Addressing Security and Privacy Issues in Low-Cost RFID Systems

Submitted by

Zeeshan Bilal

for the degree of Doctor of Philosophy

of the

Royal Holloway, University of London

2015

Declaration

I, Zeeshan Bilal, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed.....(Zeeshan Bilal)
Date:

I dedicate my thesis to my parents who has always been instrumental in every success of my life.

Abstract

Radio Frequency Identification (RFID) systems are being used in numerous applications such as transportation ticketing, animal tracking, supply chain management, medical records, electronic passports and identity cards. These systems consist of three main components, namely: server, reader and tag. A *tag* is a small microchip with antenna attached to an item which needs identification. A *reader* scans a tag(s) and collects the identification information. This information is then passed on to a *server* by the reader for further operations.

Providing security and preserving privacy of these systems come with a cost. In sensitive applications such as e-passports, the embedded tags are resourceful enough to accommodate standard cryptographic functionality. These resourceful tags are high-cost. However in the most widely deployed RFID systems, such as in supply chain management of daily consumer goods, it is not feasible to use such high-cost tags. Therefore the tags used in these applications are low-cost tags which are constrained in their resources. Since these tags cannot afford the luxury of conventional cryptographic primitives, low-cost RFID systems are prone to both passive as well as active adversaries. Some of the typical threats related to an RFID system include tag cloning, impersonation, replay, relay, de-synchronization, DoS, content privacy leakage, tracing and tracking attacks, etc. Therefore it is imperative to think out of the box to provide security and privacy to these low-cost RFID systems.

This thesis makes six contributions in this regard. In the first and second contribution, very basic low-cost tags are considered. These tags are very constrained with respect to their resources. To secure such tags, researchers have proposed *ultra-lightweight mutual authentication protocols* (UMAPs). First we demonstrate multiple attacks in detail on two of such UMAPs. Then we carry out analysis of existing UMAPs and highlight weaknesses. We also propose a new UMAP which overcomes the weaknesses of existing discussed schemes.

The next three contributions focus on the most widely used application of RFID systems, supply chain management. This application generally uses a standard *EPC-global Class-1 Gen-2* (EPCC1G2). We contribute by first proposing a scheme which provides security and privacy to tagged items throughout a supply chain cycle with online as well as offline readers. Then we focus our work on the counterfeit problem in supply chain management, which causes huge losses to businesses. We propose a hierarchical anti-counterfeit mechanism to counter the problem of counterfeiting during the supply chain cycle. Finally we devise a framework to provide an anti-counterfeiting feature to individual customers who cannot afford the luxury of standard readers and access to a back-end database server.

Lastly we discuss the problem of ownership transfer in RFID systems. Since tags travel to different geographic locations, there is a need of ownership transfer, where an *owner* is an entity which can interact with the tag using a shared secret key. A simple ownership transfer involves transfer of a shared secret key from old owner to new owner. This raises concerns where an old owner would retain a copy of the key and can still interact with the tag even after its ownership is revoked. Similarly, if the key is not changed before transfer, a new owner can trace past transactions of an old owner. We propose a secure ownership transfer scheme which meets certain requirements. We further elaborate on additional properties required to achieve a robust ownership transfer process.

Acknowledgement

First of all I would like to thank Almighty God who has bestowed his countless blessings on me. Then I would like to express my deepest gratitude to my supervisor, Professor Keith Martin, for his care, guidance and patience during the course of my research. He transformed me into an independent researcher by giving me freedom and providing me insightful discussions about my research. He has also made a major impact on how to convey my research to wider audiences in the form of research papers, posters and presentations.

I would also like to thank my advisor Dr. Kostas Markantonakis for his availability and guidance whenever required. A token of thanks to Dr. Gerhard Hancke and Dr. Geraint Price for their patience and useful feedback during my review meetings. I am thankful to Qasim Saeed and Dr. Colin Walter for our joint work.

My parents have always been my inspiration. I have achieved everything only due to their prayers and efforts. My two brothers, my sister, and their respective families always brought a smile to my face in my low times.

Finally I would like to say thanks to the whole Information Security Group for making my stay very comfortable and providing me whatever I needed. To Royal Holloway University of London and specially to Founder's building, thank you for giving me such an amazing and unforgettable time of my life.

Contents

List of Figures

List of Tables

List of Notation

T	A tag.
R	A reader.
S	A back-end database server.
$\oplus$	Exclusive-OR operator.
$\|\|$	Concatenation operator.
$HW(X)$	Hamming weight of bit string X.
$Rot(X, Y)$	Left rotation of argument X by Y.
EPC	An EPCC1G2 tag's 96-bit static and unique identity.
$Access$	A built-in 32-bit unique access password in each EPCC1G2 tag.
r	A random challenge.
KS	A shared symmetric key.
(K_p, K_s)	A public and private key pair.
(K_{sign}, K_{ver})	A signing and verification key pair.
$E_K(M)$	An encryption function.
$D_K(M)$	A decryption function.
$H(M)$	A hash function.

List of Abbreviations

RFID	Radio Frequency Identification
UMAP	Ultra-lightweight Mutual Authentication Protocol
ALU	Arithmatic and Logic Unit
EPCC1G2	EPCglobal Class-1 Gen-2 Standard
EPC	Electronic Product Code
UHF	Ultra-High Frequency
LSB	Least Significant Bit
MSB	Most Significant Bit
XOR	Exclusive OR

Chapter 1

Introduction

This chapter outlines this thesis. Section 1.1 gives an overview of technology, application and challenges which form the basis for this research. The focus of this research is given in Section 1.2. The structure of the thesis is presented in Section 1.3.

1.1 Overview

A *Radio Frequency Identification* (RFID) system is used to identify an object remotely using radio waves and is made up of tags, readers and a back-end server.

A small transponder is attached to the object which needs identification. This small transponder is called a *tag*, which is a small chip with antenna. The chip has memory that stores the identification information of a particular object. This chip may also have a small processor to perform some computations if required. The antenna is used to transmit and receive information. A tag is classified according to its power source. A *passive tag* does not have its own power source whereas an *active tag* has one.

A *compatible reader* is a device which can scan/read a tag in its vicinity. Compared to a tag, a reader is a resourceful entity with antenna, modem, processor, storage and its own power supply. It transmits signals at a prescribed frequency, power and format to not only query a tag, but also to power up a passive tag.

A *back-end server* is connected to multiple readers. This gathers identification information from tags using these readers as intermediary devices. It then stores/verifies this information in its database for further processing.

RFID systems have been deployed in numerous applications. Some examples include access control, transportation ticketing, animal tracking, patient medical history, toll payments, vehicle identification, library administration, electronic passport control,

inventory and supply chain management [146]. RFID systems are also diversified in their standardization. Some systems are proprietary while others follow application-specific standards [151]. RFID system are used in these applications to achieve the following objectives [63]:

1. **Unique Identification.** Each tagged object is identified uniquely, including objects within a homogeneous collection. For example, a tagged biscuit pack can be uniquely identified in a crate of other biscuit packs of the same manufacturer and brand.

2. **Automation.** Tags are identified automatically without any requirement for a line-of-sight or physical connection. This allows RFID tags to be identified anywhere within scan range.

Although there are many benefits of using this technology, RFID systems have associated security and privacy concerns which arise due to the following reasons [63]:

- Communication between a reader and a tag is wireless and hence can be eavesdropped.
- Tags can be read by any compatible reader promiscuously.
- A rogue reader may emit a stronger signal than prescribed to scan tags at longer distances.
- Tags are not only inconspicuous, but a tag holder does not know when a tag is transmitting information or to whom.

Therefore there is a clear need to address security and privacy concerns in RFID systems.

1.2 Focus

While considering security and privacy of RFID systems, it is imperative to keep the following questions in mind:

- In which application will the RFID system be deployed and what are the desired security properties?
- Does the RFID system under consideration has to comply with any standards?
- What resources are available to the RFID tag?

Providing security and privacy for RFID systems is a challenge because of the resource constraints of the tag. Basic cryptographic primitives require considerable amounts of storage, computation, power consumption and communication overheads. These additional resources increase the overall cost of the system. Therefore, while designing security for RFID systems, it is imperative to make a trade-off between cost, performance and level of security.

The main security and privacy requirements vary depending on the application of an RFID system. For example, in an e-passport system cost is not an issue but the level of security is important. Therefore RFID tags employed in such an application are expensive and resourceful. Whereas, in a supply chain management system of consumer goods, the level of security can often be degraded in order to keep the cost of the RFID tags low. There is no standard criterion but generally a *low-cost tag* should cost a few pence whereas a *high-cost tag* can cost as much as several pounds sterling.

In [22], a tag classification based on the operations supported on-chip is proposed. High-cost tags are divided into two classes: *full-fledged* and *simple*. Likewise, there are two classes for low-cost RFID tags: *lightweight* and *ultra-lightweight*.

1. **Full-fledged.** These tags support on-board conventional cryptography like symmetric encryption, cryptographic one-way functions and even public key cryptography.

2. **Simple.** The chip on these tags can support random number generators and one-way hash functions.

3. **Lightweight.** These tags are those whose chip supports a random number generation and simple functions like a Cyclic Redundancy Code (CRC) checksum, but not a cryptographic hash function.

4. **Ultra-lightweight.** These tags can only compute simple bitwise operations like XOR, AND, OR, etc.

Low-cost tags (lightweight and ultra-lightweight) pose the biggest challenge in terms of security and privacy. These tags are in widespread use and have very limited resources to accommodate security primitives. The focus of this thesis will be to address security and privacy requirements in low-cost tags. A comparison, between low and high-cost tags, is shown in the Table 1.1.

Table 1.1: Comparison between Low-cost and High-cost RFID Tags.

Specifications	**Low-cost Tags**	**High-cost Tags**
Cost	Few pence	Several pounds
Standards	EPC Class-1 Generation-2 ISO/IEC 18000-6C	ISO/IEC 14443 A/B
Storage	64 bits - 1 kilobytes	up to 128 kilobytes
Power Source	Passive	Passive and Active
Computation	250-4000 gates simple ALU	Fully capable microprocessor

1.3 Structure

This thesis focuses on analyzing schemes for addressing security and privacy issues in low-cost RFID systems. The thesis is divided into the following parts:

Part 1: Setting the Scene

This part consists of Chapter 2, which provides a fundamental background to RFID systems. This includes a detailed discussion of the main components of an RFID system, its interface, regulations, standardization and various applications. Finally it explains why security and privacy risks arise in these system and how security solutions are classified.

Part 2: Ultra-lightweight Mutual Authentication Protocols (UMAPs) : Weaknesses and Countermeasures

This part consists of Chapter 3 and Chapter 4. Here we will examine secure solutions appropriate for ultra-lightweight tags.

- **Chapter 3.** We analyze weaknesses found in two ultra-lightweight mutual authentication protocols (SIDRFID and DIDRFID) presented in [80] and discuss

multiple attacks on both protocols. This work has been accepted for publication [12].

- **Chapter 4.** This chapter generalizes weaknesses in a number of existing UMAPs. We then suggest countermeasures to overcome the highlighted weaknesses. The countermeasures are presented in the form of a new UMAP which builds on the strengths of the existing schemes. This work has been published in [9].

Part 3: RFID Systems in Supply Chain Management

This part consists of Chapter 5, Chapter 6 and Chapter 7. One of the most researched applications of RFID systems is their use in supply chain management. While RFID systems for supply chain management can provide high performance, there are many outstanding security and privacy issues which need to be addressed.

- **Chapter 5.** In supply chain management, a tagged item travels from manufacturer to end-user/customer. The tag starts its journey in a secure environment where readers share secrets with corresponding tags (*online readers*) and moves to insecure environment where readers are positioned at different geographic locations and do not possess secrets corresponding to the tags (*offline readers*). A tag's user privacy can be easily compromised in insecure environment if appropriate measures are not taken. This chapter presents an online/offline adaptive approach to achieve desired security and privacy goals throughout a supply chain management system. The suggested scheme is designed for *EPCglobal Class-1 Gen-2* (EPCC1G2) standard compliant tags [48] but can be modified in order to be suitable for similar resource-constraint environments. This work has been published [11].

- **Chapter 6.** Detecting a counterfeit in supply chain management is a laborious and time consuming task. Though RFID systems can speed up the process of identification, these systems are vulnerable to a genuine tag being cloned and attached to a counterfeit item. Tagged items travel in groups in supply chain management depending on their type, lot number and expiry date, etc. In this chapter, a hierarchical anti-counterfeit mechanism is designed that can detect both counterfeit and missing items. This mechanism also helps to identify dishonest middle parties. The proposal is suitable for EPCC1G2 standard compliant tags [48] and can be extended to other standards. This work has been published in [10].

- **Chapter 7.** Since *ultra-high frequency* (UHF) readers are not available to individual customers, it is not feasible to verify the authenticity of tagged items at a customer level. We design a customer level anti-counterfeit framework that uses *near field communication* (NFC) technology in smart phones to detect counterfeits. A valuable item is linked to two tags (one EPC compliant and one NFC compliant). The tagged item is processed in the supply chain using the EPC tag until the item reaches the end-user/customer. The customer then uses their NFC enabled device to determine the legitimacy of the item by running an authentication protocol with the NFC tag. This work has been published in [125].

Part 4: Ownership Transfer in RFID Systems

This part consists of Chapter 8 which deals with the scenario where a tagged item changes its ownership. The ownership is associated with the possession of a secret key. Thus only an owner of a tag can interact with it using its respective shared secret key. However when a particular tag is transferred/sold, the new owner needs this secret so that it can also interact with the tag. This transfer should be secure, where the secret key of old owner should not be exposed to new owner and vice versa. We propose a robust ownership transfer process which is not only secure but also achieves additional properties.

Part 5: Conclusions and Future Work

The conclusions of this research are drawn in Chapter 9. Future research directions are also discussed.

Part I

Setting the Scene

Chapter 2

Background

This chapter provides a background to RFID systems. Section 2.1 defines the main components of an RFID system, the type of interface used is discussed in Section 2.2, and regulations and standardizations governing RFID systems are listed in Section 2.3. Different applications of RFID systems are given in Section 2.4. Section 2.5 explains some of the reasons why risks arise relating to RFID systems. An introduction to formal analysis methods is given in Section 2.6.

2.1 Components of RFID Systems

RFID is a wireless technology that enables identification of tags attached to items over a radio link. A short history of RFID is given in Appendix A. RFID systems can support a larger set of identifiers than bar codes [144]. They can also handle additional information such as manufacturer, product type and even monitor environmental factors such as temperature, humidity, etc. In this section, the main components of RFID system are discussed.

2.1.1 RFID Tags

An RFID tag consists of a microchip with logic gates for computation, memory for storage, and a coupling interface, such as an antenna coil for communication. Cost is a major concern when designing different components of a tag. A diagram of an RFID tag is as shown in Figure 2.1.

- **Power Source.** Tags can be classified as either *active* tags having their own power supply (some tags are classified as *semi-active* as their batteries are only activated in the presence of a reader) or *passive* tags drawing power from the

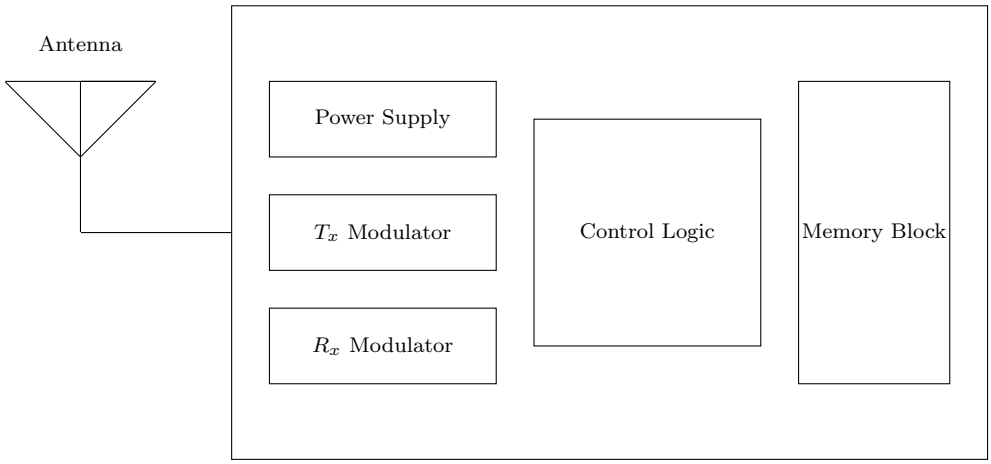

Figure 2.1: Block Diagram of a Tag.

signal received from the reader. The battery, on active tags, adds sensory and data logging capabilities, supports larger memories and increases the scan range. Passive tags receive power from the reader in order to perform computations and transmit data. Low-cost tags tend only to be passive.

- **Transmitter and Receiver Modulators.** These modulators are used for coding bit-streams into radio waves and decoding radio waves into bit-streams between readers and tags.

- **Control Logic.** Depending on the functionality and cost of an RFID tag, it can have different computational capabilities. Low-cost tags generally have a simple *arithmetic and logic unit* (ALU) consisting of 250-4000 logic gates. High-cost tags can afford the luxury of a fully capable microprocessor. Depending on the tag category, its programming can be done at the manufacturing level or at the application level.

- **Memory Block.** Each tag has *read only memory* (ROM) and *random access memory* (RAM) depending on its application. These memory blocks can be read-only, write-once read-many, or fully rewritable. Typically passive tags have a range of 64 bits to 1 kilobyte of non-volatile memory. These passive tags normally use *electrically erasable programmable read only memory* (EEPROM). Some passive tags are laser programmed at the manufacturing process. Active tags have memories as high as 128 kilobytes and use battery-supported *static random access memory* (SRAM).

- **Antenna.** The antenna is used for transmitting/receiving the signals between

reader and tag. More details about its coupling with the communication is given in Section 2.2.

2.1.2 RFID Readers

RFID readers have a radio frequency module, a control unit, and a coupling element in order to interrogate tags using radio frequency communication. RFID readers are not considered to have issues with regard to internal storage and processing capabilities. Therefore, any costly cryptographic operations such as random number generation tend to be handled by RFID readers rather than tags. These readers are connected to a back-end server through a secure communication channel such as SSL/TLS.

2.1.3 Back-end Server

Since tags can store, process and communicate relatively few bits of information due to resource limitations, tag data tends to be transmitted to a back-end server and used as an index (pointers, randomized identifiers, etc.) to a database for retrieval of detailed information associated with a particular tag. The reader acts as an intermediary between tag and server to exchange this information. It is assumed that the connection between readers and the back-end server is secure.

2.2 RFID System Interface

Active tags generally have different transmitter and receiver functionalities supported by a power source. Therefore active tags may respond at a different frequency than the reader's interrogation signal. These tags normally operate at 433MHz *ultra high frequency* (UHF) in military applications, at microwave and ultra-wide band ranges.

Passive tags receive power from readers for computation and communication. Energy is transferred using coupling via electromagnetic fields [47]. RFID tags use either electric field or magnetic field (or both) to receive power from a reader. The signal sent from reader to tag must be used simultaneously to transmit both information and energy. Most RFID systems operate in ISM bands [149] which are designated by the International Union of Telecommunications. The most commonly used ISM frequencies for RFID systems are 13.56 MHz and 860-960 MHz. Each band has its own radiation power and bandwidth regulations.

There are various methods of transferring the data to a reader. Passive tags usually use *passive backscatter* or *inductive coupling* (see Figure 2.2). In passive backscatter, reader transfers energy to the tag by emitting electromagnetic waves through the air.

The tag uses RF energy to charge up, receives command/data signals and responds accordingly. Inductive coupling is used by *low frequency* (LF) and *high frequency* (HF) band RFID devices. The reader's antenna uses a current to generate the magnetic field. The antenna on the tag, when exposed to this magnetic field, generates a current in the tag that powers up its circuitry. Circuitry on the tag switches the impedance load of the tag's antenna according to the data stream, causing modulation of the magnetic field joining reader and tag. This is demodulated by the reader to extract useful information.

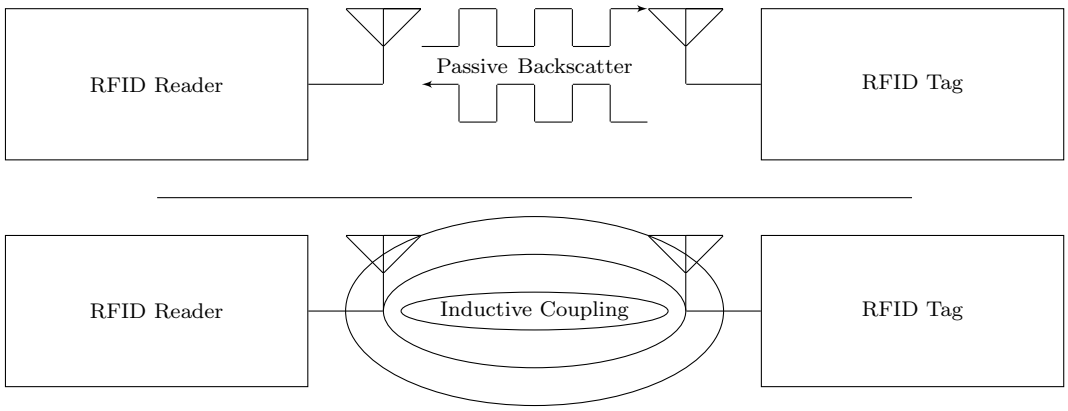

Figure 2.2: RFID System Interface for Passive Communication.

2.2.1 Coding and Modulation.

The exchange of data between reader and tag, and vice versa, must be produced efficiently; so both coding and modulation are used. The coding/modulation are defined according to the existing limitations in the forward (reader to tag) and the backward (tag to reader) channels. Readers are able to transmit greater power, but have bandwidth limitations. Tags have power limitations. The power source classification of an RFID system is as shown in Table 2.1.

The modulation scheme determines how the bit-stream is transmitted between readers and tags, and vice versa. Some solutions include: *amplitude shift keying* (ASK), *frequency shift keying* (FSK) and *phase shift keying* (PSK). The choice of modulation type is based on power consumption, reliability and bandwidth requirements. In the forward channel, either *manchester* or *non-return-to-zero* (NRZ) is used. Whereas in the backward channel, either *pulse-position modulation* (PPM) or *pulse-width modulation* (PWM) coding techniques are preferred.

Table 2.1: Power Source Classification

Type of Tags	Internal Circuitry	Types of Communication
Passive	Power from Reader	Passive backscatter or inductive coupling
Semi-active	Internal Power	Passive backscatter or inductive coupling
Active	Internal Power	Transmits and receives RF signal

2.2.2 Collisions in RFID System

Collisions between tags happen when multiple tags simultaneously answer a reader signal. The anti-collision algorithms used in RFID systems are quite similar to those applied in networks, but they take into account that RFID tags are generally more limited than the average network device. Both probabilistic or deterministic approaches are used. In practice, however, a combination of both is often deployed.

In the case of collision between readers, several readers interrogate the same tag at the same time. The tag in such a case may not respond. One possible solution to this problem consists of allocating frequencies over time to a set of readers by either a distributed or a centralized approach.

2.3 Regulation and Standardization of RFID System

Companies were mainly using proprietary systems before RFID standards began to evolve. Organizations including the International Organization for Standardization (ISO), the International Electro-technical Commission (IEC), ASTM International, DASH7 Alliance, and EPCglobal have set standards for RFID systems. Depending on the availability of frequency bands, the regulations for RFID systems [18, 40] can be categorized as follows:

- **Low Frequency (LF).** These use 125-134.2 kHz and 140-148.5 kHz. LF tags can be read at a distance up to 10 cm. Applications include animal identification, car key-locks and data collection, etc.
- **High Frequency (HF).** This uses the ISM band at 13.56 MHz. HF tags have a scan range from 10 cm to 1 m. These tags are used in smart cards, library books

and clothing identification, etc.

- **Ultra High Frequency (UHF).** This frequency range is 860-960 MHz. UHF tags are read at a distance from 1 to 12 m. The main application is inventory and supply chain management and most wide deployments follow the EPCglobal standardization framework.

- **Microwave.** This transmits at 2.45 GHz and has a read range of up to 30 meters approximately. Applications of microwave tags are highway toll collection and vehicle fleet identification, etc.

The main standard of interest to us is the EPCC1G2 standard [48] since it forms the basis of the work in Part 3 of the thesis (details of other standards can be found in Appendix A).

2.3.1 EPCglobal Class-1 Gen-2 (EPCC1G2) Standard.

The Auto-ID Center, founded in the late 1990s at Massachusetts Institute of Technology (MIT), started working on a standard that would put RFID technology into various applications across the globe, notably supply chain management. In 2003, the work on this standard was taken over by another organization, EPCglobal, which was a joint venture between the European Article Numbering (EAN) and Uniform Code Council (UCC).

The aim of EPCglobal is to establish an international standard for identification of tagged products in supply chains across the globe using passive RFID tags, each having a unique *electronic product code* (EPC). Class-0 and Class-1 were the two protocols used in commercial applications between 2003 and 2005. These protocols specified how to exchange information between a tag and a reader and are known as the *Air Interface Protocols*. The EPCC1G2 Version 1.2.0 standard [48] specifies low-cost UHF tags which operate in the frequency range of 860-960 MHz and have a read range of 2-10 meters.

Each tag is identified with its unique EPC, which is a 96-bit long string (see Figure 2.3). The first 8 bits represent the version number. The next 28 bits are for the organization number as assigned by the EPCglobal consortium. This is followed by 24 bits of product class identification. The last 36 bits carry the unique serial number of the tagged product. Rather like a URL, EPC can be used as an identifier in a global database to uniquely identify a particular product. Further details are given in Chapter 5.

The new version of this standard, Version 2.0.0 [49] was released in November, 2013. The new standard proposes an optional cryptographic suite to be implemented

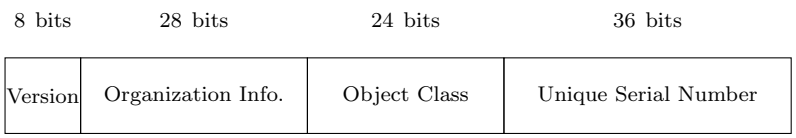

Figure 2.3: Electronic Product Code (EPC) Format.

in RFID tags. It also supports new optional security commands including *Authenticate*, *AuthComm*, *SecureComm*, *KeyUpdate* and *TagPrivilege*. A Tag may support zero, one, or more than one, cryptographic suites. A cryptographic suite defines how a tag and a reader implement a cryptographic algorithm and its functions. A reader selects one from among the implemented cryptographic suites using the Cryptographic Suit Indicator (CSI) field in the Challenge and Authenticate commands. A tag may support up to 256 keys, numbered Key_0 to Key_{255}. The tag manufacturer chooses the number and type of cryptographic suite and number of available keys, and assigns them to the cryptographic suite(s); this assignment is not be alterable in the field. No two keys have the same number, even if used for different cryptographic suites. A tag does not indicate where in memory it stores its keys, nor does it allow a reader to read this memory location. This new version of the standard does not affect the UHF Air Interface Protocol and also supports our work carried out on previous version with regard to cryptographic implementations.

2.4 Applications of the RFID Systems

RFID systems can be used in a variety of applications. Low-cost tags are considered to have widespread potential in future applications [40, 63, 76, 93]. According to a report in 2012 [152], the global RFID market is expected to grow at a compound annual growth rate of around 18% through 2014 to reach approximately USD 19.3 billion. This phenomenal growth surpasses other identification technologies including bar codes. A few of the many applications of RFID system are as follows (see [146] for further examples):

- **Access Control.** Contactless proximity cards (with embedded tags) are used for controlled access to buildings [21]. Car ignition keys are fitted with tags to counter theft and access to vehicles.
- **Automated Payments.** A toll is paid automatically using tags attached to the windscreen of vehicles. The SpeedPass token for petrol station payments, con-

tactless credit-cards, like American Express ExpressPay and Mastercard PayPass are a few examples.

- **Animal Tracking.** Tags embedded in animals are a way to identify and track animal habitat, medication and extract other useful information [138]. Millions of pets have tags for tracking and returning to their owners in case of loss.
- **Public Transport.** RFID systems have led to considerable improvements in the public transportation sector [128]. Tagged tickets/cards are re-usable and can be pre-paid. Losses due to customers not paying and manual checking has reduced considerably.
- **Smart Appliances.** Smart appliances [142] can interact with tags used in the consumer products such as medicine, food and garments. A smart cabinet can set reminders for medicines, a smart fridge can send notifications about expiry dates and, similarly, a smart washing machine can set cycles depending on the nature of a garment. These smart devices can later interact with the web to pass a shopping list on to a home delivery service.
- **Automated Shopping.** Customers can shop in a retail store and all the items in their trolley will be read automatically while passing through exit doors installed with compatible readers. Items will be indexed in the database for their prices. Even the cost can be automatically deducted from the customer's contactless payment card and an e-bill could be sent to their phone.
- **Interactive Objects.** Smart posters enable someone to use an NFC-enabled phone [103] to obtain information, for example show timings, reviews and cast for a movie poster. Similarly a customer can use a smart phone to read a promotional leaflet to obtain detailed information about a product.
- **Supply Chain Management.** One of the biggest impact of RFID systems is considered to be in supply chain management, which also involves logistics and assets tracking. This application will be discussed in detail in Part 3 of this thesis.

2.5 Risks to an RFID System

RFID systems suffer risks just like any other types of system. A research survey on RFID security and privacy issues is given in [63]. The following are the aspects of an RFID system that give rise to security risks, and we indicate how to counter those risks.

- **Resource Constraints.** Since the cost has to be kept low, a tag, particularly in low-cost RFID systems, has constraints on its resources. This in turn lowers the level of security that can be supported. Therefore researchers have to discuss lightweight secure solutions for RFID systems.

- **Wireless Channel.** The communication between a tag and a reader uses radio waves transmitted through the air. This communication can be eavesdropped by any adversary and can be used to carry out different attacks. This channel thus should be encrypted to ensure security of transmitted information.

- **Promiscuous Technology.** Any compatible reader can scan a particular tag and obtain useful information. An authentication mechanism is thus required to restrict access so only legitimate and authorized readers can scan a particular tag.

- **Remote Reading.** Tags can be read at a distance through materials like cardboard, cloth, and plastic. A compatible reader can scan in its wireless range to look for tags. Different scan ranges can be identified as follows [63]:

  - *Nominal Read Range.* The maximum distance at which a normally operating reader can reliably scan tags.
  - *Rogue Scanning Range.* A rogue reader can normally emit a stronger signal and read tags from a larger distance than the nominal range.
  - *Tag-to-Reader Eavesdropping Range.* Read range limitations result from the requirement that the reader powers a passive tag. However, one reader can power up the tag, while another one can monitor its emission (eavesdrop at a longer read range).
  - *Reader-to-Tag Eavesdropping Range.* Readers transmit at much higher power than tags. Reader's transmissions can be eavesdropped from much further.

  One approach is to cover a tag with shielding against scanning. Another approach is to program a tag to reject a stronger signal than prescribed. Distance bounding protocols [35, 73, 78] determine whether a reader querying a tag is inside nominal range or not. These protocols make decision based on the time that a message is sent and when its response is received.

- **Stealthy Scanning.** Tags are not only inconspicuous, a tag holder often does not even know when they are transmitting information or to whom. Therefore an entity authentication mechanism should be incorporated which ensures that only authorized readers can read a tag.

This thesis considers lightweight solutions to provide security and preserve privacy in RFID deployments. These lightweight solutions involve mutual authentication protocols, anti-counterfeit mechanisms and an ownership transfer scheme. We also carry out the formal analysis of different protocols suggested in this thesis as described in next section.

2.6 Formal Analysis

We formally analyze the protocols and schemes proposed in our thesis using Casper and FDR tools. First we describe a security protocol in simple and abstract language understandable by Casper [90]. Casper is a program that will compile this description and produce a *communicating sequential processes* (CSP) description of the same protocol [55]. The CSP description is then checked using another program known as *failures-divergence refinement* (FDR) [45]. FDR uses the assumptions of the Dolev-Yao model [30] to find attacks upon protocols, or to show that no such attack exists. The Dolev-Yao model assumes that the intruder may overhear or intercept messages, decrypt and encrypt messages with keys that he knows, and fake messages, but not perform any cryptological attacks.

2.7 Summary

In this chapter, we have discussed RFID technology in detail. The various components which form an RFID system and their roles are explained. Regulations and standardizations of RFID systems are discussed. Various risks related to using this technology are presented. We also carry out formal analysis of the suggested protocols in our thesis using Casper and FDR tools.

Part II

Ultra-Lightweight Mutual Authentication Protocols (UMAPs) : Weaknesses and Countermeasures

Chapter 3

Weaknesses in Existing UMAPs

This chapter reviews how design flaws can be exploited in existing proposals for a family of mutual authentication protocols belonging to the ultra-lightweight class, which are designed for low-cost RFID systems. Section 3.1 introduces this family and security analysis of two authentication protocols, SIDRFID and DIDRFID belonging to the same family. Section 3.2 defines both these protocols in detail. These protocols are considered to employ ultra-lightweight functions and are very efficient. However, Section 3.3 demonstrate design flaws in SIDRFID while Section 3.4 analyzes DIDRFID resulting in full secret disclosure and other attacks in both protocols. These disclosure attacks undermine the security of both protocols. Further analysis highlights additional attacks including traceability and reader impersonation.

3.1 Introduction

As discussed in Section 2.1, RFID systems consist of three main components: *tag*, *reader* and *server*. The communication channel between server and reader is assumed to be secured while the channel between reader and tag is wireless and can be eavesdropped. The wide deployment of RFID systems is being constrained due to many security and privacy issues, as shown in Section 2.5, concerning the eavesdropping of the channel between reader and tag.

To secure the communication on this channel researchers have proposed various cryptographic solutions, including mutual authentication protocols between the two communicating parties. Based on the computational cost and operations supported by the tags, these authentication protocols are divided into four classes: *full-fledged*, *simple*, *lightweight* and *ultra-lightweight*, as discussed earlier in Section 1.2. In the

ultra-lightweight class, UMAPs are proposed for the low-cost RFID systems that are most widely deployed [22] and most likely to replace bar-codes. The main limiting factor in these tags is the strict resource constraints. Since cost has to be kept low, these tags cannot afford a state-of-the-art CPU, large memory, or transmit large data. Generally, low-cost RFID tags consist of a few thousand gates, a simple ALU performing simple operations, and no active power source, as explained in Section 2.1. Therefore UMAPs proposed for these low-cost RFID tags should consist of extremely lightweight and efficient functions. However, it is easy to propose a weak UMAP if not carefully designed. We analyze UMAP proposals to illustrate what can go wrong while suggesting such ultra-lightweight schemes.

Yung-Cheng Lee proposed two such ultra-lightweight authentication protocols [80]. In one of the protocols, the tag and reader do not share any secrets and use their respective identities as shared secrets. These identities are, therefore, not transmitted in the clear. Moreover, these identities do not update and are static. This protocol is called *ultra-lightweight RFID protocol with static identity* (SIDRFID). In the other protocol, tag and reader share a secret key K. After authenticating the reader, the tag sends its unique secret identity IDT. Both K and IDT are updated in each authentication round, hence this protocol is called *ultra-lightweight RFID protocol with dynamic identity* (DIDRFID). Both protocols claim to provide mutual authentication and implement very efficient and extremely lightweight functions. We discuss these protocols in greater depth in Section 3.2.

3.1.1 Our Contribution

We focus our work on highlighting weaknesses in existing proposals belonging to the ultra-lightweight class. These proposals outline mutual authentication protocols to provide security and privacy properties to low-cost RFID systems. We carry out multiple attacks on a couple of UMAPs proposed in [80]. Avoine et al. [3] have also carried out a security analysis of both protocols. They observe that using a single master key in SIDRFID is a single point of failure if compromised. However, they do not elaborate on any specific technique to recover the master key. Our work shows how to recover this single master key and break the entire SIDRFID system. This part of the work as shown in Section 3.3.2 is a joint work. Further, Avoine et al. [3] highlight an attack on the secret key used in DIDRFID. This attack involves eavesdropping two rounds of authentication session and L^2 possible guesses (where L is the length of key). Our work demonstrates a passive full disclosure attack that determines the correct key after eavesdropping approximately $\sqrt{\pi L}$ rounds. This work has been accepted to get published [12].

Table 3.1: Notation used in Chapter 3

Notation	Description
IDT	Tag's static identity.
$DIDT_i$	Tag's dynamic identity used in i^{th} authentication round.
IDR	Reader's static identity.
K_i	Secret key shared between tag and reader in i^{th} authentication round.
R_i	Random number generated by reader in i^{th} authentication round.
$\oplus$	Bitwise XOR operation.
$\vee$	Bitwise OR operation.
$\wedge$	Bitwise AND operation.
$A \rightarrow B : M$	A sends to B, message M.
X	A 96-bit string $x_{95} \cdots x_0$, where x_0 and x_{95} are the least significant and most significant bits respectively.
$HW(X)$	Hamming weight of bit string X.
$Rot(X, Y)$	Left rotation of argument X by HW(Y) bits.

3.2 Two Ultra-lightweight Protocols

In this section, the two protocols proposed in [80] are presented. These protocols belong to the ultra-lightweight class designed for low-cost RFID tags and claim to provide mutual authentication. Additionally, these protocols claim to resist attacks including traceability, replay, de-synchronization and impersonation. Importantly, the computation cost is kept low by incorporating lightweight functions. In the proposed protocols, the pseudo-random number generator is only installed in the reader (which is a resourceful entity when compared to the tag). The low-cost tag only performs simple bit-wise operations (XOR, AND, OR) and left rotation of bits $Rot(A, B)$. The notation used in this chapter are given in Table 3.1.

3.2.1 Protocol with Static Identity (SIDRFID)

This protocol assumes that tag and reader each have identities IDT and IDR, respectively. Both these identities are secret values shared by both entities (it is assumed that tag and reader have these pre-installed prior to activation of the scheme). The i^{th}

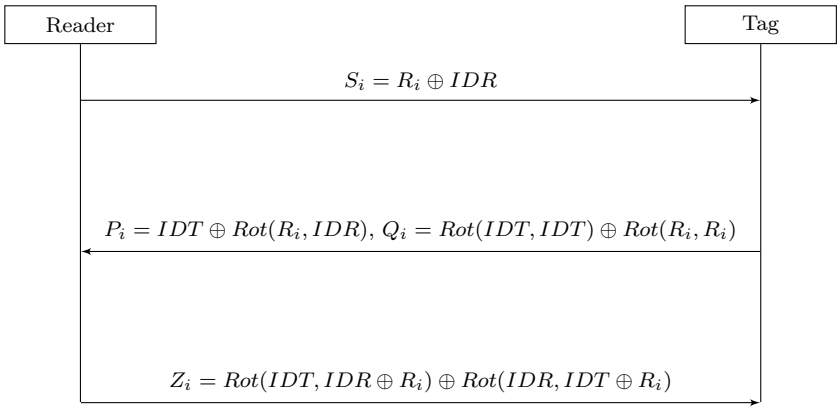

Figure 3.1: Protocol with Static Identity SIDRFID.

round of authentication is shown in Figure 3.1 and consists of the following steps:

- **Step 1.**
  - Reader generates R_i.
  - Reader computes:
    $$S_i = R_i \oplus IDR.$$
  - $Reader \rightarrow Tag : S_i$.

- **Step 2.**
  - Tag computes:
    $$R_i = S_i \oplus IDR,$$
    $$P_i = IDT \oplus Rot(R_i, IDR),$$
    $$Q_i = Rot(IDT, IDT) \oplus Rot(R_i, R_i).$$
  - $Tag \rightarrow Reader : (P_i, Q_i)$.

- **Step 3.**
  - Reader computes:
    $$IDT = P_i \oplus Rot(R_i, IDR),$$
    $$Q_i^{'} = Rot(IDT, IDT) \oplus Rot(R_i, R_i).$$
  - Reader authenticates tag as follows:

if $Q_i^{'} = Q_i$ **then**
Tag is authenticated,
else
Protocol is abandoned.
end if

- **Step 4.**
  - In case of successful tag authentication, the reader computes:

$$Z_i = Rot(IDT, IDR \oplus R_i) \oplus Rot(IDR, IDT \oplus R_i).$$

  - $Reader \rightarrow Tag : Z_i$.

- **Step 5.**
  - Tag computes:

$$Z_i^{'} = Rot(IDT, IDR \oplus R_i) \oplus Rot(IDR, IDT \oplus R_i).$$

  - Tag authenticates reader as follows:

if $Z_i^{'} = Z_i$ **then**
Reader is authenticated,
else
Protocol is abandoned.
end if

3.2.2 Protocol with Dynamic Identity (DIDRFID)

This protocol assumes that tag and reader share a secret key K (it is assumed that tag and reader have this pre-installed prior to activation of the scheme). The i^{th} round of authentication is as shown in Figure 3.2 and consists of the following steps:

- **Step 1.**
  - $Tag \rightarrow Reader : DIDT_i$.

- **Step 2.**

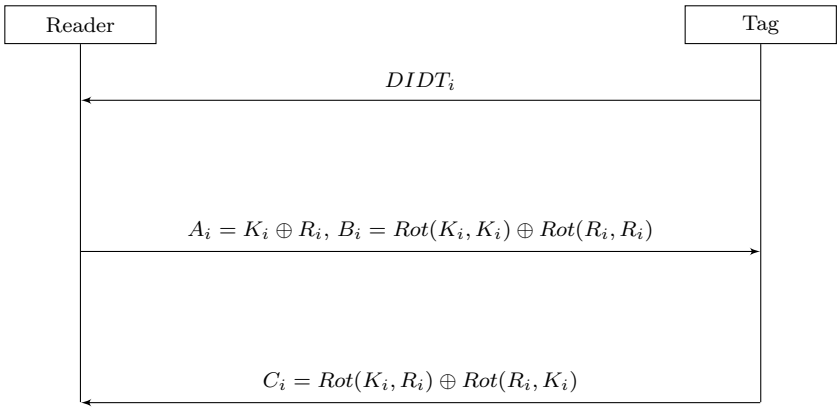

Figure 3.2: Protocol with Dynamic Identity DIDRFID.

- Reader uses $DIDT_i$ as index to extract the corresponding secret key K_i from the database.
- Reader generates a random number R_i.
- Reader computes:

$$A_i = K_i \oplus R_i,$$
$$B_i = Rot(K_i, K_i) \oplus Rot(R_i, R_i).$$

- $Reader \rightarrow Tag : (A_i, B_i)$.

- **Step 3.**
  - Tag computes:

$$R_i = A_i \oplus K_i,$$
$$B_i^{'} = Rot(K_i, K_i) \oplus Rot(R_i, R_i).$$

  - Tag authenticates reader as follows:

    if $B_i^{'} = B_i$ **then**
    Reader is authenticated,
    else
    Protocol is abandoned.
    end if

- **Step 4.**

– In case of successful reader authentication, the tag computes:

$$C_i = Rot(K_i, R_i) \oplus Rot(R_i, K_i).$$

– $Tag \rightarrow Reader : C_i$.

- **Step 5.**

  – Reader computes:

$$C_i^{'} = Rot(K_i, R_i) \oplus Rot(R_i, K_i).$$

  – Reader authenticates tag as follows:

  if $C_i^{'} = C_i$ **then**
  Tag is authenticated,
  else
  Protocol is abandoned.
  end if

- **Key Updating Step.** After successful mutual authentication, tag and reader update their values:

  – Tag and Reader compute:

$$DIDT_{i+1} = Rot(R_i, R_i \vee K_i) \oplus Rot(K_i, R_i \wedge K_i),$$
$$K_{i+1} = Rot(R_i, R_i \wedge K_i) \oplus Rot(K_i, R_i \vee K_i).$$

  – Tag and Reader both keep $(DIDT_i, K_i)$ and $(DIDT_{i+1}, K_{i+1})$ in their respective memory locations.

3.3 Security Analysis of SIDRFID

In this section, we carry out a security analysis of SIDRFID as presented in Section 3.2.1. Avoine et al. [3] have suggested that SIDRFID is a weak protocol because it uses a single master key which in many situations is considered unacceptable. However, there may be applications, such as issuing temporary RFID tags for access control to a team visiting an organization, where use of a single master key may be justified. In such scenarios, we do not need to generate new keys on every access attempt and thus

avoid the need for secure distribution of these secret keys to each tag. Nonetheless we show that, even in situations where a fixed master key is justified, the secret entities can easily be recovered thus demonstrating that SIDRFID is a very weak protocol. A formal analysis of this protocol is presented in Appendix B.

3.3.1 Passive Hamming Weight Disclosure (PHWD) Attack

We first present a passive attack which reveals $HW(IDR)$. We make the realistic assumption that the channel between tag and reader is wireless and can be eavesdropped. The attacker simply needs to eavesdrop any two rounds of authentication. Moreover, the resources available to the attacker are also limited so it cannot perform complex computations (a realistic assumption in lightweight cryptography). The attack executes as follows:

- **Step 1.** Attacker eavesdrops two legitimate authentication rounds to obtain S_1, P_1 and S_2, P_2.

- **Step 2.** The attacker computes:

$$\begin{aligned} A &= S_1 \oplus S_2, \\ &= (R_1 \oplus IDR) \oplus (R_2 \oplus IDR), \\ &= R_1 \oplus R_2. \end{aligned} \tag{3.1}$$

$$\begin{aligned} B &= P_1 \oplus P_2, \\ &= (IDT \oplus Rot(R_1, IDR)) \oplus (IDT \oplus Rot(R_2, IDR)), \\ &= Rot(R_1, IDR) \oplus Rot(R_2, IDR), \\ &= Rot(R_1 \oplus R_2, IDR). \end{aligned} \tag{3.2}$$

From (3.1) and (3.2), we get:

$$B = Rot(A, IDR). \tag{3.3}$$

Since A and B are known from (3.1) and (3.2), $HW(IDR)$ can easily be obtained from (3.3).

After disclosing $HW(IDR)$, an attacker can carry out a selective brute force attack to find the exact value, where each value has correctness probability (considering L as the length of bit string IDR):

$$prob = \frac{1}{\binom{L}{HW(IDR)}}.$$

This value is much higher than 2^{-L}, which is the probability of brute force attack success against an L-bit value. If we assume that IDR is similar to those assigned as EPC values (96-bit [48]), IDR consists of only 36 random bits (which we denote IDR^*) and the remaining 60 bits are publicly known (these determine the header, manufacturer and type of item details). This further raises the correctness probability $prob'$ of a guess to:

$$prob' = \frac{1}{\binom{36}{HW(IDR^*)}},$$

which is substantially fewer trials to conduct.

3.3.2 Full Disclosure Active (FDA) Attack

We now present a full disclosure active attack against $SIDRFID$. We assume that either the attacker is in possession of the tag or there is no restriction on accessing the tag. This attack involves eavesdropping one round of legitimate communication and 95 chosen public messages sent to the tag (considering the length of variables to be 96 bits as in the EPCC1G2 standard [48]).

The FDA attack is explained as follows:

- **Step 1.** The attacker eavesdrops a legitimate authentication round and records S_1, P_1, Q_1 and Z_1 (described in Section 3.2.1), where the labels of individuals bits in each of these strings is as for string X in Table 3.1.

- **Step 2.** The attacker impersonates a legitimate reader and sends S_2, which is a manipulated version of S_1 with the two least significant bits flipped as $s_0^{'}$ and $s_1^{'}$ (the subscript of S represents the round number and subscript of s represents the bit position).

- **Step 3.** Tag computes R_2 as follows:

$$R_2 = S_2 \oplus IDR. \tag{3.4}$$

  Since IDR is fixed, R_2 is the same as R_1 except that the least significant two bits are flipped as $r_0^{'}$ and $r_1^{'}$ as follows:

$$\begin{aligned} R_1 &= r_{95}r_{94}r_{93}\cdots r_2 r_1 r_0, \\ R_2 &= r_{95}r_{94}r_{93}\cdots r_2 r_1^{'} r_0^{'}, \\ M &= R_1 \oplus R_2, \\ &= 00\cdots 011. \end{aligned} \tag{3.5}$$

The tag now computes P_2 and Q_2 where:

$$\begin{aligned} P_2 &= IDT \oplus Rot(R_2, IDR), \\ Q_2 &= Rot(IDT, IDT) \oplus Rot(R_2, R_2). \end{aligned}$$

and sends them to the attacker.

- **Step 4.** After receiving P_2 and Q_2, the attacker computes:

$$\begin{aligned} N &= P_1 \oplus P_2, \\ &= (IDT \oplus Rot(R_1, IDR)) \oplus (IDT \oplus Rot(R_2, IDR)), \\ &= Rot(R_1, IDR) \oplus Rot(R_2, IDR), \\ &= Rot(R_1 \oplus R_2, IDR), \\ &= Rot(M, IDR). \end{aligned} \tag{3.6}$$

Since N and M are known in (3.6), HW(IDR) can be calculated.

- **Step 5.** The attacker now computes:

$$\begin{aligned} T &= Q_1 \oplus Q_2, \\ &= (Rot(IDT, IDT) \oplus Rot(R_1, R_1)) \oplus (Rot(IDT, IDT) \oplus Rot(R_2, R_2)), \\ &= Rot(R_1, R_1) \oplus Rot(R_2, R_2). \end{aligned} \tag{3.7}$$

- **Step 6.** R_2 is the same as R_1 except that the least two bits are flipped as $r_0^{'}$ and $r_1^{'}$, as discussed before when deriving (3.5). The two least significant bits of R_1 will either be the same or different with probability one half. The attacker thus analyzes (3.7) according to two conditions as follows:
  - **Case 1.** The two flipped bits of R_1 are different, which results in:

$$HW(R_1) = HW(R_2).$$

This simplifies (3.7) as follows:

$$\begin{aligned} W &= Rot(R_1 \oplus R_2, R_1), \\ &= Rot(M, R_1). \end{aligned} \tag{3.8}$$

Since M is a string of all 0's except for two consecutive 1's in the least significant positions (as described for (3.5)), W will also consist of all 0's except for two 1's at two consecutive positions in the string. The position of the first 1 starting with the least significant bit as zero determines HW(R_1). The attacker marks the least significant bit of R_1 as x and the next bit as $x^{'}$ (in this case the first two LSBs are inverses of each other).

– **Case 2.** The two flipped bits of R_1 are the same which results in either:

$$HW(R_1) = HW(R_2) + 2,$$

or

$$HW(R_1) = HW(R_2) - 2.$$

Since HW(R_1) $\neq$ HW(R_2), this does not simplify (3.7). In this case the string T will be a random string of 0's and 1's without any pattern. The attacker marks the least significant bit of R_1 as x and the next bit as x, since both bits are either 0 or 1.

- **Step 7.** The attacker continues sending the next chosen plaintext S_3 by flipping (s_0, s_2). The resultant string T in this case will reveal whether r_2 is the same as r_0.

  if $r_2 = r_0$ **then**
  $r_2 = x$,
  else
  $r_2 = x^{'}$.
  end if

  In general, the attacker continues sending chosen plaintexts by flipping two bits (s_0, s_k) where $k = 1 \cdots 95$ as shown in Figure 3.3. For the k^{th} round of authentication, the string T in (3.7) reveals two bits of R_1, (r_0, r_k), to be either the same

or otherwise.

- **Step 8.** At the end of this attack, R_1 is represented as a string of x and $x^{'}$ with known HW(R_1) from (3.8). The attacker now replaces x's with 1's and $x^{'}$'s with 0's, or vice versa according to HW(R_1).

- **Step 9.** The only non-trivial value will be when HW(R_1) = 48. In this case, x can either be a 1 or a 0, thus R_1 has two possible values. In this case, the attacker uses the eavesdropped legitimate round of Step 1 and checks which of the two possible values of R_1 satisfies the values of the public messages S_1, P_1 and Q_1.

- **Step 10.** Once we get the value of R_1, we can easily determine IDR and IDT from any of the public messages. It now becomes very easy to launch multiple attacks on a tag including tag cloning, tag tracking and inventorying [63].

3.3.3 Other Attacks

We have just shown a full disclosure attack which completely disrupts the authentication process in SIDRFID. We now highlight further weaknesses in the design of this protocol which can be exploited to launch multiple attacks.

- **Traceability Attack.** We assume that a low-cost RFID tag is unable to keep track of the current status in an authentication round. It thus replies to every query sent by a compatible reader. In SIDRFID, the public messages P and Q are different in every authentication round because of the different random R generated by the reader. The attacker thus eavesdrops one round of authentication and keeps on sending the same S, thus forcing the tag to calculate similar public messages. This will facilitate tracking of a particular tag.

- **Reader Impersonation.** The order of authentication is important in SIDRFID protocol and can counter active attacks. The reader should be authenticated first so the tag may transmit its secret information only to a legitimate reader. The wrong order of authentication leads to a reader impersonation attack. An attacker can eavesdrop a legitimate authentication round. The attacker can then impersonate a legitimate reader and replay the eavesdropped response as legitimate and get itself authenticated. This attack is possible because secret values are not updated in each fresh round of authentication.

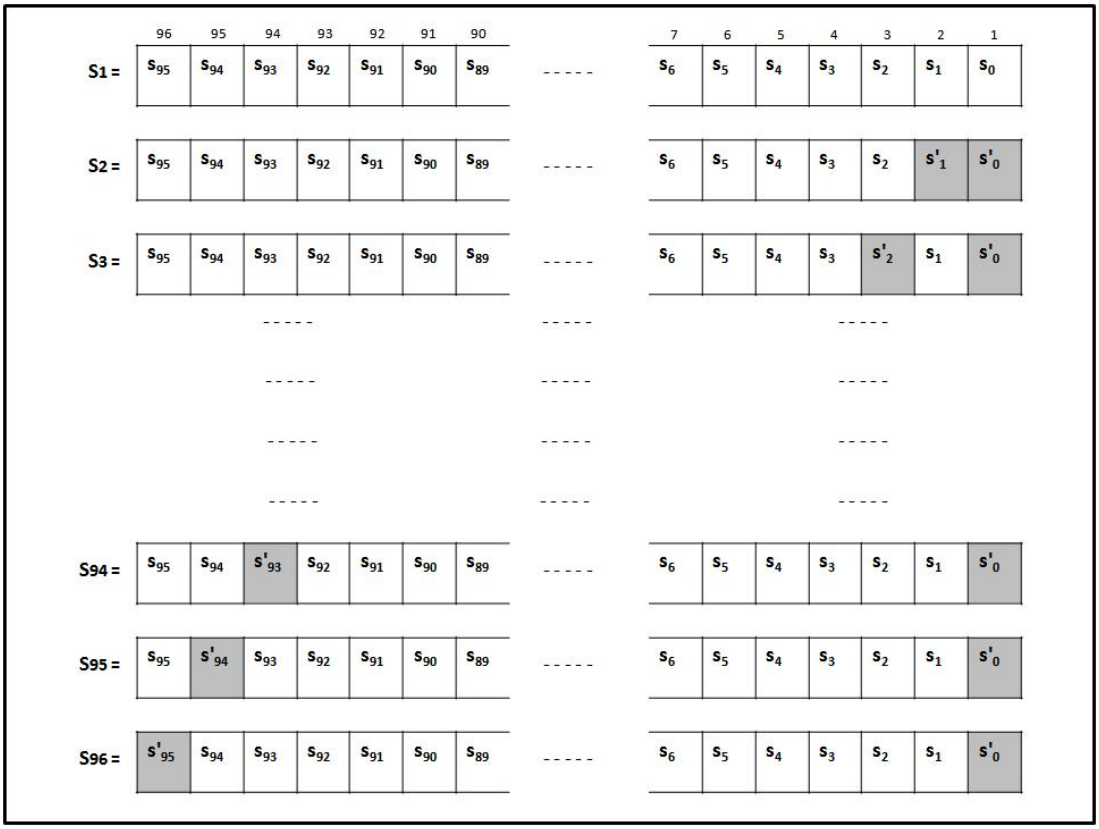

Figure 3.3: Full Disclosure Attack.

- **Identification of Reader.** SIDRFID does not specify how the tag determines which IDR is to be used to generate the public values. Therefore, a further limitation of this protocol is that it can only be implemented in scenarios where there is only one particular reader (or many readers with the same IDR value).

3.4 Security Analysis of DIDRFID

In this section, we carry out a security analysis of DIDRFID as given in Section 3.2.2. Avoine et al. [3] presented a key guessing attack against DIDRFID. This attack requires eavesdropping two authentication session and a total of L^2 possible guesses, where L is the length of the secret key. Whilst this is a serious attack, we present another variant of full disclosure attack which uniquely determines the key. This further demonstrates that DIDRFID is a very weak protocol. A formal analysis of this protocol is presented in Appendix B.

3.4.1 Passive Weight Disclosure (PWD) Attack

We assume that the channel between the tag and reader is wireless and can be eavesdropped. This attack first obtains $HW(K)$ which we will then show allows us to uniquely determine the correct secret K.

The details of this protocol are given in Section 3.2.2 and our attack, which extracts the secret key K, is as follows:

- **Step 1.** Attacker scans the communication channel until he observes that the message B_i in (3.9) sent by reader to tag (forward channel) is equal to the message C_i in (3.10) sent by tag to reader (backward channel):

$$B_i = Rot(K_i, K_i) \oplus Rot(R_i, R_i), \tag{3.9}$$

$$C_i = Rot(K_i, R_i) \oplus Rot(R_i, K_i). \tag{3.10}$$

  It is evident from (3.9) and (3.10) that $B_i = C_i$ when:

$$HW(K_i) = HW(R_i). \tag{3.11}$$

- **Step 2.** The probability *prob* of meeting the condition in (3.11) for two random L bits values is as follows:

$$prob = \sum_{i=0}^{L} \frac{\binom{L}{i}^2}{(2^L)^2}. \tag{3.12}$$

- **Step 3.** Once the condition in (3.11) is satisfied, the attacker re-writes (3.9) and (3.10) as follows:

$$B_i = Rot(K_i \oplus R_i, K_i), \tag{3.13}$$

$$C_i = Rot(K_i \oplus R_i, K_i). \tag{3.14}$$

- **Step 4.** Since message A is:

$$A_i = K_i \oplus R_i, \tag{3.15}$$

  as described in Section 3.2.2, (3.13) and (3.14) can be written as:

$$B_i = C_i = Rot(A_i, K_i). \tag{3.16}$$

Since A_i, B_i and C_i are known, HW(K_i) can be computed from (3.16) and thus HW(R_i) from (3.11).

- **Step 5.** Since message A_i, HW(K_i) and HW(R_i) are known, the attacker uses (3.15) to infer the following information:

$$HW(A_i) = HW(R_i) + HW(K_i) - 2j, \tag{3.17}$$

where j determines the number of 1's in K_i overlapping with R_i at the same bit positions.

- **Step 6.** The attacker determines j using (3.17) to infer the following information:

$$HW(R_i \vee K_i) = HW(A_i) + j, \tag{3.18}$$

$$HW(R_i \wedge K_i) = j. \tag{3.19}$$

- **Step 7.** We now XOR the update equations as given in Section 3.2.2 as follows:

$$\begin{aligned} DIDT_{i+1} \oplus K_{i+1} &= Rot(R_i \oplus K_i, R_i \vee K_i) \oplus Rot(R_i \oplus K_i, R_i \wedge K_i), \\ &= Rot(A_i, R_i \wedge K_i) \oplus Rot(A_i, R_i \vee K_i). \end{aligned} \tag{3.20}$$

Since DID_{i+1} and A_i are public values and we use (3.18) and (3.19) to deduce the correct K_{i+1}.

3.4.2 Comparison between Our Attack and Avoine's Attack

The complexity of revealing the secret K for both attacks depends on the number of bits of K. The number of operations in Avoine's attack corresponds to the number of guesses before revealing the correct K. Avoine's attack thus requires a total of L^2 guesses using (3.20) and eavesdropping a second round of a DIDRFID authentication session for testing.

Our attack requires a small number of rounds to be eavesdropped, but once this is done there is no further "guesswork" required since the key K is then revealed. We now show that the number of rounds required can be approximated as $\sqrt{\pi L}$. This number corresponds to $\frac{1}{prob}$, which from (3.12) is given by:

$$rounds = \sum_{i=0}^{L} \frac{(2^L)^2}{\binom{L}{i}^2}. \tag{3.21}$$

Putting $m = n = p = L$ in Vandermondes convolution formula (also called the ChuVandermonde formula, see [50, 132]) we see that:

$$\sum_{i=0}^{L} \binom{L}{i}^2 \approx \binom{2L}{L}. \tag{3.22}$$

From Stirling's approximation [38]:

$$\binom{2L}{L} \approx \frac{4^L}{\sqrt{\pi L}}. \tag{3.23}$$

Hence it follows that:

$$rounds \approx \sqrt{\pi L}. \tag{3.24}$$

We note that for the case of an EPCglobal tag, $L = 96$ and hence $rounds = 17$. Since eavesdropping the tag-reader channel is easy, our attack can be very effective in dense reader environments where tags can be read multiple times. In other cases an ongoing authentication round can be interrupted and repeated until $B_i = C_i$. The relationship between these two attacks is summarized in Table 3.2.

3.4.3 Traceability Attack

We note an additional weakness of DIDRFID. If the final message C_i sent by the tag does not reach the reader due to a transmission error, or the attacker disrupts it, the reader does not recognize the updated value $DIDT_{i+1}$. The reader in this case asks for older values of $DIDT_i$ (this is not mentioned in [80]). In such a scenario, the attacker can track the tag by eavesdropping $DIDT_i, A_i, B_i$ and then disrupting message C_i. The attacker can then repeatedly ask for an older value $DIDT_i$ and send A_i, B_i in response, thus tracking the tag.

Table 3.2: Comparison between Our Attack and Avoine Attack.

Type of Attack	**No. of rounds to be eavesdropped**	**No. of guesses before revealing secret key**
Avoine Attack [3]	2	L^2
Our Attack	$\sqrt{\pi L}$ approx.	1

3.5 Summary

In this chapter, we have highlighted design flaws in two existing UMAPs by carrying out security analysis of the two RFID authentication protocols proposed in [80]. Earlier analysis carried out by Avoine et al. [3] on SIDRFID mentions only that the use of a single master key is a potential weakness. We have shown how to recover this single master key, thus allowing this weakness to be fully exploited. Similarly, the attack on DIDRFID presented in [3] can successfully determine the correct key in L^2 attempts (where L is the length of key). We have presented another variant of a full disclosure attack which only requires the attacker to eavesdrop approximately $\sqrt{\pi L}$ rounds but requires no further computation in order to disclose the secret key. We conclude that both SIDRFID and DIDRFID are extremely weak protocols.

Chapter 4

Proposing a new UMAP

In this chapter we propose a new UMAP which builds on the strengths of existing schemes and overcomes their weaknesses. Section 4.1 summarizes the weaknesses in existing schemes and our contribution. Our proposed protocol is explained in detail in Section 4.2. We then carry out a security and performance analysis of our proposed scheme in Section 4.3. An implementation design is also suggested in Section 4.4. Another family of ultra-lightweight class is introduced in Section 4.5.

4.1 Introduction

Low-cost RFID systems are the most widely deployed RFID systems. Mutual authentication protocols belonging to the ultra-lightweight class (UMAPs) are suggested for these systems because tags cost a few cents and have severe resource constraints. As mentioned in Chapter 3, and will be shown in Section 4.2.4, existing proposals for UMAPs have significant drawbacks such as the use of triangular functions [74] only, use of a fresh random nonce for updates in every authentication attempt (whether successful or not), and notable overheads. In Chapter 3, we saw how multiple attacks can be launched against existing UMAPs exploiting their vulnerabilities. This raises the need for proposing new UMAPs, which will not suffer from the same flaws.

4.1.1 Our Contribution

The weaknesses in existing schemes exploit design flaws to carry out secret disclosure and de-synchronization attacks, whereas most countermeasures use additional overheads. We propose a UMAP which overcomes the weaknesses highlighted in these earlier schemes and builds on their strengths to provide mutual authentication be-

tween a tag and a reader (connected with a server). Significantly our suggested scheme also uses fewer resources than other countermeasures proposed for the same class. This work has been published in [9].

4.2 Proposed UMAP

We now explain our proposed UMAP.

4.2.1 Assumptions

We first make the following assumptions that must hold prior to running our protocol.

- Each tag shares secrets (specifically a key and a static identity) with a server.
- The server holds a database which records details about a particular tag, including its shared secrets (key and static identity).
- This database is indexed by a dynamic and publicly known index-pseudonym unique to each tag.
- The reader is an intermediary which relays the messages from the tag (prover) to the server (verifier).
- The reader, querying the tag, is connected to the server and is legitimate (the communication channel between the reader and server is secured).

4.2.2 Adversarial Model

We consider that our scheme is vulnerable to both passive and active attackers. The abilities and limitations of our potential adversary are as follows:

- The adversary is capable of listening to both forward and backward channels (the reader to the tag and vice versa).
- We assume that our adversary has two options: either to jam (active) or to eavesdrop (passive) the radio conversation between server and tag. However we also assume that our adversary cannot function in full duplex mode; i.e., the adversary cannot transmit and receive on the same frequency slot, at the same time.
- The adversary cannot take over an ongoing authentication round because, when the tag detects a collision of readers, it stops responding (we assume the use of a reader anti-collision algorithm, see Section 2.2.2).

Table 4.1: Notation used in Chapter 4

Notation	Description
T	A tag participating in an authentication round.
R	A reader participating in an authentication round.
S	A server holding the database and authenticating a tag.
Adv	Both passive as well as active adversary.
$Index^i$	A dynamic index-pseudonym uniquely associated to each tag in the i^{th} authentication round.
KS^i	A dynamic secret key shared between tag and server in the i^{th} authentication round.
ID	A tag's static and unique identity.
L	Length of the secret key and static identity.
r^i	A random number generated by the server in the i^{th} authentication round.
$+$	Addition modulo 2^L since all values are assumed to be L-bits long.
$A \rightarrow B : M$	A sends to B, message M.
$A\|\|B$	Concatenation of two messages (or values) A and B.
$HW(X)$	Hamming weight of bit string X.
$\lambda(X)$	Integer value of L-bit string X reduced modulo L.
$Rot(X, \mu)$	Left rotation of argument X by μ.
$f(X, Y)$	A secure lightweight pseudo random function (PRF) which takes two inputs X, Y and outputs a pseudo-random value where $f(X, Y) \neq f(Y, X)$ (such as $MixBits$, specified in [111]).

- Defenses against relay attack (man-in-the-middle), physical capture and tampering are not in the scope of this work.

The notation used in our scheme is summarized in Table 4.1.

4.2.3 Goals

A UMAP should achieve the following goals considering the variety of potential threats (as discussed in Section 2.5):

- **Mutual Authentication:** Our scheme should provide mutual (entity) authentication.

- **Tag Content Privacy:** The secret static identity of the tag should not be transmitted in the clear since it is linked to the contents of the item it is attached to.
- **Availability:** Authenticating parties should stay synchronized and always be available to communicate.
- **Tag Anonymity:** The adversary should not be able to track a target tag by listening to the communication channel.
- **Forward Security:** If a tag is compromised at any stage, the adversary should not be able to compromise any future communication.
- **Performance:** Since UMAPs are designed for low-cost RFID systems:
  - storage space should be as low as possible,
  - cryptographic functions should be extremely lightweight in nature and efficient to compute,
  - the amount of data communicated should be kept as low as possible.

4.2.4 Literature Review of UMAPs

UMAPs are designed to provide mutual authentication between a tag and a reader (connected with a back-end server). These schemes are proposed for extremely low-cost RFID tags costing a few pence approximately. Many UMAPs have been proposed, but all existing proposals have significant flaws, as we now outline.

Initial proposals were based only on the use of *triangular functions* (T-Functions) [74] including XOR, AND, OR and addition modulo 2^L (where L is the length of variables in bits). These schemes [107, 108, 110] are considered very efficient and lightweight in their design. However, T-functions have very poor diffusion and use of AND and OR produces biased results. Weaknesses in these proposals were highlighted after publication [5, 6, 86, 87].

$SASI$ [22] is the first UMAP to use a lightweight non-triangular function $RotBits$ (left rotation of bits) with triangular functions. This protocol was initially acclaimed, but later weaknesses were highlighted in its design which resulted in de-synchronization and full-disclosure attacks [16, 17, 134]. A full-disclosure attack on $SASI$ in [17] used the properties of the $RotBits$ function. The main idea behind the attack was that if $RotBits$ does not rotate the values, $SASI$ can be treated as a scheme with T-functions only. As a result, the *Gossamer* UMAP [111] was proposed, which introduced a new non-triangular lightweight function known as $MixBits$. However, the weakness which

caused de-synchronization in *SASI* [16,134] was not addressed in *Gossamer*, resulting in further de-synchronization attacks [13, 135, 154]. This weakness arises since the reader generates new random numbers in each authentication round and both reader and tag use these random numbers to update their values. An adversary can thus use this property to its advantage by de-synchronizing the authentication process [4].

David et al. [24] and Tagra et al. [135] proposed countermeasures to prevent de-synchronization attacks. However, these countermeasures require additional valuable resources at the tag end. Hernandez-Castro et al. [54] presented a passive attack on the scheme in [24] which can recover a tag's secret using linear cryptanalysis. Lee et al. [81] also presented a scheme which requires additional memory and communication overheads and has some privacy issues [112]. Yeh et al. [154] suggested reducing the storage overheads on the tag's memory, however, it becomes very easy to force de-synchronization [4]. Moreover, a passive adversary can carry out a traceability and a full-disclosure attack [109] on the Yeh et al. protocol. Similarly a protocol suggested by Eghdamian et al. [32] is cryptanalyzed by Avoine in [3]. In most UMAPs, the main vulnerability exploited by an adversary is the stateless nature of a tag. The attacker runs many incomplete protocols and gathers information from each in order to disclose secret values.

Most of the existing proposals for UMAPs thus have flaws. The existing countermeasures to overcome these flaws given in [72, 135] have notable overheads. In this work, we propose a new UMAP which overcomes the flaws and uses strengths of existing protocols.

4.2.5 Design Features

Our protocol has the following design features, intended to overcome flaws as outlined in Section 4.2.4:

- **Combination of Functions:** The protocol uses a combination of lightweight non-triangular functions and triangular functions. We employ modular addition, which is not biased like OR and AND functions. We use $Rot(X, \lambda(Y))$ as left rotation of bit string X by $\lambda(Y)$ positions, where $\lambda(Y)$ is computed by first converting the L-bit string Y into an integer and then reducing it modulo L. Some of the existing schemes have used $Rot(X, HW(Y))$ as left rotation of bit string X by the hamming weight of bit string Y. Since HW(Y) does not follow a uniform distribution, this weakens the security property given by the rotation function, whereas $\lambda(Y)$ follows a uniform distribution. We also use $MixBits$ function [111], which is defined as follows:

$$
\begin{aligned}
Z &= MixBits(X, Y) \\
Z &= X; \\
for(i &= 0; i < L; i++)\{ \\
Z &= (Z << 1) + ((Z + Y) >> 1);\}
\end{aligned}
$$

Inputs (X and Y) and output (Z) are L-bits of length and the function has a loop of L iterations. Addition is carried out modulo 2^L, $<<$ denotes bitwise left shift and $>>$ denotes bitwise right shift. This function is also considered to be efficient as it consists of modular additions and rotation functions. It also provides good security by resisting common attacks as shown in [114].

- **Use of Random Nonce:** In our protocol, the server generates a random nonce for data freshness. In existing schemes random nonces change on every communication attempt, even in the event of a failed authentication. Avoine et al. [4] mention this as a potential vulnerability which can lead to de-synchronization attacks. Our scheme overcomes this vulnerability by recording each random nonce in a database for a particular tag's $Index$. It then uses the random value to calculate internal secret values for updating the tuple (KS, $Index$). The server generates a new random nonce only after a successful authentication. This resists de-synchronization attacks and provides tag anonymity and forward security.

- **Provision for Re-synchronization:** In the event of a failed authentication attempt, either due to communication error or intentional interference by an adversary, both server and tag may become de-synchronized. Our scheme re-synchronizes, as the tag does not update its values in a failed authentication attempt and the server keeps a copy of older values. In some existing schemes [22, 24,32,72,81,111,135], re-synchronization is attempted using older values of $Index$ and shared secrets stored in the tag's memory. This not only places additional overheads on the tag's valuable memory but also leads to a potential weakness which allows the server to ask for older values of $Index$ of the tag if the updated $Index$ is not recognized. This weakness leads to denial-of-service attacks as mentioned in [4, 13]. In our scheme, if the server asks for older values, this will be an indication of a replay attack carried out by an impersonating server.

- **Cost, Performance and Security Trade-offs:** Our scheme provides a trade-off between cost, performance and level of security. It uses lightweight functions which can easily be incorporated in the simple ALU of low-cost RFID tags. Our protocol consumes a small amount of storage on these tags and completes the

protocol using two messages. The schemes mentioned in [24, 32, 72, 81, 135, 155] require additional messages and memory requirements in order to overcome existing weaknesses. Moreover, many of these schemes are still vulnerable and have been analyzed to highlight weaknesses in the design [3, 4, 54, 109, 112].

4.2.6 The Protocol

We now propose a new UMAP that provides the security goals mentioned in Section 4.2.3 and has the design features identified in Section 4.2.5. The i^{th} round of the authentication protocol is described in the subsequent sections.

Identification Stage

A compatible T in the vicinity of a compatible R is identified as follows:

- **Step 1.** $R \rightarrow T : Hello$.
- **Step 2.** $T \rightarrow R : Index^i$.
- **Step 3.** $R \rightarrow S : Index^i$.
- **Step 4.** S now searches for this $Index^i$ in its database. If it matches an existing entry, S proceeds to the next stage, otherwise it does not respond to T.

Server Authentication and Update Stage

On successful identification, S is authenticated as follows:

- **Step 1.** S uses $Index^i$ sent by T to extract KS^i associated with this particular T.
- **Step 2.** S now generates a random value r^i and calculates the internal secret values n_1^i,n_2^i using tuple (KS^i, r^i) as follows:

$$\begin{aligned} n_1^i &= f(KS^i, r^i), \\ n_2^i &= f(r^i, KS^i). \end{aligned} \tag{4.1}$$

- **Step 3.** S now generates public message A^i using tuple $(n1^i, n2^i, Index^i, KS^i, ID)$ as follows:

$$A^i = Rot(Rot(n2^i + Index^i + KS^i + ID, n1^i) + n1^i, n2^i). \tag{4.2}$$

- **Step 4.** $S \rightarrow R : A^i \| r^i$.
- **Step 5.** $R \rightarrow T : A^i \| r^i$.
- **Step 6.** T calculates internal secrets $n1^i$ and $n2^i$ as in (4.1) and uses these to calculate a local copy $A^{i'}$ of A^i using (4.2).
- **Step 7.** T now checks:
  if $A^{i'} = A^i$ **then**
  S is authenticated; proceed to next stage,
  else
  Protocol is abandoned.
  end if
- **Step 8.** S, after sending $A^i \| r^i$, also updates its tuple $(Index^i, KS^i)$ as follows:

$$\begin{aligned} Index^{i+1} &= Rot(Rot(n1^i + Index^i, n1^i) + n2^i, n2^i), \\ KS^{i+1} &= Rot(Rot(n2^i + KS^i, n1^i) + n1^i, n2^i). \end{aligned} \tag{4.3}$$

- **Step 9.** In addition, S keeps a copy of tuple $(Index^i, KS^i, r^i)$ in its memory.

Tag Authentication and Update Stage

Once S is authenticated, T is now authenticated as follows:

- **Step 1.** T generates the public message B^i using tuple $(n1^i, n2^i, Index^i, KS^i, ID)$ as follows:

$$B^i = Rot(Rot(n1^i + Index^i + KS^i + ID, n2^i) + n2^i, n1^i). \tag{4.4}$$

- **Step 2.** $T \rightarrow R : B^i$.
- **Step 3.** $R \rightarrow S : B^i$.
- **Step 4.** S calculates a local copy $B^{i'}$ of B^i using (4.4).
- **Step 5.** S now checks:
  if $B^{i'} = B^i$ **then**
  T is authenticated,
  else
  Protocol is abandoned.
  end if

- **Step 6.** T after sending B^i updates its values of tuple $(Index^i, KS^i)$ only after authenticating S using (4.3).

The message B^i can only be verified by a legitimate S. Successful mutual authentication concludes the protocol and S grants access to T. Both T and S have updated their values as shown in (4.3). S, after successfully authenticating T, deletes the old values of the tuple $(Index^i, KS^i, r^i)$ in its database to avoid tag impersonation. Our proposed UMAP is summarized in Figure 4.1.

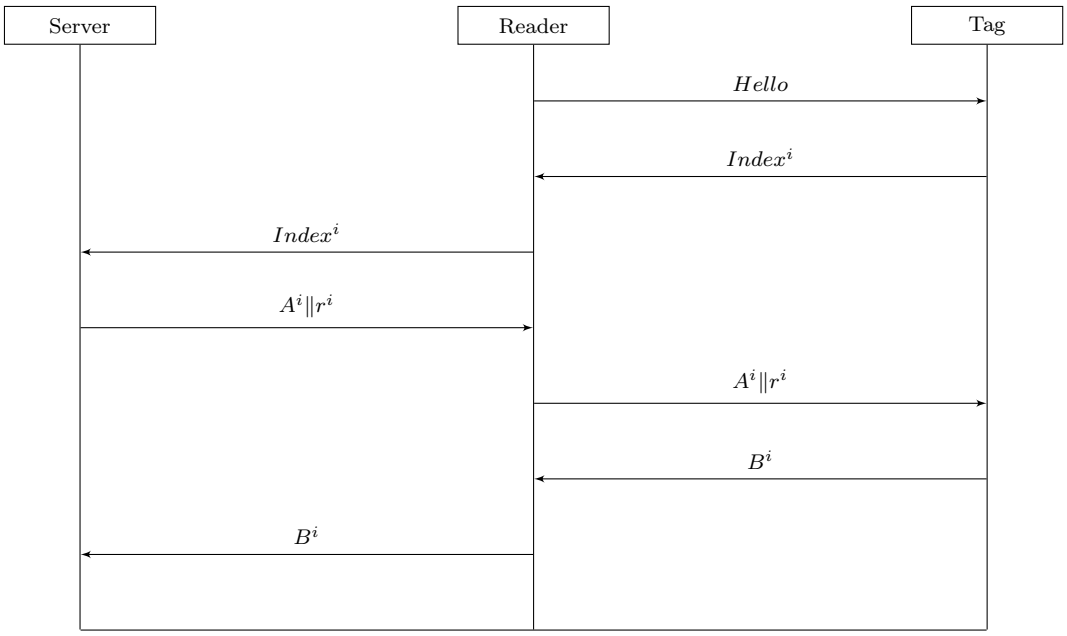

Figure 4.1: Proposed UMAP.

4.3 Security and Performance Analysis

We now conduct a security analysis to show how our UMAP meets the goals of Section 4.2.3, as well as a performance analysis which demonstrates that our scheme uses fewer resources than schemes given in [72, 135]. A formal analysis of our proposed scheme is also presented in Appendix C.

4.3.1 Mutual Authentication

We first show that our scheme provides mutual authentication by demonstrating that only a valid pair of S and T (in possession of KS) can generate public messages A and

B, respectively, that will be accepted by the other party. The freshness of these public messages is ensured by the use of a random nonce in every authentication round.

1. **Authentication of the server:** S is authenticated by checking the authenticity of public message A. This message is generated using shared secrets known only to legitimate authenticating parties. Therefore, only a legitimate T can check the legitimacy of the message. The correctness of public message A thus determines the authenticity of S.
2. **Authentication of the tag:** Once T authenticates S successfully, it transmits its shared secrets in the form of a public message B. S can check the legitimacy and correctness of this message and hence authenticates T.

We consider whether an adversary Adv without shared secrets can generate the public messages. To do so, Adv has to take over the authentication round after disrupting message $A||r$ and replaying it later by impersonating a genuine S, or Adv has to eavesdrop $Index$ and message $A||r$ and then take over the authentication round after disrupting and eavesdropping message B to replay it for T's impersonation. However, this is infeasible due to the following reasons:

- Adv cannot take over an ongoing authentication round (see Section 4.2.2).
- Adv cannot disrupt and eavesdrop at the same time (see Section 4.2.2).
- Adv has to perform a relay attack (see Section 4.2.2).

So, server and tag impersonation attacks are not feasible.

4.3.2 Tag Content Privacy

Each T has a unique static identity ID and is linked to the content of the particular tagged item. We want to transmit this ID confidentially so that an adversary is unable to read, copy or track it. In our scheme, S and T share a secret dynamic KS. Our scheme uses this KS to calculate two internal secret values $n1$ and $n2$ using a secure PRF f. We then use the tuple of ($n1$,$n2$,KS) to generate public messages which are used for transmitting the secret ID confidentially. Recall from Section 4.2.6 that each of the two public messages has the following form:

$$P = Rot(Rot(s2 + p + K + S, s1) + s1, s2), \tag{4.5}$$

where P and p are public values, $s1$, $s2$ are dynamic secret values, K is a shared secret key and S is a static secret (ID of T). The goal of the adversary is to disclose S. The complexity of recovering S is as follows:

- The outer rotation from (4.5) is undone with complexity $O(\log_2 s2)$:

$$\begin{aligned} Q &= Rot^{-1}(P, s2), \\ &= Rot(s2 + p + K + S, s1) + s1. \end{aligned} \tag{4.6}$$

- It requires complexity $O(2^{s1} \times \log_2 s2)$ to subtract all possible values of $s1$ from the right hand side of (4.6):

$$\begin{aligned} R &= Q - s1, \\ &= Rot(s2 + p + K + S, s1). \end{aligned} \tag{4.7}$$

- Further inner rotation is undone from (4.7) from all corresponding $2^{s1} \times \log_2 s2$ values (this doubles the complexity as $O(2 \times 2^{s1} \times \log_2 s2)$):

$$\begin{aligned} T &= Rot^{-1}(R, s1), \\ &= s2 + p + K + S. \end{aligned} \tag{4.8}$$

- We now subtract public value p from (4.8) (this doubles the overall complexity as $O(2 \times 2 \times 2^{s1} \times \log_2 s2) = O(2^2 \times 2^{s1} \times \log_2 s2) \approx O(2^{s1} \times \log_2 s2)$:

$$\begin{aligned} U &= T - p, \\ &= s2 + K + S. \end{aligned} \tag{4.9}$$

- Subtracting the corresponding values of $s2$ from (4.9) requires an overall complexity of $O(2^{s1} \times \log_2 s2 \times \frac{2^{s2}}{\log_2 s2}) = O(2^{s1} \times 2^{s2})$:

$$\begin{aligned} V &= U - s2, \\ &= K + S. \end{aligned} \tag{4.10}$$

Concluding, we have a total of 2^{3K} (considering $s1$, $s2$ and K are of the same length) possible values of S. Therefore only a brute force attack is the best available option to guess the shared secret key, which requires 2^K guesses. Since $s1$, $s2$ and K change in every authentication round (and $s1$, $s2$ are output by a secure PRF), our protocol provides privacy of the tag's content.

4.3.3 Availability

In our scheme, both S and T update their shared secret KS and $Index$ after every successful authentication round in synchronization with each other. This synchroniza-

tion is based on the receipt and authenticity of public messages A and B. Since update only takes place after a successful authentication, and public messages A and B can only be generated by legitimate parties, we consider the following threats which can break the synchronization:

1. **Adversary disrupts message A||r:** Since T does not receive message $A||r$ sent by S, it will not update its values and keep the tuple $(Index^i, KS^i)$ in its memory. Although S updates to new tuple $(Index^{i+1}, KS^{i+1})$, it still has an entry for the old tuple $(Index^i, KS^i, r^i)$ in its database. In this case, S identifies T with $Index^i$ which is still not updated and both remain synchronized.

2. **Adversary disrupts message B:** Since S does not receive message B, it has both old and new values as $((Index^i, KS^i, r^i), (Index^{i+1}, KS^{i+1}))$ stored in its database. Whereas T, on sending message B, has already updated its tuple to $(Index^{i+1}, KS^{i+1})$. This avoids de-synchronization as T is identified by S using $Index^{i+1}$.

3. **Adversary tampers with A||r or B:** If an adversary tampers with the public messages A or random number r, a genuine T will calculate a different value of $A^{'}$, which indicates that the message has been altered. Similarly, a genuine S can check the integrity of public message B.

4.3.4 Tag Anonymity

Two of the main privacy concerns in RFID systems are tracking and content privacy [63]. In our scheme, the $Index$ and public messages (A,B) change in every authentication round. This avoids tracking the location of a tag.

4.3.5 Forward Security

In our UMAP, S generates a random value to calculate internal secrets using f. These internal secrets are used to update the $Index$ and KS after every successful authentication round. Therefore, if a tag is compromised, it does not reveal any of its past and future communications.

4.3.6 Performance Analysis

We now briefly carry out a comparative analysis of performance parameters compared with the UMAPs given in [72, 135], which are the only existing ones that appear to meet the security goals detailed in Section 4.2.3.

- **Storage Overhead:** S stores the next potential and old values of the tuple $(Index, KS)$. Since S is considered to have less resource constraints, this lifts the burden on T's memory. Moreover, on successful completion of the protocol, S deletes the old entry thus saving storage space. Tag T requires $2L$ bits storage on RAM for tuple $(Index, KS)$ and L bits of ROM to store its ID, which is less compared to other protocols of the same family, as shown in Table 4.2.
- **Computation Overhead:** We have used lightweight functions (modular addition, left rotation and a lightweight PRF) similar to other related protocols. In our scheme, T has to verify one public message and calculate another message using lightweight functions that can be easily implemented in the ALU of T. Therefore it computes two public messages, which is fewer compared to other schemes (which use the same functions, i.e., a lightweight PRF to generate internal secrets and then adding, XOR-ing and left rotating these with other secret and public values) as shown in Table 4.2. Moreover, we have also reduced the call to f to two, compared to three in other protocols, and do not require XOR.
- **Communication Overhead:** Our scheme communicates $2L$ bits during an authentication round (considering each public message to be L bits) which is less than the other schemes in Table 4.2.

Table 4.2: Comparative Analysis of Different Protocols

Protocol	**Storage**	**Computation**	**Communication**
Tagra et al. Protocol [135]	6L	4	4L
SULMA Protocol [72]	6L	4	4L
Our Protocol	2L	2	2L

Chien [22] categorized RFID tags into four classes depending on the resources, cost and application (see Section 1.2). The ultra-lightweight class is considered to be very restricted in its resources. We consider that achieving security goals as mentioned in Section 4.2.3 using fewer resources is important in this class. UMAPs are designed using

a trade-off between cost, performance and level of security. Thus our protocol reduces the cost (in terms of storage) and enhances the performance (in terms of computation and communication) while achieving the desired goals of a UMAP.

4.4 Implementation Design

In this section, we will explain the proposed design architecture for implementing our proposed UMAP in a tag. First of all we need to consider whether to choose a parallel or a serial implementation. In parallel, each operation will be carried out on a block of bits treated as a word. Whereas, in serial, operations are conducted on one bit at a time. Generally considering the low-power restrictions of RFID tags, the internal clock frequency is set to 100 KHz [37]. As shown in Figure 4.2, we calculate the basic requirements of our targeted platform as follows:

- **MixBits : Lightweight PRF.** This function uses efficient triangular (addition mod 2^L) and non-triangular (left and right bitwise shifts). For $L = 32$ bits, it requires a total of 740 *gate equivalents* (GE) including control logic, 128 clock cycles per block (32 bits) and thus a throughput of approximately 25 Kbps as calculated in [114].
- **Modular Addition.** This function will require 11 GE per bit and L clock cycles for implementation as shown in [60].
- **Rot(X,λ(Y)) : Left Rotation.** This requires a maximum of $2L$ clock cycles to determine $\lambda(Y)$ and then rotating the argument X. If we use an LFSR to implement the rotation, it requires 8 GE per bit.
- **Comparator.** We need two L-bit registers each requiring 8 GE per bit and some control logic. This function uses a maxium of L clock cycles.

4.5 Introduction to HB Protocols

Another family of ultra-lightweight protocols is based on human based identification and uses the hardness of *learning parity with noise* (LPN) first proposed by Nicholas J. Hopper and Manuel Blum [56], hence the name HB. The first proposal based on the HB protocol for RFID systems was suggested by Ari Juels and Stephen A. Weis in [68]. The original HB protocol put in RFID settings is shown in Figure 4.3. The HB protocol requires lightweight functions for implementation. Only bitwise AND and XOR operations are required to compute the binary inner product $a \bullet x$. This product

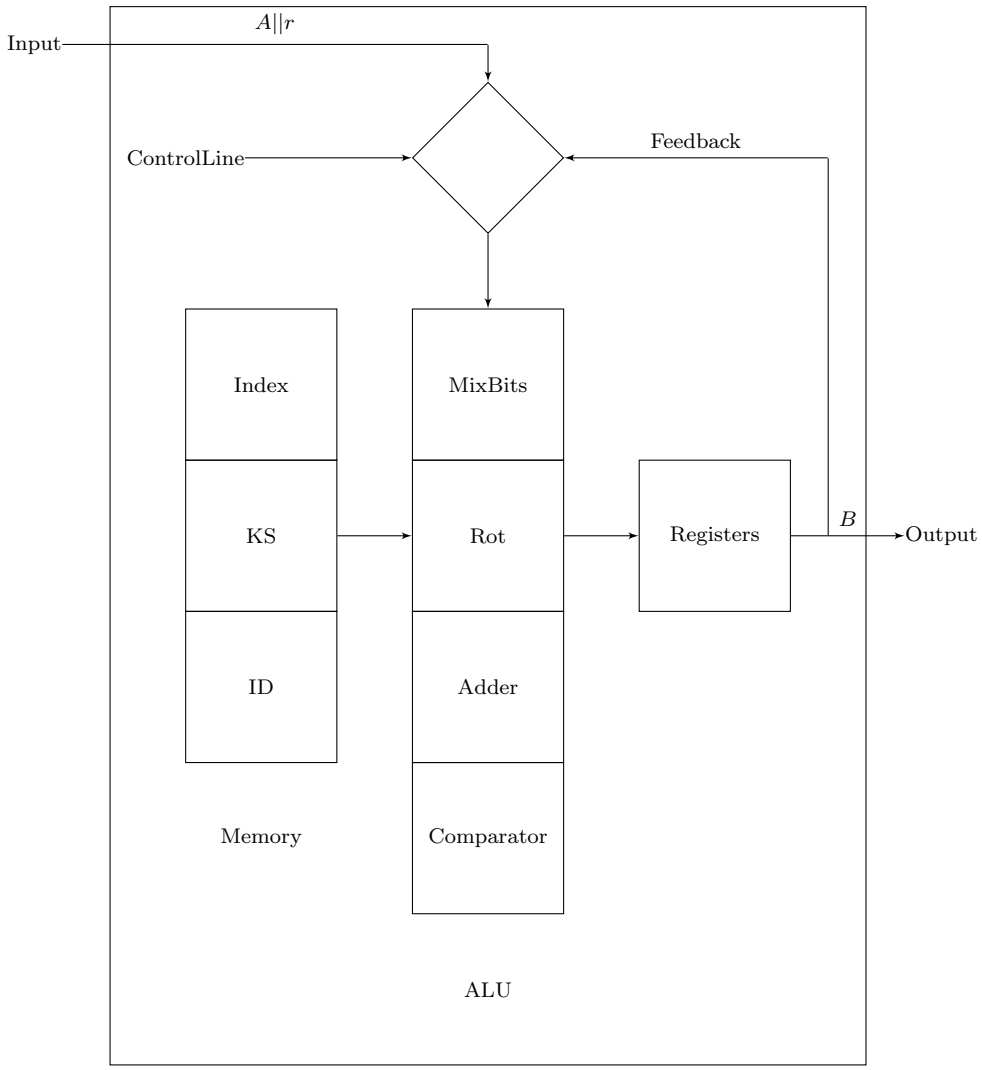

Figure 4.2: Design of Proposed UMAP.

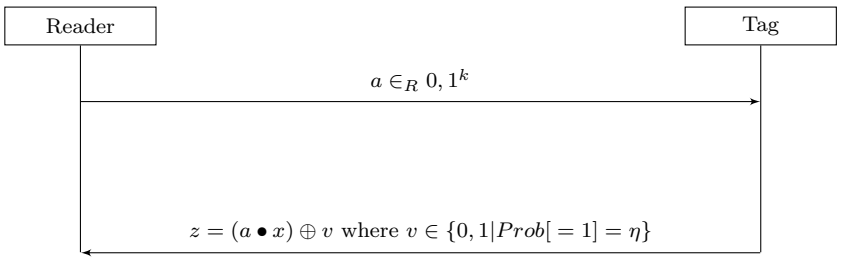

Figure 4.3: HB Protocol setting for RFID.

can be computed on the fly as each bit of a is received and there is no requirement for a buffer. The noise bit v can be generated from physical properties like thermal noise, shot noise, diode breakdown noise, metastability, oscillation jitter, or other methods. Based on these lightweight properties, several proposals have been published based on the HB protocols for RFID systems. Some of the recent developments include [27, 89, 92, 121].

4.6 Summary

In this chapter, we propose a new UMAP designed for use in RFID devices with limited resources. We have shown why our protocol overcomes weaknesses in previous UMAP designs and demonstrated that our protocol involves lower overheads.

Part III

RFID Systems in Supply Chain Management

Chapter 5

Adaptive Online/Offline RFID Scheme

This chapter provides a solution to various requirements related to RFID systems deployed in supply chain management. Section 5.1 provides an overview of an RFID system's application in supply chain management. Section 5.2 comments on existing approaches to preserving privacy of users of RFID tags in supply chain management. Section 5.3 outlines our proposed scheme. Section 5.4 carries out analysis of our proposal.

5.1 Introduction

RFID systems are extensively used in many applications as shown in Section 2.4. In this chapter, we discuss their deployment in supply chain management, where an RFID system is capable of identifying products throughout the supply chain process [79]. As previously explained in Section 2.1, RFID systems in supply chain management have three main components: 1) a *server* (usually centralized), 2) *readers* (from tens to hundreds, depending on the application size), and 3) *tags* (potentially millions).

To better understand RFID system deployment, we reproduce an illustrative example of a supply chain management system as given in [65]. Figure 5.1 depicts the journey of a pack of razor blades from its manufacturer to a consumer. We start with the manufacturer, where one pallet consists of 90 cases with each case containing 72 packs. Considering the pallet, cases and packs are all tagged, a total of 6571 tags reach to a distribution center in one large group. This large pallet is then de-palletized and assembled back into smaller pallets depending on the orders placed by retail stores. Considering a smaller pallet can hold up to 10 cases, each pallet will now carry 730

tags stored in the backroom of a retail store. Normally up to two cases are displayed on the store shelf and a consumer may pick a few packs to purchase. The following is a typical hierarchy of some of the objects [65]:

- *Razor blades:* $6571 \rightarrow 730 \rightarrow 144 \rightarrow 5$
- *DVDs:* $5040 \rightarrow 2520 \rightarrow 400 \rightarrow 24$
- *Pharmaceuticals:* $7200 \rightarrow 1920 \rightarrow 150 \rightarrow 6$

These hierarchies may differ for various objects and retailers. The important point to note here is that the number of tags (tagged items) reduces in size from manufacturer to end-user. The larger group of tags is read by readers in a physically secure environment, whereas as the smaller number of tags, reaching to store shelf and consumers, is exposed to adversaries. Considering a typical supply chain process, we divide the lifecycle of a tag into the following two zones:

1. **Secure Zone with Online Readers.** This zone is assumed to be secure from all adversaries. A large number of tags are scanned by a limited number of known readers in this zone. Since the position of all the readers is known, these readers either share the database held with the back-end server which stores shared secrets for each tag, or secrets can be securely transferred to those reader's local databases. The main requirement in this zone is fast reading of the large number of passing tags.

2. **Insecure Zone with Offline Readers.** This zone is assumed to be insecure and open to all adversaries. A comparatively smaller number of tags are scanned by unknown readers in this zone. The position of readers is unknown and their local servers do not share secrets with the tags. The main requirement in this zone is to preserve privacy, while it is reasonable to compromise on read speed since the number of tags is smaller.

As discussed earlier in Section 2.3.1, EPCC1G2 standard [48] compliant tags are typically deployed in supply chain management for automated inventory checks. The UHF air interface protocol defines the standard of communication between a reader and a tag. The reader first selects a group of tags to be read in its vicinity. The reader then initiates an inventory round to read the tag's credentials until the whole group is read. Finally the reader enters into an access phase where it can write into a tag's memory (if required) using a built-in *Access* password. This protocol is further explained in Section 5.3.3.

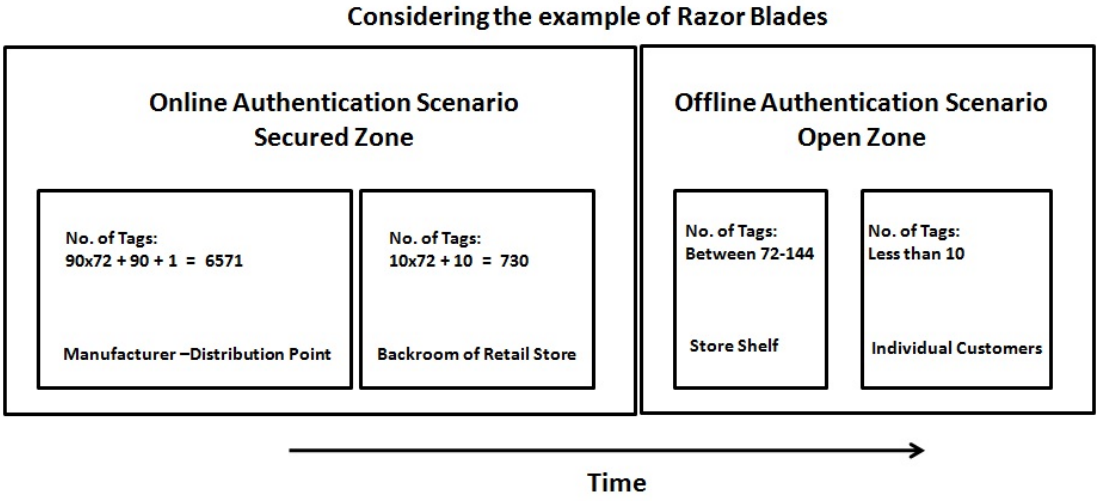

Figure 5.1: Object's Journey in RFID-enabled Supply Chain Management.

However, there are risks associated with this class of tags as given in Section 2.5. Since the standard does not elaborate on any specific authentication mechanism, a tag will respond to every query sent by a compatible reader. This causes privacy concerns [63] as follows:

- **Content Privacy.** An illegitimate reader can learn sensitive information associated with a tag's identifier, such as type, price, expiry, etc. This can further be used to profile the tag holder such as shopping habits, medical history and other private information.
- **Location Privacy.** An attacker can track a tag carrier since the tag's *EPC* is a unique and static identifier.

RFID systems using such tags cannot implement computationally intensive privacy-preserving protocols due to their limited resources. These tags have limited memory and computation capabilities. These are passive tags and draw power from a reader in order to compute and communicate. In addition, the amount of data transmitted between a tag and a reader should not be excessive, bounded by the available bandwidth.

5.1.1 Our Contribution

RFID tagged objects are read by multiple readers both in known locations (secure zone with online readers) as well as unknown locations (insecure zone with offline readers). In the secure zone, the primary requirement is to read a large number of tags with high speed. In the insecure zone, the primary requirement is to preserve the privacy of a user

of tagged object. We present an EPCC1G2 standard compliant scheme which allows RFID tags to be authenticated by readers throughout the supply chain lifecycle while meeting the requirements of both the secure and insecure zones. Our scheme adapts between online and offline authentication without requiring user-intervention. In this work we propose a scheme without involvement of any cryptographic primitive and using built-in functionalities in a tag. This work has been accepted to get published in [11].

5.2 Existing Work

Various ideas for addressing privacy issues in supply chain management have been suggested. Some of these proposals [61–63] are based on shared secrets (online authentication schemes) and do not address the requirements for tags to be scanned by offline readers. Furthermore, some of these [35, 77] are not EPCC1G2 standard compliant, while some [57, 127] require user intervention in order to preserve the privacy of a tagged object's user.

5.2.1 Password Protected Online Authentication Schemes.

The scheme given in [63] involves disabling RFID tags at checkouts using the existing *Kill* password. However, secure transfer of *Kill* passwords to offline readers with unknown locations is not feasible. By disabling tags, after-sales features such as receipt-less returns, automated warranty claims and recycling are not automatically facilitated. The scheme in [62] uses built-in *Kill* and *Access* passwords in an EPCC1G2 compliant tag for mutual authentication. While this mechanism avoids killing the tags permanently, a source must know its end destination in order to transfer corresponding passwords. Thus, readers must know all the passwords of potential tags, which could be millions in number, and thus requires a dedicated database. A small retail store cannot afford the luxury of a back-end database and an end-user cannot carry IT equipment in order to transfer all the passwords related to their tags. The proposal in [61] suggests using pseudonyms instead of the original identifiers of tags. However, fixed pseudonyms facilitate tracking, whereas cryptographically changing pseudonyms require readers to possess the same key and stay synchronized. Moreover, a central repository storing all pseudonyms requires access tokens. All of these schemes thus only work with online readers.

5.2.2 Additional Privacy Preserving Devices.

Another scheme proposed in [66] uses appropriate prefixes to EPC and an additional blocker tag to preserve the privacy of tags. For example, all the tags attached to sold items are declared to be private (no reader can query the tag) by setting their EPC's prefix bit(s) to some predetermined value. If an unauthorized reader queries these tags, the blocker tag, acting as intermediary, suppresses its queries. As well as requiring an additional blocker tag, this scheme also requires writing/setting the appropriate prefix into a tag's memory (for example at point of sale). This scheme is based on querying a tag using a binary-tree search algorithm and is not EPCC1G2 compliant.

The proposals given in [41, 67, 120] use a proxy device to suppress the stealth scanning of a tag's content. The proxy device acts as an intermediary between reader and tag. This smart device makes intelligent decisions in determining the legitimacy of a reader. However in these proxy devices, acquire and release control of tags during ownership transfer is difficult. It is also difficult to entirely suppress reader's commands and tag's replies.

5.2.3 Distance Bounding Protocols.

There are many proposals for distance bounding protocols [35, 73, 78] which determine the legitimacy of a reader based on its proximity, typically calculated from signal strength and query-to-response time measurements. However, since the read ranges vary considerably depending on the transmitted powers, antenna sensitivities and environment, the adversary may send a stronger signal than prescribed and read over a longer distance with a better signal-to-noise ratio. Therefore, these schemes can fail against such attacks. Moreover these protocols typically require additional circuitry in low-cost tags and are not EPCC1G2 compliant.

5.2.4 Relabeling and Partial Destruction.

Similarly some proposals suggest partial destruction of important and secret information of a tag. Relabeling [127] is one such proposal which requires changing the tag's label from secret to some public value in order to preserve the tag's privacy when the tag travels in the insecure zone. Partial destruction using splitting [57] requires two tags (one carrying the private information while the other has public information) on every item. The tag carrying secret information is removed to preserve the privacy when in the insecure zone. Both of these schemes require user interaction.

5.2.5 Bit Throttling and Secret Sharing Schemes.

To deter sporadic reading of a tag's secret content, the scheme in [77] reveals the secret content one bit at a time and thus delays the process of promiscuous reading of the tag's content. This makes it harder for a sporadic adversary to disclose or track a particular tag. However the data rate of this scheme is very low and it also requires additional circuitry to perform this task. Determining the sequence of bits for transmission is also a problem as sequential transmission (starting from the least significant bit) can reveal important information through only the first few disclosed bits (for example, the first four bits of the *EPC* reveal the commercial code and the next four suggest the size).

The scheme suggested in [65] adopts secret sharing where shares are distributed amongst different tags across time and space. When individual tags are sold to different customers, their privacy is preserved as an individual share does not reveal any sensitive information. However, warranty claims become cumbersome in these scenarios because an individual customer carries only one share of the secret and also needs to collect other shares which are distributed amongst other unknown customers. Another potential problem with this scheme is clandestine tracking as secret shares are static and do not change.

The proposal in [1] is based on delayed transmission of the secret value using *linear feedback shift register* (LFSR). This proposal is suitable in scenarios where the number of tags is small as it takes time to transmit the complete secret. It therefore does not address the requirement of high speed reading of a large number of tags in the secure zone. It also requires additional functionality other than the standard.

5.2.6 Our Scheme.

In this work, we consider taking an EPCC1G2 compliant approach that fulfills the requirements of both fast read speed, when a large number of tags are read by online readers in the secure zone, as well as preserving the privacy of a tag when read by offline readers in the insecure zone. Our unified scheme is based on delaying the disclosure of the secret (tag's content) until a certain time threshold is achieved, and adapts between online and offline authentication without user intervention. We focus our comparative analysis on the schemes presented in [1, 65, 77] since these are the only other schemes which use related techniques.

5.3 Proposed Scheme

We now explain our proposed scheme which provides privacy to EPCC1G2 compliant RFID systems deployed in supply chain management.

5.3.1 Adversarial Model

We make the following assumptions about the capability of an adversary:

- An adversary can conduct both passive and active attacks. Our scheme protects against passive attacks (eavesdropping both the forward and backward channels) and active attacks, except for physical capture and tampering attacks.
- An adversary cannot take over an ongoing authentication round because when the tag receives queries from multiple readers, it detects a collision and stops responding as explained in Section 2.2.2 (we assume the use of a reader anti-collision algorithm, see [48]).
- An adversary cannot learn the update values as only a legitimate reader in possession of the tag can update its memory.

The notation required to describe our scheme is shown in Table 5.1.

5.3.2 Goals

Considering a supply chain process consisting of the two zones identified in Section 5.1, our scheme is designed to achieve the following goals in the presence of an adversary as defined in Section 5.3.1.

- **Content Privacy.** Support privacy of a tag's content, wherever this is required.
- **Location Privacy.** Support privacy of the location of a tag in order to prevent tracing and tracking of the tag, wherever this is required.
- **Conformance to Standard.** The ownership transfer scheme designed for a particular RFID standard should conform with the standard's operations and functionalities as much as possible.
- **Fast Read Speed.** Support a fast read speed, particularly required when the number of tags is large.
- **User Transparency.** Adapt according to the status of the reader (i.e., online or offline) without user intervention.

Table 5.1: Notation used in Chapter 5

Notation	Description
Query	A command sent by the reader to a tag/group of tags it wants to read.
QueryRep	A command sent by the reader to a tag/group of tags if it receives no response, or multiple of responses from more than one tag.
SlotCounter	A counter implemented in the tag which loads a random number and decrements with every Query and QueryRep command.
$RN16$	A 16-bit standard random number generated by the tag and transmitted to the reader once its SlotCounter reaches zero.
ACK	A 16-bit acknowledgment sent by the reader to the tag.
$PC + EPC + CRC$	A tag's content plus its cyclic redundancy check.
Access	A built-in 32-bit unique access password in each tag.
r	A 16-bit random number generated by the tag.

5.3.3 Overview of Protocol

We use the existing functionality of EPCC1G2 standard tags [48]. The standard defines the UHF air interface protocol shown in Figure 5.2. We now give an overview of our proposed protocol. Note that we need to make a couple of very minor changes to the standard in order to support an authentication mechanism (see Section 5.4.3).

1. **Initialization.** In the original standard [48], each tag generates a random 16-bit number $RN16$ on the fly. We suggest that each tag is initialized using a unique random $RN16$ in its local group. It is important to note that this only limits a group size to 2^{16} tags and does not affect the EPC, which is a 96-bit unique code. This modification can easily be incorporated into the standard.

2. **Unique Allocation of RN16 within a Group.** Initially manufacturers can write this into the tag's memory and later the back- end server, in possession of the corresponding *Access* password, can update the value of $RN16$ by writing into the tag's memory using a compatible reader. Since a server keeps updated record of groups of tags, the former can ensure unique allocation of the updated

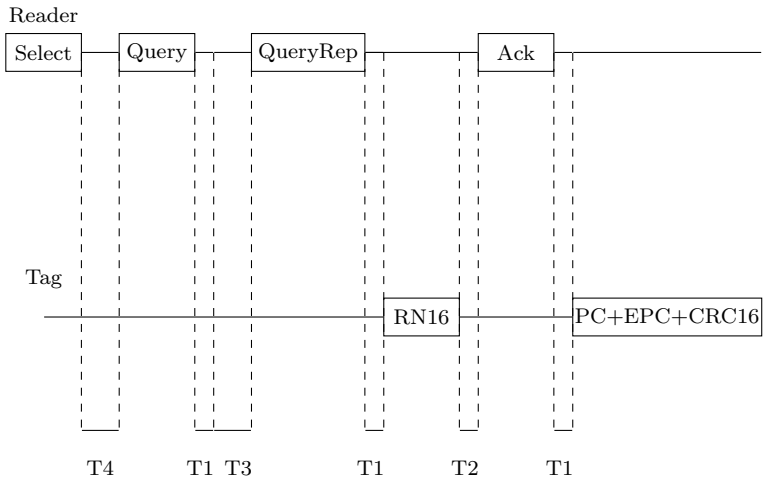

Figure 5.2: UHF Air Interface Protocol for Class-1 Gen-2 Tags.

$RN16$. Following are the steps to ensure that each tag in a group is allocated a unique $RN16$ depending on the number and size (number of items within one group) of the group:

- A back-end database server has a total of $2^{16} = 65536$ values of $RN16$ which can be allocated to items uniquely within one group.
- Since the server has knowledge of total number of groups and size of each group within a certain supply chain cycle, it can create a pool of available $RN16$ values for each group.
- A certain group can be identified within a supply chain using different variables like geographic location, date and time, lot numbers, item classification. Additionally, another workaround to identify a certain group can be ensured by putting an additional tag on the pallet.

We now consider an example to clarify how unique allocation is ensured within one group. Since a server can trivially identify a certain group, we suppose that this group $Group - X$ has a total of 20 items where each tag incorporates a 5-bit identifier. This is depicted in Table 5.2. Now when each tag from $Group - X$ is read, it is allocated an $RN16$ from unallocated pool randomly while its older allocated value of $RN16$ is added to unallocated value. This ensures that each tag is allocated a unique $RN16$ within one group and any $RN16$ value is not repeated in succession in order to resist a traceability attack.

3. **Initial Identification.** The unique random $RN16$ is used to identify a tag in

Table 5.2: Unique Allocation of RN16 within Group-X

S/No	Allocated	Unallocated
1.	00101	00000
2.	00111	00001
3.	01000	00010
4.	01010	00011
5.	01011	00100
6.	01100	00110
7.	01110	01001
8.	01111	01101
9.	10010	10000
10.	10011	10001
11.	10100	10110
12.	10101	11101
13.	10111	
14.	11000	
15.	11001	
16.	11010	
17.	11011	
18.	11100	
19.	11110	
20.	11111	

the reader's back-end server.

4. **Mutual Authentication.** We incorporate a mutual authentication stage inside the inventory round (see [48]). The standard defines two secret values, *Kill* and *Access* passwords, that are embedded into every EPCC1G2 compliant tag. The *Kill* password is used for disabling a tag and the *Access* password is used for read/write access to the tag. Both passwords are 32-bits long. We use the *Access* password for both mutual authentication and read/write access, while retaining the *Kill* functionality where required. We divide the *Access* password into two parts consisting of the 16 LSB (used for reader authentication) and the 16 MSB (used for tag authentication).

5. **Standard Protocol.** After successful mutual authentication, tags are read as

per the standard [48], as shown in Figure 5.2.

6. **Update.** We use the access round (see [48]) to enable a legitimate reader to update the values of $RN16$ and the *Access* password by writing into the tag's memory. Note that this update can only be carried out by a back-end server in possession of the tag's *Access* password.

7. **Determining Threshold.** Our offline authentication stage is based on a time threshold value (as will be explained in Section 5.3.5). Therefore it is important to determine a suitable threshold value which prevents an adversary from disclosing the contents of the tag or identifying its location. As per the standard, the reader powers up the tag, sends the select and query commands, receives the response $RN16$ from the tag, and then transmits an ACK in response. If the ACK is valid, the tag answers back by transmitting its content. The reader then powers down the tag. The whole process, ignoring the proposed mutual authentication messages, takes approximately 35 milliseconds (see [48]). A legitimate offline reader does not power down and power up the tag until the required time threshold is achieved. Considering power down time is 1 millisecond, one cycle of the standard scanning process without powering down the tag will take approximately 34 milliseconds for the first cycle and then 32.5 milliseconds (since power up time is 1.5 millisecond) for each subsequent cycle. Consider a realistic scenario for supply chain management where legitimate offline readers are present in retail stores and smart home appliances. These readers can scan the tags for a relatively long time and then change their status to online after obtaining the shared secret when the time threshold is achieved. The precise time threshold value can be set by a manufacturer depending on the application.

The overview of the protocol is provided in Figure 5.3. Our scheme starts when a reader sends the acknowledgment, which is compared with the value of the *Access* password stored in the tag. If it matches, our online part of the authentication scheme takes over, otherwise it switches to offline mode. Since a reader is only an intermediary device between a server and a tag, each reader is connected to either a back-end server with stored shared secrets with the tag (online readers) or local servers without any information about tags. This connection between reader and server is assumed to be secure, hence we use the term *reader* to encompass reader, server and their communicating channel in this work. Both the online and the offline mechanisms are explained in subsequent sections.

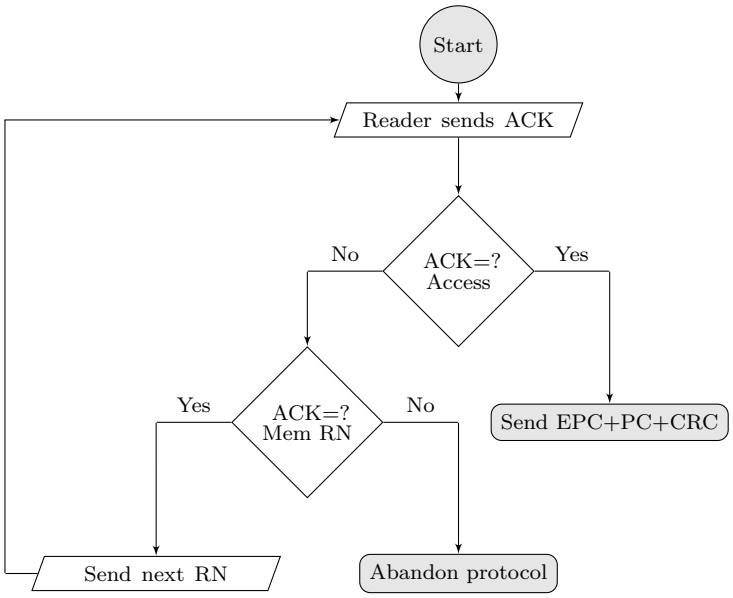

Figure 5.3: Overview of the Proposed Scheme.

5.3.4 Online Authentication Stage

Online authentication is based on shared secrets. Online readers have known locations, and secret passwords (*Kill* and *Access*) for each tag are securely distributed to every reader in the chain (more precisely all readers share the database storing secret passwords of each tag). The main requirement in this stage is to achieve a fast read rate, since the number of tags is large and the area is considered to be physically secure (see Section 5.1). Since the UHF Air Interface Protocol does not define any authentication mechanism [48], we modify the standard functionality by changing the $RN16$ sent by the tag and the ACK sent by the reader to achieve mutual authentication. Our online authentication scheme is motivated by [62] and defined as follows:

1. **Initialization.** Each tag is initialized with a unique fixed $RN16$.

2. **Initial Identification.** Online readers identify a particular tag using $RN16$ as an index to a database (held at a back-end server) and extract *Access* password.

3. **Mutual Authentication.** A valid ACK is now the 16 LSBs of the *Access* password. The tag authenticates the reader by comparing this value sent by the reader with the value stored in the tag's memory. If both are equal, then the reader is considered to be online and legitimate, else either the reader is offline or not legitimate. In case of successful authentication, the tag now sends the 16 MSBs of its *Access* password, which the reader uses to authenticate the tag.

4. **Standard Protocol.** After successful mutual authentication, the standard as shown in Figure 5.2 is followed. The reader now sends a standard ACK (which is the same $RN16$ sent initially by the tag) and the tag in return sends its information to the reader.

5. **Update.** The legitimate reader updates $RN16$ and the *Access* password values in the tag using existing *Access* password. The same update is carried out in the back-end database as well.

The online authentication scheme is summarized in Figure 5.4, assuming the protocol follows the standard until the slot counter of a particular tag reaches zero.

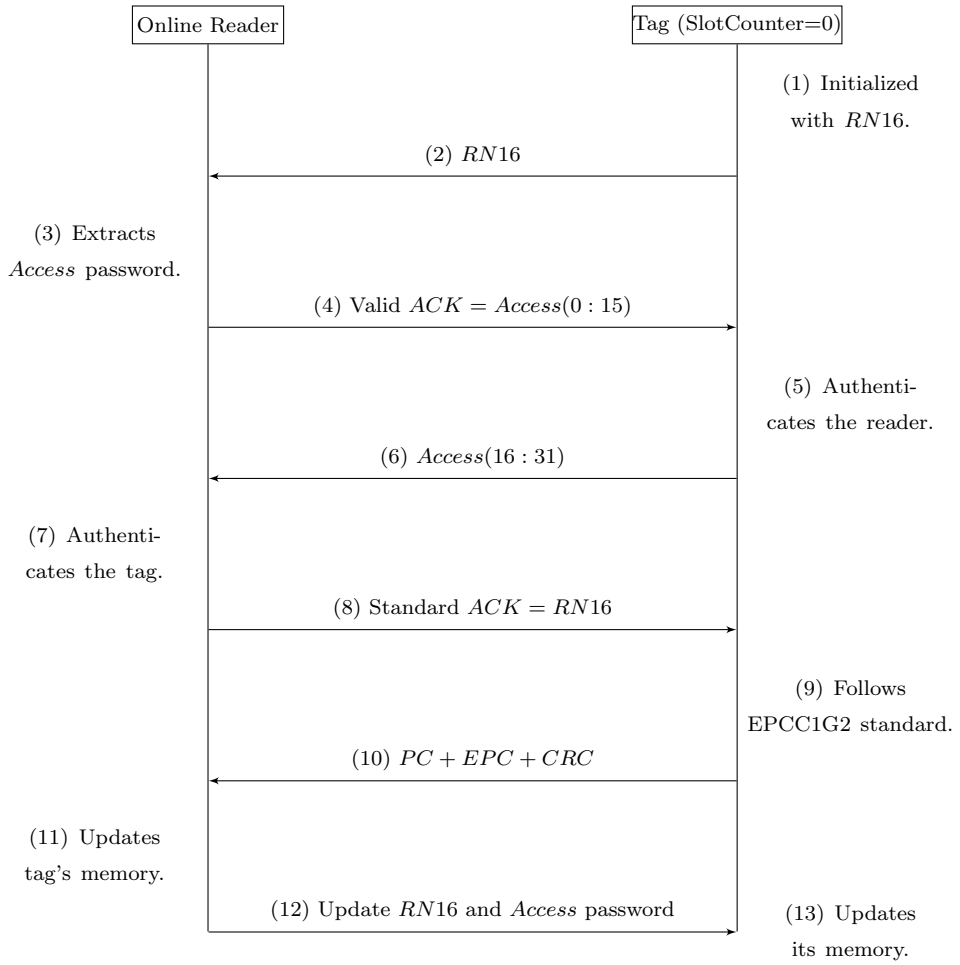

Figure 5.4: Online Authentication Scheme for Class-1 Gen-2 Tags.

5.3.5 Offline Authentication Stage

Offline readers have unknown locations and it is infeasible to distribute secret passwords (*Kill* and *Access*) for each tag securely to every such reader. The main requirement in this stage is to preserve privacy, with a willingness to compromise on read speed since the number of tags is small and the area is considered to be physically insecure (see Section 5.1). The UHF Air Interface Protocol works as in the standard except that the $RN16$ sent by the tag and ACK sent by the reader changes in the proposed scheme (see Section 5.3.3). The ACK is checked by the tag in order to establish which of the following three states apply:

1. *Valid* if ACK is equal to the 16 least significant bits of the *Access* password.
2. *Semi-valid* if ACK is equal to the random values generated by tag.
3. *Invalid* otherwise.

The offline part of our authentication scheme is motivated by [1]. This scheme is defined as follows:

1. **Initialization.** Each tag is initialized with a unique fixed $RN16$.
2. **Initial Identification.** Offline readers cannot identify a particular tag using $RN16$, so it cannot send a valid ACK, which is the 16 LSBs of the *Access* password of the corresponding tag.
3. **Mutual Authentication.** An offline reader sends a semi-valid ACK, which is equal to the $RN16$ (as per the existing standard [48]) sent by the tag and stores a copy of $RN16$ in its memory. The tag first checks its validity by comparing it with the 16 LSBs of the built-in *Access* password. In case of failure, it checks its semi-validity by comparing this with the $RN16$ stored in its memory. If the ACK is semi-valid, the tag switches to offline mode. The tag now generates another 16 bit random number r_1, XORs it with the previous $RN16$, transmits the result sum_1 ($sum_1 = r_1 \oplus RN16$) to the reader, and stores r_1 and sum_1 in its memory. The reader, on receiving this new sum_1, stores its value and performs the same operation ($r_1 = sum_1 \oplus RN16$) and sends the result r_1 to the tag (see Figure 5.5). The tag continues checking for a valid, semi-valid or invalid ACK, and responds accordingly. Once this repeated communication reaches a certain threshold, and the tag determines (by comparing the r_{Th-1} sent by the reader with its stored value) that the reader has spent enough time in pairing up, it performs an XOR of the previous value of sum_{Th-1} stored in its memory with the 16 LSBs of its

Access password and sends the result as sum_{Th} to the reader. On receiving this 16-bit number, the reader also performs the XOR of this new value sum_{Th} with the previous one sum_{Th-1} and extracts the 16 LSBs of the *Access* password. Once the reader transmits these 16 bits as an ACK, the tag regards it as valid. On receiving a valid ACK, the tag switches to online mode.

4. **Standard Protocol.** After successful mutual authentication, the EPCC1G2 standard is followed as shown in Figure 5.2. The reader now sends a standard ACK (which is the same $RN16$ sent initially by the tag) and the tag in return sends its content to the reader.

5. **Update.** The legitimate reader updates $RN16$ and the *Access* password values in the tag using existing *Access* password. The same update is carried out in the reader's local database as well.

The scheme is summarized in Figure 5.5, assuming the protocol follows the standard until the slot counter of a particular tag reaches zero.

5.4 Analysis

In this section, we carry out an analysis of our protocol for the desired goals stated in Section 5.3.2, and compare it to existing proposals [1, 65, 77] which are based on a similar mechanism as mentioned in Section 5.2. We summarize this comparison in Table 5.3.

5.4.1 Content Privacy

A common criticism of the use of RFIDs is that the tags reveal content promiscuously to any compatible reader. Our scheme protects the content of a tag by sending content to only authorized or trusted readers. In the secure zone with online readers, the tag sends its content only after successful mutual authentication. Considering the area is secured, we rule out the possibility of content disclosure to any adversary. In the insecure zone with offline readers, the tag first sends random information if it does not trust a reader until a certain trust threshold is achieved, then the content of the tag are sent after a successful mutual authentication phase. A recent proposal [1] based on the concept of transmitting a shared secret in parts tends to leak information after every transmission unless the secret is revealed. Our scheme does not reveal any information until the trust threshold is achieved. A comparison is shown in Figure 5.6. We analyze the strength of our scheme by considering the following adversarial behaviour:

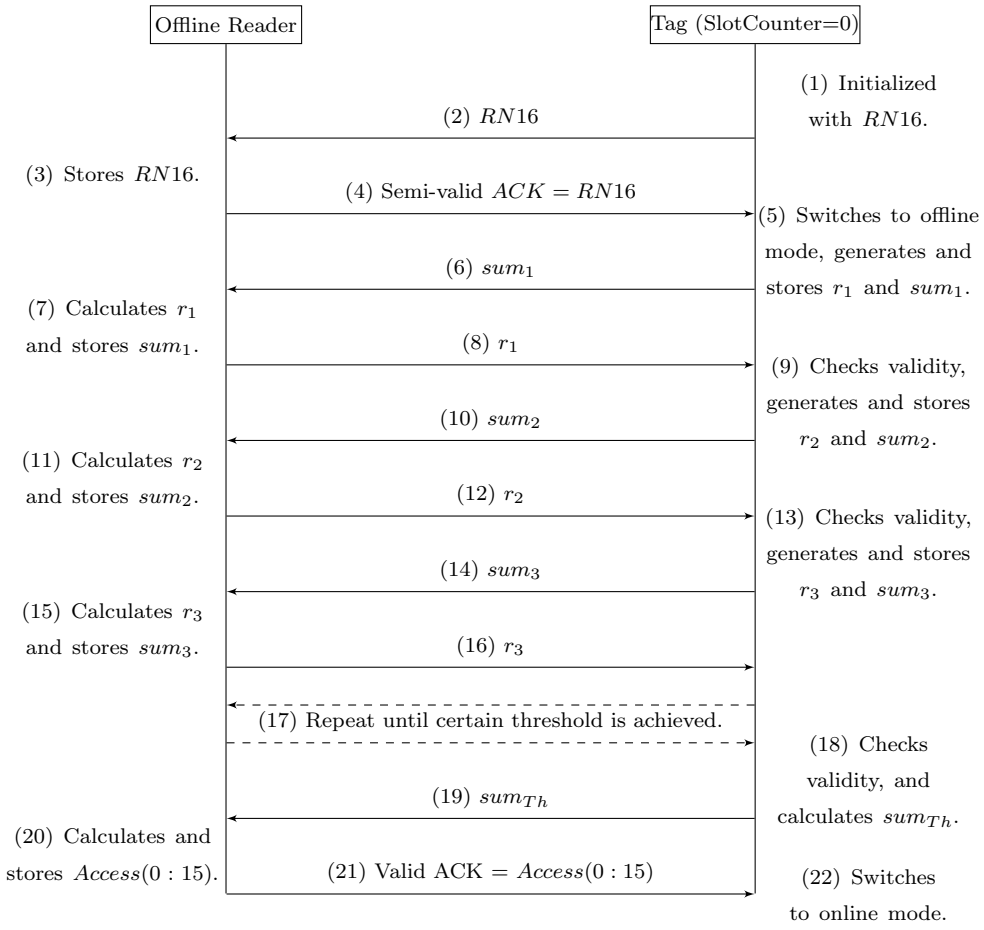

Figure 5.5: Offline Authentication Scheme for Class-1 Gen-2 Tags.

- **Online Adversary:** We assume that online readers scan the tag in a secure area (see Section 5.3.1). Therefore, we rule out the possibility of a passive adversary listening to communication between an online reader and a tag. However, an active adversary can act as online (in the insecure area) and the secret *Access* password can be retrieved by a brute force attack. Simply, a reader can send a random ACK to a tag until the tag sends back its content, which means that the reader has found the correct password. In each guess, the online adversary has to complete the scanning cycle as mentioned earlier in Section 5.3.3. If the tag does not answer back with its content, the reader powers down the tag and repeats the sequence with a different value of the ACK. Considering the EPCC1G2 specification, each try takes 35 milliseconds and a 16-bit password is thus exhausted in about 38.23 minutes (19.11 minutes on average). We consider that an adversary who is not in possession of the tag will generally not have

Table 5.3: Comparative Analysis of Proposed Scheme vs Existing Schemes

Security Features	**Marc [77]**	**Juels [65]**	**Amariucai [1]**	**Proposed**
Unified Approach	No	Yes	No	Yes
EPCC1G2 Compliance	No	Yes	No	Yes
Read Speed	Slow	Fast	Slow	Fast(secure zone)
Content Privacy	Reveals pattern	Preserved	Preserved	Preserved
Location Privacy	Preserved	Not preserved	Preserved	Preserved
Information leakage	Gradual	Gradual	Linear	No Leakage

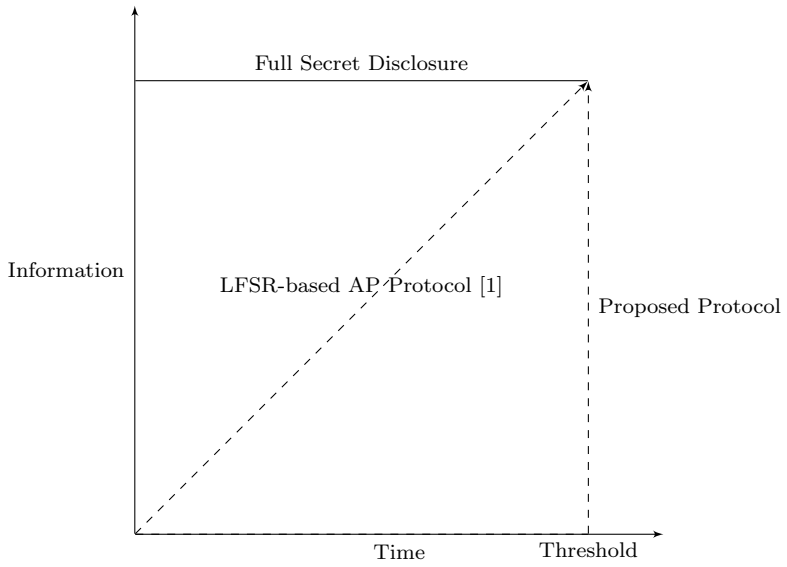

Figure 5.6: Information Leakage Comparison.

sufficient time to do this before being detected.

- **Sporadic Offline Adversary:** A more realistic scenario is of a sporadic adversary who is capable only of scanning or eavesdropping some of the random information exchanged between a reader and a tag. This random information will not be sufficient to acquire the threshold or disclose the tag's content. Thus the adversary has to keep track of all the communication sessions. However, a sporadic adversary can eavesdrop either the last session, or the second-last session (see steps after the threshold is achieved in Figure 5.5) by chance. The probability of success will be $1/n$ for a threshold of $n-1$ random sessions, since each session is independent. Moreover, the adversary cannot take over an ongoing authentication round (see Section 5.3.1) and has to wait for it to complete. Once an authentication round is complete, the adversary cannot replay the eavesdropped values or act as online since these values are updated in the tag (see Section 5.3.3).

- **Dedicated Offline Attacker:** A dedicated offline adversary is assumed to act like a legitimate offline reader. This adversary is able to scan the tag until a threshold is achieved. Therefore, the adversary is able to disclose the *Access* password and content of the tag. After achieving the *Access* password, the adversary will impersonate as an online reader. It can thus downgrade the legitimate owner to offline by updating the tag to its own values of $RN16$ and *Access* password. However, if the adversary is not in possession of the tag, this success will be one time only. The adversary will no longer be able to disclose this tag's content since the *Access* password is updated by the legitimate owner (in its next communication with the tag). The countermeasure for such an adversary is to set the time threshold value to be sufficiently high that this adversary can be detected before the tag reveals its secrets.

5.4.2 Location Privacy

Our scheme preserves the location privacy of a tag and hence prevents its tracking. Since $RN16$ and the *Access* password are changed in every authentication round, and the tag sends different random numbers when queried by an unauthorized reader, its location cannot be tracked. The tracking depends on the properties of the random number generator on the tag. The specification in the standard [48] is as follows:

Probability of a single $RN16$: The probability that any single $RN16$ drawn from RNG has value $RN16 = j$, for any j, shall be bounded by $0.8/2^{16} < P(RN16 = j) < 1.5/2^{16}$.

Probability of simultaneously identical sequences: For a tag population of 10,000 tags, the probability that any two or more tags simultaneously generate the same sequence of $RN16$ shall be less than 0.1%, regardless of when tags are energized.

Probability of predicting $RN16$: An $RN16$ drawn from a tag's RNG $10ms$ after the end of Tr (rise time of reader power-up waveform) shall not be predictable with a probability greater than 0.025% if the outcomes of the prior draws from the RNG, performed under identical conditions, are known.

5.4.3 Conformance to Standard

Many of the earlier proposals cannot be implemented in low-cost environments (see Section 5.2), particularly EPCC1G2 standard compliant tags, or require considerable changes to the existing standard. Our scheme can easily be implemented in these tags with very minor changes to the standard and uses existing functionality as defined in the standard [48]. The proposed scheme conforms with the standard operations as follows:

- **Select.** This operation is used to select a single or a population of many tags. We use the existing select operation as per the standard.
- **Inventory.** Once a tag selected for identification, inventory operation is initiated. This operation concludes when the selected tag sends its content to reader. We incorporate a mutual authentication phase in inventory round for online authentication while a time based threshold pairing phase is included for offline authentication.
- **Access.** Access involves read from/write to and other operations requiring to access the tag's memory. We use the built-in *Access* password for updating the tag's memory.

Following are the additional overheads in addition to standard functionalities:

- **Storage.** Our scheme requires the tag to store two additional 16-bit values (8 GE/bit for temporary storage) in addition to storing a random number as in the standard.
- **Computation.** The additional mechanism uses the existing functionalities of an EPCC1G2 compliant tag for generating a 16-bit random nonce and conducting an XOR computation.
- **Communication.** There are additional communication overheads to achieve mutual authentication and a time threshold. In the online authentication scheme,

there is an additional mutual authentication mechanism which is completed in two additional messages and authentication is based on the existing built-in *Access* password. In the offline authentication scheme, the reader has to acquire a time threshold in order to read the tag's content.

5.4.4 Fast Read Speed

EPCC1G2 certified readers have two read modes namely fast and slow. The read speeds are automatic and depend entirely on the actual read conditions for each tag. In multi-tag environments, where thousands of tags are passing in front of readers, speed is of the utmost importance. Fast read speed requirement exists in the secure zone with online readers. Our proposed scheme reads the tag using the same standard functionality in the secure zone with online readers. Thus this requirement is fulfilled using our proposed scheme.

5.4.5 User Transparency

As discussed in Section 5.2, some of the earlier schemes require user intervention to preserve the privacy of the tag. These systems are prone to errors and are labour-intensive. Our proposal adapts between online and offline authentication modes without any user intervention.

5.5 Summary

In this chapter, we have proposed a scheme that provides a unified approach to tackling privacy and performance issues in RFID-tagged supply chain management. Unlike any existing proposal in the literature, it is easy to implement in the existing EPCC1G2 standard, it provides fast read speed in the secure zone, and preserves privacy in the insecure zone, and it adapts between online and offline authentication without user intervention.

Chapter 6

A Hierarchical Anti-counterfeit Mechanism

In this chapter, we address the counterfeiting threat to supply chain management using RFIDs. Section 6.1 provides an overview of the counterfeit problem in RFID systems. The existing solution and their drawbacks are discussed in Section 6.2. Our proposed anti-counterfeit mechanism is presented in Section 6.3. The analysis of our scheme is carried out in Section 6.4 to determine whether it achieves the desired goals.

6.1 Introduction

RFID technology has replaced barcode mainly because items can be individually identified without line-of-sight requirements (see Section 1.1). Although RFID systems face similar challenges to those faced by barcode technologies, such as cloning and impersonation, RFID systems have the advantage that they are capable of providing identification as well as authentication. However, counterfeiting, caused by cloning and impersonation attacks, has been a problem for some RFID systems [63]. The counterfeiting of products is one of the most serious threats to modern commerce according to estimates by the Counterfeiting Intelligence Bureau (CIB) of the International Chamber of Commerce (ICC), which claims that counterfeit goods account for up to 5-7% of world trade [58]. Counterfeits products can be found everywhere starting from low cost items such as biscuit packs, food tins, DVDs and medicine bottles, to high value goods such as watches, designer clothes, cars, motorcycles and bicycles.

RFID tags are attached to products for remote identification. Among these, EPCC1G2 compliant tags are the most widely used because of their world-wide standardization.

The EPCC1G2 standard is used for supply chain management (see Section 5.1) and can be used as a tool for anti-counterfeiting [130]. A tagged object starts its journey from manufacturer to customer as part of a large group [65]. During this journey, the object may be read by multiple readers located from the manufacturing company through to retail stores. Finally the object is sold to the end-user/customer. To address counterfeiting, RFID researchers have designed many schemes which trade-off between cost, security, and performance, however existing approaches all have significant drawbacks which we outline in Section 6.2. In this chapter, we shall propose a mechanism to counter counterfeiting in the supply chain management system. However, this mechanism cannot address counterfeits when items reach customer level. In the next chapter, we shall propose an anti-counterfeit framework for individual customers.

Since a tag will respond to every query sent by any compatible reader, if no authentication mechanism is employed, an adversary can query a genuine tag and learn the sensitive information associated with the tag's identifier, which can then be used to make counterfeit tags (see Section 2.5). When using authentication, a tag will respond to every query sent by a compatible reader that has been authenticated as legitimate. However, the adversary can still eavesdrop the tag's identifier and then copy this information to a counterfeit tag. So there is a need for *secure identification with authentication*, in which case a tag will securely provide its information in response to every query sent by a compatible reader that has been authenticated as legitimate. Although an adversary cannot learn the sensitive information, if this information is static then it can be copied or replayed by counterfeit tags to impersonate genuine tags. Finally the adversary can collude with legitimate but dishonest middle parties to gain benefits from counterfeiting. Considering these threats and capabilities of the adversary, there is a need of an anti-counterfeit mechanism to identify dishonest middle parties involved in both counterfeiting and processing stolen/missing items.

6.1.1 Our Contribution

Counterfeiting is a very serious threat to supply chain management systems. RFID systems are widely used to automate and speed up the process of remotely identifying products, however these systems are vulnerable to counterfeiting as shown in Section 6.2. In this chapter, we propose a hierarchical anti-counterfeiting mechanism for EPCC1G2 standard (see Section 2.3 compliant tags. Our mechanism uses three layers of verification. Our layered approach, based on the use of shared secrets to generate dynamic verification codes which change in each transaction, offers scalability and is suitable for different sizes of groups of tagged items, as well as individual tags. Our scheme not only provides protection against counterfeiting but also identifies dishonest

middle parties. Additionally, it can detect any missing or stolen items and is sufficiently scalable to be applicable to the complete lifecycle of a tagged object within a supply chain management. This work has been published in [10].

6.2 Existing Work

There are several existing approaches to managing RFID counterfeiting (see for example [83,84,124]). We briefly review some of these schemes and identify their drawbacks.

6.2.1 Unique Serial Numbers

Every item equipped with an EPCC1G2 compliant tag carries a 96-bit code to uniquely identify and manage the item in a supply chain. Several proposals [75,136] use unique serial numbers to identify products. These numbers are used to track the physical location of a tag and update the results in an online database to check legitimacy and highlight any missing items. The *EPC* of a counterfeit product will appear twice (at least) in the database, assuming the counterfeit product is equipped with a cloned tag. However, this technique is detection only and does not prevent counterfeiting since the serial number is transmitted in the clear and any adversary can eavesdrop the serial number in order to clone or impersonate it. If a genuine tag is removed, and a counterfeit tag is impersonated as genuine, this scheme cannot detect it.

A number of proposed schemes [75, 106, 130] include a *track and trace* method where a counterfeit or missing item can be tracked down and traced back anywhere in the process. This is done using the complete trail of the exchanges of a cloned tag which is updated by each shipping and receiving record. The main disadvantage of this approach is its significant communication and computation overheads. Every reader has to update records in the online database server in real time. The online server has to track and trace each code received from the online reader and generate triggers in case of any abnormality. Thus, this approach is time consuming and creates bottlenecks if multiple clones are detected at the same time, as each cloned tag is individually checked using its complete shipping record from the database. In addition to these overheads, there are also some privacy concerns associated with this approach: for example, tracking of individuals from the products they carry, or tracking medicines, etc [2]. Another drawback of this approach is that a genuine but dishonest retailer can copy a genuine tag and attach this copy to a counterfeit product. They can then sell the counterfeit to a customer, who verifies it to be legitimate using a track and trace process that is not updated by the retailer or the middle parties [130]. Since track and trace process needs an update by each middle party, it is therefore vulnerable to both

intentional and unintentional errors [84].

6.2.2 Cryptographic Anti-counterfeit Mechanisms

Cryptographic mechanisms can be used to tackle counterfeits. The basic idea is to base authentication on a secret value possessed by each tag, which is then disclosed to the verifier as a proof of authenticity in a challenge-response, protocol [140]. Generally, this uses an encrypted challenge-response protocol, as it may be eavesdropped and the secret cloned if sent in the clear. This approach may be based on symmetric cryptography or asymmetric cryptography.

If symmetric cryptography is used [31, 36, 37, 115], the secret is already known to the verifier who matches it with the secret value received from the tag. To avoid a single point of failure, each tag is given a unique secret key, hence there will potentially be millions of such keys. This results in a requirement for a secure and efficient key distribution mechanism to distribute the tags' secrets among the readers. One approach to establishing all these keys is to distribute all keys to each reader in the form of a local database. However if a reader's local database is compromised then this approach results in the breaking of the whole system. A preferred approach is to store all the keys in an online database which each reader can access. This server is online at all times to provide the secret values of tags to readers. Assuming millions of tags are deployed in the supply chain with hundreds of compatible RFID readers, this approach incurs even more extensive communication, computational and storage overheads than the track and trace approach [2, 83], and even higher than the unique serial numbers approach. In addition, the reader needs to be trusted by the supplier since the reader stores or accesses the secret values of the tags in any system based on symmetric cryptography.

In contrast, asymmetric cryptography can be used [2, 7, 94, 153] to distribute keys. However, this still requires each tag to have a unique secret key and involves considerable computational overheads. Although researchers have proposed some lightweight public key cryptographic systems such as WIPR [104], it is still unclear whether such schemes can be deployed in the resource-constrained low-cost RFID systems (EPCC1G2 compliant tags) used in supply chain management.

6.2.3 Unclonable RFID tags

Physical Unclonable Functions (PUFs) are tamper-proof, unclonable items of hardware which produce a unique signature, given an input. In [139] an offline authentication scheme based on physically printed challenge-response pairs from a certain PUF was proposed for tag authentication. However, the printout has to be physically read and

cannot easily be automated. Further, it is relatively easy to program a cloned tag to give responses to particular challenges instead of using a PUF. These issues were addressed in another PUF-based scheme [82], but tracking of a tag is possible in this scheme as the PUF identifier is unique and does not change. Moreover, it is infeasible to maintain a large number of challenge-response pairs for one tag, potentially resulting in the few challenge-response entries being eavesdropped and the cloning of the tag.

6.2.4 Built-in Passwords

Juels has suggested a solution based on the tag's built-in passwords to counter the threat of cloning [62]. The idea is to use the existing *Kill* and *Access* password PINs to perform mutual authentication between reader and tag in order to avoid cloning. The reader sends a set of apparently random values except that one is the correct password PIN. The tag in response has to send the position of the correct PIN to get its legitimacy verified. However, legitimate but dishonest readers can store the complete set of PINs with a tag's responses and can thus clone the tag. Even if the reader is honest, the challenge set of PINs and responses can be eavesdropped. Juels also noted that this scheme is not secure against a simple three-step attack [62] based on skimming a tag identifier, interacting with the reader to obtain the challenge set of PINs, and then using these to obtain the correct PIN.

6.3 Proposed Anti-counterfeiting Mechanism

We now propose a new approach to prevent counterfeiting in supply chain processes where tags travel in groups. Our mechanism is based on a hierarchical model which involves three layers of verification. The three layers can be considered to range from low to high complexity with respect to trade-offs between cost, performance and level of security. If an upper layer verification fails, verification drops down to the next layer. We design new first and second layers, and then use the track and trace approach [75] for the third layer.

6.3.1 Goals

Our anti-counterfeiting mechanism is designed to achieve the following goals:

- **Anti-cloning.** Protection against copying the data from a genuine tag attached to a legitimate product and cloning it onto another tag attached to a counterfeit (see Section 6.2.1).

- **Anti-spoofing.** Protection against replay (impersonation) attacks (see Sections 6.2.3 and 6.2.4).
- **Anti-theft.** Detection of stolen or missing items.
- **Scalability.** Ability to operate efficiently when tags are in large groups as well as when a tag is attached to a single item.
- **Conformance to Standard.** Conform with the standard's operations and functionality as much as possible.
- **Efficient Key Management.** Supportable using an efficient key management scheme (see Section 6.2.2).
- **Good Throughput.** Avoidance of bottlenecks which degrade the overall supply chain system throughput (see Section 6.2.2).

6.3.2 The Layered Approach

The three hierarchical layers used for the legitimacy verification of a product (see Figure 6.1) are:

1. **Group Verification (GV) Layer.** For most of their journey in the supply chain products travel in groups (based on their type, specification, manufacturer and lot number, etc.) Our first layer verifies a complete group. In this layer the reader does not need continuous access to a central repository for verifying each tag because the complete group is read first and then verified as a whole.
2. **Product Verification (PV) Layer.** If *GV* layer verification fails, product verification is initiated using an individual tag's verification code. This lowers the performance and throughput since the reader has to access the database multiple times. Since the server verifies the legitimacy of a single product, the additional computational overheads are acceptable since the server is anticipated to be powerful. The *PV* layer identifies individual products that are either counterfeit or missing, their values and complete specifications.
3. **Track and Trace (TT) Layer.** After the *PV* layer has identified counterfeit or missing products, track and trace is initiated using the complete shipping/receiving record of the product. This gathers important information that includes the location of the anomaly and the type of anomaly (dishonest reader, counterfeit tag, or tag completely missing).

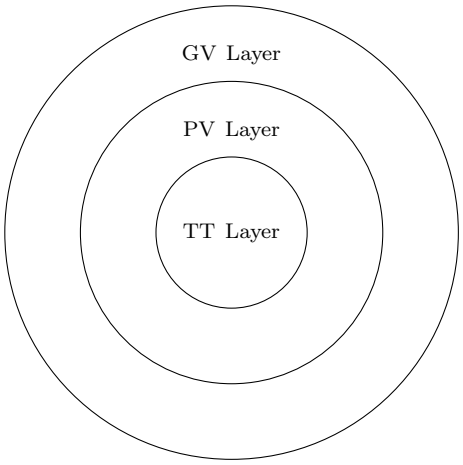

Figure 6.1: Hierarchical Verification Model.

Each layer of hierarchical verification detects anomalies in the supply chain in the following order:

- **GV Anomaly Detection.** *GV* mainly fails if the reader is not legitimate, a counterfeit is detected, or a tag is completely missing. When *GV* fails, this will generate an alarm in the server. The server will record the location and details of the reader where the alarm is raised. The server then switches to the *PV* layer.
- **PV Anomaly Detection.** *PV* identifies the exact cause of *GV* failure. It highlights the exact tagged product which is either counterfeit or missing. The server makes a corresponding entry relating to this particular tag.
- **TT Anomaly Detection.** *TT* is carried out as a last step which recovers the complete shipping/receiving record of the tag that was identified in *PV* anomaly detection. This further shows whether more clones exist in the supply chain, or whether the original product is completely missing. The server records the details of anomalies detected by *TT*.

6.3.3 Hierarchical Anti-counterfeiting Mechanism

We now explain the detailed operation of our hierarchical anti-counterfeiting mechanism. The notation used is summarized in Table 6.1.

Key Distribution Phase.

In this phase, the supplier who is responsible for shipping the tagged items in groups (or standalone as explained in Section 6.3.1) to different geographic locations holds a

Table 6.1: Notation used in Chapter 6

Notation	Description
S	A server holding the database with shared secrets.
R_h	A reader scanning a group of tags with index h.
G_j	A group of many tags with index j.
d	The number of tags in a particular group.
HVT_i	A tag participating in hierarchical verification with index i.
EPC_i	A tag's (with index i) static and unique identity.
KS_{tj}	A group secret key for every tag in a group with index j.
KM_r	A master key given to each reader in supply chain management.
KDF	A key derivation function agreed between server and all readers.
SK_{hj}	A session key derived from master key by reader (with index h) scanning a group (with index j).
r_j	A 96-bit random challenge generated by server for a group or a single tag (both with index j).
TVC_i	A 96-bit tag verification code used to verify the legitimacy of a tag with index i.
RVC_h	A reader verification code used to verify the legitimacy of a reader with index h.
GVC_j	A group verification code used to verify the legitimacy of a group of tags with index j.
$EGVC_j$	Encrypted group verification code for group with index j.
t_{out}	The maximum time limit of sending a message and receiving its reply.
$F_K(X)$	A lightweight secure PRF such as Hummingbird-2 [34] designed for EPCC1G2 compliant tags.
$E_K(X)$	A secure PRP such as AES.
$X \oplus Y$	Exclusive-OR of two values X and Y.

database with shared secrets. This database is securely held at a central back-end server S. The supplier distributes the keys as shown in Figure 6.2. There are n tags grouped in m groups depending on their type, specification, application, date of manufacture, lot number, date of expiry and geographic location, etc. Since, $n \gg m$, it is easy to distribute a total of m keys to n tags (the same key KS for each tag belonging to one group). The number of readers that scan these groups is denoted by s. The supplier distributes one master key KM_r to each reader.

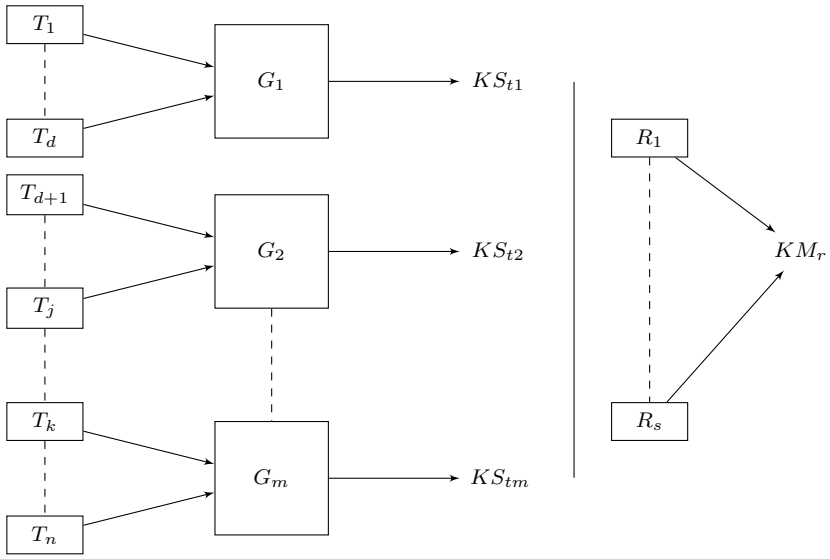

Figure 6.2: Key Distribution Phase.

Group Verification Phase.

After the key distribution phase is complete, and the supplier makes corresponding entries in the database, the groups of tagged items are shipped to their respective locations. When a group reaches a particular reader in the supply chain process, the GV phase is initiated. The complete protocol is shown in Figure 6.3 and is as follows:

1. The reader R_h initiates an EPCC1G2 standard UHF protocol.
2. The first tag (whose slot counter is zero, see [48]) HVT_1 responds showing that it is employing hierarchical verification belonging to group G_j.
3. The reader sends this group identifier G_j and its own identifier R_h to server S.
4. The server S generates a random nonce r_j and sends it to the reader R_h along with the total number of tags d in group G_j.
5. The reader R_h then forwards $r_j + 1, r_j + 2, \cdots, r_j + d$ to the tags in succession.
6. Each tag computes its verification code depending on which random value it received and sends it to reader R_h. Tag HVT_i belonging to group G_j computes its TVC_i as follows:

$$TVC_i = EPC_i \oplus F_{KS_{tj}}(r_j + i). \tag{6.1}$$

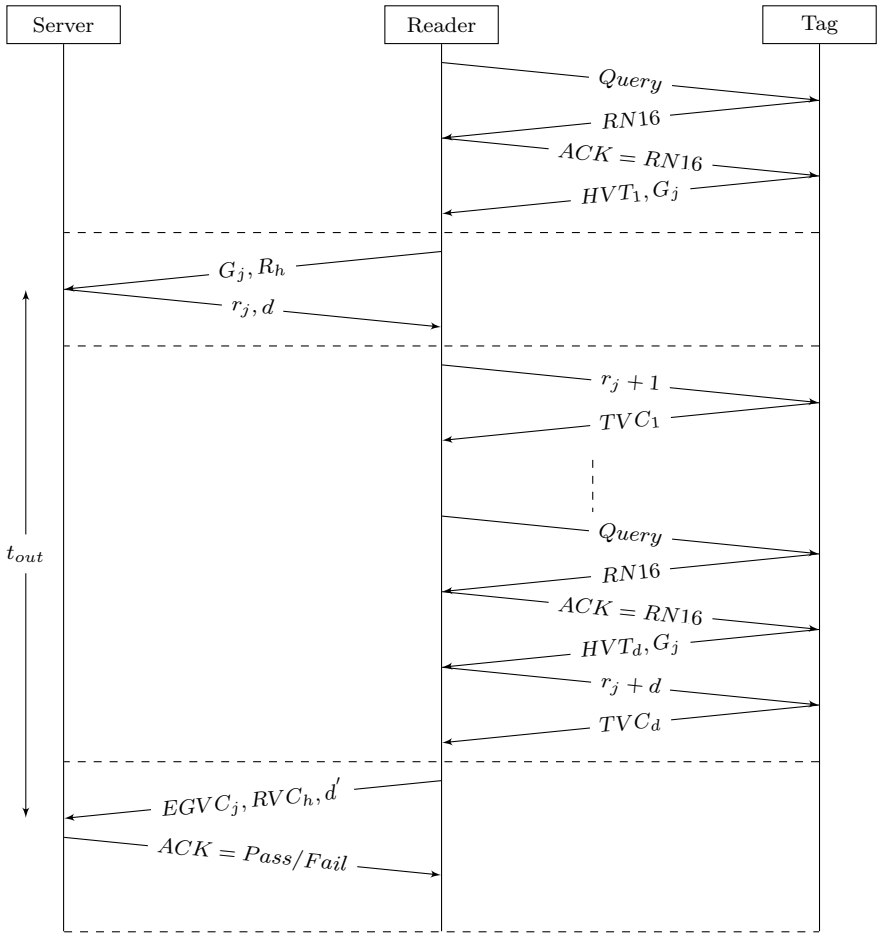

Figure 6.3: Group Verification Protocol.

7. The reader R_h computes a GVC by XOR with the previous TVC every time it receives a new TVC, until all d tags have responded:

$$GVC_j = TVC_i \oplus ... \oplus TVC_d. \tag{6.2}$$

8. The reader R_h computes a session key as:

$$SK_{hj} = KDF(KM_r, R_h, G_j). \tag{6.3}$$

9. The reader R_h encrypts r_j to compute RVC_h and GVC_j using SK_{hj}, and sends it as $EGVC_j$ along with the total number of tags $d^{'}$ that it read within time t_{out} to the server S. RVC_h and $EGVC_j$ are computed as follows:

$$RVC_h = E_{SK_{hj}}(r_j), \tag{6.4}$$

$$EGVC_j = E_{SK_{hj}}(GVC_j). \tag{6.5}$$

10. The use of XOR in calculation of TVC and GVC offers performance efficiency where the server can calculate GVC offline for comparison with received value without requiring a continuous input from the reader irrespective of which value of random number is sent to a particular tag. This also keeps the task of the reader simple and transparent while sending incremented value of the random numbers where it does not have to keep a track of these values and their recipient tags.

11. The server first checks the legitimacy of reader R_h by decrypting RVC_h. The server S next checks that reader R_h has read all the tags from the value of $d^{'}$ (to determine any missing/dummy tags). The server S finally decrypts the $EGVC_j$ sent by reader R_h to check whether GVC_j is correct:

if $D_{SK_{hj}}(RVC_h) == r_j$ **then**
  R_h is legitimate,
  Check:
  if $d^{'} == d$ **then**
    All tags have been read,
    Check:
    if $D_{SK_{hj}}(EGVC_j) == GVC_j$ **then**
      G_j is successfully authenticated,
      Send $ACK = Pass$ to R_h.
    end if
  else
    GV has failed,
    Send $ACK = Fail$ to R_h.
  end if
else
  R_h is not legitimate,
  Abandon the protocol.
end if

If the final $ACK = Pass$, this shows that group G_j has passed the GV phase successfully. A corresponding entry is made in the database for the group G_j scanned by

reader R_h, which also helps in future transactions with this particular reader in terms of trust level. The construction of the GV layer is as shown in the example given in Figure 6.4, where the group G_1 consisting of four tags is being scanned by the reader R_5.

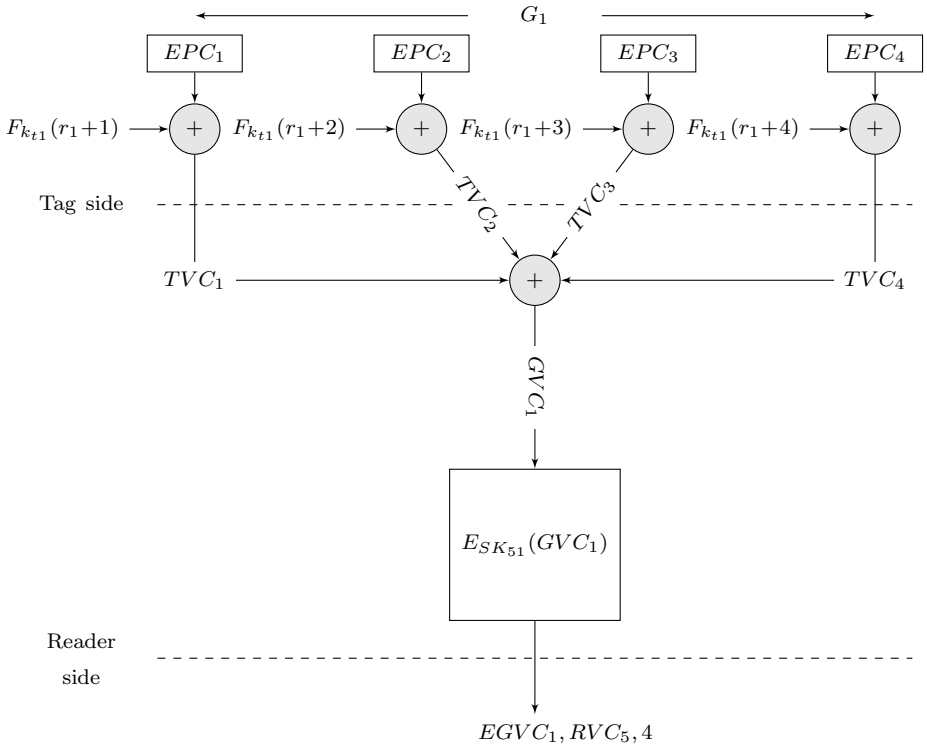

Figure 6.4: Construction of the GV Layer.

Product Verification Phase.

When $ACK = Fail$ is sent to reader R_h, this shows that the GV layer has not verified the authenticity of the group. In this case the PV phase is initiated as shown in Figure 6.5. PV is carried out as follows:

1. The reader R_h sends the tag identifiers HVTs to the server.
2. The server S generates a random nonce r for each tag as a challenge.
3. The reader R_h forwards this challenge to the corresponding tag, receives the TVC and forwards it back to the server S.
4. The server S verifies the legitimacy of an individual tag as follows:
   if $TVC_i == EPC_i \oplus F_{KS_{tj}}(r_i)$ **then**

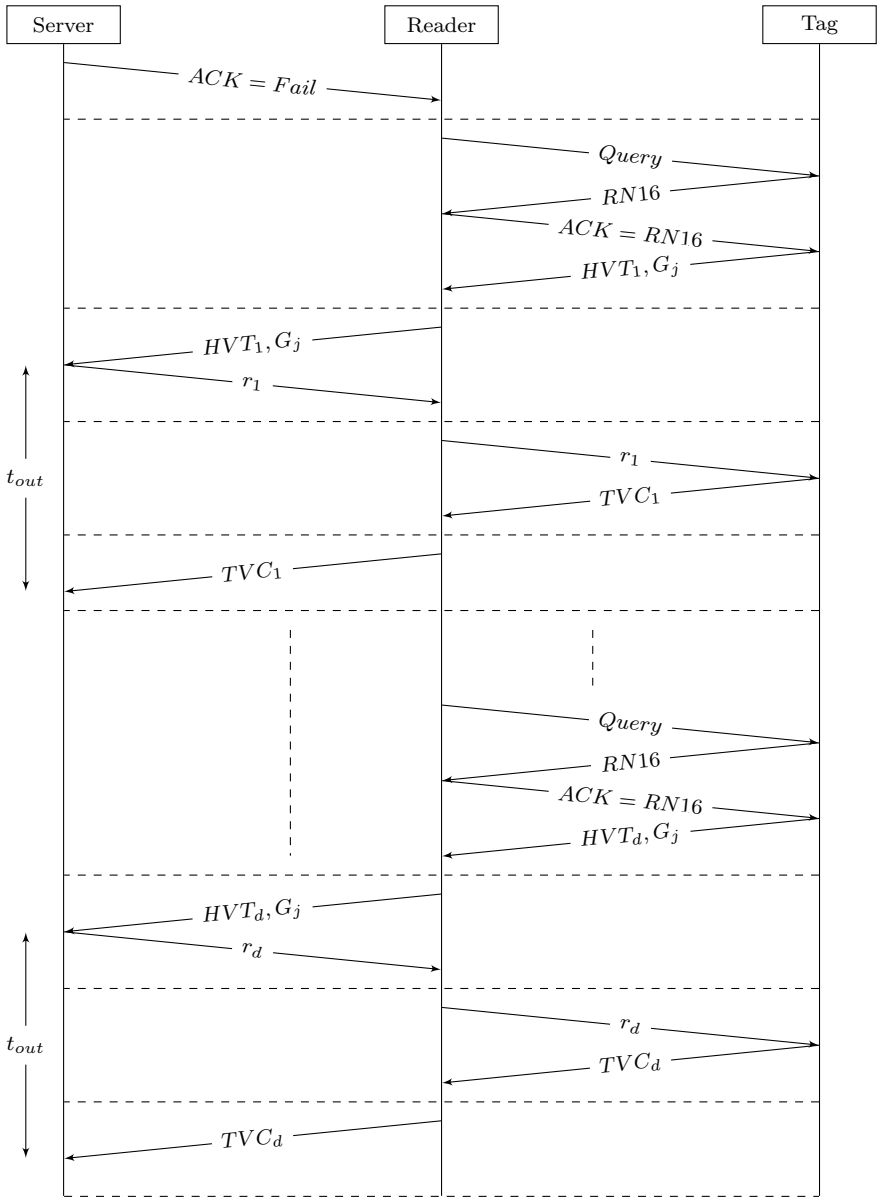

Figure 6.5: Product Verification Protocol.

Tag with identifier HVT_i is a genuine tag,

else

Tag with identifier HVT_i is a counterfeit tag.

end if

5. At the end of this protocol, the server S is able to identify the counterfeit tags as well as missing/dummy tags.

Track and Trace Phase.

In the EPCC1G2 standard, the unique and secret identifier EPC is used to track and trace the tag's movement throughout the supply chain. We give an example in Figure 6.6 to explain the TT phase. Suppose that a particular item travels in a group

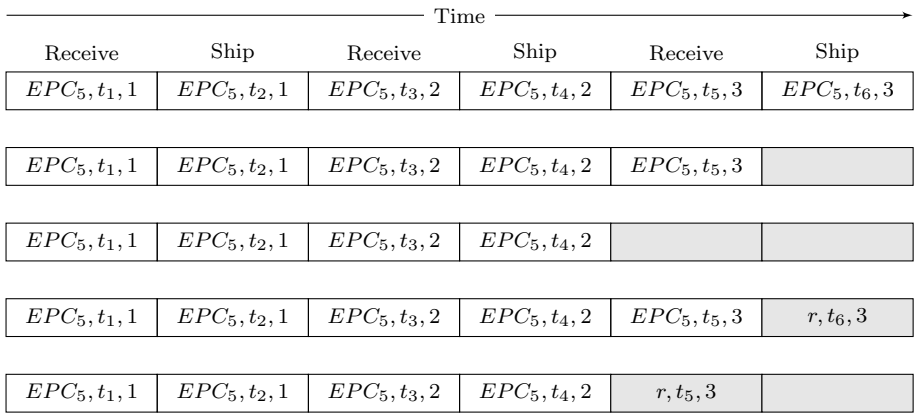

Figure 6.6: Track and Trace Example.

through three different companies (Company 1, 2 and 3) before reaching its retailer. The server S maintains its receiving and shipping record at each company. The entry $EPC_5, t_1, 1$ shows that the item with tag identifier EPC_5 was received at time t_1 by Company 1. A track and trace operation results in one of the following:

- **Case 1: No anomaly.** The first record in Figure 6.6 shows an ideal case where a particular item EPC_5 is successfully shipped to the retailer.
- **Case 2: Missing item within company.** The second record shows that Company 3 received the item at t_5 but never shipped it to the retailer.
- **Case 3: Missing item en-route.** The third record shows that the item was shipped by Company 2 but was never received by Company 3.
- **Case 4: Counterfeit item within company.** The fourth record shows that Company 3 received the original authentic item at t_5 but shipped a suspected counterfeit to the retailer.
- **Case 5: Counterfeit item en-route.** The last record shows that Company 2 shipped the original authentic item at t_4 but the item received by Company 3 is a suspected counterfeit.

6.4 Analysis

We now carry out an analysis of our proposed anti-counterfeiting mechanism as to whether it achieves the desired goals of Section 6.3.1. A formal analysis of our proposed scheme is also presented in Appendix D.

6.4.1 Anti-cloning

As discussed in Section 6.2.1, if a tag transmits its static secret identity EPC in the clear then it can easily be copied. This unique identity of the tag is linked with its associated information, which potentially includes value, composition and other useful supplier-related data. In the proposed mechanism the tag hides this secret static identity in its verification code. The tag thus transmits its verification code which appears to be random data. Thus an adversary cannot make a copy of the secret static identity from a genuine tag. However the adversary can copy the public identifier HVT of a tag to a counterfeit tag, but this counterfeit tag will not be able to reproduce the correct verification code and thus will fail the legitimacy verification. We discuss the following scenarios:

- **Adversary learns EPC.** If an adversary learns about the static secret identity EPC of a genuine tag and wants to clone it, the former still has to generate a correct product verification code to go undetected. The TVC is calculated as follows:

$$TVC = EPC \oplus F_{KS}(r). \tag{6.6}$$

  From (6.6), a cloned tag has to calculate the correct $F_{KS}(r)$ after receiving r from a reader. This requires a complexity of $O(2^{128})$, since Hummingbird-2 uses 128-bit key [34].

- **Adversary learns KS.** If an adversary learns about the secret key KS of a group of tags and wants to clone its tags, the former has to generate a correct product verification code to go undetected. From (6.6), a cloned tag has to know the EPC of the genuine tag. This requires a complexity of $O(2^{96})$, since a tag uses a 96-bit EPC [48].

We therefore suggest storing these values of EPC and KS in a secure memory location (assuming the tag is tamper-resistant).

6.4.2 Anti-spoofing

As discussed in Sections 6.2.3 and 6.2.4, static secret identities can be replayed by a counterfeit tag in order to spoof as a genuine tag. In the proposed mechanism, the secret information which is transmitted for verification changes in every transaction because of the use of a fresh random nonce that is generated by the server. An adversary can thus not replay this secret information during a later transaction because the use of a new nonce will result in verification failure. The only possibility for an adversary is to steal a genuine tag and then relay its verification code. However, our scheme also provides protection against missing/stolen item as explained subsequently.

6.4.3 Anti-theft

The proposed mechanism employs a layered approach in order to detect stolen/missing products. As described in Section 6.3.2, this layered approach can be used to identify the exact cause and location of any anomaly since the final Track and Trace layer provides the complete shipping and receiving record of the identified stolen/missing item. After tracing the root cause of the anomaly, suitable processes can be undertaken to hold the responsible parties accountable. Appropriate countermeasures can then be applied in order to prevent this anomaly from occurring during future transactions.

We note that a smart adversary can prevent this detection by relaying the genuine verification code from a stolen/missing item that is not physically present in the vicinity of the scan range. To counter such an adversary, we have employed a time-out clock within our scheme. The server pre-computes this time, depending on the number of tags involved in the scheme. The server then expects the reader to answer back within that specified time. If the reader delays its response, this is an indication of a potential relay attack. The server can thus ask the respective reader for a physical check for the completeness of this group and makes a corresponding entry for this anomaly in its database.

6.4.4 Scalability

In supply chain environments, tags travel for most of their journey in groups. These groups can be large, medium or even consist of a single item, depending on their size, value, and application. Sometime these groups change in their size en-route from manufacturer to end-user. Our proposed group verification layer can verify any group irrespective of its size and, during the product verification layer, the legitimacy of a single tagged item is checked. Therefore, our scheme scales well from large groups, through to medium and small groups, and even to standalone items.

6.4.5 Conformance to Standard

Our mechanism is proposed for EPCC1G2 compliant tags. The conformance with the standard's operation is as follows:

- **Select.** There is no change in the standard select operation which is used for selecting a tag population.
- **Inventory.** We suggest a two way challenge-response protocol to verify the authenticity of the tag in addition to the standard inventory round.
- **Access.** We do not suggest any change in the access operation which is used to read from/write to and other operations related to tag's memory.

The following are the additional requirements to be incorporated in the standard's functionality:

- **Storage.** The tags need to store a secret key KS, a public identifier HVT (both require 3 GE/bit for long term storage) and a group identifier G (8 GE/bit for temporary storage) in addition to data specified in the standard.
- **Computation.** The tags have to perform encryption to generate TVC. The designers of the standard have already proposed an encryption alogrithm [34] to be incorporated into EPCC1G2 compliant tags [28].
- **Communication.** The proposed mechanism uses the standard UHF Air Interface Protocol as specified in [48]. The two additional messages are sent to carry out a challenge-response protocol to verify the authenticity of the tag. If a tag is not employing hierarchical verification, it can be read as per the existing standard.

6.4.6 Efficient Key Management

The proposed mechanism avoids some of the scalability problems of using symmetric cryptography by providing all tags belonging to a specific group with a unique key. Since the number of groups is much smaller than the number of tags, it is comparatively easy to manage the keys in the database. Additionally, all readers involved in the supply chain management system are only given one master key. By reducing the overall number of keys in the system, the key management is considerably more efficient than schemes with a unique key for each tag as mentioned in [2].

6.4.7 Good Throughput

The layered approach deployed by the proposed mechanism is partly designed to reduce the likelihood of potential bottlenecks arising from readers having to stay online with the server during authentication and verification, and having to regularly interact with the database. By first deploying relatively lightweight group-level checks we avoid bottlenecks in the top layer of the hierarchical verification process. Overall performance decreases in the lower layers, but these are only activated if anomalies are detected during the group-level checking. In this way a reasonable throughput can be expected for the system.

6.4.8 Economic Analysis

Please note that this analysis is informal and based on assumptions and rough estimates and may not be taken as a basis for making business decisions. As discussed earlier in Section 6.1, around 5-7% of world trade is composed of counterfeit goods according to estimates by the Counterfeiting Intelligence Bureau (CIB) of the International Chamber of Commerce (ICC) [58]. We carry out a rough estimate of the economic impact of our proposed scheme in this section. We make the following assumptions:

- The additional cost of key generation, writing into respective tags, cryptographic operations in the tag, etc. is included in the cost of an EPCC1G2 compliant tag.
- The 5-7% counterfeit items only cause losses to the sales revenue. Other losses including decline in sales due to reputation loss and associated impacts are not considered for simplicity.

We now use notation as shown in Table 6.2. We want to figure out what will be the ideal value of Δ which shall provide us with a break even price of using RFID tags.

$$n \cdot (\Delta - c) = n \cdot \Delta \cdot (1 - p),$$
$$c = \Delta \cdot p.$$

c for an EPCC1G2 varies from different vendors. SmartCode offers it at a price of \$0.05 apiece in volume of 100 million or more [122]. Suppose including all associated cost as earlier discussed, c is assumed to be \$0.1. p is assumed to be 0.07 in worst case scenario. Therefore if our $\Delta >$ \$1.43 we shall gain profit per item using our suggested scheme.

Table 6.2: Notation used in Economic Analysis of Hierarchical Verification Scheme

Notation	**Description**
x	The production cost per unit.
y	The sale price per unit.
$\Delta = y - x$	The profit per unit.
n	The market demand over some fixed period.
$n \cdot \Delta$	Ideal profit for original manufacturer.
p	The percentage of counterfeits in the market.
$n(1-p)$	Products sold by original manufacturer.
np	Counterfeit items from other suppliers in the market.
$n \cdot \Delta \cdot (1-p)$	Actual profit by the manufacturer.
c	Unit cost of RFID tag usage on a product.
$n \cdot (\Delta - c)$	Profit by manufacturer considering counterfeits are eliminated.

6.5 Summary

In this chapter, we have proposed a hierarchical anti-counterfeiting mechanism which uses three layers of verification to determine the legitimacy of a tagged item. This mechanism is designed for EPCC1G2 compliant tags used in supply chain management, where counterfeit items present a very serious threat. This threat is countered using dynamic verification codes generated using symmetric cryptography. Our model detects the stolen/missing items, provides efficient key management, avoids bottlenecks, and is scalable to the complete lifecycle of tags in the supply chain. The layered approach also potentially lends itself to deployment in schemes based on other standards.

Chapter 7

A Customer Level Counterfeit Detection Scheme

In this chapter, a counterfeit detection scheme for tagged products is proposed at customer level. Types of counterfeits which reach to individual customers is introduced in Section 7.1. Section 7.2 introduces NFC technology and the different types of NFC tags. This is followed by an overview of the EPC network and its application in supply chain management. Section 7.3 analyzes existing work with a detailed description of the scheme proposed by Alex Arbit et al [2]. Section 7.4 describes the proposed scheme, which overcomes the weaknesses in the existing work. A detailed analysis of suggested proposal is presented in Section 7.5.

7.1 Introduction

As previously explained in Section 6.1, counterfeit products are one of the major threats to modern commerce. We have proposed a hierarchical anti-counterfeit mechanism in Chapter 6 which caters for counterfeits in supply chain management. Whereas in this chapter, we propose a scheme to detect counterfeit products by individual customers using *electronic product code* (EPC) and *near field communication* (NFC) tags. Counterfeit products reaching individual customers can be classified into four categories [8].

1. The first category consists of those products that are inexpensive, lower quality and may lack original packaging. This category is often called 'knockoff'. These products are being sold as counterfeits and the customer is aware of it.

2. In the second category of counterfeit, a genuine product is reverse engineered

and identical copies are sold as the genuine product. It is hard for a customer to differentiate between a genuine and a counterfeit product. This category is meant to deceive the customer.

3. These are the products that are produced by an outsourced manufacturer without knowledge of the original owner. For example, an outsourced manufacturer manufactures further products after termination of its contract with the original owner without notifying the original owner.

4. These are genuine products that do not meet the manufacturer's standards but are not labeled as faulty.

One of the major outlets for counterfeit products is Internet e-commerce, where the customer has no means of authenticating a product before delivery. Even after delivery, the customer has very limited resources to determine the legitimacy of a product. Auction websites, such as eBay, have further expanded the market of counterfeit products. For example, test purchases from 300,000 Dior products and 150,000 Vuitton items offered on eBay during 2006 found 90% conterfeits [95]. Tiffany & Co. purchased 186 random items from eBay and found only 5% to be genuine [19].

These circumstances call for mechanisms to fight counterfeiting at customer level. Analysis shows that money spent in this way prevents a much greater loss from counterfeit goods. According to the U.S. Chamber of Commerce, $5 is gained for every $1 invested in this battle [137].

7.1.1 Our Contribution

We focus our work on detecting counterfeits that fall into the categories 2 and 3 mentioned in Section 7.1. Category 1 counterfeits are not a major concern as customers are aware of the fact that the products they are buying are counterfeits. The loss in sales of the original product is also negligible as very few genuine goods purchasers would purchase a knock off [8]. Category 4 counterfeits can be restricted by enforcing an efficient quality control measure by the genuine product owner. Categories 2 and 3 are most critical as not only is the customer unaware of the illegitimacy of the product, but also the genuine owner has no, or minimal, control over the production, marketing and selling of such products. Our scheme helps to detect counterfeit products at the customer level pertaining to category 2 and 3 products, thus providing an efficient tool to detect counterfeits.

In this chapter, we analyze the anti-counterfeiting scheme which Alex Arbit *et al.* proposed in [2] and highlight a few of its short-comings. The main drawback of

their scheme is its semi-offline structure, which render it incapable of authenticating a product at customer level despite using public key cryptography.

We revise and extend their work in two main ways. Firstly, we restore the EPC tag to the original standard rather than using the modified EPC tag in the Alex Arbit *et al.* scheme. This resolves any modification-related problems in the existing EPC framework. Secondly, we supplement the EPC tag with an NFC tag which can perform the necessary computations that were not within the capability of the EPC tag. The main advantage of being offline is that a customer can authenticate a product without any online communication with the supplier's database. We believe that our offline product authentication at customer level is an efficient anti-counterfeiting tool. This scheme not only helps customers authenticate a product, but any verifier such as a law enforcement agency can also use this scheme to detect counterfeit products.

We resolve the problem of provisioning of a UHF RFID reader for product authentication to every customer by using an NFC tag for the EPC tag. Such NFC technology is now available on mobile phones and so a customer's mobile phone can act like an RFID reader to read the EPC. Since our scheme is completely offline, the customer is able to distinguish between a legitimate and a counterfeit product by using his mobile phone without accessing the supplier's database.

We also resolve the issue of trust in the reader for an offline framework. In the work of Alex Arbit *et al* [2], the reader is a secure module storing a verification key and a decryption key, as noted earlier in this section. These keys cannot be stored on any reader that is not trusted by the supplier. Although the customer's mobile phone is not trusted by the supplier, this issue can be addressed by using certificate-based public key cryptography, thereby all but eliminating any key storage requirement on the reader side. In many cases, the NFC tag can also be accessed and authenticated during product distribution without having to resort to the greater read range of the EPC tag.

This work is published in [125] and conducted jointly with Qasim Saeed and Colin Walter (ISG-RHUL).

7.2 RFID Technologies

In this section we introduce the two different classes of RFID technology that are related to our scheme.

7.2.1 Near Field Communication

NFC is a wireless technology that operates at a distance of less than about 4 cm. This technology is compatible with contactless smart cards based on the ISO/IEC 14443 standard [151]. The frequency of operation falls in the HF band operating at 13.56 MHz. The limited 4cm range means that their use in supply chain management can be problematic. Access to tags embedded in products which are packaged in rigid expanded polystyrene foam requires precise location markers printed on the boxes, and the ability to place a reader on that location. This may not be possible in a warehouse.

An NFC link is established between a tag and a reader on a single touch. This makes it a user friendly technology, where no input is required from a user apart from touching the tag to the reader. NFC has three modes of operation enabling a variety of applications: peer-to-peer mode, read/write mode and emulation mode [91]. The latest mobile phones are equipped with NFC technology [101].

We only focus on the read/write mode of operation as only this mode is applicable to our proposed scheme. In read/write mode, an NFC device acts as an RFID reader/writer to read or write an NFC tag. In order to maintain the interoperability of NFC devices and tags, the NFC Forum (a forum to standardize the applications related to NFC) [148] has specified four different types of tags [99]: Type-1, Type-2, Type-3 and Type4. Type-1 has the least resources in terms of computational power and memory, whereas Type-4 is much more powerful and contains a cryptographic processor.

7.2.2 EPC in the Supply Chain

We have previously explained EPC tags in Section 2.3.1 and its deployment in supply chain management system in Section 5.1. We present the following brief summary of EPC tags in supply chain. The EPCglobal Class-1 Gen-2 standard [48] specifies low-cost UHF tags which operate in the frequency range of 860-960 MHz and have a read range of 2-10 metres. This longer range makes UHF tags more easily read in containers and warehouses than is the case with NFC tags. Electronic Product Code (EPC) tags are typically deployed in supply chain management systems for automated inventory checks. The EPC is a 96-bit identifier stored in the EPC tag which helps to identify each tagged product uniquely.

In this work, EPC tags are used in conjunction with NFC tags to cover the complete journey of a tagged product from manufacturer to end-user/customer. A hierarchical anti-counterfeit mechanism using EPC tags is discussed in Chapter 6 while this proposal provides counterfeit detection at customer level.

7.3 Existing Work

EPC tags can be used as a tool for anti-counterfeiting [130] as already discussed in detail in Section 6.2. Using unique serial numbers (refer to Section 6.2.1), these numbers should be distributed to individual customers to counter counterfeiting. However, there are many overhead and also some privacy concerns associated with this approach. Therefore this approach is not suitable for customer-level anti-counterfeit mechanism.

Another anti-counterfeiting approach is based on cryptography (refer to Section 6.2.2). In this approach, each tag contains a secret value, knowledge of which is established by the reader in an authentication proof. This approach may be based on symmetric key or asymmetric key cryptography. The BRIDGE project [83] analyzed various anti-counterfeiting approaches based on RFID tags. This work analyzed the secure distribution and management of secret keys in a symmetric key anti-counterfeiting framework, and showed that it results in ten times more communication and computational overheads than in a track-and-trace anti-counterfeiting system. Anti-counterfeiting approach based on cryptography can be categorized into two main categories as also described in Section 5.1; *Off-line* and *On-line*. In a supply chain, it is very unlikely that the login credentials are provided to a customer to access the database in order to verify the authenticity a product. This makes former approach more suitable for product authentication at the customer level.

As observed earlier, one of the major factors in the upsurge in counterfeit products is online shopping. With the advancement in Internet technology, the volume of online shopping is growing rapidly. It is not feasible at present to tailor any symmetric key approach for product authentication to online shopping. The reason is obvious: a customer receiving a product through online shopping does not possess an RFID reader to communicate with the tag attached to the product. Even in the very unlikely scenario where a customer possesses an RFID compatible reader, the supplier will have to provide login credentials to access the database. This situation is far from practical. Thus, product authentication at the customer level remains an open challenge, especially for the Internet shopping framework.

In contrast to the symmetric key approach, public key cryptography can also be used to authenticate a product. Considering the limitations of the symmetric key approach described above, the case for public key cryptography in product authentication is thus very strong. The main restriction in using on RFID tags, such as EPC tags, is the limited computational and storage capabilities of these tags. Work to reduce the computational overheads in public key cryptography is also in progress and various lightweight public key cryptosystems are being designed. The CRYPTOGPS is

a light-weight public-key cryptosystem and can be implemented in around 2800 GE (Gate Equivalent) with a processing time of around 720 cycles [15, 116]. The Rabin cryptosystem is the first to be implemented in a wireless sensor network in [46]. It takes about 16,700 GEs to implement 512-bit encryption. This cryptosystem is considered unsuitable for resource-constrained RFID tags.

Recently, Alex Arbit *et al.* presented a working implementation of a PKC-based anti-counterfeiting scheme [2]. They selected WIPR, an ultra-low-power public key cryptosystem developed by Oren and Feldhofer [104]. WIPR is a lightweight version of 1024-bit Rabin encryption [46], with a minimal hardware footprint of under 4700 gates. The scheme presented by Alex Arbit *et al.* is semi-offline, where the verification and decryption keys are dispatched to the reader using a smart card and the reader is considered as a secure module for storing these keys.

7.3.1 Alex Arbit *et al* Anti-Counterfeiting Scheme

Alex Arbit *et al* proposed an anti-counterfeit scheme based on EPC tags and public key cryptography [2]. Their scheme is described in Figure 7.1. The figure represents the various entities involved in the anti-counterfeiting scheme. The scheme consists of the following sequence of operations.

- **Step 1:** The scheme is initiated by the *tag integrator* (TI), who wishes to deploy anti-counterfeiting technology in EPC tags. It creates two public-private key pairs: 1) a *private signing key* K_S, together with its *pubic verification key* K_V, 2) and a *public encryption key* K_E with its *private decryption key* K_D. The signing key K_S is never disclosed to any entity of the scheme. The TI generates a list of *tag identifiers* (TIDs) and signs each TID with the key K_S. He then sends the list of signed TIDs to the tag manufacturer along with the encryption key K_E. Since the tag manufacturer lacks K_S, he is unable to generate arbitrary signed TIDs, thus ensuring the integrity of the TIDs.

- **Step 2:** The tag manufacturer produces and deploys the tags, each with an individually signed TID from the list along with the public key K_E.

- **Step 3:** The reader receives K_D and K_V from the TI. Once these keys are delivered to the reader, the system can operate in an offline mode. The reader then carries out a challenge-response protocol to determine that the tag possesses a valid, signed TID.

This is a semi-offline scheme as it requires an initial key distribution mechanism to distribute keys to readers through some secure channel. The authors suggests dis-

tributing keys through a secure module such as a smart card.

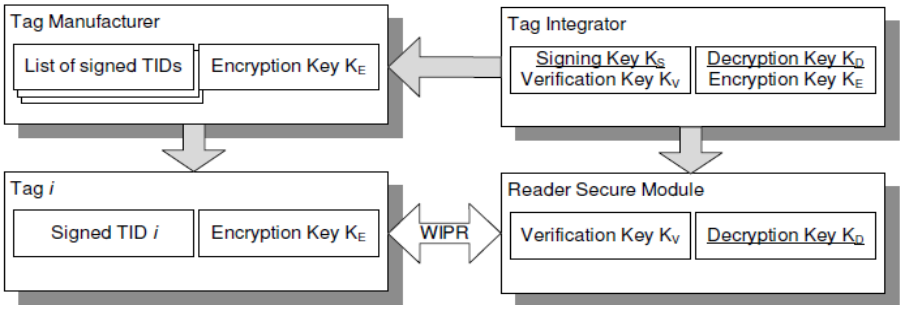

Figure 7.1: Alex *et al* Anti-Counterfeiting Scheme

7.3.2 Analysis of Existing Scheme

There are several weaknesses highlighted in this scheme.

- The scheme is semi-offline where the reader stores K_V and K_D. This puts a limit on its utility for product authentication at customer level, as it is infeasible to communicate K_D to each customer.

- K_V and K_D have to be delivered to a reader through some secure channel such as a smart card. Since the same set of keys are distributed to each reader, this results in a single point of failure where the loss of a single smart card will compromise the entire system. Moreover, if a single retailer is dishonest, he can break the entire system as all the readers use the same set of keys K_V and K_D.

- The authors have not discussed the storage location and accessibility of K_E inside an EPC tag. If K_E is stored at an accessible location, an attacker can make a successful counterfeit tag by simply copying all the content of the EPC tag, including K_E, to a counterfeit tag. If K_E is stored at some inaccessible location inside the EPC tag, it can prevent tag cloning, but still the scheme is prone to single point of failure. Since K_E is identical in each tag, it only needs an adversary to attack a single tag to compromise the entire system.

- **Bypass Attack.** The scheme is prone to a "Bypass" attack where the anti-counterfeiting protocol is circumvented in a counterfeit tag in the following way. The scheme is designed to handle both WIPR-modified and standard EPC tags. During the handshake protocol between a reader and an EPC tag, the tag responds with an indication of being WIPR modified or not. This is achieved by the modified tag sending a special WIPR EPC message to the reader instead

of the actual EPC value according to the standard (see Figure 4 in [2]). The special WIPR message acts as a flag to the reader to execute the anti-counterfeit protocol.

The scheme does not provide integrity protection to the special WIPR EPC message contents and so alterations to this message may not be detected. An attacker just needs to replace the message with the actual EPC value in the counterfeit tag, thereby making the tag claim to follow standard EPC protocol. On receipt of the actual EPC value from a counterfeit tag, the reader does not execute the anti-counterfeiting protocol, instead assuming the tag to be unmodified as the flag (the special WIPR message) is not received from the tag. Thus, the anti-counterfeit protocol is bypassed and the counterfeit tag remains undetected. Of course, if the reader knows the TID belongs to a tag which follows the WIPR modified protocol, then the counterfeit should be detected.

7.4 Proposed scheme

In this section, we propose our counterfeit detection scheme that uses RFID technology. This scheme is a modified version of the Alex Arbit *et al.* [2] scheme. We use an NFC tag in addition to EPC tag, thereby providing a product authentication mechanism at the customer level.

NFC technology is used mainly for two reasons. Firstly, this technology can support public key cryptography on tags and, secondly, it is available on mobile phones enabling them to act as RFID readers. The former supports our scheme in an offline mode where a connection to the supplier's database is no longer required. The latter helps extend the authentication scheme to the customer, where a customer uses his mobile phone to authenticate a product.

7.4.1 Initialization Phase

Our proposed anti-counterfeit scheme is executed in two phases; the first, namely initialization, being illustrated in Figure 7.2. This phase is initiated from the production line where a serial number and an EPC are allocated to the product. The serial number, EPC and the product specification are communicated to TI. Meanwhile, the product is dispatched to the tag embedding department.

On receipt of the information from the production line, the TI generates a public/private key pair (K_p, K_s). This pair is unique for each tagged product item. The TI must be a secure platform as it is responsible for the generation of anti-counterfeit keys. It stores the EPC in an EPC tag and forms a string S_1 defined by

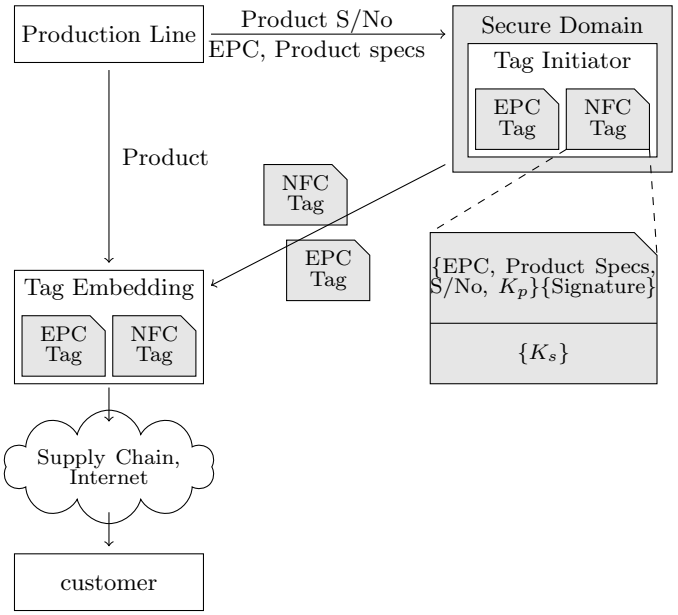

Figure 7.2: Initialization Phase of Proposed Scheme.

$$S_1 = \text{EPC} \| \text{Product Specs} \| \text{S/N} \| K_p.$$

The TI digitally signs this string S_1 with its signing key K_{sign} and stores the string along with its signature on the NFC tag. The signature on the tag is stored as a 'Signature Record' according to the NFC Forum's Signature Record Type Definition [100]. According to this specification, the signature record consists of a digital signature along with a digital certificate containing the corresponding verification key K_{ver}. S_1 and its signature are stored at a memory location accessible to any NFC reader. However, the TI also stores the secret key K_s inside the tag but at a secure location. This location of K_s is only accessible to the tag's processor and is therefore inaccessible to a reader. The corresponding public key K_p is part of S_1, and therefore accessible to any NFC reader. After storing the relevant information on both tags, the TI configures both tags as write protected and dispatches them to the Tag Embedding department.

On receipt of the tags from the TI, the tag embedding department embeds both tags on the product. Since the tags are physically embedded we shall assume that any attempt to remove the tags will destroy them. After embedding the tags, the products are shipped to the supply line, from whence they may be delivered to a department store or direct to a customer through online shopping.

7.4.2 Verification Phase

This phase is executed by the verifier on receipt of the product. Since this is an offline scheme, the verifier does not require any connection to the supplier's database. Therefore the verifier may be a customer, a warehouse employee, a member of law enforcement or, indeed, any individual wishing to authenticate the product. The verification phase is executed in two phases as shown in Figure 7.3. The first is visual and the second is cryptographic. The visual verification process is executed as follows:

- The customer checks the claimed identity of the product itself and the integrity of the tag which should be bound to the product item in a tamper-evident manner.
- The verifier places his mobile phone on the NFC tag to read its contents. The accessible data on the NFC tag (string S_1 and corresponding signature) is communicated to the mobile phone.
- The mobile phone verifies the signature. A successful verification is an indication that the string S_1 is legitimate.
- The mobile phone displays the product specification and its serial number to the customer.
- The customer checks the two product descriptions match each other.

In the case of a successful visual verification, the customer should initiate the second phase of product verification, which is a cryptographic challenge-response protocol:

- The mobile phone sends a random challenge r to the NFC tag.
- The tag signs r with the secret key K_s and returns the result $sign(r)$ to the mobile phone.
- The mobile phone verifies the signature using the corresponding verification key K_p which it knows from S_1.

A successful verification is a strong indication of a genuine product, as a counterfeit tag lacks the signing key K_s and so cannot compute a valid signature on r.

7.5 Analysis

In this section, we analyze the proposed scheme from various angles. Our scheme addresses category 2 and 3 of counterfeits as mentioned in Section 7.1.1. Categories 1 and 4 are not a focus of our work since, in the former case, the products are being

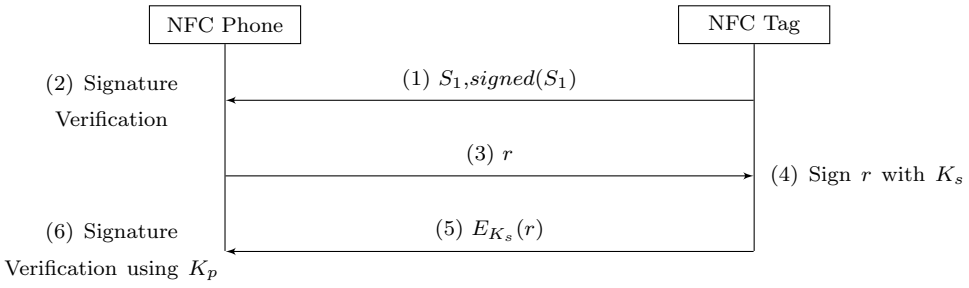

Figure 7.3: Product Verification Phase at Customer Level.

identified by the customers as counterfeits and, in the latter, can be countered with an appropriate quality control. Categories 2 and 3 are critical as the customer is not aware of the legitimacy of the product. Since our scheme is designed to detect counterfeits at the customer level, it provides a tool for customers to determine the legitimacy of a product. We also carried out a formal analysis of suggested scheme as shown in Appendix E.

7.5.1 Detection of Cat 2 Counterfeits

In the case of category 2 counterfeits where the original product is reverse engineered, the NFC tag attached to the original product cannot be reverse engineered, i.e., the secret data on the NFC tag cannot be copied as explained in Section 7.5.4. A customer can therefore determine the illegitimacy of a reverse-engineered product by the unsuccessful verification of the data on the NFC tag.

7.5.2 Detection of Cat 3 Counterfeits

In our scheme, the TI is responsible for generating and storing the secret keys on the NFC tags. The tags are then embedded on the product by another department termed the 'Tag Embedding Department'. In the case of out-sourced manufacturers, the product manufacturing and tag embedding are done by the out-sourced manufacturer. The TI remains a part of the genuine owner. The genuine owner provides NFC and EPC tags to the out-sourced manufacturer only in same quantity as specified in the contract. If an out-sourced manufacturer is dishonest and produces more than the quantity mentioned in the contract (category 3 counterfeits), he will have to produce the product either without the NFC tag or with a fake NFC tag. This counterfeit product is then detected by the customer because making a fake NFC tag is not feasible (explained in Section 7.5.4). Thus, our scheme helps in the detection of category 3 counterfeits at customer level.

7.5.3 Justification for Two RFID Tags

We use two types of tags in our scheme, an EPC tag and an NFC tag. Although both are RFID tags, they have very different characteristics. The main difference is the operating frequency: EPC tags operate at 860-960 MHz whereas NFC tags operate at 13.56 MHz frequency band. The range is consequently different in the two tags. EPC tags can be read from 2 to 10 meters whereas NFC tags have a very short communication range of no more than approximately 4 centimeters. This property makes only the EPC tag suitable for countering counterfeits in the supply chain management as explained in Chapter 6. Since EPC tags are already deployed in the market for supply chain management, we use EPC tags in our scheme in order to maintain the backward compatibility and normal supply chain needs.

NFC tags are used because of two main requirements that cannot be fulfilled by EPC tags. Firstly, EPC tags are very resource constrained when compared to NFC tags: EPC tags have very limited computational power and much less memory, whereas NFC tags, specially NFC Type-4 tags, are much more powerful. Since our scheme is based on public key cryptography, where the tag has to perform extensive computation, we need a reasonably resourced tag. Secondly, our scheme needs to provide authentication at the customer level. Without an NFC tag, this would require every customer to be equipped with a UHF EPC tag reader, which is far from practical. The issue is resolved with the inclusion of the NFC tag, as the customer's mobile phone can act as a reader for the tag.

7.5.4 Security Analysis

In this section, we analyze our scheme from the security perspective. The goal of an attacker is to develop a cloned tag or a tag with a valid signature. To develop a cloned tag, the attacker must know the private key K_s of the original tag. This key is stored at an inaccessible location in the tag's memory and so it is normally secure from the attacker. The alternative solution open to an attacker with a cloned tag is to replace the legitimate public key K_p with the attacker's public key K'_p in S_1 and store the corresponding private key K'_s in the tag. However, this is not possible as the legitimate K_p is digitally signed (it is in a digital certificate) so that any alteration will invalidate the signature. Of course, the verifier must have a trusted source for the certificate's public key in order not to be duped.

In case an attacker spends time and money to reverse engineer a single tag and recover its private key K_s, it will not affect the entire system as the pair K_s, K_p is unique to each tag. The tags will have few counter-measures to side channel analysis,

which will be a significant threat in some markets. However, this will avoid a single point of failure as experienced in the Alex *et al* scheme.

Our scheme is resistant to the bypass attack. The existence of K_p in S_1 is an indication that the tag is equipped with the anti-cloning feature. This key can neither be removed nor altered as it is digitally signed. The user's application on the mobile phone, once it has detected K_p, will execute the anti-counterfeiting protocol, thereby resisting the bypass attack.

In addition to cryptographic authentication, our scheme also provides visual product authentication. After scanning the NFC tag, the product specification and product serial number is visually displayed on the user's mobile phone display. The user can visually check and verify the information from the product or product packaging. Needless to say, there are many other sources of compromise. For example, the NFC tag could just return a QR code which connects the customer's phone to the attacker's website and displays the expected protocol output and the verification data for the counterfeit product. Alternatively, the merchant may direct customers who lack the verification app to the attacker's website to download a compromised app that confirms the authenticity of any product.

The tags have to be tamper-evident. This is to ensure that they cannot be reused on counterfeit products. If the tag were to contain the URL for registering the product under the manufacturer's guarantee, customers could be encouraged by their app to register, the manufacturer could check its database for duplicate registrations that would flag a clone, and the manufacturer could advise the customer if there were such a problem.

One critical factor in securing the system is the physical location of the NFC tag in the product. This is an industry specific decision and requires careful consideration. It is assumed that the tags are physically embedded on the main assembly of the product and not on casing/packing or on any easily replaceable component of the product, very much in the same way as a watermark or hologram is an integral part of the item it is protecting. As in the latter case, an attacker just needs to place the tag embedded component from a legitimate product into the counterfeit product.

7.5.5 Economic Analysis

We use our existing analysis as discussed in Section 6.4.8 to determine the profit per unit which provides us with benefits of using this scheme. The percentage p of counterfeit products depends on various factors like brand, geographical location, in-store or on-line, etc. It is difficult to find an exact value of p for a specific brand as the counterfeit products of categories 2, 3 and 4 are indistinguishable. Fortunately, the

surveys mentioned in Section 7.1 regarding counterfeit products on eBay are only measuring a fraction of the total market for the goods in question – although this may change. Assuming the price of implementing RFID tags with the required infrastructure is \$2/unit, and assuming p as 7% in worst case scenario, (which is an estimate by the Counterfeiting Intelligence Bureau (CIB) of the International Chamber of Commerce (ICC)) [58]), our scheme is suitable for those businesses where the profit/unit Δ is greater than \$28.50, i.e., around \$30. This is a very rough estimate as it is based on very simple assumptions. Of course, with higher values of p, the profit/unit threshold at which the NFC RFID scheme becomes cost effective decreases. This means that it becomes suitable for more businesses.

7.6 Summary

This chapter presents an RFID based anti-counterfeiting scheme at the customer level. There are two main constraints related to this authentication level. Firstly, the individual customer cannot afford to keep an RFID reader to authenticate a product; and secondly, customers cannot be provided with access to the supplier's database because of intellectual property rights and communication overheads. We addressed both these constraints by using NFC technology: an NFC tag is used along with an EPC tag for customer level authentication on the reasonable assumption that most individuals will carry an NFC-enabled mobile phone in the near future. We provided a dual layer verification mechanism to a customer. In the first phase of verification, the product specifications are displayed to the customer on his mobile phone for visual verification of the actual product. After successful verification, a cryptographic challenge-response protocol is executed to authenticate the product. Our proposal is based on certificate-based public key cryptography and successfully detects the counterfeit products.

Part IV

Ownership Transfer in RFID Systems

Chapter 8

A Robust Ownership Transfer Scheme

This chapter provides a robust scheme of transferring ownership of a tag in RFID systems. The requirement of ownership and associated terminology are presented in Section 8.1. Section 8.2 carries out a literature survey of existing ownership transfer schemes, and highlights their weaknesses and limitations. Section 8.3 elaborates our proposed scheme for ownership transfer. Section 8.4 analyzes our proposed scheme with respect to the necessary properties.

8.1 Introduction

As discussed in Section 1.1 and 2.1, RFID tags are small microchips attached to physical objects such as medicine bottles, car keys or smart home appliances, etc. Tags bears a unique serial number linked to useful information related to the object, such as identification, manufacturer, ingredients, expiry date, lot numbers, location, environment and other sensitive data such as medical history, credit card numbers and biometric information.

The purpose of using an RFID system is to enable a user to interact with a tagged object remotely using a compatible reader. The interaction between a user and a tag normally involves reading the tag content. When an RFID tag is queried by a reader, the tag transmits a unique serial number which is linked to the information about the tagged product. If a tag transmits this serial number in the clear, anyone can eavesdrop this as the communication between a reader and a tag uses radio waves. This raises serious risks such as cloning a tag, impersonation, and tracking a tag holder, etc.

Therefore this information should be sent encrypted using a secret key shared with a tag-reading entity.

There are many scenarios where tagged objects are physically transferred between different users who require access to the tag content. Examples include transfer of a tagged file of a patient's medical history between hospitals and supply of products across different geographic locations, etc. This process requires an old user to transfer the shared secret key to a new user in order to enable interaction with the tag attached to the transferred object. However there are some security and privacy concerns associated with transfer of a secret key between different users. If this transfer only involves transferring the shared secret key, the old user could retain a copy of the secret key and thus jeopardize the new user's security and privacy by, for example, tracking location, tracing transactions, changing content, cloning or blocking the tag's functionality. Similarly a new user, while tracing back transactions of the tag with an old user, might learn the latter's previous location information. Therefore, there is a need for a secure process for such transfer.

8.1.1 Our Contribution

In this chapter, we propose a robust scheme for transferring a secret key between old and new users. We call this scheme *robust* because it is not only secure but also provides additional properties, as will be explained in Section 8.3.1. We also overcome flaws and limitations in existing proposals. We first define the following important terminology:

- **Owner.** An owner of a tag (tagged product) is an entity who is able to interact with the tag using a shared secret key.
- **Ownership Transfer.** The process of transferring the shared secret key to a new owner in order to enable the new owner to interact with the transferred tag.
- **Secure Ownership Transfer.** A secure ownership transfer scheme is a process of transferring the shared secret key of a physically transferred tag to a new owner so that only the new owner can interact, identify and modify the tag's content, and also transfer the ownership to the next owner if required. This process should meet the following requirements:
  1. **Old Owner's Security.** The new owner (and the adversary) do not learn the old owner's secret key.
  2. **New Owner's Security.** The old owner (and the adversary) do not learn the new owner's secret key.

8.2 Existing Work

There are several ownership transfer schemes proposed in the previous literature. However existing schemes have a number of drawbacks and limitations, as we shall now discuss.

8.2.1 Ownership Transfer without Old Owner's Security

In this section, we discuss proposed schemes where the secret key of the old owner is revealed. Ownership transfer is first discussed explicitly in [98]. In this work, a trusted center controls the owner(s) who can interact with the tag based on an access policy which uses a counter value in the tag. With each access by some owner, the counter value increments until it reaches a maximum value. After that the owner's access expires and it can no longer interact with the tag. However, once temporary access has already been delegated to some owner, the ownership cannot be transferred as this access cannot be revoked until the access expires (the tag's counter reaches its maximum value).

To deal with such a scenario, two methods are proposed. In the first method, the new owner increments the tag's counter by continuously querying it until the ownership of the previous owner has expired. This method does not meet the requirements of secure ownership transfer as the new owner is given the same pseudonym as the old owner to interact with the tag while the old owner can still interact with the tag until the expiry of its own access. In the second method, the new owner establishes mutual authentication with the tag to send it a counter value which eventually expires the validity of the old owner. The details of how this mutual authentication is established and how the new owner can write into the tag's memory are not clearly explained.

Similarly, two schemes are presented in [126] for ownership transfer. The first scheme involves a *trusted third party* (TTP). The old owner transmits its key K_{old} to the new owner. The new owner generates a new key K_{new} and sends both keys (K_{old} and K_{new}) to the TTP. The TTP shares a key K_{TTP} with the tag and uses it to encrypt both the old and the new keys ($E_{K_{TTP}}(K_{old}||K_{new})$). The TTP then sends this ciphertext to the new owner which forwards it to the tag. The tag decrypts this ciphertext, checks the validity of K_{old} and if successful updates from K_{old} to K_{new}.

This scheme has significant drawbacks because the old owner sends its private key K_{old} directly to the new owner. The author suggests that the old owner could change its key K_{old} to some temporary key before ownership transfer. This raises another problem where there is no other mechanism to recognize which tag needs ownership transfer apart from K_{old}. This requires that the key K_{TTP} shared between the tag

and the TTP is the same for all tags. This also causes a single point of failure, where compromise of a single tag breaks down the whole system. Therefore, if the tag has to validate, and is to be identified by, the old key K_{old}, then the old owner cannot change this key by itself prior to ownership transfer without involving the TTP and the tag.

The scheme presented in [42, 43] suggests ownership transfer scheme based on two keys K_p (shared between the owner and the tag) and K_u (shared between the server and the tag for update only). The new owner receives $N_T, N_R, f_{K_p}(N_T \oplus N_R)$, where N_T and N_R are random numbers generated by the tag and the old owner respectively and f is some encryption function, from the tag and forwards it to the server with an ownership transfer request. The shared secret keys with the tag have already been transferred to the new owner's server. The server sends the credentials of the tag to the new owner along with N_T, $f_{K_u}(N_T|\delta)$, which is forwarded to the tag. The tag and the server then update their keys as $K_{p_{new}} = K_p \oplus N_T \oplus \delta$, and $K_{u_{new}} = K_u \oplus N_T \oplus \delta$. The author suggests that this new owner's server can recover the old credentials for warranty claims. This reveals the old owner's secret key to the new owner.

8.2.2 Ownership Transfer without New Owner's Security

In the scheme proposed in [105], the old owner changes its key to a temporary key before ownership transfer. It then transmits its temporary key K_{temp} and the tag's credentials ID to the new owner. The new owner calculates an XOR of the encryption of the tag's identifier with the temporary old key and its own new key and sends to the tag ($e = E_{K_{temp}}(ID) \oplus E_{K_{new}}(ID)$). The tag thus recovers the encryption of the tag's identifier with the new key by again using XOR of this received message with the identifier encrypted using the temporary key ($E_{K_{new}}(ID) = E_{K_{temp}}(ID) \oplus e$). However, this scheme has a major drawback because the old owner has knowledge of the encrypted identifier and can thus recover this new encrypted identifier easily from the eavesdropped message between the new owner and the tag. Furthermore, this scheme is vulnerable to attacks, including traceability and de-synchronization (details in [69, 129, 141]).

The de-synchronization and traceability attacks on this scheme [105] are also given in [85], where the author presents a countermeasure by adding another message for integrity checking the message sent by the new owner for key update. Moreover, the author proposed that the database server generates a new key for the new owner. The encrypted identifier with the old key is hashed, XOR-ed with the encrypted identifier with the new key and sent to the tag ($e = H(E_{K_{old}}(ID)) \oplus E_{K_{new}}(ID)$). Another message is sent with this message to determine the legitimacy of the key update message ($m = H(E_{K_{old}}(ID)||e||s)$, where s is a random number earlier generated and transmit-

ted by the tag). This additional message avoids de-synchronization. Similarly, hashing the value of the identifier which is encrypted using the old key ($H(E_{K_{old}}(ID))$), provides security to the old owner's secret key. However, the new owner's security is still compromised as previously. Since hashing is public, the old owner can determine the credentials of the new owner ($E_{K_{new}}(ID) = H(E_{K_{old}}(ID)) \oplus e$).

In the second proposal [70] without the TTP, the old owner can easily eavesdrop the nonce N_T generated and communicated in clear by the tag. The old owner can thus calculate $N = N_{R1} \oplus N_T$. Therefore, this scheme can result in knowledge of the new key and the new owner's security is thus compromised.

8.2.3 Ownership Transfer without both Old and New Owner's Security

In a scheme presented in [59], the authors propose sending two messages ($m1 = E_{K_{old}}(ID) \oplus E_{K_{new}}(ID)$ and $m2 = H(E_{K_{new}}(ID) \oplus r),\ r$). The old owner's security is compromised because the new owner can calculate $E_{K_{old}}(ID) = m1 \oplus E_{K_{new}}(ID)$, and the new owner's security is also compromised because the old owner can calculate $E_{K_{new}}(ID) = m1 \oplus E_{K_{old}}(ID)$. A de-synchronization attack has also been described against this scheme in [69]. Moreover, the schemes mentioned in [129, 156] have vulnerabilities that lead to their compromise (details in [113, 141]).

8.2.4 Ownership Transfer with Limitation

There are some ownership transfer schemes which work with some limitations. Such schemes, and the reasons why these limitations are undesirable, are explained as follows:

- *Tag should be in vicinity of a trusted third party (TTP).* The scheme mentioned in [70] suggests two solutions. The first requires the TTP to directly communicate with the tag, which is not feasible considering the diversity of a tag's potential geographic location (similar schemes are mentioned in [158,159]). The tag, whose ownership is required to be transferred, has to be in physical proximity of the TTP for this protocol to succeed.

- *New owner should update the tag's secret key in private.* In the second scheme suggested in [126], not involving a TTP, the old owner changes its shared key to some temporary key K_{temp} before ownership transfer and transmits this temporary key to the new owner. The new owner then sends a query to the tag, the tag generates a random key K_r in response, and transmits it back to the new owner. The new owner then encrypts the temporary key sent by the old owner and its

own new key using the random key generated by the tag ($E_{K_r}(K_{temp}||K_{new})$). The tag finally decrypts this to update its key to K_{new}. The success of this scheme is based on the assumption that K_r will not be eavesdropped because of the small range of the backward channel (tag to reader). However, this range can vary significantly depending on the power of the adversary and the environment. If K_r is eavesdropped, it can easily compromise the secret keys of both the old and the new owner (these vulnerabilities are also mentioned in [70]). Similarly, schemes presented in [20, 29, 33, 39, 42, 43, 88, 102, 129] suggest that the new owner should update its secrets on the tag during private communication (outside the range of the old owner).

8.3 Proposed Ownership Transfer Scheme

In this section, we propose a robust ownership transfer scheme that resolves the fundamental flaws and limitations discovered during the review of existing work.

8.3.1 Properties

Our proposed ownership transfer scheme should have the following properties:

- **Old and New Owner's Security.** The proposed scheme should ensure that during and after an ownership transfer is carried out, the secret keys of both the old and the new owner should not be revealed. After ownership has been transferred, only the new owner should be able to interact with the tag and ownership of the old owner should be revoked. Similarly, the new owner should not be able to trace back any past transaction of the tag with the old owner.
- **Old and New Owner's Proximity.** Ownership transfer scheme using a TTP requires the tag to be read by the TTP for transferring ownership. This results in limitations to ownership transfer since the tag and the TTP may be at distant geographic locations. We thus specify that only the old and the new owners are required to read the tag in its proximity in order to conduct ownership transfer.
- **Public Credential Update.** Another limitation of many existing schemes is that update of the tag's credentials needs to be done in private to achieve security for the new owner's secret key. Considering that wireless communication between reader and tag can easily be eavesdropped, the use of private update is undesirable. Therefore, we propose that the tag's credentials can be updated in public whilst still preserving the secrecy of the new owner's secret key.

Additional Properties

In addition to the above properties, we suggest that an ownership transfer scheme should also offer the following properties:

- **Authorization Recovery.** This ensures that only the server (a trusted entity) can temporarily recover ownership (authorization to access the tag) on behalf of the old owner in order to facilitate after-sales/warranty-claim services without compromising the security of the old owner's secret key.
- **Tag Assurance.** This ensures that the new owner can determine the authenticity of a tag, i.e., the tag's credentials are the same as those claimed by the old owner.
- **Non-repudiation of Ownership.** The old owner is not able to deny its ownership of a tag before ownership transfer. Similarly the new owner is not able to deny that it owns this tag after ownership transfer.
- **Conformance to Standard.** The ownership transfer scheme designed for a particular RFID standard should conform with the standard's operations and functionality as much as possible.

An ownership transfer scheme offering all of these properties will hereafter be referred to as being *robust*.

8.3.2 Overview of Scheme Design

Suppose an old $Owner_o$ connected to $Server_o$ is transferring ownership of a particular (item with) Tag_k to a new $Owner_n$ connected to $Server_n$. The relationship between different entities involved in setup of an ownership transfer is shown in Figure 8.1.

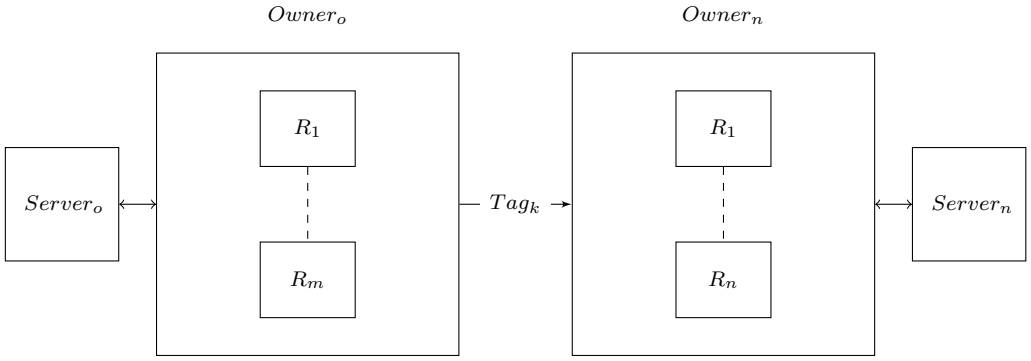

Figure 8.1: Relationship between Entities in Ownership Transfer.

If $Server_o$ is the same as $Server_n$ then a simplified version of the ownership transfer scheme can be run. We now give an overview of the proposed ownership transfer scheme which is illustrated in Figure 8.2:

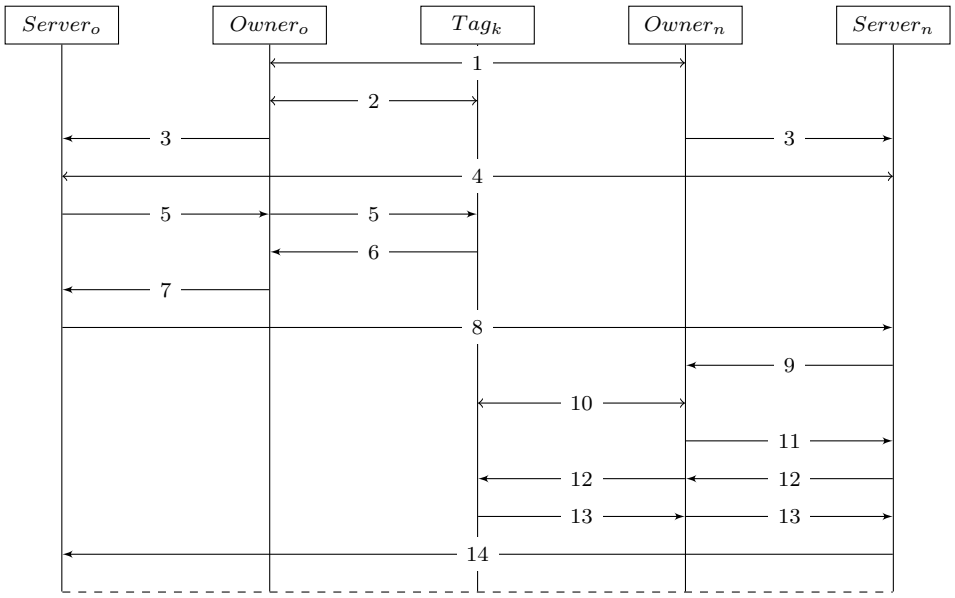

Figure 8.2: Ownership Transfer Overview.

1. **Ownership Transfer Agreement.** Both owners exchange the necessary details in order to facilitate ownership transfer.

2. **Establish Ownership.** We confirm that the entity transferring the ownership is the rightful owner of the tag. This step can be carried out earlier while an owner is communicating with a tag.

3. **Release and Acquire Credentials.** $Owner_o$ presents the release credentials to $Server_o$, which shows that it is willing to release its ownership of Tag_k. Similarly, $Owner_n$ presents the acquire credentials which indicates that it wants to acquire the ownership of Tag_k.

4. **Establish Transfer Credentials.** Both $Server_o$ and $Server_n$ establish transfer credentials, which are used to carry out the ownership transfer from $Owner_o$ to $Owner_n$.

5. **Transfer Credentials.** $Server_o$ sends the transfer credentials to $Owner_o$, who presents them to Tag_k.

6. **Ready to Transfer Acknowledgment.** Tag_k updates its credentials for transfer and sends an acknowledgment to $Owner_o$.

7. **Released Ownership Acknowledgment.** $Owner_o$ forwards Tag_k's readiness to transfer along with its own acknowledgment of releasing the ownership to $Server_o$.

8. **Ready to Transfer Acknowledgment.** $Server_o$ sends an acknowledgment to $Server_n$ that Tag_k is ready for ownership transfer.

9. **Transfer Credentials.** $Server_n$ sends the transfer credentials to $Owner_n$.

10. **Establish Ownership.** Since Tag_k has already updated its transfer credentials, $Owner_n$ establishes ownership of Tag_k using these credentials.

11. **Acquired Ownership Acknowledgment.** $Owner_n$ sends $Server_n$ an acknowledgment that it has acquired ownership of Tag_k using the transfer credentials.

12. **Key Update Credentials.** $Server_n$ sends key update credentials to $Owner_n$ who forwards it to Tag_k for updating the key from transfer to a new private key known only to $Owner_n$.

13. **Key Update Acknowledgment.** Tag_k updates its ownership credentials to the new private key and sends an acknowledgment to $Owner_n$, who also forwards it to $Server_n$.

14. **Ownership Transfer Complete Acknowledgment.** Finally $Server_n$ sends an acknowledgment to $Server_o$ that ownership transfer has been completed successfully.

8.3.3 Detailed Design

We now explain the detailed design and operation of our proposed scheme. We make the following assumptions before explaining our scheme:

- The manufacturer writes a unique secret seed into each tag.
- Each owner of a tag uses a compatible reader to interact with the tag.
- Each owner shares a secret key with its server.
- Each server is a trusted entity which shares a secret key with each corresponding tag owned by its owners. This key is delegated to the tag's current owner for offline authentication.

- The tag is capable of performing encryption and hashing (EPCC1G2 compliant tags will support on-tag encryption using the HB-2 algorithm [28]).

The notation used is summarized in Table 8.1.

Table 8.1: Notation

Notation	**Description**
KS_{old}	A shared secret key between the old owner and the tag.
KS_{new}	A shared secret key between the new owner and the tag.
KS_{so}	A shared secret key between the old owner and its server.
KS_{sn}	A shared secret key between the new owner and its server.
SK	A secret session key agreed between both old and new owner's servers.
$E_K(M)$	Encryption of a message M with key K.
EPC_k	A tag's (with index k) static and unique identity.
O	An owner's identifier.
S	A server's identifier.
$Access$	A built-in 32-bit unique access password in each tag.
r	A random number generated as a challenge by the server.
RTO	A release tag ownership signal.
ATO	An acquire tag ownership signal.
ts	A current time stamp.
$SETOT$	Set ownership transfer flag in the tag.
$RSTOT$	Reset ownership transfer flag in the tag.
$Released$	A signal to acknowledge that ownership of a tag has been released.
$Acquired$	A signal to acknowledge that ownership of a tag has been acquired.
$Updated$	A signal to acknowledge that the secret key of a tag has been updated.
s	A seed written in the tag by the manufacturer.
$H^i(s)$	The hash value after the i^{th} hashing of seed s.

8.3.4 Tag Ownership Release Phase.

In this phase, $Owner_o$ releases ownership of a particular Tag_k (with EPC_k) so that $Owner_n$ can take over its ownership. Referring to Section 8.3.2, release of tag ownership is carried out as shown in Figure 8.3 and proceeds as follows:

- **Step 1 : Ownership Transfer Agreement.** Both $Owner_o$ and $Owner_n$ exchange necessary details in order to facilitate ownership transfer.

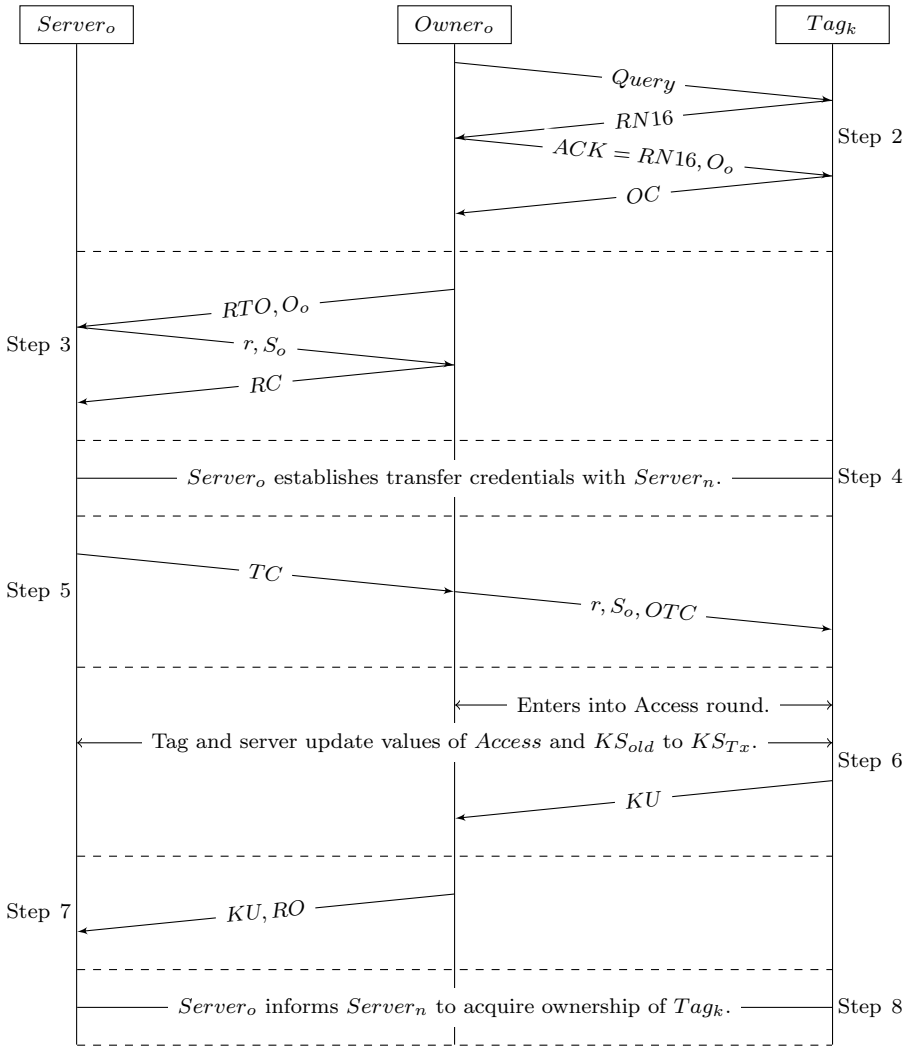

Figure 8.3: Tag Ownership Release Phase.

- **Step 2 : Establish Ownership.** $Owner_o$ initiates an EPCC1G2 communication protocol (see [48]) with Tag_k whose ownership needs to be transferred using shared secret key KS_{old}. This is concluded when Tag_k sends OC which has ownership credentials for $Owner_o$. OC is calculated as follows:

$$OC = E_{KS_{old}}(RN16, EPC_k, O_o). \tag{8.1}$$

  The correct decryption of this message determines that $Owner_o$ is an owner of this tag and can initiate an ownership transfer.

- **Step 3 : Release Credentials.** $Owner_o$ now contacts $Server_o$ for releasing

Tag_k's ownership. This is concluded when $Owner_o$ sends RC using its shared secret key KS_{so} which has release credentials as follows:

$$RC = E_{KS_{so}}(r||EPC_k||O_o||O_n||S_o||S_n||ts). \tag{8.2}$$

The correct decryption of RC determines that a rightful $Owner_o$ wants to release Tag_k's ownership and to transfer it to $Owner_n$ connected with $Server_n$.

- **Step 4 : Establish Transfer Credentials.** After obtaining the details, $Server_o$ then contacts $Server_n$. After establishing a shared key SK, $Server_o$ transfers necessary details to $Server_n$ (see Section 8.3.5).

- **Step 5 : Transfer Credentials.** This ownership transfer key SK is sent to $Owner_o$ by $Server_o$ along with details of Tag_k (EPC_k and its $Access$ password) as TC which is generated as follows:

$$TC = E_{KS_{so}}(EPC_k||O_o||S_o||SK||Access||ts). \tag{8.3}$$

When $Owner_o$ obtains the $Access$ password for Tag_k, it now can write new values into the Tag_k's memory by presenting a correct $Access$ password (see [48]). $Owner_o$ now forwards the $Server_o$'s random challenge and identifier (r and S_o) along with the transfer credentials as OTC as follows:

$$OTC = E_{KS_{old}}(EPC_k||O_o||Access||SK||SETOT||ts). \tag{8.4}$$

- **Step 6 : Updates and Ready to Transfer Acknowledgment.** Tag_k, after checking the correctness of the $Access$ and freshness of ts, sets its ownership flag and updates its values (assuming the new owner is the $(i+1)^{th}$ owner of the tag) of KS_{old} and $Access$ password as follows:

$$KS_{Tx} = SK \oplus H^{i+1}(s),$$
$$Access = Access \oplus H^{i+1}(s).$$

Tag_k finally sends an acknowledgment of key update as KU to the $Owner_o$ as follows:

$$KU = E_{KS_{Tx}}(r||EPC_k||O_o||S_o||ts). \tag{8.5}$$

- **Step 7 : Released Ownership Acknowledgment.** Since $Owner_o$ cannot decrypt KU (as it cannot calculate KS_{Tx} since s is a secret known only to the

tag and the server), it forwards this acknowledgment KU along with its own acknowledgment of releasing the tag as RO to $Server_o$, where RO is generated as follows:

$$RO = E_{KS_{so}}(Released||O_o||O_n||ts). \tag{8.6}$$

- **Step 8 : Ready to Transfer Acknowledgment.** $Server_o$ on receiving this acknowledgment and checking its authenticity informs $Server_n$ to take over ownership of the Tag_k (see Section 8.3.5).

8.3.5 Tag Ownership Transfer Phase.

When $Owner_o$ sends the information about $Server_n$ (in RC) to its $Server_o$ as explained in Section 8.3.4, the transfer phase starts as shown in Figure 8.4 and the details are as follows (see Section 8.3.2):

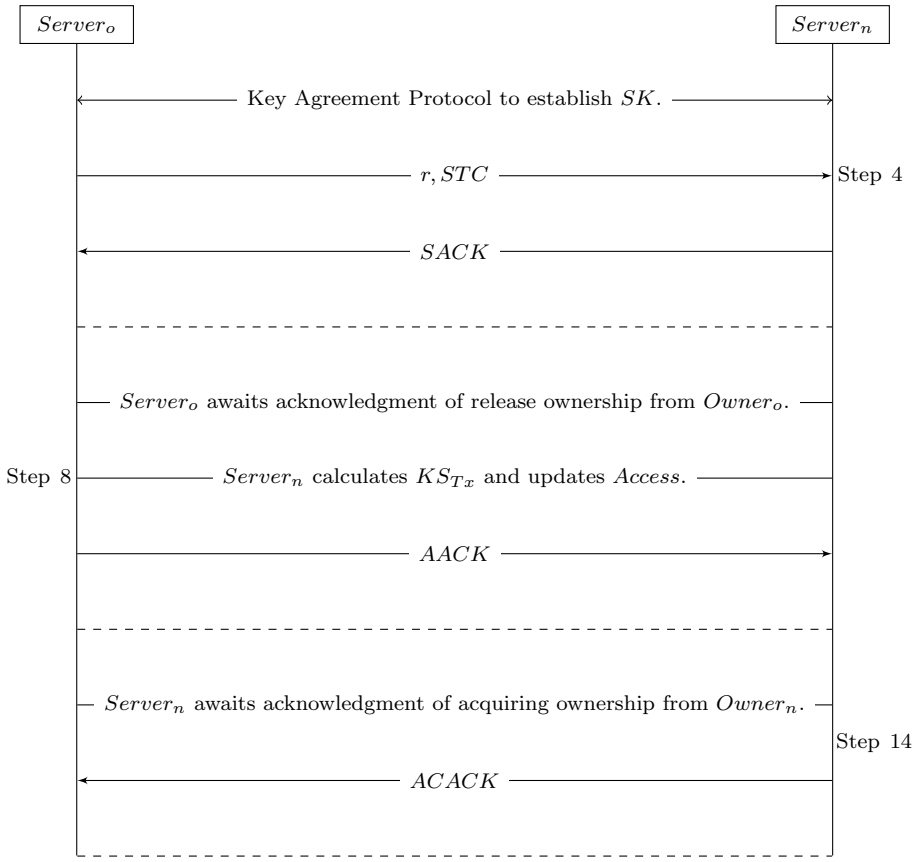

Figure 8.4: Tag Ownership Transfer Phase.

- **Step 4: Establish Transfer Credentials.** $Server_o$ contacts $Server_n$. Both

these servers use a secure key agreement protocol to establish a shared key SK between them. A new key SK is established between the two servers to achieve two goals:

- Preserve privacy of $Owner_o$'s private key KS_{old}.
- Facilitate ownership transfer using a shared secret key SK.

$Server_o$ then sends a random challenge r and transfers the details required for ownership transfer to $Server_n$ as STC which is calculated as follows:

$$STC = E_{SK}(EPC_k||Access||H^i(s)||O_o||ts). \tag{8.7}$$

$Server_n$ responds to the challenge acknowledging that it has received the information correctly by sending $SACK$ as follows:

$$SACK = E_{SK}(r||ts). \tag{8.8}$$

- **Step 8 : Ready to Transfer Acknowledgment.** $Server_o$ now waits for the acknowledgment (KU, RO) from $Owner_o$ (see Section 8.3.4) with regards to the release of Tag_k ownership. Meanwhile, $Server_n$ calculates KS_{Tx} and updates the value of $Access$ password as follows:

$$KS_{Tx} = SK \oplus H^{i+1}(s),$$
$$Access = Access \oplus H^{i+1}(s).$$

After receiving this release acknowledgment from $Owner_o$, $Server_o$ informs $Server_n$ that the latter can now acquire ownership of Tag_k. This acquire acknowledgment $AACK$ is calculated as follows:

$$AACK = E_{SK}(ATO||EPC_k||O_o||ts). \tag{8.9}$$

- **Step 14 : Ownership Transfer Complete Acknowledgment.** $Server_n$ waits until $Owner_n$ acquires ownership and update credentials ($UACK$) of Tag_k (see Section 8.3.6). $Server_n$ informs $Server_o$ that its $Owner_n$ has successfully acquired ownership of Tag_k and ownership of $Owner_o$ has been revoked by sending acquire complete acknowledgment as $ACACK$ which is calculated as follows:

$$ACACK = E_{SK}(Acquired||EPC_k||O_n||ts). \tag{8.10}$$

8.3.6 Tag Ownership Acquire Phase.

In this phase, $Owner_n$ acquires ownership of a particular Tag_k (with EPC_K) and ownership of $Owner_o$ is revoked. $Server_n$ updates its records by adding the details of $Owner_n$ (O_n), Tag_k whose ownership is acquired (EPC_k) and the timing information (ts). The acquire phase is as shown in Figure 8.5 and the details are as follows:

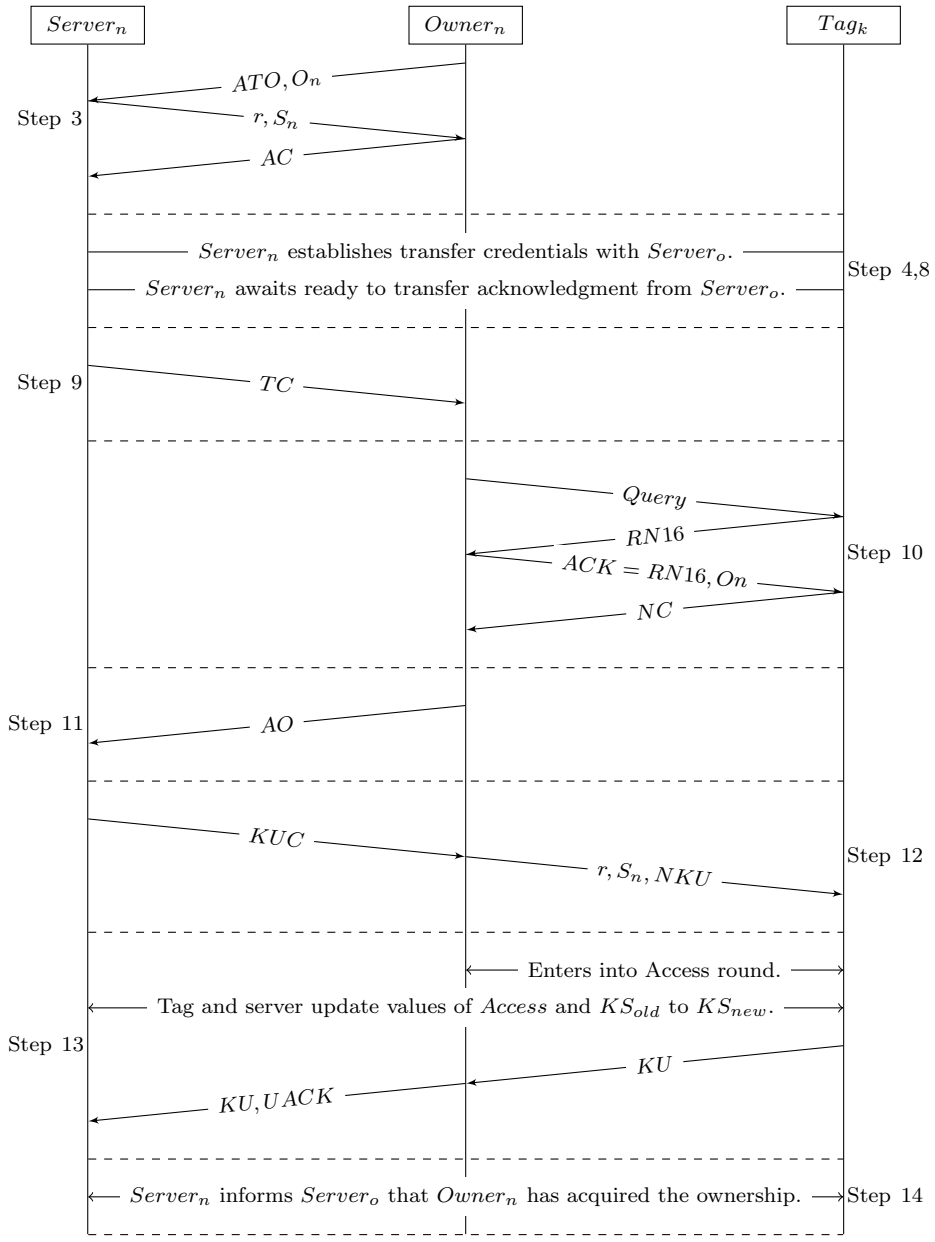

Figure 8.5: Tag Ownership Acquire Phase.

- **Step 1 : Ownership Transfer Agreement.** $Owner_o$ is transferring ownership of a particular item tagged with Tag_k to a new $Owner_n$. Both owners exchange necessary information in order to facilitate ownership transfer.

- **Step 3 : Acquire Credentials.** $Owner_n$ contacts its $Server_n$ for acquiring Tag_k's ownership using its shared secret key KS_{sn}. This is concluded by sending AC as follows:

$$AC = E_{KS_{sn}}(r||Tag_k||O_n||O_o||S_n||S_o||ts). \tag{8.11}$$

  The correct decryption of AC determines that a legitimate $Owner_n$ wants to acquire Tag_k's ownership from $Owner_o$ connected with $Server_o$ (Tag_k used in the message is the public information about tagged item).

- **Step 4 : Establish Transfer Credentials.** After receiving the public details of Tag_k and $Owner_o$ (O_o), $Server_n$ approaches (if it has not already been approached as explained in Section 8.3.4) $Server_o$ for acquiring the necessary details (see Section 8.3.5) for ownership transfer.

- **Step 8 : Ready to Transfer Acknowledgment.** After $Server_o$ receives acknowledgment from $Owner_o$ that ownership of Tag_k has been release (see Section 8.3.4), it informs $Server_n$ that the tag can now be acquired (see Section 8.3.5).

- **Step 9 : Transfer Credentials.** After receiving acknowledgment from $Server_o$ and getting necessary details of Tag_k (EPC_k), $Server_n$ forwards the transfer key KS_{Tx} to $Owner_n$ in transfer credentials message TC which is generated as follows:

$$TC = E_{KS_{sn}}(EPC_k||O_n||S_n||KS_{Tx}||ts). \tag{8.12}$$

- **Step 10 : Establish Ownership.** $Owner_n$ now initiates a standard communication protocol (see [48]) with Tag_k. This is concluded when Tag_k sends its EPC_k encrypted using the transfer key KS_{Tx} as follows:

$$NC = E_{KS_{Tx}}(RN16||EPC_k||On). \tag{8.13}$$

  Correct decryption of this message ensures tag assurance to $Owner_n$ and its ownership.

- **Step 11 : Acquired Ownership Acknowledgment.** $Owner_n$ sends the acknowledgment to its $Server_n$ that it has acquired ownership of Tag_k from previous $Owner_o$ by sending an acquired ownership acknowledgment message AO

generated as follows:

$$AO = E_{KS_{sn}}(Acquired||EPC_k||O_n||O_o||S_n||ts). \tag{8.14}$$

- **Step 12 : Key Update Credentials.** $Server_n$ now transfers $Access$ password and new private key KS_{new} to $Owner_n$ in message KUC which is calculated as follows:

$$KUC = E_{KS_{sn}}(EPC_k||O_n||S_n||KS_{new}||Access||ts). \tag{8.15}$$

After obtaining the $Access$ password for Tag_k, $Owner_n$ forwards KS_{new} to Tag_k and resets the ownership transfer flag ($RSTOT$) by sending a message NKU as follows:

$$NKU = E_{KS_{Tx}}(EPC_k||O_n||Access||KS_{new}||RSTOT||ts). \tag{8.16}$$

- **Step 13 : Update and Key Update Acknowledgment.** After receiving the correct credentials, Tag_k resets its ownership flag, replaces the value of KS_{Tx} with KS_{new} and updates its $Access$ password as follows:

$$Access = Access \oplus H^{i+2}(s).$$

An acknowledgment of key update (KU) is finally sent to $Owner_n$ as follows:

$$KU = E_{KS_{new}}(r||EPC_k||O_n||S_n||ts). \tag{8.17}$$

KS_{new} is written into the tag's memory for two reasons:

  - The ownership of $Owner_o$ is revoked.
  - $Owner_n$'s private key KS_{new} is unknown to $Owner_o$.

$Owner_n$ acknowledges $Server_n$ about the update by sending KU and its own acknowledgment $UACK$ as follows:

$$UACK = E_{KS_{sn}}(Updated||O_n||O_o||ts). \tag{8.18}$$

- **Step 14 : Ownership Transfer Complete Acknowledgment.** Finally $Server_n$ informs $Server_o$ that $Owner_n$ has taken over the ownership of the Tag_k and $Owner_o$'s ownership has been revoked (see Section 8.3.5).

8.4 Analysis

We now carry out an analysis of our proposed robust ownership transfer scheme with respect to the desired properties mentioned in Section 8.3.1. A formal analysis of release and acquire phases for any possible attacks is carried out using CasperFDR in Appendix F.

8.4.1 Old and New Owner's Security

An ownership transfer scheme should be able to change the ownership of a particular tag. It is important that ownership is only transferred by an old owner. In our proposed scheme $Owner_o$ therefore runs a standard protocol (see [48]) with Tag_k to determine its ownership with respect to KS_{old}. $Server_o$ also determines whether $Owner_o$, requesting the ownership transfer, has ownership or not. Once ownership is determined, $Server_o$ sends the shared secret key SK. This provides privacy to $Owner_o$'s own private key KS_{old} delegated by $Server_o$. Similarly, $Server_n$ finally delegates KS_{new} to $Owner_n$ to ensure the new ownership of Tag_k. During ownership transfer, Tag_k updates its shared secret key as follows:

- Secret key from KS_{old} to KS_{Tx} during release and transfer.
- Secret key from KS_{Tx} to KS_{new} during transfer and acquire.

This ensures secure ownership to both $Owner_o$ and $Owner_n$.

Old Owner's Security

$Owner_o$ has its own private shared key KS_{old} with Tag_k. To achieve old owner's security, this private key should not be exposed to $Owner_n$. A temporary key KS_{Tx} derived from SK and secret seed s is used for ownership transfer. If this transfer key is compromised during ownership transfer, it cannot relate to KS_{old}. Thus our scheme provides security for the previous $Owner_o$'s transactions.

New Owner's Security

Since only a trusted server and a genuine tag can calculate $H^{i+1}(s)$ and hence KS_{Tx}, therefore KS_{new} is never exposed to either $Owner_o$ or the adversary eavesdropping communication. So even if KS_{Tx} is compromised at some point, it is independent of KS_{new}. Moreover, it is only used during ownership transfer and then discarded.

8.4.2 Old and New Owner's Proximity

As discussed in Section 8.2.4, some of the existing schemes based on a TTP require a tag to communicate directly with a TTP in order to carry out ownership transfer. This approach is not feasible as a tag and a TTP can be at distant geographic locations. In our suggested scheme, we use back-end servers as trusted entities. Our scheme transfers the ownership using compatible readers held by $Owner_o$ and $Owner_n$ in close proximity with Tag_k and does not require this tag to communicate directly with a server.

8.4.3 Public Credential Update.

Another limitation of some of the existing schemes (see Section 8.2.4) is to update a new owner's key in private, i.e., outside the read range of the old owner (and adversary). We consider this as a limitation since the communication is wireless and can easily be eavesdropped either outside the range using non-standard equipment, or within range using stealthy equipment, by an adversary. In our scheme, we suggest that key update should be secure irrespective of whether an adversary is eavesdropping this communication. KS_{new} is transmitted to Tag_k in an encrypted message with KS_{Tx} which is unknown to $Owner_o$ and transmitted to $Owner_n$ securely only after $Owner_o$ has released its ownership of Tag_k.

8.4.4 Authorization Recovery

Since the update of a tag's credentials is only possible by using an *Access* password (see [48]), in our scheme *Access* is transferred to both $Owner_o$ and $Owner_n$ during ownership transfer. Once Tag_k updates its credentials with respect to $Owner_o$, both Tag_k and $Server_n$ update as $Access = Access \oplus H^{i+1}(s)$ (unknown to $Owner_o$) and then to $Access = Access \oplus H^{i+2}(s)$ (unknown to $Owner_n$). The server thus can revoke or recover ownership of any owner using this *Access* password.

8.4.5 Tag Assurance

Since $Owner_n$ runs a protocol with Tag_k using KS_{Tx} before updating the tag's secret key to its own private key KS_{new}, $Owner_n$ can determine whether Tag_k is the same as that claimed by $Owner_o$ (checked by the correctness of EPC_k).

8.4.6 Non-Repudiation of Ownership

During the tag ownership transfer phase as explained in Section 8.3.5, $Server_o$ also transfers the credentials of $Owner_o$ and the time when Tag_k's ownership was released,

and $Server_n$ transfers the details of $Owner_n$ and the time when ownership of Tag_k was acquired. This data can later be reproduced by servers to ensure non-repudiation of ownership by any owner.

8.4.7 Conformance to Standard

Our scheme is designed for EPCC1G2 standard compliant RFID system. This standard uses three basic operations for tag identification [48]. The proposed scheme conforms with the standard operations as follows:

- **Select.** This operation is used to select a tag population for inventory and access operations. A select command can be used a number of times to select a particular tag population using user-specified criteria. We preserve the standard select operation.
- **Inventory.** This operation is carried out for identifying the tags in the selected tag population. We preserve the standard functionality except that the EPC is sent encrypted in our suggested scheme (this feature is already in the process of being incorporated into the EPCC1G2 standard [28]).
- **Access.** This operation involves reading from/writing to a particular tag's memory. Access is granted using the standard *Access* password unique to each tag.

The following are the additional requirements to be incorporated in the standard [48]:

- **Storage.** EPC tags need to store an additional secret key (8 GE/bit for temporary storage) and a hash value of the seed (3 GE/bit for long term storage). The new version of this standard [49] also supports storage of secret keys.
- **Computation.** The computation involves an encryption and a hash function which is already in the process of incorporation in the standard [28].
- **Communication.** The proposed scheme uses the standard UHF Air Interface Protocol as specified in [48]. If a tag is not employing encryption, it can be read as per the existing standard.

8.5 Summary

In this chapter, we have proposed an ownership transfer scheme for RFID systems. A tag may be required to change its ownership several times during its life time. Our

proposed scheme overcomes limitations of existing ownership transfer schemes, since the tag is not required to be physically moved to a different location in order to be read and we do not need to update the tag's credentials in private. Finally our scheme is designed for EPCC1G2 tags but can be customized to fit into other similar environments.

Part V

Conclusion and Future Work

Chapter 9

Conclusion and Future Work

In this chapter, we shall summarize the main contributions of our research. We shall also discuss the future course of action to further ideas in this thesis.

9.1 Contributions Summary

In this thesis we have focused our work on addressing security and privacy issues in low-cost RFID systems. The contributions are summarized as follows:

- **Chapter 3.** Ultra-lightweight mutual authentication protocols (UMAPs) are designed to provide security and privacy properties to low-cost RFID systems. These systems consist of cheap tags which have constraints on their resources (computation, communication and storage). This chapter carries out security analysis of two such ultra-lightweight mutual authentication protocols (SIDRFID and DIDRFID) presented in [80]. In SIDRFID (RFID protocol with static identity), both reader and tag use their respective static identities as shared secrets. In DIDRFID (RFID protocol with dynamic identity), reader and tag share a secret key which is updated along with tag's identity after every authentication round. Both these protocols use lightweight and efficient functions such as XOR and left rotation of bits. However our security analysis helps in highlighting weaknesses present in both protocols and launching multiple attacks. The salient features of this work include:
  - A passive attack on SIDRFID reveals the hamming weight of secret identities.

  - A full disclosure active attack on SIDRFID reveals the secret identities used to provide mutual authentication.
  - Traceability and reader impersonation attack are carried out on SIDRFID.
  - A passive secret disclosure attack to uniquely determine the shared secret key is carried out against DIDRFID.
  - A traceability attack is launched on DIDRFID.

  This work appears in [12].

- **Chapter 4.** In this chapter, we further carry out analysis of several existing ultra-lightweight mutual authentication protocols (UMAPs). We contribute by generalizing weaknesses in a number of existing UMAPs and proposing a new one. The main points of this work include:

  - The weaknesses of using triangular functions only to design a UMAP can result in weaknesses which can be exploited to launch disclosure attacks.
  - The use of random nonces for update causes de-synchronization.
  - The suggested countermeasures use unreasonable overheads for low-cost tags.
  - Proposal of a new UMAP which addresses the weaknesses highlighted in earlier schemes.
  - A comparative security and performance analysis with other schemes of the same family.

  This work was published in [9].

- **Chapter 5.** RFID systems are widely used in supply chain management, however there are many outstanding security and privacy issues which need to be addressed. This includes preserving privacy of tagged items throughout the supply chain life-cycle, where a tagged item travels from manufacturer to end-user/customer. The tag starts its journey in large groups in a supply chain process and the group size reduces as it reaches the end-user. During this journey, the tag is read by online readers with known locations inside a secure zone, as well as offline readers with unknown locations in a potentially insecure zone. When the tags are in the secure zone in large numbers, the main requirement is fast read speed. This requirement changes to a need to provide security and privacy once a tag enters an insecure zone. We present an EPCglobal Class-1 Gen-2 standard

compliant [48] online/offline adaptive approach to tag security. This contribution includes:

- A scheme that achieves high tag read speed when the number of tags is large and the area is secure (online authentication scheme).
- When the number of tags reduces in size and the area becomes insecure, the scheme provides the necessary security and privacy properties (offline authentication scheme).
- This scheme switches between online and offline without user-intervention.

This work was published in [11].

- **Chapter 6.** Counterfeit items account for around 5-7% of world trade according to the International Chamber of Commerce [58]. RFID systems automate and speed up the process of item identification. However these systems can fall victim to counterfeits if appropriate measures are not taken. A counterfeit item is very difficult to detect in a supply chain management system. We propose an EPC-global Class-1 Gen-2 standard compliant [48] hierarchical anti-counterfeit mechanism that helps in not only detecting a counterfeit, but also a missing/stolen item. Features of this contribution include:

  - The proposed mechanism uses three layers to verify the legitimacy of a tagged product.
  - Each layer provides a mean for anomaly detection.
  - The mechanism is scalable, implementable and also uses efficient key management.
  - It detects not only the counterfeit/stolen items but also identifies the responsible party for such an anomaly.
  - The layered approach grows in complexity only in the event that an anomaly is detected.

  This work was published in [10].

- **Chapter 7.** E-commerce and online shopping has become widespread. However, there is also increasing fraud, where counterfeit items are sold to individual customers. End-users cannot carry UHF readers to read UHF supply chain tags and also have no access to a back-end database to verify the authenticity of a product bought online. We therefore design a customer-level anti-counterfeit framework

that uses *near field communication* (NFC) technology in smart phones to detect counterfeits in a supply chain. The main points of this work are:

  - The proposed framework suggests using two tags: one EPC tag for detection in the supply chain, and one NFC tag for online shopping and customer verification.
  - The customer can then run an authentication protocol with the product using NFC technology to ascertain the legitimacy of the product.
  - The additional cost of using an NFC tag is justified using an economic analysis.

  This work was published in [125].

- **Chapter 8.** A tagged product travels to different locations, is read by different readers, and owned by different entities. When a tag is transferred/sold to another entity, the relevant secret key should also be transferred in order to facilitate interaction of the tag with the new owner. However, if this transfer does not involve updating the key when it reaches a new owner, some concerns arise. An old owner can still retain a copy of this key or a new owner can have access to past transactions with the old owner. We propose a robust ownership transfer process which is not only secure but also achieves additional properties. The salient points of this contribution include:

  - Security is provided to both the old and the new owner's ownership credentials.
  - The proposed scheme overcomes limitations of previous schemes such as use of a trusted third party and the need to update in private.

9.2 Future Work

There is potential for further research into topics discussed in the thesis.

- **Implementation Results.** The contributions given in this thesis are theoretical and formally analyzed on paper. Future work could involve practical implementation of suggested schemes. Conventional RFID tags perform simple, hard-coded computations using the harvested power from readers. It is possible to use *computational RFID* (CRFID) tags to carry out experimental work. CRFIDs have microcontrollers, power buffers to store the harvested power, sensors, actuators and non-volatile memory. The following are well-known examples of CRFID tags:

- *WISP.* The *wireless identification and sensing platform* [118] acts as an emulator for passive EPC tags. It incorporates a microcontroller which can be programmed to carry out cryptographic computations. This platform is an open source project developed by Intel Research Seattle.
- *UMass Moo.* This is developed by the Computer Science Department at the University of Massachussets, Amherst [157]. It is an improvement to existing WISP in terms of computation, storage and other related features. It also emulates a UHF passive RFID tag which can interact with a standard UHF reader.

Another open source project known as Rifidi [117] is also a good source to simulate different business processes using RFIDs. It is a complete software simulation test bed to check the effects of different environments and thus helps in design decisions.

- **Proposing Lightweight Cryptographic Primitives.** Parts 3 and 4 of the thesis have looked into RFID applications using existing lightweight cryptographic primitives. Further research could analyze these lightweight primitives. This may also include proposing new lightweight ciphers and other cryptographic primitives for low-cost RFID systems. The list of existing lightweight encryption algorithms considered fit for use in RFIDs is shown in Table 9.1.

Table 9.1: Lightweight Encryption Algorithms Comparison

Algorithm	**Key Size (bits)**	**Area (GE)**	**Throughput (clocks/bit)**
HB2 [34]	128	2159	0.25
Grain-128 [53]	128	1857	1
Trivium [25]	128	2599	1
Present-80 [14]	80	1561	0.5
Present-128 [14]	128	2681	0.5
Katan32 [26]	80	462	8
Katan48 [26]	80	588	5.31
Katan64 [26]	80	1054	3.98
Iceberg [131]	128	7732	0.25
AES-128 [37]	128	3400	1.25

- **Issues in High-Cost Tags.** As discussed in Section 1.2 and shown in Table 1.1,

high-cost tags can support standard cryptographic primitives including public key cryptography. The well-known standard for high-cost tags is ISO/IEC 14443. Applications include e-passports, oyster card and NFC technology, to name a few. Though high-cost tags have lesser resource constraints, there is still potential for weaknesses in their implementation. Security and privacy issues in e-passports have been discussed in [64]. Similarly, the evolution of transportation ticket systems and fraud controls have been discussed at length in [96] and NFC security and privacy issues are still evolving. Therefore, analysis of the protocols suggested for high-cost tags can be carried out in future, since these tags are used in sensitive applications.

- **Other Attacks.** Some attacks are mostly assumed to be out of scope of this research, such as physical tampering and relay attacks. These attacks may be considered in more detail in future work. Relay attack [97] is a powerful adversarial attack which uses the ghost-leech model [71]. The ghost device impersonates as a genuine tag and leech as a genuine reader. The information exchanged between a legitimate tag and a legitimate reader is thus relayed using ghost-leech model. Both the communicating parties are duped to think that they are communicating within each other's vicinity. A lot of research material can be found on practical relay attacks [44, 51]. There are some countermeasures to resist such attacks including distance bounding protocols [35, 73, 78], time-out assumptions, on/off button on cards [150], metallic sleeves [145], multi-factor authentication [147] and context-aware communication [23].

- **EPC Class-1 Gen-2 Version 2.2.0.** Recently a new version of the standard [49] has been released in November, 2013. In this new standard, a tag may support one or more cryptographic suites. The two security commands *Challenge* and *Authenticate* include a *cryptographic suite indicator* (CSI) field. The four most significant bits represent the suite assigning authority and the four least significant bits represent one of the sixteen cryptographic suite assigned. For example, $CSI = 00000000_2$ is the first and $CSI = 00000001_2$ is the second suite that ISO/IEC 29167 may assign. The new security commands include *Challenge* during *Select* and *Authenticate*, *AuthComm*, *SecureComm*, *KeyUpdate*, *Untraceable* as *Access* commands. Since the UHF air interface protocol is the same as before and all these security commands are also optional, our work (on the previous Version 1.2.0) has not been affected by the new standard. A detailed security analysis of this new standard and its features can also be a related part of future work.

Bibliography

[1] G. T. Amariucai, C. Bergman, and Y. Guan. An Automatic, Time-Based, Secure Pairing Protocol for Passive RFID. In *7th International Workshop on RFID Security and Privacy*, volume 7055 of *Lecture Notes in Computer Science*, pages 108–126, Amherst, USA, 2012. Springer.

[2] A. Arbit, Y. Oren, and A. Wool. Toward Practical Public Key Anti-Counterfeiting for Low-Cost EPC Tags. In *International IEEE Conference on RFID*, pages 184–191, Orlando, USA, 2011. IEEE.

[3] G. Avoine and X. Carpent. Yet Another Ultralightweight Authentication Protocol that is Broken. In *Radio Frequency Identification. Security and Privacy Issues*, pages 20–30, Nijmegen, Netherlands, 2013. Springer.

[4] G. Avoine, X. Carpent, and B. Martin. Privacy-Friendly Synchronized Ultralightweight Authentication Protocols in the Storm. *Journal of Network and Computer Applications*, 35(2):826–843, February 2012. Elsevier Publishers.

[5] M. Bárász, B. Boros, P. Ligeti, K. Lója, and D. Nagy. Breaking LMAP. In *Conference on RFID Security*, volume 7, pages 11–16, Malaga, Spain, July 2007.

[6] M. Bárász, B. Boros, P. Ligeti, K. Lója, and D. Nagy. Passive Attack Against the M^2AP Mutual Authentication Protocol for RFID Tags. In *1st International EURASIP Workshop on RFID Technology*, pages 37–48, Vienna, Austria, September 2007.

[7] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. An Elliptic Curve Processor Suitable For RFID-Tags. *Cryptology ePrint Archive, Report 2006/227*, 2006. IACR.

[8] B. Berman. Strategies to Detect and Reduce Counterfeiting Activity. *Business Horizons*, 51(3):191 – 199, 2008. Elsevier.

[9] Z. Bilal and K. M. Martin. Ultra-lightweight Mutual Authentication Protocols: Weaknesses and Countermeasures. In *8th International Conference on Availability, Reliability and Security*, pages 304–309. IEEE, 2013.

[10] Z. Bilal and K. M. Martin. A Hierarchical Anti-Counterfeit Mechanism: Securing the Supply Chain Using RFIDs. In *Foundations and Practice of Security*, volume 8352 of *Lecture Notes in Computer Science*, pages 291–305. Springer, 2014.

[11] Z. Bilal and K. M. Martin. Adaptive Online/Offline RFID Scheme for Supply Chain Management Systems. In *International Conference on Privacy and Security in Mobile Systems*, pages 1–9. Global Wireless Summit, IEEE, 2014.

[12] Z. Bilal, K. M. Martin, and Q. Saeed. Multiple Attacks on Authentication Protocols for Low-Cost RFID Tags. *Applied Mathematics and Information Sciences*, 9(2):561–569, 2014.

[13] Z. Bilal, A. Masood, and F. Kausar. Security Analysis of Ultra-lightweight Cryptographic Protocol for Low-cost RFID Tags: Gossamer Protocol. In *International Conference on Network-Based Information Systems*, pages 260–267, Indianapolis, Indiana, USA, August 2009. IEEE, IEEE Computer Society.

[14] A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[15] S. Canard, L. Ferreira, and M. Robshaw. Improved (and Practical) Public-Key Authentication for UHF RFID Tags. In *11th International Conference on Smart Card Research and Advanced Application*, volume 7771 of *Lecture Notes in Computer Science*, pages 46–61. Springer, 2013.

[16] T. Cao, E. Bertino, and H. Lei. Security Analysis of the SASI Protocol. *Transactions on Dependable & Secure Computing*, 6(1):73–77, 2009. IEEE.

[17] J. C. H. Castro, J. M. Estévez-Tapiador, P. Peris-Lopez, and J.-J. Quisquater. Cryptanalysis of the SASI Ultralightweight RFID Authentication Protocol with Modular Rotations. In *International Workshop on Coding and Cryptography*, Ullensvang, Norway, May 2009.

[18] J. Chamberlain. *IBM Websphere RFID Handbook: A Solution Guide*. IBM, International Technical Support Organization, 2006.

[19] L. Chao. What Happens When an eBay Steal Is a Fake. *The Wall Street Journal*, June 2006. `http://online.wsj.com/article/SB115154214225593742.html`.

[20] C.-L. Chen and C.-F. Chien. An Ownership Transfer Scheme using Mobile RFIDs. *Wireless Personal Communications*, 68(3):1093–1119, 2013. Springer.

[21] O. G. Chiagozie and O. G. Nwaji. Radio Frequency Identification (RFID) based Attendance System with Automatic Door Unit. *Academic Research International*, 2(2):168–183, 2012.

[22] H.-Y. Chien. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. *Transactions on Dependable & Secure Computing*, 4(4):337–340, 2007. IEEE.

[23] A. Czeskis, K. Koscher, J. R. Smith, and T. Kohno. RFIDs and Secret Handshakes: Defending against Ghost-and-Leech Attacks and Unauthorized Reads with Context-Aware Communications. In *15th ACM conference on Computer and Communications Security*, pages 479–490. ACM, 2008.

[24] M. David and N. R. Prasad. Providing Strong Security and High Privacy in Low-Cost RFID Networks. In *Security and Privacy in Mobile Information and Communication Systems*, volume 17 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 172–179, Turin, Italy, June 2009. Springer Berlin Heidelberg.

[25] C. De Cannire. Trivium: A Stream Cipher Construction Inspired by Block Cipher Design Principles. In *Information Security*, volume 4176 of *Lecture Notes in Computer Science*, pages 171–186. Springer, 2006.

[26] C. De Cannire, O. Dunkelman, and M. Kneevi. KATAN and KTANTAN A Family of Small and Efficient Hardware-Oriented Block Ciphers. In *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer, 2009.

[27] G. Deng, H. Li, Y. Zhang, and J. Wang. Tree-LSHB+: An LPN-Based Lightweight Mutual Authentication RFID Protocol. *Wireless Personal Communications*, pages 1–16, January 2013.

[28] W. Diffie, C. Hanebeck, D. W. Engels, and A. Nathanson. Real-World Solutions for RFID Security and Privacy. `http://www.rfidjournal.com/videos/view?742`, 2012. The RFID Journal.

[29] T. Dimitriou. RFID-DOT: RFID Delegation and Ownership Transfer made simple. In *4th International Conference on Security and Privacy in Communication Networks*, pages 1–8, Istanbul, Turkey, September 2008. IEEE, IEEE Computer Society.

[30] D. Dolev and A. C. Yao. On the Security of Public Key Protocols. *Transactions on Information Theory*, 29(2):198–208, 1983. IEEE.

[31] S. Dominikus, E. Oswald, and M. Feldhofer. Symmetric Authentication for RFID Systems in Practice. In *Handout of the Ecrypt Workshop on RFID and Lightweight Crypto*, Graz, Austria, July 2005. Ecrypt.

[32] A. Eghdamian and A. Samsudin. A Secure Protocol for Ultralightweight Radio Frequency Identification (RFID) Tags. In *Informatics Engineering and Information Science*, volume 251 of *Communications in Computer and Information Science*, pages 200–213, Kuala Lumpur, Malaysia, November 2011. Springer Berlin Heidelberg.

[33] K. Elkhiyaoui, E.-O. Blass, and R. Molva. ROTIV: RFID Ownership Transfer with Issuer Verification. In *Workshop on RFID Security and Privacy*, volume 7055 of *Lecture Notes in Computer Science*, pages 163–182, Amherst, Massachusetts, USA, 2012. Springer.

[34] D. W. Engels, M.-J. O. Saarinen, P. Schweitzer, and E. M. Smith. The Hummingbird-2 Lightweight Authenticated Encryption Algorithm. In *7th International Workshop on RFID Security and Privacy*, volume 7055 of *Lecture Notes in Computer Science*, pages 19–31. Springer, 2012.

[35] A. Falahati and H. Jannati. Application of Distance Bounding Protocols with Random Challenges over RFID Noisy Communication Systems. In *IET Conference on Wireless Sensor Systems*, pages 1–5, London, UK, June 2012. IET.

[36] M. Feldhofer, M. Aigner, and S. Dominikus. An Application of RFID Tags using Secure Symmetric Authentication. In *International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, pages 43–49, Santorini Island, Greece, July 2005. IEEE, IEEE Computer Society.

[37] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong Authentication for RFID Systems using the AES Algorithm. In *6th International Workshop on Cryptographic Hardware and Embedded Systems*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370, Boston, Massachusetts, USA, August 2004. IACR, Springer.

[38] W. Feller. Stirling's Formula. *An Introduction to Probability Theory and Its Applications*, 1(3):50–53, 1968. New York: Wiley.

[39] A. Fernàndez-Mir, R. Trujillo-Rasua, J. Castella-Roca, and J. Domingo-Ferrer. A Scalable RFID Authentication Protocol supporting Ownership Transfer and Controlled Delegation. In *Workshop on RFID Security and Privacy*, volume 7055 of *Lecture Notes in Computer Science*, pages 147–162. Springer, Amherst, Massachusetts, USA, 2012.

[40] K. Finkenzeller and R. Waddington. *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*. Wiley New York, 1999.

[41] C. Floerkemeier, R. Schneider, and M. Langheinrich. Scanning with a Purpose - Supporting the Fair Information Principles in RFID Protocols. In *2nd International Symposium on Ubiquitous Computing Systems*, volume 3598 of *Lecture Notes in Computer Science*, pages 214–231, Tokyo, Japan, November 2005. Springer.

[42] S. Fouladgar and H. Afifi. A simple Privacy Protecting Scheme enabling Delegation and Ownership Transfer for RFID Tags. *Journal of Communications*, 2(6):6–13, 2007.

[43] S. Fouladgar and H. Afifi. An efficient Delegation and Transfer of Ownership Protocol for RFID Tags. In *1st International EURASIP Workshop on RFID Technology*, volume 160, Vienna, Austria, September 2007. Citeseer.

[44] L. Francis, G. P. Hancke, K. Mayes, and K. Markantonakis. Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones. *IACR Cryptology ePrint Archive*, 2011:618, 2011.

[45] P. Gardiner, M. Goldsmith, J. Hulance, D. Jackson, B. Roscoe, B. Scattergood, and P. Armstrong. *Failures-Divergence Refinement-FDR2 User Manual*. Formal Systems (Europe) Ltd., Oxford University Computing Laboratory, 9 edition, October 2010.

[46] G. Gaubatz, J.-P. Kaps, E. Öztürk, and B. Sunar. State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks. In *The 3rd IEEE Conference on Pervasive Computing and Communications Workshops*, pages 146–150, Kauai Island,USA, March 2005. IEEE.

[47] B. Glover and H. Bhatt. *RFID Essentials*. O'Reilly Media, Inc., 2006.

[48] GS1 EPCGlobal. *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz, Version 1.2.0*, October 2008. http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2/uhfc1g2_1_2_0-standard-20080511.pdf.

[49] GS1 EPCGlobal. *EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Communications at 860 MHz 960 MHz Version 2.0.0 Ratified*, November 2013. http://www.gs1.org/sites/default/files/docs/uhfc1g2/uhfc1g2_2_0_0_standard_20131101.pdf.

[50] J. M. Gutiérrez, M. A. Hernández, P. J. Miana, and N. Romero. New Identities in the Catalan Triangle. *Journal of Mathematical Analysis and Applications*, 341(1):52–61, 2008. Elsevier Inc.

[51] G. P. Hancke. A Practical Relay Attack on ISO 14443 Proximity Cards. Technical report, University of Cambridge Computer Laboratory, 2005.

[52] R. F. Harrington. Theory of Loaded Scatterers. *Proceedings of the Institution of Electrical Engineers*, 111(4):617–623, 1964. IET.

[53] M. Hell, T. Johansson, A. Maximov, and W. Meier. The Grain Family of Stream Ciphers. In *New Stream Cipher Designs*, volume 4986 of *Lecture Notes in Computer Science*, pages 179–190. Springer, 2008.

[54] J. C. Hernandez-Castro, P. Peris-Lopez, R. C. Phan, and J. M. Estevez-Tapiador. Cryptanalysis of the David-Prasad RFID Ultralightweight Authentication Protocol. In *Workshop on RFID Security*, volume 6370 of *Lecture Notes in Computer Science*, pages 22–34, Turkey, June 2010. Springer.

[55] C. A. R. Hoare. *Communicating Sequential Processes*, volume 178. Prentice Hall Englewood Cliffs, 1985.

[56] N. J. Hopper and M. Blum. Secure Human Identification Protocols. In *Advances in cryptologyASIACRYPT 2001*, pages 52–66. Springer, 2001.

[57] S. Inoue and H. Yasuura. RFID Privacy Using User-controllable Uniqueness. In *RFID Privacy Workshop*, pages 1–9, MIT, Massachusetts, USA, November 2003. Citeseer.

[58] Counterfeiting Intelligence Bureau. http://www.iccwbo.org/products-and-services/fighting-commercial-crime/counterfeiting-intelligence-bureau/, 2014.

[59] P. Jappinen and H. Hamalainen. Enhanced RFID Security method with Ownership Transfer. In *International Conference on Computational Intelligence and Security*, volume 2, pages 382–385. IEEE, 2008.

[60] P. Jäppinen and M. Lampi. Hardware Cost Measurement of Lightweight Security Protocols. *Wireless personal communications*, 71(2):1479–1486, July 2013. Springer.

[61] A. Juels. Minimalist Cryptography for Low-Cost RFID Tags. In *4th International Conference on Security in Communication Networks*, volume 3352 of *Lecture Notes in Computer Science*, pages 149–164, Amalfi, Italy, 2005. Springer.

[62] A. Juels. Strengthening EPC Tags against Cloning. In *4th ACM workshop on Wireless security*, WiSe '05, pages 67–76, Cologne, Germany, 2005. ACM.

[63] A. Juels. RFID Security and Privacy: A Research Survey. *Journal on Selected Areas in Communications*, 24(2):381–394, 2006. IEEE.

[64] A. Juels, D. Molnar, and D. Wagner. Security and Privacy Issues in E-passports. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 74–88. IEEE, 2005.

[65] A. Juels, R. Pappu, and B. Parno. Unidirectional Key Distribution Across Time and Space with Applications to RFID Security. In *17th USENIX Security Symposium*, pages 75–90, San Jose, California, USA, July 2008. USENIX.

[66] A. Juels, R. L. Rivest, and M. Szydlo. The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. In *10th ACM Conference on Computer and Communications Security*, pages 103–111, Washington, DC, USA, October 2003. ACM, ACM Press.

[67] A. Juels, P. F. Syverson, and D. V. Bailey. High-Power Proxies for Enhancing RFID Privacy and Utility. In *Privacy Enhancing Technologies, 5th International Workshop*, volume 3856 of *Lecture Notes in Computer Science*, pages 210–226, Cavtat, Croatia, 2006. Springer.

[68] A. Juels and S. A. Weis. Authenticating Pervasive Devices with Human Protocols. In *Advances in Cryptology–CRYPTO 2005*, pages 293–308. Springer, 2005.

[69] G. Kapoor and S. Piramuthu. Vulnerabilities in some recently Proposed RFID Ownership Transfer Protocols. *IEEE Communications Letters*, 14(3):260–262, March 2010. IEEE Computer Society.

[70] G. Kapoor and S. Piramuthu. Single RFID Tag Ownership Transfer Protocols. In *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, volume 42, pages 164–173, Los Alamitos, California, USA, 2012. IEEE Computer Society.

[71] Z. Kfir and A. Wool. Picking Virtual Pockets using Relay Attacks on Contactless Smartcard. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 47–58. IEEE, 2005.

[72] M. Kianersi, M. Gardeshi, and M. Arjmand. SULMA: A Secure Ultra Light-Weight Mutual Authentication Protocol for Lowcost RFID Tags. *International Journal of UbiComp*, 2(2):17–24, 2011. AIRCC Publishing Corporation.

[73] J. S. Kim, K. Cho, D. H. Yum, S. J. Hong, and P. J. Lee. Lightweight Distance Bounding Protocol against Relay Attacks. *IEICE Transactions on Information and Systems*, 95(4):1155–1158, April 2012. The Institute of Electronics, Information and Communication Engineers.

[74] A. Klimov and A. Shamir. Cryptographic Applications of T-Functions. In *Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 248–261. Springer, 2004.

[75] R. Koh, E. W. Schuster, I. Chackrabarti, and A. Bellman. Securing the Pharmaceutical Supply Chain. *White Paper, Auto-ID Labs*, 2003. Massachusetts Institute of Technology.

[76] J. Landt. The History of RFID. *Journal on Potentials*, 24(4):8–11, 2005. IEEE.

[77] M. Langheinrich and R. Marti. Practical Minimalist Cryptography for RFID Privacy. *IEEE Systems Journal, Special Issue on RFID Technology*, 1(2):115–128, December 2007. IEEE.

[78] S. Lee, J. S. Kim, S. J. Hong, and J. Kim. Distance Bounding with Delayed Responses. *Journal on Communications Letters*, 16(9):1478–1481, 2012. IEEE.

[79] Y. Lee, F. Cheng, and Y. T. Leung. Exploring the Impact of RFID on Supply Chain Dynamics. In *36th Conference on Winter Simulation*, volume 2, pages 1145 – 1152. IEEE, December 2004.

[80] Y.-C. Lee. Two Ultralightweight Authentication Protocols for Low-Cost RFID Tags. *Applied Mathematics and Information Sciences*, 6(2S):425–431, 2012.

[81] Y.-C. Lee, Y.-C. Hsieh, P.-S. You, and T.-C. Chen. A New Ultralightweight RFID Protocol with Mutual Authentication. In *WASE International Conference on Information Engineering*, volume 2, pages 58–61, Taiyuan, Shanxi, August 2009. IEEE, IEEE Computer Society.

[82] Y.-S. Lee, T.-Y. Kim, and H. J. Lee. Mutual Authentication Protocol for Enhanced RFID Security and Anti-Counterfeiting. In *26th International Conference on Advanced Information Networking and Applications Workshops*, pages 558–563. IEEE, 2012.

[83] M. Lehtonen, J. Al-Kassab, F. Michahelles, and O. Kasten. Anti-Counterfeiting Business Case Report. Technical report, Technical report, BRIDGE Project, December 2007.

[84] M. Lehtonen, T. Staake, F. Michahelles, and E. Fleisch. From Identification to Authentication - A Review of RFID Product Authentication Techniques. In *Networked RFID Systems and Lightweight Cryptography*, pages 169–187. Springer, 2008.

[85] H. Lei and T. Cao. RFID Protocol enabling Ownership Transfer to protect against Traceability and DoS attacks. In *1st International Symposium on Data, Privacy, and E-Commerce*, pages 508–510. IEEE, 2007.

[86] T. Li and R. H. Deng. Vulnerability Analysis of EMAP - An Efficient RFID Mutual Authentication Protocol. In *2nd International Conference on Availability, Reliability and Security*, pages 238–245, Vienna, Austria, April 2007. IEEE.

[87] T. Li and G. Wang. Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols. In *22nd International Information Security Conference*, volume 232 of *IFIP*, pages 109–120, Sandton, Gauteng, South Africa, May 2007. IFIP, Springer.

[88] C. H. Lim and T. Kwon. Strong and Robust RFID Authentication enabling perfect Ownership Transfer. In *Information and Communications Security*, volume 4307 of *Lecture Notes in Computer Science*, pages 1–20, Raleigh, North Carolina, USA, December 2006. Springer.

[89] Z. Lin and J. S. Song. An Improvement in HB-Family Lightweight Authentication Protocols for Practical Use of RFID System. *Journal of Advances in Computer Networks*, 1(1):61–65, January 2013.

[90] G. Lowe. Casper: A Compiler for the Analysis of Security Protocols. *Journal of Computer Security*, 6(1):53–84, 1998. IOS Press.

[91] G. Madlmayr, J. Langer, C. Kantner, and J. Scharinger. NFC Devices: Security and Privacy. In *3rd International Conference on Availability, Reliability and Security*, pages 642–647. IEEE, 2008.

[92] M. S. Mamun and A. Miyaji. A fully-secure RFID authentication protocol from exact LPN. In *International Conference on Trust, Security and Privacy in Computing and Communications*, pages 102–109, Melbourne, Australia, July 2013.

[93] K. Markantonakis and K. Mayes. *Secure Smart Embedded Devices, Platforms and Applications*. Springer, 2013.

[94] S. Martínez, M. Valls, C. Roig, J. M. Miret, and F. Giné. A Secure Elliptic Curve-Based RFID Protocol. *Journal of Computer Science and Technology*, 24(2):309–318, 2009. Springer.

[95] C. Matlack and T. Mullaney. Fed Up With Fakes. *Bloomberg Businessweek*, 4004:56, October 2006. `http://www.businessweek.com/stories/2006-10-08/fed-up-with-fakes`.

[96] K. E. Mayes, K. Markantonakis, and G. Hancke. Transport Ticketing Security and Fraud Controls. *Information Security Technical Report*, 14(2):87–95, 2009. Elsevier.

[97] A. Mitrokotsa, M. R. Rieback, and A. S. Tanenbaum. Classifying RFID Attacks and Defenses. *Information Systems Frontiers*, 12(5):491–505, 2010. Springer.

[98] D. Molnar, A. Soppera, and D. Wagner. A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags. In *Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 276–290. Springer, 2006.

[99] NFC Forum Tag Type Technical Specifications. `http://www.nfc-forum.org/specs/spec\_list/\#tagtypes`, 2010.

[100] NFC Forum. *Signature Record Type Definition: Technical Specification*, signature 1.0 edition, November 2010. `http://www.nfc-forum.org/specs/spec_list/\#rtds`.

[101] A Definitive List of NFC Phones. `http://www.nfcworld.com/nfc-phones-list/`, 2012.

[102] C. Y. Ng, W. Susilo, Y. Mu, and R. Safavi-Naini. Practical RFID Ownership Transfer Scheme. *Journal of Computer Security*, 19(2):319–341, 2011. IOS Press.

[103] M. C. O'Connor. NXP, Sony Partner to Make Chip for NFC Apps. *Accessed on RFID Journal*, pages 01–31, 2007.

[104] Y. Oren and M. Feldhofer. A low-resource Public-Key Identification Scheme for RFID Tags and Sensor Nodes. In *2nd ACM Conference on Wireless Network Security*, pages 59–68, Zurich, Switzerland, March 2009. ACM, ACM Press.

[105] K. Osaka, T. Tsuyoshi, K. Yamazaki, and T. Osamu. An Efficient and Secure RFID Security Method with Ownership Transfer. In *International Conference on Computational Intelligence and Security*, volume 4456 of *Lecture Notes in Computer Science*, pages 778–787. Springer, 2007.

[106] J. Pearson. Securing the Pharmaceutical Supply Chain with RFID and Public-key Infrastructure (PKI) Technologies. Texas Instruments White Paper, June 2005.

[107] P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda. EMAP: An Efficient Mutual-Authentication Protocol for Low-Cost RFID Tags. In *On the Move to Meaningful Internet Systems*, volume 4277 of *Lecture Notes in Computer Science*, pages 352–361. Springer, 2006.

[108] P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda. M^{2}AP: A Minimalist Mutual-Authentication Protocol for Low-Cost RFID Tags. In *International Conference on Ubiquitous Intelligence and Computing*, volume 4159 of *Lecture Notes in Computer Science*, pages 912–923, Wuhan and Three Gorges, China, September 2006. Springer.

[109] P. Peris-Lopez, J. C. H. Castro, R. C.-W. Phan, J. M. Estévez-Tapiador, and T. Li. Quasi-Linear Cryptanalysis of a Secure RFID Ultralightweight Authentication Protocol. In *6th China International Conference on Information Security and Cryptology*, volume 6584 of *Lecture Notes in Computer Science*, pages 427–442, Shanghai, China, 2011. Springer.

[110] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID Tags. In *2nd Workshop on RFID Security*, page 6, Graz, Austria, July 2006. Ecrypt.

[111] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol. In *Workshop on Information Security Applications*, volume 5379 of *Lecture Notes in Computer Science*, pages 56–68, Jeju Island, Korea, September 2008. Springer.

[112] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and J. C. A. van der Lubbe. Security Flaws in a Recent Ultralightweight RFID Protocol. In *Workshop on RFID Security*, volume 4 of *Cryptology and Information Security*, pages 83–93, Singapore, Republic of Singapore, February 2010. IOS Press.

[113] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Tapiador, T. Li, and Y. Li. Vulnerability Analysis of RFID Protocols for Tag Ownership Transfer. *Computer Networks*, 54(9):1502–1508, 2010. Elsevier.

[114] P. Peris-Lopez, L. Tieyan, and J. C. Hernandez-Castro. Lightweight Props on the Weak Security of EPC Class-1 Generation-2 Standard. *IEICE transactions on information and systems*, 93(3):518–527, 2010. The Institute of Electronics, Information and Communication Engineers.

[115] A. Poschmann, G. Leander, K. Schramm, and C. Paar. A Family of Light-Weight Block Ciphers Based on DES Suited for RFID Applications. In *Workshop on RFID Security*, Graz, Austria, July 2006. Ecrypt.

[116] A. Poschmann, M. J. B. Robshaw, F. Vater, and C. Paar. Lightweight Cryptography and RFID: Tackling the Hidden Overhead. In *International Conference on Information Security and Cryptology*, pages 129–145. Springer, 2009.

[117] L. Pramari. Rifidi. *Software Defined RFID*, 2007. `www.rifidi.orgoronSourceforge(sourceforge.net/projects/rifidi/)`.

[118] L. Radoslav. *Wireless Identification and Sensing Platform*. PON PRESS, 2012.

[119] R. Richardson. Remotely Activated Radio Frequency Powered Devices, 1963. US Patent 3,098,971.

[120] M. R. Rieback, B. Crispo, and A. S. Tanenbaum. RFID Guardian: A Battery-Powered Mobile Device for RFID Privacy Management. In *10th Australasian Conference on Information Security and Privacy*, volume 3574 of *Lecture Notes in Computer Science*, pages 184–194, Brisbane, Australia, July 2005. Springer.

[121] P. Rizomiliotis and S. Gritzalis. GHB#: A Provably Secure HB-Like Lightweight Authentication Protocol. In *10th International Conference on Applied Cryptography and Network Security*, Singapore, China, June 2012.

[122] M. Roberti. A 5-Cent Breakthrough. *The RFID Journal*, 2007. `http://www.rfidjournal.com/articles/view?2295`.

[123] M. Roberti. The History of RFID technology. *The RFID Journal*, 2011. `http://www.rfidjournal.com/articles/view?1338`.

[124] M.-J. O. Saarinen and D. Engels. A Do-It-All-Cipher for RFID: Design Requirements. *Cryptology ePrint Archive, Report 2012/317*, 2012. IACR.

[125] M. Q. Saeed, Z. Bilal, and C. D. Walter. An NFC based Consumer-Level Counterfeit Detection Framework. In *11th Annual International Conference on Privacy, Security and Trust (PST)*, pages 135–142. IEEE, 2013.

[126] J. Saito, K. Imamoto, and K. Sakurai. Reassignment Scheme of an RFID Tag's Key for Owner Transfer. In *Embedded and Ubiquitous Computing Workshops*, volume 3823 of *Lecture Notes in Computer Science*, pages 1303–1312. Springer, 2005.

[127] S. E. Sarma, S. A. Weis, and D. W. Engels. RFID Systems and Security and Privacy Implications. In *4th International Workshop on Cryptographic Hardware and Embedded Systems*, volume 2523 of *Lecture Notes in Computer Science*, pages 454–469, Redwood Shores, CA, USA, 2003. Springer.

[128] M. Schneider, J. Bräuer, H. Heiss, and B. Lutterbeck. Application and Scenarios of RFID Technology. *Seminar on Privacy Issues in the Environment of Pervasive Computing*, 2004.

[129] B. Song. RFID Tag Ownership Transfer. In *Workshop on RFID Security*, Budapest, Hungary, July 2008.

[130] T. Staake, F. Thiesse, and E. Fleisch. Extending the EPC Network: The Potential of RFID in Anti-Counterfeiting. In *ACM Symposium on Applied Computing*, volume 6, pages 1607–1612, New York, USA, 2005. ACM.

[131] F.-X. Standaert, G. Piret, G. Rouvroy, J.-J. Quisquater, and J.-D. Legat. ICEBERG : An Involutional Cipher Efficient for Block Encryption in Reconfigurable Hardware. In *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 279–298. Springer, 2004.

[132] R. P. Stanley. *Enumerative Combinatorics*. Number 49. Cambridge University Press, 2011.

[133] H. Stockman. Communication by means of Reflected Power. *Proceedings of the IRE*, 36(10):1196–1204, 1948. IEEE.

[134] H.-M. Sun, W.-C. Ting, and K.-H. Wang. On the Security of Chien's Ultralightweight RFID Authentication Protocol. *Transactions on Dependable & Secure Computing*, 8(2):315–317, 2011. IEEE.

[135] D. Tagra, M. Rahman, and S. Sampalli. Technique for Preventing DoS Attacks on RFID Systems. In *18th International Conference on Software Telecommunications and Computer Networks*, pages 6–10, Bol, Island of Brac, Croatia, September 2010. IEEE, IEEE Computer Society.

[136] K. Takaragi, M. Usami, R. Imura, R. Itsuki, and T. Satoh. An Ultra Small Individual Recognition Security Chip. *IEEE MICRO*, 21(6):43–49, 2001. IEEE Computer Society.

[137] The Spread of Counterfeiting: Knock-offs Catch On. `http://www.economist.com/node/15610089`, 2010.

[138] A. Trevarthen. The National Livestock Identification System: The Importance of Traceability in E-Business. *Journal of Theoretical and Applied Electronic Commerce Research*, 2(1):49–62, 2007.

[139] P. Tuyls and L. Batina. RFID-Tags for Anti-Counterfeiting. In *The Cryptographers' Track at the RSA Conference*, volume 3860 of *Lecture Notes in Computer Science*, pages 115–131, San Jose, California, USA, February 2006. Springer.

[140] I. Vajda and L. Buttyán. Lightweight Authentication Protocols for Low-Cost RFID Tags. In *2nd Workshop on Security in Ubiquitous Computing*, Seattle, Washington, USA, October 2003. ACM.

[141] T. van Deursen, S. Mauw, S. Radomirović, and P. Vullers. Secure Ownership and Ownership Transfer in RFID Systems. In *14th European Symposium on Research in Computer Security*, volume 5789 of *Lecture Notes in Computer Science*, pages 637–654, Saint-Malo, France, September 2009. Springer.

[142] B. Violino. Merloni unveils RFID appliances. *The RFID Journal*, 2003. `http://www.rfidjournal.com/article/articleview/369/1/1`.

[143] J. Vogelman. Passive Data Transmission Techniques using Radar Echoes, 1968. US Patent 3,391,404.

[144] R. Want. An Introduction to RFID Technology. *Journal on Pervasive Computing*, 5(1):25–33, Jan 2006. IEEE.

[145] Identity Stronghold. `http://www.idstronghold.com/`.

[146] International Journal on RFID. `http://www.rfidjournal.com`.

[147] Privaris plusID Products. `http://www.privaris.com/products/index.html`.

[148] The NFC Forum. `http://www.nfc-forum.org/home/`, 2004.

[149] ITU Page on Definitions of ISM Bands. `http://www.itu.int/ITUR/terrestrial/faq/index.html`, September 2005.

[150] SmartCode Corp. Solves Privacy Issue Relating To Potential Unauthorized Reading Of RFID Enabled Passports And ID Cards. `http://www.rfidsolutionsonline.com/doc/smartcode-corp-solves-privacy-issue-relating-0001`, 2006.

[151] ISO RFID Standards : A Complete List. `http://www.rfid.net/basics/186-iso-rfid-standards-a-complete-list`, 2012.

[152] World RFID Market to Reach 20 Billion USD in 2014? `http://www.rfidworld.ca/world-rfid-market-to-reach-20-billion-usd-in-2014/769`, 2012.

[153] J. Wolkerstorfer. Is Elliptic-Curve Cryptography Suitable to Secure RFID Tags? In *Handout of the Ecrypt Workshop on RFID and Lightweight Crypto*, Graz, Austria, July 2005. Ecrypt.

[154] K.-H. Yeh and N. Lo. Improvement of Two Lightweight RFID Authentication Protocols. *Information Assurance and Security Letters*, 1:6–11, 2010. Dynamic Publishers Inc.

[155] K.-H. Yeh, N. Lo, and E. Winata. An Efficient Ultralightweight Authentication Protocol for RFID Systems. In *Workshop on RFID Security*, volume 4 of *Cryptology and Information Security*, pages 49–60, Singapore, Republic of Singapore, February 2010. IOS Press.

[156] E.-J. Yoon and K.-Y. Yoo. Two Security Problems of RFID Security Method with Ownership Transfer. In *IFIP International Conference on Network and Parallel Computing*, pages 68–73. IEEE Computer Society, 2008.

[157] H. Zhang, J. Gummeson, B. Ransford, and K. Fu. Moo: A Batteryless Computational RFID and Sensing Platform. Technical report, University of Massachusetts Computer Science Technical Report UM-CS-2011-020, June 2011.

[158] W. Zhou, E. J. Yoon, and S. Piramuthu. Varying Levels of RFID Tag Ownership in Supply Chains. In *On the Move to Meaningful Internet Systems Workshops*, volume 7046 of *Lecture Notes in Computer Science*, pages 228–235, Crete, Greece, October 2011. Springer Berlin Heidelberg.

[159] W. Zhou, E. J. Yoon, and S. Piramuthu. Hierarchical RFID Tag Ownership and Transfer in Supply Chains. In *10th Workshop on E-Business*, volume 108 of *Lecture Notes in Business Information Processing*, pages 390–398, Shanghai, China, 2012. Springer Berlin Heidelberg.

Appendix A

Appendix of Chapter 2

A.1 A short History of RFID Systems

This section presents a historical perspective of the development of RFID systems in use today.

- **1846-1930.** RFID history dates back to 1846, when Michael Faraday, an English experimentalist, proposed electromagnetic energy using light and radio waves. James Clerk Maxwell, a Scottish physicist, published his electromagnetic theory in 1864. Heinrich Rudolf Hertz, a German physicist, experimentally proved that radio waves can be transmitted and received in 1887. In the experiment he generated a spark using energy and saw a similar spark a little distance away with no wires or connections. This experiment was quickly followed by Aleksander Popov in Russia. It was Marconi who actually shaped this energy and transferred data over it. Guglielmo Marconi transmitted radio signals across the Atlantic in 1901. Morse code were sent using these radio waves and the first voice broadcast was carried out in 1906 by Ernst F.W. Alexanderson. This marks the beginning of modern radio communications [76].

- **1930-1950.** The general belief is that first application of RFID technology can be traced back to World War II. In 1935, Sir Robert Alexander Watson Watt, a Scottish physicist, discovered the use of radar to signal approaching aircraft. The British, Germans, Japanese and Americans were using this radar, however there was no information that could accurately identify whether an approaching aircraft is a friend or foe. The first passive RFID system was when the Germans devised a method of changing radio signals by rolling their aircraft and thus identifying themselves to the radar. The first active RFID system was when the British invented the first *identify friend or foe* (IFF) system which, on receiving a signal

from the radar, broadcast a signal back to identify itself [123]. The first public appearance of RFID systems can be traced back to Harry Stockman's famous paper *Communications by means of Reflected Power* in 1948 [133].

- **1950-1970.** Famous work [52, 119, 143] carried out during this period was a prelude to RFID explosion in industry. Research and experimentation started to use RFID technology into different applications. As a result *electronic article surveillance* (EAS) was developed to identify electronic articles using 1-bit tags.
- **1970-1990.** The first US patent for an active RFID tag was claimed by Mario W. Cardullo on January 23, 1973 and first patent for a passive RFID tag was claimed by Charles Walton, a California entrepreneur, in the same year, who developed a passive transponder used to unlock a door without a key [123]. More applications were developed including nuclear material and cows tracking, and access token using low frequency (LF 125 kHz) and high frequency (HF 13.56MHz).
- **1990-2003.** Ultra high frequency (UHF 860-960MHz) tags were developed. Standardization of RFID system started during this period and most notable achievement of this technology was its use in supply chain management systems.
- **2003-till date.** RFID system's rapid development is being challenged by security and privacy issues. Tag supporting cryptography are being developed.

A.2 RFID Standards

Following is the comprehensive list of standards to best of knowledge:

- ISO 11784 : Radio frequency identification of animals (code structure)
- ISO 11785 : Radio frequency identification of animals (technical concept)
- ISO 14223 : Specifies the air interface between the transceiver and the advanced transponder used in the radio frequency identification of animals under the condition of full upward compatibility according to ISO 11784 and ISO 11785.
- ISO/IEC 14443 : Identification cards – Contact-less integrated circuit(s) cards – Proximity cards
  - Part 1 : Physical characteristics
  - Part 2 : Radio frequency power and signal interface
  - Part 3 : Initialization and anti-collision

  - Part 4 : Transmission protocol

- ISO/IEC 15434 : Transfer Syntax for High Capacity ADC Media
- ISO/IEC 15459 : Unique identifier for transport units
  - Part 1 : Unique identification of transport units
  - Part 2 : Registration procedures
  - Part 3 : Common rules for unique identification
  - Part 4 : Unique item identification for supply chain management
  - Part 5 : Unique Identification of Returnable Transport Items (RTIs)
  - Part 6 : Unique identification for product groupings in material life cycle management
- ISO/IEC 15961 : Information technology – Radio frequency identification (RFID) for item management (Data protocol: application interface)
  - Part 1 : Application interface
  - Part 2 : Registration of RFID data constructs
  - Part 3 : RFID data constructs
- ISO/IEC 15962 : Information technology – Radio frequency identification (RFID) for item management (Data protocol: data encoding rules and logical memory functions JTC 1/SC 31)
- ISO/IEC 15693 : Identification cards – Contact-less integrated circuit(s) cards – Vicinity cards
  - Part 1 : Physical characteristics
  - Part 2 : Air interface and initialization
  - Part 3 : Anti-collision and transmission protocol
- ISO/IEC 18000 : RFID for Item Management
  - Part 1 : Defines the foundation for all air interface definitions in the ISO/IEC 18000 series.
  - Part 2 : Parameters for air interface communications below 135 kHz (Type A (FDX): 125 kHz and Type B (HDX): 134.2 kHz)
  - Part 3 : Parameters for air interface communications at 13.56 MHz

  - Part 4 : Parameters for air interface communications at 2.45 GHz
  - Part 6 : Parameters for air interface communications at 860 MHz to 960 MHz– Type A and type B with the primary difference being the anti-collision algorithm used. Type C - also know as EPCglobal Class 1 Gen 2.
  - Part 7 : Parameters for active air interface communications at 433 MHz

- ISO/IEC 18001 : RFID for Item Management - Application Requirements Profiles (ARP)

- ISO/IEC TR 18046 : Radio frequency identification device performance test methods

- ISO/IEC TR 18047 : Information technology – Radio frequency identification device conformance test methods

  - Part 1 : Not available
  - Part 2 : Parameters for Air Interface Communications below 135 kHz
  - Part 3 : Parameters for Air Interface Communications at 13.56 MHz
  - Part 4 : Parameters for Air Interface Communications at 2.45 GHz
  - Part 5 : Not available
  - Part 6 : Parameters for Air Interface Communications at 860 to 960 MHz
  - Part 7 : Parameters for Air Interface Communications at 433 MHz

- ISO 18185 : RFID for electronic seal tags (ISO TC 104 - Freight Containers)

- ISO/IEC 19762 : Information technology – Automatic identification and data capture (AIDC) techniques – Harmonized vocabulary

  - Part 3: Radio frequency identification (RFID)

- ISO 23389 : Freight Containers - Read-Write Radio-frequency identifications (RFID) (ISO TC 104)

- ISO/IEC 24710 : Information technology, automatic identification and data capture techniques Radio frequency identification for item management Elementary tag license plate functionality for ISO/IEC 18000 air interface definitions

- ISO/IEC 24729 : Information technology Radio frequency identification for item management Implementation guidelines

  - Part 1 : RFID-enabled labels

  - Part 2 : Recyclability of RF tags
  - Part 3 : RFID interrogator/ antenna installation
- ISO/IEC 24730 : Real Time Locating Systems (RTLS)
  - Part 1 : Application programming interface(API)
  - Part 2 : 2.4 GHz
  - Part 3 : 433 MHz
  - Part 4 : Global Locating Systems (GLS)
- ISO/IEC 24752 : Information technology - Automatic Identification and Data Capture Techniques- Radio Frequency Identification (RFID) for Item Management - System Management Protocol
- ISO/IEC 24753 : Information Technology - Automatic Identification and Data Capture Techniques - Radio Frequency Identification (RFID) for Item Management - Air Interface Commands for Battery Assist and Sensor Functionality
- ISO/IEC 24769 : Information Technology, Automatic Identification and Data Capture Techniques - Real Time Locating Systems (RTLS) - RTLS Device Conformance Test Methods
- ISO/IEC 24770 : Information Technology, Automatic Identification and Data Capture Techniques - Real Time Locating Systems (RTLS) - RTLS Device Performance Test Methods
- ASTM D7434 : Standard Test Method for Determining the Performance of Passive Radio Frequency Identification (RFID) Transponders on Palletized or Unitized Loads
- ASTM D7435 : Standard Test Method for Determining the Performance of Passive Radio Frequency Identification (RFID) Transponders on Loaded Containers
- ASTM D7580 : Standard Test Method for Rotary Stretch Wrapper Method for Determining the Readability of Passive RFID Transponders on Homogeneous Palletized or Unitized Loads
- DASH7 Alliance : An international industry group formed in 2009 to promote standards and interoperability among extensions to ISO/IEC 18000-7 technologies.

- EPCglobal : this is the standardization framework that is most likely to undergo international standardization according to ISO rules as with all sound standards in the world, unless residing with limited scope, as customs regulations, air-traffic regulations and others. Currently the big distributors and governmental customers are pushing EPC heavily as a standard well-accepted in their community, but not yet regarded as for salvation to the rest of the world (Class 0, Class 1 and Class 1 Gen 2 standard compliant tags).

Appendix B

Appendix of Chapter 3

B.1 Formal Analysis of SIDRFID

We carry out formal analysis of SIDRFID using Casper-FDR tools as explained earlier in Section 2.6 for possible attacks. We describe SIDRFID in Casper as follows:

Protocol with Static Identity SIDRFID

#Free variables
T,R : Agent
ri : Nonce
idr,idt : IdenKeys
InverseKeys = (idr,idr),(idt,idt)

#Processes
INITIATOR(R,ri,idr)
RESPONDER(T,idr,idt)

#Protocol description
0. $\rightarrow R : T$
$[R! = T]$
1. $R \rightarrow T : \{ri\}\{idr\}$
2a. $T \rightarrow R : \{ri, idt\}\{idr\}$
2b. $T \rightarrow R : \{ri\}\{idt\}$
3a. $R \rightarrow T : \{idr, ri\}\{idt\}$
3b. $R \rightarrow T : \{idt, ri\}\{idr\}$

#Specification
Secret(R,idr,[T])
Secret(T,idt,[R])
Agreement(T,R,[idr,idt])
Agreement(R,T,[idr,idt])

#Actual variables
Tag,Reader,Mallory : Agent
Ri,Rm : Nonce
IDR,IDT,IDM : IdenKeys
InverseKeys = (IDR,IDR),(IDT,IDT),(IDM,IDM)

#Functions

#System
INITIATOR(Reader,Ri,IDR)
RESPONDER(Tag,IDR,IDT)

#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Tag,Reader,Mallory,Rm,IDM}

Note here that we have not calculated actual values as in the protocol and used *idr* and *idt* as shared secrets since this analysis does not involve any cryptographic attack. The testing is carried out for desired specifications as follows:

1. **Secret(R,idr,[T]) and Secret(T,idt,[R])** : The tag and the reader share *idr* and *idt* as secrets. We have already carried out a full disclosure attack to reveal these secret values as shown in Section 3.3.2.

2. **Agreement(T,R,[idr,idt])** : The tag should be authenticated to the reader and both agree on the values of *idr* and *idt*. However FDR discovers following attack:

   0. $\rightarrow Reader : Tag$
   1. $Reader \rightarrow I\_\{Tag\} : \{Ri\}\{IDR\}$
   1. $I\_\{Mallory\} \rightarrow Tag : \{Ri\}\{IDR\}$
   2a. $Tag \rightarrow I\_\{Mallory\} : \{Ri, IDT\}\{IDR\}$
   2a. $I\_\{Tag\} \rightarrow Reader : \{Ri, IDT\}\{IDR\}$

2b. $Tag \rightarrow I\_\{Mallory\} : \{Ri\}\{IDT\}$
Tag believes (s)he is running the protocol, taking role RESPONDER, with Mallory, using data items IDR, IDT.
2b. $I\_\{Tag\} \rightarrow Reader : \{Ri\}\{IDT\}$
3a. $Reader \rightarrow I\_\{Tag\} : \{IDR, Ri\}\{IDT\}$
3b. $Reader \rightarrow I\_\{Tag\} : \{IDT, Ri\}\{IDR\}$
Reader believes (s)he has completed a run of the protocol, taking role INITIATOR, with Tag, using data items IDR, IDT.

3. **Agreement(R,T,[idr,idt])** : The reader should be successfully authenticated to the tag with both parties agreeing on the values of *idr* and *idt*. When this specification is checked using FDR, following attack is discovered:

   0. $\rightarrow Reader : Mallory$
   1. $Reader \rightarrow I\_\{Mallory\} : \{Ri\}\{IDR\}$
   1. $I\_\{Tag\} \rightarrow Tag : \{Ri\}\{IDR\}$
   2a. $Tag \rightarrow I\_\{Tag\} : \{Ri, IDT\}\{IDR\}$
   2a. $I\_\{Mallory\} \rightarrow Reader : \{Ri, IDT\}\{IDR\}$
   2b. $Tag \rightarrow I\_\{Tag\} : \{Ri\}\{IDT\}$
   2b. $I\_\{Mallory\} \rightarrow Reader : \{Ri\}\{IDT\}$
   3a. $Reader \rightarrow I\_\{Mallory\} : \{IDR, Ri\}\{IDT\}$
   3a. $I\_\{Tag\} \rightarrow Tag : \{IDR, Ri\}\{IDT\}$
   3b. $Reader \rightarrow I\_\{Mallory\} : \{IDT, Ri\}\{IDR\}$
   3b. $I\_\{Tag\} \rightarrow Tag : \{IDT, Ri\}\{IDR\}$
   Reader believes (s)he is running the protocol, taking role INITIATOR, with Mallory, using data items IDR, IDT. Tag believes (s)he has completed a run of the protocol, taking role RESPONDER, with Tag, using data items IDR, IDT.

Since these specifications are failed with attacks discovered on SIDRFID, we thus formally verify that this is a weak protocol.

B.2 Formal Analysis of DIDRFID

We now analze DIDRFID using Casper-FDR to discover any attacks on the protocol. DIDRFID in Casper is described as follows:

Protocol with Dynamic Identity DIDRFID

#Free variables
T,R : Agent
ri : Nonce
idt : SessionIdenTag
ki : SessionKey
InverseKeys = (ki,ki)

#Processes
INITIATOR(T,idt,ki)
RESPONDER(R,idt,ki,ri)

#Protocol description
0. $\rightarrow T : R$
$[T! = R]$
1. $T \rightarrow R : idt$
2a. $R \rightarrow T : \{ri\}\{ki\}$
2b. $R \rightarrow T : \{ri, ki\}\{ki\}$
3. $T \rightarrow R : \{ki, ri\}\{ki\}$

#Specification
Secret(R,ki,[T])
Secret(T,ki,[R])
Agreement(R,T,[ri,ki])
Agreement(T,R,[ri,ki])

#Actual variables
Tag,Reader,Mallory : Agent
Ri,Rm : Nonce
IDT,IDM : SessionIdenTag
Ki,Km : SessionKey
InverseKeys = (Ki,Ki),(Km,Km)

#Functions

#System
INITIATOR(Tag,IDT,Ki)
RESPONDER(Reader,IDT,Ki,Ri)

#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Tag,Reader,Mallory,Rm,IDM,Km}

The messages are constructed based on the values of ri and ki to simulate the actual protocol. The desired specifications are tested as follows:

1. **Secret(R,ki,[T]) and Secret(T,ki,[R])** : The tag and the reader share ki as a secret key. As shown in Section 3.4.1, this secret key can be compromised.

2. **Agreement(R,T,[ri,ki])** : The reader should be authenticated to the tag successfully and both use the values of ri and ki. Following attack is discovered on this specification:

   0. $\rightarrow Tag : Reader$
   1. $Tag \rightarrow I\_\{Reader\} : IDT$
   1. $I\_\{Mallory\} \rightarrow Reader : IDT$
   2a. $Reader \rightarrow I\_\{Mallory\} : \{Ri\}\{Ki\}$
   2a. $I\_\{Reader\} \rightarrow Tag : \{Ri\}\{Ki\}$
   2b. $Reader \rightarrow I\_\{Mallory\} : \{Ri, Ki\}\{Ki\}$
   Reader believes (s)he is running the protocol, taking role RESPONDER, with Mallory, using data items Ri, Ki.
   2b. $I\_\{Reader\} \rightarrow Tag : \{Ri, Ki\}\{Ki\}$
   3. $Tag \rightarrow I\_\{Reader\} : \{Ki, Ri\}\{Ki\}$
   Tag believes (s)he has completed a run of the protocol, taking role INITIATOR, with Reader, using data items Ri, Ki.

3. **Agreement(T,R,[ri,ki])** : The tag should be authenticated to the reader and both agree on the values of ri and ki. Testing shows that following attack exists on this specification:

   0. $\rightarrow Tag : Mallory$
   1. $Tag \rightarrow I\_\{Mallory\} : IDT$
   1. $I\_\{Tag\} \rightarrow Reader : IDT$
   2a. $Reader \rightarrow I\_\{Tag\} : \{Ri\}\{Ki\}$
   2a. $I\_\{Mallory\} \rightarrow Tag : \{Ri\}\{Ki\}$
   2b. $Reader \rightarrow I\_\{Tag\} : \{Ri, Ki\}\{Ki\}$
   2b. $I\_\{Mallory\} \rightarrow Tag : \{Ri, Ki\}\{Ki\}$

Tag believes (s)he is running the protocol, taking role INITIATOR, with Mallory, using data items Ri, Ki.

3. $Tag \rightarrow I\_\{Mallory\} : \{Ki, Ri\}\{Ki\}$

3. $I\_\{Tag\} \rightarrow Reader : \{Ki, Ri\}\{Ki\}$

Reader believes (s)he has completed a run of the protocol, taking role RESPONDER, with Tag, using data items Ri, Ki.

Attacks are discovered on DIDRFID, we thus formally verify that this is also a weak protocol.

Appendix C

Appendix of Chapter 4

C.1 Formal Analysis of Proposed UMAP

As explained in Section 2.6, we use Casper-FDR tools to carry out formal analysis of our suggested scheme for any possible attacks. We describe our UMAP in Casper as follows:

Ultra-lightweight Mutual Authentication Protocol

#Free variables
T,R : Agent
S : Server
hello,rn : Nonce
secretID : TagSecretId
ks : SessionKey
InverseKeys = (ks,ks)

#Processes
INITIATOR(R,S,hello)
RESPONDER(T,S,secretID,ks)
SERVER(S,rn,secretID,ks)

#Protocol description
0. $\rightarrow R : T$
$[R! = T]$
1. $R \rightarrow T : hello$
2. $T \rightarrow R : T$

3. $R \rightarrow S : T$

4a. $S \rightarrow R : rn$
4b. $S \rightarrow R : \{rn, T, secretID\}\{ks\}\%A$
5a. $R \rightarrow T : rn$
5b. $R \rightarrow T : A\%\{rn, T, secretID\}\{ks\}$
6. $T \rightarrow R : \{T, secretID, rn\}\{ks\}\%B$
7. $R \rightarrow S : B\%\{T, secretID, rn\}\{ks\}$

#Specification
Secret(T,ks,[S])
Agreement(S,T,[secretID,ks])
Agreement(T,S,[rn,ks])

#Actual variables
Tag,Reader,Mallory : Agent
Sam : Server
Hello,Hellom,Rn,Rm : Nonce
SecretID,IDm : TagSecretId
Ks,Km : SessionKey
InverseKeys = (Ks, Ks),(Km, Km)

#Functions

#System
INITIATOR(Reader,Sam,Hello)
RESPONDER(Tag,Sam,SecretID,Ks)
SERVER(Sam,Rn,SecretID,Ks)

#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Tag,Reader,Sam,Mallory,Rm,IDm,Km}

Note here that we have not calculated internal secrets and used rn as it is easy for implementation. Similarly the generation of messages A and B is carried out by encryption of variables using shared secret key ks. The $Index$ of the tag is replaced by T which is a public value. The testing is carried out for desired specification as follows:

1. **Secret(T,ks,[S])** : The tag and the server share a secret key as ks.

2. **Agreement(S,T,[secretID,ks])** : The server is successfully authenticated to the tag after message A is successfully verified and both parties agree on the values of $secretID$ and ks.

3. **Agreement(T,S,[rn,ks])** : The tag is successfully authenticated to the server after message B is successfully verified and both parties agree on the values of rn and ks.

Since these specifications are passed without any attack using FDR2, our suggested scheme is verified to achieve desired functionality.

Appendix D

Appendix of Chapter 6

D.1 Formal Analysis of Hierarchical Anti-counterfeit Mechanism

The Casper-FDR tools as explained in Section 2.6 are used to carry out the formal analysis of our suggested scheme. We shall analyze the product verification phase only. This analysis will also verify that if no attacks exist on product verification phase then group verification phase does not encounter any attacks. The code compiled by Casper is as follows:

Product Verification Phase : Anti-Counterfeit Mechanism

#Free variables
T,R : Agent
S : Server
q,rn16,rand : Nonce
gid : GroupId
secID : TagSecretId
kt,kr : SessionKey
ts,ts1 : TimeStamp
InverseKeys = (kt,kt),(kr,kr)

#Processes
INITIATOR(R,S,q,kr)
RESPONDER(T,S,secID,rn16,gid,kt)
SERVER(S,rand,secID,gid,kt,kr)

#Protocol description
0. $\rightarrow R : T$
$[R! = T]$
1. $R \rightarrow T : q$
2. $T \rightarrow R : rn16$
3. $R \rightarrow T : rn16$
4. $T \rightarrow R : T, gid$
---
5. $R \rightarrow S : T, gid$
6. $S \rightarrow R : rand$
---
7. $R \rightarrow T : rand$
8. $T \rightarrow R : \{rand, secID, T, ts\}\{kt\}\%tvc$

9a. $R \rightarrow S : tvc\%\{rand, secID, T, ts\}\{kt\}$
9b. $R \rightarrow S : \{rand, R, ts1\}\{kr\}$
$[ts + 1 == now$ or $ts + 2 == now$ and
$ts1 == now$ or $ts1 + 1 == now]$

#Specification
Secret(T,kt,[S])
Secret(R,kr,[S])
TimedAgreement(T,S,3,[secID,kt])
TimedAgreement(R,S,2,[rand,kr])

#Actual variables
Tag,Reader,Mallory : Agent
Sam : Server
Q,Rn16,Rand,Rm : Nonce
GID, Gm : GroupId
SecretID,IDm : TagSecretId
Kt,Kr,Km : SessionKey
InverseKeys = (Kt,Kt),(Kr,Kr),(Km,Km)
TimeStamp = 0 .. 0
MaxRunTime = 0

#Functions

#System
INITIATOR(Reader,Sam,Q,Kr)
RESPONDER(Tag,Sam,SecretID,Rn16,GID,Kt)
SERVER(Sam,Rand,SecretID,GID,Kt,Kr)

#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Tag,Reader,Sam,Mallory,Rm,Gm,IDm,Km}

Product verification code involves all the variables as in the original scheme. However the construction is different for ease of implementation and considering that this analysis does not involve cryptographic attacks. The desired specification are tested as follows:

1. **Secret(T,kt,[S])** : The tag and the server share a secret key as kt.
2. **Secret(R,kr,[S])** : The reader and the server share a secret key as kr.
3. **TimedAgreement(T,S,3,[secID,kt])** : The tag is successfully authenticated to the server after tvc is successfully verified and both parties agree on the values of $secID$ and kt. Moreover, this authentication should be completed in three time units to avoid relay attack (simulating t_{out} assumption in our scheme).
4. **TimedAgreement(R,S,2,[rand,kr])** : The reader is successfully authenticated to the server after $rand$ is correctly decrypted and both parties agree on the values of $rand$ and kr. This authentication should be completed in two time units to avoid any relay attack (simulating t_{out} assumption in our scheme).

When CSP file is loaded in FDR2, no attack is detected.

Appendix E

Appendix of Chapter 7

E.1 Formal Analysis of Customer Level Counterfeit Detection Scheme

We carry out the formal analysis of the suggested scheme. The tools used are as explained in Section 2.6. This analysis will determine whether the suggested scheme achieves its goal without encountering any attacks. The Casper code is as follows:

Customer Level Counterfeit Detection Scheme

#Free variables
T,R : Agent
rand,hello : Nonce
secID : TagSecretId
pkt : TagPublicKey
skt : TagSecretKey
ksign : SignatureKey
kverify : VerificationKey
InverseKeys = (pkt,skt),(ksign,kverify)

#Processes
INITIATOR(R,kverify,rand,hello)
RESPONDER(T,secID,pkt,skt,ksign)

#Protocol description
0. $\rightarrow R : T$

$[R! = T]$
1. $R \rightarrow T : hello$
2. $T \rightarrow R : secID, T, pkt, \{secID, T, pkt\}\{ksign\}$
3. $R \rightarrow T : rand, R$
4. $T \rightarrow R : \{rand, R\}\{skt\}$

#Specification
Agreement(T,R,[rand,pkt])

#Actual variables
Tag,Reader,Mallory : Agent
Rand,Rm,Hello : Nonce
SecretID,IDm : TagSecretId
PKt,PKm : TagPublicKey
SKt,SKm : TagSecretKey
Ksign,Ksignm : SignatureKey
Kverify,Kverifym : VerificationKey
InverseKeys = (PKt,SKt),(Ksign,Kverify),(PKm,SKm),(Ksignm,Kverifym)

#Functions

#System
INITIATOR(Reader,Kverify,Rand,Hello)
RESPONDER(Tag,SecretID,PKt,SKt,Ksign)

#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Tag,Reader,Mallory,Rm,IDm,PKm,SKm,Ksignm,Kverifym}

We have included the identity of the reader in messages 3 and 4 considering that the verification process is being carried out by a party who does not need to authenticate itself first. Visual verification can be replayed by the adversary, however cryptographic verification cannot be replayed because of the random challenge generated by a legitimate user. The desired specification is tested as follows:

1. **Agreement(T,R,[rand,pkt])** : The tag is successfully authenticated to the reader after a successful challenge-response protocol and both parties agree on the values of *rand* and *pkt*.

The above mentioned Casper code is compiled to generate a CSP file which is further loaded in FDR2. The specification passes successfully without any attack.

Appendix F

Appendix of Chapter 8

F.1 Formal Analysis of Tag Ownership Release Phase

We use CasperFDR as explained in Section 2.6 to formally analyze our ownership transfer scheme. We analyze our tag ownership release phase for any possible attacks. The following complete Casper code is compiled to generate a CSP and loaded into FDR2:

Tag Ownership Release Phase

#Free variables
T,O : Agent
S : Server
q,rn,rands : Nonce
rto,setot : Flag
access : InitialSeq
kold,ks,kso,ktx : SessionKey
ts,ts1,ts2,ts3,ts4 : TimeStamp
InverseKeys = (kold,kold),(ks,ks),(kso,kso),(ktx,ktx)

#Processes
INITIATOR(O,S,q,kold,setot,rto,kso)
RESPONDER(T,rn,kold,access,setot,ktx)
SERVER(S,rands,kso,access,ks,ktx)

#Protocol description
0. $\rightarrow O : T$

$[O! = T]$
1. $O \rightarrow T : q$
2. $T \rightarrow O : rn$
3. $O \rightarrow T : rn, O$
4. $T \rightarrow O : \{rn, T, O\}\{kold\}$

---

5. $O \rightarrow S : rto, O$
6. $S \rightarrow O : rands, S$
7. $O \rightarrow S : \{rands, T, O, S, ts\}\{kso\}$
$[ts == now$ or $ts + 1 == now]$
8. $S \rightarrow O : \{T, O, S, ks, access, ts1\}\{kso\}$
$[ts1 == now$ or $ts1 + 1 == now]$

9a. $O \rightarrow T : rands, S$
9b. $O \rightarrow T : \{T, O, access, ks, setot, ts2\}\{kold\}$
$[ts2 == now$ or $ts2 + 1 == now]$

---

10. $T \rightarrow O : \{rands, T, O, S, ts3\}\{ktx\}\%ku$
11a. $O \rightarrow S : ku\%\{rands, T, O, S, ts3\}\{ktx\}$
11b. $O \rightarrow S : \{O, ts4\}\{kso\}$
$[ts4 == now$ or $ts4 + 1 == now$ and
$ts3 + 1 == now$ or $ts3 + 2 == now]$

#Specification
Secret(O,kold,[T])
Secret(O,kso,[S])
Secret(T,ktx,[S])
Agreement(T,O,[rn,kold])
TimedAgreement(O,T,2,[access,ks,kold])
TimedAgreement(O,S,2,[rands,kso])
TimedAgreement(S,O,2,[kso])
TimedAgreement(T,S,3,[rands,ktx])

#Actual variables
Tag,OldOwner,Mallory : Agent
Sam : Server
Query,RN16,Rands,Nm : Nonce

RTO,SETOT : Flag
Access,Accessm : InitialSeq
Kold,Ks,Kso,Ktx,Km : SessionKey
InverseKeys = (Kold, Kold),(Ks, Ks),(Kso, Kso),(Ktx, Ktx),(Km, Km)
TimeStamp = 0 .. 0
MaxRunTime = 0

#Functions

#System
INITIATOR(OldOwner,Sam,Query,Kold,SETOT,RTO,Kso)
RESPONDER(Tag,RN16,Kold,Access,SETOT,Ktx)
SERVER(Sam,Rands,Kso,Access,Ks,Ktx)

#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Tag,OldOwner,Sam,Mallory,Nm,Accessm,Km,RTO,SETOT}

We divide the complete phase into following stages for testing purposes.

F.1.1 First Stage

The old owner communicates with the tag using standard protocol as follows:
1. $O \rightarrow T : q$
2. $T \rightarrow O : rn$
3. $O \rightarrow T : rn, O$
4. $T \rightarrow O : \{rn, T, O\}\{kold\}$

This achieves two goals:

- To determine that the tag and the old owner share a secret key as *kold*.
- The tag is authenticated successfully to the old owner.

Using CasperFDR analysis, these two goals are achieved successfully as following two specifications are passed without any attack:

- **Secret(O,kold,[T])** : The old owner thinks that *kold* is a secret that can be known to only himself and the tag.

- **Agreement(T,O,[rn,kold])** : The tag is correctly authenticated to the old owner, and the two agents agree on the data values rn and $kold$.

F.1.2 Second Stage

The old owner communicates with the server as follows:
1. $O \rightarrow S : rto, O$
2. $S \rightarrow O : rands, S$
3. $O \rightarrow S : \{rands, T, O, S, ts\}\{kso\}$
$[ts == now$ or $ts + 1 == now]$
4. $S \rightarrow O : \{T, O, S, ks, access, ts1\}\{kso\}$
$[ts1 == now$ or $ts1 + 1 == now]$

This achieves two goals:

- To determine that the server and the old owner share a secret key as kso.
- The old owner and the server are successfully mutually authenticated.

Using CasperFDR analysis, these two goals are achieved successfully as following specifications are passed without any attack:

- **Secret(O,kso,[S])** : The old owner thinks that kso is a secret that can be known to only himself and the server.
- **TimedAgreement(O,S,2,[rands,kso])** : The old owner is correctly authenticated to the server within two time units (one for processing and one for checking), and the two agents agree on the data values $rands$ and kso.
- **TimedAgreement(S,O,2,[kso])** : The server is correctly authenticated to the old owner within two time units (one for processing and one for checking), and the two agents agree on the data value kso.

F.1.3 Third Stage

The old owner communicates with the tag as follows:
1a. $O \rightarrow T : rands, S$
1b. $O \rightarrow T : \{T, O, access, ks, setot, ts2\}\{kold\}$
$[ts2 == now$ or $ts2 + 1 == now]$

This achieves following goal:

- The old owner is successfully authenticated to the tag.

Using CasperFDR analysis, this goal is achieved successfully as following specification is passed without any attack:

- **TimedAgreement(O,T,2,[access,ks,kold])** : The old owner is correctly authenticated to the tag within two time units (one for processing and one for checking), and the two agents agree on the data values *access*, *ks* and *kold*.

F.1.4 Fourth Stage

The tag sends its update message to old owner which forwards it to the server along with its own acknowledgment message as follows:
1. $T \rightarrow O : \{rands, T, O, S, ts3\}\{ktx\}\%ku$
2a. $O \rightarrow S : ku\%\{rands, T, O, S, ts3\}\{ktx\}$
2b. $O \rightarrow S : \{O, ts4\}\{kso\}$
$[ts4 == now$ or $ts4 + 1 == now$ and
$ts3 + 1 == now$ or $ts3 + 2 == now]$

This stage achieves following goals:

- To determine that the tag and the server share a secret key as *ktx*.
- The tag is successfully authenticated to the server.

Using CasperFDR analysis, these goal are achieved successfully as following specifications are passed without any attack:

- **Secret(T,ktx,[S])** : The tag thinks that *ktx* is a secret that can be known to only himself and the server.
- **TimedAgreement(T,S,3,[rands,ktx])** : The tag is correctly authenticated to the server within three time units (one for forwarding by the old owner, one for processing and one for checking), and the two agents agree on the data values of *rands* and *ktx*.

F.2 Formal Analysis of Tag Ownership Acquire Phase

We now formally analyze tag ownership acquire phase using CasperFDR for any possible attacks. Following Casper code is compiled to generate a CSP and loaded into FDR2:

Tag Ownership Acquire Phase

#Free variables
T,O : Agent
S : Server
q,rn,rands : Nonce
ato,rstot : Flag
access : InitialSeq
kso,ktx,knew : SessionKey
ts,ts1,ts2,ts3,ts4,ts5,ts6 : TimeStamp
InverseKeys = (kso,kso),(ktx,ktx),(knew,knew)

#Processes
INITIATOR(O,S,q,rstot,ato,kso)
RESPONDER(T,rn,access,rstot,ktx)
SERVER(S,rands,kso,access,ktx,knew)

#Protocol description
0. $\rightarrow O : T$
$[O! = T]$
1. $O \rightarrow S : ato, O$
2. $S \rightarrow O : rands, S$
3. $O \rightarrow S : \{rands, T, O, S, ts\}\{kso\}$
$[ts == now$ or $ts + 1 == now]$
4. $S \rightarrow O : \{T, O, S, ktx, ts1\}\{kso\}$
$[ts1 == now$ or $ts1 + 1 == now]$
---
5. $O \rightarrow T : q$
6. $T \rightarrow O : rn$
7. $O \rightarrow T : rn, O$
8. $T \rightarrow O : \{rn, T, O\}\{ktx\}$
---
9. $O \rightarrow S : \{T, O, S, ts2\}\{kso\}$
$[ts2 == now$ or $ts2 + 1 == now]$
10. $S \rightarrow O : \{T, O, S, knew, access, ts3\}\{kso\}$
$[ts3 == now$ or $ts3 + 1 == now]$

———————————————-

11a. $O \rightarrow T : rands, S$

11b. $O \rightarrow T : \{T, O, access, knew, rstot, ts4\}\{ktx\}$

$[ts4 == now$ or $ts4 + 1 == now]$

———————————————-

12. $T \rightarrow O : \{rands, T, O, S, ts5\}\{knew\}\%ku$

13a. $O \rightarrow S : ku\%\{rands, T, O, S, ts5\}\{knew\}$

13b. $O \rightarrow S : \{O, ts6\}\{kso\}$

$[ts5 + 1 == now$ or $ts5 + 2 == now$ and

$ts6 == now$ or $ts6 + 1 == now]$

#Specification

Secret(O,ktx,[T])

Secret(O,kso,[S])

Secret(T,knew,[S])

Secret(T,knew,[O])

Agreement(T,O,[rn,ktx])

TimedAgreement(O,T,2,[access,knew,ktx])

TimedAgreement(O,S,2,[rands,kso])

TimedAgreement(S,O,2,[kso])

TimedAgreement(T,S,3,[rands,knew])

#Actual variables

Tag,NewOwner,Mallory : Agent

Sam : Server

Query,RN16,Rands,Nm : Nonce

ATO,RSTOT : Flag

Access,Accessm : InitialSeq

Kso,Ktx,Knew,Km : SessionKey

InverseKeys = (Kso,Kso),(Ktx,Ktx),(Knew,Knew),(Km,Km)

TimeStamp = 0 .. 0

MaxRunTime = 0

#Functions

#System

INITIATOR(NewOwner,Sam,Query,RSTOT,ATO,Kso)

RESPONDER(Tag,RN16,Access,RSTOT,Ktx)
SERVER(Sam,Rands,Kso,Access,Ktx,Knew)

#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Tag, NewOwner, Sam, Mallory, Nm, Accessm, Km, ATO, RSTOT}

The acquire phase is divided into following stages for testing purposes.

F.2.1 First Stage

The new owner starts tag ownership acquire phase by communicating with the server as follows:
1. $O \rightarrow S : ato, O$
2. $S \rightarrow O : rands, S$
3. $O \rightarrow S : \{rands, T, O, S, ts\}\{kso\}$
$[ts == now$ or $ts + 1 == now]$
4. $S \rightarrow O : \{T, O, S, ktx, ts1\}\{kso\}$
$[ts1 == now$ or $ts1 + 1 == now]$

Concluding this achieves two goals:

- A secret key as kso is shared between the server and the new owner.
- Both the new owner and the server are mutually authenticated successfully.

Using CasperFDR analysis, these two goals are achieved successfully as following specifications are passed without any attack:

- **Secret(O,kso,[S])** : The new owner thinks that kso is a secret that can be known to only himself and the server.
- **TimedAgreement(O,S,2,[rands,kso])** : The new owner is correctly authenticated to the server within two time units (one for processing and one for checking), and the two agents agree on the data values $rands$ and kso.
- **TimedAgreement(S,O,2,[kso])** : The server is correctly authenticated to the new owner within two time units (one for processing and one for checking), and the two agents agree on the data value kso.

F.2.2 Second Stage

The new owner uses standard protocol to communicate with the tag:
1. $O \rightarrow T : q$
2. $T \rightarrow O : rn$
3. $O \rightarrow T : rn, O$
4. $T \rightarrow O : \{rn, T, O\}\{ktx\}$

This is carried out to achieve following goals:

- To determine a secret key ktx is shared between the tag and the new owner.
- The tag is authenticated successfully to the new owner.

Using CasperFDR analysis, these two goals are achieved successfully as following two specifications are passed without any attack:

- **Secret(O,ktx,[T])** : The new owner thinks that ktx is a secret that can be known to only himself and the tag.
- **Agreement(T,O,[rn,ktx])** : The tag is correctly authenticated to the new owner, and the two agents agree on the data values rn and ktx.

F.2.3 Third Stage

After achieving the values of $knew$ and $access$, the new owner now communicates with the tag as follows:
1. $O \rightarrow T : rands, S$
2. $O \rightarrow T : \{T, O, access, knew, rstot, ts4\}\{ktx\}$
$[ts4 == now$ or $ts4 + 1 == now]$

This achieves following goal:

- The new owner and tag now shares a secret key as $knew$.
- The new owner is successfully authenticated to the tag.

Using CasperFDR analysis, this goal is achieved successfully as following specification is passed without any attack:

- **Secret(T,knew,[O])** : The tag thinks that $knew$ is a secret that is known to both himself and the new owner.

- **TimedAgreement(O,T,2,[access,knew,ktx])** : The new owner is correctly authenticated to the tag within two time units (one for processing and one for checking), and the two agents agree on the data values *access*, *knew* and *ktx*.

F.2.4 Fourth Stage

The tag sends its update message to old owner which forwards it to the server along with its owner acknowledgment message.
1. $T \rightarrow O : \{rands, T, O, S, ts5\}\{knew\}\%ku$
2a. $O \rightarrow S : ku\%\{rands, T, O, S, ts5\}\{knew\}$
2b. $O \rightarrow S : \{O, ts6\}\{kso\}$
$[ts5 + 1 == now$ or $ts5 + 2 == now$ and
$ts6 == now$ or $ts6 + 1 == now]$

This stage achieves following goals:

- To determine that the tag and the server share a secret key as *knew*.
- The tag is successfully authenticated to the server.

Using CasperFDR analysis, these goal are achieved successfully as following specifications are passed without any attack:

- **Secret(T,knew,[S])** : The tag thinks that *knew* is a secret that can be known to only himself and the server.
- **TimedAgreement(T,S,3,[rands,knew])** : The tag is correctly authenticated to the server within three time units (one for forwarding by the new owner, one for processing and one for checking), and the two agents agree on the data values of *rands* and *knew*.