The Pennsylvania State University

The Graduate School

Department of Computer Science and Engineering

SECURITY AND PRIVACY IN LOW COST RADIO FREQUENCY IDENTIFICATION

A Thesis in

Computer Science and Engineering

by

Janani Murthy

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2009

The thesis of Janani Murthy was reviewed and approved* by the following:

Raj Acharya
Professor and Head of Department
Computer Science and Engineering
Thesis Advisor

Sencun Zhu
Assistant Professor
Computer Science and Engineering

Guohong Cao
Professor
Computer Science and Engineering

*Signatures are on file in the Graduate School

ABSTRACT

Automatic Identification (AutoID) technology has introduced innovative ways to collect, manage information and monitor everything from hospital patients and livestock to library books. Radio Frequency Identification (RFID) is the burgeoning AutoID becoming a popular tool in manufacturing, supply chain management and retail inventory control. Optical barcodes, another common automatic identification system, have been used for packaging on consumer items for years. Due to advances in silicon manufacturing technology, RFID costs have dropped significantly and in the near future, low-cost RFID electronic product codes may be a practical replacement for optical barcodes on consumer items. Unfortunately, the universal deployment of RFID devices in consumer items exposes new security and privacy risks. This thesis presents an introduction to RFID technology, identifies several potential threats to security and privacy in different authentication protocols, and proposes a new authentication protocol for incorporating security and privacy features in low cost RFID communication. I demonstrate that the proposed protocol provides a secure framework for RFID communication in addition to the fact that it does not cause any additional overhead in the singulation time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This work would not have been possible without the support and encouragement of my advisor, Dr. Raj Acharya, who introduced me to the fascinating world of RFID. His motivation and suggestions have guided me in thinking of various aspects of this work. I also thank Dr. Senchen Zhu and Dr. Guohong Cao for being a part of my thesis committee.

I thank my parents, fiancée and my brother for constantly encouraging me throughout my masters at the Academy. It is to them that I dedicate this work. Finally, I thank my friends who were always there beside me through all the tough times.

Chapter 1

Introduction

Radio Frequency Identification

Radio Frequency Identification (RFID) is a fairly new wireless technology that uses radio signals for automatic identification of objects and collecting data about them. The technology involves remotely storing and retrieving data using small devices called RFID tags that are attached to the objects. RFID tags are envisioned as a replacement of bar codes because of a number of important advantages over the older bar-code technology as discussed in later sections. Apart from the fact that RFID tags are small enabling them to be implanted within objects, identification by frequency allows objects to be read in large numbers without the need of visual contact.

RFID technology dates back to more than sixty years ago during Second World War when The Royal Air Force used it to distinguish allied aircrafts from enemy aircrafts by fitting their aircrafts with Radio transponders that would respond when interrogated. RFID has gained importance ever since and is emerging as a ubiquitous technology for automatic identification. Benefits of RFID solutions that make it favorable for use in inventorying, tracking etc can be summarized as follows:

1. Lower costs and higher productivity: RFID applications automate the collection of information about the location of objects with greater accuracy, speed and lower costs compared to manual methods.

2. Increased revenues: By tracking the sales of different products, the stocks and their production rate can be regulated thus reducing losses.

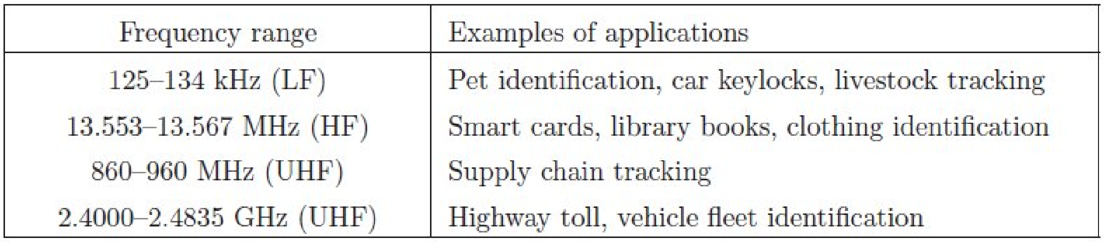
3. Improved quality: Contact-less technology offers quick, easy and more reliable scanning than the legacy schemes like bar codes.

4. Accurate and relevant information: As RFID can be used to obtain real-time data when stocks are moved around; up-to-date management information is available to stores for planning and operational management purposes.

These advantages make RFID a convenient technology that can be widely used in different areas. Some of the popular areas where RFID is being broadly used are described in the next section.

RFID Applications

Supply chain management is an area where RFID has extensive applications. Management of stock and inventories in shops and warehouses is a prime domain for low cost RFID tags. Wal-Mart requires suppliers to attach electronic tags in pallets and packing cases that are delivered. Fast-lane and E-Z pass road toll system uses RFID. The RFID is attached to the wind shield of vehicles and is read by a reader placed in the toll systems. Animal Identification is yet another area where RFID proves very helpful. An RFID tag attached to an animal enables tracking of the animal's location and appropriate sensors embedded can be used to measure health conditions. Homeland security's current concerns have prompted significant efforts on their part to implement electronic passports embedded with RFID tags. The Pin number printed on the cover is read by the reader to identify the passport. Metallic anti-skimming material added in the cover protect from skimming of passports as discussed in chapter 3. FDA is looking into RFID to secure pharmaceutical supply chain to counterfeit drugs. RFID tags can be classified into LF (low frequency), HF (High frequency), Ultra high frequency (UHF) on the basis of the radio frequency they operate in. Figure 1-1 describes the applications of tags in these frequency ranges.



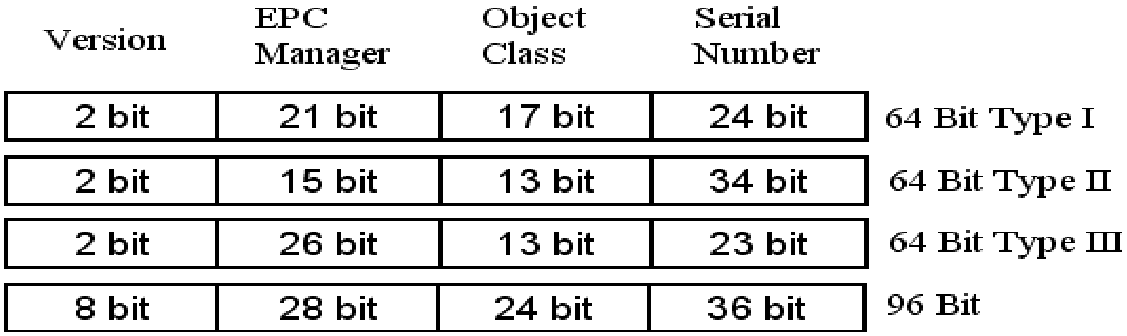
Frequency range	Examples of applications
125–134 kHz (LF)	Pet identification, car keylocks, livestock tracking
13.553–13.567 MHz (HF)	Smart cards, library books, clothing identification
860–960 MHz (UHF)	Supply chain tracking
2.4000–2.4835 GHz (UHF)	Highway toll, vehicle fleet identification

Figure **1-1**: Main Frequency ranges used in RFID and application of tags in those ranges.

Low cost RFID tags and Electronic Product Code

Tags can be manufactured with a wide range of data carrying capacities and with different processor capabilities on the tag. This calls for a standardization of the format of codes stored on the tags and the rules used to query and access data on the tags. Thus, the RFID user community have come together to create a standard for data stored in the tags. EPC Tag data standard as described below defines what information should be held on an EPC compliant RFID tag and the binary format that the information should be held in.

The Electronic Product Code (EPC) is a family of coding schemes created as an eventual successor to the bar code. The EPC was created as a low-cost method of tracking goods using RFID technology. It is designed to meet the needs of various industries, while guaranteeing uniqueness for all EPC-compliant tags. The attractiveness of EPC tags over barcodes is twofold. Firstly, EPC tags transmit information over short distances to RFID readers automatically using radio frequency. They do not require line-of-sight or physical contact to scan like barcode scanners. This eliminates manual intervention. A second benefit of EPC is that they were designed to identify each item manufactured, as opposed to just the manufacturer and class of products, as bar codes. The unique identifier could serve as a pointer to a database that contains information about the product in detail. The EPC accommodates existing coding schemes and defines new schemes where necessary. The EPC was the creation of the MIT Auto-ID Center, a consortium of over 120 global corporations and university labs. The EPC system is currently managed by EPCglobal, Inc. Every tag is identified by the EPC which contains information about the product. The structure of an EPC is illustrated in the Figure 1-2. All EPC numbers contain a header identifying the encoding scheme that has been used. This in turn dictates the length, type and structure of the EPC. EPC encoding schemes contain a serial number called the EPC code which can be used to uniquely identify objects.



Version	EPC Manager	Object Class	Serial Number	
2 bit	21 bit	17 bit	24 bit	64 Bit Type I
2 bit	15 bit	13 bit	34 bit	64 Bit Type II
2 bit	26 bit	13 bit	23 bit	64 Bit Type III
8 bit	28 bit	24 bit	36 bit	96 Bit

Figure **1-2**: Electronic Product Code

96 bit EPC is the standard for data formats on RFID tag applications that replace the barcodes. It can uniquely label all products for the next 1,000 years. The 96 bit code is made up of:

1. A version number (8 bits) for the tag type. e.g. 96 bit EPC class 1

2. An EPC manager number (28 bits) defining who is responsible for administering the tag code, e.g. “ABC Soft drinks ltd”

3. The object class (24 bits) specifies the type of product the RFID tag is attached to, e.g. “6 pack cola diet drink”

4. A unique identifier (36 bits) that together with the rest of the EPC code uniquely identifies the tag and the object it is attached to.

EPC users will have access to the EPC Discovery Service, an aggregate database of tag collected from independent readers. Anyone with access to EPC Discovery service can monitor or track the movement of a particular RFID-tagged item. Commercial information good producers will likely use the EPC format on their RFID tags. EPC works with ONS and PML. ONS (Object Naming service) links the EPC of a tag with associated information about the tag. It works very much like the DNS (Domain Name Service) of the World Wide Web providing a lookup table for translating a unique EPC code into an entry providing additional information about the tag. PML (Product Marksup Language) is a specification that provides a collection of common, standardized vocabularies to represent and distribute information about EPC network enabled objects.

Chapter 2

RFID System

RFID System Architecture

The RFID system is an information tracking system that consists of 3 main components namely the RFID backend infrastructure/database (B), RFID reader (R) and RFID tag (T) as shown in Figure 2-1. The tag is the identification device attached to the object to be tracked. The reader recognizes the presence of tags and issues commands or queries using radio frequency signals to read them and then stores the data collected in the backend database.

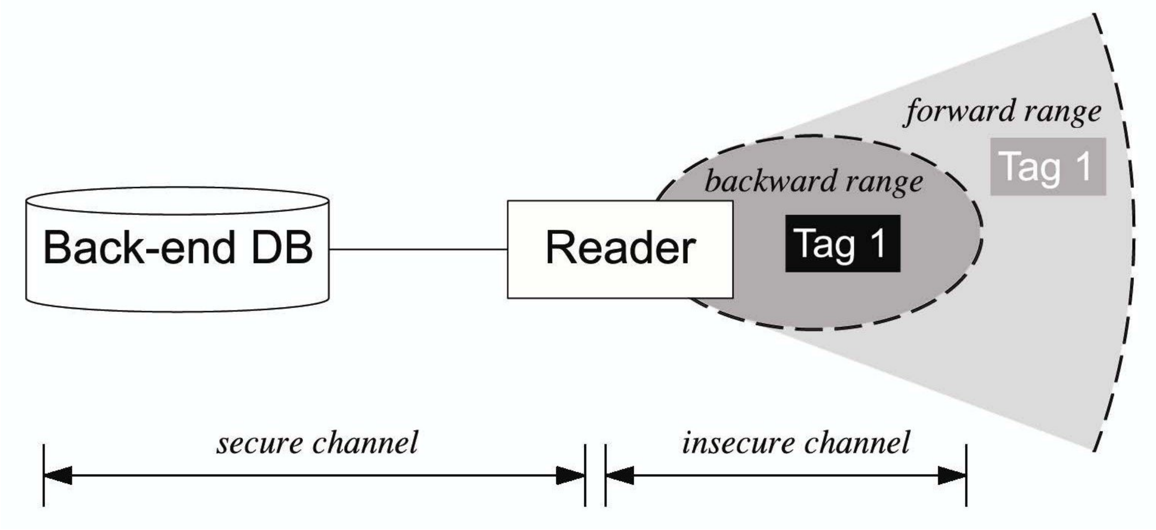


Figure **2-1**: RFID system architecture.

RFID tags are wireless, small sized devices that are placed on the objects that need to be identified. They contain information like manufacturer, brand, model and a unique serial number. Collectively, this information is called the tag’s ID or EPC code as discussed in previous chapter. A 96 bit ID would suffice for most RFID applications. The tag consists of an IC chip and an antenna. It transmits information to a reader in response to a radio frequency signal. The RFID tags can be classified into two major categories namely active and passive tags.

Passive tags have no on-board power source (batteries). They derive their transmission power from the signal of an interrogating reader. They can operate in a number of frequency bands, thus have different read ranges as described in Table 2-1 below.

Table **2-1**: RFID tag frequency and read ranges. Source [5].

RFID Tag	Frequency range	Read range
Low frequency (LF)	124kHz – 135 kHz	Half a meter
High frequency (HF)	13.56 Mhz	Order of tens of centimeters
Ultra High Frequency (UHF)	860 MHz – 960 Mhz	Tens of meters

The tags that contain batteries are of two types. The semi-passive tags have batteries that power the circuitry when they are interrogated. The active tags have batteries that power their transmissions and hence they can initiate communication, have read ranges of 100m or more. Naturally, they are expensive costing more than $20. EPC Global divides tags into six categories, Class 0-1, Class 2, Class 3, Class 4, Class 5 as a defacto standard; Class 0 and Class 1 read only passive identity tags; Class 2 are passive tags with additional functionality like memory or encryption; Class 3 are semi-passive tags that support broadband; Class 4 are the active tags and that are capable of broadband peer-to-peer communication with other active tags in the same frequency band and with readers; Class 5 are the active tags that can support and power Class 0-3 tags and communicate with Class 4 tags as well as with each other wirelessly.

An RFID reader is a device that transmits a radio frequency query signal to T, receives the information sent as response from T and forwards it to B.

The Backend database is a secure server that has a database consisting of information about the objects the tags are attached to. It manages information such as tag ID, location, read time, temperature of sensor attached to tag etc. B resolves ID of tag from the information sent by R and authenticates R as well as T.

Tag Anti-Collision

When an RFID reader attempts to read a population of tags, it is capable of communicating with a single tag at a time. The process of reading a population of tags one at a time is called singulation. The reader transmits a Query message to start the singulation process. When more than one tag respond to the query, a collision occurs and the reader cannot read any of these tags successfully any more. In such cases, the reader and tags must engage in a protocol that enables the reader to read each of the conflicting tags one at a time. These protocols are called anti-collision protocols. Similar collision problems arise in wireless networks as well as Ethernet. RFID systems have certain unique traits that make protocol design challenging. The computation power of reader and tag is comparatively low, tags cannot detect collisions which add a significant burden on the reader to detect collisions and collisions are unavoidable due to varying radio signal strengths. Taking all these factors into account, a number of anti-collision protocols have been proposed in the past. They can be categorized into either probabilistic or deterministic protocols.

Binary tree walking scheme is a deterministic protocol that is implemented in tags that operate at a frequency of 915 MHz namely EPC Class 0 tags. These are the most common type used widely. The tree walking singulation algorithm enables an RFID reader to identify the unique identifiers of tags in the population by a bit-by-bit query process using depth first search of a binary tree as shown in figure 2-2. Suppose the length of EPC of the tags is k bits. All these identifiers are the leaves of the binary tree of depth k. A node at depth d is labeled with a binary string say x of length d. Id d<k, then the node has 2 children at depth d+1; left child is labeled with the string x followed by a 0 and the right child with the string x followed by a 1. Thus, each of the 2^k leaves in the tree has a unique associated k-bit string. These represent the EPCs of the tags. The tree walking algorithm is a recursive depth first search of the constructed tree performed by the reader initiating at the root. At a given node $B=b_1b_2b_3\ldots b_d$, the reader queries all tags having EPCs in the leaves of the corresponding subtree, i.e. the tags with prefix B. All other tags remain silent. The queried tags reply to the reader with d+1 bit (which is a 0 or 1 depending on whether node is a left or right child) of their EPC. If the node has both left and right children, collision occurs and the reader recourses on the left subtree first and then proceeds to the right subtree. If there is no collision and all tags reply either with 0 or 1, then reader queries tags in the corresponding subtree ignoring the other half.

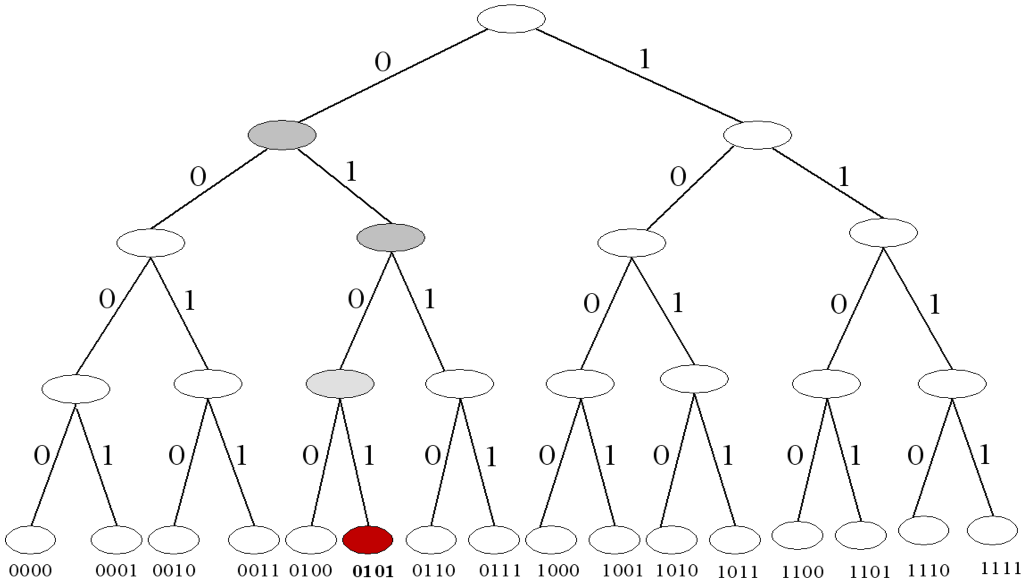


Figure **2-2**: Binary tree walking anti-collision protocol.

The EPCglobal UHF class 1 gen1 tags contain the query tree walking protocol for singulation. The reader queries the tags by using group of bits. Tags whose identifier has matching prefix respond with 8 bits as shown in figure 2-3. This protocol leads to more efficient search than the previous scheme. Tags operating at 13.56 MHz namely EPCglobal class 1 gen 2 tags use the Aloha scheme (Q protocol) for singulation. The Aloha scheme is a probabilistic protocol that is popularly used in Ethernet. In RFID context, tag-tag collision is handled by tags waiting for random intervals before responding again. Thus higher densities of tag populations would result in larger number of collisions and longer waiting intervals for tags. EPCglobal Gen2 specifies Q protocol for singulation where Q is the value that is used by the query to estimate the density of the population. Tags use a PRNG to determine their waiting time.

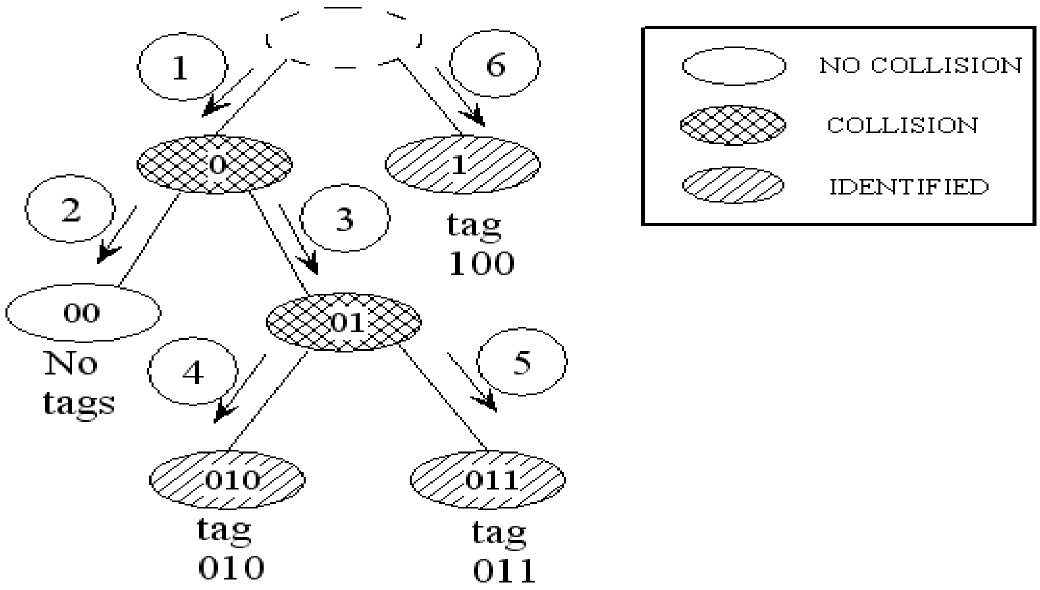


Figure **2-3**: Query tree anti-collision protocol.

Chapter 3

The Threat Model

Security Considerations

Confidentiality: It should not be possible for malicious readers to gather the data stored in a tag and use it to trace the relationship between the tag and the tag bearer. The private information of a tag must be kept secure to guarantee user privacy. The tag information must be meaningless when it is overheard by an unauthorized reader. Thus, confidentiality of the data stored in the tag is of atmost importance.

Anonymity: Although a tag's data is encrypted, the tag's unique identification information is exposed during tag-reader communication as the encrypted data is constant for each tag. An attacker can identify each T with its constant encrypted data. Therefore, it is important to make the tag's information anonymous.

Integrity: Integrity in terms of RFID environment as a security requirement is usually for data integrity between tags, readers, and back-end servers. The air-interface of the communication channel is not fault-tolerable and data synchronization between entities could fail. Thus, integrity among entities must be guaranteed and data recovery mechanisms should be provided in case data loss occurs. In addition to that, if a tag's memory is rewritable, forgery is possible, so integrity for the tag's information must be guaranteed.

Privacy Considerations

In order for RFID technology to become ubiquitous, there are certain security and privacy issues that need to be overcome. The most important of them that deals with the consumers directly is user privacy. Consumer products labeled with insecure tags may reveal sensitive information when queried by readers. For example, from the replies from the tags on these products, information such as amount of money a consumer has, the type of medicine being used by a consumer, the different types of books in the library etc can be obtained that invade user privacy.

A closely related privacy issue is location privacy. The EPC codes of tags that serve as handles to track tags when they communicate with readers can be used to track the tags. Tracking of location leads to tracking of individuals who carry the tag thereby invading their privacy. Dealing with this issue is difficult because even if the tag replies as well as the tag's content are secure, the responses if unique to each tag may help identify the location of the holder. The fact that tracking a tag would give away the location of the product or consumer owning the tag is a major privacy issue that has been discussed in various papers on RFID privacy. Schemes like hash-lock suffer from location tracking because the metaID that the tag sends to the reader during each session is constant. These schemes are explained in detail in chapter 4. To ensure untraceability, every time the tag is queried, the metaID transmitted by the tag should change.

Threats and attacks

When we speak of threat models, STRIDE is a one of the most prominent and useful models that categorizes threats into categories. It is derived from an acronym for the following six threat categories:

Spoofing identity: An example of identity spoofing is illegally accessing and then using another user's authentication information. In RFID context, a malicious reader could spoof a valid reader or tag.

Tampering with data: Data tampering involves the malicious modification of data. Examples in RFID include an attacker obtaining messages between T and R , modifying them and sending them.

Repudiation: Repudiation threats are associated with users who deny performing an action without other parties having any way to prove otherwise. This could happen if readers or tags are taken control of. Nonrepudiation refers to the ability of a system to counter repudiation threats. In this thesis, we assume that valid readers cannot be taken control of by an attacker. So this threat is not considered in the threat model.

Information disclosure: Information disclosure threats involve the exposure of information to individuals who are not supposed to have access to it. This is one of the important threats in RFID as an attacker should not obtain EPC of the tags.

Denial of service: Denial of service (DoS) attacks deny service to valid entities. This can be achieved in an RFID system when an attacker disrupts messages sent between R and T. You must protect against certain types of DoS threats simply to improve system availability and reliability.

Elevation of privilege: In this type of threat, an unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system. Elevation of privilege threats are not considered in this thesis for the same reason as repudiation threats.

Table 3-1 discusses the various mitigation techniques available in the literature to counter these threats. In an RFID system, there are certain vulnerabilities that make the system prone to many of the threats described by the model. One of the unique traits of RFID that leads to a number of attacks is the asymmetry in the communication channel between the reader and tag.

Table **3-1**: STRIDE categories and mitigation techniques.

Categories	Techniques
Spoofing identity	Authentication Protect the secrets Safe storage of secrets
Tampering with data	Authentications Hashes Message authentication codes Digital signatures Tamper-resistant protocols
Repudiation	Digital signatures Timestamps Audit
Information disclosure	Authorization Encryption Protect the secrets Secure storage of secrets Privacy-enabled protocols
Denial of service	Authentication Authorization Filtering Throttling Quality of service
Evaluation of privilege	Run with least privilege

Passive tags receive power via the forward channel, thus it is much stronger than the backward channel. As a result, the forward channel may be monitored from a much greater distance than the backward channel. For example, a 915 MHz passive tag may have a 3-meter operating range, yet its forward channel may be monitored from 100 meters. This asymmetry in channel strength could lead to eavesdropping. The forward channel eavesdropping is a more serious threat than the backward channel because for monitoring the backward channel, an eavesdropper would have to be within the short range and thus can be detected. The threat of backward-channel eavesdropping should not be discounted completely though. An attacker could still eavesdrop by attaching a malicious device on to a legitimate reader. However, these attacks are more costly and easier to detect than forward-channel eavesdropping.

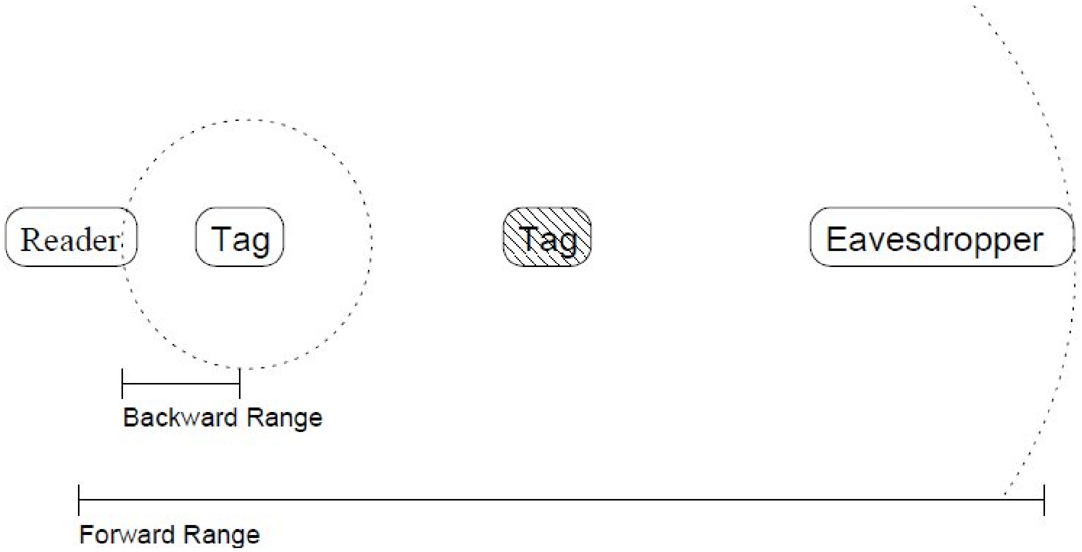


Figure **3-1**: Forward vs. Backward Channels: The reader will detect the nearby tag, but cannot detect the shaded tag. A distant eavesdropper may monitor the forward channel, but not the tag responses. Source [9].

An important attack on the security and privacy of RFID tags is T g cloning. As an example, an attacker spoofing the communication between a tag and reader can replicate the tag and replace it thereby continuously authenticating to the reader. Thus an authentication protocol needs to be designed using which the tag can prove its identity to the reader and backend database. Not just for tags but authentication of a reader is imperative as well because otherwise, an unauthorized reader deployed in an RFID environment could establish communication with valid tags and gain information about them. Thus a mutual authentication protocol needs to be deployed for secure communication between reader and tags.

The security weakness of RFID technology comes from the fact that a RFID tag wirelessly transmits its EPC identifying the object associated with the tag upon receiving the “query request” message from a reader. Using the unique EPC as reference, an attacker equipped with a compatible reader can track the moving history, the personal preferences and the belongings of a tag's holder. Absence of authentication causes the revealing of EPC to malicious readers (referred to as skimming attack). Once capturing EPC, an attacker can duplicate genuine tags and use the cloned tags for a variety of malicious purposes. An attacker could steal an EPC by eavesdropping communication channel between a tag and a reader if the EPC is not encrypted.

RFID tags are inexpensive devices that offer no tamper resistance, thus they suffer from physical attacks that can expose their memory content. They could be discarded, or easily captured, and may be highly vulnerable to side-channel attacks on the stored keys. An attacker getting hold of the tag may be able to link the tag to past communications between the tag and reader thereby giving away its identity. Thus, the protocol designed should preserve forward security; messages transmitted presently should be secure in the future, even if the tag is compromised. Forward security is essential to guarantee the privacy of past transactions if the long-term key or current session key is compromised.

Indistinguishability is a core requirement in RFID communication. If an eavesdropper listening to the communication between tag and reader or reader and database is able to decipher information about tag, then the security of the whole system is compromised. The attacker can use the gained information to clone the tag or compromise the database entry.

Replay attacks are the most common attacks on authentication protocols and they hold a great significance in an RFID environment as any attacker could impersonate a reader or valid tag by replaying messages that they overheard in previous communications if possible.

A DoS (Denial of Service) attack could be carried out by an attacker if he intercepts the messages sent between the tag and reader, thereby disrupting communication. DoS attacks are prevalent not just in RFID but other technologies like wireless networks too. A DoS attack could be carried out to desynchronize the tag and the backend server in case of protocols where these entities update shared keys based on messages they send to each other. This results in the tag being no more recognized by the database, hence invalidating the tag.

RFID Bill of Rights

Conditions, operational policy and rules for RFID privacy have been presented by Granfinkel as well as Auto-ID center for RFID deployment, these are restated as follows: Users of RFID systems and purchases of products containing RFID tags have: 1) The right to know if a product contains an RFID tag. 2) The right to have embedded RFID tags removed, deactivated, or destroyed when a product is purchased. 3) The right to first class RFID alternatives: consumers should not lose other rights (e.g., the right to return a product or to travel on a particular road) if they decide to opt-out of RFID or exercise an RFID tag's "kill" feature. 4) The right to know what information is stored inside their RFID tags. If this information is incorrect, there must be a means to correct or amend it 5) The right to know when, where, and why an RFID tag is being read.

EPCglobal Generation 2 standard specification

EPCglobal is a joint venture between EAN International (Europe) and UCC (USA) aiming at developing industry RFID standards as discussed in chapter 1. One of the most important standards proposed by EPCglobal is the EPCglobal Class-1 Gen-2 RFID specification which defines the functionality and operation of a RFID tag. According to the specification, the EPC passive tag acquires its operating energy from the RF signal generated by RFID reader. The wireless communication range of EPC passive tags is about 2 to 10 meters and EPC RFID systems operate within the frequency range of UHF band i.e. 800 – 960 MHz.

EPCglobal Generation 2 standard supports two security mechanisms namely the Kill command and the access command. The KILL password and the KILL command disable RFID gen2 tags permanently. These are further discussed in the next section. ACCESS password, ACCESS and LOCK commands provide secure access to the tag's memory. The EPCglobal gen2 tag consists of an XOR gate to perform CRC and ability to generate a 16 bit random number. It can temporarily store 2 16 bit random numbers.

The tag memory is logically separated into 4 sections namely

Reserved memory: contains the 32 bit kill and access passwords

EPC memory: contains the 16 bit CRC, 16 bit PC and the EPC

TID memory: has space for custom commands and optional features supported by the tag

User memory: reserved for user-specific data.

An example of EPCglobal Class 1 tag specifications are as described below in Table below:

Table **3-2**: EPCglobal class1 tag specifications. Source [5].

Class1 EPC tag	Passively powered with 96 bit memory
Range	3m operating, 100m forward channel, 3m backward channel
Anti-collision algorithm	Probabilistic
Performance	100 read operations per second
Clock cycles per read	10,000
Security gate count	200-2000
Physical operations	Imprint, kill
Logical Operations	Read, ping

Current security and privacy methods

EPC tags carry no explicit mechanism for authentication. A basic tag carries only the mandatory features of the EPCglobal standard. They have only one security feature, the Kill command. When a tag receives the kill command, it self-destructs. To protect against malicious readers sending kill commands and deactivating tags, the command is effective only when sent with a valid pin. The Kill pin is 32 bits in length according to the EPCglobal standards. Enhanced EPC tags respond to a command called Access. The implementation of access command is optional in EPCglobal standard. Using a 32 bit access pin, the command effectively helps tag transition to the secured state. Read access to tag's memory is possible only when the tag is in this state.

Kill tag approach: This is the most straightforward approach for protection of consumer privacy. The RFID tag is "killed" before it is placed in the hands of the consumer. A killed tag cannot be re-activated. This approach can be very useful in supermarkets where RFID tags are used for inventory management and monitoring the stocks. At checkout, the tags could be killed to protect consumer privacy. There are many cases where this approach would not work. For example, the store might want to retain the tag so it could be scanned incase of return of the product.

Faraday Cage approach: A RFID tag may be shielded in order to protect against scanning by malicious readers by using a Faraday cage, a container made of metal mesh that blocks the radio waves. There are companies offering faraday-cage-based products for privacy purposes. This is a physical method to protect consumer privacy.

Active Jamming approach: In this approach, consumer carries a device that actively broadcasts radio signals to block any malicious reader in the vicinity trying to scan the tag. This approach is not very practical if the broadcast power is too high as it could damage RFID systems nearby.

Blocker Tag approach: Blocker tag is a cheap passive RFID device that can simulate many RFID tags simultaneously, thus can be used to block a population of tags when queried by a reader. This approach was proposed by Juels et al for protecting the consumer privacy. The disadvantage of this approach is that a blocker tag is needed for every set of tags that need to be protected.

Current public key protocols cannot address security issues in EPC Global Gen2 RFID tags. Some of the popular public key algorithms in commercial use include RSA, Diffie-Hellman and Elliptic curve cryptography; the security of these systems is based on hardness of factoring but they are not feasible to be used in gen2 tags due to various reasons. Each of these protocols run in quadratic time and is inherently slow as they require multiplication and division of very large numbers. Gen2 tags have very few gates and low memory capability which makes it impossible for these public key protocols to be implemented in them.

According to the EPCglobal Class 1 generation 2 standard specifications, the two main operations in RFID namely inventory and access described in figure 3-2 and are as follows:

Inventory round is carried out as follows:

(1) A reader sends a request message called QUERY to the tag population. The query initiates an inventory round for the Q protocol.

(2) Each tag on receiving the Query picks a 16 bit random value using the PRNG and loads it into a slot counter. When the slot counter hits 0, it backscatters the RN16 to the reader.

(3) Reader on singulating the tag acknowledges the tag with AK containing the same RN16.

(4) Tag compares the received RN16 with the value it stored. If they are equal, it backscatters PC, EPC and CRC16.

After acknowledging the tag, the reader will need to access the tag. Following steps comprise the access phase.

(1) The reader issues ReqRN with the previous RN16 to the tag.

(2) Tag compares the ReqRN with the RN16 stored and if they are the same, it computes a new RN16 called the handle and backscatters the handle to the reader.

(3) The reader then uses the write, kill or access commands by generating a ciphertext string which is a xor of the 16 bit word (data or the corresponding passwords) with the new RN16 received. This is transmitted to the tag.

(4) Tag decrypts the relieved ciphertext by XORing the string with the RN16.

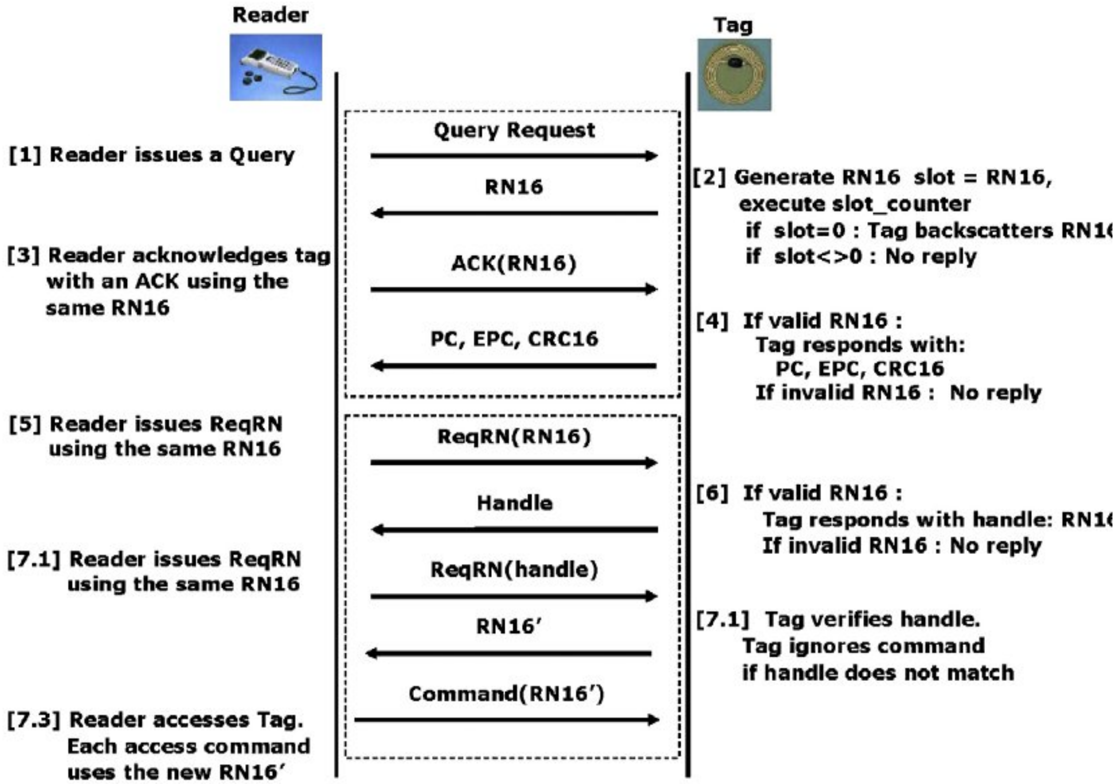


Figure **3-2**: Inventory and access rounds Source [16].

Chapter 4

Literature Review

The communication channel between the tag and reader as well as that between the reader and the backend database are insecure thus leading to various security and privacy concerns as seen in the previous sections. The first step in overcoming these threats is to come up with a mutual authentication protocol using which the tag and reader are aware of each other's identity. Initial attempts for authenticating tag – reader communication involved using the KILL command, active jamming and blocker tag. In this section, an analysis of different authentication protocols that use hash functions, PRNG, secret keys etc is discussed.

Hash Definition

A hash function is an efficiently computable function which maps an arbitrary length input to a fixed length output. The three main properties of a hash function are as follows:

- Preimage resistance – For all outputs y, it is computationally feasible to find any input x such that h(x) = y given that no corresponding input is known.
- 2^{nd} – preimage resistance – Given x, it is computationally infeasible to find x' ≠ x such that h(x) = h(x').
- Collision resistance – It is computationally infeasible to find any pair of inputs x and x' such that h(x) = h(x').

A one way hash function is one that offers preimage and 2nd preimage resistance. A collision resistant hash function has 2nd preimage resistance and collision resistance.

Pseudorandom Generator

Random number generation has its applications in cryptographic operations like session key generation and challenge-response protocols. The function outputs a sequence of binary digits such that at any stage, the next output cannot be predicted from the previous output. However, this property in a random generation function is very difficult to obtain using computers since computers are deterministic devices. To achieve true randomness would require the function's input to be dependent on physical phenomena (like rate of neutron emission from a radioactive substance) which is difficult. A nearly random sequence can be produced using pseudorandom generator (PRNG). A PRNG produces a sequence of bits that has a random looking distribution. It takes a sequence of bits called seed as input. A different seed produces a different sequence of bits. Many of the current authentication protocols proposed for RFID communication employ hash functions and PRNG to protect data and authenticate tags and readers to R.

In the following sections, different authentication schemes are discussed that use hash functions and PRNG for reader-tag communication. Security and privacy analysis of these schemes are carried out in detail.

Hash Lock Scheme

Weis et al. proposed the hash lock and randomized hash-lock protocols for mutual authentication of tag and reader. In the hash-lock scheme, the backend server and each tag share a secret key k. The key is stored in the server while the tag stores the metaID which is just h(k). The tag transmits this constant for every session hence enabling an adversary to track it across different sessions. The scheme is depicted in figure 1-1. The problem with this scheme is that the metaID transmitted by the tag which is the metaID is constant for all sessions of communication. Thus this can be used as an identifier to track the tag across different sessions. Thus location privacy of the tag bearers is compromised. Spoofing attack can be carried out easily on this protocol. A malicious reader can send the Query to the tag to obtain its metaID which it then forwards to the DB to obtain the key and ID of the tag. This enables the attacker to obtain the tag's information.

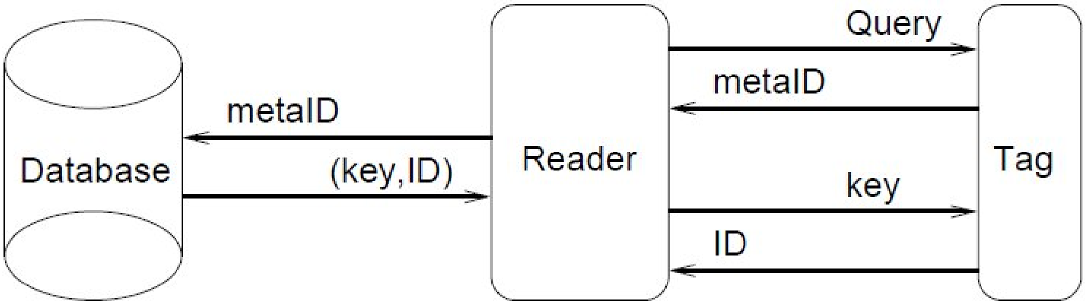


Figure **4-1**: Hash- lock scheme. Source [5].

Randomized Hash Lock Scheme

To overcome the tracing problem, the authors proposed the randomized hash-lock scheme where every tag implements a pseudorandom generator (PRNG). Tag picks a random number 'r' for each session and computes c= h (ID, r) as tag's unique identification for every session. The extended hash-lock scheme is vulnerable to replay attack where an attacker overhearing the tag's reply could forward it to the DB at a later stage to obtain the IDs which it can use to communicate with the tag. Spoofing attack is possible as an adversary can query the tag and obtain (c, r). Using this information, the attacker can impersonate as a valid tag to the reader; the reader’s response will identify the tag. Thus, mutual authentication is required between reader and tag to counter the replay and man-in-the-middle attacks.

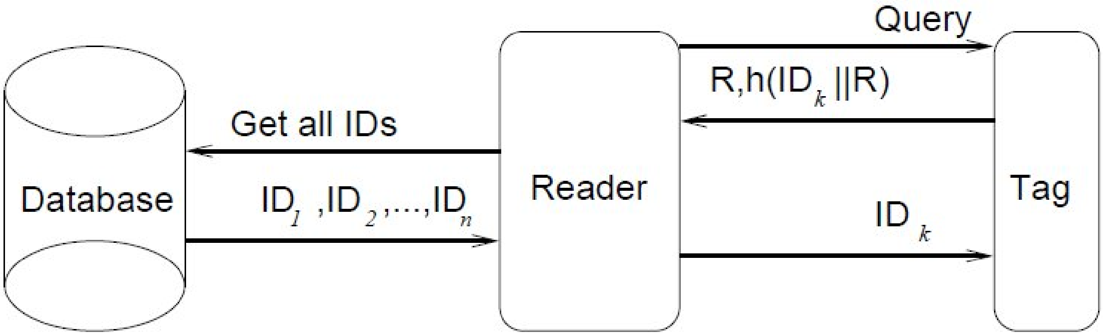


Figure **4-2**: Extended Hash Lock Scheme. Source [5]

Hash based varying identifier scheme

Another hashed approach proposed by Henrici and Muller requires the backend database to provide a random number every session to the tag to make the tag identifier random protecting location privacy. Though this scheme protects against spoofing, it is prone to the man-in-the-middle attack by an unauthorized reader interrogating the tag with a random number. The tag increases its value k every time it receives a request even when the identification fails, but it updates k_{last} only when the identification succeeds. Thus, an adversary may query the tag several times to abnormally increase k and in turn Δk. Because this value is sent in clear in the second message, the adversary is then able to later recognize its target tag using this value; if the tag sends abnormally high Δi, the adversary can conclude that this is the target tag.

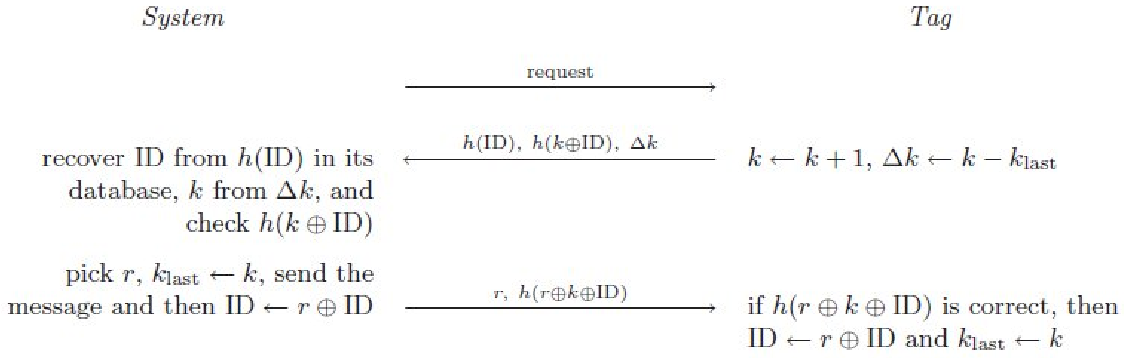

System
Tag
request
$h(\mathrm{ID}),\ h(k \oplus \mathrm{ID}),\ \Delta k$
recover ID from $h(\mathrm{ID})$ in its database, k from Δk, and check $h(k \oplus \mathrm{ID})$
$k \leftarrow k+1,\ \Delta k \leftarrow k - k_{\mathrm{last}}$
pick r, $k_{\mathrm{last}} \leftarrow k$, send the message and then $\mathrm{ID} \leftarrow r \oplus \mathrm{ID}$
$r,\ h(r \oplus k \oplus \mathrm{ID})$
if $h(r \oplus k \oplus \mathrm{ID})$ is correct, then $\mathrm{ID} \leftarrow r \oplus \mathrm{ID}$ and $k_{\mathrm{last}} \leftarrow k$

Figure **4-3**: Hash based varying identifier scheme. Source [8].

Improved hash based varying identifier Scheme

Hwang et al. proposed an improved hash-based verifying identifier scheme where the reader has a pseudorandom number generator (PRNG) to counter the man-in-the-middle attack that was described previously. The reader sends a pseudorandom number S with every query. Then the tag replies with h(ID) (for finding the corresponding record for the tag in the DB) and half of a new identifier, half_L(R) (R = h(ID||S). Then, the reader forwards h(ID), half_L(R) and S. In the authentication phase at DB, h(ID) is used to find the corresponding record of the tag and ID of the tag is obtained. With stored ID and S received from the reader, the DB can calculate R'= ID||S and the tag can be authenticated comparing half_L(R') with half_L(R) received from the tag. If the authentication is successful, then the ID of the record is updated to a new ID = R' and h(ID) to h(R'). Then, the DB replies half_R(R') with tag data to the tag through the reader. With half_R(R), the tag can check whether the reply message is valid or not. If the process is successful, the tag and the database update their ID to ID $\oplus$ (R|| R'). In this way, the session identifier for the next session is updated by both the tag and the server and the states are synchronized.

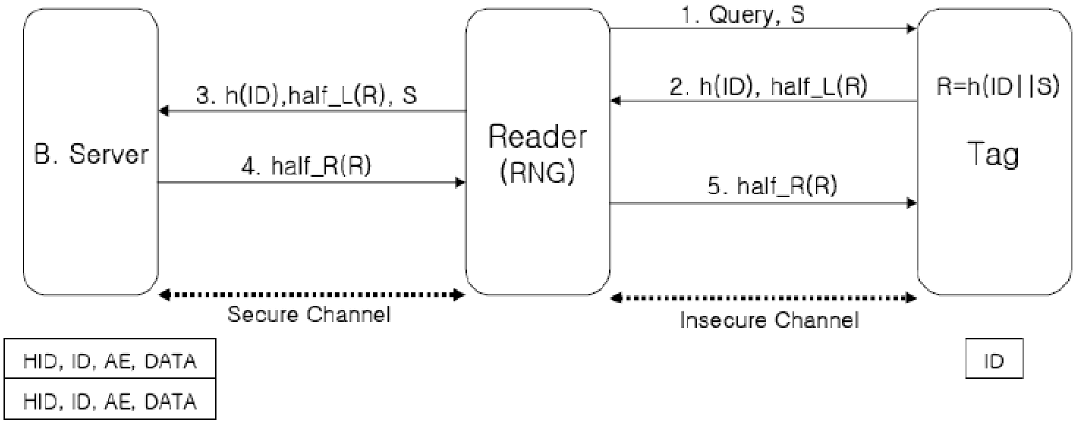


Figure **4-4**: Improved hash based varying identifier scheme. Source [5]

The scheme assumes that the reader is a trusted party and hence they shift the PRNG from backend database to the reader. Scheme protects location privacy as a tag's unique identifier is changed every session. Tag and backend server are mutually authenticated and hence replay attacks are not possible. If the reader is not a trusted third party, then man-in-the-middle attack is possible. An attacker with a malicious reader can query the tag with any pseudorandom number and hence can get hold of the messages communicated thereby successfully learning the tag's content.

Hash chain based approach

Ohkubo et al. proposed a cryptographic approach based on hash chain to ensure forward security in low cost RFID system. The hash chain technique renews the secret information contained in the tag using two one-way hash functions H and G as shown in figure 4-5. Initially, tag has information s_1. In the i^{th} transaction with the reader, the RFID tag 1) sends answer $a_i = G(s_i)$ to the reader. 2) renews the secret $s_{i+1} = H(s_i)$ from the previous secret s_i. The reader sends a_i to the database which maintains the pair (ID, s_1) where s_1 is the initial secret information and is different for each tag. On receiving a_i, database calculates $a_i' = G(H^i(s_1))$ for each s_1 in the list, and checks if $a_i' = a_i$. If they are equal, it returns ID back. The protocol ensures anonymity as the tag output changes every session as well as resistance to the denial of service attack. But it does not prevent the replay attack. Also, malicious readers cannot be distinguished from actual readers.

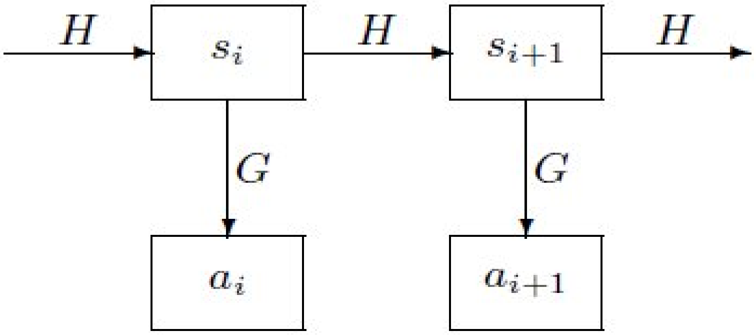


Figure **4-5**: Hash chain approach. Source [10]

A lightweight RFID protocol

T. Dimitriou proposed a simplified protocol for tag-reader communication that aims at protecting against replay and desynchronization attacks. It uses a shared secret between tag and reader. The reader is assumed to be trusted and hence they don't distinguish between reader and database. The secret can be common for all tags but in that case, compromise of a single tag would reveal the identity of the entire population of tags. The alternative is to have different secrets per tag and a mechanism for reader to map secret key to the tag. The database uses the hash of ID of tag as a key to lookup the entry for the tag. The ID of tag changes every session. The reader sends query message along with a nonce N_R. The tag computes the hash of the current ID and sends it along with a new nonce N_T as well as a keyed hash of the nonces.

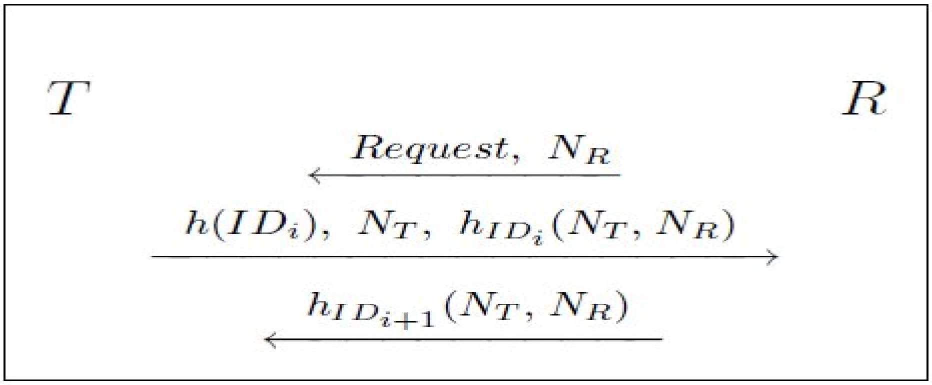


Figure **4-6**: A lightweight RFID protocol. Source [9]

The nonce N_R is to prevent an attacker from performing replay attack. Attacks by an attacker impersonating as a valid tag or reader cannot be carried out as secret is not known to the attacker. Thus desynchronization attack is prevented. The protocol demonstrates that it is essential that the tag authenticate the reader as well. A mutual authentication is essential to avoid desynchronization attacks on the RFID system. The problem with this protocol is that maintaining number of IDs per tag is not practical.

Reader aided ID refreshment scheme

Yang et al. proposed a protocol where reader refreshes the ID of the tag. Reader- aided ID refreshment was introduced to counter the replay attacks as well as tracking of tags. Figure illustrates the communication between reader and tag. At the beginning, the tag and the backend database share two keys k_1 and k_2 as well as the tag identifier C. The keys are refreshed for every session. The steps in the protocol are depicted in Figure 4-7. A possible attack against this protocol is as follows: An adversary who overhears the communication can obtain the current values of S, ID and ID', namely S_{cur}, ID_{cur} and ID'_{cur}. The adversary then can send $S_{next} = S_{cur} \oplus$

ID'_{cur} to the tag. The tag would compute $ID_{new} = (k_1 \oplus ID'_{cur}) \oplus (S_{cur} \oplus ID'_{cur}) \oplus C$ and send it to the reader. ID_{new} is same as ID_{cur}, thus using this adversary can track the tag in the subsequent sessions.

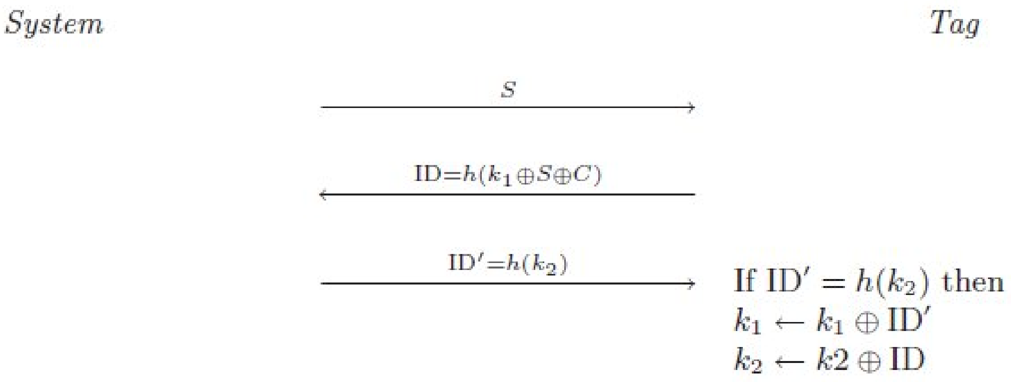

System
Tag
S
$\mathrm{ID}=h(k_1 \oplus S \oplus C)$
$\mathrm{ID}'=h(k_2)$
If $\mathrm{ID}' = h(k_2)$ then
$k_1 \leftarrow k_1 \oplus \mathrm{ID}'$
$k_2 \leftarrow k2 \oplus \mathrm{ID}$

Figure **4-7**: reader aided id refreshment scheme. Source [8].

An and Oh protocol

The authors An and Oh discuss an authentication protocol that uses hash functions and PRNG for communication in an RFID system where readers are linked to different databases and each reader is aware of the ID of the DB it is linked to. Hash functions provide anonymity and PRNG is used to resist the replay attacks. The DBs contain the tag identifier SN as well as their IDs. The protocol steps include the reader sending a query to the tag with the ID of the DB. The tag computes a random number ‘r’ and computes a hash of its identifier (SN), the DB’s ID as well as r and sends it to the reader. Reader forwards the data to the DB which then computes hash of the SN, DB ID and the r sent in the message for each entry to check if one of them is equal to the one in the message sent. If it finds such an entry, the tag has been authenticated and hence the tag data (like price, period etc) is sent back to the reader. An illustration of the protocol is shown in the figure 4-7 below.

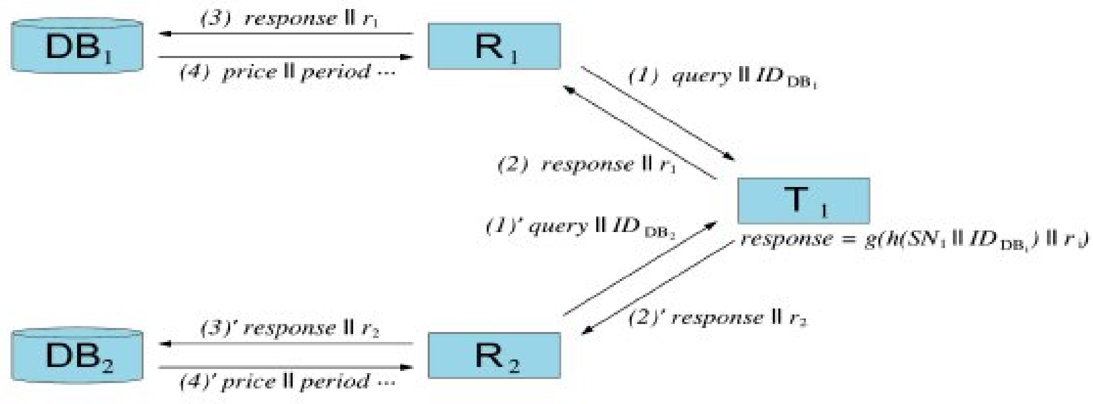


Figure **4-8**: An and Oh protocol. Source [14]

In the above protocol, the reader is not authenticated to the tag. A malicious reader could query the tag by obtaining ID of DB from eavesdropping and obtain tag's reply which can then be forwarded to the DB to obtain the tag's information.

Rhee et al protocol

The protocol proposed by Rhee et al. authenticates the tag and reader using one-way hash functions and PRNG. The reader broadcasts the Query message to the tags along with a random number R_{reader}. The tag concatenates its random number R_{tag} to Rreader along with the ID of the tag and hashes it. The tag then replies the hash along with R_{tag} back to reader. The reader forwards the hash along with the 2 random numbers to the database. The database hashes its stored ID with the two random numbers to authenticate the tag. The protocol design steps are shown in the figure 4-9 below.

Since the tag response changes every session because of the random number, tracking of the tags is not possible. Thus, replay attacks are not possible. Just like the previous scheme, the reader is not authenticated to the tag in this protocol as well.

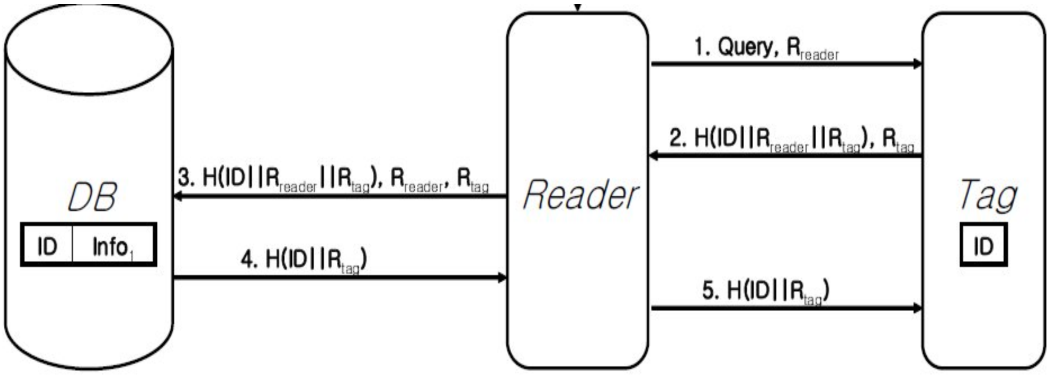

DB
ID
Info1
Reader
Tag
ID
1. Query, R_reader
2. H(ID||R_reader||R_tag), R_tag
3. H(ID||R_reader||R_tag), R_reader, R_tag
4. H(ID||R_tag)
5. H(ID||R_tag)

Figure **4-9**: Rhee et al protocol. Source [15]

A malicious reader could query the tag using its own random number and then forward the tag's reply to the DB to obtain the data about the tag. Also, the protocol does not provide forward security; an attacker eavesdropping the communication and recording the messages can recover information about the tag in future.

Comparison of related Work

In this section, the various protocols proposed in the literature are compared against the different attacks and threats that the current RFID system is susceptible to. In the below table 4-1, the schemes are compared under the most important security and privacy requirements namely data protection, tracking prevention and forward security. Tracking prevention is a very important requirement to protect user privacy. Data protection is necessary in any system where security is of prime importance. In an RFID system it is required to prevent an attacker from gaining information about the tag and the EPCs. Forward security ensures that compromise of the current session identifier does not reveal information exchanged in the past. It is necessary in an RFID system because the tag-reader communication channel is highly prone to eavesdropping.

Table **4-1**: Comparison of related work

Scheme	Data protection	Forward security	Tracking prevention	Reply attacks Resistance	Required Computation
Hash lock	√	X	X	X	Hash
Extended hash lock	√	X	√	X	Hash, PRNG
Hash based varying identifier	√	√	Δ	√	Hash, PRNG, XOR
Improved hash based varying identifier	√	√	Δ	X	Hash, PRNG,XOR
Lightweight protocol	√	√	√	√	Hash
Reader aided refreshment	√	X	X	X	Hash
Ohkubo et al. [10]	√	√	√	X	Hash
Rhee et al	√	X	√	√	Hash, PRNG
An and Oh	√	X	X	X	Hash, PRNG

Notation: √ - satisfied, x – not satisfied, Δ - partially satisfied

Chapter 5

Proposed protocol for mutual authentication in RFID System

Low cost RFID tags have very limited computational power. Thus, implementing computationally intensive cryptographic algorithms to enforce security and privacy will drain the power in these tags very quickly. There is a tradeoff between the efficiency of algorithms and the computational power consumption. A lot of research has gone into developing lightweight cryptographic protocols to protect against security and privacy threats in RFID. Solutions make use of a hash function [6, 7] for providing protection against replay attacks but the current EPCglobal Class-1 Gen-2 RFID specification does not approve cryptographic hash function like MD5 and SHA-1. Thus, solutions with the available functionalities of current RFID standards have to be sought after. Juels [3] suggested such a scheme to prevent cloned tags from impersonating legitimate tags. However, his protocol did not take eavesdropping and privacy issues into consideration. The proposed scheme [3] employs only PRNG and pre-shared secrets between tag and reader (e.g., PIN, seed to PRNG), called synchronization-based as it requires session-key synchronization between tag and reader. The proposed protocol also requires the implementation of a one-way secure hash function, a keyed hash function and XOR gates. According to the EPC Global Generation2 standard specifications, a low cost tag can implement only these basic functions.

The purpose of the one-way hash function is to anonymize the data being sent between the tag and reader. Thus an eavesdropper will not be able to understand the communication between tag and reader. Also, due to its properties as discussed earlier, it is difficult to invert and hence does not leak information about the message it is applied to. Pseudo random generator supplies the session key to be used for communication between tag and reader as well as the reader and database. The session key is refreshed every session to mitigate the location tracking of tags. The XOR gates are used for XORing the EPC of the tag with the session identifier as well as a random nonce before being sent. The random nonces used are to provide freshness of the message such that an eavesdropper cannot replay the messages in future.

A pseudo random generator (PRNG) is used to generate a new session key which is shared between RFID tag and reader for each and every session. In the EPCglobal Class1 Gen-2 RFID specification, the RFID tag is capable of generating 16-bit pseudo-random number [3]. The algorithm uses the same PRNG with the same seed at both RFID tag and reader. The PRNGs on both sides start with a common seed so that the tag can be authenticated to the database based on the session key generated by the PRNG every session. During tag fabrication, the manufacturer assigns EPC and other parameters unique to each tag. Then, it chooses a random seed number 'seed' and stores K_1 = f (seed) to tag's memory and backend server's database entry corresponding to the matching EPC. A random PIN (access PIN defined in Gen-2 specification) is also stored in both tag's memory and backend server database in a similar way.

Assumptions

The proposed protocol assumes that both the reader-tag and the reader-database channels are insecure as is the case in real world. Many of the papers in chapter 4 assumed that the reader is trusted. But there have been incidents recently where people equipped with a compatible reader could successively scan tags. Thus, for the protocol design, we assume that the reader cannot be trusted.

The database resides on a secure server and the information in the server cannot be tampered with. Security of information in the database is out of scope and is not discussed in this thesis. The server is also capable of implementing PRNG as well as cryptographic operations like encryption and decryption. The reader has the necessary power to perform basic encryption and decryption. The tag on the other hand can support only PRNG and one way hash functions. All the devices have XOR gates.

The secure server stores information about the tag along with the tuple (EPC, K_i). This tuple uniquely identifies the tag and authenticates it to the server. To determine the identifier of the tag from which the server receives message, it has to do an exhaustive search of the database, checking if the message is equal to hash of EPC and K_i for each entry. This does not scale well when tag populations are large as the search would take a lot of time. For this work, the database is implemented as a hash table and thus requires O(1) time for search.

Protocol design

To guarantee security and protect the privacy of the tag bearers, following requirements are considered during the design of the authentication protocol. For a query from the reader, the tag emits an anonymous ID rather than its EPC. The reader forwards the message to the database which then in turn sends data about the tag back to reader and an update message that the reader forwards to the tag. This message indicates a successful read attempt to the tag. For reader to initiate Q protocol, it first contacts the DB to obtain a handle that it uses to query the tags. This step is necessary to encounter man-in-the-middle attacks as discussed in detail in the next chapter.

During singulation of the tag, the reader uses the Q protocol. To make sure that a valid reader is querying the tags, the reader first contacts the database with a start message. The DB validates the reader and then issues a handle r. The reader sends the Query request that contains the value of Q (initialized to 4) along with r. Once the tags receive the query command, they initialize their slot counters with a random value. If the slot counter of any tag hits 0, the tag replies. The DB validates the reader and then issues a handle r.

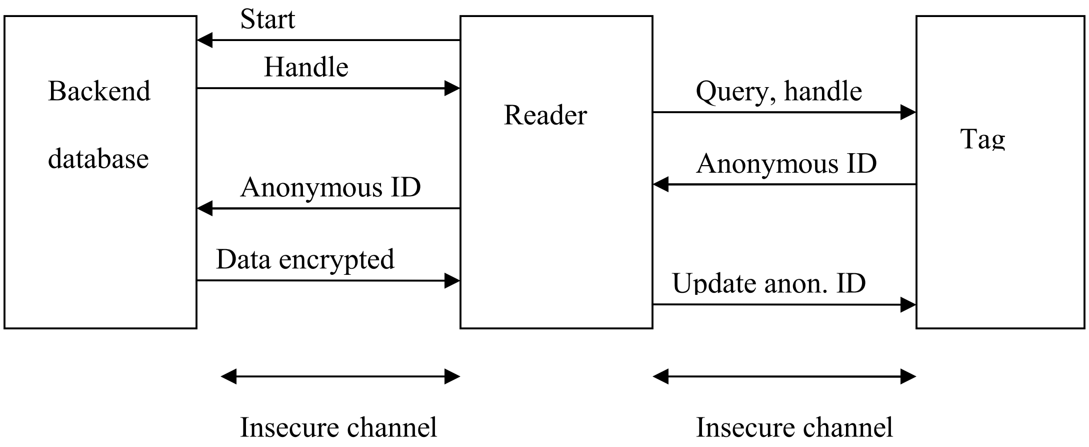


Figure **5-1**: Overall architecture of the proposed RFID system.

The reader sends the Query request that contains the value of Q (initialized to 4) along with r. Once the tags receive the query command, they initialize their slot counters with a random value. If the slot counter of any tag hits 0, the tag replies. While replying, in the existing Q protocol, the tag sends a handle which as we saw in chapter 3 can be used for launching replay attacks. In the proposed protocol, the tag uses $r\oplus K_i$ as the seed to its PRNG to obtain f(r) which it XORs with the EPC and the session identifier K_i (calculated using PRNG with shared seed) and hashes the entire value to obtain a value say M ($M= H(EPC \oplus K_i) \oplus f(r\oplus K_i)$). The tag sends M to the reader. The reader on receiving the message forwards M to the database after encrypting it with the key it shares with the database, k. The database on receiving the messages decrypts the message to obtain M. Then it calculates $f(r\oplus K_i)$ for the r it issued to the reader and XORs M with the calculated $f(r\oplus K_i)$ to obtain $H(EPC \oplus K_i)$. It uses this hash as a key to retrieve tag information from database. If it is not a valid key, the DB ignores the message else the tag has been authenticated. The server then sends the data about the tag encrypted with the shared secret k to the reader. Along with that, it sends $H(K_{i+1})$ to the reader which it forwards to the tag. Here K_{i+1} is the second random number in the sequence obtained using K_i-1 as the seed, the first random number being K_i (Note: K_i-1 is the seed used by the database and the tag in the PRNG to generate the session key K_i). The purpose of the challenge is for the tag to know that the communication was successful and the session identifier can be updated. It acknowledges the receipt by sending $H(K_{i+2})$ (K_{i+2} is the third random number in the sequence generated by using K_i-1 as the seed) back to the reader which is forwarded to the database. The DB in turn can generate the third random number and check that the hashes are equal. Using this challenge response, the database and tag update their session identifiers synchronously to the same value. The mutual authentication steps in the protocol are shown in figure 5-2.

Detailed Description

Step 1: The reader requests a handle from DB to start singulation of a population of tags.

Step 2: The DB authenticates the reader and then sends the handle r (a random number). It stores the r corresponding to the reader in the DB.

Step 3: The reader sends the Query message along with the r to initiate the Q protocol for singulating a tag. During singulation, all tags reply by hashing their EPC with their session identifiers and then XORing with $f(r\oplus K_i)$. This is to secure the tag identifiers from being read by an eavesdropper. The metaID of the tag is thus $H(EPC \oplus K_i)$.

Step 4: For a query from a reader, the tag replies by hashing EPC with the session identifier K_i and then appending $f(r\oplus K_i)$ to the result. The handle r received from the reader prevents man-in-the-middle attacks. If the message is replayed in a different session, the server would ignore the message as it would not match the entry for the tag in the database owing to the fact that the session identifier would have changed. Thus, the session key not only authenticates the tag to the database but also provides means for detecting replay attacks.

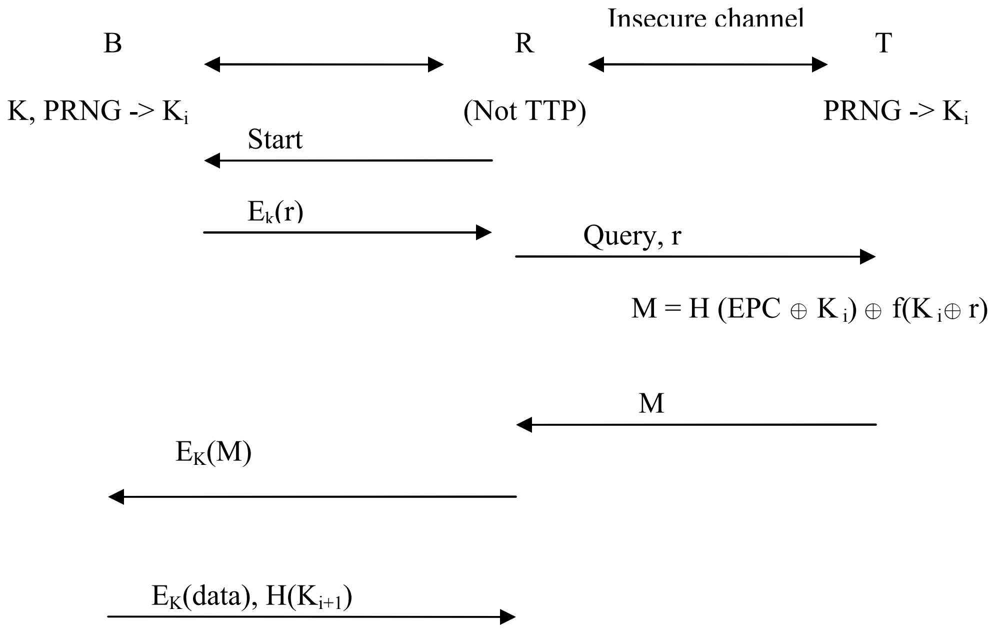


Figure **5-2**: Basic authentication in proposed protocol.

Step 5: The reader needs to forward the message it received from the tag to the database. The secret key shared between reader and the database authenticates the reader to the database.

Step 6: The database decrypts the result with the key K to obtain the message that was forwarded by the tag to the reader. It checks if the hash is equal to the calculated hash of EPC and K_i stored for each tag and if $f(r \oplus K_i)$ is valid for the corresponding r assigned for that reader. If it finds such an entry, the database has identified the tag and hence can obtain the data about the tag. The data is encrypted with K and sent to the reader. Along with it, a hash of K_{i+1} is sent to the reader.

Step 7: The reader uncovers the data for the tag from the message that was sent by the database. It then forwards the $H(K_{i+1})$ to the tag. This message is to indicate to the tag that the read attempt by the reader was successful. It indirectly authenticates the reader to the tag.

Step 8: The tag on receiving the update message modifies its session identifier to a new value (obtained by using the current session identifier as seed in PRNG) and then acknowledges the received message by sending $H(K_{i+2})$ back to the reader.

Step 9: The reader forwards this message to the database which then updates the session identifier corresponding to that tag to the new value.

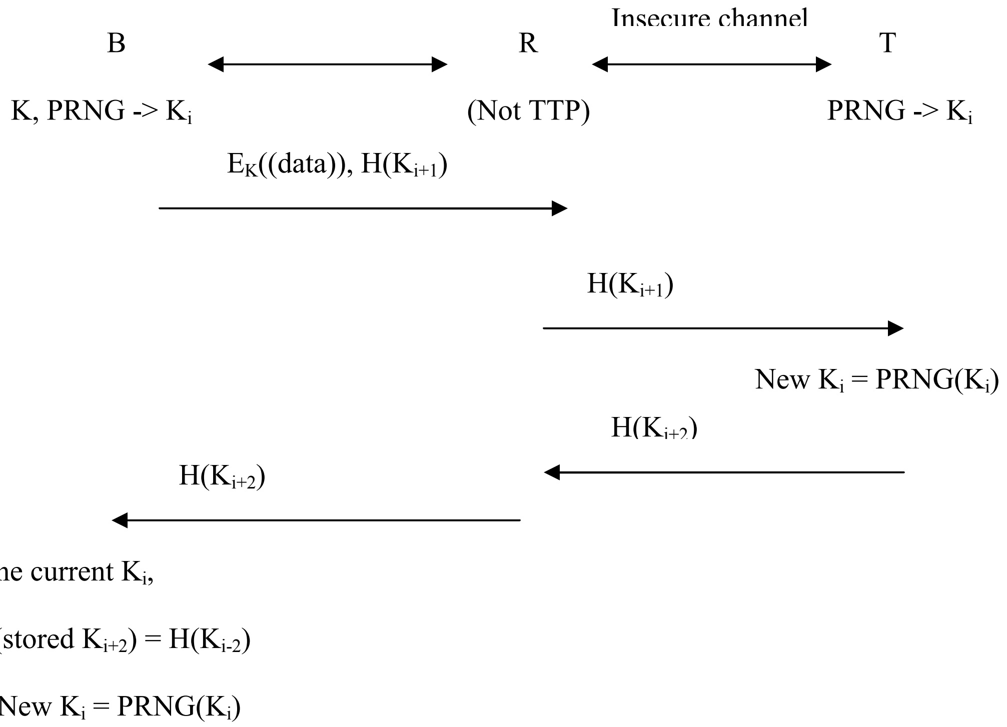


Figure **5-3**: The synchronization steps of the protocol.

Handling Collisions in the Database

Due to the limited computational capabilities of the tag, implementing a perfect collision resistant hash function is difficult. Thus, when updating keys, the new hash $H(EPC\oplus K_i)$ might collide with an entry in the DB. In such cases, adding the tag entry in the already existing entry will not help. The DB when hashing to the entry will not be able to determine which tag's information to send back to the reader. Thus we would need to update to a key such that the computed hash does not collide. This is done by walking down the sequence of random numbers generated using K_i as the seed. For each of these, calculate hash of EPC and the number and check if there is a collision. The first number that does not cause collision is the new session id. Δ tracks the number of steps walked down the sequence number to get the new session id. This operation is denoted by $f^{\Delta}(K_i)$. (f is the PRNG). Send this Δ along with the challenge to the tag. The tag would basically walk down the same sequence (generated by the PRNG with shared seed) Δ times to hit on the new session id to be used. The entire challenge response is shown in figure 5-4.

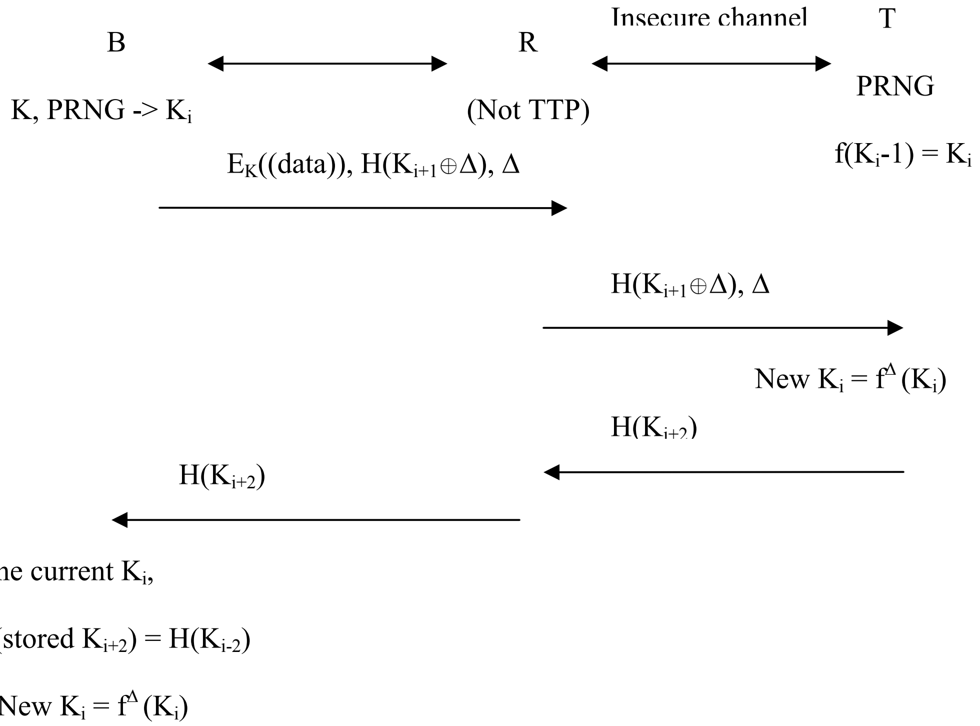

Insecure channel
B
R
T
K, PRNG -> K_i
(Not TTP)
PRNG
$f(K_i-1) = K_i$
$E_K((data)), H(K_{i+1} \oplus \Delta), \Delta$
$H(K_{i+1} \oplus \Delta), \Delta$
New $K_i = f^{\Delta}(K_i)$
$H(K_{i+2})$
$H(K_{i+2})$
For the current K_i,
if H(stored K_{i+2}) = $H(K_{i-2})$
then New $K_i = f^{\Delta}(K_i)$

Figure **5-4**: Handling collisions at DB

To make sure an adversary does not modify the value of Δ when it is sent from reader to tag, it is XORed with K_{i+1} and then hashed. The fact that DB and tag know PRNG and the current seed ensures that the communication is protected. Unless the adversary knows the current seed and PRNG, he cannot decode this message.

The different threats and privacy issues in RFID are discussed in detail in the next chapter. An analysis of the robustness of the above protocol to all these attacks is carried out.

Chapter 6

Security and Privacy analysis of the protocol

Security Analysis

Data Confidentiality: The tag id EPC is hashed with the random number generated by the PRNG. Thus, an eavesdropper cannot make sense by overhearing the message transmitted from tag to reader. Similarly, the messages transmitted between the reader and the database are encrypted with the shared key and hence confidentiality is guaranteed.

Tag Anonymity: Since the tag identifier or EPC code is hashed when sent, the tag id cannot be obtained by an attacker hence guaranteeing tag anonymity.

Data Integrity: The hash of the EPC with the pseudorandom number is sent to the reader, which is then forwarded to the database. If an attacker replaces the hash with a different message, then the tag is not authenticated to the database and hence reader will not be able to receive the tag's content. Thus, modifying the data would not help an attacker to gain information about the tag. The protocol guarantees data integrity by using one-way hash functions.

Privacy Analysis

Forward security: An eaves dropper could eavesdrop and store the communication between tag and reader at different sessions. With the knowledge of these messages, the attacker cannot ascertain the identity of tag or tag's content as the session identifier is refreshed every session and the message transmitted from tag is hashed. If the seed of the PRNG that is common between tag and database can be accessed by the attacker, then he can determine the session identifiers and track all the messages he collected in the past. To get hold of the seed, the attacker would have to physically tamper the tag. To avoid the disclosure of tag information on physical tampering, the seed is overwritten with the session identifier when it is calculated every session. In this way, forward security is guaranteed. Once the tag is compromised and the current session identifier is known, the attacker can understand all the future messages between the compromised tag and reader. This is equivalent to taking control of the tag, the cost of which is higher than the cost of tag production and hence we do not consider theses attacks in our threat model.

Location tracking prevention: Since the message sent by the tag differs every session (as it is the hash of the session identifier which is refreshed every session), a tag's location cannot be tracked from the messages sent by it to the reader, hence guaranteeing location privacy.

Threat Analysis

Man-in-the-middle attack: An attacker can impersonate as a valid reader and query a tag of population to singulate a tag. Upon singulation, he can obtain M. Then, he impersonates as a valid tag to the reader when it queries. This is called man-in-the-middle attack. This attack is very powerful because in warehouses, an adversary could obtain M and then ask his accomplice to steal tag. He could then impersonate the tag to a reader and fool it and the DB in believing the product still exists in the warehouse. Without including the initial steps where the reader obtains a handle from DB, this attack is possible. The handle r from DB authenticates the reader and the tag. The tag would embed $f(r \oplus K_i)$ in its M. Thus while impersonating as tag to the reader (whose handle is r_1 say), an attacker will not be able to modify the message to include $f(r_1 \oplus K_i)$instead of $f(r \oplus K_i)$. Thus the message would be ignored at DB.

Impersonating as a valid reader: An unauthorized reader can impersonate as a valid reader by replaying the message valid reader sent to the DB in the previous session. For the reader to forward this message to the database, it would need a shared secret key. Thus, it will be unable to authenticate itself to the database thereby cannot obtain information about the tag.

Tag cloning: A tag can be cloned only if the seed is obtained. But since the seed is replaced by the session identifier every session, reply from a cloned tag cannot be identified by the database and thus the attacker will not be successful.

Replay attack: A malicious reader could replay message from an authentic reader to the database in efforts to obtain the tag content. If this happens in the current session, the database would reply with the tag content hashed with current session id which can not be deciphered by the malicious reader. The malicious reader could attempt to replay the message in a different session. But in that case, the tag would not be identified as the session id would have changed. The attacker could impersonate as a valid tag and replay a previously overheard reply from a tag to a query. This would end at the database as the tag would not be identified for the same reason as above. Thus the protocol is robust against replay attacks.

Attack against communication between tag and reader: The handle f(r), the hash values make the messages tamper resistant. An attacker trying to modify the messages will not succeed.

Attack against location tracking: The tag response changes every session; hence the tag cannot be tracked across multiple sessions by overhearing the communication between the tag and the reader.

Attack against user privacy: Since the tag content is protected by the hash and the session id, user owning the tag cannot be tracked and neither can an attacker eavesdropping the communication between R and T obtain any information about the user based on the messages overheard. Thus user privacy is maintained by this protocol.

Forgery resistance: The EPC of the tag is maintained in the reserved memory and cannot be tampered with. The only information that would be available to an attacker on tampering with a tag would be the current session id. This does not yield information about past transactions.

Data recovery: During synchronization phase, if an active attacker disrupts either the update message from database to the tag or from the tag to the database, then the two ends would have different session keys thereby being desynchronized. The tag cannot be identified afterwards. Thus, it is necessary to maintain the old state (i.e. state in previous session) to recover data incase of loss of messages or desynchronization attack. This can be carried out at the database by having a pointer stored to the previous entry.

Attack against seed: The session key is generated using a PRNG and the current session id is stored temporarily during that session only. The initial seed is overwritten by the id every session and hence tag tampering will not reveal the seed to the attacker.

Chapter 7

Design and Implementation

The RFID system is implemented using threads in C++. Each of the tags, reader and DB are implemented as separate classes that override functions implemented in a parent class. Q protocol for tag singulation is implemented. Once the tag is singulated, the proposed authentication protocol is carried out to obtain tag's content. Air interface is simulated using event programming in C++. The observer-observable pattern is used to simulate reader broadcast messages. Malicious reader and tag are implemented as separate classes.

All objects (tags, readers, server) maintain queue to store messages. Queue processing is implemented using threads for concurrent processing of messages by the tags. The database is implemented using hash table with the metaID of the tag being the key that is used to lookup the tag information.

For every key update, the key of the hash table corresponding to the tag entry is updated to the new session id. Hash tables support efficient lookup, insertion and deletion of elements in constant time O(1).

The protocol starts at the Q protocol for singulation and then runs the proposed protocol. The simulation ends when the entire tag population is singulated once. By increasing the number of iterations, the population of tags can be singulated more than once. In each of the iterations, the attacks discussed below are carried out by having separate class for malicious reader. After the iteration completes, we check that all tags have been singulated by examining the log; the protocol is then robust to the attacks.

A measurement of the average singulation time per tag for different tag populations shows that the proposed protocol performs in constant time and thus does not add any overhead to the singulation time. The singulation time is dominated by the performance of the Q protocol.

Attack Implementation

Impersonating reader: A malicious reader overhears the FWD_TAG_REPLY message and then resends it to the DB. The malicious reader object stores this message in its queue and resends the message when it hears the END_SESSION message from the DB. If the replay attack is carried out after the session ends, then the tag's session identifier would have changed and hence the DB will not find an entry in the hash table corresponding to the old hash thereby ignoring the message. If the replay attack is carried out during the session before the session identifier is updated, then the database would reply back to the message with the hash of tags content and session identifier. The malicious reader does not know the session identifier and hence cannot understand the message.

Impersonating tag: A malicious reader tries to reply to a reader with TAG_REPLY it overheard in the previous session. Like above, since the tag hash would have changed, the DB would ignore the message forwarded by the reader.

Desynchronization attack: To overcome desynchronization, we explicitly make the tag ack the update key message. This ack is forwarded to the DB to confirm the update of key at the tag side. Disrupting ACK_UPDATE_KEY would leave an updated id at tag but not at DB. The DB in subsequent rounds will not be able to recognize the tag. To overcome this problem, we have pointers to the previous entry of the DB. The pointer is updated to the last recently used hash for the tag during the update phase.

Man-in-the-middle attack: The malicious reader object queries tags with a random r_1 and then stores the tag's reply in the queue. When the valid reader sends the 'Query' message with the handle ‘r’, the malicious reader replies with the stored message. The valid reader then forwards this message to the DB. Since the forwarded message would have $f(r_1 \oplus K_i)$ while the reader would have sent r to the DB, the message will be ignored. Hence, man-in-the-middle attack is effectively countered.

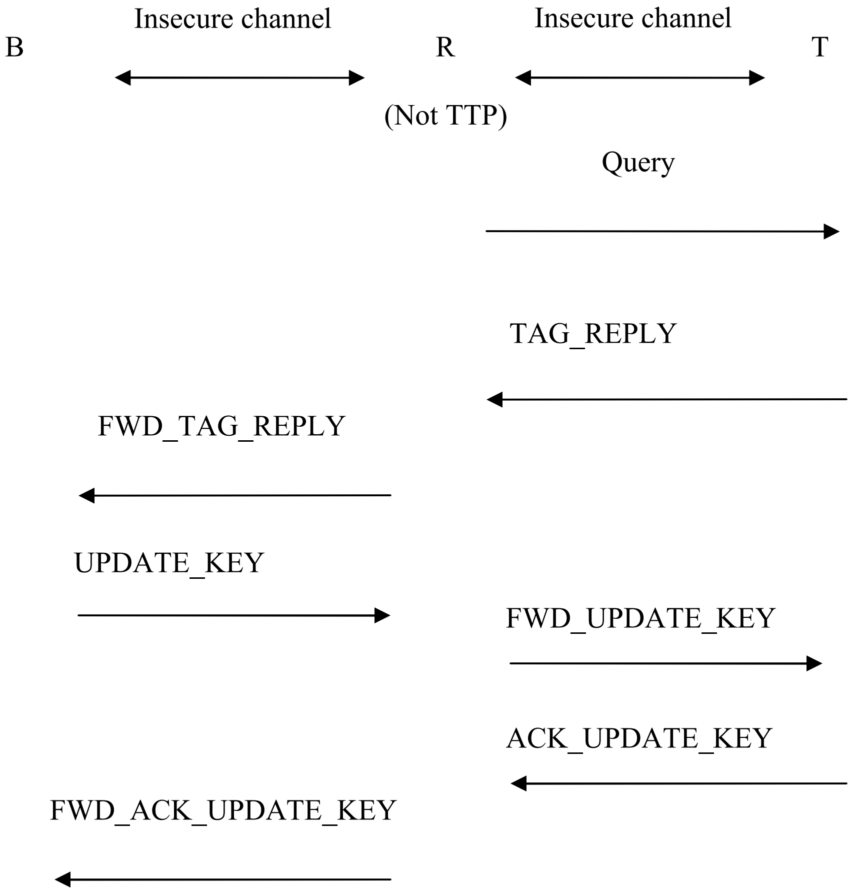


Figure **7-1**: The protocol framework

Chapter 8

Conclusion

In this thesis, the design and analysis of authentication protocols for low-cost RFID systems have been studied. Work on different hash-based protocols, one time pads as well as security schemes for RFID have been reviewed. Some of these have been the basis for the protocol proposed in the thesis. The proposed mutual authentication protocol for reader-tag communication provides security against eavesdropping, man-in-the-middle attacks, location tracking, desynchronization attacks, cloning of tags as well as denial of service attacks. It is computationally light-weight and anonymously interacts between entities. The proposed protocol basically fits the low-cost RFID system environment. The tag has a hash function and a pseudorandom generator whose input would fit in the small memory. With this minimal cryptographic primitive, our protocol provides the mutual authentication between the tag and the back-end server and anonymously interacts. The protocol is robust since it counteracts the replay attack and man-in-the-middle attack even when the reader is not a trusted third party and the communication channels are insecure. As all authentication messages are randomized and the tag contains just its unique identification data, the user data privacy and the location privacy are guaranteed. The protocol provides a secure framework that counters all attacks in addition to the fact that it does not provide any additional overhead to the singulation time. In the proposed protocol, reader authentication and prevention of active attacks are provided based on the assumption that a reader is no more a trusted third party and the communication channel between the reader and the back-end server is insecure. As tags only have hash function, PRNG and exclusive-or unit, the proposed protocol is practical for low-cost RFID environment.

Bibliography

[1] Juels, A.: RFID Security and Privacy: A Research Survey. RSA Laboratories (2005)

[2] Dimitriou, T.: A lightweight RFID protocol to protect against traceability and cloning attacks. In: SecureComm 2005, Security and Privacy for Emerging Areas in Communications Networks – 2005, pp. 59-66 (September 2005)

[3] EPCglobal, Available at http://www.EPCglobalinc.org/home

[4] Hung-Yu Chien and Che-Hao Chen, "Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 Standards, "In Computer Standards and Interfaces, 2006

[5] Jeongkyu Yang, Jaemin Park, Hyunrok Lee, Kui Ren and Kwangjo Kim, "Mutual Authentication Protocol for LOW-COST RFID," In the Encrypt Workshop on RFID and Lightweight Crypto, 2005

[6] D. Henrici, and P. Muller, Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers, In the Proceedings if PerSec'04 at IEEE PerCom, Mar. 2004.

[7] Simson. L. Garfinkel, "Adopting Fair Information Practices to Low Cost RFID Systems", Ubicomp 2002.

[8] G. Avoine. Online bibliography: Security and privacy in RFID systems, 2008.

[9] S. Weis, S. Sharma, R. Rivest and D. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems", in 1st Intern. Conference on Security in Pervasive Computing (SPC), 2003.

[10] M. Ohkubo, K. Suzuki, and S. Kinoshita, Cryptographic Approach to Privacy-Friendly Tags, RFID Privacy Workshop 2003, MIT, MA, USA, Nov. 2003.

[11] A. Juels, Minimalist Cryptography for Low-Cost RFID Tags, Available at http://www.rsasecurity.com/rsalabs/node.asp?id=2033.

[12] S. Weis, Security and Privacy in Radio-Frequency Identification Devices, Master's thesis, MIT, 2003.

[13] David Molnar and David Wagner, Privacy and security in library RFID: Issues, Practices, and Architectures, In Conference on computer and communications security – CCS'04

[14] Younghwa An and Soohyun Oh, RFID System for user's privacy protection, In Asia-Pacific Conference on Communications, 2005.

[15] Keunwoo Rhee, Jin Kwak, Seungjoo Kim and Dongho Won, Challenge response Based RFID Authentication Protocol for Distributed Database Environment, In International Conference on Security in Pervasive Computing – SPC 2005

[16] Pedro Peris-Lopez, Julip Cesar Hernandez-Castro, Juan M. Estevez-Tapiador, Arturo Ribagorda, LAMED – A PRNG for EOC Class-1 Genration-2 RFID soecification.

Appendix A

Glossary

Active tags - An RFID tag with on-board power source, such as battery, and an active transmitter, page 5.

Auto-ID - automatic identification, page 2 and 10

Backward channel - The communication channel from tag to reader, page 8

Collision - Interference resulting from simultaneous transmissions on the same frequency, pages 6, 11 and 14

Collision resistant hash function - A hash function that provides 2nd preimage resistance and collision resistance, page 14

EPC - Electronic product code, page 2

Forward channel - The communication channel from reader to tag, page 8

Gen2 tags – EPCglobal generation 2 RFID tags, page 11

Hash function - An efficiently computable function which maps an arbitrary length input to a fixed length output, pages 12 and 14

Hash lock - Access control mechanism, page 15

IC - Integrated circuit, page 4

Location privacy - The ability to prevent other parties from learning one's current and past locations, page 8

Low cost tags - RFID tags priced at around 5 cents, pages 1, 2 and 12

metaID - The hashed key value which acts as a temporary ID for a tag, page 8

nonce - A random string used to pad messages; used for introducing freshness to the message, page

One way hash function - A hash function offering preimage and 2nd preimage resistance, page 14

Passive tags - An RFID tag which receives power from a reader, necessarily with a passive transmitter, page 5

RFID - Radio Frequency Identification, page 2

Semi-passive tags - An RFID tag with an on-board power source, but with a passive transmitter, page 5

Singulation - addressing and isolating a single tag from a population of tags by using anti collision protocols, pages 6 and 21

Appendix B

List of Notations

DATA – Information about the RFID tag

T – RFID tag

R – Reader

B – Database/ Server

EPC – The Electronic Product Code of the tag

E_k() – Encryption using key k

f(r) – Pseudorandom generator with seed r

h() – One way hash function

K_i – Temporary session identifier

k – Shared secret between the reader and database

PRNG – Pseudo random generator

r_1, r_2 – Nonce

$\oplus$ **–** Exclusive OR function

|| – concatenation