Towards Secure and Scalable Tag Search approaches for Current and Next Generation RFID Systems

Farzana Rahman
Marquette University

TOWARDS SECURE AND SCALABLE TAG SEARCH APPROACHES FOR CURRENT AND NEXT GENERATION RFID SYSTEMS

by

Farzana Rahman

A Thesis submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

August 2010

ABSTRACT
TOWARDS SECURE AND SCALABLE TAG SEARCH APPROACHES FOR CURRENT AND NEXT GENERATION RFID SYSTEMS

Farzana Rahman

Marquette University, 2010

The technology behind Radio Frequency Identification (RFID) has been around for a while, but dropping tag prices and standardization efforts are finally facilitating the expansion of RFID systems. The massive adoption of this technology is taking us closer to the well known ubiquitous computing scenarios. However, the widespread deployment of RFID technology also gives rise to significant user security issues. One possible solution to these challenges is the use of secure authentication protocols to protect RFID communications. A natural extension of RFID authentication is RFID tag searching, where a reader needs to search for a particular RFID tag out of a large collection of tags. As the number of tags of the system increases, the ability to search for the tags is invaluable when the reader requires data from a few tags rather than all the tags of the system. Authenticating each tag one at a time until the desired tag is found is a time consuming process. Surprisingly, RFID search has not been widely addressed in the literature despite the availability of search capabilities in typical RFID tags. In this thesis, we examine the challenges of extending security and scalability issues to RFID tag search and suggest several solutions.

This thesis aims to design RFID tag search protocols that ensure security and scalability using lightweight cryptographic primitives. We identify the security and performance requirements for RFID systems. We also point out and explain the major attacks that are typically launched against an RFID system. This thesis makes four main contributions. First, we propose a serverless (without a central server) and untraceable search protocol that is secure against major attacks we identified earlier. The unique feature of this protocol is that it provides security protection and searching capacity same as an RFID system with a central server. In addition, this approach is no more vulnerable to a single point-of-failure. Second, we propose a scalable tag search protocol that provides most of the identified security and performance features. The highly scalable feature of this protocol allows it to be deployed in large scale RFID systems. Third, we propose a hexagonal cell based distributed architecture for efficient RFID tag searching in an emergency evacuation system. Finally, we introduce tag monitoring as a new dimension of tag searching and propose a Slotted Aloha based scalable tag monitoring protocol for next generation WISP (Wireless Identification and Sensing Platform) tags.

ACKNOWLEDGMENTS

Farzana Rahman

This work is the result of my Master's studies at Marquette University as well as of all the months I spent at UbiComp Lab to work on the fascinating topics of RFID Security. During this period, many persons happened to cross my way. Sure I will forget some important names. But here I want to shortly recall some of those I more deeply interacted with and therefore had - in a way or another - an impact on my thesis work.

Firstly, I would like to thank my supervisor, Dr. Sheikh Iqbal Ahamed for his invaluable advice throughout the last two years at Marquette. I have learned great things from him in these years. I will always be thankful to Dr. Ahamed for giving me the opportunity to work with him.

I'm also grateful to the thesis committee members Dr. Douglas Harris and Dr. Praveen Madiraju for their invaluable comments and patience during the preparation of the thesis. I would like to appreciate my fellow UbiComp lab members for their benevolent support and valuable advice, which were very useful to drive the research towards the right direction.

I would like to thank my parents for giving me the wonderful childhood, love, and support. My sincere gratitude goes to my in-laws who were very supportive of my every bit of work. I want to thank my loving sister for putting up with me in spite of all the childish fights and arguments we've had. Many thanks must go to my wonderful friends in Milwaukee, Bangladesh and everywhere else.

Above all, my special thanks go to my husband, Md. Endadul Hoque, for his never ending support, for making me believe that I am capable, and for always being there with me. I would like to thank him for encouraging me, helping me, and tolerating me! Thank you as you know without you I would never have made it this far. Lastly, thanks to everybody advising me or somehow contributing to my future career, which starts here where my Master's studies end.

TABLE OF CONTENTS

LIST OF TABLES

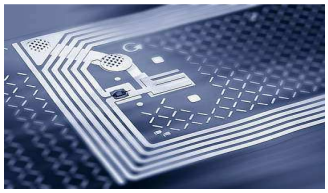
LIST OF FIGURES

LIST OF ABBREVIATIONS AND ELABORATIONS

TERM	ELABORATION
RFID	Radio Frequency IDentification
RF	Radio Frequency
EPC	Electronic Product Code
CBI	Cross-Band Interrogation
IFF	Identify Friend or Foe
WWII	World War II
S-Search	Scalable Search protocol
S^3PR	Secure, Serverless Search Protocol
EDSA	Enhanced Distributed Scalable Architecture
MonAC	Monitor And Collect
PRNG	Pseudo Random Number Generator
XOR	Exclusive Or
AES	Advanced Encryption Standard
DoS	Denial-of-Service attack
WISP	Wireless Identification and Sensing Platform
CRFID	Computational RFID
ICU	Intensive Care Unit
BR	Bit Record
SP	Slot Position
HB	Hopper-Bloom Protocol

Chapter 1: Introduction

Radio Frequency IDentification (RFID) (see figure 1.1) is the classic pervasive computing technology. RFID is plugged as the replacement for traditional barcodes and its' wireless identification capabilities promise to revolutionize our industrial, commercial, and medical experiences. What makes RFID unique is that it facilitates information gathering about physical objects easy. Information about RFID tagged objects can be read through physical barriers and from a distance. In line with Mark Weiser's concept of ubiquitous computing [Weiser93, Pervasive1, and Pervasive2], RFID tags could turn our interactions with computing infrastructure into something subconscious.



Source: [Bocchetti08]

Figure 1.1 Design of an RFID tag

Each RFID system has three main components: tag, reader, and database. An RFID reader and an RFID tag communicate via a wireless radio communications channel. The base idea of an RFID technology is an automatic identification technique, which relies on storing and remotely retrieving data about objects we want to manage using RFID tags. Some popular applications of RFID are product tracking in a supply chain [Li07], toll payments [Mayes09], access control [Juels05b], patient recognition in hospitals [Juels05b], automatic vehicle identification [Juels05b], point of sale applications [Juels05b], library book administration [Juels05b], and e-passports [Juels05c].

We envision that low-cost RFID will be attached to every object in our daily lives, from clothes, books, and pens, to very small objects such as pins and buttons. Annotating objects around us with tags gives us enormous advantage in connecting the physical world with the cyber-world so that people can easily obtain information about the environment and physical

objects. We believe that more powerful tags and readers in the future promise many more applications based on how we may use those tags.

Unit cost per tag is a major consideration for RFID tags because some applications need low cost tags. Cost may be a secondary consideration in passports or credit cards because security is paramount and these devices may pass that cost on to the consumer without much concern. In an application like product tagging, cost is paramount, and the cost per tag needs to be low; otherwise, the benefits of RFID are outweighed by the cost. Securing RFID tags and providing privacy in consumer applications, while limiting cost per tag, has been the focus of much academic work. Due to the constraints on memory, power consumption, and amount of logic on RFID devices, standard cryptographic primitives are often unsuitable.

In recent years, numbers of papers have been published providing solutions to RFID security and privacy challenges. One approach to addressing such privacy and security threats is to use a tag authentication scheme in which a tag is both identified and verified in a manner that does not reveal the tag identity to an attacker. However, RFID tags have limited computation power and storage because of the tag cost requirements. As a result, protocols for RFID systems should not only be designed to address privacy and security threats, but should also take into account the limited capabilities of RFID tags.

1.1. Security and Scalability in RFID Infrastructures

In this section, we explain the meaning of two important terms in perspective of RFID systems. These two terms are: *Security* and *Scalability*. Every RFID system must be secure enough to be used by mass level end users. Scalability of an RFID system is related to its performance and the RFID system must be scalable to satisfy the needs of large number of users. However, from RFID system's perspective, it has been found in literature that security and scalability are two conflicting issues.

1.1.1. Security

Security and privacy of data (and of consumers) is one of the major concerns that have hindered the adoption of RFID technology for many applications. The absence of protocols for privacy and security introduce concerns such as scanning and tracking, cloning, eavesdropping, and replay attacks. However, a major problem of designing cryptographically secure RFID protocols is the lack of computational resources on RFID tags. This prohibits the use of common cryptographic operations to enhance privacy and security in RFID infrastructures. Therefore RFID protocol designers need to keep in mind all the challenges to find some new lightweight alternatives.

1.1.2. Scalability

A protocol is said to be scalable if the number of nodes can be significantly increased without imposing an unacceptable workload on any entity in the network. The interpretation of scalability will vary depending on the context (and the size of the network). Any security protocol deployed in an RFID network should not significantly affect its scalability. In the context of secure RFID systems, we would typically require that the workload on the server, to complete a single transaction, should not be a linear function of the number of deployed RFID tags.

1.2. Motivation

Recent advances in wireless technologies and cost reductions in sensor industries are causing the entire world to shift toward broad adoption of radio frequency identification (RFID) technology. Considering the expanding nature of RFID applications, we believe, one important functionality that an RFID system should provide is tag search, where a reader can detect if a particular tag is present or not. To better understand the situation, we describe some scenarios:

- ***Scenario 1-Container search within seaports:*** Usually there are hundreds and thousands of containers within a seaport. Containers are parked and stacked by hundreds of

employees and countless drivers who deliver containers from remote locations. Moreover, containers are also unloaded from ships in order to deliver them to different customers and locations. Whether a particular container has already been unloaded from the ship or not, whether a specific container has arrived at the seaport for shipment or not, are some of the major tasks performed within seaports. But it is quite impossible to search for a particular container manually. That is why seaports in different countries have long been searching for technologies that can identify specific containers and that can confirm the existence of containers within seaports. One solution to the aforementioned problem can be to use RFID tags for container identification. Now through the use of our serverless search protocols, it will be quite easy to search for a particular container by searching the tag. If a container's tag *id* is known, then a search operation can be invoked with the *id* within the seaport. If the container is present within the seaport then according to our protocol, definitely that particular tag will reply. Thus we can be sure about the container's existence.

- ***Scenario 2-Product Search in a warehouse:*** Let us imagine a warehouse full of tagged items and a manager of the warehouse wants to know if a particular item is present in the warehouse or not. The manager can use a reader to query the tag attached to that item and listen for a correct response from the tag to detect the presence of the item. Using an authentication technique to securely identify the desired item is very inefficient as the reader has to authenticate each tag one at a time. However, using a search technique within the warehouse can make the entire operation secure, efficient and easy for the manager and the reader.

Based on this example application, we define tag search problem and some other terms related to tag searching as follows:

➢ *Definition 1: Tag Searching*
Tag Searching is a process invoked by an RFID reader to determine among a number of tags whether a particular one is present.

- *Definition 2: Target Tag or Desired Tag*

We define the tag being searched for as the *Target Tag* or *Desired Tag*. We assume that the reader knows the identity (*id*) of the target tag and therefore the reader can initiate a search with this id.

(However, initiating the search with the tag id is not secure and therefore the reader needs to encrypt or apply some other techniques to make the search protocol secure)

From the above mentioned two scenarios and definitions, it is easy to infer that tag searching poses challenge to security and privacy. A naive search protocol is that the reader broadcasts the id and the target tag sends back a response. However, this protocol involves severe privacy and security problems. For example, an adversary can easily track the location of the tag using its id he/she overheard, or the attacker can forge the presence of the tag by replaying the overheard response. To solve these problems, we demand a secure search protocol. By a secure search protocol between a reader and a tag, we mean that the following two properties should be satisfied.

Property 1: Only the reader is aware of the identity of the target tag, but an eavesdropper cannot infer the tag's identity from the communication between the reader and the tag.

Property 2: The reader can determine the presence of the tag, but an adversary is not able to forge the tag's presence if it is not present. However, the protocols ensures strong security if the attacker is not able to determine the presence of the tag.

However satisfying the above two properties will make the search protocols secure but the protocols will not be efficient. If we use a naive search approach to find a tag, the computational complexity will increase linearly with the number of the tags and this technique will raise scalability issues.

Suppose we have a large library where each book is equipped with a tag. A book can be easily misplaced by any chance (e.g., because of a visitor's negligence or a librarian's mistake).

Using a randomized authentication protocol to find a specific book is in efficient as the server needs to authenticate half of the books in the library on average. Therefore, designing an efficient, secure search protocol is essential in an RFID system.

In an efficient search protocol, the server would expect to only receive a response from a designated tag. Otherwise, the server would need to handle responses from multiple tags. On the other hand, a tag should not respond before properly authenticating the server since a query may not be from an authentic server, but from an attacker who wants to track the tag. Therefore, the protocol should be a one-round protocol, and a tag should authenticate the server without giving any challenge. When designing a secure, anonymous, untraceable search protocol, we face scalability problems as it increases computational complexity in the reader/back-end server. In other words, there is a tradeoff between scalability and other security parameters. Search protocols for RFID systems should not only be designed to address security threats and scalability issues, but should also consider the little capacities of RFID tags.

A wide variety of authentication protocols for RFID systems have been proposed. Each of the protocols has their own strengths and weakness. Many of these protocols have privacy, security, and/or performance drawbacks. However, tag searching is a relatively new issue and it has been mentioned in limited research literatures [Tan07, Ahamed08b, Kulseng09, and Lee10].

For these reasons, this thesis focuses on the design of RFID search protocols that ensure security and scalability. The thesis begins by identifying the security, scalability and performance requirements for such protocols. We aim to propose novel RFID search protocols that meet the identified requirements. We also aim to propose a new type of tag searching that we name as *tag monitoring* for the next generation tags such as WISP (Wireless Identification and Sensing Platform) tags.

1.3. Major Contributions

In this thesis we consider RFID tag searching protocols that ensure security and scalability. The main contributions of the thesis are as follows:

- We summarize all the possible attacks that can be launched against RFID systems.
- We point out the security requirements that should be guaranteed by the RFID protocol designers to protect against the major security attacks.
- We also point out the scalability and performance requirements for RFID protocols.
- We introduce the notion of serverless (without a central server) RFID tag searching. From this perspective, we propose a lightweight, secure, and serverless search protocol (S^3PR) for RFID systems. The unique feature of this protocol is that it can provide the same level of security and searching capacity as an RFID system with a back end server. Moreover, this protocol is not vulnerable to single point-of-failure as it does not rely on central server.
- We address the tradeoff between security and scalability. From this perspective, we propose a secure and scalable RFID tag searching protocol (*S-Search*) for large scale RFID systems using Slotted ALOHA based technique. This protocol is also lightweight as makes use of simple hash function to provide security. The unique feature of this protocol is that it is highly scalable and therefore it is suitable to be used in large scale RFID networks, such as supply chain and inventory control.
- We propose hexagonal cell based distributed scalable architecture (*EDSA*) for RFID tag searching in an emergency evacuation system. This standard architecture can be used in different RFID applications for scalable tag searching. We analyze and compare our architecture with a prior work. We also prove that our hexagonal cell structure increases the performance of the RFID systems and outperforms the prior work.
- We introduce the concept of *tag monitoring* as a new dimension of tag searching. We propose a tag monitoring protocol (*MonAC*) for WISP based sensor networks. To the best of our

knowledge, this is the first proposal to address the tag monitoring approach for a network of Gen 2 tags, i.e. WISP tags based networks.

For the rest of the thesis, we consider typical RFID tags that are capable of generating Pseudo Random Number (PRNG), performing simple hash function and XOR operation.

1.4. Thesis Organization

The rest of this thesis is structured as follows:

- In chapter 2, we give a brief description of RFID technology. We compare RFID systems with the existing barcode technology. Then, we discuss some popular application areas of RFID technology. Next in this section, we discuss different components of RFID systems and their constraints. Then, we describe RFID standards and point out the differences between different types of EPC class tags. In this chapter, we also compare tags of Gen 1 and Gen 2.

- In chapter 3, we start by pointing out the attack objectives and goals of the RFID system attacker. Then we briefly discuss the security requirements of RFID systems and RFID protocols. Next we define different types of adversary. This is followed by a detailed discussion of different types of attacks in RFID systems. Finally, we explain the attack intentions of an adversary who may have various purpose of attacking the system.

- In chapter 4, we discuss related works relevant to RFID search techniques. Although tag search is a major issue for RFID systems, the assortment of research literature on RFID searching is inadequate. Since RFID tag searching is an extension of RFID authentication, we therefore discuss some famous RFID authentication techniques in this section.

- In chapter 5, we address the problem of secure serverless tag searching. First, we describe the problems of central server based RFID networks and illustrate some situations where serverless RFID searching can be very important. Next, we describe some trivial approaches to solve the problem and point out their shortcomings. We then continue to present our protocol

(*S^3PR*) for a serverless RFID system. Finally, we perform a security analysis of our proposed protocol.

- In chapter 6, we address the problem of scalable tag searching. First, we describe the problem of un-scalable searching approaches for large scale RFID networks. Next, we present a secure and scalable search protocol (*S-Search*) using Slotted ALOHA technique. Finally, we evaluate our protocol by doing a security analysis.

- In chapter 7, we address the problem of a lack of standard architecture to perform scalable tag searching in an RFID system. This is followed by a description of an existing architecture and its shortcomings. Then, we present an enhanced distributed scalable architecture (*EDSA*) with hexagonal cell. This is followed by the comparison between our proposal and the prior work. Finally, we explain the application of our architecture in an emergency evacuation system.

- In chapter 8, we start by giving a brief introduction of a Gen 2 tag (Wireless Identification and Sensing Platform or WISP). Next, we discuss a potential application scenario of WISP. Then, we introduce a new notion of tag searching, tag monitoring, for WISP based networks. This is followed by a brief discussion of the security and scalability problems that may occur while WISP tag monitoring. We then propose a monitoring technique (*MonAC*) which does not require the reader to collect ids from each WISP tag. Finally, security proofs of our proposed protocol are presented.

- In chapter 9, we make our conclusions and describe our future work in the area of securing WISP networks and simple RFID networks.

- The appendix contains definitions of different terms mentioned within the thesis.

1.5. Publications

This thesis contains material that has been published in [Ahamed08a, Ahamed08b, Ahamed08d, Hoque09, and Hoque10]. The contents of [Ahamed08b] form the basis for chapter

5, the contents of [Hoque10] form the basis for chapter 6 and the contents of [Ahamed08a] forms the basis for chapter 7. The contents of [Ahamed08b] have been updated since publication, and an updated version can be found in [Ahamed08d and Hoque09].

Chapter 2: Overview of RFID Technology

The goal of this chapter is to discuss some basics of RFID technology. It starts by highlighting the current evolution of automatic identification from barcodes to RFID and compares the existing auto-id systems. Subsequently current RFID systems are classified and compared. After a system overview has been given, the technical background of RFID readers and tags are discussed. Finally, properties of various RFID standards are discussed at the end of this chapter.

2.1. Historical Perspective of RFID

RFID is the acronym of Radio Frequency IDentification. It designates a large family of technologies and devices all having in common the aim to identify objects or persons with RFID tags. Even if RFID is often thought of as a very new domain, actually it dates back to World War II. British technology IFF (Identify Friend or Foe) has been developed in the late 1930s to help the Royal Air Force to distinguish between friendly and hostile aircrafts and it is the ancestor of RFID technology. Basically, the IFF of WWII and Soviet era systems used coded radar signals (called Cross-Band Interrogation, or CBI) to automatically trigger the aircraft transponder in an aircraft “painted” by the radar. An aircraft responding to an IFF request was then considered a friend, one not responding a foe. This technique was intended to reduce friend-fire. Since then RFID has seen new forms and applications.

Starting in the late 80’s battery powered active RFID devices have been used for automatic toll collecting on motorway (*e.g. Telepass* in Italy). Nevertheless the big revolution, bringing RFID to the attention of common people and media, has certainly been due to the progresses in miniaturization which leaded to very small and cheap tags which are well suited for being applied on single packages of products.

2.2. From Barcodes to RFID

Barcodes are predominantly used for identifying and tracking products throughout the supply chain. Even though they can achieve efficiencies in the order of 90% [Fin03], they still show some limits in the technology, for which RFID is able to provide a better solution and further optimization. Bar coding is a cost-effective and low-risk method of encoding information. RFID on the other hand enables users to encode information for many items simultaneously with no line-of-sight requirement. Unlike bar codes, for which many standards already exist, RFID is just at the beginning of standardization. There are common frequency ranges for example, but the reader power output and specific frequency may vary by company and manufacturer. In addition, systems within the same frequency range may have their own chip set, protocol for memory storage, air protocol and antenna design. With no-contact, no-line-of-sight reading, the RFID tag's position is not as crucial as it is for barcodes. Furthermore RFID tags are more robust than barcodes in foggy and dusty environments. With decreasing equipment and tag costs, RFID gains competitive edge over barcodes.

RFID technology already has started to be applied in several practical situations where barcodes were used to be applied before. For instance, Wal-Mart has recently asked to all its suppliers to embed RFID tags into their products to allow per item tracing of goods, from the producer to the final consumer. Similar experiments have been conducted by Gilette and Benetton. Recently Hitachi has presented its μ-chip (see figure 2.1.2), just 4 mm^2 big and 60 microns thick. Currently the retail price for a passive RFID tag is about 0.10$ and a further reduction of the cost is anticipated for the next few years. Moreover, RFID passive tags are, in most of the cases, very simple devices with few or no intelligence on board. Nevertheless all the efforts of the producers are in the direction of reduction of cost more than in that of feature enhancing. For all these reasons, RFID technology is going to be in the next years a big player in logistic, health care, automation and many other areas. At the same time, the broad diffusion of RFID devices introduces a problem related to privacy of persons owning or carrying objects identifiable by means of tags. Solutions to these concerns are far from being trivial especially

because standard cryptographic tools used to enforce privacy cannot be employed in cheap and less powerful devices as the ones used for mass distribution.

2.3. RFID Applications

Next we discuss some popular application areas of RFID technology.

2.3.1. EPC

EPC stands for Electronic Product Code. It is proposed by EPC Global, a nonprofit organization made up of several companies and academics. It aims to standardize the use of RFID technology for inventory by establishing an *Electronic Product Code (EPC)* Network as a global standard for automatic and accurate identification of any item in the supply chain of any company, in any industry, and anywhere in the world. The EPC global Network was developed by the Auto-ID Center, an academic research project headquartered at the Massachusetts Institute of Technology (M.I.T) with labs at five leading research universities around the globe.

The Electronic Product Code (EPC) is indented to be the way of providing product identification. It was intended to standardize the way in which tag's *id*s are structured and assigned. Similarly to a bar code its goal is to identify products but it differs from printed codes as these usually identify a broad category of products (*e.g.* 1 liter milk box) while EPC links to a specific item of a product (*e.g.* 1 liter milk box, produced on July 6*th* 2006, item n. 21389432287). Like many current numbering schemes used in commerce, the EPC is divided into numbers that identify the manufacturer and product type, in addition to a supplementary set of digits which identifies each specific item. The EPC is the key to the information about the product it identifies that exists in the EPC global Network.

2.3.2. Access Control

One of the first applications of RFID technology has probably been to ski pass. Starting several years ago, skiers in many resorts have been provided with an RFID contactless card in

order to quickly gain access to ski lifts. Contactless cards are lately spreading fast in access control applications, classical contact chip-card being substituted by RFID contactless cards. Figure 2.1.7 shows the new *Camipro* card which in 2006 takes the place of the former contact chip-card, which has been in use in the past 15 years for authentication of students and personnel at the Swiss Federal Institute of Technology of Lausanne (EPFL), Switzerland.

2.3.3. Anti-counterfeit

Many products are subject to counterfeit and imitation. To reduce this phenomenon, many producers are starting to embed RFID tags in their merchandise (clothes, watches, spare parts, *etc.*). Stolen or counterfeit items can then be easily identified by RFID scanning.

2.3.4. Implantable Devices

Verichip, an American company manufacturing RFID tags, develops human implantable RFID tags. These special tags (see figure 2.1.3) are passive transponders (it would be extremely difficult to replace batteries once the tag has been implanted) and are injected under skin with a sort of special needle. The applications of these types of tags may go from access control to health care (patient identification, infant protection, *etc*.)

2.3.5. Libraries

RFID allows a fast and automatic tracing of items. This feature is particularly suited to applications as library automation. In libraries RFID are starting taking the place of barcodes. The barcodes need visual contact to be scanned and they are easily deteriorated by use. In addition they cannot perform multiple scan at the same time. On the other hand, RFID technology (see figure 2.1.1 and figure 2.1.4) allows autonomous checkouts where the patron just passing under library's batters is identified (via a contactless card) and so are the books that are identified. The system automatically checks if the patron can borrow the books and updates library's data base setting a "lent" flag.

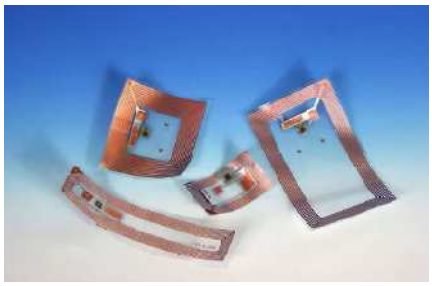


Figure 2.1.1 Misc smart labels inlay tags

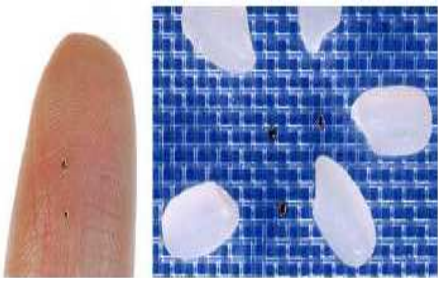


Figure 2.1.2 Hitachi produces the smallest RFID tag (just 0.33mm^2)

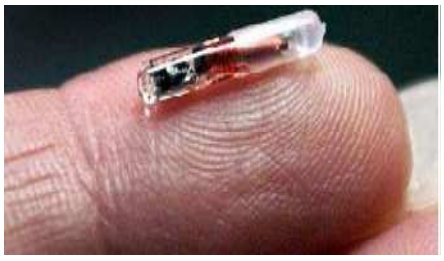


Figure 2.1.3 Human implantable tag by Verichip

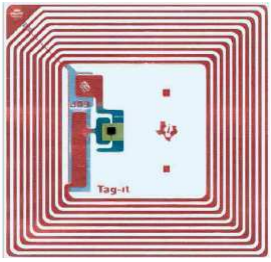


Figure 2.1.4 A square label tag

Figure 2.1.5 Baggage tracking with RFID labels

Figure 2.1.6 The new Ford keys containing RFID device to check the authenticity of the key

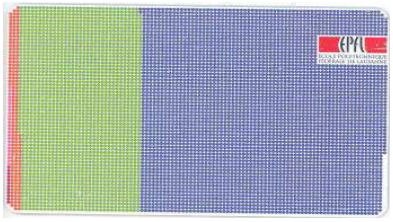


Figure 2.1.7 *Camipro* contactless card are the new cards in use at EPFL for access control

Figure 2.1.8 RFID used in Supply Chain

Source: [Bocchetti08]

Figure 2.1 Different types of RFID devices used in different RFID systems or applications

2.3.6. Supply Chain

The supply chain is a multi-stage process, which involves everything from the supplying of prime materials, used to develop products, to the products delivery to customers via

warehouses and distribution centers. Supply chains exist in service, manufacturing and retail organizations. Although, the complexity of the chain changes greatly from one industry branch to another, its management can be seen as the organization of the flows of these materials, as they move through the various processes. The efficiency of the supply chain has a direct impact on the profitability of a company. Therefore any major company striving for competitive edge needs to invest in infrastructures to control inventory, track products and manage associated finance.

By increasing transparency in the supply chain, RFID allows the optimization of logistic processes. The primary goal is the discovery of inefficiencies in the value chain within and between the companies thus rationalizing the material, information and financial flows. RFID (see figure 2.1.8) enables the fine grained tracking of the entire objects within the network, thus facilitating the detection and the locating of losses and shrinkage, the result of misplaced orders, theft and inefficient stock management.

2.3.7. Car Ignition Control

An RFID tag is embedded in the ignition key (see figure 2.1.6). When starting the car the tag in the key is used to assure of the key's genuineness. If the authentication fails the car does not start. Companies employing this technology declare that so far not even one case has been reported in which this system has been defeated. All stolen cars which employ this technology have been taken towing the car with a trailer.

2.4. RFID Systems

RFID systems are made up of three main components: RFID tag, RFID reader, and the back-end database. Figure 2.2 illustrates an example of a typical RFID system. In the following subsections, we explain the details of different components of an RFID system.

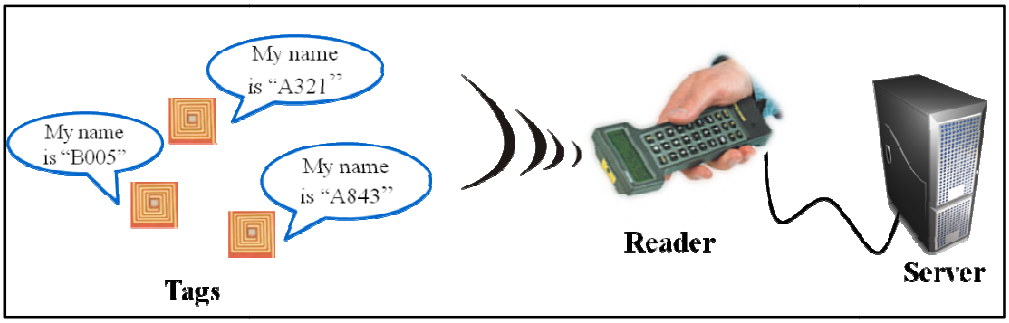


Figure 2.2 A simple RFID system

2.4.1. RFID tags or transponder

In an RFID system, each object will be labeled with a tag. Each tag contains a microchip with some computation and storage capabilities, and an antenna coil for communication. Tags can be classified according to three main criteria (see figure 2.3):

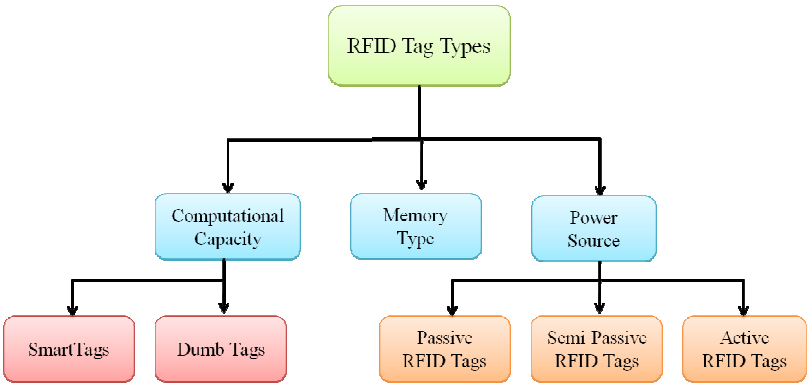


Figure 2.3 Types of RFID Tags

A) Memory Type: The memory element serves as writable and non-writable data storage. Tags can be programmed to be *read-only*, *write-once read-many*, or *fully rewritable*. Depending on the kind of tag, tag programming can take place at the manufacturing level or at the application level.

B) Power Source: A tag can obtain power from the signal received from the reader, or it can have its own internal source of power. The way the tag gets its power generally defines the category of the tag.

- ***Passive RFID tags.*** Passive tags do not have an internal source of power. They harvest their power from the reader that sends out electromagnetic waves. They are restricted in

their read/write range as they rely on RF energy from the reader for both power and communication.

- ***Semi-passive RFID tags*.** Semi-passive tags use a battery to run the microchip's circuitry but communicate by harvesting power from the reader signal.

- ***Active RFID tags.*** Active tags possess a power source that is used to run the microchip's circuitry and to broadcast a signal to the reader.

C) Computational capability: Based on the computational capacity of RFID tags, there are mainly two types [Song09] of them: dumb and smart.

- ***Dumb tags:*** A dumb tag has very low computation capacity and it has a unique identifier that is of a fixed unique length (usually 10 or 16 hexadecimal digits long) value. The memory capacity of a dumb tag is likely to be fairly small (i.e. hundred bytes to 2kBytes).

- ***Smart tags:*** Smart tags have a small processor built within it that has the capability do some cryptographic operation [Laurie07]. They usually have a larger memory capacity (32kBytes or more) compared to the dumb tags. Smart tags can perform authentication before allowing access to the stored data. Such a tag can encrypt communications to avoid some major attacks [Laurie07].

2.4.2. Constraints on the Tag

1) Tag is passive: It has no batteries. It can operate just when interrogated by a reader and only for a short time after each interrogation.

2) Tag has limited memory: Each tag has on board only a few kilobits of memory to store its *id* and its secrets. At present the majority of the tags can just save a fixed 96 bit *id.* Nevertheless we consider more sophisticated tags where some more memory is available otherwise there would be no space for any cryptographic data.

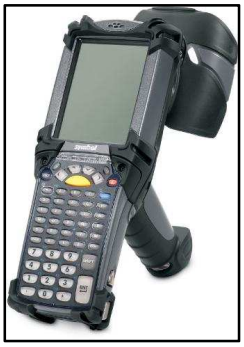
3) Tag has limited computational abilities: Each tag can perform only basic calculations, hash calculations, PRNG, AES 2. Public-key cryptography is quite expensive.

4) Tag provides no physical security: Each tag can be physically opened, thus revealing the complete contents of its memory.

5) Tag communicates at up to a fixed distance: The tag-to-reader communication is limited to a few meters but the reader-to-tag communication could be eavesdropped at a greater distance.

All these choices of tags are arbitrary and one could find tags with different Characteristics (*e.g.* more expensive). Nevertheless our choice is at present quite realistic.

2.4.3. RFID Readers or Transceiver



Source: http://www.thebarcodewarehouse.co.uk/Assets/Images/Products/16006.jpg

Figure 2.4 A simple RFID reader

RFID readers are generally composed of an RF module, a control unit, and an antenna element to interrogate electronic tags via RF communication. Readers may have better internal storage and processing capabilities, and frequently connection to backend databases. Complex computations, such as all kind of cryptographic operations, may be carried out by RFID readers, as they do not have more limitations than those found in modern handheld devices or PDAs. Figure 2.4 shows an RFID reader.

2.4.4. Constraints on the Reader

While having constraints on the tag seems quite obvious, one could think that no real concern should arise about characteristics of the readers. We should therefore explain where the concerns about the complexity of reader-side algorithms arise from.

Many RFID systems are composed by millions of tags. Think, as an example, about a big library where an RFID tag could be attached to each book. While checking out from the library the system has to recognize, before a patron crosses the door, which book he brings with him, determine if he can borrow it and update its record on a database, stating the "borrowed" status of the book. Of course all these operations have to be accomplished in a matter of fractions of a second. Having a high search complexity could lead to an unrealistic scenario where the user has to wait 30 seconds next to the reader at the library exit while the system performs its calculations. Some applications are even more time-critical. Therefore, efficient and scalable search protocols need to be installed in the reader. However, the main concern on the reader is the number of cryptographic operations to perform to identify tags.

2.4.5. Back-end server

The information provided by tags is usually an index to a back-end server (pointers, randomized *id*s, etc.). This limits the information stored in tags to only a few bits, which is a sensible choice due to severe tag limitations in processing and storing. It is generally assumed that the connection between readers and back-end databases is secure, because processing and storing constraints are not so tight in readers.

2.4.6. Constraints on the RFID System

The constraints on the two main ingredients of an RFID system (tags and readers) have already been highlighted, but still some limits on the characteristics of the whole system should be delineated.

1) Connection: Unless otherwise specified, transceivers and the back-end server are interconnected by means of a secure channel with constant infinite available bandwidth.

2) Scalability: More tags could be added to the system at any time.

2.4.7. Cryptography for RFID Systems

We make the following assumptions about the availability of cryptographic functions in simple RFID tags.

- There are sufficiently secure hash functions which are suitable for a low-cost tag.
- There is a sufficiently secure pseudo-random number generator for a low-cost tag.

2.5. RFID Standards

In any technology, lack of standards leads to inefficiencies because customers have to rely on a single equipment provider. Even the well known EPC standard is not yet fully standardized in its details. Another problem is that frequency regulations are not internationally standardized. EPC Global standardizes different categories of devices, in relation with the technical characteristics and the functionalities provided by the tag. Each class includes all the properties of the previous and adds some new. The summary of EPC class is showed in table 2.1.

Class 0: Class 0 tags are the simplest type of tags, where the data, which are usually a simple *id* number (EPC), are written into the tag only once during manufacture. No further updates are possible. These tags announce their presence when passing through an antenna field.

Table 2.1 EPC class types

Class type	Specification
Class 0	Read only tags
Class 1	Write once, read many tags
Class 1 Gen 2	Write once, read many tags, UHF Gen 2 protocol
Class 2	Rewritable tags
Class 3	Semi-passive tags
Class 4	Active tags

Class 1: Class 1 tags are manufactured with no data written into the memory. Data can either be written by the tag manufacturer or by the user, but only once. After this no further update is possible and the tag can only be read.

Class 2: Class 2 tags allow users to both, read and write data into the tag's memory. They are typically used as data loggers, and therefore contain more memory space than tags which carry only simple ID numbers.

Class 3: Class 3 tags are just like class 2 tags except that they contain on-board sensors for recording parameters like temperature and pressure, which are recorded into the tags memory. As sensor readings must be loaded into memory in absence of the reader, the tags are either semi-passive or active, thus requiring an on-board power source.

Class 4: Class 4 tags are equipped with integrated transmitters. These tags are similar to radio devices, which can communicate with other tags and devices in the absence of a reader.

Presently deployed Gen 1 UHF RFID systems are based on a number of competing protocols, most notably Matric's Class 0 and Alien Technology's Class 1. There is a problem that these protocols are proprietary. Beyond that, they lack the features, reliability and power to adequately serve a growing number of applications, particularly when taking worldwide operability into account. MIT's Auto-ID Center recognized these problems and created a single open standard that would firstly create an environment of interoperability and international regulatory compliance and secondly would raise the bar on RFID system performance in a significant way. These two values formed the backbone of the EPC Gen 2 UHF standard. With a single worldwide specification in place, UHF RFID-based systems are expected to become faster, easier to use, less costly to deploy and more robust.

2.6. Generation 2 vs Generation 1

The EPC global Class-1 Gen-2 RFID specification [Claas-1] was adopted by EPC global in 2004 and was sent to ISO. These specifications provide a great advance to consolidate the adoption of RFID technology. Where previously there were several specifications such as EPC Class-1 and EPC Class-0, a single UHF specification is now established. In order to ease a worldwide deployment, emerging UHF regulations in different regions have been taken into

account. Additionally, the best features of the preceding specifications have been improved, and a range of future applications including higher-function sensor tags have been foreseen.

2.7. Integration Costs

Currently the prices of tags are still too high for many companies to make RFID an investment. However business analysts project that the tag costs will be falling rapidly with increasing mass production. Moreover, significant investments in the infrastructure have to be made for the flourish deployment of RFID system. This includes equipment, such as terminals and networks for the collection, processing, and evaluation of the data supplied by the RFID system. Additionally the restructuring of business process and parallel operation during the initial phase are also major cost factors.

2.8. Summary

RFID makes use of radio transmission to recognize, categorize, locate and track objects. In this chapter, we discuss the components of RFID systems that are: readers, tags and a back-end database for storage and management of the collected data. The tags are attached to the products and can be read when they enter a reader's antenna field. We also discuss properties and capabilities of different categories of RFID tags. This is followed by the discussion of constraints of RFID tags, readers, back-end server, and the system. We also discuss RFID standards and the details of different types of EPC classes.

Chapter 3: Attacking RFID Systems

RFID technology is a pervasive technology, perhaps one of the most pervasive in history. However security and privacy concerns are the major drawback of this technology. One should be aware that the ways of collecting, storing and analyzing vast amounts of information about consumers existed even before the appearance of RFID technology. For example, we usually pay with credit cards, give our names and address for merchandizing, use cookies while surfing the Internet, etc.

For RFID systems a great variety of attacks can be identified. Attacks against the RFID systems opened the door for the development of both classical and modern security techniques, ranging from signal jamming to challenge-response identification. And it is just as likely that RFID will continue to inspire progress in security and privacy research in the future, as it has done for decades.

The major goal of this chapter is to give an overview of the primary security requirements of RFID systems and the traditional mechanisms to fulfill those requirements. Another objective is to categorize the existing weaknesses of RFID systems so that a better understanding of RFID attacks can be achieved.

3.1. Attack Objectives

In an RFID system the objectives of each attack can be very different. It is important to identify the potential targets in order to understand all the possible attacks. The target can be the complete system (i.e. disrupt the whole of a business system) or only a section of the entire system (i.e. a particular item). A great number of information systems focus solely on protecting the transmitted data. However, when designing RFID systems, additional objectives, such as tracking or data manipulation should be considered. Let us imagine the following example in a store: an attacker modifies the tag content of an item reducing its price from 100 to 9.90 ¤. This

leads to a huge loss for the store. In this scenario, the data may be transmitted in a secure form and the database has not been manipulated. However, attack is carried out because part of the system has been modified. Therefore, in order to make a system secure, all of its components should be considered. Neglecting one component, whatever the security level of the remaining components, could compromise the security of the whole system. As shown in the above example, the attack may be perpetrated to steal or reduce the price of a single item, while other attacks could aim to prevent all sales at a store. An attacker may introduce corrupt information in the database to render it inoperative. Some attacks, such as the active jamming attack, are inherent in the wireless technology employed. Other attacks focus on eliminating physical access control, and ignore the data. Some involve identity stealing from legitimate e-passports, and etc.

3.2. Security Requirements

RFID technology may bring spontaneous risks because of the proliferation of RFID tags. Certain security requirements must be addressed by every RFID protocol to maintain the security and privacy of the overall RFID system. Number of research literatures [Ahamed08c, Avoine05, Bringer06, Cai09, Chien07, Choi04, Conti07, Cui07, Gilbert05, Henrici04, Hoque09, Hopper00, Hopper01, Juels05a, Juels05b, Juels05c, Juels06, Lee10, Molnar04, and Ohkubo03] deals with several privacy and security issues of RFID. Therefore, we try to point out the security goals that should be guaranteed by a protocol:

- ***Privacy protection*:** A tag cannot be distinguished by an adversary without tampering it and realizing the data stored in the tag.

- ***Anti-tracking:*** It is tough for an adversary to track a tag if the adversary does not have any information about the tag. But the attacker can track a tag, if the tag replies with a constant response each time it is queried. So protocols should be designed such that a tag neither reveals its id nor replies with constant response.

- ***Anti-cloning:*** In order to clone a tag, an adversary needs to know the secret key shared between a tag and the authorized reader. So, to be secured against cloning attack, protocols should never reveal the shared secret key.

- ***Synchronization*****:** Attacker should not able to update the key used by the tag or the reader to secure the communication.

- ***DoS resiliency*****:** *Denial-of-Service* (DoS) attack means an authorized entity is prevented from accessing its authorized entities. In order to ensure successful communication between a reader and its authorized tags, it should be guaranteed that an adversary cannot desynchronize them.

- ***Not susceptible to replay attack:*** Security must be ensured against replay attacks so that an adversary cannot impersonate a legitimate tag by replaying an eavesdropped message.

- ***Forward secrecy*****:** An adversary compromising a tag will not be able to identify the previous outputs of the tag.

- ***Backward secrecy:*** An adversary compromising a tag will be unable to track future transactions even if it has access to the tag's present internal state.

3.3. Adversary Types

The adversary can be categorized into the following classes:

- ***Weak adversary:*** This type of adversary cannot corrupt any tags.

- ***Strong adversary:*** This type of adversary has no limitations on corrupting tags, and can do anything at its wish. For each category of adversary defined above, it is also defined a *narrow* variant, where a narrow adversary cannot access the outputs of the players (i.e., reader and tags) for any protocol run.

- ***Forward adversary:*** This type of adversary can corrupt tags under the limitation that once the adversary corrupts a tag, it can do nothing subsequently except for corrupting more tags.

- ***Destructive adversary:*** This type of adversary can do anything after a tag corruption, but under the limitation that the adversary cannot reuse a tag after corrupting it. Specifically, once a tag is corrupted it will be virtually destroyed. In particular, a destructive adversary cannot observe or interact with a corrupted tag nor can the adversary impersonate a corrupted tag to the reader.

3.4. Classification of Different Attacks

This upcoming section discusses the major classes of attacks that are usually launched against RFID systems.

3.4.1. Modification of data

This type of attack deals with the alteration of data saved within the memory of the tags. By unauthorized write access, the data stored on the tag can be modified. This attack is only effective if the identifier and security information such as keys remain unchanged. Otherwise this attack leads to denial-of-service. The attack is only possible if additional data along with the identifier are stored.

3.4.2. Deactivation of tags

In this type of attack, the tag is made inoperative by executing a dedicated command or by physical intervention. Depending on the degree of deactivation the identity or the presence of the tag can no longer be determined.

3.4.3. Active jamming

Although passive interference is usually unintentional, an attacker can take advantage of the fact that an RFID tag listens indiscriminately to all radio signals in its range. Thus, an adversary may cause electromagnetic jamming by creating a signal in the same range as the reader in order to prevent tags from communicating with readers.

3.4.4. Sniffing or tracking

RFID tags are designed to be readable by any compliant reader. Unfortunately, this allows unauthorized readers to scan tagged items, oftentimes from great distances. This type is attack is called *sniffing* or *tracking* and this is one of the major attacks launched in most of the RFID systems. This type of attack can also be launched by eavesdropping on the wireless channel between the tag and the reader. Tracking of RFID tags allows monitoring of individuals' whereabouts and actions. RFID readers placed in strategic locations (like 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 collections which are recurring groups of tags that are associated with an individual. In such cases, RFID technology also enables the monitoring of entire groups of people. Moreover, tracking attack will also lead to unrestricted access to tag data or tagged object's information. Unrestricted access to tag data can have serious implications and collected tag data might reveal information like medical predispositions or unusual personal inclinations, which could cause denial of insurance coverage or employment for an individual.

3.4.5. Spoofing or cloning

In this type of attack, the attackers can create 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.

3.4.6. Replay attack

Replay devices are capable of intercepting and retransmitting RFID queries, which could be used to abuse a variety of RFID applications. These types of attacks usually occur in situations where RFID components use a challenge response based protocol. RFID tags and readers usually

share a secret and use a challenge response protocol to authenticate their identities. Nevertheless, very often this approach is subject to replay attacks. In a replay attack, an adversary broadcasts a tag's response recorded from a past transaction in order to impersonate the tag to a reader. Typical example of this attack is the unauthorized access to restricted areas by broadcasting an exact replay of the radio signal sent from a legitimate tag to the reader that grants access.

3.4.7. Relay attack

In a relay attack an adversary acts as a man-in-the-middle. An adversarial device is placed surreptitiously between a legitimate RFID tag and reader. This device is able to intercept and modify the radio signal between the legitimate tag and reader. Subsequently, a momentary connection is relayed from the legitimate tag/reader through the adversarial device to the legitimate reader/tag. The legitimate tag and reader are fooled into thinking that they are communicating directly with each other. To make this type of attack even more sophisticated, separate devices could be used, one for the communication with the reader and one for the communication with the RFID tag.

A number of factors combine to make relay attacks on RFID technology. Tags are read over a distance and activated automatically when close to a reader. This allows an attacker to communicate with a tag without the knowledge of its owner. Two devices, as shown in figure 3.1, are involved in the relay attack: the ghost and the leech [Czeskis08]. The ghost is a device which fakes a card to the reader, and the leech is a device which fakes a reader to the card. A fast communication channel between the legitimate reader and the victim card is created by the ghost and the leech:

1. The legitimate reader sends a message to the ghost

2. The ghost receives it and forwards this message to the leech through the fast communication channel

3. The leech fakes the real reader, and sends the message to the legitimate tag

4. The legitimate tag computes a new message and transmits it to the leech

5. The leech receives it and forwards this message to the ghost through the fast communication channel

6. The ghost forwards this message to the real reader

This sort of attack dispels the assumption that readers and tags should be very close to communicate. Additionally, even if communications were encrypted, the attack is feasible because messages are only relayed through a fast communication channel, without requiring knowledge of their contents.

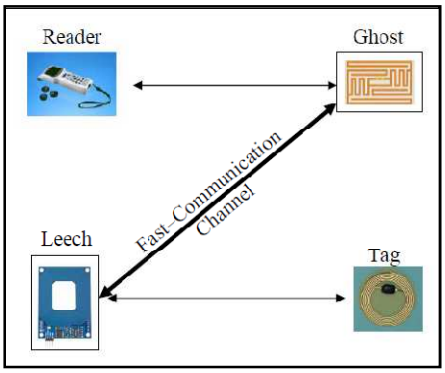


Figure 3.1 Example of relay attack (ghost and leech attack)

3.4.8. Denial-of-Service (DoS)

This is a type of attack in which an attacker causes RFID tags to reach to such a state from which they can no longer function properly. This results in the tags becoming either temporarily or permanently out of operation. More precisely, in this attack a tag is attacked with queries from an illegitimate reader. As a result, that tag is not able to respond to a further query from the legitimate reader. In other words, a genuine reader cannot communicate with its legitimate tags. A similar attack is also possible on the reader, but since the tag is much more resource constrained than the reader, they are more susceptible to such attacks than the readers. Such attacks are often intensified by the mobile nature of the tags, allowing them to be manipulated at a distance by covert readers. This type of attack can be a serious threat to the integrity of automated inventory and shipping applications.

3.4.9. Server impersonation attacks

Server impersonation means that an adversary is able to impersonate a valid server to a tag. One reason that this is a genuine threat is because de-synchronization can occur if a tag updates its stored data when the server does not. More specifically, an attacker that compromises a tag's stored secrets can impersonate an authorized server to the tag. If the attacker executes an authentication session with the tag, impersonating a valid server, then it can make the tag to update its stored secrets, although the genuine server does not update the secret corresponding to the tag entry. Then the tag and the real server can be desynchronized.

3.4.10. Eavesdropping attack

As RFID technology operates through radio channel, so communication can be covertly overheard. In eavesdropping an unauthorized individual uses an antenna in order to record communications between legitimate RFID tags and readers. In this type of attack, the communication between tag and reader over the air interface is intercepted, decoded and interpreted. A passive adversary can eavesdrop on messages between a reader and a tag and can keep records of the messages. The information recorded can be used to perform more sophisticated attacks later. The feasibility of this attack depends on many factors, such as the distance of the attacker from the legitimate RFID devices.

There are two possible distances at which an attacker can listen to the messages exchanged between a tag and a reader. They are:

Forward Channel Eavesdropping Range: In the reader-to-tag channel (forward channel) the reader broadcasts a strong signal, allowing its monitoring from a long distance.

Backward Channel Eavesdropping Range: The signal transmitted in the tag-to-reader (backward channel) is relatively weak, and may only be monitored in close proximity to the tag.

3.5. Attack Intentions

Table 3.1 Intentions behind attacks in RFID systems

	Privacy Protection	Access of data	Denial of service	Spoofing
Modification of data				
Tag Spoofing				
Deactivation of tags				
Removal of tags				
Eavesdropping				
Jamming				
Reader Spoofing				

Table 3.1 various intentions that an adversary might have while attacking an RFID system. An attacker may want to access sensitive information or exploit an RFID system by spoofing an RFID tag. An attacker's intention might be to make an RFID system unavailable (DoS attack). Even a user might launch an attack because he feels his right for privacy is violated.

3.6. Summary

Although RFID networks have many advantages, they also present a number of inherent vulnerabilities with serious potential security implications. In this chapter, we analyzed the security issues that arise with RFID. Firstly a discussion of the attack objectives of an adversary in an RFID system is given. Then the security requirements of RFID systems are pointed out. After that, some major possible attacks are identified and discussed. Finally, attack intentions of an RFID system attacker are identified.

Chapter 4: Related Work

There are several attacks in RFID systems that are obstacles to make RFID more popular, and widespread then before. However, researchers have been working for long time to prevent those attacks in RFID systems and to facilitate the expansion of RFID technology. One key research area that focuses on securing RFID systems against major attacks is to design secure authentication methodologies. These authentication techniques are designed to execute while a reader communicates with an RFID tag for identification purpose.

One extension of RFID tag authentication is known as tag searching. Tag searching means searching for an RFID tag from a large collection of tags. Any RFID authentication protocol which provides security and privacy can be used for this purpose. However, as the number of RFID tags increases, the cost of collecting data can be very high. More efficient methods for performing RFID tag search are needed. Search is a basic and invaluable tool for sifting through large amounts of data. Consider for example, a large pharmacy stocked with RFID embedded medication. A pharmacist wanting to find a particular drug can broadcast his query and receive an answer. Due to the limited broadcast range of RFID readers, the pharmacist can even determine the approximate locality of the medication by directing the RFID reader at different locations, i.e., towards different shelves.

Though RFID tag searching is an important issue for most RFID systems, the assortment of research literature on RFID searching is inadequate. Therefore, the goal of this chapter is to discuss some famous authentication techniques along with the proposed search protocols so far.

4.1. Authentication Related Prior Works

In this section we present some classic identification/authentication protocols for RFID.

RFID security based research area can be divided into two categories. The first category is protocol based. This category mainly focuses on implementing protocols using secure,

lightweight primitives on small RFID tags in order to ensure security and privacy. The second category is hardware based and this category focuses on improving RFID tag hardware so that it can provide additional security primitives. All of our proposed protocols in this thesis fall in the first category. So we will not discuss about the hardware based category. However, interested readers can refer to [Juels05b] and [Rieback07] for more details. In this section, we will mainly discuss the research background related to the protocols based category. Within the area of the protocol based category numbers of techniques have been proposed for ensuring RFID security and the assortment of authentication protocols is quite extensive. Thus we shall avoid a broad review and focus on those works that are related to our contribution. Interested readers may refer to [Juels05b] and [Juels06].

- ***The Weis-Sarma-Rivest-Engels Protocol:*** Weis et al. [Weis03] proposed authentication protocol which used back-end database to perform the authentication. In this protocol, an RFID tag replies with a $metaID$ when it is queried by a reader. The reader forwards this $metaID$ to the back-end database which finds out the real ID of the tag for the reader. An RFID tag replies with the same $metaID$ each time it is queried by a reader. So this protocol is not secured against tracking attack which hampers privacy of the tag holder. So the authors proposed a randomized hash lock scheme to solve this problem. In this scheme, a tag replies with $(r, ID \oplus f_k(r))$, when it is queried by a reader. Here, k is the tag's secret, f_k is a pseudorandom function and r is a random number generated by the tag. The reader forwards this reply to the secure database which then searches for the ID/tag secret key pair that matches with the reply. Under this scheme, an RFID tag replies with a different value each time it is queried by a reader as each reply of the tag involves a random number.

- ***The Tsudik Protocol***: Tsudik proposed a protocol, YA-TRAP, in [Tsudik06] that ensures high efficiency at the server side. It is a famous authentication protocol that places little burden on the back-end server. The principle advantage of this protocol is that the central

database avoids any real time processing. Authors proposed that YA-TRAP is really advantageous in situations where tag information is processed in batches rather than in real time. The fundamental idea of this protocol is based on monotonically increasing timestamp which makes this protocol secured against tracking. But the use of the timestamp makes this protocol unsecured against DOS attack. In this protocol, an RFID tag update its timestamp based on a value provided by the reader. At the same time each tag stores T_{max} , where T_{max} is the maximum value that can be reached by the timestamp. When the timestamp reaches T_{max} a tag does not answer to the reader's queries. Hence an adversary can send the tag a large enough timestamp so that it goes beyond T_{max}. Thus it becomes quite easy for a malicious reader to create DOS attack. Although the solution to DOS was proposed in YA-TRAP+ [Avoine05], this protocol still lacks forward secrecy.

- ***The Ohkubo-Suzki-Kinoshita Protocol:*** Another lightweight protocol is OSK [Ohkubo03]. Ohkubo, Suzuki and Kinoshita proposed that two hash function H and G are sufficient to provide indistinguishability and forward secrecy. Here, H is a one way hash function and G has random oracle. According to this protocol, a tag is initialized with a shared secret s_1 and the back-end server maintains a list of tags (id, s_i). The tag updates its secret key after each query according to the following formula $s_{i+1} = H(s_i)$. And in response to the query from a reader, the tag replies $a_i = G(s_i)$. The server on the other hand uses a_i to identify the tag by performing a brute force search through the list of tags. OSK does not ensure scalability. In [Avoine05], Avoine and Oechslin modified OSK which removed the scalability problem. They introduced a time-memory tradeoff which reduced the computational complexity for inverting the hash function. Another problem of OSK is that a malicious reader may easily desynchronize a tag which eventually results in DOS attack.

- ***The Henrici-MÄuller Protocol:*** In [Henrici04], Henrici and MÄuller relies one-way hash function to thwart tag tracking attacks. In this solution, a tag responses a reader's query with

two hash values and updates its stored values after a successful authentication. This solution does not provide full-degree of anti-tracking since a tag always replies with the same response before it is successfully authenticated. In addition, it does not provide forward security as a strong adversary could derive tag identifiers in previous sessions from the tag's current identifier and the server's random number.

- ***The Molnar-Wagner Protocol:*** Molnar and Wagner [Molnar04] pointed out that the randomized hash lock scheme does not defend against an eavesdropper. An adversary can eavesdrop on the communication between reader and tag to learn the tag replies. The adversary then uses this information to impersonate the RFID tag to fool a reader. In this protocol, both the reader and tag share a secret (x). Both reader and tag generate random nonces (r_a, r_b) and share them. By refreshing the random nonces during every instantiation of the protocol, replay attacks through eavesdropping are avoided.

- ***The Hopper-Blum Protocols:*** Hopper and Blum propose a secure human authentication protocol in [Hopper00 and Hopper01]. Here, $r_A \cdot x$ and $r_A \oplus x$ represent scalar product and exclusive-or (XOR) of k-bit binary vectors r_A and x respectively. The HB protocol relies on the computational hardness of Learning Parity with Noise (LPN) problem. It is meant only to be secure against passive attacks, and it is not secure against active attacks. A simple active attack, where an adversary pretending to be the reader, transmits a fixed r_A to the tag several times can retrieve the value of x. While humans may get suspicious with repeated, failed login attempts if they are actively queried by a computer, a simple tag will blindly reply to active queries. In other words, HB would not protect against skimming attacks.

- ***The HB+ Protocol:*** An alternative method for RFID authentication is based on a "challenge and response" between a reader and a tag. Juels et. al. [Juels05a] observed that human authentication protocols can be applied to RFID, since RFID tags, like humans, have weak computational capabilities. They introduced HB protocol, in which a reader issues a new

challenge to a tag each time it queries an RFID tag. The tag computes the binary inner product based on the reader's challenge, and returns the answer to the reader. The reader authenticates the tag by verifying the tag response. The HB+ protocol is an improvement over the HB protocol by using an additional binding factor from the tag to defend against an active adversary. Later work by [Piramuthu06], [Gilbert05], [Bringer06] improves on this idea.

- ***The Seo-Kim Protocol:*** Seo et al. [Seo06] proposed a hash function based authentication protocol that ensures high scalability. This protocol is also untraceable. Here back-end server $\mathcal{B}$ has the following four fields associated with each tag: EPC, $h(ID_i)$, ID_i and the access PIN. Each tag saves the last timestamp TS send by an authorized $\mathcal{R}$ as TS_{last}. Based on its own timestamp TS and shared secret key k, reader computes $h(k, TS)$ and transmits it to the tag $\mathcal{T}_k$ together with TS. Tag recognizes an authorized reader if TS received from the reader is greater than TS_{last} and replies with $h(ID_i)$. Reader $\mathcal{R}$ forwards $h(ID_i)$ and TS to $\mathcal{B}$ and here the back-end server comes into play. It updates the ID of corresponding tag and asks the reader to pass on the message to the tag for synchronization. Upon reception of the message, tag $\mathcal{T}_k$ updates its ID and TS_{last}. The most significant contribution of this paper is scalability and forward secrecy. Updating ID with a one way hash function ensures forward secrecy. Scalability is ensured in a sense that back-end server needs time complexity $O(\beta)$ to find a tag in multi tag environment where β is the number of tags that have same key k within the operating range of a reader. The drawback of this protocol is that ownership transfer requires external intervention.

- ***The Seo-Lee-Kim Protocol:*** Seo et al. proposed another authentication protocol [Seo06b] that ensures high scalability and ownership transfer. It is a lightweight authentication protocol that employs a proxy in addition to the back-end server. The protocol is based on Universal Re-encryption which allows the back-end server to get the tag identifier only after a simple decryption. This decryption requires a constant time which makes it one of the highest scalable authentication protocol. But its application area is restricted because of the use of proxy.

This protocol is best suited for personal use. But it suffers from the problem of traceability and some other security issues such as DOS attack and swapping.

- ***The Tan-Sheng-Lee Protocol:*** In [Tan07], Chiu et al. proposed a serverless authentication protocol. In this protocol reader maintains an access list L_i which is used for tag authentication purpose. And each tag has a secret t which is not shared with anyone. Reader and tag both know $f(r,t)$, where r is reader identifier. Here in response to the query from a reader, tag replies with some of the bits of $h(f(r,t) \parallel n_i \parallel n_j))$ where n_i and n_j are two random numbers generated by the reader and the tag respectively and $h(.)$ is a one way hash function. Since only a legitimate tag can generate $h(f(r,t) \parallel n_i \parallel n_j))$, it works as tag's certificate to the reader. At the same time tag queries reader with a question string. Only a legitimate reader replies with valid answer string which introduces the reader as an authorized reader to the tag. Tag releases its data only after realizing that the reader is legitimate. But here again the reader has to do a lot of computation to find out id of the required tag. But their protocol 2 is not purely and strongly anonymous as they return tag id by performing XOR operation with hash value for authentication. Moreover, they didn't propose any technique for ownership transfer.

- ***The Chien-Chen Protocol:*** In [Chien07], Chien and Chen used a challenge-response protocol to prevent replay attacks. To prevent denial of service attacks, both new key and old key for authenticating a tag are stored in back-end database. However, a strong adversary can still identify a tag's fixed EPC code, thus identify the tag's past and future interactions after compromising a tag.

4.2. Search Related Prior Works

Tag searching is different than tag authentication. Though a single tag can be searched using a secure authentication protocol, it will decrease the performance and response time of the overall RFID system. There have not been many attempts to produce a secure search protocol for RFID systems. RFID search protocols have not gathered much attention so far but research

literature in this area is also in an emerging state. In this section, we will explain different search protocols that are proposed in research literatures till now.

- ***Hash based Serverless Search Protocols:*** Serverless RFID searching protocols were also proposed in [Tan07] for the first time. The authors produced a series of search solutions that require very little storage, and can be distributed without an explicit need for a back-end server. Their solutions base themselves on the RFID tag's ability to perform hash computations. According to this protocol, a reader wishes to find out whether a specific tag is within its vicinity by broadcasting $h(f(r_i, t_j)||n_r) \oplus id_j$, n_r and r_i . Based on this search query, only the intended tag, if exists, reply with its encrypted id. Other tags within the reader's vicinity reply a random number based on certain probability. Tags authenticate the reader based on the search query and reader authenticates tags based on the reply "string". Both valid query and valid replies are generated by legitimate parties. In their protocol they used to use noise to mask the tag replies. Each tag receiving a search query that does not match the request replies with some probability. This technique facilitates the protocol to be secured against some major attacks, such as tracking, or physically determining a tag's location.

- ***Lightweight Secure Search Protocols:*** Lars et al. propose a lightweight secure search protocol in [Kulseng09]. The authors proposed three lightweight secure search protocols, all of which can prevent the adversary from learning the identity of tags or impersonating tags. In the basic protocol, the target tag responds to any query, so an adversary may replay any previous query and know the presence of a target tag. Their synchronization-based protocol mitigates the impact of replaying attacks by reducing the number of queries that a target tag should respond. Their best protocol is the multi-response protocol from which the adversary learns nothing about the target tag. Their protocols are built on top of Linear Feedback Shift Registers (LFSR) and Physically Unclonable Functions (PUF), which are very efficient for implementation in low-cost tags. The authors use LFSR to generate random numbers for encrypting communication and reply

on PUF to authenticate the identity of tags. The author also performed evaluation of their protocol and the experimental results show that their solutions have negligible processing time and require no more than 1400 hardware gates. So, they are very suitable for low-cost RFID systems with at most 2000 gates available for security purposes.

- ***Lee et al.'s Search Protocol:*** Lee et al. proposed a novel search protocol [Cai09] which allows for privately querying a particular tag. In their protocol, the server (or a reader) can efficiently query for a specific tag, without compromising the tag's privacy. The authors first designed a two-round protocol and reduce it to a one-round protocol. In order to reduce it to a one-round protocol, they change the protocol such that the server generates a challenge instead of receiving it from a tag. According to their protocol, in order to prevent replay attack, each tag is allowed to keep a counter and update it each time a valid message is received. As a result, the received counter is always bigger than the stored one. After verifying the message from the server, a tag can respond to the server. Only the server in their system can generate valid messages. After the search protocol is executed, in order to make sure that the proper tag is responding to the server, a tag-to-server authentication protocol is invoked. The search protocol itself (without combining it with an authentication protocol) requires the server and a tag to perform two EC point multiplications each. The authors proved the security properties of the proposed search protocol. The performance results of their experiment show the feasibility of the proposed protocols, even for a passive tag. According to the authors, their protocol outperform other privacy-preserving protocols.

4.3. Summary

In this section, we have reviewed a number of recently proposed RFID authentication and search protocols. We have also assessed their security and performance properties against the requirements identified in chapter 3.

Chapter 5: A Secure Serverless Search Protocol (S^3PR)

5.1. Introduction

Usually an RFID system is composed of three main components: tag, reader and back-end database. Every tag carries an object identifying data. When a tag receives a query from a reader, the tag transmits information to the reader using RF signals. The RFID reader reads and sometimes re-writes the stored data in a tag. After a reader queries a tag and receives information from the tag, the reader forwards the information to a Back-end server. The back-end server is powerful in computational capacity and manages lots of information related to each tag. Actually in server based system, back-end server plays an essential role and it is quite easy to check validity of tags or reader, which is very important for privacy protection and security issues.

But the major drawback of the central server based system is that the readers always have to be connected to the server, which limits usage of RFID systems in remote locations where connectivity with server cannot be ensured. Besides, having a single database makes the whole system more vulnerable to privacy attacks. Central server has knowledge of all the tag secrets and tag information. Therefore, if the database is collapsed by an adversary, the entire user community's privacy is jeopardized.

An alternative, analogous to using central database, is to store all information of the central server in the reader. Because of the mobile nature of readers, they can be stolen. An adversary with a stolen reader will have access to the information found in the central database and the stolen reader can be easily compromised. The compromised reader may hold *id* and tag secret pair that can be loaded by an adversary into a blank tag. This fake tag can impersonate a legitimate tag and a reader cannot distinguish between the two. This is a severe breach in the security of an RFID system.

Security and privacy protection is a major issue in another situation where a single reader and multiple tags are present. In all such practical situation, often a reader needs to determine

whether a particular tag exists within a group of tags. This is referred to as RFID searching. Tag searching with the help of a central database is not a challenging issue. However, without the help of the server, the reader has to search a tag entirely by itself. This type of tag searching is a critical task as it is vulnerable to privacy and security threats. For example, through the broadcast of a search query, a reader in a warehouse wants to search for a tag which belongs to a precious object. Now if the tag exists, it will reply and an adversary will become sure that a valuable object exists around it.

RFID tag searching can be thought as an extension of RFID authentication. By authenticating every tag within a group, we can find out the desired tag. As the number of tags increase, the ability to search RFID tags becomes invaluable when the reader requires data from a few RFID tags rather than all the tags in the collection. If the reader has to authenticate each tag one at a time then the entire searching process will become very time consuming. Though tag searching is very useful in many RFID applications, secure searching methods have not received enough attention in research literatures so far. We firmly believe that in near future tag searching will be a significant issue in RFID based pervasive systems.

In this chapter, we try to find solutions to the following questions: a) *how can the readers search for a particular tag without the help of the server?* b) *how does a tag identify that the communicating reader is legitimate?* Here, we propose a low cost, secured, serverless search protocol that provides solutions to the preceding questions. All these characteristics are ensured without a back end server which makes our proposal suitable for various application areas. A version of this proposal has been published in [Ahamed08b].

In serverless system, a reader has to search, authenticate as well as provide security without the server's intervention. This departure from a server based system may also reduce the cost for RFID system deployment in many areas where tag searching is done frequently, like inventory management, retail store product managements, supply chain management, E-passport, etc.

5.2. Existing Trivial Solutions

Back-end database played an essential role in most early works on RFID security. Researchers came up with highly secure protocols but authentication was done mostly by the back-end server rather than the reader itself.

Serverless RFID searching protocols were proposed in [Tan07] for the first time. According to this protocol, a reader wishes to find out whether a specific tag is within its vicinity by broadcasting $h(f(r_i, t_j)||n_r) \oplus id_j$, n_r and r_i . Based on this search query, only the intended tag, if exists, reply with its encrypted id. Other tags within the reader's vicinity reply a random number based on certain probability. Tags authenticate the reader based on the search query and reader authenticates tags based on the reply "string". Both valid query and valid replies are generated by legitimate parties. But here the reader has to do a lot of computation and their protocols are not strongly anonymous as they return tag id by performing XOR operation with hash value for authentication.

5.3. Proposed Solution

Our major contributions in this chapter are as follows:

- We propose serverless, forward secure, anonymous and secure searching protocols for RFID tags. Our protocol makes use of the simple Pseudo random Number Generator (PRNG) and hash function to ensure security.
- According to the protocols, the tag identifier is not passed to the reader in response to a reader's query. Here, the tag sends certifying information to the reader in such a way that only the authorized reader is able to find out whether this is the desired tag. One unique feature of our protocol is that it is not vulnerable to single point-of-failure.
- We consider security of both tags and readers as both can be attacked by adversaries. We consider all the major attacks and our search protocols are secure against those attacks.

5.3.1. System architecture

An RFID system usually consists of three main components: a reader, a tag, and a back-end database/server. The communication channel between the reader and the tag is wireless, while that between the reader and the database can be either wired or wireless. The tag presents its identification number or other stored information to the reader upon request. The reader will then communicate with the database. We assume that the communication between the reader and the database is secure due to the use of some kind of standard encryption technique. We further assume that an adversary can hear all transactions between a reader and a tag.

However, our RFID system is a serverless system. Therefore, our serverless RFID system mainly consist two parties, one of them is the reader *R* and the other is a set of tags. A certification authority *CA* is involved in the system to certify readers and authorize them to particular tags. In this protocol, we focus on passive tags, which are low-cost and resource-constrained. For example, the most popular passive tag, EPC Class 1 Gen 2, has at most 2000 hardware gates available for security features [Juels05a].

5.3.2. Preliminaries

All readers and tags have knowledge of a pseudorandom number generator $\mathcal{P}(.)$ and a function $\mathcal{M}(.)$. $\mathcal{P}(.)$ is a fairly simple random number generator that can be implemented at low cost. $\mathcal{P}(.)$ takes a seed as an argument and outputs a pseudorandom number according to its distribution. $\mathcal{M}(.)$ is used by all readers and tags to update the seed of the pseudorandom number generator by passing the current seed as input. We assume $\mathcal{M}(.)$ as an irreversible one way hash function. Therefore a current seed cannot be linked to its previous one.

We refer an RFID reader as R. Each R has a unique identifier r and a contact list $\mathcal{L}$. We will describe the contents of $\mathcal{L}$ later. R obtains r and $\mathcal{L}$ from a certification authority, CA, after authenticating itself. The CA is a trusted party who deploys all the RFID tags and authorizes any

RFID reader. For the sake of simplicity we assume that R and CA communicate through a secure channel. Each tag T contains a unique id and a unique secret t in its nonvolatile memory.

Table 5.1 Summary of notations for S^{3}PR Protocols

Symbol	Meaning
R_i	RFID reader i which wishes to execute search
$T_{desired}$	Desired RFID tag that the reader is searching for
$seed_{desired}$	seed residing in the contact list of $R_i$for the RFID tag $T_{desired}$
$seed_{T_{desired}}$	seed residing in the RFID tag $T_{desired}$
$n_{desired}$	Pseudorandom number generated by the reader R_i for tag $T_{desired}$, based on $seed_{desired}$
$T*$	All tags within the vicinity of the reader R_i
$seed_{T*}$	seed residing in each tag that is within the vicinity of the reader R_i

Subscripts are used to describe a particular R or T and their respective variables. Thus a particular RFID reader i will be R_i with an identifier r_i and contact list $\mathcal{L}_i$ stored in its nonvolatile memory. An RFID tag j is T_j having a secret t_j. The contact list $\mathcal{L}_i$ contains information about the tags which R_i has access to. The information about each tag comprises a seed and the id of the tag. If R_i is authorized to access tags $T_1,\cdots,T_n$, $\mathcal{L}_i$ will take the following shape after authenticating itself to CA,

$$\mathcal{L}_i = \begin{Bmatrix} seed_1 : id_1 \\ \cdots \quad : \cdots \\ seed_n : id_n \end{Bmatrix}$$

where, for any tag T_j and $1 \le j \le n$, $seed_j$ is a seed used by R_i to communicate with T_j and id_j is T_j's identifier. $seed_j$ is initialized by $seed_j = f(r_i, t_j) = h(r_i \parallel t_j)$ where $h(.)$ is a one way hash function and $\parallel$ represents concatenate. Note that R_i does not know the tag secret t_j. R_i only knows the outcome of the function $f(r_i, t_j)$ as $seed_j$. The initial $seed_j$ is computed by CA and stored in R_i. The tag T_j will contain only one seed for its only one authorized reader R_i. While T_j is deployed by CA, T_j will get $f(r_i, t_j) = h(r_i \parallel t_j)$ as $seed_{T_j}$ from CA. T_j stores $seed_{T_j}$

in its nonvolatile memory. We also assume that CA cannot be compromised. We denote an adversary as ϱ. The notations for serverless search protocols are summarized in Table 5.1.

5.3.3. Attack model

RFID systems face many threats launched by attackers. Attackers can be either active or passive. Passive attackers mainly launch eavesdropping attacks to capture the messages transmitted between the reader and the tag. They intend to learn some secret or private information about the communicating parties. This information can then be used for the purpose of tracking or finding secrets in other messages by utilizing bit manipulation or other offline methods. The active attackers can jam wireless communication, send out bogus messages, or compromise some tags. In our protocol, we focus on the majority of attacks launched by the active attackers.

The major goal of an adversary in any RFID system is to counterfeit a real tag such that it has a small probability of being distinguished from the real one. Evidently, the fake tag embedded within the fake product can let the product to be identified as a legitimate one.

For our serverless protocol, we denote an adversary as $\tilde{A}$. The adversary can control a number of readers and tags. The reader and the tag controlled by the adversary is denoted as $\tilde{R}$ and $\tilde{T}$, respectively. $\tilde{R}$ is unauthorized to have access to any real tags as it is not connected with the backend server. Similarly, $\tilde{T}$ is not valid as it has no idea about *S* and *ID*. We assume that the backend server cannot be compromised. Moreover we assume that all the entities such as tags, readers, adversaries, adversarial tags and adversarial readers have polynomially bounded resources.

We assume that $\tilde{A}$ is more powerful than a passive attacker. Like a passive attacker, $\tilde{A}$ can eavesdrop on the channel between a valid reader and a valid tag. However, like an active attacker, $\tilde{A}$ can install a rouge reader $\tilde{R}$ that can communicate with a valid tag. In addition, $\tilde{A}$ can install a fake tag $\tilde{T}$ to communicate with a legitimate reader. In both cases the ultimate goal of the

adversary is to counterfeit a tag with the learned information. In spite of these attacks, $\tilde{A}$ can launch hardware based physical attacks. A successful hardware based physical attack can give adversaries the ability to create fake tags, or impersonate a legitimate tag using some other device. But we will not study such attacks as hardware based physical attacks are beyond the scope of this paper.

5.3.4. Search Protocols

Intuitively, to satisfy the two properties of secure search protocol, we need to encrypt both query and response in order to prevent an eavesdropper from learning the identity of the target tag. Meanwhile, the messages should be changed for each search in order to prevent an adversary from replaying them. Based on these ideas, we design several secure search protocols. Each of our protocols consists of two phases, a preliminary setup phase and an online search phase. In the setup phase, the reader and all the tags are preloaded with some secrets. Then, in the search phase, the reader and the tag exchange their secrets for the reader to detect the presence of the target tag. Next, we discuss the detail of the online search phase. Suppose, a reader R_i is searching for a tag denoted as $T_{desired}$. One way of searching may be according to our *Search Protocol 1* which we name as *Simple Search Protocol* (see figure 5.1).

Search Protocol 1: **Simple Search Protocol**

(1) $R_i \rightarrow T* : Broadcast\ r_i$

(2) $R_i : Compute\ n_{desired} = \mathcal{P}(seed_{desired})$

(3) $T* : m = \mathcal{P}(seed_{T*})$

(4) $R_i \leftarrow T* : m$

(5) R_i : for each m received from each tag in the group

(6) if $(m == n_{desired})$ then

(7) $T_{desired}\ found$

(8) else

(9) $T_{desired}\ not\ found$

Figure 5.1 Simple Search Protocol

One main shortcoming of this protocol is that it is a one side authenticated search protocol. In this type of search, tags do not authenticate the readers before replying. So tags cannot know whether they are replying to an adversary or to a valid reader. But the tags should reply only to the authorized reader. Here the tags reply upon receiving a search query. So by querying a group of tags, an adversary may succeed in his/her attempt of searching a particular valuable tag, if that tag is present. Therefore, the tags need to authenticate the reader before replying. So when R_i broadcasts the search query, all tags, including the tag which satisfies the query, need to authenticate R_i before replying.

Search Protocol 2: **Enhanced Search Protocol**

(1) R_i : $Compute\ n_{desired} = \mathcal{P}(seed_{desired})$
(2) $R_i \rightarrow T*$: $Broadcast\ n_{desired}$
(3) $T*$: $a = \mathcal{P}(seed_{T*})$
(4) if $(a == n_{desired})$ then
(5) Let $k = \mathcal{M}(seed_{T*})$
(6) Let $x = \mathcal{P}(k)$
(7) $seed_{T_{desired}} = \mathcal{M}(k)$
(8) $R_i \leftarrow T_{desired}$: x
(9) else
(10) $R_i \leftarrow T_j$: $rand$ with probablity λ
(11) R_i : Let $s = \mathcal{M}(seed_{desired})$
(12) Let $m = \mathcal{P}(s)$
(13) for each $response$ from the group of tags
(14) if (m is equal to a $response$) then
(15) $seed_{desired} = \mathcal{M}(s)$
(16) $T_{desired}\ found$
(17) else
(18) $T_{desired}\ not\ found$

Figure 5.2 Enhanced Search Protocol

Moreover, since seeds are not updated in both parties after each search, the tags will reply with the same answers in subsequent search queries. If an adversary queries with a previously

learned r_i, tags will reply with the same values as before. Although the adversary will not be able to figure out which tag the reader was searching for, the adversary will be sure that the same search is taking place. By querying several times with different r_i, the adversary can learn a pattern for queries and replies.

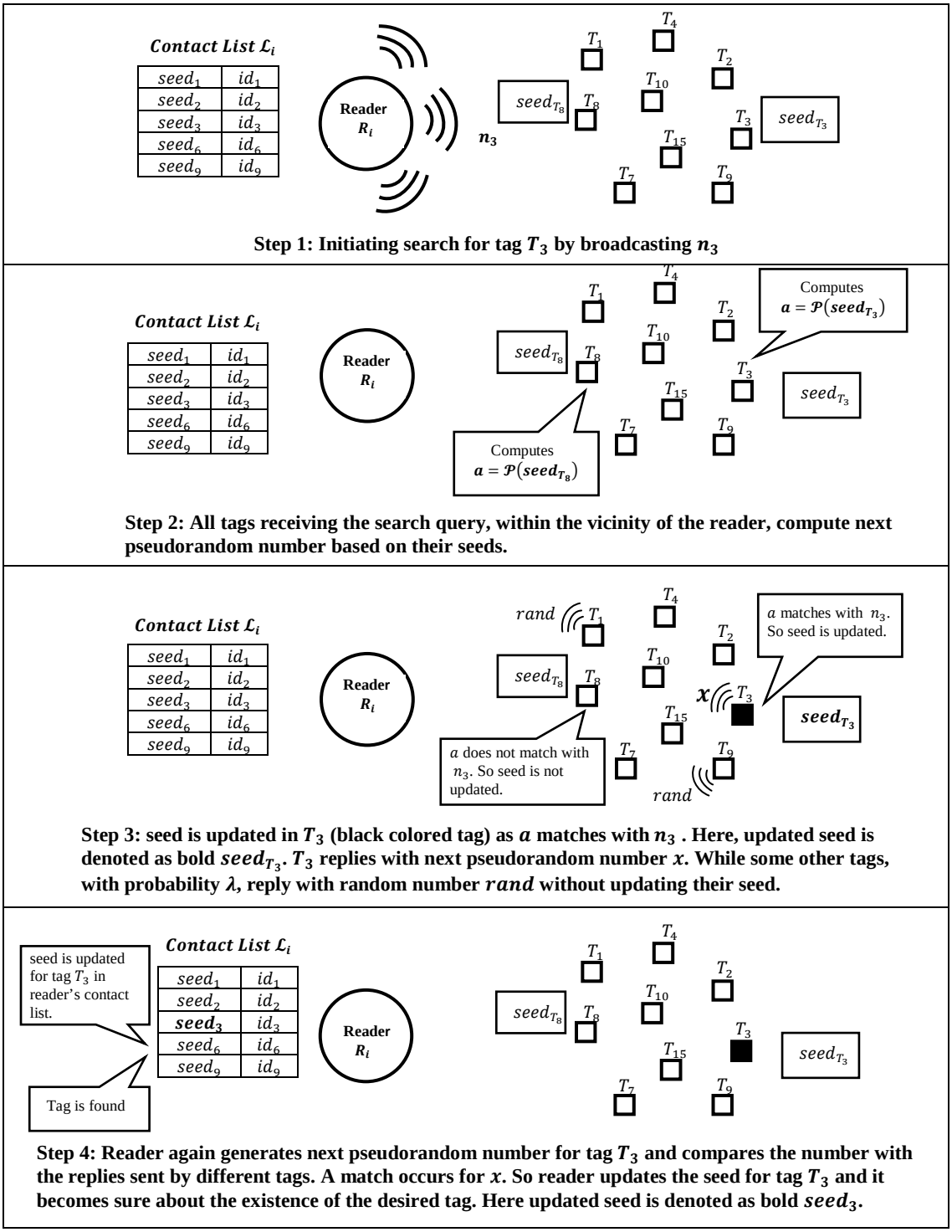
To solve the problems of simple search protocol, we can set up our goals for searching as follows. A tag should respond only to its authorized reader. A reader should query only the tags it is authorized to access to. Both parties (i.e. tags and reader) should update their seeds after a successful search. All these properties are incorporated in our next search protocol which is *Search Protocol 2* which we name as *Enhanced Search Protocol* (see figure 5.2). In this protocol, a reader issues a query in a way that only a legitimate tag can understand and a tag replies in such a manner that only an authorized reader can understand.

In this protocol, R_i computes $n_{desired}$ and broadcasts it to find out $T_{desired}$. All tags receiving $n_{desired}$ compare this number with the pseudorandom number a that is produced by using their own $seed_{T*}$. If a match occurs, a tag will be sure of the reader's authority. In fact only legitimate $T_{desired}$ can find a match because only an authorized reader can generate valid $n_{desired}$. Hence after authenticating the reader, $T_{desired}$ will reply with next pseudorandom number x from the sequence and update its own $seed_{T_{desired}}$. Now R_i computes the next pseudorandom number m from its sequence and compares it with each received $response$. If any $response$ is equal to m, then the reader can be sure that the tag is valid. Consequently reader R_i now updates the seed for $T_{desired}$. Security analysis for this protocol is discussed in the next subsection.

In enhanced search protocol, we let some other tags reply in addition to the desired tag to put the actual reply in disguise. Each tag that receives a search query will have some probability λ of replying with a random number. So by observing the tag replies, an adversary cannot recognize the tag that the reader is searching for.

5.3.5. Interaction diagram

The following figure (see figure 5.3) shows a detailed interaction diagram of enhanced search protocol.



Step 1: Initiating search for tag T_3 by broadcasting n_3

Step 2: All tags receiving the search query, within the vicinity of the reader, compute next pseudorandom number based on their seeds.

Step 3: seed is updated in T_3 (black colored tag) as a matches with n_3 . Here, updated seed is denoted as bold $seed_{T_3}$. T_3 replies with next pseudorandom number x. While some other tags, with probability λ, reply with random number $rand$ without updating their seed.

Step 4: Reader again generates next pseudorandom number for tag T_3 and compares the number with the replies sent by different tags. A match occurs for x. So reader updates the seed for tag T_3 and it becomes sure about the existence of the desired tag. Here updated seed is denoted as bold $seed_3$.

Figure 5.3 Interaction diagram of Enhanced Search Protocol when R_i is searching tag T_3

5.4. Protocol Analysis

In this section, we analyze our proposed authentication protocol against different types of attacks. For every attack, we describe how the attack is performed by an adversary and we explain how our protocol protects against the attack. R_i and T_j are referred to as a legitimate reader and tag.

5.4.1. Security analysis

***Tracking*:** Tracking attack in searching is slightly different from the one found in authentication related security literatures. In case of tag searching, an adversary cannot pick a particular tag to track. Rather, the adversary can only track a tag that has been searched for by a legitimate reader. For example, through the broadcast of a search query, a reader in a shopping mall wants to search for a tagged object, which may be worth thousands of dollars. Now if the object exists within the mall, the tag within the object will reply and an adversary will become aware that a valuable object exists around him/her. Therefore, the attacker may be able to track the location of the object and find out which store or owner the object belongs to.

Furthermore, the adversary has to iteratively query every tag in a group individually before determining what tag he is tracking. These reasons increase the difficulty of launching a tracking attack via the RFID search protocol. The very act of replying to a query can be used to identify a tag. So as long as a search query produces a unique reply, the reply becomes an identifier for a particular tag. Encryption does not solve the problem, since encryption only prevents an adversary from learning the content of a message, but not that a message has been sent. Our enhanced search protocol is resistant against tracking.

Let us consider the following attack. $\tilde{A}$ eavesdrops on the transaction between a reader R_i and a group of tags. So $\tilde{A}$ knows the queries and replies. $\tilde{A}$ will not be able to reverse compute the replies or learn the query but it can certainly be sure that a searching has been taken place. However $\tilde{A}$ cannot be sure which tag $T_{desired}$ the reader was searching for, since besides the

desired tag other tags also replied with probability λ. Now $\tilde{A}$ can replay previously learned $n_{desired}$ to track $T_{desired}$. After the previous successful searching between R_i and $T_{desired}$, both parties have changed their seeds. So $n_{desired}$, sent by the adversary, does not match with the one computed by $T_{desired}$. As a result, $T_{desired}$ responds with a random number. At the same time other tags will also reply with random numbers. If $\tilde{A}$ continues to query with different $n_{desired}$, all tags including the desired tag will reply randomly. Therefore $\tilde{A}$ is not able to track a tag.

***Cloning*:** Consider the following cloning attack. R_i queries to search a tag $T_{desired}$. If $T_{desired}$ is present it will reply. At the same time other tags will also reply. Suppose, $\tilde{A}$ finds out the tag the reader was searching for. Now if $\tilde{A}$ is able to clone $T_{desired}$, then $\tilde{A}$ can fool R_i by not replying or even giving a false reply. As a result, R_i will assume that the desired tag $T_{desired}$ does not exist in this group. In our protocol, this attack is impossible as $\tilde{A}$ is unable to find out the tag the reader was searching for.

***Eavesdropping*:** Here $\tilde{A}$ observes all the queries between a reader and tags. The goal of $\tilde{A}$ is to use the data to impersonate a fake reader R_i or a fake tag T_j. Our protocol is powerful against this attack. In our protocol $\tilde{A}$ will not be able to find out the expected reply of the reader as more than one tag will reply. $\tilde{A}$ can only observe $n_{desired}^{k}$ send by the reader. With this little knowledge $\tilde{A}$ cannot impersonate R_i or T_j, because after the last successful searching between R_i and $T_{desired}$, both of them have updated their seeds. So both of them, R_i and $T_{desired}$, are now expecting new values which are not known by $\tilde{A}$. Therefore by eavesdropping $\tilde{A}$ cannot launch a replay attack by using previous values.

Forward Secrecy: Forward secrecy means that an adversary will not be able to realize any previous output transmitted by the entity even if he/she compromises that entity. Enhanced search protocol ensures forward secrecy. The secret $seed$ of the desired tag, $T_{desired}$, shared between the tag and the reader, is updated each time using irreversible one way hash function.

After compromising a valid entity, $\tilde{A}$ cannot realize earlier responses based on the former secret $seed$ as it cannot derive the former secret $seed$s from the current one.

Privacy Protection: Users carrying various tagged items do not want to hamper their own privacy. If an adversary comes by any private information of the tag, by querying or eavesdropping, it may cause several vulnerabilities to the owner's day to day life. Our protocol protects users' privacy strongly. According to our enhanced search protocol, a tag never sends its own *id* to anyone, not even to the authorized reader. The tag sends its responses in disguise so that only an authorized reader can identify the tag. Moreover, along with the desired tag, additional tags also reply to the readers search query to preserve anonymity of the desired tag.

5.4.2. Cost analysis of enhanced search protocol

There are only two hash functions, $f(\cdot,\cdot)$ and $\mathcal{M}(\cdot)$, involved in our Enhanced Search Protocol. However $f(\cdot,\cdot)$ is used only at the deployment phase of tags when *CA* deploys all the RFID tags and authorizes the reader. So, it is logical to estimate the cost of our protocol based on the computation of $\mathcal{M}(\cdot)$ hash function. Moreover, since readers have high computation capacity, we calculate the cost of our protocol from the tag's perspective. From the Enhanced Search Protocol described above, we see that $\mathcal{M}(\cdot)$ is executed twice, first in line 5 and second in line 7. So, the cost for our protocols is little higher than alternative protocol [Tan07] which require the tag to perform only one hash function. The additional hash functions allow our protocols to be serverless and yet avoid exposing the tag secret to the reader. Considering communication cost, assuming that both reader and tag *id*s have the same length, the search protocol requires ($|n| + |rand| + |x|$) bits, where $|n|$ is the length of random numbers n_i or n_j. $|rand|$ is the length of $rand_i$ or $rand_j$ and $|x|$ is the length of x (see Enhanced Search Protocol, figure 5.2).

5.5. Comparison with Other Protocols

There have not been many attempts to produce a secure search protocol for RFID systems. In [Tan07], Tan et al. produced a series of search solutions that require very little storage, and can be distributed without an explicit need for a back-end server. Our solutions offer similar functions as Tan's, asking very little in terms of memory usage by tags or readers, and in addition we provide better security features. As tag search technique is one type of authentication and there are few search protocols proposed so far, we will compare our proposed search protocols with some existing famous authentication techniques along with other proposed search protocol [Tan07] based on the security features and other additional features.

Table 5.2 Comparison between different protocols

Protocols	Privacy Protection	Anti-Tracking	Anti-Cloning	Synchronization	Forward Secrecy	Serverless feature	Scalability Assurance
Seo-Lee-Kim [Seo06a]	Yes	Yes	Yes	Yes	Yes	No	Yes
Seo-Lee-Kim [Seo06b]	Yes	No	Yes	Yes	Yes	No	Yes
OSK [Ohkubo03]	Yes	Yes	Yes	Yes	Yes	No	No
YA-TRAP [Tsudik06]	Yes	Yes	Yes	No	No	No	Yes
YA-TRAP+ [Molnar04]	Yes	Yes	Yes	Yes	No	No	Yes
Av-ech[Avoine05]	Yes	Yes	Yes	Yes	Yes	No	Yes
Chiu-Bo-Qun [Tan07]	Yes	Yes	Yes	Yes	Yes	No	No
Serverless search protocol	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**No**

The protocol proposed by Seo et. al [Seo06a] provides high security. However, external intervention is required in order to perform ownership transfer, which is considered as a major

flaw of this protocol. Another highly scalable and highly secure protocol was proposed by the same authors in [Seo06b]. But this protocol has problem regarding untraceability and other security issues such as DoS attack. Another famous but novel authentication protocol is YA-TRAP [Tsudik06]. YA TRAP is secure against tracking and cloning attack and it does not require any computational overhead. However, in YA-TRAP, a simple DoS attack can be performed, exhausting the capability of the tag to respond to a legitimate reader. Moreover, this protocol does not assure forward secrecy. YA-TRAP+ [Molnar04] solves the problem of YA-TRAP, but inherits from it the inability to provide forward secrecy. The protocol of Avoine and Oechslin [Avoine05] guarantees firm security such as untraceability, forward secrecy, anti - cloning property. This protocol is also scalable but it offers no protection against DoS attack. The protocol proposed in [Tan07], is highly secure against most of the attacks. But this serverless protocol is not scalable.

Our proposed *Enhanced Search Protocol* is secure against tracking, cloning, eavesdropping, and physical attack. Moreover it can ensure forward secrecy, privacy protection, synchronization between tag and reader. But the biggest strength of our protocol is that it is serverless and it requires much less computation than the techniques mentioned in table 5.2. The serverless and lightweight nature of our search protocol makes it suitable for application areas where back-end servers are unreachable or unavailable. Moreover, S^3PR protocol is not vulnerable to single point-of-failure.

5.6. Application Areas of S^3PR

In this section we discuss two potential application areas of S^3PR protocol.

1. Mishandled bag search within Airports: Passengers suffer a lot due to inefficient bag handling system in the airports. Passengers have to deal with customer service representative in search of their lost baggage. The industry refers to this as “Mishandled bag”. Every missing or mishandled bag costs the responsible airline approximately $80 to $120, or an average of $100

per bag. And yearly this figure rises to approximately $146 million. Moreover, this type of events degrades the reputation of the responsible airline. However a simple, cost-effective, efficient solution to Mishandled Bag can be achieved using our search protocol. Whenever a passenger arrives to customer service representative to report about missing bags, the representative can get the tag *id*s of bags from airport operations database (AODB) and can request a search operation. Mobile readers can be used to identify the exact location of the missing bag by directing those readers to different location within airport.

2. User Interactions in a smart space: A smart space typically contains multiple smart objects offering several invisible services. Users' personal devices are usually used to interact with the smart space. Discovering invisible services securely and authenticating the users are interesting research problems in the smart space domain. Our approach offers promising solutions to both of these problems. Iconic images embedded with RFID tags can advertise invisible services and user terminals can be equipped with an RFID reader. A user can search for a specific service (tags in this case) or can initiate a service by touching the tag. Considering the pre-negotiation between the reader and the tags, secure discovery and searching mechanism can be easily achieved applying our protocol.

5.7. Summary

The application areas of RFID systems are unlimited. In spite of this, secure RFID tag searching has not gathered much attention till now. Bur it will become very important when RFID tags will be deployed at a larger scale. Therefore, in this chapter we introduced various problems incurred while performing secure serverless tag search. Moreover, we analyzed different attacks that can be launched against RFID tag searching. Finally we proposed a secure serverless RFID tag search protocol that can safeguard against the major attacks without the server's intervention.

Chapter 6: A Scalable and Efficient Search Protocol (S-Search)

6.1. Introduction

RFID holds the potential of changing how businesses operate today but its implementation is not straightforward. Since RFID tags are extremely constrained in time and space, enforcing high level of security with excessive cryptographic computation is not possible. Secured mechanisms for tag authentication have been in the midst of researcher's interest for almost a decade. A number of challenges such as security/privacy concern, scalability, high cost, reliability of the technology, efficient performance of the system, and even some more issues need to be addressed.

In RFID systems, tags equipped within different objects have unique identification information. This information is applicable in various fields such as supply chain management, and product maintenance etc. In all such practical implementations, often a reader needs to determine efficiently whether a particular tag exists within a group of tags no matter how large the size of the tag set is. This is referred to as scalable RFID tag searching. RFID tag searching is one sort of extension of RFID authentication, which has not been given much attention so far. But with the massive deployment of RFID technology, tag searching will become a very significant issue.

In practical RFID systems, the number of tags within the system is extensive. Searching a particular RFID tag among this immense number of tags needs to be efficient. Which means that searching of tags need to be scalable. Scalability means that a reader will be able to search a tag with constant computational time regardless of the number of tags that is owned by it. Non-scalable tag search protocols are not feasible as they are not implementable in real life RFID systems that consist of large number of tags.

In this chapter, we try to find solutions to the following questions: a*) how can a reader search a particular tag within a set of tags efficiently?* b) *how can the search protocol maintain scalability?* Here, we propose a scalable, low cost, and secure search protocol that provides solutions to the preceding questions. A version of this proposal has been published in [Hoque10].

6.2. Existing Trivial Solutions

The assortment of research literature on RFID searching is inadequate although it is a major issue in its real life implementation. We will mainly concentrate on the search protocols proposed so far in [Tan07] and [Ahamed08b] that are relevant to our proposal.

Serverless RFID search protocols were proposed in [Tan07] for the first time. According to this protocol, a reader wishes to find out whether a specific tag is within its vicinity by broadcasting $h(f(r_i, t_j)||n_r) \oplus id_j$, n_r and r_i. Based on this search query, only the intended tag, if exists, reply with its encrypted id. Other tags within the reader's vicinity reply a random number based on certain probability. Tags authenticate the reader based on the search query and reader authenticates tags based on the reply "string". Both valid query and valid replies are generated by legitimate parties.

Another serverless search protocol was proposed in [Ahamed08b]. In this paper, the authors proposed different search protocols in which tag identifier is not passed to the reader in response to a reader's query. Tag sends certifying information to the reader in such a way that only the authorized reader is able to find out whether this is the desired tag. However, both of these search protocols lack scalability when the number of tags increases drastically in the system.

Another major drawback of both of these protocols is, multiple tags reply at the same time when reader broadcasts a search query. This creates data and signal collision in the communication channel between the tag and the reader. Because of collision, those tags whose

data were distorted, needs to reply again. As a result, both of these protocols are not efficient with respect to time.

In an RFID system, a collision occurs when multiple tags try to transmit data to the reader at the same time. This results in the reader being unable to obtain any useful information. Prior works [Bonuccelli06, Cha05, Lee05 and Micic05] have focused on improving protocols to reduce collisions. However these solutions are ultimately bounded by the number of tags.

Another approach is to use probabilistic techniques to determine some features of a large collection of RFID tags. These include methods to estimate the cardinality of a set of tags [Kodialam06], and to determine popular categories of tags [Sheng08]. For a reader to successfully receive data from multiple tags, anti-collision protocols must be designed so that replied data from multiple tags will not be distorted because of collision. In general, two approaches are used to regulate collision. The first is based on the ALOHA protocol [Metcalfe75, Lee05, Schoute83, Vogt02, and Wieselthier89]. A representative protocol used in RFID systems is the framed ALOHA [Metcalfe75], a variation of ALOHA [Abramson70]. In this protocol, a frame is divided into multiple time slots. The communication is initialized when the reader broadcasts a frame size, i.e., the number of slots in the frame. Every RFID tag responds only in a particular slot in the current frame. The reader can successfully receive data in a certain slot if only one tag picks the slot for transmission. This process is repeated until all data are collected.

The second approach uses the tree traversal technique [Choi04, and Cidon88]. The reader broadcasts an ID prefix, and those tags whose IDs match the prefix will respond. If a collision is detected, the reader will append ‘0’ or ‘1’ to the prefix and send new prefixes again. It is equivalent to traversing a binary tree, where each tag’s ID is a leaf node. The expansion of prefix stops if only one tag responds.

In this chapter we propose a scalable tag search protocol using Slotted ALOHA based communication between legitimate tag and reader. We present a lightweight solution that does

not require expensive tag hardware such as an accurate on-chip timer or cryptographic MAC functions which are unavailable on passive RFID tags.

6.3. Proposed Solution

The objective of secure RFID searching should be: the reader will search a specific RFID tag which he is authorized to access. Tags will reply with valid answers only if the reader is legitimate. Our major contributions are as follows:

- We propose scalable, forward secure, anonymous, and efficient search protocol, (***S-Search***), for searching RFID tags efficiently within a system.
- The *S-Search* protocol does not require the reader to collect *id*s from each RFID tag, but is still able to accurately find out a specific RFID tag.
- The major focus of this search protocol is to keep the searching scalable so that it can implemented efficiently in real life/practical RFID systems.
- Our Searching technique provides privacy protection by neither broadcasting tag *id*s in public, nor revealing *id*s to the RFID reader.

6.3.1. System architecture

Usually, the RFID system consists of wireless tag, *T*, wireless reader, *R*, and back-end database. A certification authority *CA* is involved in the system to certify readers and authorize them to particular tags. We discuss the roles of different components of an RFID system and the communication techniques between them.

Tag: Each tag *T* is comprised of an IC chip and antenna. Tags can be of two types. There are active tags, which have a battery, and passive tags, which have no battery. We focus on the passive tag, which is expected to be the most common type of RFID tags. In our system, each tag is able to communicate with one reader at a time.

Reader: A reader *R* is a device that sends some query using radio frequency signal to a tag, receives the information sent by the tag and performs some important computation on those data.

Server: By server we mean a secure server. It has a database and manages various types of information related to each *T*. The server resolves the *id* of *T* from the information sent by *T* through authenticated *R*.

Communication: The wireless communications between the reader and the tag is assumed to be vulnerable to eavesdropping. Communications between the reader and the Certification Authority (that we refer as *Trusted Authority - TC* in the rest of the chapter) are assumed to be conducted over a secure channel.

6.3.2. Problem Definition

We assume that a server has a set of tags. Each RFID tag has a unique *id*. A set of tags once created is assumed to be static which means that no tags are added or removed from the set. The problem is to search a particular RFID tag among this set of tags.

Protocol Goals. The goal of the server is to search a specific tag remotely, quickly and efficiently so that the search can be scalable even with large number of tags in the system.

Adversary Goals. The goal of an adversary in any RFID system is to counterfeit a real tag with its real data such that it can only be distinguished from the real one with small probability. Evidently, this fake tag can let a fake product to be identified as an authentic one just by embedding the fake tag into the fake product. We denote an adversary is denoted as $\tilde{A}$. The adversary can control a number of readers and tags. Each reader and tag controlled by $\tilde{A}$ is denoted as $\tilde{R}$and $\tilde{T}$, respectively. $\tilde{R}$ is unauthorized to have access to real tags as it cannot get any tag secret *t* and *id* (see section 7.3.3) from *CA.* Similarly, $\tilde{T}$ is not valid as it does not have secret and identifying information of any tag. Moreover, we assume that all the entities (tags, readers,

TC including adversary, adversarial readers and adversarial tags) have polynomially bounded resources.

Attack Model. We assume that $\tilde{A}$ is more formidable than a passive attacker. In addition to eavesdropping on the channel between a valid reader and a valid tag, $\tilde{A}$, like an active attacker, can install a rogue reader $\tilde{R}$ that can communicate with a valid tag. Even $\tilde{A}$ can also install a fake tag $\tilde{T}$ to communicate with an authorized reader. In both cases, the adversary wants to counterfeit a tag with the learned information. Furthermore $\tilde{A}$ can launch physical attacks. However hardware-based defenses against physical attacks are beyond the scope of this proposal.

6.3.3. Preliminaries

Table 6.1 Notations for S-Search protocol

Symbol	Meaning
$T*$	Set of RFID tags
$T_{desired}$	RFID tag for which the reader executes a search operation within T_*
n	Number of tags within T_*
$h(.)$	One way hash function
SP	Slot position within frame
BR	Bit Record generated by the reader with the replies of tags
t_i	Tag secret of T_i
$t_{desired}$	Tag secret of $T_{desired}$
$id_{desired}$	id of $T_{desired}$
$rand_m$	First m bits of a random number
x_m	First m bits of number x

Like many other earlier research, here we have assumed that RFID tags are capable of performing cryptographic hash function. But cryptographic hash function requires additional gates to be implemented within the tag. This eventually increases price per tag. So due to the higher production cost, most RFID tags do not provide these hash function. Some common hash functions like MD4, SHA-1, SHA-256 requires between 7350 and 10868 additional gates [Feldhofer06]. So majority of the proposed protocols can be used with expensive RFID tags which are likely to be attached with more valuable items.

We refer an RFID reader denoted as *R*. The *TC* is a trusted party who deploys all the RFID tags and authorizes any RFID reader. For the sake of simplicity we assume that *R* and *TC* communicate through a secure channel. According to our proposal, Each RFID tag *T* contains a unique value *id* a unique secret *t* in its nonvolatile memory. We denote the frame size as *f* and the random number generated by the reader/tag as *r*. Both the server and the reader contains a table of tag entries. Each entry of the table contains the corresponding tag *id* and the associated tag secret *t.* The necessary notations for S-Search protocol are summarized in table 6.1.

Our communication model is based on the slotted ALOHA. We assume that an RFID reader is able to distinguish the slots with no reply, single reply, or multiple replies. We define these slots as empty slot, single-reply slot, or collision slot respectively. In our approach, every tag does not transfer the long *id*, but a short random bit string (usually < 10 bits, which we denote as *m*), as long as the RFID reader can detect the presence of the signal. In this proposal, we assume that RFID tags resolve collisions using a slotted ALOHA scheme [Hernandez01]. In our protocol, the server sends a frame size *f* and a random number *r* to the reader. The reader broadcasts *(f, r)* and $h(r \oplus t_{desired})$ to all tags. Each RFID tag uses the random number *r* and its *id* to hash to a slot position *SP* between *[1, f]* to return their reply where,

$$SP = h(id \oplus r) \bmod f$$

The tags simply reply with a few random bits signifying the tag has chosen that slot. In other words, instead of the reader receiving

{… | id_1 | 0 |...| collision | 0 |…},

where 0 indicates no tag has picked that slot to reply, and collision indicates multiple tags trying to reply in the same slot, the reader will receive

{…| random bits | 0 |… | random bits | 0 |…}

This is more efficient since the tag *id* is much longer than the random bits transmitted. This is even more secure as the tags do not have to transmit *id* to the reader. So an adversary

cannot find out *id*'s of tags by eavesdropping within the channels. From the reply, the reader can generate the Bit Record (*BR*)

BR = {…| 1 | 0 | 1 | 0 | …},

Here, 1 indicates one tag has picked that slot. The server knows all the *id*s and tag secrets of all the tags and the parameters *(f, r)*. Therefore, it will be able to determine the resulting Bit Record (*BR*) for an intact set of tags ahead of time. The intuition behind this is to let the server pick a *(f, r)* for the reader to broadcast to the set of tags. The server uses the Bit Record (*BR*) generated by itself and Bit Record (*BR*) returned from the reader to determine whether the searched tag is present or not.

6.3.4. Search protocol

Algorithm 1: *Interaction between server and reader*
1. Server sends *(f, r)* to the reader *R*
2. *R* executes Algorithm 4
3. All nearby tags executes Algorithm 3
4. Compute Slot Position for the desired tag $T_{desired}$ by $SP_{desired} = h(id_{desired} \oplus r) \bmod f$
5. Receive Bit Record (*BR*) from *R*
6. **if** $(BR\ (SP_{desired}) = 1)$ **then**
7. $T_{desired}$ is present
8. **else**
9. $T_{desired}$ is not present

Figure 6.1 Algorithm for interaction between server and reader in S-Search Protocol

Algorithm 2: *Interaction between reader and tags*
1. Reader broadcasts *(f, r)* and $h(r \oplus t_{desired})$ to all tags
2. Reader *R* executes Algorithm 4
3. Each tag T_i (where i = 1 to n) executes Algorithm 3
4. Reader returns Bit Record (*BR*) to the server

Figure 6.2 Algorithm for interaction between reader and tags in S-Search Protocol

Algorithm 3: *Algorithm executed by tags*

1. Receive *(f, r)* and $h(r \oplus t_{desired})$ from *R*
2. Each tag T_i (where i = 1 to n) Compute Slot Position (*SP*) by $SP_i = h(id_i \oplus r) \bmod f$
3. **while** *R* broadcasts Slot Position (*SP*) **do**
4. **if** (*SP* == SP_i) **then**
5. compute $h(r \oplus t_i)$
6. **if** $(h(r \oplus t_i) = h(r \oplus t_{desired}))$ **then**
7. return $\left(h(id_i \oplus t_i \oplus r)\right)_m$ to *R*
8. **else**
9. return $rand_m$ to *R* with probability λ

Figure 6.3 Algorithm executed by the tags in S-Search Protocol

Algorithm 4: *Algorithm executed by the reader R*

1. Compute Bit Record (*BR*) of length *f*
2. Initialize all entries of *BR* to 0
3. Compute $h(id_{desired} \oplus t_{desired} \oplus r)$ for $T_{desired}$
4. **for** Slot Position *SP =1 to f* **do**
5. **if** *receive reply* or *collision* **then**
6. **set** *BR[SP]* = 1
7. **if** (*reply* = $h(id_{desired} \oplus t_{desired} \oplus r)_m$) **then**
8. $T_{desired}$ is present
9. **else**
10. $T_{desired}$ is not present

Figure 6.4 Algorithm executed by the reader in S-Search Protocol

6.3.5. Protocol description

In case of single RFID tag search, the reader first broadcasts a frame size and a random number, *(f, r)*, together with $V_{desired} = h(r \oplus t_{desired})$ to all the tags. Each RFID tag T_i uses its own tag secret t_i and r to generate $V_i = h(r \oplus t_i)$. If V_i is equal to $V_{desired}$, tag T_i becomes sure

that the reader is searching for itself. Only the desired tag returns first *m* bits of $h(id_{desired} \oplus t_{desired} \oplus r)$ during the slot position specified by $SP_{desired} = h(id_{desired} \oplus r) \bmod f$. And all other tags reply random number with probability λ during the slot position specified by $SP = h(id \oplus r)\ mod\ f$. The reader searching for tag $T_{desired}$ calculates $h(id_{desired} \oplus t_{desired} \oplus r)$ ahead of time. At the time of receiving replies from different tags, the reader checks the content of slot position $SP_{desired}$. If the received content matches with first *m* bits of $h(id_{desired} \oplus t_{desired} \oplus r)$, the reader becomes sure that $T_{desired}$ is present. Now reader forms the Bit Record (*BR*) of length *f* (frame size) to transmit to the server. Initially reader assigns 0 to all the slot positions. However, the reader stores 1 in all those slot positions in which it receive a reply (either $h(id_{desired} \oplus t_{desired} \oplus r)$ or random number). We assume that the frame size (*f*) is large enough and there are more slot positions within the frame than total number of tags (i.e. $f > n$). Reader stores 1 in those slots in which it identifies a collision. Therefore, some slot position of the *BR* contains 0 and some contains 1. But the adversary cannot find out in which slot position the desired tag replied. This technique of bit assignment allows our search protocol to be secured against some major attacks which we will discuss in section 8. The Bit Record is transmitted to the server. We assume that the channel between server and reader is secure.

Next, the server only checks the slot position $SP_{desired}$ of the *BR* to find out whether the desired tag is present or not. Slot position $SP_{desired}$ containing 1 indicates that the desired tag is present. Algorithm 1(figure 6.1) shows the overall interaction between the server and the reader. Algorithm 2 (figure 6.2) shows the interaction between the reader and tags. Each tag in the set executes algorithm 3 (figure 6.3) independently. Algorithm 4 (figure 6.4) generates the Bit Record (*BR*) and returns it to the server. In algorithm 2, we see that tag does not need to return the tag *id* to the reader. They return a much shorter number (*m* bits), either random number or $h(id_{desired} \oplus t_{desired} \oplus r)$, to inform their presence.

6.4. Protocol Analysis

In this section, we analyze our proposed search protocol against different types of attacks.

6.4.1. Security analysis

***Tracking*:** Our search protocol is resistant against tracking. Consider the following attack. $\tilde{A}$ eavesdrops on the transaction between a reader *R* and tags. So he knows the queries and replies. He will not be able to reverse compute the replies or learn the query but he can certainly be sure that a searching has taken place. However he cannot be sure, which tag $T_{desired}$, reader was searching for. Since besides the desired tag some other tags also replied with probability λ. Adversary can find out which tag replied in which slot but he will not be able to determine what were replied by the tags. Since outputs of all tags will seem to be pure random to the adversary.

***Cloning*:** Consider the following cloning attack. *R* queries to search a tag $T_{desired}$. If $T_{desired}$ is present it will reply. At the same time other tags will also reply. Suppose, $\tilde{A}$ finds out the tag the reader was searching for. Now if he is able to clone $T_{desired}$, then he can fool R by not replying or even giving a false reply. As a result, R will assume that the desired tag $T_{desired}$ does not exist in this group. In our protocol, this is impossible. Because $\tilde{A}$ is unable to find out, which tag the reader was searching for.

***Eavesdropping*:** Here $\tilde{A}$ observes all the queries between a reader and tags. And his goal is to use the data to impersonate a fake reader R or a fake tag T_j. Our protocol is powerful against this attack. In our protocol $\tilde{A}$ will not be able to find out the expected reply of the tags as more than one tag will reply. He can only observe the data send by the reader. With his little knowledge he cannot impersonate R or T_j. The output of the desired tag consists of the random number, tag secret *t*, and *id*. The tag secret and *id* is not known to the adversary. So he is not capable to generate the new outputs of the desired tag. Therefore by eavesdropping $\tilde{A}$ cannot launch a replay attack by using previous values.

6.5. Comparison with Other Protocols

In this section (table 6.2), we compare S-Search Protocol with some existing protocols.

Table 6.2 Comparison between different protocols

Protocols	Privacy Protection	Anti-Tracking	Anti-Cloning	Synchronization	Forward Secrecy	Scalability Assurance
Seo-Lee-Kim [Seo06a]	Yes	Yes	Yes	Yes	Yes	Yes
Seo-Lee-Kim [Seo06b]	Yes	No	Yes	Yes	Yes	Yes
OSK [Ohkubo03]	Yes	Yes	Yes	Yes	Yes	No
YA-TRAP [Tsudik06]	Yes	Yes	Yes	No	No	Yes
YA-TRAP+ [Molnar04]	Yes	Yes	Yes	Yes	No	Yes
Av-ech[Avoine05]	Yes	Yes	Yes	Yes	Yes	Yes
Chiu-Bo-Qun [Tan07]	Yes	Yes	Yes	Yes	Yes	No
Serverless search protocol	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**No**
S-Search Protocol	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

6.6. Summary

In this chapter, we propose scalable and efficient RFID tag search protocol that can safeguard against some major attacks without performing complex cryptographic computation. This protocol only requires tags to be capable of generating hash function and performing XOR operation. Our approach differs from prior works in that our technique does not require the reader to collect the *id* from every tag. Also it requires little computation to search a particular tag which makes our protocol scalable and highly suitable for practical large scale RFID systems.

Chapter 7: A Hexagonal Architecture for Tag Search (EDSA)

7.1. Introduction

Several researches were conducted to ensure security and privacy of consumers and it has brought some fruitful outcomes. Numbers of privacy problem were identified in [Rieback06, Juels05b] and many privacy preserving cryptographic techniques were identified in [Juels05b]. The definition of strong privacy given by Juels and Weis in [Juels06] seems to have a conflicting relation with scalability. According to Juels and Weis in [Juels06], private tag identification involves decryption of the ID of the tag being identified by exhaustive search. Definitely, this technique will not ensure scalability when the number of tags will increase. But both strong privacy protection and scalability are very important for the real life implementation of RFID technology.

One such real life situation is emergency evacuation system. In such circumstances RFID tag can be used to keep track of each and every person stuck in danger, persons who are unable to leave the danger premises and persons who are undetected. This RFID system will raise scalability issues if typical RFID identification techniques and infrastructure is used. One solution to this problem can be suggested by using distributed architecture of [Solanas07] which ensures scalability by using typical hash lock scheme. In this chapter we propose an ***E***nhanced ***D***istributed ***S***calable ***A***rchitecture (***EDSA***) that provides even more scalability and security by using serverless tag authentication [Ahamed08c] and search protocol [see chapter 5]. The use of serverless search and authentication protocol ensure efficiency by incorporating a back-end server. A version of this proposal has been published in [Ahamed08a].

The main reason of using serverless search protocol is: it reduces set up, maintenance cost and mostly traffic from reader to back-end server. Back-end server is now devoted to some higher level maintenance of real life application as it hands over the responsibility of authenticating and searching tags to readers. It is practical and feasible to use serverless search

and authentication in emergency evacuation system as it can operate without the involvement of server.

The goal of this chapter is to discuss an RFID system architecture to solve the scalability issues in practical application scenarios. It starts by an introduction to the scalability requirements in real life RFID systems. Subsequently existing trivial solutions to solve the scalability issues are discussed. After that an overview of the proposal has been given. This is followed by the technical details of our architecture. Finally, we describe the application of our scalable architecture in a practical scenario that is in emergency evacuation system.

7.2. Existing Trivial Solutions

A cell-based distributed architecture was proposed by A. Solanas et al. in [Solanas07]. In this paper an area is divided into cells, where each cell is assumed to be a square. Scalability is ensured by using information sharing protocol suites, though the system could be more scalable by assuming different structure cells. A single authentication operation to search a particular tagged object, in the system costs much computation. According to A. Solanas et al. in [Solanas07], tags capable of simple cryptographic computation can use improved randomized hash lock [Juels06], in a scalable manner to send its encrypted ID to the reader. Here, other authentication techniques can be used in addition to improved randomized hash lock. However, our proposal of using search protocols can achieve more scalability for the system.

7.3. Proposed Solution

In subsequent subsections, we will discuss the following major contributions:

- We propose a distributed, hexagonal, cell based architecture, ***EDSA*** (***E****nhanced* ***D****istributed* ***S****calable* ***A****rchitecture)*, which can be used for secured tag identification and tag searching without compromising scalability.
- We point out the challenges of an emergency evacuation system.

- We also offer solutions to these challenges using EDSA and our proposed serverless search protocol.

To the best of our knowledge, the integration of a serverless and a servered technique within the same architecture is addressed for the first time in this proposal.

7.3.1. System architecture

We propose a distributed architecture for large scale application where not only secure RFID authentication is needed, but also efficiency, cost-effectiveness and accuracy are great concerns. Our system is an improved version of the architecture of [Solanas07]. We try to alleviate the shortcomings of the architecture proposed in [Solanas07]. In our system, the use of a different cell structure provides more scalability than the one of [Solanas07].

RFID reader, tags and back end server are defined as main components of the system. Inclusion of back end server is completely different approach in comparison with the previous literature [Solanas07]. The tags are assumed to be passive. We also assume that the tags can compute simple one way hash functions and generate random numbers. Tags deployed in the system need to have enough non-volatile memory to execute the Enhanced Search Protocol and serverless authentication protocol [Ahamed08c]. We also assume that the mobility of the tags are enabled in our system i.e., tags can change their location at any time.

In our system readers are static and active devices. They are capable of detecting the tags and performing crucial functions to do authentication and exploration of tags according to our serverless search protocols. To cover an area, readers are logically distributed. In [Solanas07], the area is divided into equal squares such that each square is covered by a single reader. However, hexagon is a better choice to partition the area. We refer to each hexagon as a cell. Hexagonal cell improves our system. Each single reader covers a specific cell. We also assume that our system facilitates a secure reader to reader communication channel.

The backbone of our system is a back-end server. It accesses the database of the *id*s of tags. On basis of requirements, the server can communicate with each reader while monitoring the system. In spite of having a back-end server, ours system does not comply with a centralized scheme. It is a servered as well as serverless scheme (see section 7.3.4). The functionalities of the back-end server and the readers are described in details later. Again we assume that the communication and exchanging of information between the server and the readers are performed through a secure communication channel.

7.3.2. Coverage area

In this section we describe the distribution of components in our system. The readers are spatially distributed and the tags are scattered among them. Consider an area S which can be covered by a couple of readers. We have two permitted points called ENtrance Point (ENP) and EXit Point (EXP) for tags to enter or exit the area S, respectively. We assume all the readers are of same read range. The size of the cell, covered by each reader, is equal. C_i denotes the i^{th} cell of S. We consider

$$S = \bigcup_i C_i \mid C_i \cap C_j = \emptyset, \ \ \forall i, j \ \wedge \ i \neq j$$

Suppose, cell C_i is covered by reader R_i. Also, $Adj(R_i)$ is the set of readers adjacent to R_i. Next we describe some other related topics to point out how hexagonal cell improves our system. After that we again come to the improvement point (see section 7.3.5).

7.3.3. Privacy and search

Identification protocols of tags are vulnerable due to eavesdropping and other attacks. However, authentication protocols are more so protected than identification protocols. The authentication protocol proposed in [Ahamed08c] is indeed a secure protocol that never negotiates with privacy of both the reader and the tags. Therefore, the above mentioned serverless authentication protocol can be used in our system.

However, the use of authentication protocols to search a particular tag will raise scalability issues. Therefore we need to use secure search protocol for tag exploration in our system. Many real life applications, for example emergency evacuation system have an appreciable requirement for tag search. Even though the assortment of search protocol is limited, we have to incorporate a search protocol in our system. The enhanced search protocol [see chapter 6] is entirely appropriate for RFID systems it does not compromise privacy and security while searching for a tag.

7.3.4. Protocols and functionalities

In fact there will be three types of communications in our system. These are: $tag \leftrightarrow reader$, $reader \leftrightarrow reader$ and $reader \leftrightarrow back-end\ server$. We now point out the communication protocols those are to be used for the system.

A. $tag \leftrightarrow reader$

Between tags and reader, there will be two types of functionalities. One is for authentication and other is for search. The authentication protocol proposed by [Ahamed08c] can serve the purpose of authentication between tag and reader. The enhanced search protocol serves the purpose of searching a tag. Since the protocols are proposed for a serverless reader, it seems to be implausible for the readers of this system. Here, each reader can perform like a serverless reader as well as a reader backed by a server. More about these readers are described below.

B. $reader \leftrightarrow reader$

A reader can share its information with adjacent readers. The shared information contains the $seed$ used for a tag along with the tag id and reader id of the reader which locates the particular tag within its cell. There is subtle difference between this shared information and ownership information of a reader [Ahamed08c]. From now on we refer to the shared information as Ownership Information in this context. Now, consider a reader R_i locates a tag T_j within C_i. The ownership information is $Info_{R_i}^{T_j} = seed_j \parallel id_j \parallel r_i$ where, $seed_j$is the $seed$ for T_j stored in

R_i, id_j is the identifier of T_j, r_i is the identifier of R_i, and ‖ denotes concatenate. After authentication, R_i sends $Info_{R_i}^{T_j}$ to reader $R_k \in Adj(R_i)$. As a result, reader R_k stores $Info_{R_i}^{T_j}$ in its contact list so that it can authenticate T_j whenever T_j enters C_k in future.

Our EDSA system is a servered as well as serverless system, we here emphasize on its serverless property. R_{ENP} and R_{EXP} indicate the readers of ENP and EXP respectively. The readers in the system need three protocols. A brief description of each protocol is given below.

***Arrival Protocol*:** This protocol starts when a tag first enters the system through ENP. At the very beginning all readers other than that at ENP own no tags, i.e., they all have empty contact lists. R_{ENP} (i.e. the reader at the entrance) is supposed to be authorized by back-end server for all possible tags which can enter the system through it. Whenever an authorized tag T_j enters the system, after authentication, R_{ENP} sends$Info_{R_{ENP}}^{T_j}$ to adjacent reader R_{in} which appends the ownership information in its contact list. Otherwise, R_{ENP} alerts the system about the attempt of an unauthorized tag. To roam into the system T_j has to move into cell C_{in}. Upon entering the cell, R_{in} locates T_j and authenticates it without any involvement of back-end server as R_{in} has ownership information in its contact list. Then R_{in} sends $Info_{R_{in}}^{T_j}$ to all the adjacent readers $Adj(R_{in})$ and thus causes adjacent readers to be authorized for the tag T_j.

***Roaming Protocol*:** This protocol sets off when a tag enters another cell equipped with a reader from the cell of its current reader. Whenever T_j enters the cell of the reader R_i, it locates the tag T_j. R_{own} is the reader of the cell where T_j was before its detection by R_i. Due to the spatial distribution of readers, $R_i \in Adj(R_{own})$ and R_i contains $Info_{R_{own}}^{T_j}$. Hence, R_i is authorized for T_j. After authentication, R_i sends its ownership information $Info_{R_i}^{T_j}$ to all adjacent readers $Adj(R_i)$. Now depending on the information in its contact lists, each adjacent reader R_k behaves differently.

a) If $R_k \in Adj(R_i) \cap Adj(R_{own})$, R_k has to just replace $Info_{R_{own}}^{T_j}$ with $Info_{R_i}^{T_j}$ in its contact list.

b) If $R_k \in Adj(R_i) - Adj(R_{own})$, R_k appends $Info_{R_i}^{T_j}$ in its contact list.

c) If $R_k = R_{own}$, it changes its ownership information and passes on $Info_{R_i}^{T_j}$ to its adjacent readers in $Adj(R_{own})$. Now each reader R_l adjacent to R_{own} behaves in two different ways.

c1) If $R_l \in Adj(R_i) \cap Adj(R_{own})$, then do nothing.

c2) If $R_l \in Adj(R_{own}) - Adj(R_i)$, it erases $Info_{R_{own}}^{T_j}$ from its contact list.

At the end of this protocol, only $\{R_i\} \cup Adj(R_i)$ readers know $Info_{R_i}^{T_j}$ and only they have capability to authenticate the tag T_j. Again, server is not involved for any responsibilities.

***Departure Protocol*:** Whenever a tag is about to exit the system through EXP, this protocol starts. When the tag T_j reaches the EXP to exit from the system, R_{EXP} sends $Info_{R_{EXP}}^{T_j}$ to readers in $Adj(R_{EXP})$ to erase ownership information because there is no chance to go back. Moreover, the previous owner (reader) propagates this information to its neighboring readers to remove ownership information of T_j from their contact lists. Hence nothing remains in the system about the departed tag. Therefore, we can appreciably refer the system as serverless despite the presence of a server.

The dynamic way of authorizing readers for tags and removal of authority from readers implies that a reader has to deal with a moderate number of tags. Therefore system sustains its scalability. In fact, system can even be more scalable by incorporating hexagonal cell which we will describe later.

C. $reader \leftrightarrow back-end\ server$

Association of a back-end server strengthens the system and its efficiency. Therefore our system is now capable of performing in many real life applications. Our system is equipped with

a back-end server which can efficiently access the database of all the *id*s of the tags. The server can communicate with each reader as we've mentioned earlier. Server can authorize all readers. But it authorizes only R_{ENP} for tags. Server monitors the system constantly.

As searching is unavoidable requirement for most real life application, the server can do a search whenever it faces a request from application. For simplicity, the server just sends a search request to all the readers in the system along with the *id* of the tag, id_j, for which readers have to perform a search. However, not all the readers have ownership information related to id_j.Only those readers who have ownership information can invoke search for id_j according to enhanced, while others remain silent. Whenever a reader locates id_j within its cell, it replies to the server with its consequence. Since we assume the communication between the reader and the server is performed through a secure channel, the reader can just send the successful search result. Otherwise, unsuccessful readers have to send fake messages to the server to fool an adversary. Through the search of a tag, server determines the cell in which the tag resides. This technique can be used in the application where locating or tracking of something is required.

For this back-end server, we cannot refer our system as entirely serverless. Though the intervention of server is limited to some special purposes such as search, authorization, monitoring, etc., we can't deny the presence of back-end server.

7.3.5. Enhanced cell organization

So far we have just mentioned the improvement of our system by organizing cell as hexagonal. Here we justify our claim. In fact, hexagonal cell based architecture, unlike the previous one [Solanas07], is another prominent feature of our system.

Let D be the radius of that circle that circumscribes cell (hexagon or square) (see figure 7.1 and figure 7.2) and d be the radius of the circle inscribed in a cell. As each reader has same read range, we assume read range is D.

1. Depending on the mobility properties, a tag can be at different locations at different times. Therefore, for locating a tag, the readers face five different situations in case of square cell (see figure 7.2) and four different situations in case of hexagonal cell (see figure 7.1). Using square area as a cell, a tag can be located by at most four readers. However using hexagonal cell, a tag can be located by at most three readers. The use of hexagon cells reduces the communication traffic of the channel between the reader and the server, as fewer readers will reply in response of the server's search request.

2. $|Adj(R_i)|$ in a hexagonal cell organization is less than that in square cell organization. At a time, at most 7 readers know about a tag in case of hexagonal cell organization (see figure 7.1). But in square cell organization at most 9 readers know about a single tag (see figure 7.2). In our system, whenever a tag changes its location from one cell to another, at most 3 readers have to insert the ownership information of the particular tag into their contact list and at most 3 readers have to erase the information. While in previous system, at most 5 readers do insertion and at most 5 readers do deletion. Thus, our system ensures more scalability.

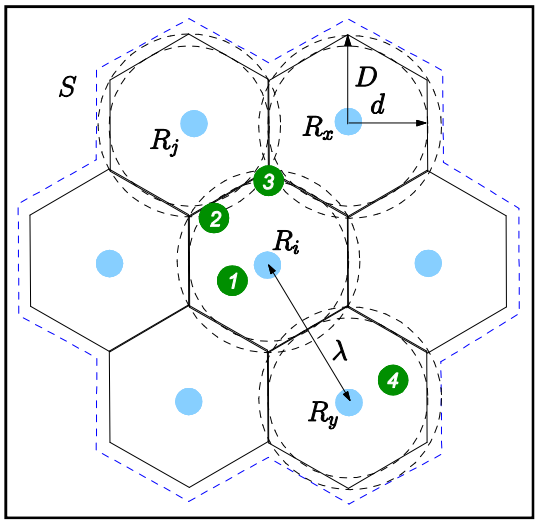


Figure 7.1 The coverage of a set of readers while cell is hexagonal. Number denotes different tag location situations. 1 denotes only R_i locates the tag. 2 denotes both R_i and R_j locate the tag. In position 3, R_i, R_j and R_x detect the tag. 4 indicates R_i cannot locate the tag.

3. Radio frequency is omnidirectional. So, a cell should be circular. But, practically circular cell will not be possible. A hexagon has more resemblance to a circle than a square. In

fact, a precise hexagonal pattern is not used in all instances due to topographical limitations, local signal propagation conditions, and practical limitations on signal antennas.

4. In square pattern the neighboring readers of a reader are not at an equal distance. Some neighbors are at λ distance while others are at $\sqrt{2}\lambda$ distance. In contrast, all neighboring readers in hexagonal pattern are to be at equal distance λ. This property resolves a shortcoming in communication between readers.

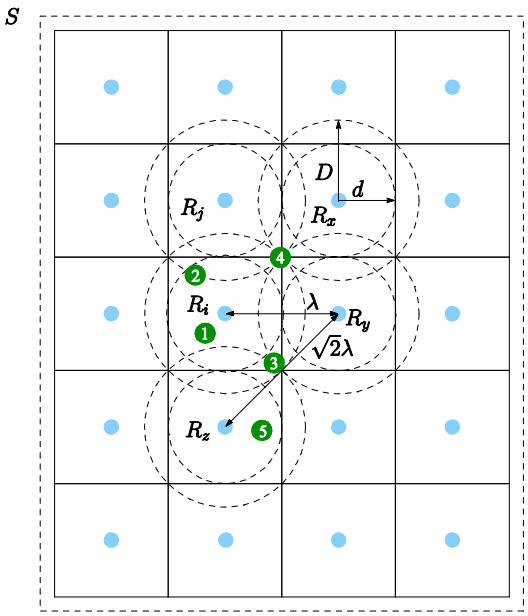


Figure 7.2 Coverage of set of readers while cell is square. The numbers are used to indicate different tag location situations. 1 denotes only R_i locates the tag. 2 denote both R_i and R_j locate the tag. In position 3, R_i, R_y and R_z detect the tag. Location 4 meansR_i, R_j, R_y and R_x locate the tag. 5 indicate R_i cannot locate the tag.

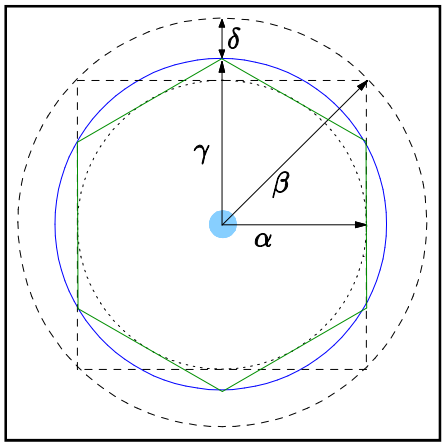


Figure 7.3 Overlapping area of two different cell patterns.

5. The common area between any two cells is referred to as overlapping area. A hexagonal pattern provides a reduced overlapping area. The area is reduced by $\pi(\beta^2 - \gamma^2)$ for a single cell (see figure 7.3). If tags are uniformly distributed in S and there are ρ tags per unit area, then $\rho\pi(\beta^2 - \gamma^2)$ amount of tags within a cell will neither be authenticated nor be found by more than one reader.

The above justifications prove the enhancement of our system that uses hexagonal cell.

7.4. Application of EDSA

Emergency Evacuation System: Safety at the workplace and saving human lives in emergency situations has always been one of the highest priorities in all civilized countries. Fast and efficient evacuation of building complexes, and keeping account of all involved in unpredictable circumstances with hundreds or even thousands of people escaping from danger zones, is an essential component of any emergency system. In the case of emergency, conventional evacuation strategies rely on emergency authority (Fire Brigade, Police etc) to check each and every floor and to direct the personnel to come out of the building in the case of emergency situation. This approach has experienced limited success for safe and effective evacuation operation. A better mechanism or process is needed. Here we focus on some major issues and functionalities of an emergency evacuation system and how these can be accomplished through the usage of EDSA.

The emergency evacuation system must be able to keep track of who is entering and leaving the system on a hands-free basis. It must cover all entrances/exits and handle people on a one-by-one basis or when rushing through in numbers. As there is no time for a personnel to think in which pocket he/she might have left the card to use it to exit the building in emergency case. It must automatically keep track of the whereabouts of all personnel and visitors within the building on 24/7 basis. Actually it has to know more specifically who has entered and if those who entered are still inside.

RFID system can be applied as a solution to the evacuation process and EDSA can be the appropriate architecture to be used in the system. Implementation of more than one ENP or EXP in EDSA, for more than one entrance or exit, is straight forward. To account for every personnel building occupants must have ID card, badge or other cards with embedded RFID tags. Even visitors should be equipped with such type of temporary cards so that they can also be accounted for in case of emergency situation. As a tag needs to be authenticated to enter the system, the ENPs of EDSA can be authorized by back-end server for all possible tags that can enter through them. The set of all possible tags may be comprised the tags for personnel as well as visitors. As all personnel and visitors have to enter the building, i.e., the system, through ENPs, it is straight forward to account for all humans. Those who have entered and who are still inside can be available to back-end server by getting information from contact lists of readers. Indeed, back-end server can come to know about the sparse distribution of people throughout the whole building. The back-end server can keep track of whereabouts of people whenever necessary by executing a search operation with respective tags, because if a tag enters the system, it must be somewhere in the system until its departure.

After the initialization of evacuating process, facility managers and first responders need to track the progress of the whole process in an easy approach. Monitoring can be possible by linking their PDAs or laptop computers with the back-end server. The software must display either the number of personnel left in the building or the names of those not yet accounted for. All these tasks must be updated in real time. Back-end server can provide total numbers of people left in the building, who are leaving the building through EXPs and who have not been accounted for. The software can request a search operation to back-end server which it passes on to all the readers for the respective tags. All the monitoring process including search operations can be done in real time. This allows first responders to know where to target their search and rescue efforts. Even by tagging rescuers before entering the emergency location, the back-end server can track them during evacuation process.

The automated system concept must be based on a 'hands-off' approach and require no user intervention upon entry or exit. In fact, EDSA pursues the hands-off approach. The protocols to facilitate the communication between readers indeed maintain this approach by insertion and deletion of ownership information.

As buildings grow and workplace increase in size, the need for more sophisticated emergency systems grows. Accurate location information is essential to any emergency system and thus the implementation of EDSA is very important for the society. The possibility of damage to the system is beyond scope of this thesis.

7.5. Summary

RFID systems are used in selected industries for quite a few years now, yet there exists many applications of this technology. However, the question as to whether RFID systems will be widely used in the future depends on the strength of privacy protection and the improvement of performance features such as scalability. Unfortunately, there is a tradeoff between allowing scalability and ensuring security. In order to incorporate these two conflicting goals, we propose a hexagonal cell based distributed architecture using RFID tag identification which provides more scalability. In this architecture readers can co-operate with one another through a secure channel for scalable and secure tag identification. To the best of our knowledge, this is the first proposal to integrate a serverless and a severed technique in the same architecture to enhance RFID system scalability. At the end of the chapter, we also illustrate the incorporation of EDSA in a real life example, such as emergency evacuation system, and discuss the capabilities of EDSA to overcome the challenges.

Chapter 8: Monitoring Missing WISP Tags in CRFID Networks

8.1. Introduction

The past decade has seen significant effort and progress towards the original ubiquitous applications. Particularly wireless sensor networks (WSNs) based on mote sensing platforms have been applied to many real-world problems. Remote monitoring applications have sensed animal behavior and habitat, structural integrity of bridges, volcanic activity, and forest fire danger [Hartung06], are to name only a few successes. Due to low power design and careful networking protocols these sensor networks had lifetimes measured in weeks or months, which were generally sufficient for the applications. Despite these successes, WSNs have not led to an approximation of sensing embedded in the fabric of everyday life, where walls, clothes, products, and personal items are all equipped with networked sensors. For this type of deployment, truly unobtrusive sensing devices are necessary. The size and finite lifetime of motes make them unsuitable for these applications. For the last few years, it is argued that Radio Frequency Identification (RFID) technology has a number of key attributes that make it attractive for such applications. RFID is a technology for automated identification of objects and people. But future RFID applications will require tags that can also perform minimal sensing, computation, and storage. One recent extension of RFID, Computational RFID (CRFID), presents exciting possibilities for ubiquitous computing applications. CRFID combines the advantages of RFID with those of sensor networks.

Table 8.1 Comparison of different technologies

	CPU	Sensing	Size (inches)	Range	Power	Lifetime
WSN (Mote)	Yes	Yes	3.0 x 1.3 x .82 (2.16 in^3)	Any	Battery	< 3 yrs
RFID tag	No	No	6.1 x 0.7 x .02 (.08 in^3)	30 ft	Harvested	indefinite
CRFID (WISP)	Yes	Yes	5.5 x 0.5 x .10 (.60 in^3)	10 ft	harvested	indefinite

As discussed, two technologies, wireless sensor networks and RFID, have been widely used to realize real-world applications. But CRFID presents the combination of both of these networks. The comparison of these three technologies is presented in table 8.1.

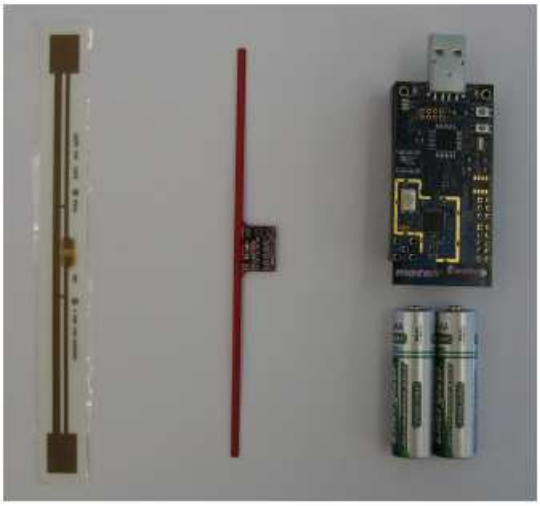
In this chapter, we explore a third class of sensors that aims to provide the best of both worlds: RFID sensor networks based on Wireless Identification and Sensing Platforms (WISPs) [Sample08]. We consider the problem of how to accurately and efficiently monitor a set of WISP tags for missing tags. The task of monitoring for a missing WISP tag within a set of tags can be considered as a tag search approach. This is a special type of tag searching approach where the reader needs to monitor for missing tags and find out the tag that is missing.

8.2. What is WISP?

The Wireless Identification and Sensing Platform (WISP) from Intel Research Seattle [Buettner08, Sample08] is an instance of CRFID. WISPs combine passive UHF RFID technology with traditional sensors. A current WISP is shown alongside a commercial UHF RFID tag and a common wireless sensor node (mote) in figure 8.1. WISPs have the capabilities of RFID tags but also support sensing and computation. Like any passive RFID tag, WISP is powered and read by a standard off-the-shelf EPC “Gen 2” RFID reader, harvesting the power it uses from the reader's emitted radio signals. To an RFID reader, a WISP is just a normal EPC class-1 or gen-2 tag; but inside the WISP, the harvested energy is operating a 16-bit general purpose microcontroller. The microcontroller also has an analog to digital converter within itself. The microcontroller can perform a variety of computing tasks, including sampling sensors, and reporting that sensor data back to the RFID reader.

WISP uses an integrated 802.15.4 radio for communication to talk with reader. WISPs can sense quantities such as light, temperature, acceleration, strain, and liquid level. Though the feasibility of WISPs has been discussed in some research literatures, how to harness many such devices to create a WISP sensor network is till now an open question. In near future, sensor

network will consists of multiple WISPs and one or more readers. Consequently, realizing a full-scale network will require development at both the WISP and the reader end, because new protocols and techniques must be developed unlike those of either RFID or WSNs.



Source: [Buettner09]

Figure 8.1 A standard UHF Class 1 Gen 2 RFID tag, Intel WISP, and Telos Mote (left to right)

While WISPs are currently assembled from discrete components that have a cost of roughly $25, they are intended to be mass manufactured like RFID tags at price points closer to $1 [Buettner09]. One disadvantage of using the WISP tags is that they need to be placed within 1-2 meters of the reader.

8.3. Research Problem of WISP Networks

For simple RFID sensor networks, the data of interest is simply each tag's identity. However, for WISP sensor networks, it is difficult to develop efficient protocols for gathering sensor data that changes over time. With RFID, the reader is able to transmit messages to all the tags and the tags can re-transmit messages to the reader. Currently, WISP tags with new sensor data must wait until they are interrogated by a reader. This increases the likelihood of many WISP tags wanting to use the bandwidth limited channel at the same time when replying to the reader query. However, the standard RFID strategy of identifying and then communicating with

each device is wasteful as only some devices would have relevant data. Because of all these differences, the trivial RFID protocols securing RFID network cannot be applied or even adapted to WISP sensor network.

Let's consider a WSN deployed in a battlefield. Quick response time of sensor network along with high data accuracy and integrity is very important in such networks. A reader might have hundreds of accelerometer WISPs in its field of view. Because all the WISPs share a single reader channel, the update rate per tag would be very low if every tag were simply queried for sensor data sequentially. At any given moment, the reader may want to find out whether all the tags are present in the battlefield or not. The reader may also want to find out the particular WISP tag that is missing from the battlefield.

There are two kinds of methods used to solve this type of problems. One is ALOHA based algorithms and the other one is tree-based algorithms [Fin03]. The ALOHA based algorithms reduce the probability of tag collisions since tags are scheduled to transmit at distinct times. However, with the increase of the number of tags, the identification performance will be deteriorated sharply. We propose to apply Slotted ALOHA based technique to solve this problem.

8.4. Motivation

Let us consider WISP network installed in a hospital to monitor patients who are in ICU (Intensive Care Unit). ICU patients are usually in a very critical situation (i.e. they are out of any kind of movement) and they are kept in ICU for a very small period of time (for example 3-4 days). But patients in ICU need special care. They are treated with highest medical facilities and devices. For these types of patients, one important fact that the doctors look for is the quality and quantity of sleep of each patient. WISP tags have accelerometer and it can be attached with the patients' mattress/bed to monitor for sleep quality. These WISP tags can also be used to collect other information such as, surrounding sound, temperature, air density, identification of patient, etc. In this scenario, the reader installed in the ICU may need to perform two tasks:

Task 1. Monitor for any missing WISP tag in the ICU (it can be an indication of administration error or life threat from any enemy of the patient). The goal of this task is to find out the tag that is missing within the system.

Task 2. Collect data of the WISP tags to monitor for patient's state. The goal of this task is collect the sensor data from all the existing WISP tags of the systems and pass it to the server for further processing.

The technique of determining patient's health and environment status from the collected raw data is done by the server and it is out of the scope of this thesis. Next we investigate the methodologies that can be used to perform the above mentioned two tasks.

Hospital authority could first attach a WISP tag to each object/person to be monitored. Each tag contains a unique *id* which is recorded and stored on a secure server. The authority then deploys a Gen 2 reader to periodically collect all the *id*s from the tags and match them against the *id*s stored on server. This way, the doctor can be immediately notified of any errors. We term this simple approach as *collect all*. However, collect all suffers from two drawbacks.

First, collecting tag *id*s for comparison is time consuming when there are a lot of tags due to presence of collisions. A reader collects *id*s by first broadcasting the number of available time slots. Each tag will independently pick a time slot to reply. When multiple tags pick the same slot, a collision occurs and the reader obtains no information and must repeat the process again. When the set of tags is large, the number of collisions will rise, increasing the data collection time. The increase of data collection time may have an adverse effect on patients' lives since ICU patients needs to be observed continuously. The system response delay of 1 to 2 minute can cause serious vulnerabilities to the patients' lives at ICU.

Second, collect all requires the WISP tags to reply their sensor data values in a second round of message. This increases the communications cost of the system.

In this chapter, we consider the problem of accurately and efficiently monitoring for missing WISP tags. We assume that the gen 2 reader interacts with the tags and passes the

collected information to the server. The server will do further processing and will issue a warning if there are any missing tags. Our approach is unique in a sense that the tags will reply with their identification and data (i.e. sensor data) at the same pass of message. This technique reduces the communication cost drastically when there is large number of tags in the system.

8.5. Existing Works on WISP

The existing literatures on CRFID based security are not much extensive. However, extensive works have been done to secure RFID sensor networks and sensor networks in general. Next we discuss some important literatures on sensor network security and WISP security that are relevant to this chapter.

In the recent years, wireless sensor network security problem [Chan03] has been able to catch the attentions of a number of researchers around the world. Wood et al. [Wood02] studies DoS attacks against different layers of sensor protocol stack. JAM [Wood03] presents a mapping protocol which detects a jammed region in the sensor network and helps to avoid the faulty region to continue routing within the network, thus handles DoS attacks caused by jamming. [Wang06] presents a brief summary of various security schemes of wireless sensor networks proposed so far. Recently, Abdelzaher et al. have started research on sensor networks to investigate network protocols, services, and programming paradigms tailored to the Cyber physical system involving sensors and RFID tags (excluding WISP tags) [Zaher]. However, these techniques are not suitable for WISP sensor networks because of different architecture, operating platform, sensing technique, and new challenges introduced by WISP tags.

So far, the security aspects of WISP sensor network have not been explored in literature extensively since the usage of these tags are still a new technology. However, [Czeskis08] presents an overview of low power wireless security research for WISP enabled RFID network. [Intel] provides information of the entire WISP related literatures that have been proposed so far.

But, most of these literatures focus on the improvement of power aware parameters of WISP sensor network.

In CRFID networks collision occurs when multiple WISP tags try to transmit data to a reader at the same time. This results in the reader being unable to obtain any useful information. Works [Bonuccelli06, Cha05, Lee05, and Micic05] have been done to improve the protocols to reduce collisions, and secure search techniques to isolate particular tags [Tan07] one at a time. While these techniques improve monitoring performance, such solutions are ultimately bounded by the number of tags. Regardless of the protocol used, the RFID reader will still have to isolate each tag at least once to obtain data. Our approach does not require the reader to isolate every tag.

8.6. Proposed Solution

We make the following contributions in this chapter ─

- We propose the notion of tag monitoring which is a new dimension of tag searching for a WISP tag based network.
- We propose a tag monitoring protocol that does not require the reader to collect *id*s from each WISP tag, but is still able to accurately monitor for missing tags.
- Our monitoring technique provides privacy protection by neither broadcasting tag *id*s in public, nor revealing *id*s to the reader.
- Our technique is unique as it allows the WISP tags to reply their sensor data in the same pass of message in which the tag identification data is replied.
- To the best of our knowledge, the tag searching/monitoring technique for the WISP based network is proposed for the first time in this thesis.

8.6.1. Problem definition

In our system, we assume that a server has a group of objects, and a WISP tag with a unique *id* is attached to each object. We refer to this group of objects as a set of tags. A set of tags

once created is assumed to remain static. It means that no tags are added to or removed from the set. We consider this set of tags to be "*intact*" if all the tags in the set are physically present together at the same time.

8.6.2. Protocol goals

The goal of a server is to remotely, quickly, and accurately determine whether a set of WISP tags is intact. The server first instructs a reader to scan all the tags to collect a Bit Record (BR). The server then uses this result to determine whether there are any missing tags. Our protocols succeed if the server is able to determine a set of tags is not intact when any of the tag is missing.

8.6.3. Attack model

The goal of the adversary is to steal RFID tags. The adversary launches the attack by physically removing tags from the set. We do not consider more involved attacks such as "clone and replace". In such an attack, the adversary steals some tags, clones the stolen tags to make replicate tags, and replaces the replicate tags back into the set. Cloning creates replicate tags that are identical to the stolen tags. In this scenario, the server cannot detect any missing tags since the replicate tags are identical to the removed tags. In our proposal, the adversary simply attempts to steal some tags. Once the tags are stolen, the tags are assumed to be out of the range of the reader. Therefore, when a reader issues a query, the stolen tags will not reply.

We denote an adversary is denoted as $\tilde{A}$. The adversary can control a number of readers and tags. Each reader and tag controlled by $\tilde{A}$ is denoted as $\tilde{R}$and $\tilde{T}$, respectively. We assume that all the entities (tags, readers, *TC* including adversary, adversarial readers and adversarial tags) have polynomially bounded resources.

8.6.4. Preliminaries

We consider an RFID reader, R, and a set of N WISP tags, $T*$. The *TC* is a trusted party who deploys all the WISP tags and authorizes any reader. For the sake of simplicity we assume that *R* and *TC* communicate through a secure channel. We assume that an RFID reader is able to distinguish the slots with no reply, single reply, or multiple replies. We define these slots as empty slot, single-reply slot, or collision slot respectively. We denote the frame size as *f* and the random number generated by reader/tag as *r*. The server contains a table of tag entries. Each entry of the table contains the corresponding tag *id.* Table 8.2 summarizes the remaining notations.

Table 8.2 Notations for MonAC protocol

Symbol	Meaning
$T*$	Set of RFID tags
R	RFID Reader
r	Random number
N	Number of tags within T_*
$h(.)$	One way hash function
SP	Slot position within frame
BR	Bit Record generated by the reader with the replies of tags
e_dat	Encrypted sensor data
r_dat	Raw sensor data
$reply\_string$	Reply sent by the tags

8.6.5. MonAC (Monitor And Collect) protocol

In this section, we present our *Monitor And Collect* protocol, *MonAC*, where the gen 2 reader is assumed to be always honest. Given a set of WISP tags, *MonAC* returns a Bit Record to the server to check if the set of tags is intact and to let the server collect the sensor values.

In our protocol, we assume that WISP tags resolve collisions using a slotted ALOHA type scheme. The reader first broadcasts a frame size and a random number, (f, r), to all the tags. Each tag uses the random number *r* and its *id* to hash to a Slot Position, SP, between [1, f] where

$$SP = h(id \oplus r) \bmod f$$

In order to raw send the sensor data, r_dat, in the same slot, each tag creates an encrypted version of the sensor data, e_dat , in the following way:

$$e\_dat = h(id) \oplus r\_dat$$

Finally tag send the data, e_dat, in the slot position *SP*. Tags that successfully transmit their data are instructed to keep silent. Tags that pick the same slot to reply will be informed by the reader to retransmit in subsequent rounds where the reader will send a new (f, r). The reader repeats this process until all tag *id*s are collected.

We modify the slot picking behavior used in *collect all* so that instead of having a tag pick a slot and return its *id*, we let the tag reply with the encrypted sensor data value e_dat, signifying the tag has chosen that slot. In other words, instead of the reader receiving

$$\{\cdots \mid id1 \mid 0 \mid id6 \mid collision \mid \ collision \ \mid \cdots\},$$

where 0 indicates no tag has picked that slot to reply, and collision denotes multiple tags trying to reply in the same slot, the reader will receive

$$\{\cdots \mid r\_dat \mid 0 \mid r\_dat \mid collision \mid \ collision \mid \cdots\}.$$

After receiving the replies, the reader can insert a random number, *r*, in the collision slot. From the reply, the reader can generate the bit string

$$BR = \{\cdots \mid r\_dat \mid 0 \mid r\_dat \mid r \mid r \mid \cdots\}.$$

This is more secure since the tag is not returning its *id*, and the sensor data is sent in encrypted form which seems purely random to the adversary. Our protocol exploits the fact that a low cost RFID tag picks a reply slot in a deterministic fashion. Thus, given a particular random number r and frame size f, a tag will always pick the same slot to reply. The server knows all the *id*s in a set, as well as the parameters (f, r). Therefore, the server will be able to determine the resulting *BR* for an intact WISP tag set ahead of time. However, the server will know that it is supposed to get a random number in the collision slots and random number alike sensor data in other slot positions where corresponding to a tag presence.

From the reply of the reader, the server can generate a new Bit Record, BR_{new},

$$BR_{new} = \{\cdots \mid 1 \mid 0 \mid 1 \mid 1 \mid 1 \mid \cdots\}$$

The intuition behind our protocol is to let the server pick a (f, r) for the reader to broadcast to the set of tags. The server then compares the BR returned by the reader with the BR_{new} generated from the server's records. A match will indicate that the set is intact. The server can collect the sensor data values corresponding to each tag from the BR returned by the reader.

Algorithm 1: *Interaction between server and reader(R)*

1. Server sends (f, r) to the reader R
2. R executes Algorithm 4
3. All nearby tags executes Algorithm 3
4. Calculate BR_{server} for all tags $T*$
5. Receive BR from R
6. **for** $i = 1 : f$ **do**
7. **if** $(BR\ (i)! = 0)$ **then**
8. **assign** $BR_{new}\ (i) = 1$
9. **if** $(BR_{server} == BR_{new})$ **then**
10. $all\ tags\ are\ present$
11. **else**
12. $tag\ coresponding\ to\ the\ mismatching\ position\ of\ BR\ is\ missing$

Figure 8.2 Algorithm for interaction between server and reader in MonAC protocol

Algorithm 2: *Interaction between WISP tags and reader* (R)

1. Reader broadcasts (f, r) to all tags $T*$
2. Each tag T_i executes Alg. 3
3. Reader executes Alg. 4
4. Reader returns BR to the server

Figure 8.3 Algorithm for interaction between WISP tags and reader in MonAC protocol

Algorithm 3: *Algorithm executed by WISP tags*

1. Receive (f, r) from R
2. **for** Each tag T_i (where i = 1 to N) **do**
3. **compute** $SP_i = h(id_i \oplus r) \bmod f$
4. **compute** $e\_dat_i = h(id_i) \oplus r\_dat_i$
5. **while** R broadcasts Slot Position (SP) **do**
6. **if** ($SP == SP_i$) **then**
7. return e_dat_i to R

Figure 8.4 Algorithm executed by WISP tags in MonAC protocol

Algorithm 4: *Algorithm executed by reader R*

1. **compute** BR of length f
2. Initialize all entries of BR to 0
3. **for** Slot Position $SP = 1$ to f **do**
4. Broadcast SP and listen for reply
5. **if** ($reply\_string != collision$) **do**
6. $BR[SP] = reply$
7. else
8. $BR[SP] = r$
9. return BR to the server

Figure 8.5 Algorithm executed by the reader in MonAC protocol

The reader uses a various (f, r) pair each time he wants to check the intactness of $T*$. The server can either communicate a new (f, r) each time the reader executes *MonAC*, or the server can issue a list of different (f, r) pairs to the reader ahead of time.

Alg. 1 shows the overall interaction between the reader and the server. Each tag in the set executes Algorithm. 2 independently. The reader executes Algorithm. 3 to generate the BR and return it to the server. Notice that unlike the *collect all* method which requires several rounds to collect the tag information, our MonAC algorithm only requires a single round. Furthermore, in Algorithm. 3 Line 7 the tag does not need to return the tag *id* to the reader. Rather the tag sends

the encrypted sensor value (that seems random to the attacker) to inform the reader of its presence. This reduces the communication cost since a second round of messages is not required to send the sensor data to the reader.

8.6.6. Protocol description

In MonAC protocol, there are two phases of operation. One is *Monitor phase* and the other is *Collect phase*. In the Monitor phase, the server monitors for the missing tags. In the Collect phase, the server collects the sensor data for further processing. Next we discuss the details of two phases.

***Monitor Phase*:** The reader first broadcasts a frame size and a random number, *(f, r)*, to all the tags. Each WISP tag T_i uses its own tag id_i and *r* to generate $SP_i = h(id_i \oplus r) \bmod f$. At the same time, each tag calculates its own sensor data, $e\_dat = h(id) \oplus r\_dat$. When the slot position broadcasted by the reader matches with SP_i, tag T_i replies e_dat in that slot position to the reader. At the time of receiving replies from different tags, the reader checks the content of slot position SP. After receiving replies from all the tags, the reader forms the Bit Record (*BR*) of length *f* (frame size) to transmit to the server. Initially reader assigns 0 to all the slot positions. However, the reader stores $reply\_string$ in those slot positions where it receives a reply. The reader stores a random number in the slot position where it receives a collision. This technique of bit assignment allows our search protocol to be secured against some major attacks which we will discuss in next section. The BR is then transmitted to the server. We assume that the channel between the server and the reader is secure. We also assume that the frame size (*f*) is large enough and there are more slot positions within the frame than total number of tags (i.e. $f > N$).

The server calculates Bit Record, BR_{server}, for all the tags ahead of time. After receiving BR from the reader, the server stores 1 in those positions of BR where there is no 0. Let this new Bit Record be BR_{new}. Next, the server compares between BR_{new} and BR_{server}. If these two Bit Records do not match, the server becomes sure that at least one of the tags is missing. The server

can find out the slot position of the missing tag. The server can look up the table to find out the id of the missing tag corresponding to mismatched slot position.

Collect Phase: Collect phase is executed by the server after Monitor phase is over. In this phase, the server collects sensor data for all the existing tags. The server determines the raw sensor data from the $reply\_string$ corresponding to each tag. The $reply\_string$ is an encrypted form of the raw sensor data. However only the server can determine the correct raw sensor value as the server knows ids for different tags. Therefore, the server can compute the hash of the id, i.e. $h(id)$. Then the server can XOR the hash, $h(id)$, with the e_data to collect the r_data.

8.7. Protocol Analysis

In this section, we analyze our proposed search protocol against different types of attacks.

8.7.1. Security analysis

Privacy Preservation: Our protocol can preserve the privacy of individual WISP tag. The adversary is not able to find out the original sensor data. Each tag replies with an encrypted sensor data, $h(id)\oplus r\_dat$ which can be decrypted only by the server. Since the server only knows the id of different tags, only it can compute the hash value. Therefore, none but the server can decrypt the encrypted sensor data to collect the raw data.

***Tracking*:** MonAC is resistant against tracking. Let an adversary $\tilde{A}$ eavesdrops on the transaction between a reader *R* and tags. So he/she knows the queries and replies. But he/she will not be able to reverse compute the replies or learn the query but the adversary can certainly be sure that a monitoring has taken place. However, the attacker cannot be able to figure out which tag replied in which slot. Since outputs of all tags will seem to be pure random to the adversary.

***Eavesdropping*:** Here $\tilde{A}$ observes all the queries between a reader and tags. And his/her goal is to use the data to impersonate a fake reader R or a fake tag T_j. Our protocol is powerful against this attack. In our protocol $\tilde{A}$ will not be able to find out the expected reply of the tags.

$\tilde{A}$ will not be able to find out any relation between the Slot Positions and tag replies. In each monitoring pass, all tags will pick a different slot based on the random number sent by the reader. $\tilde{A}$ can only observe the data send by the reader and the tags. But $\tilde{A}$ will not be able to link the queries of two parties. $\tilde{A}$ will not be able to decrypt or even replay the messages. Therefore, $\tilde{A}$ cannot impersonate R or T_j. Therefore by eavesdropping $\tilde{A}$ cannot launch a replay attack by using previous values.

8.8. Summary

In this chapter, we introduced the concept of WIPS tags. We discussed some specific applications of WISP based networks (i.e. ICU of a hospital) and try to provide security solutions for them. We considered a unique issue of CRFID based systems, the problem of monitoring for missing WISP tags. We proposed a secure protocol to monitor for missing tags and also for collecting different sensor values of WISP tags. To the best of our knowledge, this is the first proposal to address the tag monitoring issue for WISP based networks.

Chapter 9: Conclusions and Future Works

In this chapter, we summarize the contributions of the thesis and identify some future research directions.

9.1. Research Achievements

RFID technology is increasingly being deployed in diverse applications ranging from inventory management to anti-counterfeiting protection. Nonetheless, RFID tags have yet to supplant the ubiquitous barcode found on almost every grocery product. This slow adoption is partly due to the security and privacy concerns over the pervasive deployment of RFID tags. This security and privacy concerns are mostly addressed by RFID authentication protocols. However, the aim of this thesis was to address the security and scalability challenges of RFID tag searching and to devise new solutions. Next, we summarize our contribution in this thesis.

- **Attack Summary:** This thesis focuses on RFID search protocols that ensure strong security and scalability. We summarized all the possible attacks that can be launched against RFID systems.

- **Security and Performance Requirements:** We addressed the security and performance requirements that should be guaranteed by RFID protocols to protect against the major security attacks.

- **Secure Serverless Search Protocol (S^3PR):** We proposed a lightweight, secure, and serverless search protocol for RFID systems. The unique feature of this protocol is that it is serverless and it is not vulnerable to single point-of-failure. This protocol requires the tags to be able to compute hash function and generate pseudo random numbers.

- **Secure Scalable Search Protocol (S-Search):** We proposed a secure and scalable RFID tag search protocol for large scale RFID system using Slotted ALOHA based technique. This is a highly scalable search protocol that can be used in large scale RFID systems. This

protocol is very lightweight since it requires the tags to be able to perform hash function and XOR operation.

- **Enhanced Distributed Scalable Architecture (EDSA):** We addressed the tradeoff between scalability and security. From this perspective, we proposed a hexagonal cell based distributed scalable architecture for RFID tag searching in an emergency evacuation system. We compared our architecture with its prior work and proved that our hexagonal cell structure increases the performance of the RFID system.

- **Monitor and Collect Protocol (MonAC):** We introduced the concept of monitoring missing tags. We propose a new dimension of tag searching, i.e. tag monitoring technique (MonAC) for a WISP tag based network. MonAC protocol does not require the reader to collect *id*s from each WISP tag but it is still able to accurately monitor for missing WISP tags.

9.2. Future Directions

- For S^3PR protocol, in future, we plan to simulate the protocols with a large number of tags to see how it performs. We are also interested in finding the lower bounds for the tag's computational requirements for secure RFID communications.

- For S-Search protocol, we plan to extend our protocol to search multiple RFID tags simultaneously.

- This thesis only considers RFID protocols that can perform hash function and can generate random numbers. However, there are tags that do not have such capability. So designing secure search protocols for those tags is also desirable.

- There could be many attacks on RFID systems that we have not addressed in this thesis. Thus, further study of such protocols and possible attacks on them would be desirable.

- We have assumed that the channel between the back-end server and the reader is secure. Hence, we have not dealt with security threats arising on that channel. However, in some

applications, server-reader communications may be insecure, e.g. they may use a wireless channel. Thus, secure search protocols over this channel should be studied further.

- In future, we would like to provide formal security proofs for the protocols proposed in this thesis.

- We would also like to perform simulation in future to investigate a feasible *Spatial Density* for the MonAC protocol. We would like to determine whether the reader can maintain reasonable WISP motion detection rates even when large numbers of tags are active in front of it.

BIBLIOGRAPHY

[Abramson70] Abramson, N., (1970).*The ALOHA system - another alternative for computer communications*. In Proceedings of the AFIPS Conference, Vol. 37, ACM Press. NY, USA. pp. 295–298.

[Ahamed08a] Ahamed, S. I., and Rahman, F., and Hoque, M. E., et al. (2008). *Secured tag identification using EDSA (enhanced distributed scalable architecture)*. In Proceedings of the 2008 ACM Symposium on Applied Computing (SAC 08), ACM Press. NY, USA. pp. 1902-1907.

[Ahamed08b] Ahamed, S. I., Rahman, F., Hoque, E., Kawsar, F., and Nakajima, T., et al. (2008). *S3PR: Secure serverless search protocols for RFID.* In Proceedings of the International Conference on Information Security and Assurance (ISA 08), IEEE, IEEE Computer Society Press. New York, USA. pp. 187-192.

[Ahamed08c] Ahamed, S. I., Rahman, F., and Hoque, Kawsar, E. F., and Nakajima, T., et al. (2008). *YA-SRAP: Yet another serverless RFID authentication protocol*. In Proceedings of the International Conference on Intelligent Environment (IE 08), IEEE, IEEE Computer Society Press. New York, USA. pp. 1-8.

[Ahamed08d] Ahamed, S. I., Rahman, F., and Hoque, Kawsar, E. F., and Nakajima, T., et al. (2008). *Secure and efficient tag searching in RFID systems using serverless search protocol*. In International Journal of Security and Its Applications (IJSIA), Vol.2, No.4. pp. 57-66.

[Avoine05] Avoine, G., and Oechslin., P., et al. (2005). *A Scalable and provably secure hash based RFID protocol*. In Proceedings of the International Workshop on Pervasive Computing and Communication Security (PerSec 05), IEEE, IEEE Computer Society Press. . New York, USA. pp. 110–114.

[Bocchetti08] Bocchetti, S., (2008). *Security and Privacy in RFID Protocols*. PhD Thesis.

[Bonuccelli06] Bonuccelli, M. A., Lonetti, F., and Martelli, F., et al. (2006). *Tree slotted ALOHA: a new protocol for tag identification in RFID networks*. In Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 06), IEEE, IEEE Computer Society Press. New York, USA. pp. 603-608.

[Bringer06] Bringer. J., Chabanne, H., and Emmanuelle, D., et al. (2006). *HB++: a lightweight authentication protocol secure against some attacks*. In International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 06), IEEE, IEEE Computer Society Press. pp. 28-33.

[Buettner08] Buettner, M., Greenstein, B., Sample, A., Smith, J. R., and Wetherall, D., et al. (2008). *Revisiting smart dust with RFID sensor networks*. In Proceedings of the Workshop on Hot Topics in Networks, ACM Press. NY, USA. Available at: web.media.mit.edu/~jrs/2008-hotnets-wisp.pdf

[Buettner09] Buettner, M., Prasad, R., Philipose, M., and Wetherall, D., et al. (2009). *Recognizing daily activities with RFID-based sensors*. In Proceedings of the International conference on Ubiquitous computing (Ubicomp 09), ACM Press, New York, NY, USA. pp. 51- 60.

[Cai09] Cai, S., Li, Y., Li, T. and Deng, R., et al. (2009). *Attacks and improvements to an RFID mutual authentication protocol and its extensions*. In Proceedings of the Conference on Wireless Network Security (WiSec 09), ACM Press. NY, USA. pp. 51-58.

[Cha05] Cha, J. R. and Kim, J. H., et al. (2005). *Novel anti-collision algorithms for fast object identification in RFID system*. In Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS 05), IEEE, IEEE Computer Society Press. NY, USA. pp. 63 - 67.

[Chan03]Chan, H. and Perrig, A., et al. (2003). *Security and privacy in sensor networks*. In IEEE Computer Magazine. IEEE, IEEE Computer Society Press. NY, USA. pp. 103–105.

[Chien07] Chien, H.Y., and Chen, C. H., et al. (2007). *Mutual authentication protocol for RFID conforming to epc class 1 generation 2 standards*. In Computer Standards Interfaces, Vol. 29, Ed. 2. pp. 254-259.

[Choi04] Choi, H. S., Cha, J. R. and Kim, J. H., et al. (2004). *Fast wireless anti-collision algorithm in ubiquitous id system*. In Proceedings of the Vehicular Technology Conference, IEEE, IEEE Computer Society Press. NY, USA. pp. 4589–4592.

[Cidon88] Cidon, I., and Sidi, M., et al. (1988). *Conflict multiplicity estimation and batch resolution algorithms*. In IEEE Transaction of Information Theory, Vol. 34, Ed. 1. pp. 101-110.

[Claas-1] Class-1 Generation-2 UHF air interface protocol standard version 1.0.9: "Gen2". Available at: http://www.epcglobalinc.org/standards/.

[Conti07] Conti, M., Pietro, R. D., Mancini, L. V., and Spognardi, A., et al. (2007). *RIPP-FS: an RFID identification, privacy preserving protocol with forward secrecy*. In Proceedings of the International Workshop on Pervasive Computing and Communication Security (PerSec 07), IEEE, IEEE Computer Society Press. NY, USA. pp. 229-234.

[Cui07] Cui, Y., Kobara, K., Matsuura, K., and Imai, H., et al. (2007). *Lightweight Asymmetric Privacy-Preserving Authentication Protocols Secure against Active Attack.* In Proceedings of the International Workshop on Pervasive Computing and Communication Security (PerSec 07), IEEE, IEEE Computer Society Press. NY, USA. pp. 223-228.

[Czeskis08] Czeskis, A., Koscher, K., Smith, J. R., and Kohno, T., et al. (2008). *RFIDs and Secret Handshakes: Defending Against Ghost-and-Leech Attacks and Unauthorized Reads with Context-Aware Communications.* In Proceedings of the Conference on Computer and Communications Security (CCS 08), ACM Press. NY, USA. pp. 479-490.

[Feldhofer06] Feldhofer, M., and Rechberger, C., et al. (2006). *A case against currently used hash functions in RFID protocols*. In Proceedings of the OTM Workshops (1). pp. 372-381. Available at: http://dx.doi.org/10.1007/11915034_61

[Fin03] Finkenzeller, K., (2003). *RFID Handbook: Fundamentals and applications in contactless smart cards and identification*. John Wiley and Sons, NY, USA.

[Gilbert05] Gilbert, H., Robshaw, M., and Sibert, H., et al. (2005). *An active attack against HB+ – a provably secure lightweight authentication protocol*. Manuscript, 2005.

[Hartung06] Hartung, C., Han, R., Seielstad, C., and Holbrook, S., et al. (2006). *Firewxnet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments*. In Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys 06). ACM Press. NY, USA. pp. 28-41.

[Henrici04] Henrici, D., and MÄuller, P., et al. (2004). *Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers*. In Proceedings of the International Workshop on Pervasive Computing and Communication Security (PerSec 04), IEEE, IEEE Computer Society Press. NY, USA. pp. 149-153.

[Hernandez01] Hernandez, P., Sandoval, J., Puente, F., and Perez, F., et al. (2001). *Mathematical model for a multiread anticollision protocol*. In IEEE Pacific Rim Conference on Communications, Computers and signal Processing (PACRIM 02). Vol. 2. pp. 647 - 650. DOI: 10.1109/PACRIM.2001.953716

[Hopper00] Hopper, N., and Blum, M., et al. (2000). A secure human-computer authentication scheme. Tech. Rep. CMU-CS-00-139, Carnegie Mellon University, 2000.

[Hopper01] Hopper, N. J., and Blum, M., et al. (2001). *Secure human identification protocols*. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 01), Springer-Verlag. pp. 52–66.

[Hoque09] Hoque, M. E., and Rahman, F., and Ahamed, S. I., and Park, J. H., et al. (2009). *Enhancing privacy and security of RFID system with serverless authentication and search protocols in pervasive environments.* In Springer Wireless Personal Communication. http://dx.doi.org/10.1007/s11277-009-9786-0

[Hoque10] Hoque, M. E., and Rahman, F., and Ahamed, S. I., et al. (2010). *S-search: finding RFID tags using scalable and secure search protocol*. In Proceedings of the ACM Symposium on Applied Computing (SAC 10). ACM Press. NY, USA. pp. 439-443.

[Intel] Intel Research Seattle, http://seattle.intel-research.net/wisp/#pub

[Juels05a] Juels, A., and Weis, S. A., et al. (2005). *Authenticating pervasive devices with human protocols*. In Advances in Cryptology, CRYPTO 2005, pp. 293-308.

[Juels05b] Juels, A., (2005). *RFID security and privacy: A research survey*. RSA Laboratories.

[Juels05c] Juels, A., Molnar, D. and Wagner, D., et al. (2005). *Security and Privacy Issues in E-passports*. In Proceedings of the Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm 05), IEEE, IEEE Computer Society Press. NY, USA. pp. 74-88.

[Juels06] Juels, A., and Weis, S., et al. (2006). *Defining strong privacy for RFID*. In Proceedings of the Cryptology ePrint Archive, Report 2006/137, IACR.

[Kodialam06] Kodialam, M., and Nandagopal, T., et al. (2006). *Fast and reliable estimation schemes in RFID systems*. In Proceedings of the International Conference on Mobile Computing and Networking (MobiCom 06), ACM Press. NY, USA. pp. 322-333.

[Kulseng09] Kulseng, L., Yu, Z., Wei, Y., and Guan, Y., et al. (2009). *Lightweight secure search protocols for low-cost RFID systems*. In Proceedings of the Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems. IEEE, IEEE Computer Society Press. Washington, DC, USA. pp. 40-48.

[Laurie07] Laurie, A., (2007). *Practical attacks against RFID*. In Network Security. pp. 4-7.

[Lee05] Lee, S. R., Joo, S. D., and Lee, C. W., et al. (2005). *An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification*. In Proceedings of the International ICST Conference on Mobile and Ubiquitous Systems (Mobiquitous 05), ACM Press. NY, USA. pp. 166 - 172.

[Lee10] Lee, Y. K., Batina, L., Singel, D. and Verbauwhede, I., et al. (2010*). Low-cost untraceable authentication protocols for RFID.* In Proceedings of the Conference on Wireless network security, IEEE, IEEE Computer Society Press. NY, USA. pp. 55-64.

[Li07] Li, Y., and Ding, X., et al. (2007). *Protecting RFID communications in supply chains*. In Proceedings of the Symposium on Information, Computer and Communications Security, (ASIACCS 07). ACM Press, NY, USA. pp. 234-241

[Mayes09] Mayes, K., Markantonakis, K., and Hancke, G. et al. (2009). *Transport ticketing security and fraud controls*. In Elsevier Information Security Technical Report, Vol. 14, Ed. 2. pp. 87-95,

[Metcalfe75] Metcalfe, B., (1975). *Steady-state analysis of a slotted and controlled ALOHA system with blocking*. In SIGCOMM Computing Communication. Vol. 5, Ed. 1. PP. 24–31.

[Micic05] Micic, A., Nayak, A., Simplot-Ryl, D., and Stojmenovic, I., et al. (2005). *A hybrid randomized protocol for RFID tag identification.* In Proceedings of the International Workshop on Next Generation Wireless Networks (WoNGeN 05), IEEE, IEEE Computer Society Press. New York, USA. pp.147 - 154.

[Molnar04] Molnar, D., and Wagner, D., et al. (2004). *Privacy and security in library RFID: Issues, practices, and architectures*. In Proceedings of the Conference on Computer and Communications Security (CCS 04), ACM Press. Washington DC, USA, pp. 210-219.

[Ohkubo03] Ohkubo, M., Suzuki, K., and Kinoshita, S., et al. (2003). *Cryptographic Approach to "Privacy-Friendly" Tags*. In Proceedings of the RFID Privacy Workshop, MIT. MA, USA.

[Pervasive1] Pervasive Computing definition, URL: http://www.parliament.vic.gov.au/sarc/EDemocracy/Final_Report/Glossary.htm

[Pervasive2] Pervasive Computing framework, URL: http://framework.v2.nl/archive/archive/node/text/default.xslt/nodenr-156647

[Piramuthu06] Piramuthu, S., (2006). *HB and related lightweight authentication protocols for secure RFID tag/reader authentication*. In *CollECTeR 2006*.

[Rieback06] Rieback, M., Crispo, B., and Tanenbaum, A., et al. (2006). *Is your cat infected with a computer virus?* In Proceedings of the International Conference on Pervasive Computing and Communications (PerCom 06), IEEE, IEEE Computer Society Press. Washington, DC, USA. pp. 169-179.

[Rieback07] Rieback, M., Crispo, B., and Tanenbaum, A., et al. (2006). *The evolution of RFID security*. In the Journal of IEEE Pervasive Computing. Vol 5, Num. 1. pp. 62-69.

[Sample08] Sample, A. P., Yeager, D. J., Powledge, P. S., Mamishev, A.V., and Smith, J. R., et al. (2008). *Design of an RFID-based batteryfree programmable sensing platform*. In

Proceedings of the IEEE Transaction on Instrumentation and Measurement. IEEE, IEEE Computer Society Press. New York, USA.

[Schoute83] Schoute, F. C., (1983). *Dynamic frame length ALOHA*. In IEEE Transactions on Communications, Vol. 31, IEEE, IEEE Computer Society Press. New York, USA. pp. 565–568.

[Seo06a] Seo, Y., and Kim, K., et al. (2006). *Scalable and untraceable authentication protocol for RFID*. In Proceedings of the International Workshop on Security in Ubiquitous Computing Systems (Secubiq 06), Lecture Notes in Computer Science, Seoul, Korea.

[Seo06b] Seo, Y., Lee, H., and Kim, K., et al. (2006). *A lightweight authentication protocol based on universal re-encryption of RFID Tags*. Available at: caislab.icu.ac.kr/Paper/paper_files/2006/CISC_1115_Youngjoon.pdf

[Sheng08] Sheng, B., Tan, C. C., Li, Q. and Mao, W., et al. (2008). *Finding popular categories for rfid tags*. In Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc 08), ACM Press. NY, USA. pp. 159-168.

[Solanas07] Solanas, A., Domingo-Ferrer, J., Martínez-Ballesté, A., and Daza, V., et al. (2007). *A distributed architecture for scalable private RFID tag identification*. In Computer Networks, Elsevier, Vol. 51, Ed. 9. pp. 2268-2279.

[Song09] Song, B., (2009). *RFID authentication protocols using symmetric cryptography*. Thesis.

[Syverson94] Syverson, P., (1994). *A taxonomy of replay attacks*. In Proceedings of the Computer Security Foundations Symposium (CSF 94). IEEE, IEEE Computer Society Press. New York, USA. pp. 187-191.

[Tan07] Tan, C. C., Sheng, B., and Li, Q., et al. (2007). *Severless search and authentication protocols for RFID.* In Proceedings of the International Conference on Pervasive Computing and Communications (PerCom 07), IEEE, IEEE Computer Society Press. New York, USA. pp. 3-12.

[Tan08] Tan, C. C., Sheng, B., and Li, Q., et al. (2008). *How to monitor for missing RFID tags*. In Proceedings of the International Conference on Distributed Computing Systems. pp. 295–302.

[Tsudik06] Tsudik, G., (2006). *YA-TRAP: yet another trivial RFID authentication protocol*. In Proceedings of the International Conference on Pervasive Computing and Communications (PerCom 06), IEEE, IEEE Computer Society. New York, USA. pp.-643.

[Vajda03] Vajda, I. and Butty´an, L., et al. (2003). *Lightweight authentication protocols for low-cost RFID tags*. In Proceedings of the Second Workshop on Security in Ubiquitous Computing (Ubicomp ‘03). Seattle, WA, USA

[Vogt02] Vogt, H., (2002). *Efficient object identification with passive RFID Tags*. In Proceedings of the International Conference on Pervasive Computing, Springer-Verlag. pp. 98-113.

[Wang06] Wang, Y., Attebury, G., and Ramamurthy, B., et al. (2006). *A survey of security issues in wireless sensor networks*. In IEEE Communications Surveys and Tutorials, Vol. 8.

[Weis03] Weis, S.A., Sarma, S.E., Rivest, R.L. and Engels, D.W., et al. (2003). *Security and privacy aspects of low-cost radio frequency identification systems*. In Proceedings of the International Conference on Security in Pervasive Computing (SPC 03), Springer-Verlag. Vol. 2802. pp. 454-469.

[Weiser93] Weiser, M., (1993). *Some computer science problems in ubiquitous computing.* In Communications of the ACM, Vol. 36, No. 7. pp. 75-84.

[Wieselthier89] Wieselthier, J., Ephremides, A., and Michaels, L., et al. (1989). *An exact analysis and performance evaluation of framed ALOHA with capture*. In IEEE Transactions on Communications, IEEE, IEEE Computer Society Press. New York, USA. pp.125-137.

[Wood02] Wood, A. D. and Stankovic, J. A., et al. (2002). *Denial of service in sensor networks*. In Computer Magazine, Vol. 35, Iss. 10, IEEE, IEEE Computer Society Press. pp. 54 - 62.

[Wood03] Wood, A. D. and Stankovic, J. A., et al. (2003). *JAM: A jammed-area napping service for sensor networks*. In Proceedings of the IEEE Real-Time Systems Symposium, (RTSS 03), IEEE, IEEE Computer Society Press. pp. 286-297.

[Zaher] http://www.cs.uiuc.edu/homes/zaher/cyberphysical/sensors.html

Appendix A

Glossary of Terms

Term	Definition
Pervasive computing	Pervasive computing provides an environment where information and services can be accessed remotely from the environment specially through wireless technologies
RFID systems	RFID is an abbreviation of Radio Frequency IDentification. It is a data collection technology that uses electronic tags for storing data.
RFID tags	A microchip attached to an antenna that is packaged in a way that it can be applied to an object. The tag picks up signals from and sends signals to a reader. The tag contains a unique serial number.
Reader	A device used to communicate with RFID tags. The reader has one or more antennas, which emit radio waves and receive signals back from the tag.
Gen 2	The second generation air interface for communication between an RFID reader and tag, administered by EPC global Inc. It deals with the modulation scheme, packet structure, command language and methods for dealing with collision.
WISP	WISP stands for Wireless Identification and Sensing Platform. WISPs have the capabilities of RFID tags, but also support sensing and computing.
Security	Process of creating a computing platform that ensures only allowed actions are performed.
Scalability	A network protocol is said to be scalable if the number of nodes can be significantly increased without imposing an unacceptable workload on any entity in the network.
Anonymity	Anonymity is the state of not being identifiable within a set
Lightweight Cryptography	Cryptographic operations that require low computational and processing power to be performed
Serverless System	An RFID system consisting of tags and readers but without a central database
Eavesdropping	Eavesdropping is the act of secretly listening to the private conversation between two parties
Nonce	A random number that never repeats its value