2.b Ultra-High Frequency EPC Reader [5]; Figure 2.c High-Frequency Handheld Reader [5]

The backend application is a high-end computer or a server that manages the data required for the application that is using the RFID system. Typically, it could be a database interfaced with application-specific functions. It also defines the business logic for interpreting raw RFID data and the actions associated with it. For this reason, the backend application can never go down and it's connectivity with the readers should be maintained and standardized, always.

1.1 History of RFID

In 1946 Léon Theremin invented an espionage tool for the Soviet government which retransmitted incident radio waves with audio information. Even though this device was a passive covert listening device, not an identification tag, it has been attributed as the first known device and a predecessor to RFID technology. A more similar technology, the IFF transponder, was invented by the British in 1939, and was routinely used by the allies in World War II to identify airplanes as friend or foe [1]. In 1950, RFID was experimented on with all the explorations that went on for more than a decade. The 1990s saw commercial RFID applications enter mainstream and standards being prescribed. Today, it's being widely used and the RFID explosion still continues [6].

1.2 How it works

In a typical system, tags are attached to objects. Each tag has a certain amount of internal memory (EEPROM) in which it stores information about the object, such as its unique ID (serial) number. When these tags pass through a field generated by a reader, they transmit this information back to the reader, thereby identifying the object. The communication process between the reader and tag is managed and controlled by one of several protocols, such as the ISO 15693 and ISO 18000. Basically what happens is that when the reader is switched on, it starts emitting a signal at the selected frequency band. Any corresponding tag in the vicinity of the reader will detect the signal and use the energy from it to wake up and supply operating power to its internal circuits. Once the Tag has decoded the signal as valid, it replies to the reader, and indicates its presence by modulating (affecting) the reader field [5].

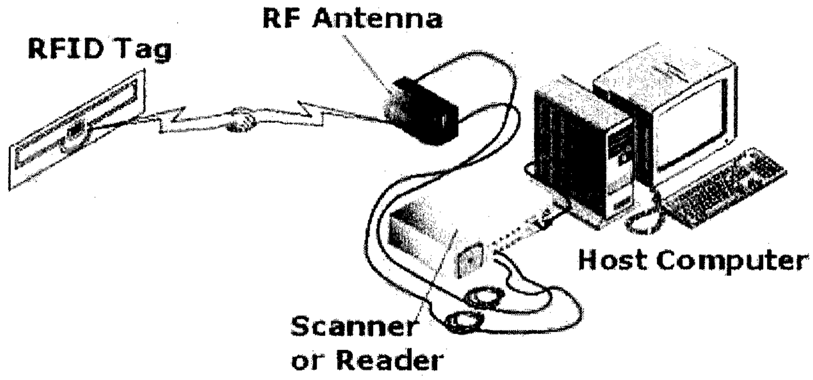
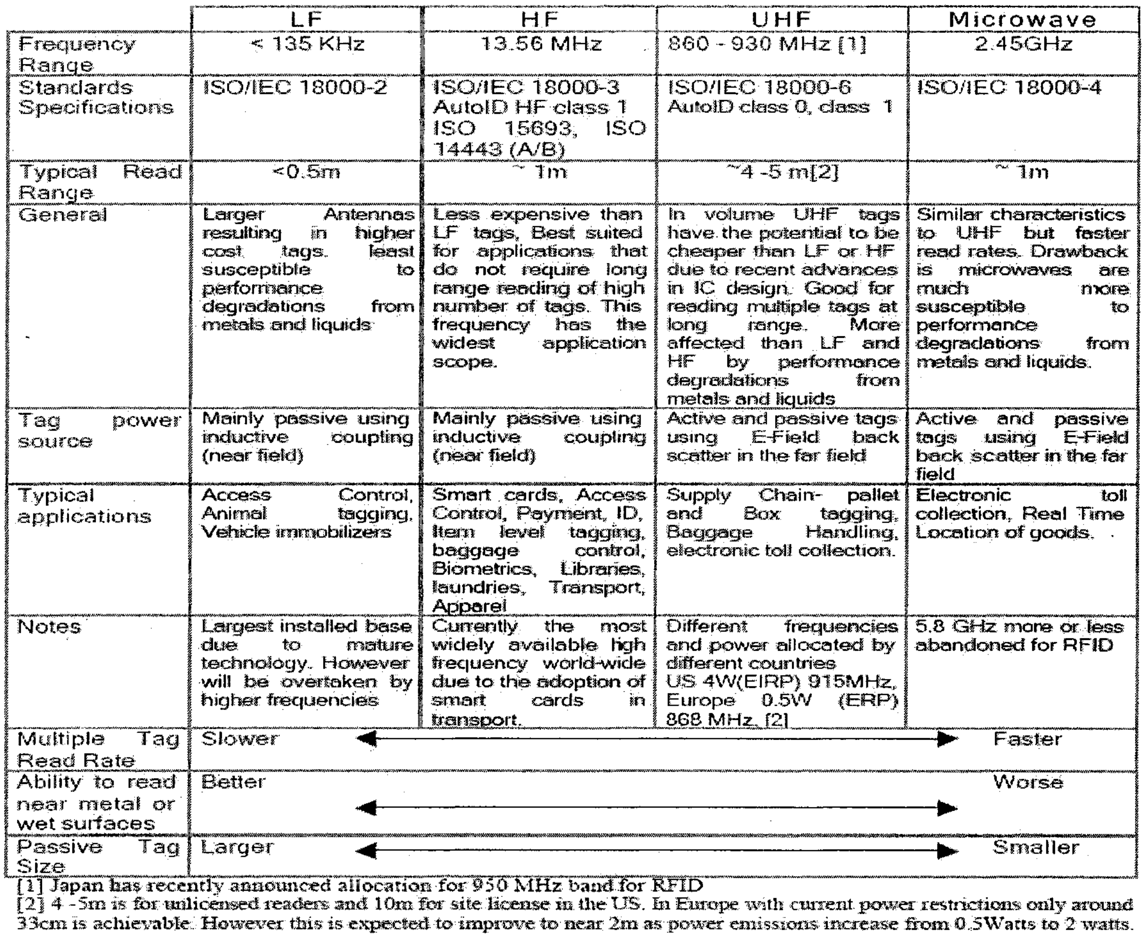


Figure 3 RFID System [5]

If more than one valid tag is present then all those tags will respond at the same time, which causes a signal collision at the reader end and is seen as an indication of multiple tags. The reader manages this problem by using an anti-collision algorithm designed to allow tags to be sorted and individually selected. There are many different types of algorithms (Binary Tree, Aloha, etc.) which are defined as part of the protocol standards. Once the appropriate tag is selected, the reader is then allowed to perform various operations like read or a read/ write operation [5].

The following table describes the various properties of an RFID system for different frequency ranges that it can operate in [5]:



	LF	HF	UHF	Microwave
Frequency Range	< 135 KHz	13.56 MHz	860 - 930 MHz [1]	2.45GHz
Standards Specifications	ISO/IEC 18000-2	ISO/IEC 18000-3 AutoID HF class 1 ISO 15693, ISO 14443 (A/B)	ISO/IEC 18000-6 AutoID class 0, class 1	ISO/IEC 18000-4
Typical Read Range	<0.5m	~ 1m	~4 -5 m[2]	~ 1m
General	Larger Antennas resulting in higher cost tags. least susceptible to performance degradations from metals and liquids	Less expensive than LF tags, Best suited for applications that do not require long range reading of high number of tags. This frequency has the widest application scope.	In volume UHF tags have the potential to be cheaper than LF or HF due to recent advances in IC design. Good for reading multiple tags at long range. More affected than LF and HF by performance degradations from metals and liquids	Similar characteristics to UHF but faster read rates. Drawback is microwaves are much more susceptible to performance degradations from metals and liquids.
Tag power source	Mainly passive using inductive coupling (near field)	Mainly passive using inductive coupling (near field)	Active and passive tags using E-Field back scatter in the far field	Active and passive tags using E-Field back scatter in the far field
Typical applications	Access Control, Animal tagging, Vehicle immobilizers	Smart cards, Access Control, Payment, ID, Item level tagging, baggage control, Biometrics, Libraries, laundries, Transport, Apparel	Supply Chain- pallet and Box tagging, Baggage Handling, electronic toll collection.	Electronic toll collection, Real Time Location of goods.
Notes	Largest installed base due to mature technology. However will be overtaken by higher frequencies	Currently the most widely available high frequency world-wide due to the adoption of smart cards in transport.	Different frequencies and power allocated by different countries US 4W(EIRP) 915MHz, Europe 0.5W (ERP) 868 MHz. [2]	5.8 GHz more or less abandoned for RFID
Multiple Tag Read Rate	Slower	←	→	Faster
Ability to read near metal or wet surfaces	Better	←	→	Worse
Passive Tag Size	Larger	←	→	Smaller

[1] Japan has recently announced allocation for 950 MHz band for RFID

[2] 4 -5m is for unlicensed readers and 10m for site license in the US. In Europe with current power restrictions only around 33cm is achievable. However this is expected to improve to near 2m as power emissions increase from 0.5Watts to 2 watts.

Figure 4 RFID system properties for different operating frequencies

1.3 Security Issues in RFID systems

Although RFID systems may emerge as the most pervasive computing technologies, there are still some security issues related to it that need to be addressed. As the applications of RFID keep growing in complexity and see wider use, the need for updated security measures that are rugged but still suit the simple working standards of RFID is also abounding.

One of the major issues related to RFID security is privacy. Tags reveal sensitive data when queried by readers and they do it indiscriminately [7]. Any perpetrator who wants to access classified data can do so, by querying the tag, the way it is supposed to be. The other most important security issue in using RFID is tracking. The tags usually allow a third party to easily establish an association between a given tag and its holder or owner, by revealing the location to any reader that queries it [7]. Apart from these issues, an RFID system is always vulnerable to attacks like Spoofing, Denial of Service, Eavesdropping, Counterfeiting and Physical attacks.

There has been extensive research carried out in the field of RFID security, since RFID technology's invasion into every other field is beckoning, and a standardized security protocol for RFID systems is a possibility in the near future. This thesis aims at providing an efficient solution to the security issues of an RFID system, that have been addressed later on.

CHAPTER 2

RFID SECURITY LITERATURE REVIEW

Since its invention in the 1940s, RFID has been an obvious target for abuse. Wireless identification is a powerful capability, and RFID reveals both a physical object's nature and location. Anyone can easily gain unauthorized access to RFID data because they don't need a line of sight to gather it. For example, in the original RFID-based application—Identification Friend or Foe (IFF) systems— security breaches resulted in Allied planes being shot down [8].

Despite concerns over security issues, because of its simplicity, low-cost and uniquely identifying capability, RFID systems are scaled with additional implementation to make it more suitable to use for a particular application and are achieving wide-scale deployment. There has always been a call for a consensus on a secure RFID communication protocol that would standardize all the security measures of a low-cost RFID system. However, with each passing day, the applications of RFID are tremendously growing only to put forth the question of the efficiency of existing security protocols when implemented in the new and more sophisticated applications. This stresses the need for constant extensive research to find an appropriate solution, to secure the RFID applications.

This chapter discusses the security issues in the predominant applications that use RFID, and also the research and concepts that try to resolve these issues. As mentioned, the earliest application of RFID was in World War II when it was used for the Identification Friend or Foe systems, which were compromised.

2.1 RFID Application Study

RFID has been quite extensively used in Inventory management, Livestock tracking, Automobile tracking, Airport Security, IDs and badges, etc.

The most common application of RFID in Inventory managements would be its replacement of barcodes to handle product and customer information and stocking and warehousing purposes. Wal-mart was the first retailer to start using RFID in place of barcode, in 2006. But early efforts to use RFID technology for inventory management have raised concerns that tags will he used to develop consumer profiles. Civil-liberties advocates similarly fear that RFID technology may he used by the government to monitor people [9].

RFID is used in by a gamut of applications for tracking purposes. They are used in livestock farming where each animal has its own tag so that it can be tracked individually. [10] talks about how the US Department of Agriculture is planning to use RFID to uniquely identify every cow in the US, to step up efforts to track animal diseases more efficiently. It is used the same way in tracking cars in parking lots and in rental car lots. [11, 12, 13, 14] explain how RFID tags can be used to not manage access control in parking lots, but also to track stolen cars and high end parts of a car, not to mention the possibility of adding a tag to the license plate to facilitate the police or anyone concerned, to retrieve any relevant data about the vehicle.

The use of RFID tags can aid the detection and identification of possible threats in a diverse range of applications from passenger check-in at airports to the detection of intruders. [15] investigates an automatic tracking system to improve airport efficiency and security by means of a cellular network of passive RFID devices, which may be issued along within the boarding passes or as security badges.

2.2 RFID Security Issues

It is obvious that in the applications mentioned above and in the other applications of RFID, privacy and data integrity are highly essential, as they deal with sensitive and classified data that could be worth a fortune to an adversary. Hence, there have been several security measure and protocols that have been proposed over the years, to protect an RFID application from any of the possible attacks.

RFID security threats have been broadly classified into five categories and have been describes, as follows, in [8]:

2.2.1 Sniffing

RFID tags are indiscriminate— they're designed to be readable by any compliant reader. Unfortunately, this lets unauthorized readers scan tagged items unbeknownst to the bearer, often from great distances. People can also collect RFID data by eavesdropping on the wireless RFID channel. Unrestricted access to tag data can have serious implications.

2.2.2 Tracking

RFID technology facilitates clandestine monitoring of individuals' whereabouts and actions. RFID readers placed in strategic locations (such as doorways) can record

RFID tags' unique responses, which can then be persistently associated with a person's identity. RFID tags without unique identifiers can also facilitate tracking by forming constellations, recurring groups of tags that are associated with an individual. RFID technology also enables monitoring entire groups of people.

2.2.3 Spoofing

Attackers can mimic authentic RFID tags by writing appropriately formatted data on blank RFID tags. For example, thieves could retag items in a supermarket identifying them as similar, but cheaper, products. Tag cloning is another kind of spoofing attack, which produces unauthorized copies of legitimate RFID tags.

2.2.4 Replay attacks

Relay devices can intercept and retransmit RFID queries, which offenders can use to abuse various RFID applications. England's new RFID-enabled license plates, *e-Plates*, are one example of a modern RFID system that's susceptible to attack by a relay device. The active e-Plate tags contain an encrypted ID code that is stored in the UK Ministry of Transport's vehicle database. An attacker can record the encrypted identifier when another car's license plate is scanned and replay it later.

2.2.5 Denial of service

RFID systems only work when RFID tags and back-end databases are available. Thieves can exploit this to steal RFID-tagged items by removing tags from the items completely or by putting them in a foil lined booster bag (Faraday cage) that blocks RFID readers' query signals and temporarily deactivates the items. Another attack takes the opposite approach—flood an RFID system with more data than it can handle. Anti-RFID activists could remove RFID tags and plant them on other items, causing RFID systems to record useless data, discrediting and devaluing RFID technology.

2.3 Existing Security Mechanisms and Protocols

Low cost Radio Frequency Identification (RFID) tags are highly resource limited, and cannot support strong cryptography. There has been constant research on finding a light-weight protocol that can handle these issues as efficiently as possible. The most common approaches involve a Challenge-Response System[16] with or without encryption/hash technique.

2.3.1 Challenge-Response System

[16] suggests a feasible security mechanism to enhance RFID tag's security, but conforming to the low cost budget. It uses mutual authentication between reader and RFID tag that is based upon the principle of three-pass mutual authentication in accordance with ISO 9798-2, in which both entities in the communication verify the other participant's knowledge of a secret cryptographic key. In the procedure of mutual authentication, all the RFID tags and readers that belong to the same application share the same secret cryptographic key K_{AB}. When a RFID tag first enters the interrogation zone of a reader both parties will corroborate their identities to each other using three-pass mutual authentication. The illustration of the mutual authentication procedure between RFID tag and reader is shown in Figure 1.

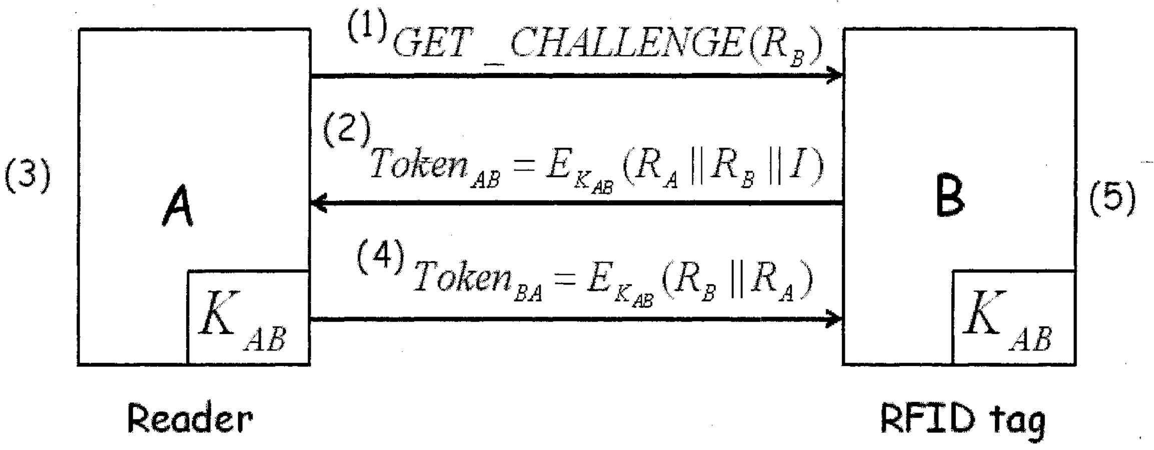

(1) $GET\_CHALLENGE(R_B)$
(2) $Token_{AB} = E_{K_{AB}}(R_A \| R_B \| I)$
(3)
A
(4) $Token_{BA} = E_{K_{AB}}(R_B \| R_A)$
K_{AB}
B
(5)
K_{AB}
Reader
RFID tag

Figure 5 Illustration of mutual three-pass authentication procedure between RFID tag and reader

Another research paper [17], suggests a similar type of Challenge-Response System, in which the reader chooses a challenge, x, which is a random number and transmits it to the tag. The tag computes x = e_k(y) and transmits the value y to the reader (here e is the encryption rule that is publicly known and K is a secret key known only to the reader and the particular tag). The reader then computes y ' = e $_K$(x). Then the reader verifies that y' = y.

2.3.2 Encryption

Encrypting the data that is transmitted is a simple idea of ensuring data privacy and providing data integrity. But incorporating even a simple encryption algorithm in an RFID system requires a hardware implementation that's resource intrinsic. TEA is an encryption algorithm designed for simplicity and ease of implementation and is recommended for RFID applications by [17]. The encryption algorithm is based on the Feistel cipher [18] and a large number of iterations to gain security without compromising simplicity. A description of the algorithm is provided in [19]. A hardware implementation of the algorithm is stated to have the same complexity as DES [19].

Implementing any of the available complex encryption algorithms is going to be an overhead on the system. Typically, a tag can only store hundreds of bits, roughly have between 5000 and 10000 logic gates, and a maximum communication range of a few meters. Within this gate counting, only between 250 and 3000 gates can be devoted to security functions. It is interesting to recall that for a standard implementation of the Advanced Encryption Standard (AES) or any other relatively complex encryption algorithm, between 20000 and 30000 gates are needed, not to mention the power restrictions [7].

2.3.3 Hash Function

Hash functions have been proposed as a replacement to encryption. One of the simple proposals described a Hash Lock mechanism [20]. According to the Hash Lock mechanism, the tag is locked during the manufacturing process and it is given a value (or meta-ID) y, and it is only unlocked from the individual by presentation of a key or PIN value x such that $y = h(x)$ for a standard one-way hash function h. [21] explains Randomized hash-lock and Hash Chain mechanisms which are also common. Other complex hash functions like SHA-1 have also been considered, but as mentioned they demand a tight high-end hardware configuration.

2.3.4 Simple Approaches

Another research paper by Zongwei Luo, Terry Chan and Jenny S. Li [18] discusses different simpler approaches to a Lightweight Mutual Authentication Protocol. The proposed approaches in [20] are briefed below:

Extra Device Added Approach – An additional device operated along with the tags and readers, to prevent attacks using Faraday Cage, Blocker Tag or Active Jamming.

Radio Frequency Modification Approach – Readers and special tags can employ random frequencies for transmission so that unauthorized users may not access the data. This also helps in preventing collision when multiple tags are being used.

Other Approach – These include methods like killing the tag, which makes the tag truly dead, and can never be reactivated, to prevent tracking and by using a Hash Lock.

A lot of security protocols that use variations of the above mentioned security mechanisms, have been proposed but with RFID growing tremendously in application, these protocols don't prove to be completely fool-proof or resource-friendly.

CHAPTER 3

SECURITY PROTOCOL DEFINITION AND IMPLEMENTATION

The working of a security protocol has to be precisely defined, in order to analyze its performance, especially when it is used with a relatively less complex and a low-cost application, like an RFID system. This Chapter describes how the proposed security protocol is defined and implemented, and details its working. This is a proposed theoretical solution to the security issues involved in an RFID system, the efficiency of which was evaluated and is elucidated on, in the next Chapter.

The proposed Security protocol guarantees data privacy and authentication between the Tag and the reader, to the maximum extent possible. It uses simple information exchange between the Tag and the Reader, to authenticate each other. The existing security protocols use simple encryption or an efficient hashing technique to authenticate the Tag and the Reader. Though the proposed protocol also uses encryption, it makes the RFID system foolproof in almost every way that the other protocols miss out on. This does increase the cost and load on the tags, but will prove to be highly secure for a system that needs it.

3.1 Why does RFID Communication need Security?

With a technology as ubiquitous as radio-frequency identification, the applications using it just keep growing by the day. Some of the current applications that use the RFID technology are:

- Inventory Control and Management
- Container/ Pallet/ Shipment Tracking
- ID Badges and Access Control
- Automatic Toll Control
- Airport Security
- Equipment/ Personnel tracking in Hospitals
- Parking Lots/ Rental Car Lots

The data exchanged in these applications, and the various others, that use RFID, are quite sensitive and need to be protected, in order to avoid identity theft, data loss, data/ product duplication, etc. Compromised RFID systems can be used to the will of the hacker, to reproduce/ modify/ misuse the system. The Interceptor can reproduce tags, by monitoring the Tag ID exchange and can, with calculated attacks, fool the reader and the database, in giving any information he needs. A compromised RFID system can hence, cause a huge monetary, commercial or personal loss. With a lot of military applications using RFID these days, security of RFID data exchange between the Tag and the reader, is of utmost importance. Security breaches can happen at the RFID tag, network, or data level. Part of the problem with adopting existing standards, at least at one level, may be the extreme low cost and extremely light functionality on the tags. All of the good security tools developed over the last 20 years won't fit into the hardware that's available on most of these RFID tags. Even encryption chews up most of the processing power of the Tag. Therefore, a security protocol that uses minimal resources and yet is efficient in authenticating the Tag and the Reader, for each other, is absolutely necessary.

3.2 Custom RFID System Description

3.2.1 RFID Reader

The Reader required for the proposed protocol, needs to be highly flexible. The protocol requires a reader that can read, as well as write, to the tag. The frequency, which the reader has to read from, is decided based on the application. A high-frequency reader that can read and write is appropriate in testing the proposed security protocol. The reader needs to support Frequency Hopping – Code Division Multiple Access (FH – CDMA) and should have circuitry for modulation and demodulation, at low, high, ultra-high or microwave frequencies, whichever is prescribed for the application. The reader should also be capable of transmitting to more than one channel. This is an overhead on the power consumed, but the system can still be optimized by using an appropriate multi-channel RF transmitter chip to get the desired results.

Texas Instruments Inc.'s product, Multi-standard High frequency RFID Reader/Writer IC, TRF7960 is a suggested reader, for any application that uses the security protocol based on random key generation. The specifications of TRF7960 include Completely Integrated Protocol handling (OSI Model Layer 3 and below), Separate internal high-PSRR power supplies, Dual receiver input with AM and PM demodulation, Reader-to-reader anti-collision, wide operating voltage range 2.7V to 5.5V and Ultra-low power modes[22]. The TRF7690, however, is not a multi-channel RF transmitter, but this can be fixed by using such a transmitter chip, like Texas Instruments Inc.'s Highly Integrated Multi-channel RF Transmitter for Low-power wireless applications, CC1150[23]. The TRF7690 describes the design of a basic RFID reader that would suit the proposed protocol. The internal circuitry of TRF7690 is shown below [24]:

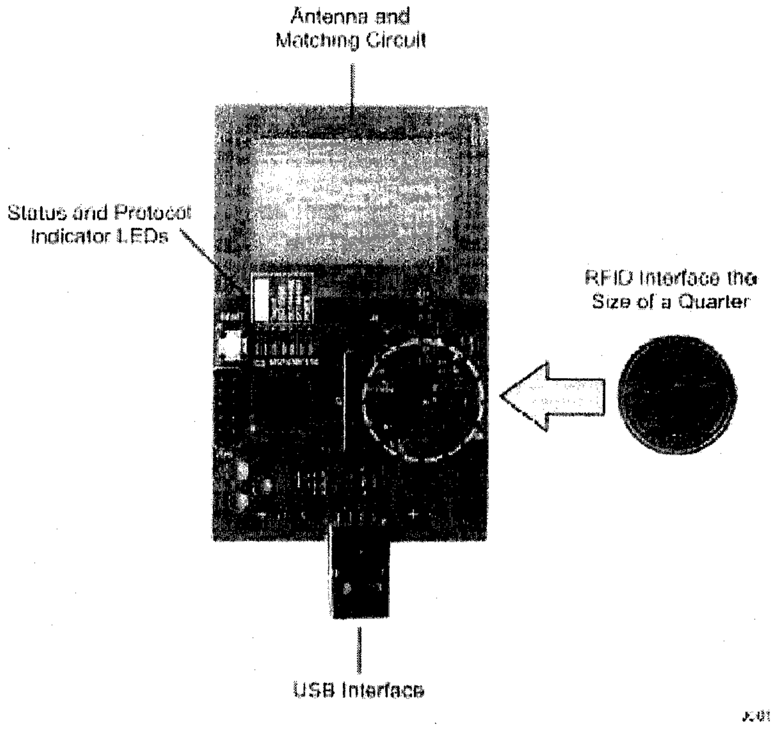

Antenna and Matching Circuit
Status and Protocol Indicator LEDs
RFID Interface the Size of a Quarter
USB Interface

Figure 6 Basic RFID reader (TRF7690) Internal Hardware Design

3.2.2 RFID Tag

The RFID Tag required for the security protocol based on random key generation, is customized to suit the protocol and is not available in the market. Hence, this protocol may not work with any of the current RFID tags that are available. However, the only customization would involve the Read-only registers and the Re-writable memory. The basic configuration and the chip configuration could resemble Texas Instruments Inc.'s

Single-Chip, Low-power, Low-cost CMOS FSK/GFSK/ASK/00K RF Transmitter for Narrowband and multi-band Apps, CC1070 [25], which are being marketed for RFID tags.

The RFID Tag needs to support multi-channel transmission and frequency modulation. The proposed measure of sophistication can never be imposed on a passive tag that does not have a power source of its own. Hence, this protocol can only be used with active tags and high-end applications that would justify the cost of using these tags, which eventually will prove to be cheaper than any other technology used in its place. The internal circuit diagram of the basic RFID Tag, CC1070, that can be modified, so as to be used with this protocol, is shown below [26]:

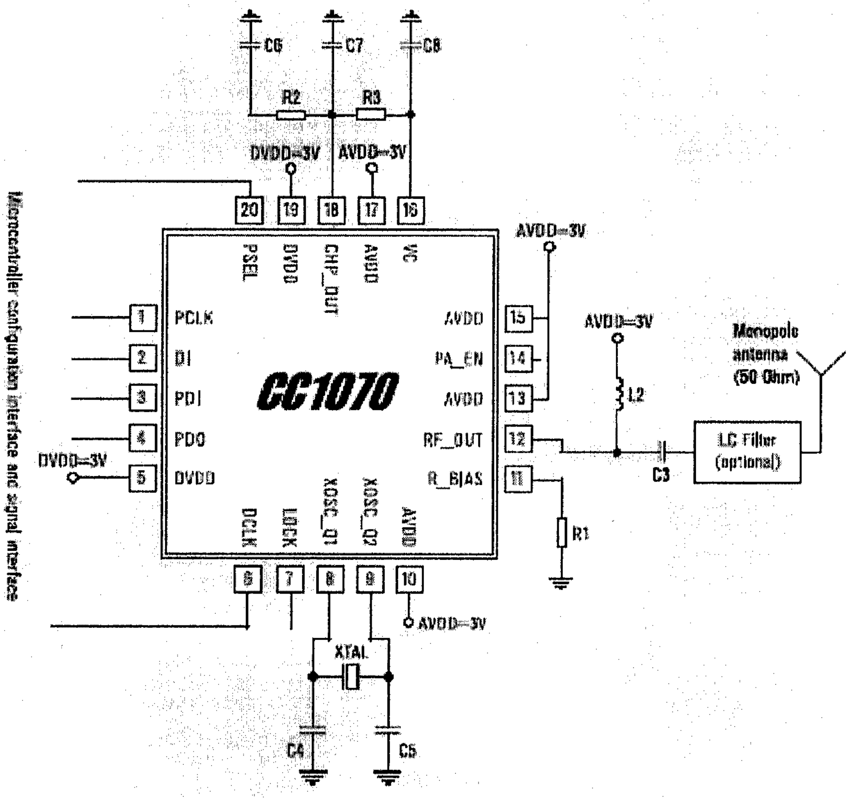


Figure 7 Basic RFID Tag internal Circuit diagram

The proposed protocol requires the Tag to have three basic registers that hold the data used to identify the tag uniquely, apart from the other registers that are needed for intermediate computation, like while encrypting and decrypting data. It is imperative to describe these memory units because they define the working of the proposed security protocol in a big way.

3.2.3 RFID Tag Memory Description

The RFID Tag has a 128-bit ROM that stores the 128 bit Tag ID. Since the Tag ID associated with a particular Tag cannot be changed, it is written to a Read-Only Memory even when it is manufactured. This value, which uniquely identifies the tag, can only be read and can never be modified.

The tag also has two 64 Bit Re-writable memories, which are used to store important data that can be read and modified. One of the 64 Bit Re-writable memories (RAM) stores the 64 Bit Encryption Key, which is a public key shared between the tag and the reader, that is, the backend application, and can be over-written. The other 64 Bit Re-writable memory is used to store a 64 Bit Authentication Code, which is used to authenticate the reader and can be over-written. The following is a schematic representation of the memory structures, needed in the RFID tag, for the proposed security protocol:

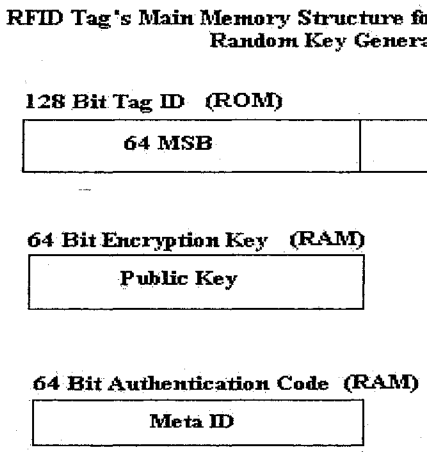

RFID Tag's Main Memory Structure for Security protocol based on Random Key Generation
128 Bit Tag ID (ROM)
64 MSB
64 LSB
64 Bit Encryption Key (RAM)
Public Key
64 Bit Authentication Code (RAM)
Meta ID

Figure 8 Memory Structure Description of the required RFID tag

3.2.4 RFID System's Backend Application

The Backend Application holds the data about all the legitimate RFID tags that are a part of the RFID application. The Backend application could be any application, running on a (remote) server, Database Management System or just a Data dump. The backend application needs to be connected to the reader at all times and the connectivity and integrity of this link, plays a crucial role in the performance of the RFID application. As mentioned, the backend application holds the data about the valid tags that are a part of the application. In a system, with the proposed security protocol being implemented, the backend application stores information which includes the 128-bit Tag ID, 64-bit Encryption Key and 64-bit Authentication Code of every legitimate RFID tag. The value for the tag ID cannot be modified but, the values of the Encryption key and the Authentication Code, associated with a particular Tag ID can be modified. The backend application also takes care of any data processing, like encryption/ decryption, generating a new random encryption key, generating a new random authentication code, etc., thus abstracting the reader from any manipulation with the data.

The reader can communicate with the backend application, using any of the basic network protocols like TCP/IP, SSL, SOAP, etc. or could use a middleware that manages the communication in a secure way. An RFID Middleware is software that manages the flow of data between tag readers and backend enterprise applications and is responsible for the quality, and therefore usability, of the information [27]. A high level of security can be incorporated in the middleware that has the capability of routing data to more than one standardized server. The backend application should be able to provide valid data, in the appropriate form, to the reader, at anytime when it is requested for.

3.3 RFID Security Implementation based on Protocol

The EPC and ISO have standardized the first two layers of the communication protocol stack between the readers and the tags. These two layers include the local wireless communication that occurs between a reader and the tags within its read field. The first layer standard is the physical, which describes the specific radio frequencies and whether tags and readers are communicating in half or full duplex mode. The second layer, referred to as the data link layer, has been standardized based on a slotted Aloha scheme [28].

The physical (PHY) layer standard functionality defines how the reader communicates with the tag and how the tag communicates with the reader using passive communication. The physical layer features include the signaling, modulation and encoding of symbols. The physical layer must be amenable to long range passive communications. The data link (DATA) layer forms the base logic layer of the class structure, which provides a framework to classify RFID tags according to their primary functional characteristics. The data link layer standard defines how communications are logically grouped, or packetized, the commands that the reader may issue to the tag, the tag's response to any commands, when and how a tag may initiate communication or functionality without being commanded by the reader, and the resource discovery, i.e., tag identification, algorithm [2].

The proposed Security protocol makes sure that its implementation secures RFID communication, in both the layers of the RFID communication protocol suite. The Physical Layer, which deals with the actual wireless data communication, uses Frequency Hopping and the Data Link Layer uses Data Encryption, according to the proposed protocol. The following figure describes the security implementation in the two layers of the RFID communication protocols suite:

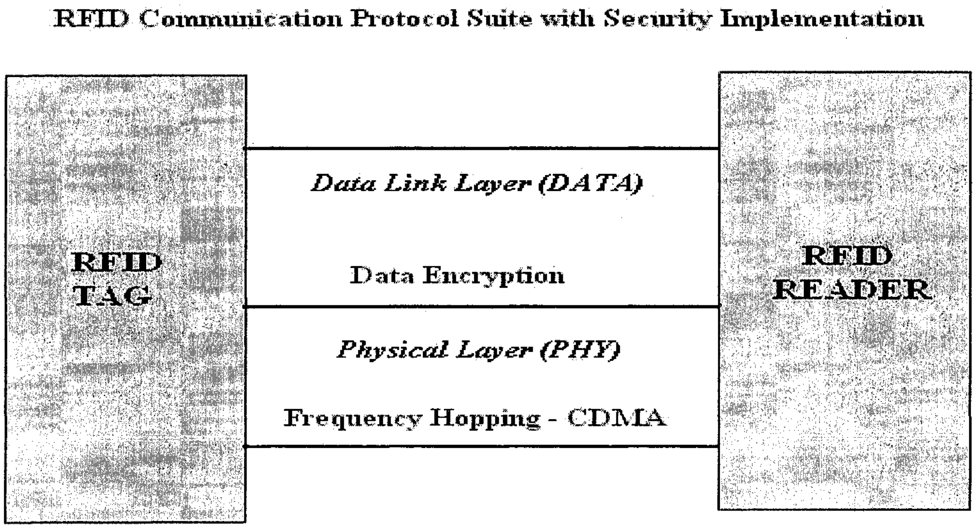


Figure 9 Security in the different layers of RFID communication protocol suite

3.3.1 Security Implementation in Physical Layer

The frequency range, in which the RFID application operates, is divided into bands or frequency channels. This type of communication between a tag and the reader, using Frequency Modulation, is pretty common [28] and is usually used when there are multiple tags, to avoid collision while reading [20]. The RFID reader and the tag use one particular channel to transmit or receive the legitimate data and send dummy data through the other channels. The dummy data could be anything that is in the same format of the original data but does not affect the RFID system's working in any way. The frequency, at which valid data is transmitted, keeps changing for every transmission, by both the tag and the reader. This frequency switching is called Frequency Hopping and this type of access method is called Frequency Hopping – Code Division Multiple Access [29]. [30, 31] define how Frequency Hopping is already being used in RFID systems. It is for this reason that the tag and the reader are chosen such that they are able to transmit to multiple channels.

The sequence of frequency hops is known both to the reader (Backend application) and the tag, so that legitimate data can be read from the appropriate frequency. This concept is similar to what is used by Bluetooth Technology [32], except for the fact that Bluetooth uses Frequency Hopping to ensure proper communication, as it permits connectivity to multiple Bluetooth enabled devices, and the proposed RFID system uses Frequency Hopping to make Eavesdropping or any other attack on the Physical layer, as non-feasible as possible.

Frequency Hopping can be used with Low-frequencies, High-frequencies, Ultra-high Frequencies and Microwave frequencies. Therefore the RFID application can choose to work in any frequency range, based on its requirements, and yet have the proposed security implementation in its Physical Layer.

3.3.2 Security Implementation in Data Link Layer

The Data Link Layer decides in what format and how exactly data is exchanged between the tag and the reader. Security implementation in this layer is imperative and the proposed protocol, like a lot of existing security protocols for RFID systems, uses data encryption. Data encryption guarantees a high level of data integrity and makes the data comprehendible only by the RFID application using it.

Since encrypting and decrypting of data, places a lot of complications on the system, like power consumption, memory resources, processing time, etc., the proposed protocol cannot be implemented in a RFID system that uses passive tags, tags that don't have a power source of their own. Hence, this protocol can be used only with a more complex system that can support its implementation. The type of encryption is chosen after careful consideration of a lot of factors that decide the performance measures of the RFID system.

A relatively less sophisticated system or a system that has its physical layer completely secure from any sort of interception, can afford a simple encryption algorithm and if the application requires a high level of data privacy and integrity, a rugged encryption algorithm could be used. The type of encryption is always a trade-off and should be chosen prudently, so as to make the system as efficient as possible, without impacting the performance of the RFID system.

The security protocol based on Random key generation, as proposed in this thesis, was tested with three different Encryption methods, namely, Boolean encryption, Tiny

Encryption Algorithm and Advanced Encryption Standards. These three encryption algorithms were chosen because they differ in the complexity of implementation, their efficiency and immunity to attacks, the Boolean encryption being the simplest algorithm and the AES being the most efficient of the three. These three encryption methods are described in detail as follows:

3.3.2.1 Boolean Encryption

Boolean encryption, as used in this implementation of the security protocol based on Random key generation, is a simple Binary XOR operation. The encryption is a simple process of computing the value of the binary value of plaintext that needs to be encrypted, XORed with the binary value of a public key. [21, 33, 34] proposes a security model that introduces a challenge-response mechanism which uses no cryptographic primitices other than simple XORs. The binary XOR operation (also known as the binary XOR function) will always produce a 1 output if either of its inputs is 1 and will produce a 0 output if both of its inputs are 0 or 1. To decrypt the encrypted value, using Binary XOR operation, the binary value of the encrypted value is XORed with the binary value of the same shared public key that was used in encrypting the plaintext. This type of encryption requires no more power than a system that uses no encryption, but can be easily cracked. This serves to just abstract the actual data instead of making it completely secure.

3.3.2.2 Tiny Encryption Algorithm

The Tiny Encryption Algorithm (TEA) is a block cipher notable for its simplicity of description and implementation (typically a few lines of code). It was designed by David Wheeler and Roger Needham of the Cambridge Computer Laboratory, and first presented at the Fast Software Encryption workshop in 1994 (Wheeler and Needham, 1994).

TEA operates on 64-bit blocks and uses a 128-bit key. It has a Feistel structure with a suggested 64 rounds, typically implemented in pairs termed *cycles*. It has an extremely simple key schedule, mixing all of the key material in exactly the same way for each cycle [35].

Routine, written in the C language, for encoding with key k[0] - k[3]. Data in v[0] and v[1]: [19]

```
void code(long* v, long* k) {
unsigned long y=v[0],z=v[1], sum=0, /* set up */
delta=0x9e3779b9, /* a key schedule constant */
n=32 ;
while (n-->0) { /* basic cycle start */
sum += delta ;
y += ((z<<4)+k[0]) ^ (z+sum) ^ ((z>>5)+k[1]) ;
z += ((y<<4)+k[2]) ^ (y+sum) ^ ((y>>5)+k[3]) ;
} /* end cycle */
v[0]=y ; v[1]=z ; }
```

The following diagram describes how TEA works [33]:

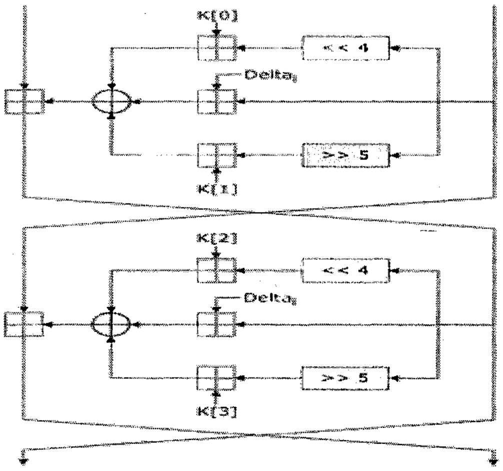


Figure 10 TEA encryption flowchart

This type of algorithm can replace DES in software, and is short enough to write into almost any program on any computer and on one implementation it is three times as fast as a good software implementation of DES which has 16 rounds [19]. The only attack that is possible is a Brute-force attack. The Tiny Encryption Algorithm was chosen because it is as good as DES, but is a lot faster and very easily implemented in a system that's as simple as an RFID system. TEA has a few weaknesses. Most notably, it suffers from equivalent keys - each key is equivalent to three others, and this means that the effective key size is only 126 bits. TEA is also susceptible to a related-key attack which requires 2^{23} chosen plaintexts under a related-key pair, with 2^{32} time complexity [35].

3.3.2.3 Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES), also known as Rijndael, is a block cipher adopted as an encryption standard by the U.S. government. The AES is the successor of DES [36]. The cipher was developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen.

AES has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits. Most of AES calculations are done in a special finite field. AES operates on a 4×4 array of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state)[37]. For encryption, each round of AES (except the last round) consists of four stages:

1. AddRoundKey — each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule.

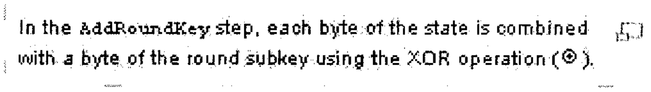
2. SubBytes — a non-linear substitution step where each byte is replaced with another according to a lookup table.

3. ShiftRows — a transposition step where each row of the state is shifted cyclically a certain number of steps.

4. MixColumns — a mixing operation which operates on the columns of the state, combining the four bytes in each column using a linear transformation.

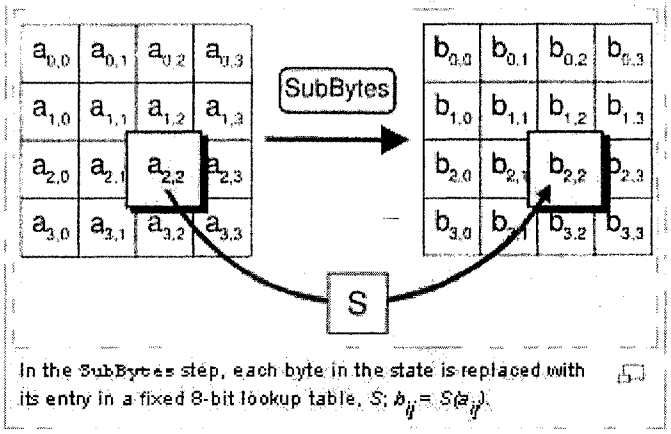
The final round replaces the MixColumns stage with another instance of AddRoundKey [37].

The following is a diagramatic representation of the different stages in AES encryption [37]:



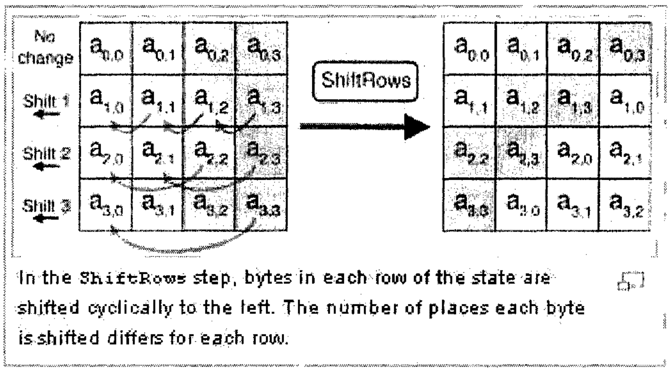
In the AddRoundKey step, each byte of the state is combined with a byte of the round subkey using the XOR operation (⊕).

Figure 11 The AddRoundKey stage of AES



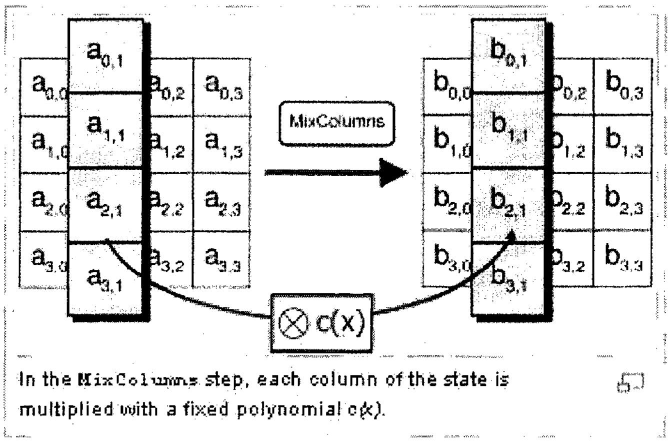
In the SubBytes step, each byte in the state is replaced with its entry in a fixed 8-bit lookup table, S; $b_{ij} = S(a_{ij})$.

Figure 12 The SubBytes stage of AES



In the ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row.

Figure 13 The ShiftRows stage of AES



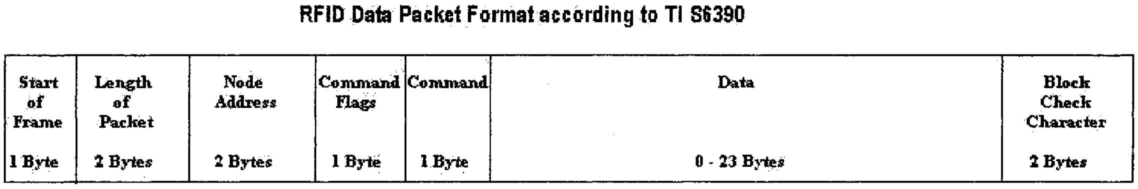
In the MixColumns step, each column of the state is multiplied with a fixed polynomial $c(x)$.

Figure 14 The MixColumns stage of AES

AES is so secure that since June 2003, it is being used by the US Government for encrypting classified data [37]. An attack against 128-bit key AES requiring 'only' 2^{120} operations would be considered a break even though it would be, now, quite unfeasible. However, implementing AES in an RFID system might be an expensive proposal and is recommended only where a high level of security is required.

3.3.3 RFID System Data Packet Format

The Data packet for the proposed RFID system looks similar to standard packet format, of any other normal RFID system. The standard data packet, as specified in Texas Instruments Inc.'s documentation of TI S6390 [38], which is used as an RF transponder, is described below:



RFID Data Packet Format according to TI S6390

Start of Frame	Length of Packet	Node Address	Command Flags	Command	Data	Block Check Character
1 Byte	2 Bytes	2 Bytes	1 Byte	1 Byte	0 - 23 Bytes	2 Bytes

Figure 15 RFID Data Packet Format according to TI S6360

The proposed security protocol makes use of the command field to hold the Operation Bits. These Operation Bits define the type of operation being performed with the data packet. The following table shows the operation corresponding to the Operation Bits:

Operation Bits	**Operation**
00	Clear contents of RAM of Tag
01	Read Request by Reader
10	Response by Tag
11	Write contents to RAM of Tag

Table 1 Operation Bits and the corresponding operations

The operating bits are set to 01 whenever the RFID reader is transmitting data or requesting data to the tag. The operating bits are set to 10 whenever the tag is transmitting data or is sending a response to an RFID reader. When the operating bits are set to 11, the Reader is sending data to overwrite the data in the RAM of the tag, the 64-bit Key, in particular. When the operating bits are set to 00, the reader is basically sending a command to the tag, to erase the content of its RAM.

3.4 Protocol Implementation

The protocol implementation defines how the Tag and the Reader authenticate each other, thus making the RFID communication safe and authentic. The proposed protocol, which is based on random Key generation uses exchange of data between the tag and the reader to make sure the communication is legitimate. The system is setup with the devices mentioned above to facilitate communication according to the proposed protocol.

The step-wise communication between the tag and the reader is described below:

1. 64-Bit Random Number Generation

The Backend Application generates a 64-bit random number, based on any of the distribution functions, and sends this random number to the reader, using any of the protocols adopted by the RFID system. It should be noted that the type of backend application and the communication link between the reader and the backend application do not affect the measure of the security for the communication between the tag and the reader, as according to the proposed protocol. But choosing the backend application with a secure middleware that is appropriate for the system that has the proposed protocol implemented is recommended. The reader that received the 64-bit random number from the backend application sends it to the tag, with the operation bits set to 01, in the format specified for the protocol.

2. 64-Bit Authentication Code Encryption

The tag receives the 64-bit random number and uses it as a key to encrypt its 64-bit Authentication Code. The type of encryption could be anything that suits the performance and scalability of the system. The protocol was tested with all three types of encryptions mentioned above. Once the Authentication Code is encrypted with the 64-bit random number, as the key, the tag then sends the encrypted Authentication Code back to the reader, with the operation bits set to 10. The reader, in turn sends it to the backend application, after validating the node address, which basically tells the reader which tag it received the data from, when it is communicating with more than one tags. If the reader does not receive a response within a specified time interval, the operation is timed out. The proposed protocol was tested with a timeout of 250 nanoseconds.

3. Encrypted Authentication Code Decryption

When the Backend Application receives the encrypted Authentication, it decrypts it, using the random number it generated, as the public key. The backend application then uses the decrypted Authentication Code to look up its database and tries to retrieve the corresponding 64-bit Key and 128-bit Tag ID values. If the values are not found, the communication is terminated, since a wrong value of the Authentication Code de-authenticates the Tag, implicitly. Otherwise, the corresponding 64-bit Key and Tag ID values are retrieved from the database.

4. Tag ID Encryption by Backend Application

Once the backend application retrieves the data, corresponding to the Authentication Code, it encrypts the 128- bit Tag ID, using the same encryption that was used before. The 128-bit Tag ID is encrypted in two 64-bit blocks. The 64 Least Significant Bits are encrypted using the 64-bit Authentication Code of the Tag, as the key and the 64 Most Significant Bits are encrypted using the 64-bit Key of the Tag, as the key. The backend application then sends the encrypted Tag ID to the reader which in turn sends it to the appropriate tag, after setting the operation bits to 01.

5. Tag ID Decryption by Tag

When the Tag receives the encrypted Tag ID, it decrypts it with its 64-bit Authentication Code and 64-bit Key. The Tag decrypts the encrypted 64-bit block corresponding to the 64 Least Significant Bits of the Tag ID, using the Authentication Code as the key and the other encrypted 64-bit block, corresponding to the 64 Most Significant Bits of the Tag ID, with the 64-bit Key of the Tag. The tag then puts the decrypted 128-bit tag ID together and compares it with its own tag ID, in the ROM. If the tag IDs match, then the tag has authenticated the reader. If the tag IDs do not match, then the tag drops the communication with that particular reader. The tag checks for an operation timeout again, as described before.

6. Tag ID Encryption by Tag

If the Reader had been authenticated, it needs to authenticate the tag too. The tag, after matching the tag IDs and authenticating the reader, encrypts the tag ID again, in the similar way, encrypting it as two 64-bit blocks. But this time, the tag encrypts the 64 Least Significant Bits of the Tag ID with its 64-bit Key and the 64 Most Significant Bits of the Tag ID with the 64-bit Authentication Code. The tag then sends the new encrypted tag ID to the reader, setting the operation bits to 10.

7. Tag ID Decryption by Reader

The reader receives the encrypted Tag ID from the tag, to decrypt, with the 64-bit Authentication Code and the 64-bit Key of the tag. If the reader did not receive the encrypted tag ID over a specified period of time (250 nanoseconds) the operation times out. When the reader receives the encrypted tag ID, it verifies the tag that sent it, and passes on the encrypted tag ID to the backend application. The backend application decrypts the block corresponding the 64 Least Significant Bits of the tag ID with the 64-bit Key and the block corresponding to the 64 Most Significant Bits of the tag ID with the 64-bit Authentication Code. If the decrypted tag ID matches the one that was retrieved from the database, the tag is a valid one. The tag is, thus, authenticated. If the tag IDs do not match, then the operation fails and the reader breaks its communication with the corresponding tag.

8. Random Key Generation

Once the tag and the reader are authenticated, the backend application makes the system more secure by changing the public keys that were used for encryption. In doing this, the backend application generated two 64-bit random numbers, one for the authentication code and one for the tag's key. The backend application then updates the value of these two for that particular tag ID, in the database and sends these two 64-bit data to the reader which in turn sends it to the tag, after setting the operation bits to 11. The tag waits for the data from the reader and if it does not receive it within the specified time, then it closes the communication. Since the reader and the tag have already been authenticated, the communication is ideally complete. When the tag receives the randomly generated 64-bit Authentic Code and the 64-bit Key, it rewrites its Authentication Code and Key, with the new values and sends an acknowledgement signal back to the reader, which could be the same randomly generated data that the reader sent it.

The final part of the protocol, where random keys are generated for the next round of communication, keeps the system highly secure because even if the keys were compromised, an attacker can never use it to fake the reader or the tag because the keys are periodically changed and are known just to the reader and the tag.

This protocol is similar to the Challenge-response mechanism described in [16, 17] but is more robust. The protocol mentioned in [16, 17] authenticates just the tag and does not let the tag authenticate the reader and is vulnerable to various attacks.

CHAPTER 4

RFID SECURITY PROTOCOL SIMULATOR

In order to observe and analyze the performance measures of the proposed security protocol and to model its functional parameters, a software program was coded, as a part of the research. It is an effort to aid the analysis of various RFID system configurations under the demands of specific RFID applications, that implement the security mechanism describes in this thesis. Although the simulator was running on a high-end computer system with 3.06GHz Pentium processor with 1GB RAM, it was coded to consider the factors of a real-time RFID system that operates based on specified RFID and air-interface standards, which are described by ISO/IEC 14443A/B, ISO/IEC 15693, ISO 18000, EN 300 400 Class 2 (Europe) and FCC CFR47 Part 15 (US).

This chapter describes the functions available in the simulator and flexibility it provides in testing the proposed protocol. The simulator software was built on Java platform with a front end in Java Swing. The minimum requirements for the simulator software to run are a computer system with optimum configuration (300Mz Processor with 128MB RAM) that is running on Windows, UNIX or Linux, Java Runtime Environment installed in the computer preferably with J2SDE 1.4.2.

The simulator software is opened by double-clicking the executable file, that contains the generated class files of the source code.

4. 1 Simulator's Main Window

The following figure shows the main application frame that opens up, as soon as the simulator software is opened:

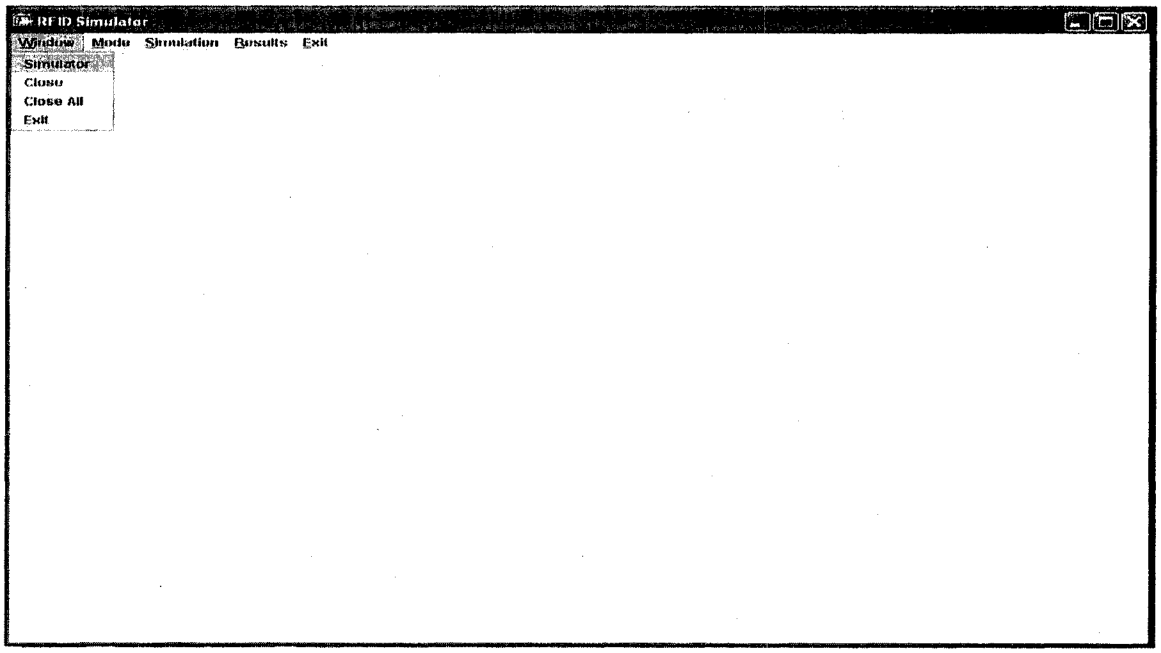


Figure 16 Simulator Software's main window

The Simulator Application looks like any other window application with similar look-and-feel and is easy to navigate. It has a Menu bar that has got Window, Mode, Simulation, Results and Exit menu options.

Window menu has four menu items, namely, Simulator which opens up the simulation run window, Close which closes the active window, Close All which closes all the open windows and Exit which terminates the simulator application.

The Mode menu again has four menu items, out which only one can be chosen, that decides the mode in which the user wants the simulator to run. Real-time mode makes the simulator run like in a real-time situation, Process mode pauses the simulation run after every process in the simulation and waits till the user wants to continue, Delay mode runs the simulation with a delay between every two processes in the run and Compare mode runs the simulator with the same set of simulation data but using all three different encryption methods and draws a comparison.

The Simulation menu has two simulation menu items, Run which opens up the simulator automatically and starts the simulation process according to the current system configuration that is chosen and Options which opens an options window, which helps the user to configure the RFID system and set its parameters.

The Results Menu again has two menu items, View Results opens the View Results window and Export Results exports the simulation run information to a custom text editor where it can be viewed, saved or printed. The Exit menu terminated the simulatór application.

The following figure shows the hierarchical menu structure of the simulator software:

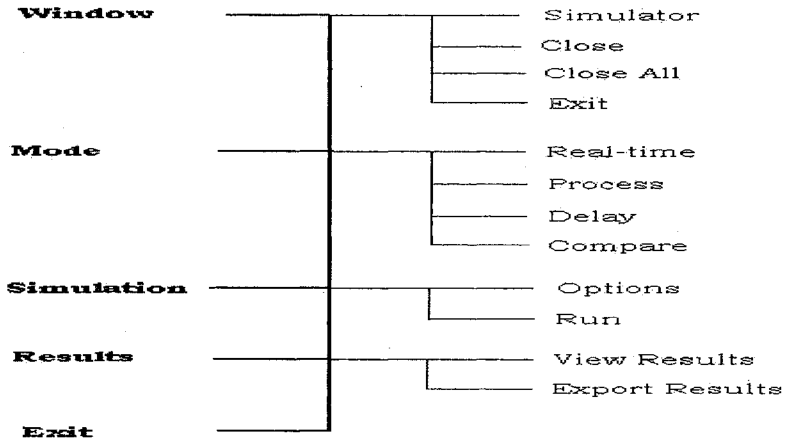


Figure 17 Hierarchical Menu structure

4.2 Simulator Window

The following figure shows the simulator window when it is opened, to run the simulation.

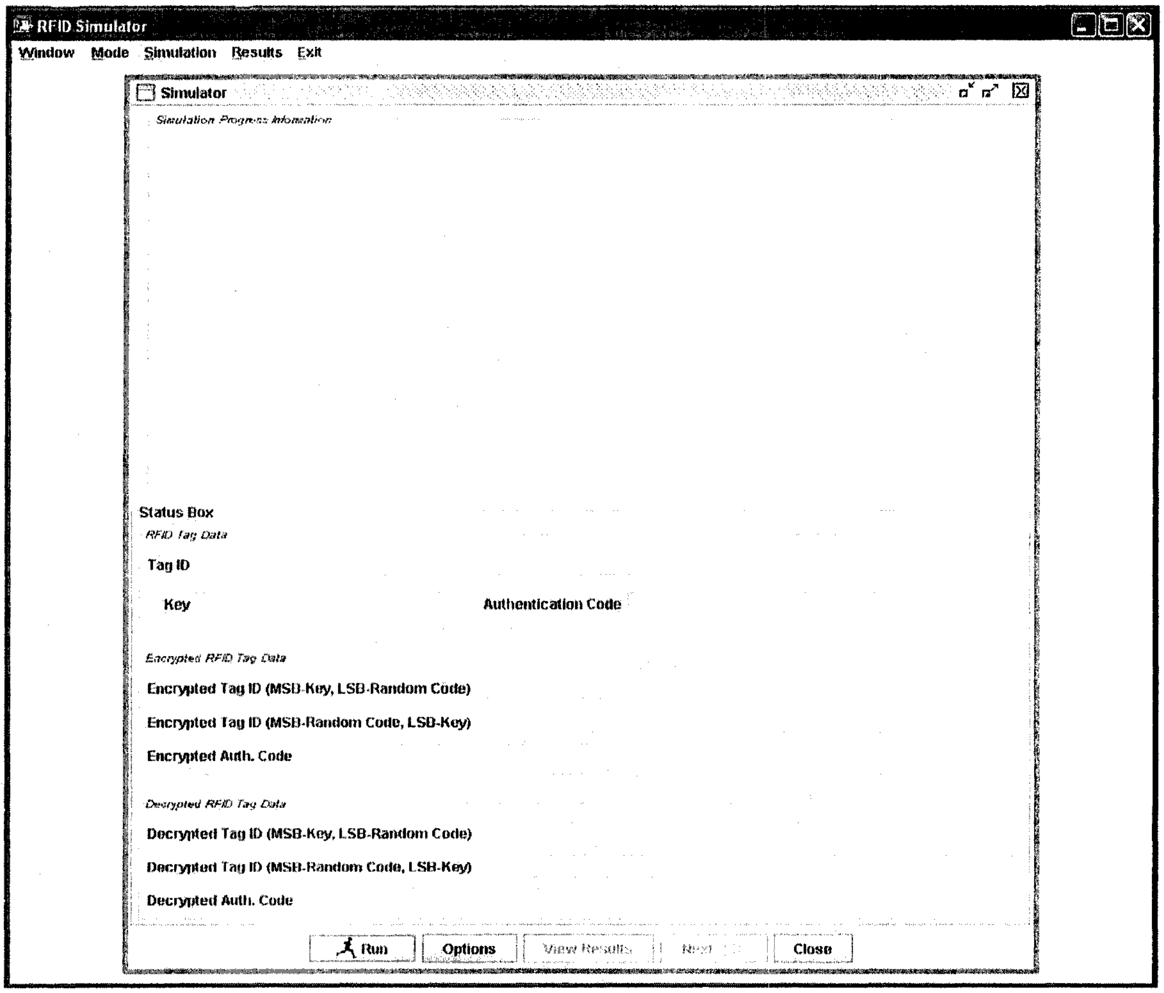
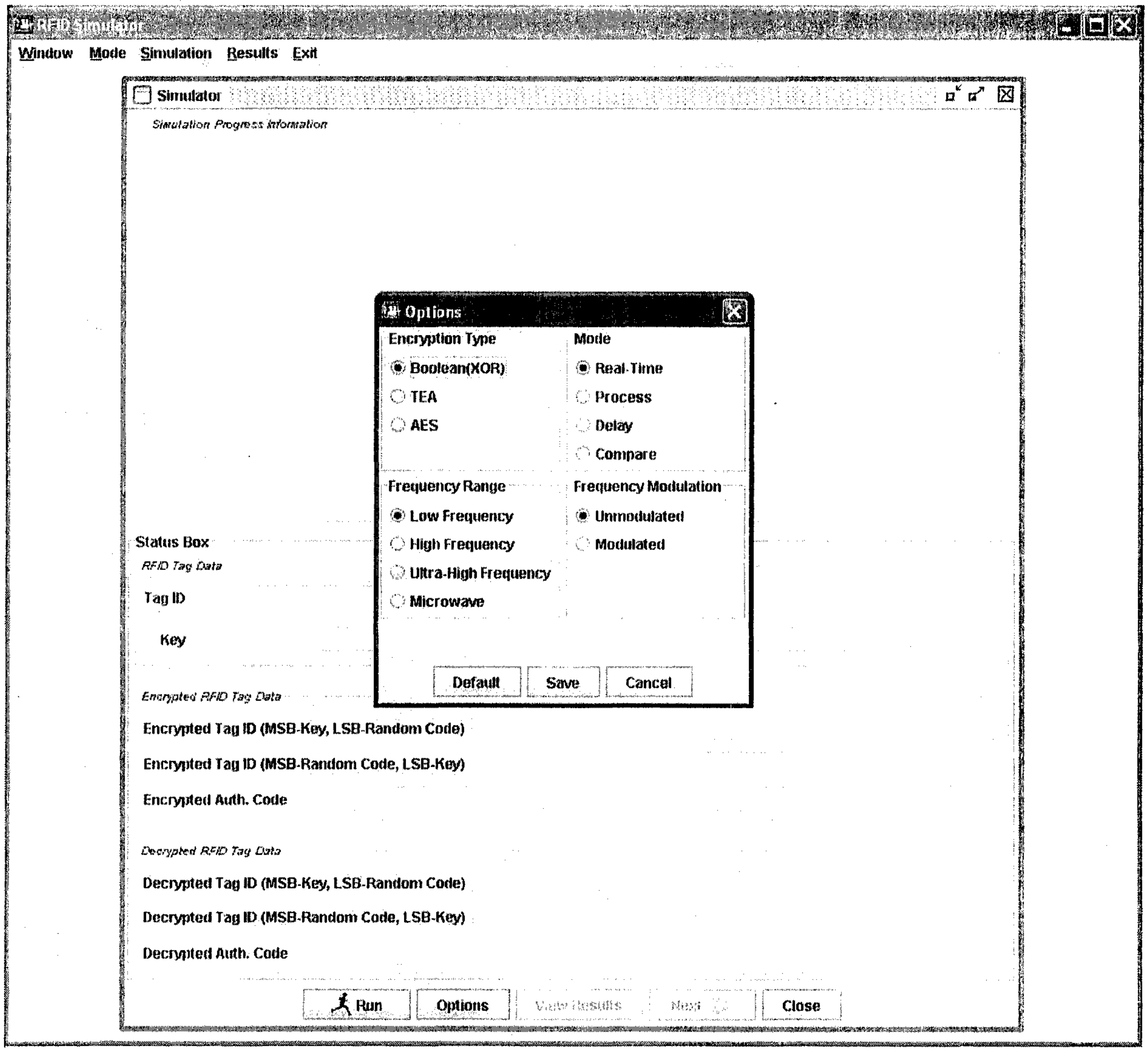


Figure 18 Simulator Window

The simulator window has a text area where the details of the simulation run are updated, even as the simulation is running. It also has fields that display the Tag's data, after the encryption process and also after decryption. The simulation is started by clicking on the Run button at the bottom. The Options and View Results button open the respective windows and Close button closes this window. The Next button is used in Process mode.

4.3 Options Window

The system configuration can be set using the options window, before running the simulation. Clicking the Options button opens the Options window.


RFID Simulator
Window Mode Simulation Results Exit
Simulator
Options
Default Save Cancel
Run Options View Results Next Close

Simulation Progress Information

Encryption Type
● Boolean(XOR)
○ TEA
○ AES

Mode
● Real-Time
○ Process
○ Delay
○ Compare

Frequency Range
● Low Frequency
○ High Frequency
○ Ultra-High Frequency
○ Microwave

Frequency Modulation
● Unmodulated
○ Modulated

Status Box

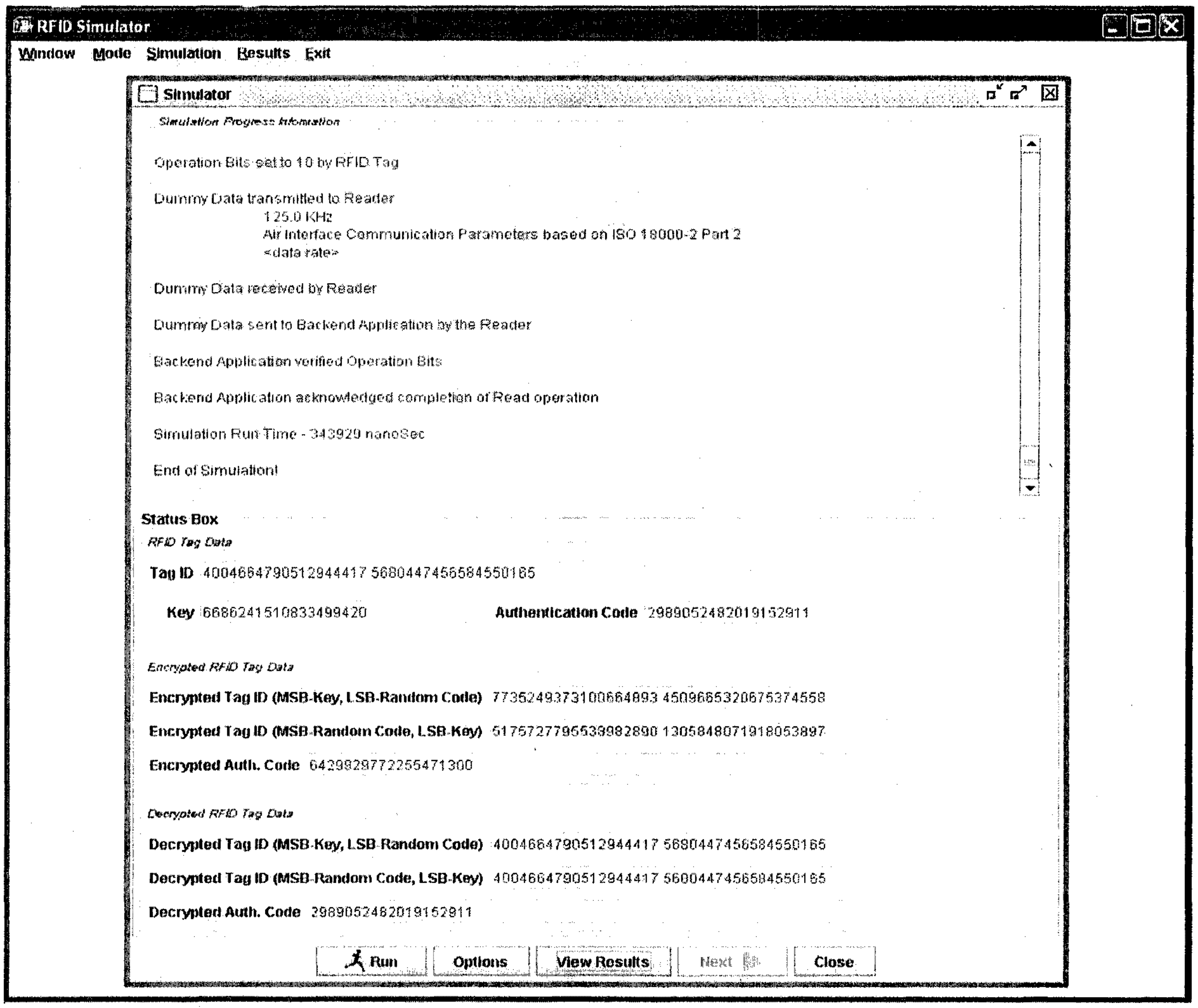
RFID Tag Data
Tag ID
Key

Encrypted RFID Tag Data
Encrypted Tag ID (MSB-Key, LSB-Random Code)
Encrypted Tag ID (MSB-Random Code, LSB-Key)
Encrypted Auth. Code

Decrypted RFID Tag Data
Decrypted Tag ID (MSB-Key, LSB-Random Code)
Decrypted Tag ID (MSB-Random Code, LSB-Key)
Decrypted Auth. Code

Figure 19 Options window of the Simulator

The following figure shows the simulator window, after a simulation run, with the default configuration (Boolean Encryption Type; Real-time mode; Low frequency range; Unmodulated frequency):



Simulation Progress Information

Operation Bits set to 10 by RFID Tag

Dummy Data transmitted to Reader
125.0 KHz
Air Interface Communication Parameters based on ISO 18000-2 Part 2
<data rate>

Dummy Data received by Reader

Dummy Data sent to Backend Application by the Reader

Backend Application verified Operation Bits

Backend Application acknowledged completion of Read operation

Simulation Run Time - 343929 nanoSec

End of Simulation!

Status Box

RFID Tag Data

Tag ID 4004664790512944417 5680447456584550165

Key 6686241510833499420 **Authentication Code** 2989052482019152911

Encrypted RFID Tag Data

Encrypted Tag ID (MSB-Key, LSB-Random Code) 7735249373100664893 4509665320675374558

Encrypted Tag ID (MSB-Random Code, LSB-Key) 5175727795539982890 1305848071918053897

Encrypted Auth. Code 6429829772255471300

Decrypted RFID Tag Data

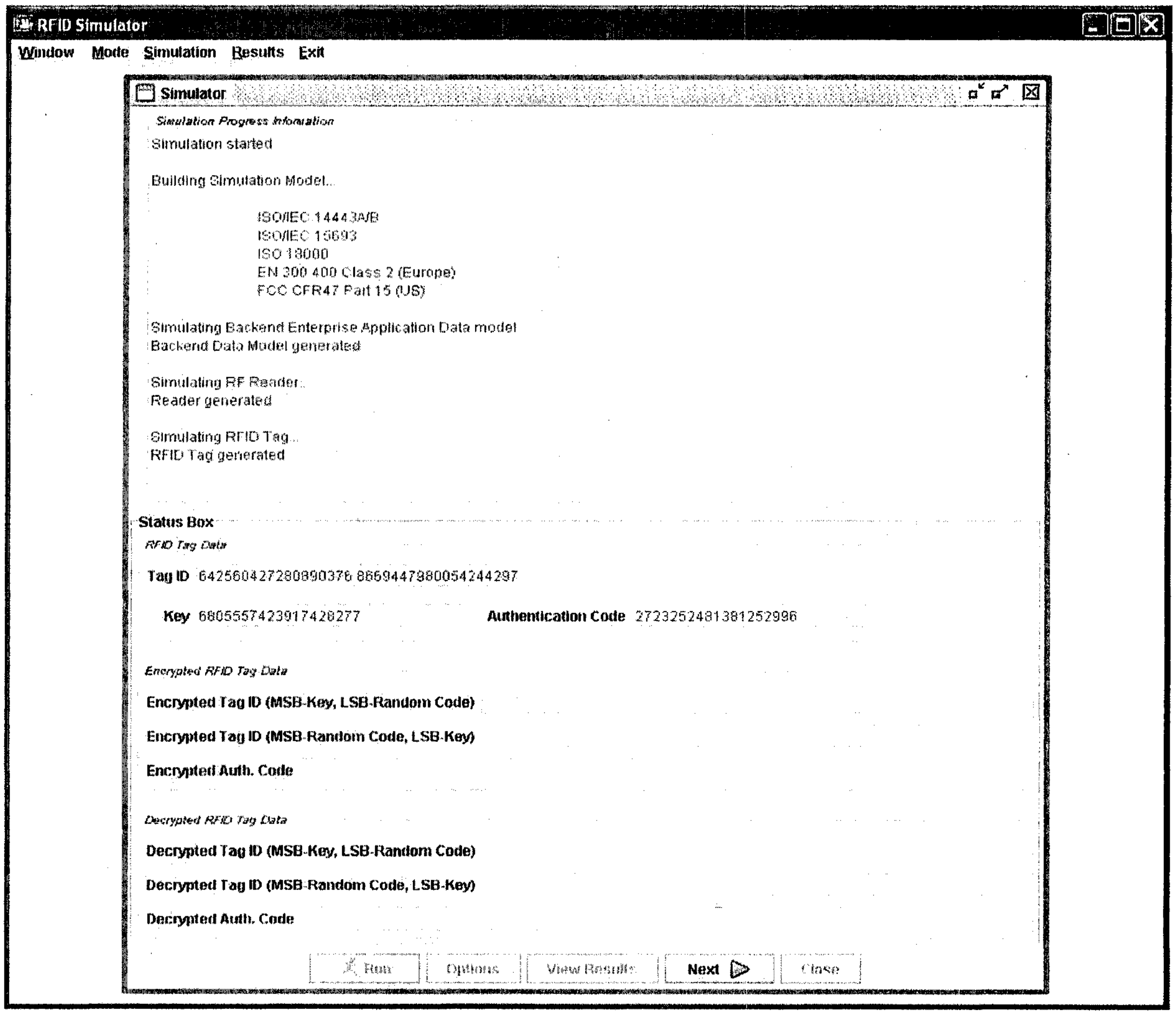
Decrypted Tag ID (MSB-Key, LSB-Random Code) 4004664790512944417 5680447456584550165

Decrypted Tag ID (MSB-Random Code, LSB-Key) 4004664790512944417 5680447456584550165

Decrypted Auth. Code 2989052482019152911

Figure 20 Simulator window after a simulation run

When running in the process mode, the simulator window freezes letting the user hit only the next button. The user may choose to stop to view the simulation process after every single step and the simulation does not proceed until the user clicks on the Next button. This mode is for a better understanding of the protocol. The following figure shows the simulator window running in Process mode:



```
Simulation Progress Information
Simulation started

Building Simulation Model...

                ISO/IEC 14443A/B
                ISO/IEC 15693
                ISO 18000
                EN 300 400 Class 2 (Europe)
                FCC CFR47 Part 15 (US)

Simulating Backend Enterprise Application Data model
Backend Data Model generated

Simulating RF Reader...
Reader generated

Simulating RFID Tag...
RFID Tag generated
```

Status Box

RFID Tag Data

Tag ID 642560427280890376 866944798005424429 7

Key 6805557423917428277 **Authentication Code** 2723252481381252996

Encrypted RFID Tag Data

Encrypted Tag ID (MSB-Key, LSB-Random Code)

Encrypted Tag ID (MSB-Random Code, LSB-Key)

Encrypted Auth. Code

Decrypted RFID Tag Data

Decrypted Tag ID (MSB-Key, LSB-Random Code)

Decrypted Tag ID (MSB-Random Code, LSB-Key)

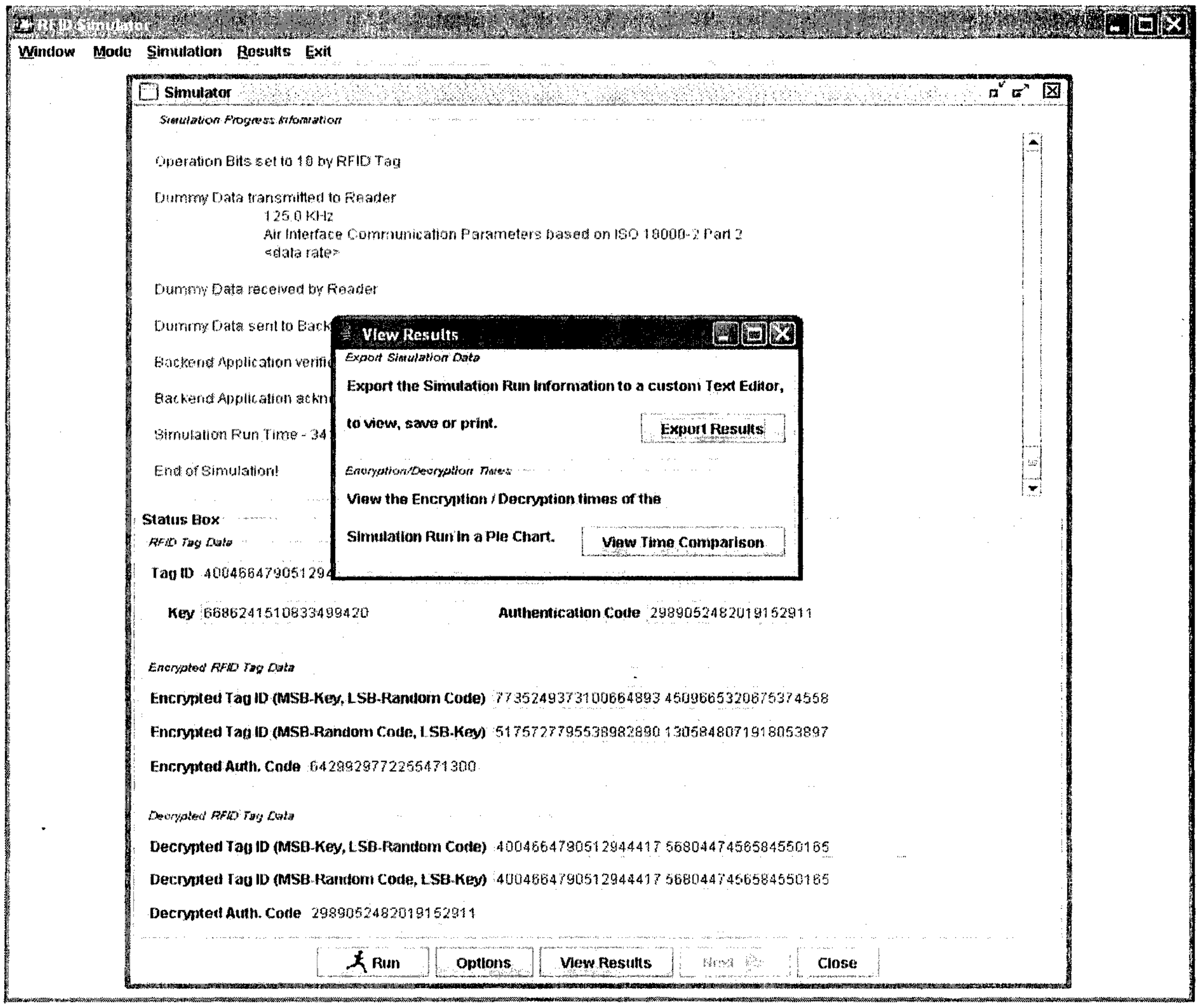
Decrypted Auth. Code

Figure 21 Simulator window when running in Process mode

Once the simulation run is completed, the View Results button can be clicked to view the results of the simulation. It opens a View Results window which gives the options of exporting the simulation data and to view the encryption time details as a pie chart. When the Export Results option is chosen, the simulation data are exported to a custom text editor, where the data can be manipulated, saved as a text file or printed. If the Chart option is chosen, the time needed for encryption/ decryption at various stages of the simulation run are displayed, as a Pie Chart, in nanoseconds.

4.4 View Results Window

The following figures show the View Result window, the exported results and the pie chart with the results of the simulation run:



Operation Bits set to 10 by RFID Tag

Dummy Data transmitted to Reader
125.0 KHz
Air Interface Communication Parameters based on ISO 18000-2 Part 2
<data rate>

Dummy Data received by Reader

Dummy Data sent to Back

Backend Application verifi

Backend Application ackn

Simulation Run Time - 34

End of Simulation!

Export the Simulation Run Information to a custom Text Editor, to view, save or print.

View the Encryption / Decryption times of the Simulation Run in a Pie Chart.

Tag ID 400466479051294

Key 6686241510833499420

Authentication Code 2989052482019152911

Encrypted Tag ID (MSB-Key, LSB-Random Code) 7735249373100664893 4509665320675374558

Encrypted Tag ID (MSB-Random Code, LSB-Key) 5175727795538982890 1305848071918053897

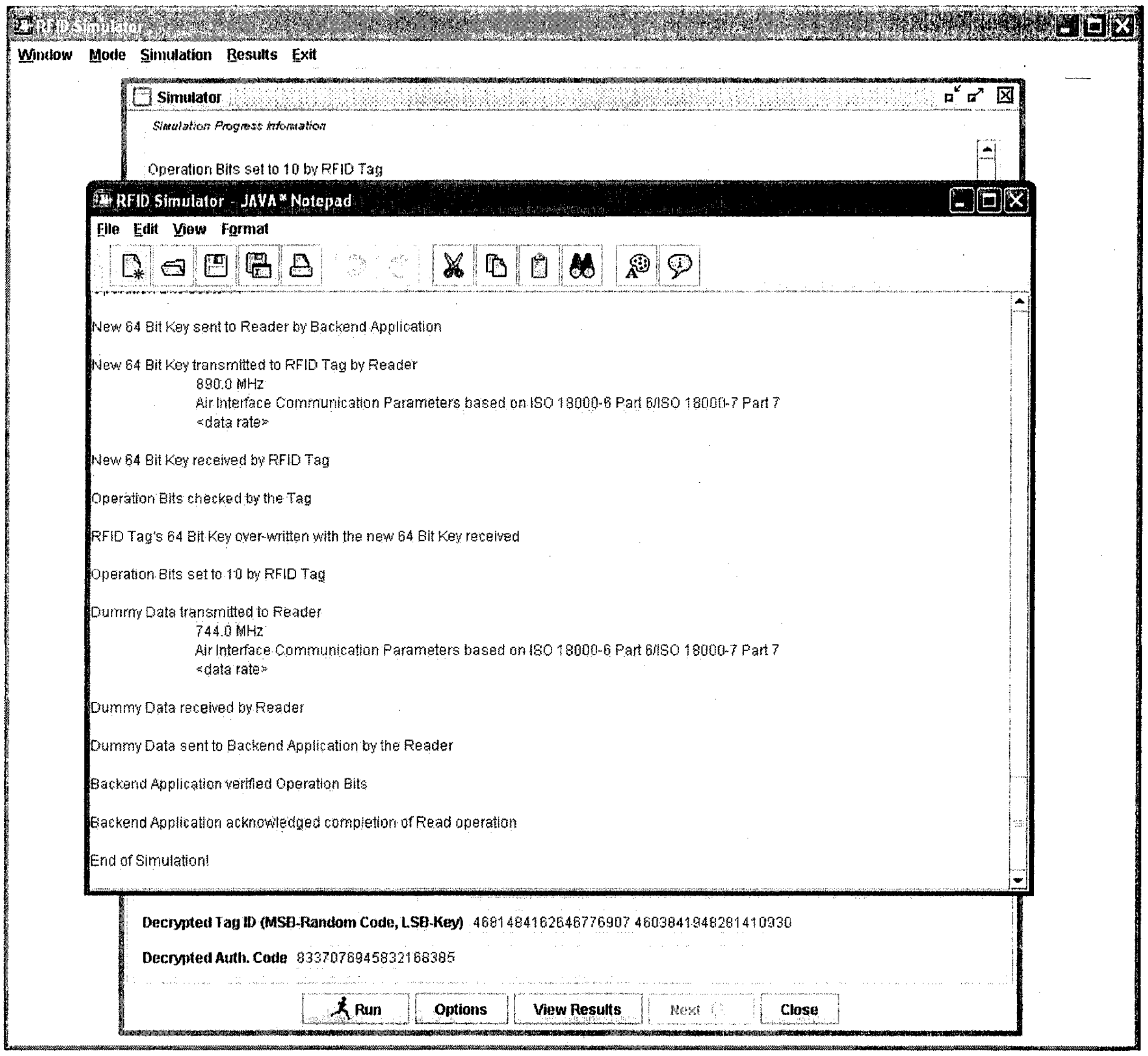
Encrypted Auth. Code 6429929772255471300

Decrypted Tag ID (MSB-Key, LSB-Random Code) 4004664790512944417 5680447456584550165

Decrypted Tag ID (MSB-Random Code, LSB-Key) 4004664790512944417 5680447456584550165

Decrypted Auth. Code 2989052482019152911

Figure 22 View Results window of the Simulator


RFID Simulator
Window Mode Simulation Results Exit
Simulator
Simulation Progress Information
RFID Simulator - JAVA* Notepad
File Edit View Format
Run
Options
View Results
Next
Close

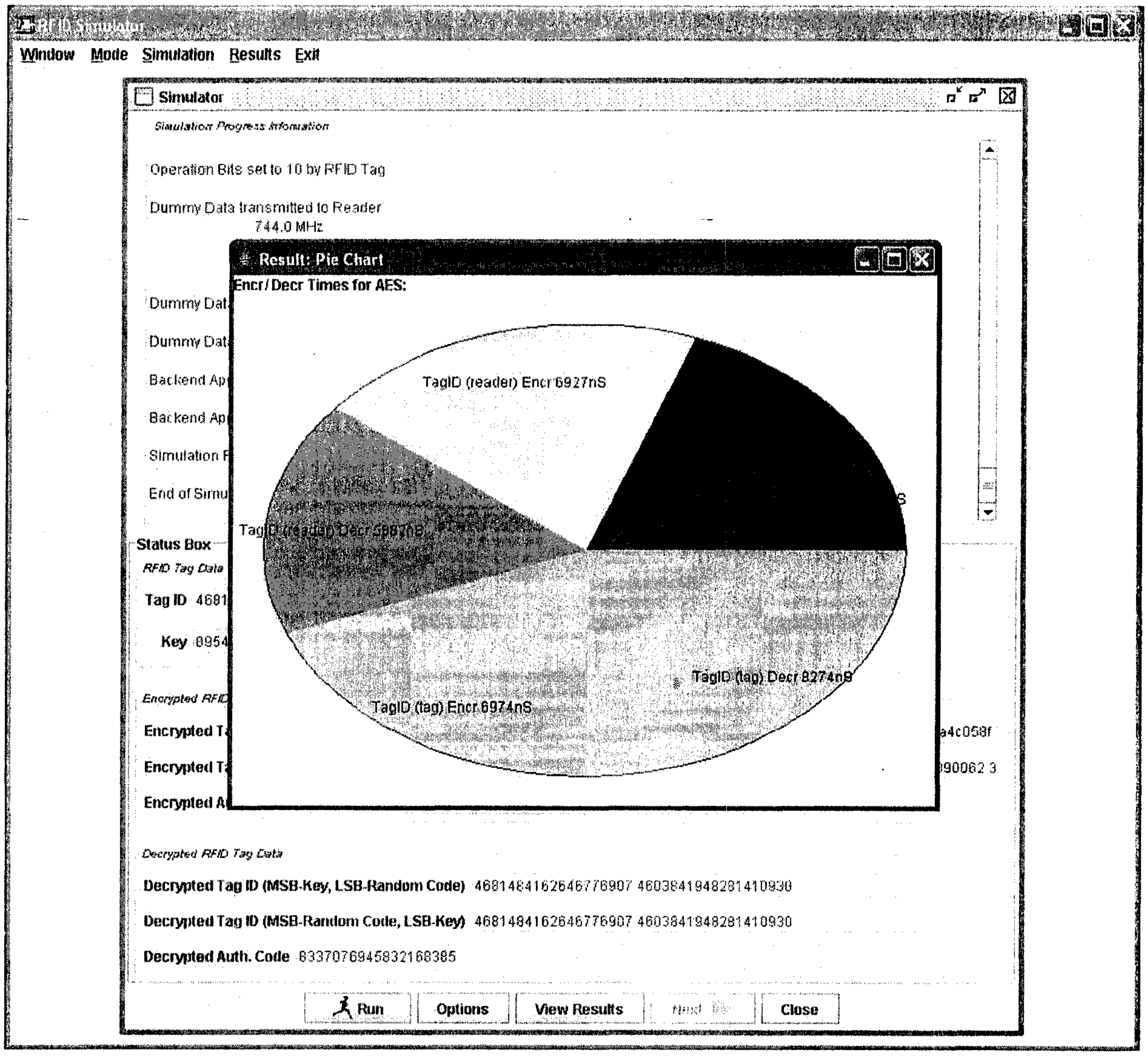
```
Operation Bits set to 10 by RFID Tag
```

```
New 64 Bit Key sent to Reader by Backend Application

New 64 Bit Key transmitted to RFID Tag by Reader
                    890.0 MHz
                    Air Interface Communication Parameters based on ISO 18000-6 Part 6/ISO 18000-7 Part 7
                    <data rate>

New 64 Bit Key received by RFID Tag

Operation Bits checked by the Tag

RFID Tag's 64 Bit Key over-written with the new 64 Bit Key received

Operation Bits set to 10 by RFID Tag

Dummy Data transmitted to Reader
                    744.0 MHz
                    Air Interface Communication Parameters based on ISO 18000-6 Part 6/ISO 18000-7 Part 7
                    <data rate>

Dummy Data received by Reader

Dummy Data sent to Backend Application by the Reader

Backend Application verified Operation Bits

Backend Application acknowledged completion of Read operation

End of Simulation!
```

Decrypted Tag ID (MSB-Random Code, LSB-Key) 4681484162646776907 4603841948281410930

Decrypted Auth. Code 8337076945832168385

Figure 23 Simulation Results exported to a custom text editor



Simulation Progress Information

Operation Bits set to 10 by RFID Tag

Dummy Data transmitted to Reader
744.0 MHz

Dummy Dat

Dummy Dat

Backend Ap

Backend Ap

Simulation F

End of Simu

Status Box

RFID Tag Data

Tag ID 4681

Key 0954

Encrypted RFID

Encrypted T a4c058f

Encrypted T 90062 3

Encrypted A

Decrypted RFID Tag Data

Decrypted Tag ID (MSB-Key, LSB-Random Code) 46814841626467769O7 4603841948281410930

Decrypted Tag ID (MSB-Random Code, LSB-Key) 4681484162646776907 4603841948281410930

Decrypted Auth. Code 8337076945832168385

Figure 24 Pie-chart of time needed for encryption/decryption in simulation run

This Simulator is based only on the protocol proposed in this thesis and cannot be used for a general RFID system, although the code can be modified to make it a general RFID simulator.

CHAPTER 5

SECURITY PROTOCOL ANALYSIS

RFID Applications are probably the most pervasive technologies, but can be exposed to new threats on security and privacy. A typical RFID Application is a really simple system, in terms of the communication between its entities. Wireless exchange of data between low-power devices makes an RFID system, highly vulnerable to a wide range of attacks.

[39] discusses in detail, the different types of attacks that are possible on low cost Identification devices. The basic possible attacks on an RFID system are listed below [40]:

- Eavesdropping
- Man-in-the-middle
- Replay attack
- Data Loss (DoS, Message hijacking)
- Forgery (Decoy Tag, etc.)
- Physical attack

The restrictions on the resources of an RFID system, rule out the possibility of a complex and fool-proof security protocol. The security implementation is always a trade-off between how secure the application is required to be, and an efficient light-weight security protocol. The eventual choice depends on the consequences of a compromised system and on how much data loss is acceptable. A simple protocol that secures the application, from the above-mentioned attacks, in the most efficient way possible, with minimal utilization of resources, is the call of the day.

5.1 Security Analysis against Attacks

The security protocol that is proposed here, tries to resist any attack effectively, as discussed below:

5.1.1 Eavesdropping

An Eavesdropping attack, as the name suggests, is tapping data that is transmitted between two legitimate objects, without their knowledge and without any modification to the data [41]. In case of an RFID system, it refers to an adversary, intercepting the data that is being transmitted between either the Tag and the Reader or the Reader and the backend application, without altering it. The data transmission between the Reader and the backend application is quite generic and can use any protocol, like TCP/IP, SOAP, SSL/TLS, etc. or can use a Middleware that controls the data transmission using one or more of these protocols, implemented with a high level of security, according to the application in question [42]. Since, there is a lot of research already being done, on the security aspects of data transmission between the Reader and the backend application, and since the security protocol proposed in this dissertation deals with the authentication of the RFID Tag and the Reader, it is not imperative, to study the middleware security aspects. The attack on the communication between the Reader and the Tag is of more relevance, here.

The fact that the Tag and the Reader communicate using RF waves, in a wireless medium, makes Eavesdropping an imminent possibility. A single antenna that is within the range of the RFID system, is sufficient to receive the RF signals that are being transmitted, and decoded later [43]. It is practically impossible to stop any other RF receiver from receiving the RF signals and hence the proposed protocol tries to overcome this by transmitting dummy data, along with the original data so that even if the attacker, or Eavesdropper, intercepts the data, it would not make much sense to him.

The transmission protocol uses Frequency Division Multiplexing, along with Frequency Hopping. The frequency range that the RFID devices operate in is divided into bands and data is transmitted through all the bands synchronously. Frequency Division Multiplexing is the traditional access method for Radio signal transmission. The RFID device transmits the required data through one of the frequency bands and dummy data through the rest of the bands. The security implementation of the RF application is such that, the frequency at which the legitimate data is sent, keeps changing for every transmission. This type of access, Frequency Hopping – Code Division Multiple Access (FH-CDMA), is used to minimize the effectiveness of unauthorized interception or jamming of communication [29]. This type of access method does not eliminate Eavesdropping but reduces the possibility of the attack by more than 90%. Thus, even if the transmitted data was intercepted, the eavesdropper would not know if it is the original data and with the frequency hops changing with a Spread function [29], it can be harder to validate any data that might be even intercepted.

5.1.2 Man-In-The-Middle

A Man-In-The-Middle attack (MITM) is an attack in which an attacker is able to read, insert and modify at will, messages between two parties without either party knowing that the link between them has been compromised [44]. This type of attack, against an RFID system would mean an attacker being able to intercept and modify data that is being exchanged between either the Reader and the Tag or the Reader and the Backend application. Again, we choose to ignore the security aspects of communication between the Reader and the Backend application.

An MITM attack on the communication between the Tag and the Reader is probably the most adverse attack and is capable of having the whole system compromised. Eavesdropping, discussed above, is a special case of the MITM attack in which the data is intercepted but not modified.

Assuming the attacker intercepts the legitimate data, the security implementation should make sure the data is encrypted so as to make it as obscure as possible. Encrypting the data before transmitting it reduces the loss by a big margin, but however, as discussed earlier, the type of encryption is always a trade-off and depends on the type of encryption the RF application can afford. The proposed security protocol recommends the use of an encryption algorithm that uses a key, which is changed after every successful read-response between the Tag and the Reader. It should be noted that, encryption and decryption are practically impossible when using passive RFID Tags, tags which don't have a power source of their own.

The proposed protocol secures the physical layer of communication by using Frequency Hopping. With frequency hopping, only the valid tag and the reader know which frequency is being used for legitimate communication, with the rest carrying useless data. For a Man-in-the-middle, figuring the frequency hop sequence is impossible as it is randomly generated, and hence even though intercepting data is possible in an RFID system, the Man-in-the-middle attack is proven to be not so effective.

5.1.3 Replay Attack

A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack by IP packet substitution (such as stream cipher attack) [45]. The various methods of preventing replay attacks are discussed in [46, 47, 49], which include session timeouts, timestamping, statistical analysis, etc.

A Replay attack on an RFID system would mean an attacker intercepting data that is transmitted between the tag and the reader, and repeating or delaying it. In a replay attack, an adversary broadcasts an exact replay of the transponder end of the radio signal recorded from a past transaction between an RF device and a reader. Where mechanisms exist to foil simple replays (such as time stamps, one-time passwords, and challenge-response cryptography) a related but more sophisticated attack can frequently still succeed. This attack, commonly known as the relay attack, uses a man in the middle adversary to relay an ephemeral connection from a legitimate reader through one or more adversarial devices to a legitimate tag which may be at a considerable distance. The distance at which the relay attack can succeed is limited only by the latency which will be tolerated by the attacked protocol [49]. According to [50], a simple security protocol based on Semi-Randomized Access Control (SRAC) is efficient enough in securing a low-cost RFID system from replay and man-in-the-middle attacks.

The proposed protocol does not define any measures to tackle a replay attack in particular, but its current implementation is impregnable against it and also relay attacks. The security protocol conceals the legitimate data by transmitting in more than one channel and even if the frequency hop sequence is cracked, the data is encrypted using keys that keep changing, which would make very little sense to the attacker. Since, this implementation averts a replay attack by itself, any specific prevention techniques like timestamps, session timeouts, etc. will just be an overhead on an RFID system. If needed, stronger measures to prevent replay attacks could be implemented at the cost of a more complex system.

5.1.4 Data Loss/Denial of Service

A denial-of-service attack (DoS attack) is an attempt to make a computer resource unavailable to its intended users. A DoS attack can either force the victim to reset or consume its resources such that it can no longer provide its intended service or obstruct the communication media between the intended users and the victim so that they can no longer communicate adequately [51].

Denial-of-service denies service to valid users. Denial-of-service attacks are easy to accomplish and difficult to guard against. We are unaware of any protocol-level technique for circumventing the effects of a malicious blocker tag, but explore simple ways of detecting the presence of such a device. An attacker kills tags in the supply chain, warehouse, or store disrupting business operations and causing a loss of revenue. Some tags have a “kill” command to destroy it to protect consumer privacy. If implemented in the tag, an attacker can “kill” the tag if the password is known. An attacker can also simulate many RFID tags simultaneously causing the anti-collision to perform singulation on a large number of tags making the system unavailable to authorized use. An attacker can also use a powerful jammer that jams the reader by creating a more powerful return signal than the signal returned from the tags and thus making the system unavailable to authorized users or could use a Faraday's Cage [52, 53].

The proposed protocol does not implement a "kill" command but keeps changing the public key used for encryption after every read-response cycle. However, it does not protect the system against Denial of Service by signal jamming. The system can be scaled to incorporate specific security against these sorts of attacks. If the data is made unavailable or if it's delayed, the proposed protocol has a timeout function, which is going to timeout the operation and the whole data exchange is dropped. The reader is going to send a new request at a different frequency which might not be jammed. If the whole frequency range is jammed or if there is any physical factor that causes the loss of data, the protocol does not provide any means of security and has to be troubleshooted manually.

5.1.5 Forgery

Forgery refers to an adversary duplicating a valid tag or a reader to fake a legitimate transaction and steal data. The proposed protocol never lets an intruder get the details of a tag from the reader or from the tag. But if it's compromised, the possibility of spoofing the tag is quite imminent. For the attacker to even duplicate, he needs to gather information about the system, like the frequency range, frequency hop sequence, or the tag ID, Authentication Code, etc.

5.1.6 Physical Attacks

Physical attacks are pretty straightforward and could be anything that could make the tag or the reader dysfunctional. The security protocol does not handle physical attacks. The RFID system needs to be physically secured from the proximity of any other electromagnetic field or an attacker who has easy access to the hardware of the RFID system setup.

The proposed protocol seems to work pretty fine against most of the attacks mentioned above. It is also proven to resist the STRIDE model of security threats for an RFID system. To test the protocol, a Simulator was coded in java, and the simulation runs are used to corroborate the fact that this security protocol is pretty strong and rugged.

5.2 Encryption Implementation Analysis

The proposed Security protocol was tested with three different encryption methods, Boolean Encryption, Tiny Encryption Algorithm and Advanced Encryption Standard. Each of these encryption methods differ in the complexity of the algorithm, in terms of time and resources. Since the efficiency of these algorithms are already known, the question is about how well these algorithms would fit in an RFID system and enhance its performace. The memory resource utilized could not be computed when using a simulator but their performance was meansured in terms of time needed for encryption and decryption, when used in an RFID system. The following tables show the time needed for encryption and decryption, for each of these algorithms, over 10 runs, when run with the Simulator.

5.2.1 Boolean (XOR) Encryption/ Decryption

Simulation Run	Auth Code Encryption Time (nS)	Tag ID LSB-Random number Encryption Time (nS)	Tag ID MSB-Key Encryption Time (nS)	Tag ID LSB-Key Encryption Time (nS)	Tag ID MSB-Auth Code Encryption Time (nS)	Average Encryption Time (nS)
1	5	3	0	3	3	2.8
2	5	4	0	3	3	3.0
3	5	4	0	3	3	3.0
4	4	4	0	3	3	2.8
5	5	4	0	3	3	3.0
6	5	3	0	3	3	2.8
7	4	3	0	3	3	2.6
8	5	4	0	3	3	3.0
9	4	4	0	3	3	2.8
10	5	4	0	3	4	3.2

Table 2 Boolean XOR Encryption

The encryption time in nanoseconds for different encryptions in the protocol over 10 simulation runs is tabulated as shown above. The average encryption time over 10 runs is 2.9 nanoseconds.

Simulation Run	Auth Code Decryption Time (nS)	Tag ID LSB-Random number Decryption Time (nS)	Tag ID MSB-Key Decryption Time (nS)	Tag ID LSB-Key Decryption Time (nS)	Tag ID MSB-Auth Code Decryption Time (nS)	Average Decryption Time (nS)
1	3	3	3	3	4	3.2
2	3	3	3	3	3	3.0
3	4	3	3	3	3	3.2
4	3	3	3	3	3	3.0
5	4	3	4	3	3	3.4
6	3	3	3	3	3	3.0
7	3	3	3	3	3	3.0
8	4	3	4	3	3	3.4
9	4	3	3	3	3	3.2
10	4	3	3	3	3	3.2

Table 3 Boolean XOR Decryption

The decryption time in nanoseconds for different decryptions in the protocol over 10 simulation runs is tabulated as shown above. The average decryption time over 10 runs is 3.16 nanoseconds.

5.2.2 TEA Encryption/ Decryption

Simulation Run	**Auth Code Encryption Time (nS)**	**Tag ID LSB-Random number Encryption Time (nS)**	**Tag ID MSB-Key Encryption Time (nS)**	**Tag ID LSB-Key Encryption Time (nS)**	**Tag ID MSB-Auth Code Encryption Time (nS)**	**Average Encryption Time (nS)**
1	1193	1152	1096	1201	1155	1159.4
2	1205	1183	1175	1192	1176	1186.2
3	1187	1164	1188	1186	1159	1176.8
4	1196	1158	1179	1164	1168	1173
5	1163	1181	1143	1159	1177	1164.6
6	1145	1201	1152	1160	1153	1162.2
7	1142	1175	1144	1161	1164	1157.2
8	1211	1167	1169	1173	1163	1176.6
9	1189	1159	1163	1145	1172	1165.6
10	1181	1163	1181	1179	1187	1178.2

Table 4 Tiny Encryption Algorithm Encryption

The encryption time in nanoseconds for different encryptions in the protocol over 10 simulation runs is tabulated as shown above. The average encryption time over 10 runs is 1169.98 nanoseconds.

Simulation Run	Auth Code Decryption Time (nS)	Tag ID LSB-Random number Decryption Time (nS)	Tag ID MSB-Key Decryption Time (nS)	Tag ID LSB-Key Decryption Time (nS)	Tag ID MSB-Auth Code Decryption Time (nS)	Average Decryption Time (nS)
1	1184	1148	1138	1144	1147	1152.2
2	1198	1179	1174	1177	1181	1181.8
3	1173	1231	1178	1195	1195	1194.4
4	1207	1152	1155	1167	1153	1166.8
5	1154	1175	1184	1163	1202	1175.6
6	1133	1172	1192	1181	1177	1171
7	1124	1176	1190	1182	1213	1177
8	1200	1200	1184	1175	1186	1189
9	1125	1185	1194	1131	1156	1158.2
10	1174	1175	1168	1164	1167	1169.6

Table 5 Tiny Encryption Algorithm Decryption

The decryption time in nanoseconds for different decryptions in the protocol over 10 simulation runs is tabulated as shown above. The average decryption time over 10 runs is 1173.56 nanoseconds.

5.2.3 AES Encryption/ Decryption

Simulation Run	Auth Code Encryption Time (nS)	Tag ID LSB-Random number Encryption Time (nS)	Tag ID MSB-Key Encryption Time (nS)	Tag ID LSB-Key Encryption Time (nS)	Tag ID MSB-Auth Code Encryption Time (nS)	Average Encryption Time (nS)
1	3341	5153	3152	3054	3576	3655.2
2	4347	5651	4308	4209	2482	4199.4
3	4731	3497	3211	3199	3415	3610.6
4	3204	2948	2666	3225	4180	3244.6
5	5977	3016	2387	2374	2376	3226
6	4170	4246	3708	3587	3470	3836.2
7	2794	2559	2393	2656	3140	2708.4
8	4574	3108	4196	4963	3472	4062.6
9	3531	3884	3043	4341	2633	3486.4
10	3290	3073	3053	4146	3034	3319.2

Table 6 Advanced Encryption Standard Encryption

The encryption time in nanoseconds for different encryptions in the protocol over 10 simulation runs is tabulated as shown above. The average encryption time over 10 runs is 3534.86 nanoseconds.

Simulation Run	Auth Code Decryption Time (nS)	Tag ID LSB-Random number Decryption Time (nS)	Tag ID MSB-Key Decryption Time (nS)	Tag ID LSB-Key Decryption Time (nS)	Tag ID MSB-Auth Code Decryption Time (nS)	Average Decryption Time (nS)
1	3054	2894	2934	4115	3154	3230.2
2	2817	2763	2739	4963	4009	3458.2
3	3229	3018	3090	4923	3066	3465.2
4	2762	2409	2414	3621	2429	2727
5	2433	2229	2318	2310	2244	2306.8
6	3257	3275	2852	3571	3162	3223.4
7	2267	2134	2223	2129	2694	2289.4
8	3085	2987	3016	2981	3627	3139.2
9	3297	2970	2912	4205	4069	3490.6
10	4103	3053	2946	3034	2942	3215.6

Table 7 Advanced Encryption Standard Decryption

The decryption time in nanoseconds for different decryptions in the protocol over 10 simulation runs is tabulated as shown above. The average decryption time over 10 runs is 3054.56 nanoseconds.

The following diagram shows the graph of both the average encryption time and the average decryption time, for the three encryption types, over 10 simulation runs:

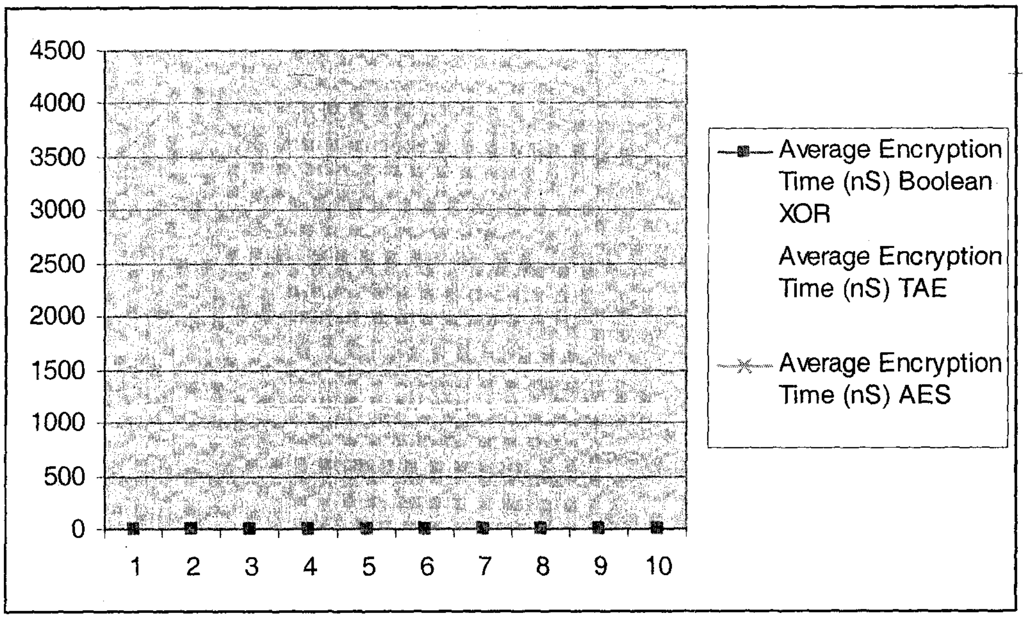


Figure 25 Simulation Run plotted against the Average Encryption Time for the 3 encryption types

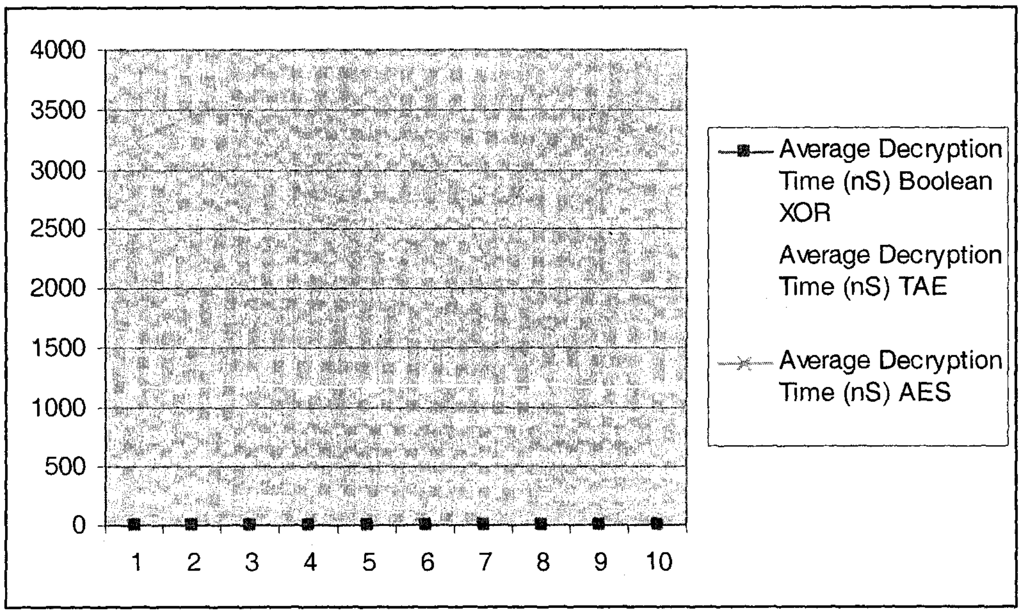


Figure 26 Simulation Run plotted against the Average Decryption Time for the 3 encryption types

CHAPTER 6

CONCLUSION AND FUTURE WORK

In the course of this thesis, it was observed that the proposed security protocol can secure an RFID system from the most sophisticated attacks. It can be implemented in any active RFID system but has to be modified in order to be implemented in an RFID system with passive tags. Tiny Encryption Algorithms seems to suit the RFID application better than any other encryption algorithm, as mentioned in [17]. Overall, the proposed protocol scores over the existing ones with a simple authentication procedure and a robust encryption and secure medium access, implemented in a typical RFID system.

With the rapid growth of RFID each day, the proposed protocol will find extensive use in future because with wider deployment of more complex applications, the effective cost of this protocol which might seem a little expensive for a light-weight RFID system, will go down. This protocol can also be tested with different types of encryption to protect data and ensure privacy. The type of encryption is a trade-off and can be chosen according to the application that uses RFID. Hence, this protocol can be substituted with any other encryption that is feasible for RFID operation and can also be modified to save power and memory resources, though at the cost of some security compromised.

BIBLIOGRAPHY

[1] Radio Frequency Identification. (2006, December 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 02:00, December 5, 2006, from http://en.wikipedia.org/w/index.php?title=Radio_Frequency_Identification&oldid=91672795

[2] Standardization Requirements within the RFID Class Structure Framework Daniel W. Engels Sanjay E. Sarma MIT AUTO-ID LABS TECHNICAL REPORT, JANUARY 2005

[3] http://www.ti.com/rfid/shtml/prod-trans-RI-I11-114A.shtml

[4] http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=54

[5] A Basic Introduction to RFID Technology and its use in the supply chain Laran RFID white paper http://www.printronix.com/library/assets/public/case-studies/rfid-laran-white-paper-english.pdf

[6] History of RFID, Jeremy Landt

[7] RFID Systems: A Survey on Security Threats and Proposed Solutions Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M. Estevez-Tapiador, and Arturo Ribagorda

[8] The Evolution of RFID Security Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum

[9] Coming Soon to a Wal-Mart Near You Rajeev Bansal

[10] http://www.cnn.com/2004/TECH/science/05/24/animalidentification/index.html

[11] http://www.activewaveinc.com/applications_parking_lots.html

[12] http://www.rfidlowdown.com/2006/04/rfid_to_help_tr.html

[13] http://www.whynot.net/ideas/1560

[14] http://www.rfidnews.org/weblog/2006/10/10/rfid-tracks-highend-car-parts/

[15] RFID for Airport Security and Efficiency Thomas McCoy, R J Bullock and PV Brennan

[16] A Feasible Security Mechanism for Low Cost RFID Tags Gwo-Ching Chang

[17] Security and Privacy Solutions for Low-Cost RFID Systems Damith C. Ranasinghe 1, Daniel W. Engels2, Peter H. Cole

[18] Menezes, A., Van Oorschot, P., and Vanstone, S., Handbook of Applied Cryptography, 3rd edition, CRC Press 1996.

[19] Wheeler, D., and Needham, R., "TEA, a Tiny Encryption Algorithm", Computer Laboratory, Cambridge University, England, 1994.

[20] A Lightweight Mutual Authentication Protocol for RFID Networks Zongwei Luo1, Terry Chan2, Jenny S. Li

[21] Authentication Protocol of the Read Only RFID Tag Using Partial ID White Paper Author not specified

[22] Multi-standard High Frequency RFID Reader/Writer IC
http://focus.ti.com/docs/prod/folders/print/trf7960.html

[23] Highly Integrated Multichannel RF Transmitter Designed for Low-Power Wireless Applications http://focus.ti.com/docs/prod/folders/print/cc1150.html

[24] http://focus.ti.com/lit/ug/slou192/slou192.pdf

[25] Single-Chip, Low-Power, Low-Cost CMOS FSK/GFSK/ASK/00K RF Transmitter for Narrowband & Multi-Ch Apps
http://focus.ti.com/docs/prod/folders/print/cc1070.html

[26] http://focus.ti.com/lit/ml/swrb016/swrb016.pdf

[27] RFID Gazette - RFID Middleware
http://www.rfidgazette.org/2005/09/rfid_middleware.html

[28] http://www.dataflows.com/RFID_Overview.html

[29] http://www.javvin.com/networksecurity/FHCDMA.html

[30] RFID Journal Glossary Frequency Hopping
http://www.rfidjournal.com/glossary/Frequency%20hopping

[31] Frequency Hopping
http://www.technovelgy.com/ct/Technology-Article.asp?ArtNum=43

[32] How Bluetooth Works http://electronics.howstuffworks.com/bluetooth1.htm

[33] A. Juels, "Minimalist cryptography for low-cost RFID tags", In 4th Intel. Conf. on Security in Communication Networks-SCN 2004 vol. 3352 LNCS, pp. 149-164.

[34] Choi, Eun Young and Lee, Su Mi and Lee, Dong Hoon, "Efficient RFID Authentication protocol for UbiquitousComputing Environment" In International Workshop on Security in Ubiquitous Computing Systems – secubiq_2005, Volume 3823 LNCS, pp. 945-954

[35] Tiny Encryption Algorithm
http://www.answers.com/topic/tiny-encryption-algorithm

[36] The Design of Rijndael: AES - The Advanced Encryption Standard (Information Security and Cryptography) Joan Daemen, Vincent Rijmen

[37] Advanced Encryption Standard
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[38] Radio Frequency Identification
http://cs.ecs.baylor.edu/~donahoo/classes/5321/projects/rfid/

[39] R. Anderson and M. Kuhn. Low cost attacks on tamper resistant devices. In IWSP: International Workshop on Security Protocols, LNCS, 1997.

[40] Mutual Authentication Protocol Mutual Authentication Protocol for Low for Low-cost RFID Jeongkyu Yang, JaeminPark, HyunrokLee, KuiRen, KwangjoKim Workshop on RFID and Lightweight Crypto, Jul. 14th, 2005

[41] Eavesdropping http://en.wikipedia.org/wiki/Eavesdropping

[42] Jieun Song, Taesung Kim, Sokjoon Lee, and Howon Kim.Security Enhanced RFID Middleware System.TRANSACTIONS ON ENGINEERING, COMPUTING AND TECHNOLOGY V10 DECEMBER 2005 ISSN 1305-5313

[43] Thomas S. Heydt-Benjamin1, Daniel V. Bailey2, Kevin Fu1, Ari Juels2, and Tom O'Hare3.RFID Payment Card Vulnerabilities Technical Report.RFID-CUSP: RFID Credit Card Vulnerabilities - Technical Report

[44] Man-in-the-middle Attack http://en.wikipedia.org/wiki/Man_in_the_middle_attack

[45] Replay Attack http://en.wikipedia.org/wiki/Replay_attack

[46] A Taxonomy of Replay Attacks Paul Syverson Code 5543 Naval Research Laboratory

[47] Strategies against Replay Attacks Tuomas Aura 10th Computer Security Foundations Workshop (CSFW '97)

[48] On Preventing Replay Attacks on Security Protocols Sreekanth Malladi, Jim Alves-Foss, Robert B. Heckendorn

[49] Vulnerabilities in First-Generation RFID-enabled Credit Cards Thomas S. Heydt-Benjamin1, Daniel V. Bailey2, Kevin Fu1, Ari Juels2, and Tom O'Hare

[50] Secure and Low-cost RFID Authentication Protocols Yong Ki Lee1 and Ingrid Verbauwhede1,2

[51] Denial of Service Attack http://en.wikipedia.org/wiki/Denial-of-service_attack

[52] The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy Ari Juels1 and Ronald L. Rivest2 and Michael Szydlo

[53] RFID SECURITY THREAT MODEL Dale R. Thompson, Neeraj Chaudhry, Craig W. Thompson

VITA

Graduate College
University of Nevada, Las Vegas

Karthik Raghavan

Home Address:
4210 Grove Circle, # 1
Las Vegas, Nevada 89119

Degrees:
Bachelor of Engineering, Computer Science, 2003
University of Madras

Master of Science, Computer Science, 2006
University of Nevada, Las Vegas

Thesis Title: Security protocol based on random key generation for an RFID system

Thesis Examination Committee:
Chairperson, Dr. Yoohwan Kim, Ph. D.
Committee Member, Dr. Ajoy Datta, Ph. D.
Committee Member, Dr. Laxmi Gewali, Ph. D.
Graduate Faculty Representative, Dr. Rama Venkat, Ph. D.