Doctoral Thesis
박사학위논문

Secure and Privacy-Preserving Protocols for RFID Applications

RFID 응용을 위한 프라이버시 보호 및 보안 프로토콜

Konidala, Divyan Munirathnam (코니달라 디비안 무니라트남)
정보통신공학과
Department of Information and Communications Engineering

한국과학기술원
Korea Advanced Institute of Science and Technology (KAIST)

2011

Secure and Privacy-Preserving Protocols for RFID Applications

RFID 응용을 위한 프라이버시 보호 및 보안 프로토콜

Secure and Privacy-Preserving Protocols for RFID Applications

Advisor : Professor Kim, Kwangjo

by

Konidala, Divyan Munirathnam
Department of Information and Communications Engineering
Korea Advanced Institute of Science and Technology (KAIST)

A thesis submitted to the faculty of the Korea Advanced Institute of Science and Technology (KAIST) in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Information and Communications Engineering

Daejeon, Korea
2010. 12. 08.
Approved by

Professor Kim, Kwangjo
Advisor

Secure and Privacy-Preserving Protocols for RFID Applications

Konidala, Divyan Munirathnam

The present thesis has been approved by the thesis committee as a Doctoral Thesis at the Korea Advanced Institute of Science and Technology (KAIST)

2010. 12. 08.

Committee chair	Kim, Kwangjo	____________
Committee member	Kim, Daeyoung	____________
Committee member	Kim, Soontae	____________
Committee member	Lee, Byoungcheon	____________
Committee member	Choi, Dooho	____________

DICE
20045328
Konidala, Divyan Munirathnam. 코니달라 디비안 무니라트남. Secure and Privacy-Preserving Protocols for RFID Applications. RFID 응용을 위한 프라이버시 보호 및 보안 프로토콜. Department of Information and Communications Engineering . 2011. 117p. Advisor Prof. Kim, Kwangjo. Text in English.

Abstract

Radio Frequency Identification (RFID) is a means to efficiently and quickly, auto-identify objects, assets, pets, people, *etc.* RFID technology is poised to automate the supply chain management system for businesses. Very soon it would become economical to attach RFID tags to items and consumer goods. As a result we can anticipate many electronic appliances including mobile/smart phones to have a RFID-reader-tag component/chip embedded in them. Such developments would allow RFID technology to also assist people in their daily lives. In this thesis we focus on three specific RFID applications that are beneficial to businesses and consumers.

At the outset this thesis focuses on RFID-based supply chain management system that adheres to the standard EPCglobal Architecture Framework specification [11]. We approach this framework with a security point-of-view. The EPCglobal Architecture Framework is composed of entities like RFID Tag, RFID Reader, RFID Middleware, Electronic Product Code Information Service (EPCIS) Repository, EPCIS Accessing Application, Object Naming Service, and Subscriber Authentication. We analyze the various security threats that affect each of these entities and their communication interfaces. We arrive at some possible security requirements and needed security solutions to ward off these threats.

After our thorough security assessment of the RFID-based supply chain management system, we narrowed down our research focus on RFID: Tag←Reader→Server/EPCIS security. We realized that cloned fake RFID tags, malicious RFID readers and consumer privacy violation pose a major threat to RFID-based supply chain management system. Fake tags can be attached to counterfeit products and medicines. Malicious readers can corrupt genuine tags and mount man-in-middle attacks on the communication channel between genuine tags and readers. A consumer carrying a tagged item can be identified, tracked and traced based solely on the tag's unique identity number. To deal with these security problems, the use of cryptographic protocols is required. However, designing cryptographic protocols for RFID tags is a challenging task as a low-cost RFID tag has very limited computational resources. As a result, in this thesis we propose two light-weight cryptographic protocols which use only light-weight primitives such as 16 bit: random number generator, cyclic redundancy check and exclusive-OR (XOR) functions and pro-

vide RFID tag←reader→Server/EPCIS mutual authentication, communicating-data confidentiality and integrity, secure key-distribution and key-protection, and tag anonymity.

The promising and beneficial RFID technology would eventually lead to the development and deployment of electronic appliances and devices that are RFID-reader-tag-enabled, *e.g.,* RFID-reader-enabled book shelves, mobile/smart phones and PDAs. Here the devices and objects, dispersed through our surroundings, can identify and communicate with each other, providing real-time information about themselves, locations, and ambient conditions around them. Therefore RFID technology will have a tremendous impact on our society assisting people in their daily lives. A right step in this direction is Mobile-RFID (mRFID) technology, where a mobile/smart phone apart from having the usual voice/data communication facilities would also have an embedded RFID-tag-reader chip thus allowing to behave as a RFID reader and a tag.

Mobile payment is a payment method, where a mobile phone is used to pay for merchandize and services. Mobile payment is gaining popularity especially in Asia and Europe. Currently, efforts are being put to deploy a mobile payment model that precisely mimics the Contactless (RFID) Card Payment model, where an mRFID-enabled mobile phone behaves as a contactless credit/debit card. However this thesis emphasizes that credit/debit card payment transactions do not protect the privacy of the customer. Once the card is handed over to the merchant for payment processing, customers are "no longer in control" on how their card details and money are handled. This leads to card fraud, identity theft, and customer profiling. Therefore for those customers who value their privacy and security of their payment transactions, this thesis proposes a choice - an alternate mobile payment model called "Pre-Paid Mobile HTTPS-based Payment model". In the proposed payment model the customer obtains the merchant's bank account information into his/her mRFID-enabled smartphone, then the customer using the smartphone instructs his/her bank to transfer the money to the merchant's bank account. The proposed payment model utilizes partially blind signature scheme to hide the customers' identity from the bank. Therefore the proposed payment model provides the customer with complete control on his/her payments and privacy protection from both the bank and the merchant.

It is anticipated that RFID technology would also play a major role in an Smart Home environment. With the availability of RFID-reader-tag-enabled devices and appliances, consumers can make use of the RFID tags attached to their purchased items in their homes. For example, a display screen on a RFID Reader-enabled refrigerator can list out the details of all the RFID tagged items inside the refrigerator, such as item name, ingredi-

ents, manufacturing date, expiry date, *etc.* This example is just one of the many RFID applications that would very soon become common in a Smart Home environment. However, to securely deploy such RFID applications in a smart home environment is not as straight forward as it seems to be. Therefore this thesis describes some of the RFID applications that are applicable to smart home environment. It then identifies their related privacy and security threats and security requirements and also proposes a secure approach, where RFID-tagged consumer items, mRFID-enabled mobile/smart phone, RFID-reader enabled appliances (e.g., refrigerator), and home server would securely interact among themselves. At the moment this approach is just a conceptual idea, but it sheds light on very important security issues related to RFID applications that are beneficial for smart home environment.

Contents

Abstract i
Contents v
List of Tables viii
List of Figures ix

1 Introduction 1
1.1 RFID for Supply Chain Management 1
1.1.1 RFID Standards for Supply Chain Management 2
1.2 RFID for Consumers: Mobile-RFID (mRFID) 4
1.2.1 mRFID/Near Field Communication (NFC) for Mobile Payments . . 5
1.2.2 RFID for Smart Home Environment 6
1.3 Discussion 7

2 Security Assessment of RFID: Supply Chain Management System 8
2.1 Overview of EPCglobal Architecture Framework 8
2.2 RFID: Supply Chain Management Scenario 14
2.3 Security Assessment 18
2.3.1 RFID Tag 18
2.3.2 Tag Interface 20
2.3.3 RFID Reader 22
2.3.4 Reader Interface 23
2.3.5 RFID Middleware 24
2.3.6 EPCIS Capture Interface 25
2.3.7 EPCIS Repository 25
2.3.8 EPCIS Query Interface 27
2.3.9 EPCIS Accessing Application 28
2.3.10 Root/Local ONS 29
2.3.11 Subscriber Authentication System 30
2.4 Summary and Discussion 33

3 RFID: Tag←Reader→Server Security 34

3.1 Security Aspects of Class-1 Generation-2 (ISO 18000:6C) RFID Protocol . . 34

3.2 Security Threats 36

3.2.1 Man-in-the-Middle Attack 36

3.2.2 Cloned Fake Tags 36

3.2.3 Malicious Readers 36

3.2.4 Insider Attack 36

3.2.5 Consumer Information and location privacy violation 37

3.3 Proposed Countermeasures 37

3.4 Proposed Tag-Reader Mutual Authentication (*TRMA*) Protocol 37

3.4.1 Contributions of the Proposed *TRMA* Protocol 38

3.4.2 Supply Chain Processing Scenario and Assumptions 38

3.4.3 Description of the Proposed *TRMA* Protocol 39

3.4.4 Pad Generation Function - PadGen(.): 41

3.4.5 Analysis of Proposed *TRMA* Protocol 45

3.4.6 Drawbacks of *TRMA* Protocol 49

3.5 Proposed *DCSTaR* Protocol 49

3.5.1 Contributions of the Proposed *DCSTaR* Protocol 49

3.5.2 Setup and Assumptions 50

3.5.3 Description of the Proposed *DCSTaR* Protocol 51

3.5.4 Analysis of Proposed *DCSTaR* Protocol 53

3.5.5 Performance Aspects of the Proposed *DCSTaR* Protocol 64

3.5.6 Comparison with Related Work 65

3.6 Discussion 70

4 Efficient and Privacy-Preserving mRFID/NFC Payment Model 71

4.1 Motivation and Related Work 71

4.1.1 Drawbacks of Credit/Debit Card Payments 71

4.1.2 Drawbacks of Contactless Credit/Debit Card & Mobile NFC Payment 73

4.1.3 Drawbacks of Prepaid Contactless Card Payment 74

4.1.4 Drawbacks of Anonymous Electronic Cash Payment 74

4.2 Goals: Offering a Choice - Alternate Payment Model 74

4.3 The Big Picture of the Proposed Mobile Payment Model 75

4.3.1 Benefits/Economic Motives for the Entities 77

4.3.2 Requirements 77

4.4 Technical Details of the Proposed Mobile Payment Model 78

4.4.1 Anonymous Pre-Paid Digital Cash Certificate Issuing Procedure . . . 78
4.4.2 Obtaining Digital Invoice Certificate Procedure 80
4.4.3 The Payment Procedure 81
4.4.4 Reclaiming Unspent Amount Procedure 81
4.5 Analysis 83
4.5.1 Customer Privacy Protection 85
4.5.2 Customer Public Key Collisions 85
4.5.3 Man-in-the-Middle Attack 85
4.5.4 Prevent Replay Attack 86
4.5.5 Prevent Double Spending 86
4.5.6 Non-Repudiation 86
4.5.7 Least Amount of Overhead 86
4.5.8 Against Stolen Smartphone & Money Laundering 87
4.6 Proof-of-Concept Implementation 87
4.7 Summary and Discussion 93

5 mRFID↔Smart Home Security and Privacy 94

5.1 Contributions of this Chapter 94
5.2 Download and View Information Represented by RFID Tags 95
5.2.1 Is that Tag Genuine or Counterfeit?: Tag Authentication 95
5.2.2 Can this EPCIS be trusted?: Secure Job Delegation & Trust Model 96
5.2.3 Security Framework for mRFID Applications in a Public Environment 96
5.3 Carrying Tagged Consumer Items 98
5.3.1 Protecting Consumer Privacy 99
5.4 Interacting with Smart Home Environment 100
5.4.1 Secure Communication between mRFID-Smartphone and Home Server 100
5.4.2 Secure Communication between Home Server and EPCIS 101
5.4.3 Secure Communication: Reader-enabled Appliance & Home Server . 103
5.4.4 Protecting Smart Home Residents Privacy 103
5.5 Summary and Discussion 106

6 Conclusion and Future Work 107

Summary (in Korean) 110

References 111

List of Tables

2.1 Security Assessment of RFID Tag 18
2.2 Security Assessment of Tag Interface 20
2.3 Security Assessment of RFID Reader 22
2.4 Security Assessment of Reader Interface 23
2.5 Security Assessment of RFID Middleware 24
2.6 Security Assessment of EPCIS Capture Interface 25
2.7 Security Assessment of EPCIS Repository 26
2.8 Security Assessment of EPCIS Query Interface 27
2.9 Security Assessment of EPCIS Accessing Application 28
2.10 Security Assessment of Root/Local ONS 29
2.11 Security Assessment of Subscriber Authentication System 30

3.1 Notations. 38
3.2 Differential Cryptanalysis Data 58
3.3 Linear Cryptanalysis Data 59
3.4 NIST (FIPS_140_2) package: testing & certification of RNGs for cryptographic applications 60
3.5 Rabbit, Alphabit and BlockAlphabit Batteries of Tests 61
3.6 Comparing DCSTaR with PRESENT and HIGHT Block Ciphers 68
3.7 Comparison with Related Work 69

4.1 Notations for mRFID/NFC Mobile Payment Model 78
4.2 Execution Time & Data Size 93

List of Figures

1.1 A Typical Passive-RFID Tag 1
1.2 Basic Components of an RFID System 3

2.1 EPCglobal Architecture Framework [11] 10
2.2 Simplified EPCglobal Architecture Framework 11
2.3 RFID: Supply Chain Management Scenario 17
2.4 Security Assessment of RFID: Supply Chain Management System 32

3.1 EPCglobal C1G2 (ISO 18000:6C) Protocol: *Access* Procedure 35
3.2 Proposed TRMA Protocol 40
3.3 Tag's Logical Memory & Access Password Map 44
3.4 Proposed Changes to the Tag's *Reserved* Memory Bank 51
3.5 Example S-box based on the one used in PRESENT block cipher [4] 53
3.6 The 32 bit version of permutation table used in PRESENT block cipher [4] 53
3.7 Proposed *DCSTaR* Protocol 54
3.8 Diffusion & Confusion procedure: $f()$ 55
3.9 C1G2 Access Procedure Air-Interface Commands 66
3.10 DCSTaR Air-Interface Commands 67

4.1 Anonymous Pre-Paid Digital Cash Certificate *crtC* Issuing Procedure 79
4.2 The Payment Procedure 82
4.3 Reclaiming Unspent Amount Procedure: Option 1 83
4.4 Reclaiming (Unspent Money) Phase: Option 2 84
4.5 Requesting and Obtaining Cash Certificate (*crtC*) from Bank 89
4.6 Execution time - Cash Certificate (*crtC*) 90
4.7 Obtaining Invoice Certificate (*crtV*) from Store 91
4.8 Pay and Receive Receipt Procedure 92

5.1 Secure Communication between mRFID-Smartphone and Home Server . . . 101
5.2 Secure Communication between Home Server and EPCIS 102
5.3 RFID Reader-enabled Appliance Configuring Process 104

5.4 Unlocking RFID tags by RFID Reader-enabled Device 105

1. Introduction

Radio Frequency Identification (RFID) [25] is a means to efficiently and quickly, auto-identify objects, assets, pets, people, *etc.* This chapter provides a brief overview of RFID applications that are beneficial to businesses and consumers, and their related standards. It briefly introduces some of the issues pertaining to these applications that this thesis addresses in the subsequent chapters.

1.1 RFID for Supply Chain Management

RFID offers businesses an automated supply chain management system [48]. With the current bar-code technology, each item's bar-code label must be brought before the bar-code scanner, and labels must be scanned one by one. This leads to laborious, human-error prone, and time consuming inventory check, and also causes customers in a store to wait in long queues at the cashier counter.

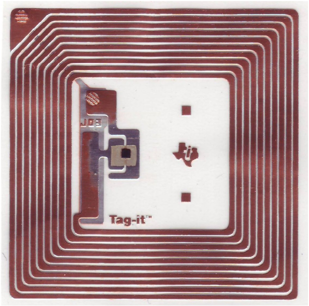


Figure 1.1: A Typical Passive-RFID Tag
(Source: Texas Instruments: Tag-it^{TM})

Whereas with RFID technology, businesses attach Passive-RFID-Tags to their products/items. As shown in Fig.1.1, these tags are low-cost electronic labels that are resource constrained *e.g.,* up to 512 bytes of memory and 3K gates. These tags contain tiny, but durable computer chips with very small antennas. They are powered-up from the interrogation Radio-Frequency (RF) signal from a RFID reader/interrogator. The tag's tiny computer chips contain an Electronic Product Code (EPC) number [11] that uniquely identifies the object to which it is attached to, and the antennas automatically transmit this EPC number without requiring line-of-sight scanning, to RFID readers within the RF range (up to 10m for ultra high frequency (UHF) passive-tags). Therefore RIFD technology allows quick scanning of products in large bulks (*e.g.,* a whole pallet at a time) thus speeding up the supply chain management

Further information associated with the item/EPC number (*e.g.,* item description, manufacturing date, packaging, shipments, item arrival and departure details, *etc.*) is captured and stored on a network of servers and databases, called EPC-Information Services (EPCIS) [11]. The unique EPC number is like an universal resource locator (URL) directing the RFID reader to the right EPCIS on the EPC Network from where the reader can download and upload real-time data about the item it scanned. Therefore, RFID and EPCIS assist geographically distributed supply-chain stakeholders (*e.g.,* manufacturers, distributors, retailers, *etc.*) with instantaneous item identification, and 'real-time' updating, querying, accessing and sharing of item information such as, shipping and receiving, track and trace, theft detection, precise item recall *etc.,* [48]. Thus enabling businesses to realize significant savings. The Fig.1.2 depicts a simplified easy-to-understand RFID system.

This thesis emphasizes that cloned fake RFID tags, malicious RFID readers and consumer privacy violation pose a major threat to RFID-based supply chain management system. Fake tags can be attached to counterfeit products and medicines and malicious readers can corrupt and snoop on genuine tags. These threats can be alleviated by incorporating a RFID tag←reader→EPCIS mutual authentication, communicating-data confidentiality and integrity, secure key-distribution and key-protection, and tag anonymity.

1.1.1 RFID Standards for Supply Chain Management

The standards like the ISO 18000: Part 1, 2, 3, 4, 6 and 7 describe the use of RFID for item management. We also have the EPCglobal Inc. [14], leading the development of industry-driven standards for the EPC to support RFID in supply chain management. This thesis is partly based on the following ratified standards from EPCglobal Inc.

- The EPCglobal Architecture Framework Version 1.3 [11]: This standard defines and describes the EPCglobal Architecture Framework. The EPCglobal Architecture Framework is a collection of interrelated standards for hardware, software, and data interfaces, together with core services that are operated by EPCglobal and its delegates, all in service of a common goal of enhancing the supply chain through the use of Electronic Product Codes (EPCs).

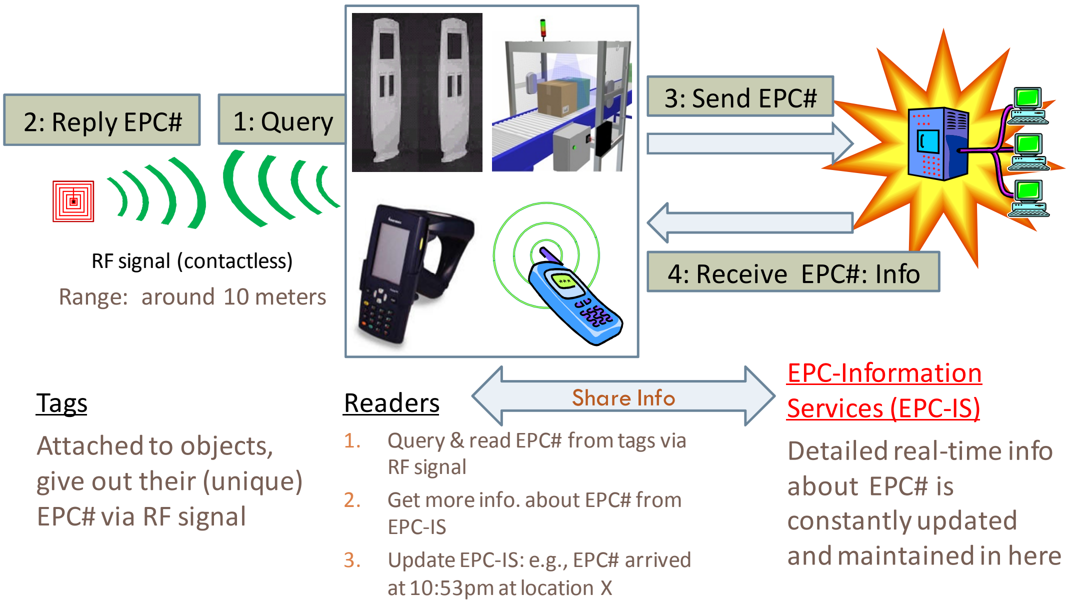

2: Reply EPC#
1: Query
3: Send EPC#
4: Receive EPC#: Info
RF signal (contactless)
Range: around 10 meters
Tags
Attached to objects, give out their (unique) EPC# via RF signal
Readers
Share Info
1. Query & read EPC# from tags via RF signal
2. Get more info. about EPC# from EPC-IS
3. Update EPC-IS: e.g., EPC# arrived at 10:53pm at location X
EPC-Information Services (EPC-IS)
Detailed real-time info about EPC# is constantly updated and maintained in here

Figure 1.2: Basic Components of an RFID System

- The Class-1 Generation-2 (C1G2) UHF (Ultra High Frequency) RFID Protocol for Communications at 860MHz - 960MHz Version 1.2.0 [12]: This standard is for low-cost, passive-backscatter, interrogator-talks-first, RFID system operating in the 860 MHz - 960 MHz frequency range. It specifies the Physical interactions (the signaling layer of the communication link) between readers and tags, and Reader and tag operating procedures and commands. This C1G2 standard has also been ratified as the ISO 18000 Part 6C standard.

- EPCglobal Certificate Profile Specification Version 2.0 [13]: The authentication of entities (subscribers, services, physical devices) operating within the EPCglobal network serves as the foundation of any security function incorporated into the network. It is expected, however, that the X.509 authentication framework will be widely employed within the EPCglobal network. To ensure broad interoperability and rapid deployment while ensuring secure usage, this document defines a profile of X.509 certificate issuance and usage by entities in the EPCglobal network. The profiles defined in this document are based upon two Internet standards, defined in the IETF's PKIX Working Group, that have been well implemented, deployed and tested in many existing environments.

Every RFID tag contains its unique EPC number. EPC is a globally unique serial number that identifies an item in the supply chain. EPC data/number contains: EPC Manager number (identifies the company), Object class (similar to a stock-keeping unit, also called product number), Serial number (specific instance of the object class being tagged, objects own unique identifier). EPCglobal allocates manufacturers specific blocks of EPC numbers, and manufacturers then add their own product codes and serial numbers to their assigned manufacturer numbers to create unique identifiers - EPCs.

1.2 RFID for Consumers: Mobile-RFID (mRFID)

Due to the above-mentioned advantages of RFID technology for supply chain management, very soon it would become economical to tag products at the item level. We would certainly see a large-scale use of tags on consumer goods. This would further lead to development and deployment of electronic appliances and devices that are RFID-reader-enabled, *e.g.,* RFID-reader-enabled book shelves, mobile phones and PDAs. As a result, we can realize one of the visions of ubiquitous computing: an "Internet of Objects", where devices and objects dispersed through our surroundings, can talk to each other, providing real-time information about themselves, locations, and ambient conditions around them. RFID technology will have a tremendous impact on our society, once it starts to assist people in their daily lives. A right step in this direction is Mobile RFID technology.

With Mobile-RFID (mRFID) technology, handheld portable devices like mobile/smart phones and PDAs, apart from having the usual voice/data communicating features, also behave as mRFID-readers and mRFID-tags. As a result, mRFID brings the RFID technology closer to consumers rather than just constraining it's usage to supply chain management. mRFID technology has a great future and it's a very challenging research area. It is poised to be one of the future killer applications and services for mobile communications field. With mRFID technology users can efficiently perform two major tasks, namely: download and view information represented by RFID tags, and machine-to-machine identification and communication.

Download and View Information represented by RFID tags: Just by bringing an mRFID-reader near to a tagged object, we can quickly and easily query that tag's EPC number and by utilizing 3G/4G/Wi-Fi network we can reach the appropriate EPCIS and download information represented by that EPC number and view this information via our mobile device's display screen. For example:

- We can download information about a particular location by scanning tagged sign posts, and landmarks.

- We can download bus routes by scanning tagged Buses.
- We can download prices of tagged merchandize sold at stores, and published in catalogs for Smart Shopping.
- We can verify whether a merchandize is genuine or not by scanning its tag and cross checking it with manufacturer's online genuine product verification service.
- We can download movies, music, trailers, show timings, and theater locations by scanning RFID tagged movie posters, music CDs, *etc.*
- We can download current menu being served at a particular restaurant by scanning its RFID tag, published in a restaurants catalog.

Machine-to-Machine identification and communication: With mRFID-tag we can consider the following applications:

- We can authenticate ourselves to another reader in order to access a particular facility (building, home, *etc.*) or services.
- We can carryout micro payments at subway stations, bus, newspaper stands, and gas stations by bringing our mobile device near to a RFID reader.
- We can give out information about our mobile device's model no. and size of it's display screen, in-order to download and view suitable multimedia content from a multimedia kiosk.

1.2.1 mRFID/Near Field Communication (NFC) for Mobile Payments

Near Field Communication (NFC) [30] is a short-range high-frequency wireless connectivity standard (ISO/IEC 18092), which enables the exchange of data between devices when they are touched or waved within four centimeters of each other. NFC is a combination of the already existing proximity-card standard (ISO/IEC 14443, contactless RFID card) and a reader into a single chip, operating at 13.56 MHz and transferring data at up to 424 Kbits/second. We also have the NFC Forum [32] to advance the use of NFC by developing specifications, and ensuring interoperability among NFC-enabled consumer electronics, mobile devices, PCs and services.

Mobile payment is a payment method, where a mobile phone is used to pay for merchandize and services. Mobile payment is gaining popularity especially in Asia and Europe. The research firm Gartner Inc. predicts that the number of mobile payment users will reach more than 190 million in 2012 [16]. Currently, efforts are being put to deploy a mobile NFC Payment model that precisely mimics the Contactless (RFID) Card Payment model, where a NFC-enabled mobile phone behaves as a contactless credit/debit card.

However this thesis emphasizes that credit/debit card payment transactions do not protect the privacy of the customer. Once the card is handed over to the merchant for payment processing, customers are "no longer in control" on how their card details and money are handled. This leads to card fraud, identity theft, and customer profiling. Therefore for those customers who value their privacy and security of their payment transactions, this thesis proposes a choice - an alternate mobile payment model called "Pre-Paid Mobile HTTPS-based Payment model". In the proposed payment model the customer obtains the bank account information of the merchant, then the customer instructs his/her bank to transfer the money to the merchant's bank account. The proposed payment model utilizes Near Field Communication (NFC) protocol to obtain the merchant's bank account information into the customer's NFC-enabled smartphone. The proposed payment model also utilizes partially blind signature scheme to hide the customers' identity from the bank. Therefore the proposed payment model provides the customer with complete control on his/her payments and privacy protection from both the bank and the merchant.

1.2.2 RFID for Smart Home Environment

The concept of Smart-Homes is becoming more and more popular. It is anticipated that RFID technology would play a major role in such an environment. With the availability of RFID-reader-tag-enabled devices and appliances, consumers can make use of the RFID tags attached to their purchased items in their homes. For example, a display screen on a RFID Reader-enabled refrigerator can list out the details of all the RFID tagged items inside the refrigerator, such as item name, ingredients, manufacturing date, expiry date, *etc.* This example is just one of the many RFID applications that would very soon become common in a Smart Home environment.

In order to make life easier in many ways, and more entertaining, a smart home environment offers a ubiquitous home network system, where different information gadgets, home appliances and other Internet-based applications communicate with each other. Smart homes exchange information and commands among these networked devices via wired and wireless communications. A Home Server or a Home Gateway operating inside this environment is considered to be the brain of this home network system. A home server supports all networking needs in the home, e.g., interacting with the home telephone, stereo system, air-conditioning system, kitchen appliances, lights, blinds, and other network-enabled devices. It also connects the home's local area network to the Internet, which allows the home network to communicate with the external world for sending messages and communicating with the residents of the home. This communication makes it possible to program the smart home from inside or outside the house.

However, to securely deploy such RFID applications in a smart home environment is not as straight forward as it seems to be. Therefore this thesis describes some of the RFID applications that are applicable to smart home environment. It then identifies their related privacy and security threats and security requirements and also propose a secure approach, where RFID-tagged consumer items, RFID-reader enabled appliances (e.g., refrigerator), and home server would securely interact among themselves. At the moment this approach is just a conceptual idea, but it sheds light on very important security issues related to RFID applications that are beneficial for smart home environment.

1.3 Discussion

In this chapter we have introduced few specific RFID applications such as RFID-based Supply Chain Management System for businesses, and mRFID/NFC Mobile Payment and RFID Smart Home Environment for consumers. However, we certainly agree that there are numerous other RFID applications. Therefore, we hope that the ideas and cryptographic protocols proposed in this thesis could be applied, modified, or improved to suit the security and privacy requirements of those RFID applications that have not been addresses in this thesis.

2. Security Assessment of RFID: Supply Chain Management System

This chapter focuses on RFID-based supply chain management system that adheres to the EPCglobal Architecture Framework specification [11] [section 1.1.1]. We approach this framework with a security point-of-view. At the outset this chapter briefly describes the EPCglobal Architecture Framework and provides an example supply chain scenario. The EPCglobal Architecture Framework is composed of entities like RFID Tag, RFID Reader, RFID Middleware, Electronic Product Code Information Service (EPCIS) Repository, EPCIS Accessing Application, Object Naming Service, and Subscriber Authentication. We analyze the various security threats that affect each of these entities and their communication interfaces. We arrive at some possible security requirements and needed security solutions to ward off these threats. Some of these threats include cloned fake RFID tags, unauthorized access and/or modification of RFID tag information and its electronic pedigree (EPCIS data), and eavesdropping, spoofing and Denial of Service (DoS) attack on EPCglobal Subscriber's network.

After our thorough security assessment of the RFID-based supply chain management system, we narrowed down our research focus to RFID Tag←Reader→Server/EPCIS security. Cloned fake RFID tags, malicious RFID readers, man-in-the-middle attacks between genuine tags and readers, and consumer privacy violation are major threats to RFID-based supply chain management system. Therefore we dedicated an entire chapter 1 in this thesis titled "RFID: Tag←Reader→Server/EPCIS security" to discuss these threats in greater detail and propose light-weight cryptographic protocols to achieve RFID tag←reader→server/EPCIS mutual authentication, their communicating-data confidentiality and integrity, secure key-distribution and key-protection, and tag anonymity.

2.1 Overview of EPCglobal Architecture Framework

Throughout this chapter we consider "EPCglobal Architecture Framework [11]" to be the typical RFID-based supply chain management system, which most of the EPCglobal Subscribers would deploy in their organization. An end-user EPCglobal Subscriber is any organization that employs EPCglobal Standards, Interfaces and EPCglobal Core Services as a part of its supply chain management system. Fig.2.1 is the EPCglobal Architecture Framework depicted in the standard document.

In Fig.2.1, the plain green bars denote interfaces governed by EPCglobal standards, while the blue "shadowed" boxes denote roles played by hardware and software components of typical RFID-based supply chain management system architecture. This figure is self descriptive following the data flow (supply chain processing) from the bottom of the figure to the top. We approach EPCglobal Architecture Framework with a security point-of-view and to maintain clarity and simplicity, we group the H/W and S/W roles proposed in the framework into six main entities, namely:

- RFID Tag
- RFID Reader
- RFID Middleware (includes EPCIS Capturing Application)
- Electronic Product Code Information Service (EPCIS) Repository
- EPCIS Accessing Application
- Local Object Naming Service (ONS)

Similarly we consider only those EPCglobal Core Services that have some significance with respect to our security assessment:

- ONS Root
- Subscriber Authentication

To clearly understand the various RFID-based supply chain management system, we also consider four supply chain stakeholders or EPCglobal Subscribers:

- Manufacturer (EPCglobal Subscriber)
- Distributor (EPCglobal Subscriber)
- Retailer (EPCglobal Subscriber)
- Consumer (Not a EPCglobal Subscriber)

Fig.2.2 depicts our simplified version of EPCglobal Architecture Framework and a feel of how this RFID-based supply chain management system is deployed among the stakeholders.

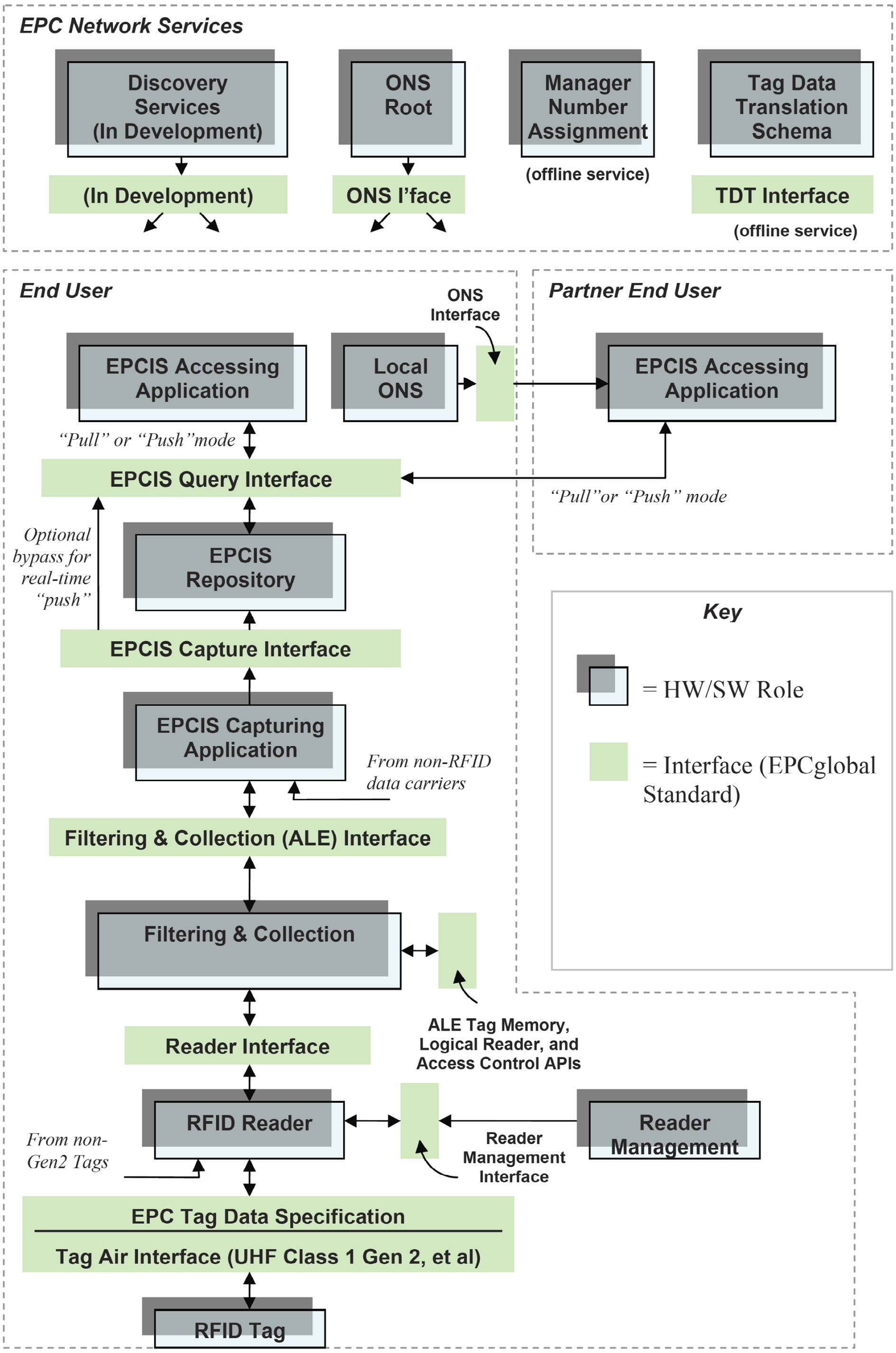

EPC Network Services
Discovery Services (In Development)
ONS Root
Manager Number Assignment
Tag Data Translation Schema
(In Development)
ONS I'face
(offline service)
TDT Interface
(offline service)
End User
ONS Interface
EPCIS Accessing Application
Local ONS
Partner End User
EPCIS Accessing Application
"Pull" or "Push" mode
EPCIS Query Interface
"Pull" or "Push" mode
Optional bypass for real-time "push"
EPCIS Repository
EPCIS Capture Interface
EPCIS Capturing Application
From non-RFID data carriers
Filtering & Collection (ALE) Interface
Filtering & Collection
ALE Tag Memory, Logical Reader, and Access Control APIs
Reader Interface
RFID Reader
Reader Management
Reader Management Interface
From non-Gen2 Tags
EPC Tag Data Specification
Tag Air Interface (UHF Class 1 Gen 2, et al)
RFID Tag
Key
= HW/SW Role
= Interface (EPCglobal Standard)

Figure 2.1: EPCglobal Architecture Framework [11]

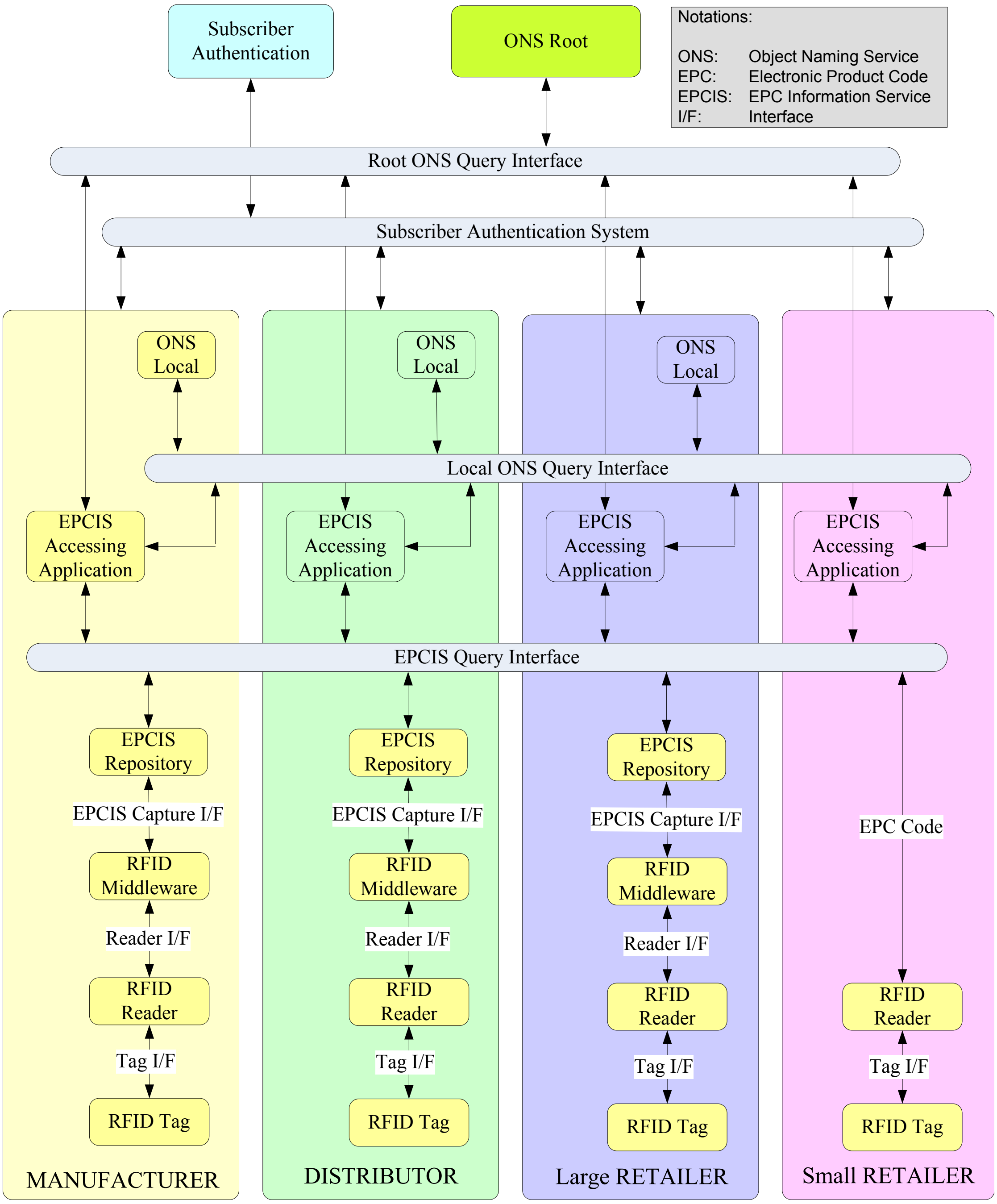


Figure 2.2: Simplified EPCglobal Architecture Framework

In our simplified version of EPCglobal Architecture Framework [Fig.2.2], it can be noticed that most of the EPCglobal Subscribers have the same RFID-based supply chain management system based on the EPCglobal Architecture Framework. Except for a small retailer (e.g., a convenient store carrying out item level sale of various products) who cannot afford to have a large-scale system setup but still has the responsibility to inform the manufacturer the arrival of a particular product in the shop to maintain the integrity of the product's pedigree. He can have only a small network connected terminal (EPCIS Accessing Application) which allows him to connect to the ONS, and EPCIS Repository. We must also consider the consumer who can scan a tag attached to a product using his mobile phone and be connected to the EPCIS Repository to verify if the product is genuine or counterfeit. The Specification of EPCglobal Architecture Framework [11] provides a detailed description of the entities and their interfaces depicted in Figure 2, we summarize some of these details below:

RFID Tag

Every RFID tag contains its unique Electronic Product Code (EPC) number. EPC is a globally unique serial number that identifies an item in the supply chain. EPC number contains: EPC Manager number (identifies the company), Object class (product number), Serial number (specific instance of the object class being tagged, objects own unique identifier). EPCglobal allocates manufacturers specific blocks of EPC numbers, and manufacturers then add their own product codes and serial numbers to their assigned manufacturer numbers to create unique identifiers - EPCs.

RFID Reader

Make multiple observations of RFID tags while they are in the read zone.

Reader Interface

Defines the control and delivery of raw tag reads from Readers to the Filtering & Collection role. Events at this interface say "Reader A saw EPC X at time T."

RFID Middleware

As mentioned before, for simplicity of our security assessment we consider EPCIS Capturing Application as an integral part of RFID Middleware. RFID Middleware filters and collects raw tag reads, over time intervals delimited by events defined by the EPCIS Capturing Application (e.g. tripping a motion detector). The filtered and collected tag read data from RFID Middleware to the EPCIS Capturing Application role may say "At Location L, between time T1 and T2, the following EPCs were observed," where the list of EPCs has no duplicates and has been filtered by criteria defined by the EPCIS Capturing Application.

EPCIS Capturing Application supervises the operation of the lower EPC elements, and provides business context by coordinating with other sources of information involved in executing a particular step of a business process. The EPCIS Capturing Application may, for example, coordinate a conveyor system with Filtering & Collection events, may check for exceptional conditions and take corrective action (e.g., diverting a bad case into a rework area), may present information to a human operator, and so on. The EPCIS Capturing Application understands the business process step or steps during which EPCIS data capture takes place. Here, the Filtering & Collection-level event and the EPCIS-level event may be so similar that no actual processing at the EPCIS Capturing Application level is necessary, and the EPCIS Capturing Application merely configures and routes filtered and collected tag read data directly to an EPCIS Repository.

EPCIS Capture Interface

The interface through which EPCIS data is delivered to enterprise-level roles, including EPCIS Repositories, EPCIS Accessing Applications, and data exchange with partners. Events at this interface say, for example, "At location X, at time T, the following contained objects (cases) were verified as being aggregated to the following containing object (pallet)."

EPCIS Accessing Application

Responsible for carrying out overall enterprise business processes, such as warehouse management, shipping and receiving, historical throughput analysis, and so forth, aided by EPC-related data. The EPCIS Accessing Application may use the Object Name Service (ONS) to locate the EPCIS service (EPCIS Accessing Application) of the EPCglobal Subscriber who is the EPC Manager of the object in question.

EPCIS Query Interface

Provides means whereby an EPCIS Accessing Application can request EPCIS data from an EPCIS Repository or an EPCIS Capturing Application, and the means by which the result is returned. Provides a means for mutual authentication of the two parties. Reflects the result of authorization decisions taken by the providing party, which may include denying a request made by the requesting party, or limiting the scope of data that is delivered in response.

EPCIS Repository

Records EPCIS-level events generated by one or more EPCIS Capturing Applications, and make them available for later query by EPCIS Accessing Applications.

ONS Query Interface

Provides a means for looking up a reference to an EPCIS service (EPCIS Accessing Application) or other service that is provided by the EPC Manager of a specific EPC.

Local ONS

Fulfills ONS lookup requests for EPCs within the control of the enterprise that operates the Local ONS; that is, EPCs for which the enterprise is the EPC Manager.

Root ONS

Provides the initial point of contact for ONS lookups. In most cases, delegates the remainder of the lookup operation to a Local ONS operated by the EPC Manager for the requested EPC. It may completely fulfill ONS requests in cases where there is no local ONS to which to delegate a lookup operation. Provides a lookup service for 64-bit Manager Index values as required by the EPC Tag Data Specification.

Subscriber Authentication

Authenticates the identity of an EPCglobal Subscriber. Provides credentials that one EPCglobal Subscriber may use to authenticate itself to another EPCglobal Subscriber, without prior arrangement between the two Subscribers. Authenticates participation in network services through validation of active EPCglobal Subscription.

2.2 RFID: Supply Chain Management Scenario

In this section we give an example scenario of a RFID-based supply chain process between a manufacturer and its distributor. In the proceeding sections we use the following scenario to describe various security threats of a RFID-based supply chain management. Fig.2.3 depicts the flow of the above-mentioned supply chain scenario.

Manufacturer's End

Step 1
- Embedding unique EPC numbers within the RFID tags.
- Attaching RFID tags involves these stages: Attaching RFID tags on each individual product items. Putting fixed number of these items into cases and attach a RFID tag on each of these cases. Putting fixed number of these cases into cartons and attach a RFID tag on each of these cartons. Putting fixed number of cartons onto a pallet and attach a RFID tag on each of these pallets. Loading fixed number of pallets onto a container and attach a RFID tag on each of these containers.
- At the end of each above-mentioned stage, RFID tags are scanned with RFID readers.

Step 2
- RFID readers scan the RFID tags and send their respective EPCs to the RFID Middleware.
- RFID Middleware associates each EPC number with some related information (e.g., one can of cola, manufacture date, expiry date, etc.).

Step 3
- The EPC number and its associated information are stored in the EPCIS repository. As a result each RFID tag's EPC number in the EPCIS repository represents some details about the items, cases, cartons, and containers to which they are attached to.
- The container is sent to the distributor.

Distributor's End

Step 4 The container with a tag arrives at the warehouse of the distributor. RFID reader is used to scan the tag attached to the container.

Step 5
- The EPC number of the container (EPC-C) is sent to the RFID Middleware.
- RFID Middleware associates this EPC-C with some related information (e.g., EPC-C, arrival location, arrival date, arrival time, etc.).

Step 6 The EPC-C and its associated information are stored in the EPCIS repository of the distributor.

Step 7 & 8 Let us assume that the distributor is only aware of the EPC-C, but has no clue on what products it contains. EPCIS Accessing Application of the Distributor (EAA-D) is used to find out further details of EPC-C.

Step 9 & 10

- Therefore, the EAA-D sends the EPC-C to the Root ONS.
- The Root ONS analyzes the EPC Manager Number, which is a part of the EPC-C, and returns the URL of the Local ONS run by the manufacturer (EPC number Manager of EPC-C).

Step 11 & 12 EAA-D sends the EPC-C to the Local ONS of the manufacturer. The Local ONS directs EAA-D to the EPCIS Accessing Application of the Manufacturer (EAA-M).

Step 13, 14, 15, & 16

- EAA-D sends EPC-C to EAA-M. The EAA-M uses the EPC-C to query the EPCIS repository of the manufacturer, and returns relevant information associated (e.g., EPC-C, cola cans, 50 pallets, 20 cartons per pallet, 15 cases per carton, 6 items per case, etc.) with EPC-C to EAA-D.
- EAA-D also informs EAA-M to update its EPCIS repository with the arrival information (e.g., EPC-C, arrival location, arrival date, arrival time, etc.) of EPC-C at the distributor's warehouse. As a result the manufacturer is now aware of the fact that his container has safely reached the intended distributor.

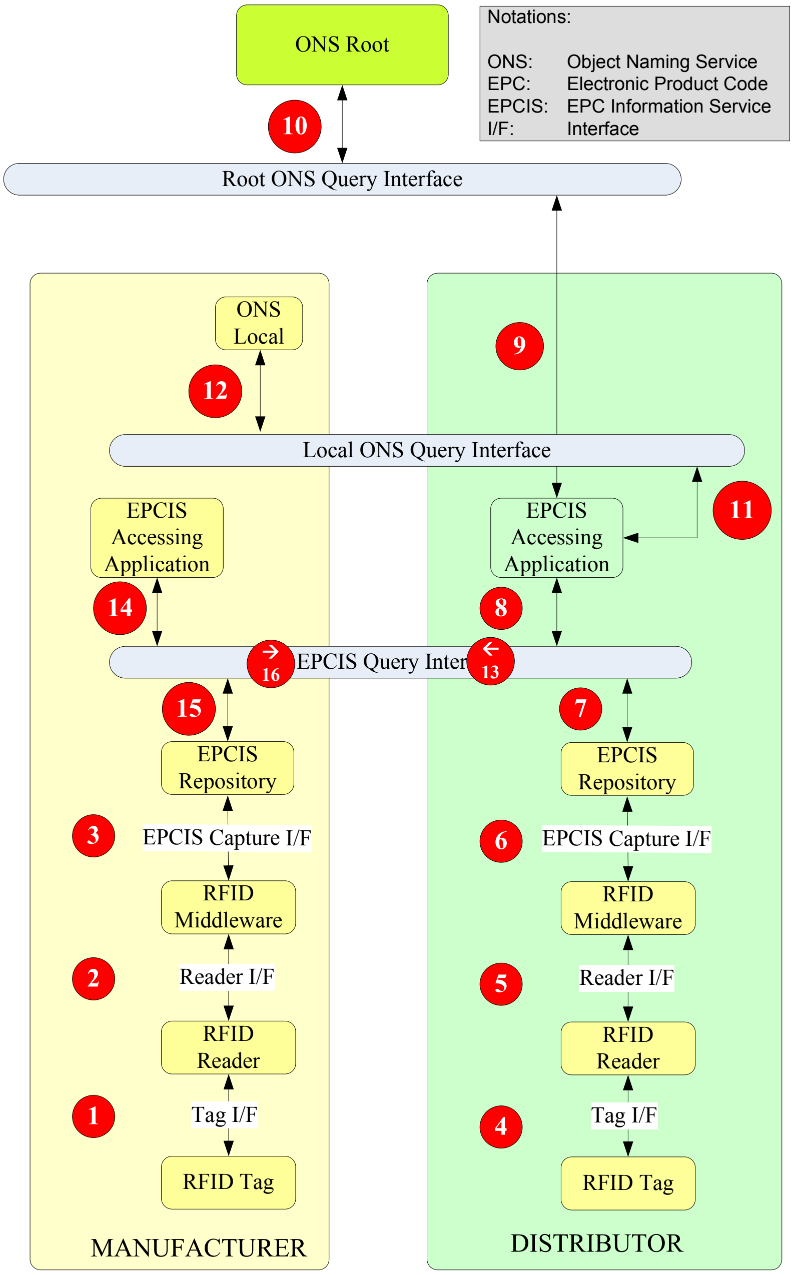


Figure 2.3: RFID: Supply Chain Management Scenario

Table 2.1: Security Assessment of RFID Tag

Security Threat	Security Requirement	Security Solution
Tag Snatching	Tamperproof Tag, Tamperproof Packaging	Physical Security
Unauthorized Tag Data Access & Manipulation	Reader to Tag Authentication	Key-based Challenge-Response Light-Weight Crypto Protocol. Give out ONLY EPC number
Cloned Tags	Tag-Reader Mutual Authentication	

2.3 Security Assessment

In this section we provide security assessment on the entities described in our simplified version of EPCglobal Architecture Framework (Fig.2.2). We analyze the security threats at each entity and suggest needed security details to overcome these threats. The following tables provide a summary of this security assessment. The Fig.2.4 depicts the entire Security Assessment of EPCglobal Architecture Framework, including the security threats, and security requirements and solutions.

2.3.1 RFID Tag

Security Threats [Table.2.1]

Tag Snatching RFID tags attached to a genuine product can be removed and pasted on a fake product, which can then be introduced into a supply chain. A shoplifter can remove a tag attached to a product, thus making it unreadable and walk away with the stolen product undetected.

Unauthorized Tag Data Access As a part of corporate espionage, tags can be illegally accessed beyond the perimeter of a particular warehouse by using powerful RFID readers.

Tag Cloning RFID tag gives out its data (EPC, TID, user data) to any interrogating RFID reader. This reader can either be genuine or malicious. If the tag gives out its data to a malicious RFID reader, then it would be very easy to create a fake tag that gives out the same information.

Tag Data Manipulation Malicious RFID reader can either corrupt or manipulate the data contained in a tag. Such a reader can write into the memory banks of a tag to suit the adversary's requirements.

Security Requirements & Solutions [Table.2.1]

Tamperproof Tag (Physical Security) The manufacture of the tags must make sure that the act of snatching a tag should cause a considerable damage to the tag itself and in the process; the tag must be rendered permanently unusable.

Tamperproof Packaging (Physical Security) The manufacturer of the products must make sure that the tag is attached to the product in such a way that the act of snatching the tag causes a significant damage to the product package itself, which can be vividly noticeable. Setting up CCTVs in the shopping mall would provide additional security in this regard.

Shielded Enclosure (Physical Security) Processing all the pallets, cases, and items in a shielded enclosure would prevent unauthorized tag data access and interferences from outside the enclosure. The enclosure can be shielded by installing RFID reader signal jamming equipment on the outside.

Give out Only EPC It must be made sure that the tag gives out ONLY its EPC code to any interrogating RFID reader. A malicious RFID reader would not have the authorization to access the EPCIS in order to gain any sensible information about the EPC code.

Tag Access Password/Key UHF C1G2 tag has the facility to lock its memory banks with a 32-bit Access Password. This access password is read/write blocked. Only when a RFID reader presents the right access password, will it be able to access the tag's memory banks (e.g., tag's user data). The manufacturer of the products must make sure that all the tags are coded with unique access passwords and that all the memory banks are locked (Kill and Access Password memory banks are permanently locked – cannot be read or modified by any reader). However this security provided in the C1G2 standard is very weak and the access password can be easily exposed, therefore in chapter 3 we discuss this threat in greater detail and proposed lightweight cryptographic authentication protocols. This Reader to Tag authentication would prevent tag data manipulation threat.

Table 2.2: Security Assessment of Tag Interface

Security Threat	Threat To	Security Requirement	Security Solution
Eavesdropping, Replay Attack, Spoofing, MIM Attack	Tag's Access Password & Data, Tag-Reader Impersonation	Cover-coding Tag's Access password & User Data, Tag-Reader Mutual Authentication	Key-based Challenge-Response Light-Weight Crypto Protocol for mutual authentication
DoS Attack, RF Jamming	RFID System	External noise / radio shielded enclosure	

Mutual Authentication The above-mentioned Reader to Tag authentication is useful to identify a genuine RFID reader (only this reader has the knowledge of the access password). But in order to differentiate between a genuine and a cloned tag we must also need Tag to Reader authentication. This leads to the concept of mutual authentication between the tag and the reader. This mutual authentication is currently not defined by the EPCglobal standard, therefore in chapter 3 we discuss this threat in greater detail and proposed light-weight cryptographic mutual authentication protocols.

2.3.2 Tag Interface

Security Threats [Table.2.2]

Eavesdropping Malicious RFID reader listening to the communications between the tag and the reader in order to retrieve the tag's sensitive user data or access password.

Denial of Service (DoS) Attack RF jamming, where the RF channel between the genuine RFID reader and the tag is distributed with a random noisy signal generated by a malicious source, thus bringing down the entire RFID system.

Spoofing (Man-in-the-Middle Attack) Malicious RFID reader hijacks the communication session between the genuine RFID reader and the tag and impersonates as one of them in order to retrieve the tag's sensitive user data or access password.

Replay Attack Malicious RFID reader captures a previous successful session between the genuine RFID reader and the tag, only to replay it back at the later stage. This attack is mounted so that even though a particular product is stolen from the supply chain, this attack can still make it look like the product is still present in the chain without raising any alarms. This attack can also be mounted to introduce counterfeit products into the supply chain.

Security Requirements & Solutions [Table.2.2]

Shielded Enclosure (Physical Security) Processing all the pallets, cases, and items in a shielded enclosure would prevent eavesdropping and DoS attack. The enclosure can be shielded by installing RFID reader signal jamming equipment on the outside.

Give out Only EPC It must be made sure that the tag gives out ONLY its EPC code to any interrogating RFID reader. A malicious RFID reader would not have the authorization to access the EPCIS in order to gain any sensible information about the EPC code.

Tag's Access Password/Key Protection Tag's access password should never be sent in the open from genuine RFID reader to the tag. UHF Class 1 Gen 2 tag generates a random 16-bit (RN16_1) number and sends it to the RFID reader. The reader uses (RN16_1) and performs XOR operation with the first half (16-bits) of the 32-bit access password and sends the result to the tag. The tag performs another XOR operation with (RN16_1) on the obtained result in order to verify the first half of the 32-bit password from the reader. This process is repeated again when the tag sends another random 16-bit (RN16_2) number to the reader to verify the second half of the 32-bit access password. This approach is called the cover-coding.

Unfortunately this approach is not all secure because both (RN16_1) and (RN16_2) are sent in the open. Therefore any eavesdropper, malicious reader or man-in-the-middle attack can easily capture (RN16_1) and (RN16_2) and obtain the access password. Therefore we need a better cover-coding approach to hide the access password. Therefore in chapter 3 we discuss this threat in greater detail and proposed lightweight cryptographic protocols that prevent key disclosure.

Mutual Authentication The above-mentioned Reader to Tag authentication is useful to identify a genuine RFID reader (only this reader has the knowledge of the access password). But in order to prevent spoofing and replay attack we must authenticate the tag every time we want to access it. Therefore we must also need Tag to

Table 2.3: Security Assessment of RFID Reader

Security Threat	Security Requirement	Security Solution
Malicious RFID reader in the vicinity	Reader Authentication, & Authorization	Identity certificates – X.509, Digital Signatures, Public Key Authentication
Hacked / Compromised RFID reader	Do not retain data with RFID readers	Plug unused communication ports

Reader authentication. This leads to the concept of mutual authentication between the tag and the reader. This mutual authentication is currently not defined by the EPCglobal standard, therefore in chapter 3 we discuss this threat in greater detail and proposed light-weight cryptographic mutual authentication protocols.

2.3.3 RFID Reader

Security Threats [Table.2.3]

Malicious RFID Reader Adversaries can place malicious RFID readers inside a particular warehouse in order to carryout corporate espionage, illegally access RFID tags's information, and to mount many attacks that have been previously discussed.

Compromised RFID Reader It could be possible that a genuine RFID reader could be compromised by an adversary and this reader is simultaneously leaking information to the adversary.

Security Requirements & Solutions [Table.2.3]

Authenticate and Authorize RFID Readers It must be made sure that every RFID reader in the vicinity must be authenticated, authorized and well accounted for before the supply chain process begins. This would ensure identification of malicious readers. This authentication and authorization can be done by verifying digital certificates, digital signatures, and public key authentication from the RFID readers (X.509 authentication framework). We suggest that "RFID Management" component can take up this task of authenticating and authorizing all the RFID readers in the system.

Constant Supervision of RFID Readers (Physical Security) All RFID readers must be placed very securely without being distracted by malicious signals and provide

Table 2.4: Security Assessment of Reader Interface

Security Threat	Threat To	Security Requirement	Security Solution
Eavesdropping, Replay Attack, Spoofing, Man-in-the-Middle Attack	Tag's Access Password/Key & Data, Reader-Middleware Impersonation	Secure Comm. Tunnel, Reader-Middleware Mutual Authentication	SSL-TLS / EAP–TLS (wired / wireless communication), Identity certificates – X.509, Digital Signatures, Public Key Authentication

appropriate physical security by which only authorized personnel can access or configure them. Install CCTV cameras monitoring these RFID readers.

Avoid Retaining Data with the RFID Reader It must be made sure that the RFID reader does not retain any data retrieved from the tag. We must plug all the unused communication ports and keep monitoring for any data leakage to the outside.

2.3.4 Reader Interface

Security Threats [Table.2.4]

Eavesdropping The communication channel between the "RFID Middleware" and the reader can be eavesdropped in order to extract sensitive RFID data and tag's access password.

Network Threats Spoofing, Man-in-the-Middle and Replay attacks.

Security Requirements & Solutions [Table.2.4]

Secure Network & Mutual Authentication, Authorization Secure communication tunnel between the mutually authenticated RFID Middleware and RFID reader must be established using services like SSL-TLS (wired), EAP-TLS (wireless), X.509 certificates, digital signatures, and public key authentication.

Table 2.5: Security Assessment of RFID Middleware

Security Threat	Security Requirement	Security Solution
Intrusion, Viruses, DoS Attack, Insider Attack	Application Server Security measures	System Authentication, Authorization, Access control, Access Control List (ACL), Ant-Virus S/W, Firewall, Intrusion detection system, Security audit, Activity logs, Data backup, Service packs & Patches
Spurious Data Attacks: Buffer Overflow	Code review, Bounds checking	Use of programming language that offer bounds checking (Java, .NET)
Spurious Data Attacks: Code Injection	Input validation, Input encoding, Output encoding	Accepting RFID data in the exact predefined format

2.3.5 RFID Middleware

Security Threats [Table.2.5]

RFID Middleware can be considered as an application server; therefore the following application server security threats are also applicable to RFID Middleware.

Application Server Threats Intrusion, viruses, DoS attack, etc.

Spurious Data attack Data Injection, Buffer Overflow

Insider Attacks Disgruntled employees, and saboteurs.

Security Requirements & Solutions [Table.2.5]

- System Authentication, Authorization, Access control, Access Control List, Ant-Virus S/W, Firewall, Intrusion detection system, Security audit, Activity logs, Data backup
- Spurious Data attacks can be prevented by accepting RFID data only if it is in the exact predefined format, code review, Use of programming languages that offer bounds checking (Java, .NET), input validation, input encoding, output encoding.
- Physical Access Control: Restricted access control to the premises, CCTV cameras.

Table 2.6: Security Assessment of EPCIS Capture Interface

Security Threat	Threat To	Security Requirement	Security Solution
Eavesdropping, Replay Attack, Spoofing, Man-in-the-Middle Attack	EPCIS Data, Middleware-EPCIS Repository Impersonation	Secure Comm. Tunnel, Reader-Middleware Mutual Authentication	SSL-TLS / EAP–TLS (wired / wireless communication), Identity certificates – X.509, Digital Signatures, Public Key Authentication

2.3.6 EPCIS Capture Interface

The security threats, security requirements and solutions are very similar to those mentioned in the above sub-section titled “Reader Interface”. But in this case we need to protect the communication channel between the RFID Middleware and the EPCIS Repository, where attacks can be mounted to extract sensitive EPCIS data.

Security Threats [Table.2.6]

Eavesdropping The communication channel between the RFID Middleware and the EPCIS Repository can be eavesdropped in order to extract sensitive EPCIS data.

Network Threats Spoofing, Man-in-the-Middle and Replay attacks.

Security Requirements & Solutions [Table.2.6]

Secure Network & Mutual Authentication, Authorization Secure communication tunnel between the mutually authenticated RFID Middleware and EPCIS Repository must be established using services like Secure Sockets Layer – Transport Layer Security: SSL-TLS (wired), Extensible Authentication Protocol: EAP-TLS (wireless), X.509 certificates, digital signatures, and public key authentication.

2.3.7 EPCIS Repository

EPCIS Repository can be considered as a database server, therefore the security threats related to database server are also applicable to EPCIS Repository. These security threats,

Table 2.7: Security Assessment of EPCIS Repository

Security Threat	Security Requirement	Security Solution
Intrusion, Viruses, DoS Attack, Insider Attack	Database Server Security Measures	System Authentication, Authorization, Access control, Role-based Access Control List, Ant-Virus S/W, Firewall, Intrusion detection system, Security audit, Activity logs, Data backup, Service packs & Patches
Spurious Data Attacks: SQL Injection	Validate and Sanitize input data before passing it to SQL Query	Use stored procedures or extended procedures to avoid granting access to tables. Strict access control policy, audit and check the logs.

security requirements and solutions are very similar to those mentioned in the above subsection titled “RFID Middleware”, but with the following additional considerations.

Security Threats [Table.2.7]

Network Threats Intrusion, viruses, eavesdropping, DoS, Man-in-the-Middle and Replay attacks.

Spurious Data attack SQL Injection.

Insider Attacks Disgruntled employees, and saboteurs.

Security Requirements & Solutions [Table.2.7]

- System Authentication, Authorization, Access control, Role-based Access Control List (RBAC), Ant-Virus S/W, Firewall, Intrusion detection system, Security audit, Activity logs, Data backup, Service packs & Patches
- SQL Injection attack can be prevented by checking for buffer overflows, validate and sanitize input data before passing it to SQL Query , disable script execution by any

outside sources, setup appropriate access rights to database, and strict access control policy, audit and check the logs.

- Physical Access Control: Restricted access control to the premises, CCTV cameras.

2.3.8 EPCIS Query Interface

Table 2.8: Security Assessment of EPCIS Query Interface

Security Threat	Threat To	Security Requirement	Security Solution
Eavesdropping, Replay Attack, Spoofing, Man-in-the-Middle Attack	EPCIS Data, EPCIS Accessing Application-EPCIS Repository Impersonation	Secure Communication Tunnel, EPCIS Accessing Application-Middleware Mutual Authentication	SSL-TLS / EAP–TLS (wire / wireless comm..), Identity certificates – X.509, Digital Signatures, Public Key Authentication

The security threats, security requirements and solutions are very similar to those mentioned in the above sub-section titled “Reader Interface”. But in this case we need to protect the communication channel between EPCIS Query Interface and EPCIS Repository, and EPCIS Accessing Application, where attacks can be mounted to extract sensitive EPCIS data. These threats could *e.g.,* allow an adversary to access the EPCIS data being retrieved/sent from/to EPCIS Repository (or EPCIS Accessing Application)and corrupt the EPCIS repository database.

Security Threats [Table.2.8]

Network Threats Threats like eavesdropping, DoS, Man-in-the-Middle and Replay attacks can be mounted on the communication channel between the EPCIS Query Interface and EPCIS Repository. These threats could e.g., allow an adversary to access the EPCIS data being retrieved/sent from/to EPCIS Repository and corrupt the EPCIS repository database.

Security Requirements & Solutions [Table.2.8]

Secure Network & Mutual Authentication Secure communication tunnel between the authenticated EPCIS Accessing Application and EPCIS Repository using services like SSL/TLS, EAP-TLS, and X.509.

Authorization After authorizing an EPCIS Accessing Application, give out only the EPCIS information that is relevant to that EPCIS Accessing Application based on its role in the supply chain e.g., distributor, wholesaler, retailer, etc.

2.3.9 EPCIS Accessing Application

Table 2.9: Security Assessment of EPCIS Accessing Application

Security Threat	Security Requirement	Security Solution
Unauthorized EPCIS Information Access	Decide which EPCIS information should be revealed to the Accessing Application of the requesting EPCglobal Subscriber (Distributor, Retailer)	Role-based Access Control, Identity certificates – X.509, Digital Signatures, Public Key Authentication
Intrusion, Viruses, DoS Attack, Insider Attack	Application Server Security measures	System Authentication, Authorization, Access control, Role-based Access Control List, Ant-Virus S/W, Firewall, Intrusion detection system, Security audit, Activity logs, Data backup, Service packs & Patches
Spurious Data Attacks: Buffer Overflow	Code review, Bounds checking	Use of programming language that offer bounds checking (Java, .NET)
Spurious Data Attacks: Code Injection	Input validation, Input encoding, Output encoding	Accepting RFID data in the exact predefined format

EPCIS Accessing Application can be considered as an application server; therefore the

Table 2.10: Security Assessment of Root/Local ONS

Security Threat	Security Requirement	Security Solution
ONS cache poisoning, File Corruption, Unauthorized Updates, IP Address Spoofing, Server to Server threat, Data Interception, Server to Client threat	Similar to Secure DNS (*e.g.,* DNS Security Extensions - DNSSEC)	Origin authentication of DNS data, Data integrity, Authenticated Denial of Existence, Digital Signatures, Digital Certificates, Data Confidentiality, Firewall, Access Control List

security threats related to application server are also applicable to EPCIS Accessing Application. These security threats, security requirements and solutions are very similar to those mentioned in the above sub-section titled "RFID Middleware", but with the following additional considerations.

Security Threats [Table.2.9]

- Unauthorized Access to EPCIS Data: Some of the EPCIS data must be available to only authorized EPCglobal Subscribers. Therefore it becomes essential for EPCIS Accessing Application to categorize EPCglobal Subscribers based on their credentials (roles and capabilities) and provide only the EPCIS data that is related and relevant to them.

Security Requirements & Solutions [Table.2.9]

- EPCglobal Subscriber Authentication, Authorization, & Access Control: EPCIS Accessing Application verifies the EPCglobal Subscribers' X.509 certificates, digital signatures, and public key authentication (X.509 authentication framework).

2.3.10 Root/Local ONS

Security Threats [Table.2.10]

- The security threats for ONS could be similar in nature to security threats for DNS such as: File Corruption, Unauthorized Updates, ONS cache poisoning, IP address spoofing, Server to Server threat, Data interception, and Server to Client threat.

Table 2.11: Security Assessment of Subscriber Authentication System

Security Threat	Security Requirement	Security Solution
Malicious EPCglobal Subscribers	Providing Credentials & Authentication	SSL-TLS / EAP–TLS (wired/wireless communication), Identity certificates – X.509, Digital Signatures, Public Key Authentication
Intrusion, Viruses, DoS Attack, Insider Attack	Application Server Security measures	System Authentication, Authorization, Access control, Access Control List (ACL), Ant-Virus S/W, Firewall, Intrusion detection system, Security audit, Activity logs, Data backup, Service packs & Patches

Security Requirements & Solutions [Table.2.10]

- Similar to Secure DNS (e.g., DNS Security Extensions - DNSSEC).
- Good system administration: secure backing-up of the files, proper read and write permissions applied. Access Control Lists.
- Origin authentication of DNS data, Data integrity, Authenticated Denial of Existence, Digital Signatures, Digital Certificates, and Data Confidentiality.
- Firewall & Intrusion Detection System.

2.3.11 Subscriber Authentication System

Subscriber Authentication Service can be considered as an application server, therefore the security threats related to application server are also applicable to Subscriber Authentication Service. These security threats and security requirements and solutions are very similar to those mentioned in the above sub-section titled “RFID Middleware”, but with the following additional considerations.

Security Threats [Table.2.11]

- Unauthorized EPCglobal Subscribers: Some of the EPCIS data must be available to only authorized EPCglobal Subscribers. Therefore it becomes essential to authorize, and authenticate EPCglobal Subscribers based on their credentials and provide only the EPCIS data that is related and relevant to them.
- Intrusion, viruses, eavesdropping, DoS, etc.

Security Requirements & Solutions [Table.2.11]

- Subscriber Authentication Service, authenticates the identity of an EPCglobal Subscriber, provides credentials that one EPCglobal Subscriber may use to authenticate itself to another EPCglobal Subscriber, without prior arrangement between the two Subscribers, and authenticates participation in network services through validation of active EPCglobal Subscription.
- System Authentication, Authorization, Access control, Access Control List (ACL), Ant-Virus S/W, Firewall, Intrusion detection system, Security audit, Activity logs, Data backup, Service packs & Patches.

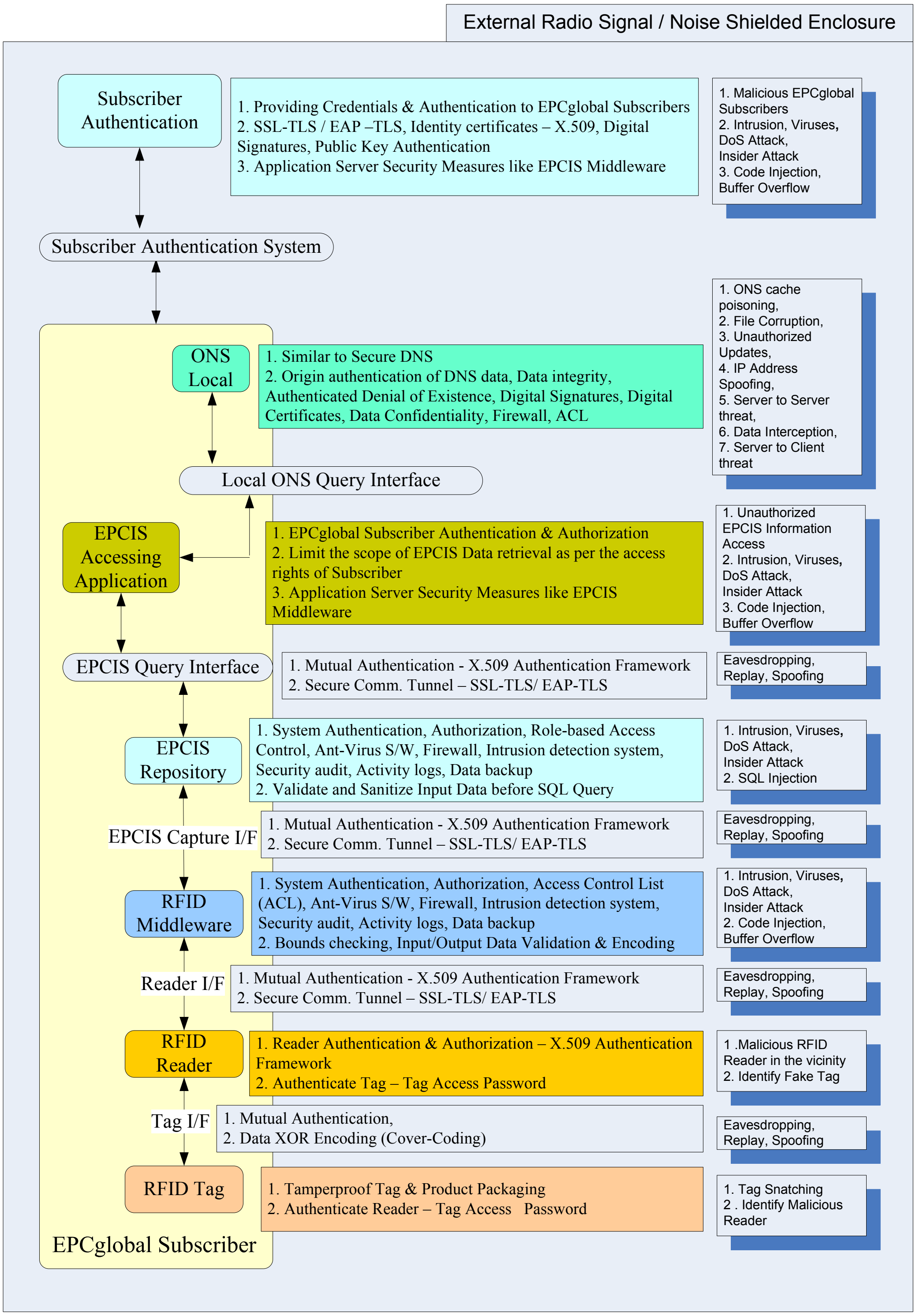

External Radio Signal / Noise Shielded Enclosure
Subscriber Authentication
1. Providing Credentials & Authentication to EPCglobal Subscribers
2. SSL-TLS / EAP –TLS, Identity certificates – X.509, Digital Signatures, Public Key Authentication
3. Application Server Security Measures like EPCIS Middleware
1. Malicious EPCglobal Subscribers
2. Intrusion, Viruses, DoS Attack, Insider Attack
3. Code Injection, Buffer Overflow
Subscriber Authentication System
1. ONS cache poisoning,
2. File Corruption,
3. Unauthorized Updates,
4. IP Address Spoofing,
5. Server to Server threat,
6. Data Interception,
7. Server to Client threat
ONS Local
1. Similar to Secure DNS
2. Origin authentication of DNS data, Data integrity, Authenticated Denial of Existence, Digital Signatures, Digital Certificates, Data Confidentiality, Firewall, ACL
Local ONS Query Interface
EPCIS Accessing Application
1. EPCglobal Subscriber Authentication & Authorization
2. Limit the scope of EPCIS Data retrieval as per the access rights of Subscriber
3. Application Server Security Measures like EPCIS Middleware
1. Unauthorized EPCIS Information Access
2. Intrusion, Viruses, DoS Attack, Insider Attack
3. Code Injection, Buffer Overflow
EPCIS Query Interface
1. Mutual Authentication - X.509 Authentication Framework
2. Secure Comm. Tunnel – SSL-TLS/ EAP-TLS
Eavesdropping, Replay, Spoofing
EPCIS Repository
1. System Authentication, Authorization, Role-based Access Control, Anti-Virus S/W, Firewall, Intrusion detection system, Security audit, Activity logs, Data backup
2. Validate and Sanitize Input Data before SQL Query
1. Intrusion, Viruses, DoS Attack, Insider Attack
2. SQL Injection
EPCIS Capture I/F
1. Mutual Authentication - X.509 Authentication Framework
2. Secure Comm. Tunnel – SSL-TLS/ EAP-TLS
Eavesdropping, Replay, Spoofing
RFID Middleware
1. System Authentication, Authorization, Access Control List (ACL), Ant-Virus S/W, Firewall, Intrusion detection system, Security audit, Activity logs, Data backup
2. Bounds checking, Input/Output Data Validation & Encoding
1. Intrusion, Viruses, DoS Attack, Insider Attack
2. Code Injection, Buffer Overflow
Reader I/F
1. Mutual Authentication - X.509 Authentication Framework
2. Secure Comm. Tunnel – SSL-TLS/ EAP-TLS
Eavesdropping, Replay, Spoofing
RFID Reader
1. Reader Authentication & Authorization – X.509 Authentication Framework
2. Authenticate Tag – Tag Access Password
1 .Malicious RFID Reader in the vicinity
2. Identify Fake Tag
Tag I/F
1. Mutual Authentication,
2. Data XOR Encoding (Cover-Coding)
Eavesdropping, Replay, Spoofing
RFID Tag
1. Tamperproof Tag & Product Packaging
2. Authenticate Reader – Tag Access Password
1. Tag Snatching
2 . Identify Malicious Reader
EPCglobal Subscriber

Figure 2.4: Security Assessment of RFID: Supply Chain Management System

2.4 Summary and Discussion

This chapter carries out a thorough security assessment of the RFID-based supply chain management system that adheres to the EPCglobal Architecture Framework specification. We identified the security threats that affect each of the entities in the framework and proposed some security requirements and needed security solutions. Securing this framework would lead to a secure and safe RFID-based supply chain management system.

The electronic pedigree of the items within the supply chain, and EPCglobal Subscriber's network can be protected by undertaking the following measures: "Subscriber Authentication" a core service of the framework can issue X.509 certificates and public-private security keys to the EPCglobal Subscribers and this helps in mutual authentication, authorization and establishing secure communication channels among the communicating EPCglobal Subscribers. Similarly all the resource rich entities like RFID reader, RFID Middleware, EPCIS Repository, EPCIS Accessing Application, and ONS can authenticate, authorize and establish secure communication channels by using X.509 Authentication Framework and technologies like SSL-TLS and EAP-TLS. We also need to protect application servers (RFID Middleware, EPCIS Accessing Application) and database servers (EPCIS Repository) by installing system authentication and role-based access control, firewall, intrusion detection system, anti-virus software, and input data and SQL query validation. But the threats from cloned RFID tags, malicious snooping RFID readers, and unauthorized tag's data access and manipulation can only be prevented by incorporating a tag-reader mutual authentication scheme, which will be addressed in details in chapter 3.

3. RFID: Tag←Reader→Server Security

This chapter first introduces the security aspects of the EPCglobals's "The Class-1 Generation-2 (C1G2) UHF RFID Protocol for Communications at 860MHz - 960MHz Version 1.2.0 [12]" standard also knows as the ISO 18000:6C standard. It then highlights the various security threats effecting Tag←Reader→Server communication and the required security countermeasures. The chapter also describes some of the related work and proposes two simple, cost-effective, light-weight, and practical cryptographic protocols to alleviate the security threats and provides their security analyses.

3.1 Security Aspects of Class-1 Generation-2 (ISO 18000:6C) RFID Protocol

As per EPCglobal C1G2 UHF RFID Protocol standard [12], a tag's chip has four memory banks: *Reserved*, *EPC*, *TID*, and *User*. The *EPC* memory bank is used to store the EPC number, TID memory bank for tag's unique manufacturer identity number, and User memory bank for additional user data. The manufacturer of the items stores a 32 bit *Access Password* ($APwd$) and a 32 bit *Kill Password* ($KPwd$) into the tags' *Reserved* memory bank. The reserved memory bank is permanently locked by the manufacturer; therefore the *Access* and *Kill* passwords can neither be read nor modified by any reader.

The tag has the capability to verify these two passwords, therefore if a reader sends the right *Access Password*, the tag enters the *Secured State*, where the reader is allowed to carry out mandatory commands such as *Read*, *Write*, and *Lock* on the tag. On the other hand if a reader sends the right *Kill Password*, the tag enters the *Killed State*, where it is permanently disabled. The C1G2 standard does not provide details on how to securely communicate the *Access Password* and *Kill Password* to the readers along the supply chain.

According to the C1G2 standard, tags can generate 16 bit random or pseudo-random numbers R_{Tx} and execute XOR ($\oplus$), and cyclic-redundancy check (CRC) operations. Initially the reader identifies the tag via a *Query* command to obtain its EPC number. Later, the reader and tag implement an *Access* command; which causes the tag to transition from the *Open* to the *Secured State*. Reader and tag can communicate indefinitely in the

Secured State.

The *Access* command is fairly easy to understand by studying the multi-step procedure shown in Fig. 3.1. Just prior to issuing the *Access* command the reader first requests a random number from the tag via the *Req_RN* command. Later, the tag sends two 16 bit random challenges R_{T1} and R_{T2}. The reader responds with $CCPwd_M = APwd_M \oplus R_{T1}$ and $CCPwd_L = APwd_L \oplus R_{T2}$. In here the R_{Tx} is used has an XOR-pad to obscure *APwd*, this is known as Cover-Coding (*CCPwd*) *Access Password.* Each XOR operation shall be performed first on APwd's 16-Most Significant Bits (MSB) $APwd_M$, followed by 16-Least Significant Bits (LSB) $APwd_L$. The tag verifies these responses in order to authenticate the reader. To ensure the validity and integrity of received data both tags and readers shall compute and send a 16 bit Cyclic-Redundancy Check (CRC) value along with their data.

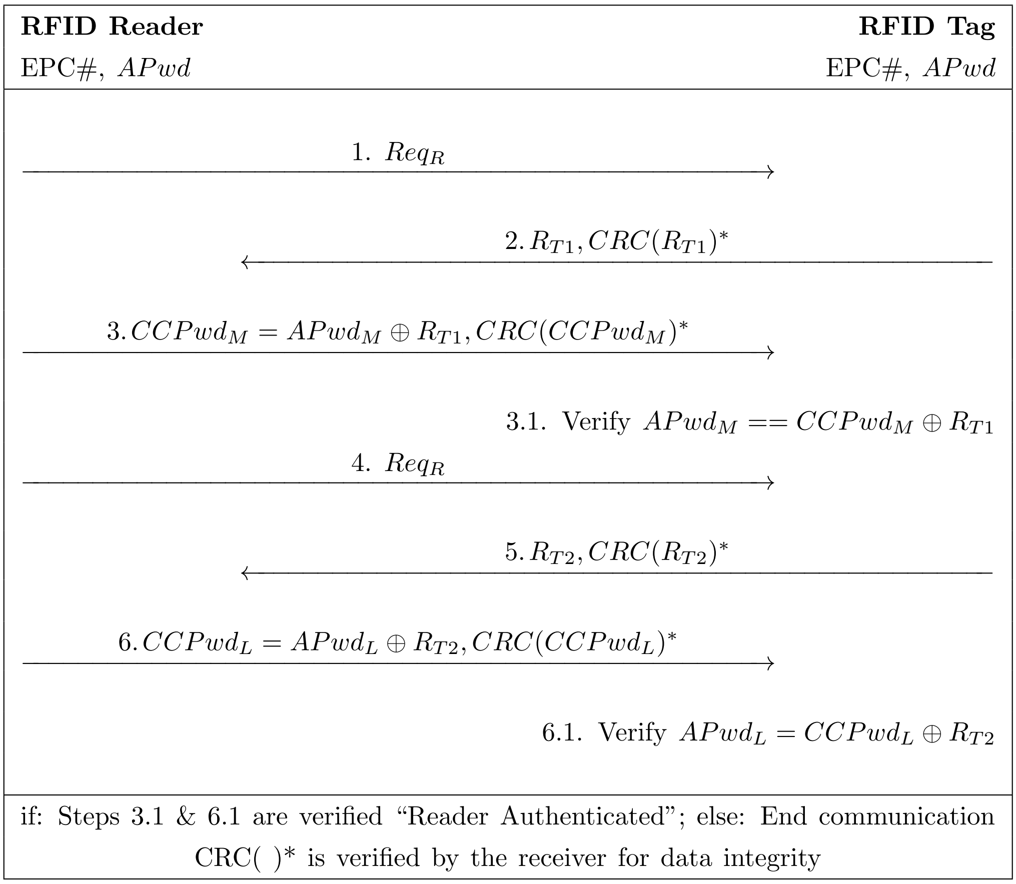

RFID Reader
RFID Tag
EPC#, *APwd*
EPC#, *APwd*
1. Req_R
2. $R_{T1}, CRC(R_{T1})^*$
3. $CCPwd_M = APwd_M \oplus R_{T1}, CRC(CCPwd_M)^*$
3.1. Verify $APwd_M == CCPwd_M \oplus R_{T1}$
4. Req_R
5. $R_{T2}, CRC(R_{T2})^*$
6. $CCPwd_L = APwd_L \oplus R_{T2}, CRC(CCPwd_L)^*$
6.1. Verify $APwd_L = CCPwd_L \oplus R_{T2}$
if: Steps 3.1 & 6.1 are verified "Reader Authenticated"; else: End communication
CRC( )* is verified by the receiver for data integrity

Figure 3.1: EPCglobal C1G2 (ISO 18000:6C) Protocol: *Access* Procedure

3.2 Security Threats

In the *Access* procedure (Fig. 3.1), the tag sends its un-obscured challenges R_{T1} and R_{T2} (steps 3 and 5) in the open. Therefore by eavesdropping on any one of the communication sessions between the tag and the reader any adversary can capture R_{T1} and R_{T2}, and reverse the $\oplus$ operation in the reader's responses - steps 3 and 6 to expose $APwd$. Because of this flaw, even though both the ISO and EPCglobal standards provide (weak) reader authentication and tag memory locking features, they suffer from the following security threats:

3.2.1 Man-in-the-Middle Attack

To accommodate quick and speedy scanning of goods in large bulks, EPCglobal C1G2 UHF RFID tags exhibit outstanding far-field performance. Readers can query and communicate with these tags over a range of 10 meters. Therefore, we can anticipate Man-in-the-Middle attacks from powerful malicious readers. This attack can be mounted to eavesdrop on the communication channel between the tag and the reader, capture the tag's EPC number, impersonate either as a tag or a reader, manipulate their communicating data, and disclose/expose the *Access Password*.

3.2.2 Cloned Fake Tags

The intrinsic functionality of a tag is to respond to any querying reader with its EPC number. Therefore a malicious reader can easily scan and copy the data (EPC number and exposed *Access Password*) on a genuine tag and embed the same data onto a fake tag. This fake tag can be attached to a counterfeit item. Even though a particular tag gives out a genuine EPC number, it must still be authenticated by the reader.

3.2.3 Malicious Readers

An exposed *Access Password* can be utilized by a malicious reader to corrupt the genuine tag. Therefore a tag must also be able to authenticate its reader. Also, only authorized readers must be allowed to access the EPCIS.

3.2.4 Insider Attack

All the hundreds of readers in the supply chain cannot be trusted with *Access* and *Kill* passwords. Any disgruntled employee can compromise authorized readers in a system and can easily obtain the *Access Password.* The *Access Password* for a tag remains the same for the rest of the item's life cycle. Therefore, an exposed *Access Password* at any of the stockholders end, would easily lead to fabrication of cloned fake tags with the same *Access Password.*

3.2.5 Consumer Information and location privacy violation

A consumer carrying a tagged item can be identified, tracked and traced based solely on the tag's unique EPC number.

3.3 Proposed Countermeasures

- Tag ← Reader → Server/EPCIS mutual authentication, alleviates the threats from tag/reader impersonation, malicious readers, and cloned fake tags.
- Communicating-data confidentiality and integrity.
- Secure key-distribution and key-protection.
- Readers must not be provided with any of the keys, but only be permitted to relay obscured data between the tag and the EPCIS/back-end server.
- Anonymity for the tags that are in the possession of a consumer.

3.4 Proposed Tag-Reader Mutual Authentication (*TRMA*) Protocol

The main advantage of this proposed protocol is that it does not require the implementation of any special cryptographic hash functions/keys within the tag. There is also no need for the tag and the reader to synchronize security keys/hash values. We in fact propose to improve the existing one-way reader-to-tag authentication *Access* procedure (proposed by EPCglobal) to also accommodate tag-reader mutual authentication. This particular protocol utilizes tag's already existing, 16-bit random number generator, XOR function, and *Access & Kill Passwords.* It is not a fully secure protocol but it is simple, cost-effective, and light-weight to be implemented on a tag, and also it is practically secure, and highly suitable to the RFID-based supply chain processing scenario. Table 3.1 provides the list of notations we used in this protocol.

Table 3.1: Notations.

Notation	**Description**
Req_R	Command Requesting 16bit Random No.
R_{Tx}	16bit Random No. Generated by Tag
R_{Mx}	16bit Random No. Generated by Manufacturer
APwd	Tag's Access Password
KPwd	Tag's Kill Password
$APwd_M$	16 MSBs of APwd
$APwd_L$	16 LSBs of APwd
$CCPwd_M$	Cover-Coded $APwd_M$
$CCPwd_L$	Cover-Coded $APwd_L$
PAD_x	Generated Pads for Cover-Coding
$\|$	Concatenates its right operand to the end of its left operand
$\oplus$	Bit-wise XOR Operation

3.4.1 Contributions of the Proposed *TRMA* Protocol

In order to alleviate the above mentioned threats, in this paper we propose the following:

- Low-cost passive-tags have tightly constrained computational and memory resources. Therefore we propose a simple, cost-effective, light-weight, and practical tag - reader mutual authentication protocol.
- A better approach to cover-code or obscure tag's *Access Password* (APwd)
- Secure distribution of obscured tags' APwd to stakeholder's RFID readers
- The manufacturer of the product plays a vital role in the tag-reader mutual authentication process. Therefore, the manufacturer can also implicitly keep track on the whereabouts of its products.
- Our protocol adheres to EPCglobal: Architecture Framework specification [11], Class 1 Gen 2 UHF RFID Protocol [12], and Certificate Profile [13]

3.4.2 Supply Chain Processing Scenario and Assumptions

Let us assume that a distributor receives a pallet of products from a manufacturer. The distributor must authenticate the tag attached to the pallet. But the reader at the distributor's end does not know the tag's *APwd.* Therefore the reader contacts the manufacturer in order to get the *APwd.* But, giving away the *APwd* to the distributor would compromise the security of the tag for the rest of its product life cycle and supply chain processing. Therefore in this proposed protocol the *Manufacturer*, Distributor's *Reader*, and the *Tag* follow a multi-step protocol procedure (Shown in Fig.3.2).

As per the EPCglobal Architecture Framework Specification [11], RFID readers are supported, monitored, and managed by many back-end computer terminals and programs such as RFID Middleware, EPCIS Accessing Application, EPCIS Query Interface, EPCIS Repository, and ONS. For reasons of clarity, we will consider the readers at the distributor's end and their back-end computer terminals and programs as one single entity called: "RFID Reader". We assume that the communication channel between the resource rich entities like *RFID Reader*, and *Manufacturer*, to be highly secure (SSL-TLS, EAP-TLS, and X.509 Authentication Framework). The trusted "Subscriber Authentication [11]" core service identifies the roles (distributor, wholesaler, and retailer) of various stakeholders and distributes appropriate X.509 type certificates [13] to them. These certificates authenticate, authorize, and secure the communication channel among them. We assume that the *RFID Reader*, and *Manufacturer* share their digital certificates and be able to execute Signature - $Sig\{.\}$ and Encryption - $Encr\{.\}$ functions.

3.4.3 Description of the Proposed *TRMA* Protocol

Our proposed protocol can be easily understood by looking at Fig.3.2. Steps 1-5 details Reader Authentication Process. Steps 6-9 describe Tag Authentication Process. Please note that Steps 1-9 are carried out in one interrogation session between the tag and the reader. After Step 9, if the verification of the tag is successful, the *Manufacturer* also updates it's EPCIS Repository with the information that a pallet to which this authentic tag is attached to has reached the distributor, and also other information associated with this event.

One of the main components of our proposed protocol is PadGen(.): *Pad Generation Function.* Detailed description of the PadGen(.) function is described in the next subsection. In short, this function takes two 16-bit random numbers each, from the Tag (R_{Tx}) and the Manufacturer (R_{Mx}), and utilizes the *Access* (APwd) and *Kill* (KPwd) *Passwords*, to generate two 16-bit Pads (PAD_x). Since ONLY the tag and the manufacturer know (APwd) and (KPwd), just by sharing the random numbers among themselves (via RFID reader), both the tag and the manufacturer can generate the same pads. Later these two pads are in-turn used to cover-code (XOR) the two 16-bit (APwd) chunks

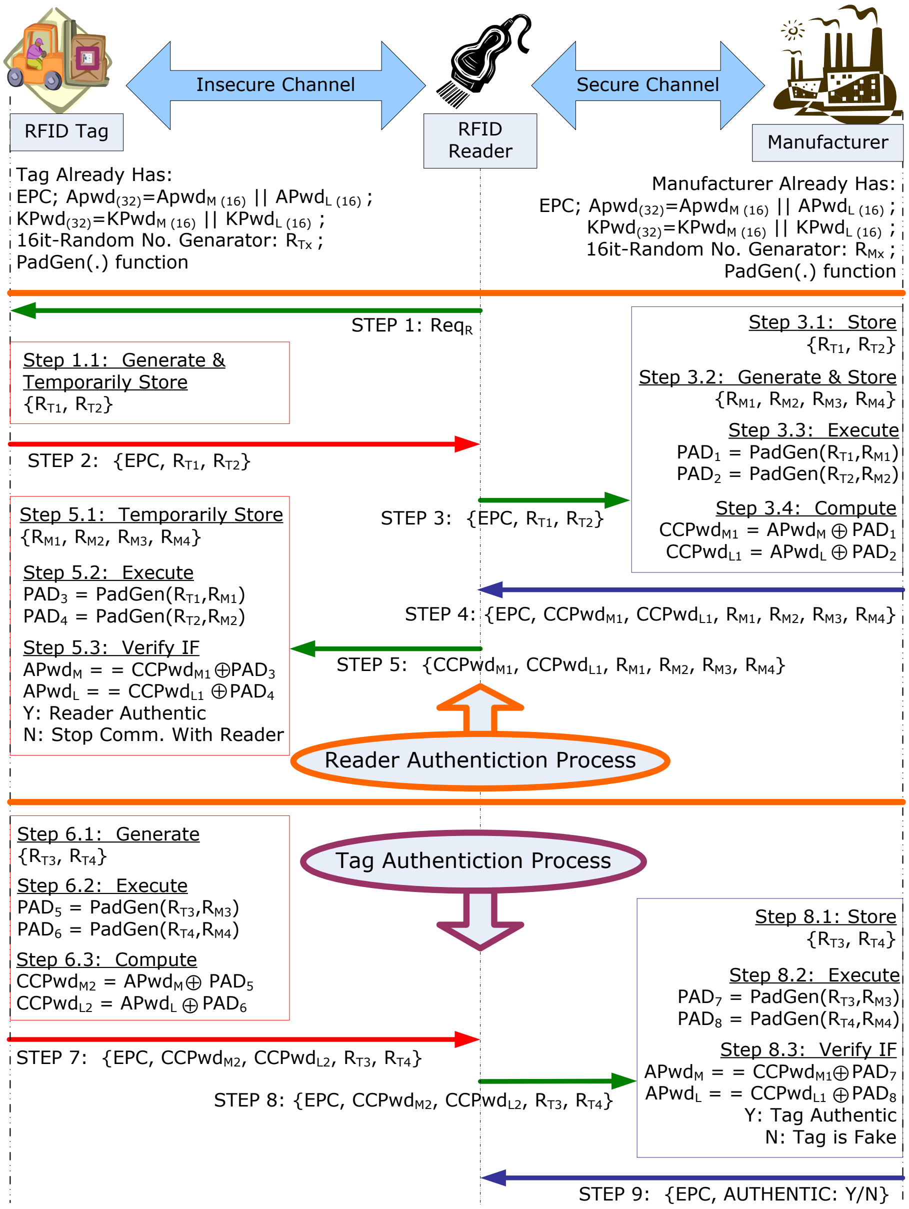

Insecure Channel
Secure Channel
RFID Tag
RFID Reader
Manufacturer
Tag Already Has:
EPC; $\text{Apwd}_{(32)}$=$\text{Apwd}_{M\ (16)}$ || $\text{APwd}_{L\ (16)}$;
$\text{KPwd}_{(32)}$=$\text{KPwd}_{M\ (16)}$ || $\text{KPwd}_{L\ (16)}$;
16it-Random No. Genarator: R_{Tx} ;
PadGen(.) function
Manufacturer Already Has:
EPC; $\text{Apwd}_{(32)}$=$\text{Apwd}_{M\ (16)}$ || $\text{APwd}_{L\ (16)}$;
$\text{KPwd}_{(32)}$=$\text{KPwd}_{M\ (16)}$ || $\text{KPwd}_{L\ (16)}$;
16it-Random No. Genarator: R_{Mx} ;
PadGen(.) function
STEP 1: Req_R
Step 1.1: Generate & Temporarily Store {R_{T1}, R_{T2}}
STEP 2: {EPC, R_{T1}, R_{T2}}
Step 3.1: Store {R_{T1}, R_{T2}}
Step 3.2: Generate & Store {R_{M1}, R_{M2}, R_{M3}, R_{M4}}
Step 3.3: Execute
PAD_1 = PadGen(R_{T1},R_{M1})
PAD_2 = PadGen(R_{T2},R_{M2})
Step 3.4: Compute
CCPwd_{M1} = $\text{APwd}_M \oplus \text{PAD}_1$
CCPwd_{L1} = $\text{APwd}_L \oplus \text{PAD}_2$
STEP 3: {EPC, R_{T1}, R_{T2}}
STEP 4: {EPC, CCPwd_{M1}, CCPwd_{L1}, R_{M1}, R_{M2}, R_{M3}, R_{M4}}
Step 5.1: Temporarily Store {R_{M1}, R_{M2}, R_{M3}, R_{M4}}
Step 5.2: Execute
PAD_3 = PadGen(R_{T1},R_{M1})
PAD_4 = PadGen(R_{T2},R_{M2})
Step 5.3: Verify IF
APwd_M = = $\text{CCPwd}_{M1} \oplus \text{PAD}_3$
APwd_L = = $\text{CCPwd}_{L1} \oplus \text{PAD}_4$
Y: Reader Authentic
N: Stop Comm. With Reader
STEP 5: {CCPwd_{M1}, CCPwd_{L1}, R_{M1}, R_{M2}, R_{M3}, R_{M4}}
Reader Authentiction Process
Tag Authentiction Process
Step 6.1: Generate {R_{T3}, R_{T4}}
Step 6.2: Execute
PAD_5 = PadGen(R_{T3},R_{M3})
PAD_6 = PadGen(R_{T4},R_{M4})
Step 6.3: Compute
CCPwd_{M2} = $\text{APwd}_M \oplus \text{PAD}_5$
CCPwd_{L2} = $\text{APwd}_L \oplus \text{PAD}_6$
STEP 7: {EPC, CCPwd_{M2}, CCPwd_{L2}, R_{T3}, R_{T4}}
STEP 8: {EPC, CCPwd_{M2}, CCPwd_{L2}, R_{T3}, R_{T4}}
Step 8.1: Store {R_{T3}, R_{T4}}
Step 8.2: Execute
PAD_7 = PadGen(R_{T3},R_{M3})
PAD_8 = PadGen(R_{T4},R_{M4})
Step 8.3: Verify IF
APwd_M = = $\text{CCPwd}_{M1} \oplus \text{PAD}_7$
APwd_L = = $\text{CCPwd}_{L1} \oplus \text{PAD}_8$
Y: Tag Authentic
N: Tag is Fake
STEP 9: {EPC, AUTHENTIC: Y/N}

Figure 3.2: Proposed TRMA Protocol

($APwd_M$, $APwd_L$). This approach prevents the major drawback of the one-way reader-to-tag authentication *Access* proposed by EPCglobal, where the random numbers sent in open, un-encrypted form, are used as pads to cover-code the APwd chunks. But in our proposed protocol the generated pads are known only to the tag and the manufacturer, and using them to cover-code the APwd chunks, provides fair amount of obscurity and security to the real APwd. Therefore we can fend off threats like exposed tag's APwd, malicious snooping readers, disgruntled employee, man-in-the-middle attacks, and cloned tags.

The manufacturer and the reader mutually authenticate and authorize each other via their digital certificates and signatures. The manufacturer sends the cover-coded ($APwd$) chunks $CCPwd_{M1}$, $CCPwd_{L1}$ to only authorized and authenticated reader. When the reader presents $CCPwd_{M1}$, $CCPwd_{L1}$ to the tag, the tag verifies them and if tallied the tag authenticates the reader to be genuine. Manufacturer Authenticates Reader, Tag Authenticates Manufacturer, therefore Tag Authenticates Reader. Similarly the tag sends $CCPwd_{M2}$, $CCPwd_{L2}$ to the reader, the reader passes them on to the manufacturer, where they are verified and if tallied, the manufacturer informs the reader that it is handling a genuine tag. Reader Authenticates Manufacturer, Manufacturer Authenticates Tag, therefore Reader Authenticates Tag.

3.4.4 Pad Generation Function - PadGen(.):

Formula:

- $CCPwd_M = APwd_M \oplus PAD$
- $PAD = PadGen(R_{Tx}, R_{Mx})$

  =KPwdBits(APwdBits(R_{Tx}, R_{Mx}), R_{Tx})

Let us represent the 32-bit APwd as:

- Hexadecimal (Base 16) Notation: $A \in \{0, 1, 3, \cdots, 9, A, B, C, \cdots, F\}$

  APwd $= A_0A_4A_8A_{12}A_{16}A_{20}A_{24}A_{28}$

- Binary (Base 2) Notation: $a \in \{0, 1\}$

  APwd $= a_0a_1a_2a_3a_4a_5a_6 \cdots\cdots a_{28}a_{29}a_{30}a_{31}$

  APwd $= APwd_M \| APwd_L$

  $APwd_M = a_0a_1a_2 \cdots\cdots a_{13}a_{14}a_{15}$

  $APwd_L = a_{16}a_{17}a_{18} \cdots\cdots a_{29}a_{30}a_{31}$

Let us represent the 32-bit KPwd as:

- Hexadecimal (Base 16) Notation: $K \in \{0, 1, 2, 3, \cdots, 9, A, B, C, \cdots, F\}$

  KPwd $= K_0K_4K_8K_{12}K_{16}K_{20}K_{24}K_{28}$

- Binary (Base 2) Notation: $k \in \{0, 1\}$

  KPwd $= k_0k_1k_2k_3k_4k_5k_6 \cdots\cdots k_{28}k_{29}k_{30}k_{31}$

  KPwd $= KPwd_M \| KPwd_L$

  $KPwd_M = k_0k_1k_2 \cdots\cdots k_{13}k_{14}k_{15}$

  $KPwd_L = k_{16}k_{17}k_{18} \cdots\cdots k_{29}k_{30}k_{31}$

Let us represent the 16-bit random number R_{Tx} generated by Tag as:

- Hexadecimal (Base 16) Notation: $ht \in \{0, 1, 2, 3, \cdots, 9, A, B, C, \cdots, F\}$

  $R_{Tx} = ht_1ht_2ht_3ht_4$

- Decimal (Base 10) Notation: $dt \in \{0, 1, 2, 3, \cdots, 9, 10, 11, \cdots, 15\}$

  $ht_i = dt_i$

  $R_{Tx} = dt_1dt_2dt_3dt_4$

Let us represent the 16-bit random number R_{Mx} generated by Tag as:

- Hexadecimal (Base 16) Notation: $hm \in \{0, 1, 2, 3, \cdots, 9, A, B, C, \cdots, F\}$

  $R_{Mx} = hm_1hm_2hm_3hm_4$

- Decimal (Base 10) Notation: $dm \in \{0, 1, 2, 3, \cdots, 9, 10, 11, \cdots, 15\}$

  $hm_i = dm_i$

  $R_{Mx} = dm_1dm_2dm_3dm_4$

Let us compute: APwdBits(R_{Tx}, R_{Mx})

- APwdBits(R_{Tx}, R_{Mx})

$$= a_{dt_1}a_{dt_2}a_{dt_3}a_{dt_4} \| a_{dt_{1+16}}a_{dt_{2+16}}a_{dt_{3+16}}a_{dt_{4+16}} \|$$
$$a_{dm_1}a_{dm_2}a_{dm_3}a_{dm_4} \| a_{dm_{1+16}}a_{dm_{2+16}}a_{dm_{3+16}}a_{dm_{4+16}} \qquad \text{[Base 2]}$$

$$= hv_1hv_2hv_3hv_4 \qquad \text{[Base 16, where } hv \in \{0, 1, 2, 3, \cdots, 9, A, B, C, \cdots, F\}\text{]}$$

$$= dv_1dv_2dv_3dv_4 \qquad \text{[Base 10, where } dv \in \{0, 1, 2, 3, \cdots, 9, 10, 11, \cdots, 15\}\text{]}$$

Let us compute: KPwdBits(APwdBits(R_{Tx}, R_{Mx}), R_{Tx})
=KPwdBits($hv_1hv_2hv_3hv_4, R_{Tx}$)

- KPwdBits($hv_1hv_2hv_3hv_4, R_{Tx}$)

$$= k_{dv_1}k_{dv_2}k_{dv_3}k_{dv_4}\|k_{dv_{1+16}}k_{dv_{2+16}}k_{dv_{3+16}}k_{dv_{4+16}}\|$$
$$k_{dt_1}k_{dt_2}k_{dt_3}k_{dt_4}\|t_{dt_{1+16}}k_{dt_{2+16}}k_{dt_{3+16}}k_{dt_{4+16}} \qquad \text{[Base 2]}$$

$$= hp_1hp_2hp_3hp_4 \qquad \text{[Base 16, where } hp \in \{0,1,2,3,\cdots,9,A,B,C,\cdots,F\}]$$

$\therefore \quad PAD = hp_1hp_2hp_3hp_4$ [Base 16]

An Example for PadGen(.)

As per the C1G2 standard, the Access Password is a 32-bit value stored in tag's Reserved memory 20_h to $3F_h$, MSB first. Also the Kill Password is a 32-bit value stored in tag's Reserved memory 00_h to $1F_h$, MSB first. Fig. 3.3 depicts tag's logical memory, Access and Kill password map. The figure includes the bit-wise storage of access password, please note that the bit-wise storage of kill password is also similar.

Initially the manufacturer's server executes $PadGen(R_{T1}, R_{M1})$, where it uses the values of R_{T1} and R_{M1} as location (Locn.) numbers to retrieve the individual Access-Password (APwd) bits stored in those locations, concatenates these bits in series to form a 16-bit temporary value - $Temp$. The server again executes $PadGen(Temp, R_{T1})$. But in here, the server uses the values of $Temp$, and R_{T1} as location (Locn.) numbers to retrieve the individual Kill-Password (KPwd) bits stored in those locations, concatenates these bits in series to form the final 16-bit PAD value: PAD_1

$CCPwd_{M1} = APwd_M \oplus PAD_1$
$PAD_1 = PadGen(R_{T1}, R_{M1})$
=KPwdBits(APwdBits(R_{T1}, R_{M1}), R_{T1})
Let $Temp$ = APwdBits(R_{T1}, R_{M1})
$\therefore PAD_1$ = KPwdBits($Temp, R_{T1}$)

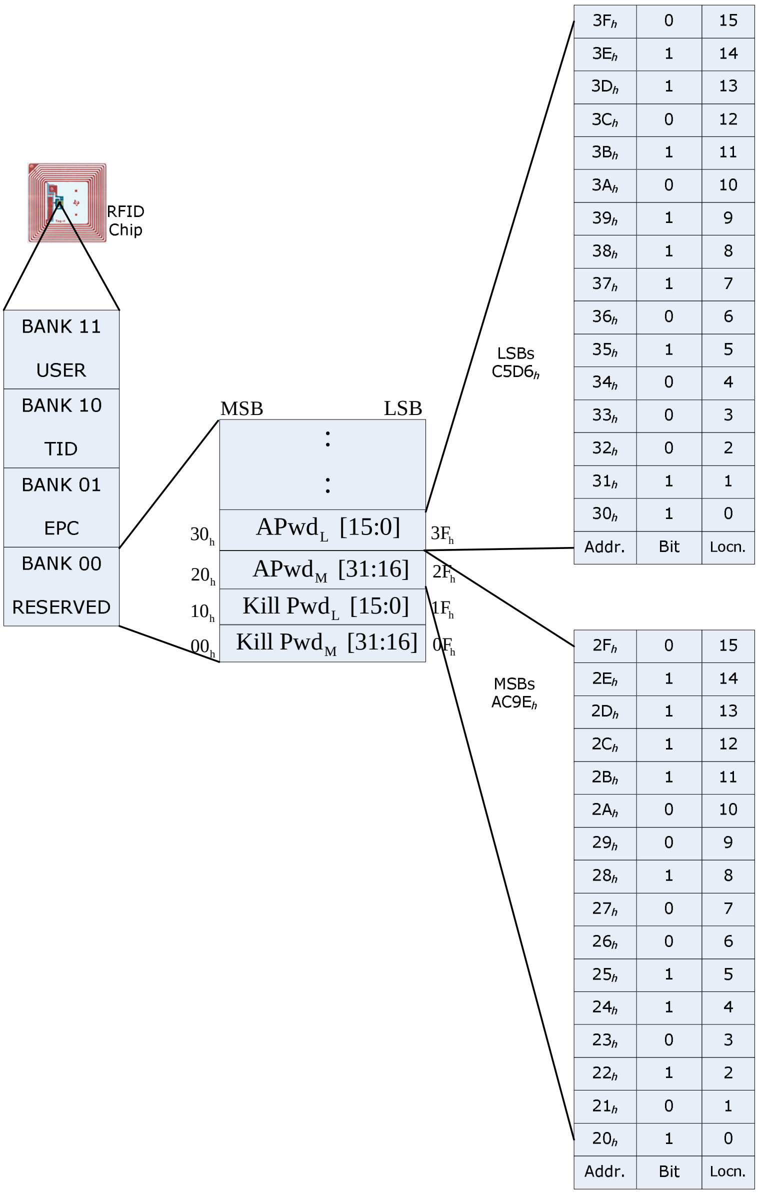
For example, let us assume:

- The Tag's Access Password ($APwd$): $AC9EC5D6_h$

  The 1st half (16 MSBs) of $APwd$ is

  $APwd_M = AC9E_h$= 1010 1100 1001 1110_2

  The 2nd half (16 LSBs) of the $APwd$ is



3F$_h$	0	15
3E$_h$	1	14
3D$_h$	1	13
3C$_h$	0	12
3B$_h$	1	11
3A$_h$	0	10
39$_h$	1	9
38$_h$	1	8
37$_h$	1	7
36$_h$	0	6
35$_h$	1	5
34$_h$	0	4
33$_h$	0	3
32$_h$	0	2
31$_h$	1	1
30$_h$	1	0
Addr.	Bit	Locn.

2F$_h$	0	15
2E$_h$	1	14
2D$_h$	1	13
2C$_h$	1	12
2B$_h$	1	11
2A$_h$	0	10
29$_h$	0	9
28$_h$	1	8
27$_h$	0	7
26$_h$	0	6
25$_h$	1	5
24$_h$	1	4
23$_h$	0	3
22$_h$	1	2
21$_h$	0	1
20$_h$	1	0
Addr.	Bit	Locn.

Figure 3.3: Tag's Logical Memory & Access Password Map

$APwd_L = C5D6_h = 1100\ 0101\ 1101\ 0110_2$

- The Tag's Kill Password ($KPwd$): $DEC59A4F_h$

  The 1st half (16 MSBs) of $KPwd$ is

  $KPwd_M = DEC5_h$= $1101\ 1110\ 1100\ 0101_2$

  The 2nd half (16 LSBs) of the $KPwd$ is

  $KPwd_L = 9A4F_h = 1001\ 1010\ 0100\ 1111_2$

- $R_{T1} = A69D_h$ and $R_{M1} = 2B5F_h$

- APwdBits(R_{T1} , R_{M1}):

  APwdBits($A69D_h$, $2B5F_h$):

  $A69D_h = 10^{th}6^{th}9^{th}13^{th}$ location of Access Password's MSBs (20_h to $2F_h$) $= 0001_2$

  $A69D_h = 10^{th}6^{th}9^{th}13^{th}$ location of Access Password's LSBs (30_h to $3F_h$) $= 0011_2$

  $2B5F_h = 2^{nd}11^{th}5^{th}15^{th}$ location of Access Password's MSBs (20_h to $2F_h$) $= 1110_2$

  $2B5F_h = 2^{nd}11^{th}5^{th}15^{th}$ location of Access Password's LSBs (30_h to $3F_h$) $= 0110_2$

- Combining the above 4 results we have a 16-bit $Temp$ value

  $Temp = 0001\ 0011\ 1110\ 0110_2 = 13E6_h$

- KPwdBits($Temp$, R_{T1}):

  KPwdBits($13E6_h$, $A69D_h$):

  $13E6_h = 1^{st}3^{rd}14^{th}6^{th}$ location of Kill Password's MSBs (00_h to $0F_h$) $= 1101_2$

  $13E6_h = 1^{st}3^{rd}14^{th}6^{th}$ location of Kill Password's LSBs (00_h to $0F_h$) $= 0111_2$

  $A69D_h = 10^{th}6^{th}9^{th}13^{th}$ location of Kill Password's MSBs (10_h to $1F_h$) $= 0111_2$

  $A69D_h = 10^{th}6^{th}9^{th}13^{th}$ location of Kill Password's LSBs (30_h to $3F_h$) $= 0111_2$

- Combining the above 4 results we have a 16-bit PAD_1 value

  $PAD_1 = 1101\ 0111\ 0111\ 0111_2 = D777_h$

3.4.5 Analysis of Proposed *TRMA* Protocol

Security Analysis

Our proposed protocol is not fully secure, it suffers from the fact that the APwd and KPwd are only 32-bits each (as per the EPCglobal standard). A simple brute-force attack or other active-attacks on the tag, can crack these two passwords. This is a trade off between keeping our protocol simple, low-cost, and adhering to EPCglobal standards, instead of proposing an expensive and a completely secure protocol. But for a RFID-based supply chain processing scenario, our proposed protocol proves to be light-weight and practically secure, this aspect is highlighted in the following sections. Our protocol provides tag-reader mutual authentication, and prevents the leakage of tag's APwd by the stakeholder's reader or by a disgruntled/compromised employee.

Practically Secure:

An active attacker may continuously eavesdrop on the communication channel between a particular tag and a reader, in order to extract that tag's APwd and KPwd. Since both the passwords are only 32-bits, the attacker can easily mount a ciphertext-only attack. Such active attacks can be prevented by processing the tagged items in an enclosure (warehouse) that is sealed off from external noise and radio signals from malicious readers. In an extremely fast paced, RFID supply chain processing environment, it is not feasible to continuously eavesdrop on one particular tag-reader communication channel for a time long enough to mount ciphertext-only attack. Several bulks of items pass through the readers with in a very short interval of time.

Secure Against Reader Impersonation Attack: The first phase of our proposed protocol is for the reader to authenticate itself to the tag. But a malicious reader does not posses both the APwd and KPwd, in order to generate corresponding CCPwd. The tag can easily detect a false CCPwd and immediately stop communication with the malicious reader. A malicious reader cannot even access the manufacturer (EPCIS) due to lack of authenticating and authorizing credentials. Therefore a Genuine Reader Impersonation Attack cannot be successful.

Secure Against Cloned Fake Tags and Tag Impersonation Attack: The second phase of our proposed protocol is for the tag to authenticate itself to the manufacturer. But a malicious tag or a cloned fake tag, do not posses both the APwd and KPwd, in order to generate corresponding CCPwd. The manufacturer can easily detect a false CCPwd and notify the reader that the tag in question is not authentic, it could be either a fake tag or a malicious tag.

On the other hand, a fake tag or a device emulating the functionalities of a malicious tag, may use the same random numbers or weak random numbers (*e.g.,* 0000_h, 1111_h, $FFFF_h$, etc.) repeatedly in order to cryptanalyze the CCPwd obtained from the manufacturer (during the reader authentication phase of our protocol). Therefore for additional security, the reader (at the distributor's end) or the manufacturer must detect and terminate the communication, if one particular tag is using the same or weak random numbers for over a certain number of consecutive sessions. Since the manufacturer is a resource rich entity, it can keep track of the random numbers and also enforce the generation of good quality random numbers from the tag. The reader or the manufacturer can easily detect an anomaly, if one particular tag is being interrogated or making its presence felt more than a certain pre-defined number of times. This means that this tag is stationary, and is not moving through the supply chain processing. Chances are that, it can be a device emulating the functionalities of a malicious tag. With the above two security measures a Genuine Tag Impersonation Attack cannot be successful.

Tag's Access Password Never Exposed

Unlike the EPCglobal's authentication *Access* procedure, our protocol does not use the random numbers sent in an un-encrypted form as pads to cover-code the tag's APwd. Instead these random numbers are used in association with the tag's APwd and KPwd to generate the pads. These generated pads are known only to the tag and the manufacturer. Using these pads to cover-code the APwd provides fair amount of obscurity and security to the tag's real APwd.

Secure Against Insider Attacks

In order to prevent leakage of APwd by disgruntled/compromised employees or readers, our proposed protocol does not deliver the tag's APwd to any of the stakeholder's reader. The reader (*e.g.,* at distributor's end) relays only the cover-coded APwd from both the Manufacturer, and the tag. Only the tag and the manufacturer can compute the right pads to verify the CCPwd. We can also adopt a "RFID system level check", where the system gives out an alert to the manufacturer, whenever a particular compromised reader at a stakeholder's location is continuously trying to interrogate only one particular tag with an intention to crack its APwd.

Secure Against Replay Attacks

To compute the pads, we use two random numbers each, generated by both the tag and the manufacturer. Therefore replaying a particular session would not serve any purpose for the adversary, as at least either the tag or the manufacturer would be genuine to generate unique random numbers for every session. As unique random numbers are used during different sessions, the computed pads are always unique.

Password Scalability

As mentioned before, a 32-bit password is not secure against active attacks like brute-force attack or ciphertext-only attack. We did not want to make major changes to the ratified standard, so we adhered to the 32-bit passwords and enhanced its security with very minor tweaks. Our proposed protocol can still be applicable, and more strengthened, in the case, where the length of the APwd and KPwd is extended for active-tags or tags for very expensive items.

Deployment

We assumed that, in order to secure their communication channel the *RFID Reader*, and *Manufacturer* share their digital certificates and be able to execute Signature - $Sig\{.\}$ and Encryption - $Encr\{.\}$ functions. These PKI-based certificate, encryption and signature schemes are expensive with respect to computational and performance factors. One may also feel that our proposed protocol may cause overhead to the RIFD-based supply chain management system, as the stakeholder's reader needs to securely communicate with the manufacturer in order to authenticate every tag.

To reduce this overhead, the manufacturer can setup a secure server at every stakeholder's supply chain processing facility. Only, the manufacturer can remotely access, monitor, and manage this server and also update the server with tags' Access & Kill passwords, and other required data. The stakeholder's RFID reader can now securely query this server in order to authenticate any tag in it's possession. We can also assume that the manufacturer's EPCIS is a highly resource rich entity, which is designed to take heavy computational and storage load. EPCIS is actually a network of high performance computer terminals and huge databases, whose main role is to assist a very large number of supply chain partners and consumers. We therefore assume that the manufacturers must have installed load balancing, firewall, bandwidth management, and backup mechanisms to support EPCIS. If the above assumption is not feasible for some reasons, during the first PKI-based authentication and encryption, reader and manufacturer can share a symmetric key. After which, we can secure the communication channel with only Keyed-Message Authentication Code (MAC), which reduces a great deal of burden.

Light-Weight Protocol

Our protocol does not use any special cryptographic functions. As per the EPCglobal Class 1 Gen 2 UHF RFID Protocol standard [12], the tag has the capability to compute XOR operations, generate random numbers, temporarily store random numbers and fetch the APwd and KPwd embedded within its Reserved Memory bank. Our protocol utilizes only these features.

Our protocol just needs an additional five 16-bit temporary storage memory slots within the tag, for four random numbers from the manufacturer and one for PadGen(.) function. Since Class-1 Gen-2 tags can have a 512-bit memory capacity or more (depending on the manufacturer), these additional five 16-bit temporary storage memory slots, can be easily incorporated. The one-way reader to tag authentication *Access* procedure proposed by EPCglobal requires two 16-bit temporary storage memory slots. Pad generation function utilizes the tag's (already existing) memory fetch capability, which collects the individual bits of the APwd and KPwd from the memory locations identified by the random numbers and concatenates these bits to form PADs. Therefore our proposed protocol is light weight and requires minor changes to the EPCglobal Class 1 Gen 2 UHF RFID Protocol standard.

3.4.6 Drawbacks of *TRMA* Protocol

Exploiting the weaknesses already mentioned in the security analysis of our *TRMA* protocol, Peris-Lopez *et al.* [41] showed that the *TRMA* protocol is prone to key-disclosure threat via impersonation and man-in-the middle attack scenarios and sending weak random numbers. As per this attack, the 16 least significant bits of the access password can be obtained with probability 2^{-2}, and the 16 most significant bits with a probability higher than 2^{-5}. It also shows how an attacker can recover the entire kill password with probability 2^{-2} within 4 eavesdropped sessions in the case of a passive attack, or just 2 consecutive sessions under an active attack.

Therefore in the next section we propose a *DCSTaR* protocol: Diffusion-Confusion based Light-weight Security for RFID Tag-Reader Communication.

3.5 Proposed *DCSTaR* Protocol

DCSTaR: Diffusion-Confusion based Light-weight Security for RFID Tag-Reader Communication

3.5.1 Contributions of the Proposed *DCSTaR* Protocol

The proposed protocol is called ***DCSTaR***, which takes a different approach, focusing and encouraging future research on the (above mentioned) simplified yet specific threats pertaining to low-cost passive-tags in the supply chain and those in the possession of the consumer. Our proposed protocol has the following salient features:

- Low-cost passive-tags have tightly constrained computational and memory resources. Therefore the proposed protocol is simple, cost-effective, light-weight, and practical.

- It is a challenge-response protocol.

- A light-weight protocol satisfying all the above-mentioned countermeasures and consisting of a simple cipher to encrypt the challenges from the tag.

- It utilizes only the primitives: RNG, CRC, and XOR and provides *Diffusion* and *Confusion* - the two fundamental properties for a secure cipher [47], taking in 32 bits and producing 32 ciphered bits. Diffusion: the output bits should depend on the input bits in a very complex way. Confusion: making the relationship between the key and the output bits as complex and involved as possible

- The novel *Diffusion* and *Confusion* cipher is simple to implement and execute in a tag. Its design is not as complex as a block cipher.

- Tag encrypts the challenges that are sent to the interrogator, but doesn't have to do any decryption to verify the response from the interrogator.

- The protocol may not provide a full-proof security but just enough security that justifies the cost of passive-tags.

- Unlike the other protocols, our proposed protocol is also an efficient way for consumers to verify an item is genuine or fake. Finally, it considers anonymity where it is needed the most, for the tags in the consumer's possession.

3.5.2 Setup and Assumptions

We propose an expansion to the tag's reserved memory bank (Fig.3.4). We included: a *96 bit Key*: K[95:0] and sixteen *4bit Substitution Keys*: U_0[63:60] $\sim$ U_{15}[3:0].

The key $K[95:0]$ is unique for each tag. The keys: $U_{0\sim15}$ must all be unique among each other, i.e., no two memory addresses should have the same key, satisfying 1:1 mapping between the address and the key. The criteria to choose s-boxes [26] [24] for block-ciphers can also be applied to choose the unique keys: $U_{0\sim15}$ that are secure against differential and linear cryptanalyses, therefore such (many) sets of good unique keys could be "wisely" re-used among different tags. All these keys are kept secret between tag and EPC-IS. Before initiating *DCSTaR*, the reader *Query* EPC from tag and relay it to the EPC-IS.

Addr.	*Reserved* Memory Bank Map	
$3\boxed{\text{F}}0_h \sim 3\boxed{\text{F}}3_h$	U_{15}[3:0]	16 x 4bit unique keys = 64bits *(1:1 mapping b/w* $\boxed{\textit{U Addr}}$ $\leftrightarrow$ $U_{0\sim15}$*)*
⋮	⋮	
$3\boxed{0}0_h \sim 3\boxed{0}3_h$	U_0[63:60]	
⋮	⋮	
$50_h \sim 5\text{F}_h$	Key: K[15:0]	
$40_h \sim 4\text{F}_h$	Key: K[31:16]	
$30_h \sim 3\text{F}_h$	Key: K[47:32]	
$20_h \sim 2\text{F}_h$	Key: K[63:48]	
$10_h \sim 1\text{F}_h$	Key: K[79:64]	
$00_h \sim 0\text{F}_h$	*96 bit* Key: K[95:80]	

Figure 3.4: Proposed Changes to the Tag's *Reserved* Memory Bank

3.5.3 Description of the Proposed *DCSTaR* Protocol

Our proposed *DCSTaR* protocol could be easily understood by studying the Fig.3.7. The Fig.3.8 describes Diffusion & Confusion procedure: $f_{S_1,S_2}(T_1,T_2)$=C, which obscures 32 bit T_1 and T_2 into a 32 bit cipher C. The S_1 and S_2 are seeds for the 16 bit *Round Keys* ($Y_{0\sim6}$).

Diffusion & Confusion Procedure: $f_{S_1,S_2}(T_1,T_2)$=C**: Fig.3.8**

Our Diffusion & Confusion Procedure could be considered as a much simplified/modified version of the PRESENT block-cipher [4]. The PRESENT block-cipher is a Substitution Permutation network consisting of 31 rounds. The block length is 64 bits and two key lengths of 80 and 128 bits are supported. Whereas our proposed Diffusion & Confusion Procedure is just 2 round substitution and permutation, with a block length of 32 bits and a key length of 96 bits (shorter key lengths can also be used). The purpose of our Diffusion & Confusion Procedure is not to provide a secure and full-proof encryption/decryption but just sufficient to *obscure* the 32 bit challenge and response pair (T_1,T_2). Therefore our Diffusion & Confusion Procedure is not as cryptographically strong than the PRESENT block-cipher, but we strongly believe that our procedure would justify the purpose of extremely low-cost passive tags in the supply chain management.

Round Key The 16 bit round keys $Y_0 \cdots\ Y_6$ are calculated as below:

$Y_0 = CRC(K, S_1, S_2)$
$Y_n = CRC(Y_{n-1}, K, S_1, S_2)$ for $n = 1, 2, 3, 4, 5, 6$

The PRESENT block-cipher [4] can support key sizes is 80 bits or 128 bits. The round keys are 64 bits each generated by taking 64 leftmost bits of the key from the key register and then the key register is rotated by 61 bit positions to the left, the left-most four bits are passed through the present S-box, and the round-counter value is exclusive-ored with 15^{th} through 19^{th} bits of the key.

Whereas our Diffusion & Confusion Procedure round keys are just 16 bits each and they are calculated by using the already existing CRC primitive in the low-cost tags.

Mapping Function (MU()) The *MU*() is a mapping function where 4 input bits are replaced by a *4 bit unique Key*; *e.g.,* please refer (Fig.3.5), where MU(**0011**) = MU($\mathbf{3}_h$) = ($\mathbf{3}\mathbf{3}0_h$∼$\mathbf{3}\mathbf{3}\mathbf{3}_h$) = $U3$[52:54]. As mentioned above, the *MU*() can be considered like a 4 Substitution Box (S-box) which provides non-linearity and is applied 16 times in parallel in each of the two round Diffusion & Confusion Procedure. Whereas PRESENT block-cipher [4] applies 4 S-box 16 times in parallel in each of its 31 rounds. Below we briefly mentions the properties of S-boxes.

Properties of S-boxes

The properties of the S-boxes [49] in a cipher are important in the consideration of a cipher's security against differential cryptanalysis [3] and linear cryptanalysis [31]. An $m \times n$ S-box, S, performs a mapping from an m-bit input X to an n-bit output Y . Considering all S-boxes, $\{S_i\}$, in a cipher, the maximum differential probability p_s is defined as:

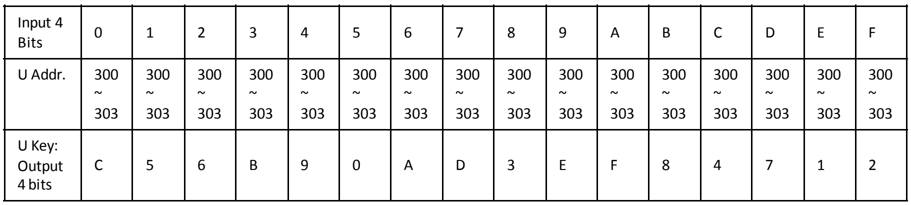
$$p_s = \max_i \max_{\Delta X \neq 0, \Delta Y} prob\{S_i(X) \oplus S_i(X \oplus \Delta X) = \Delta Y\}$$

where "$\oplus$" denotes a bitwise XOR and "Δ" denotes a bitwise XOR difference. The maximum linear probability is defined as:

$$q_s = \max_i \max_{\Gamma X \neq 0, \Gamma Y} (2 \times prob\{X \cdot \Gamma X = S_i(X) \cdot \Gamma Y\} - 1)^2$$

where "$\cdot$" denotes a bitwise inner product and ΓX and ΓY denote masking variables. All 4×4 S-boxes are assumed to satisfy $p_s, q_s \leq 2^{-2}$.

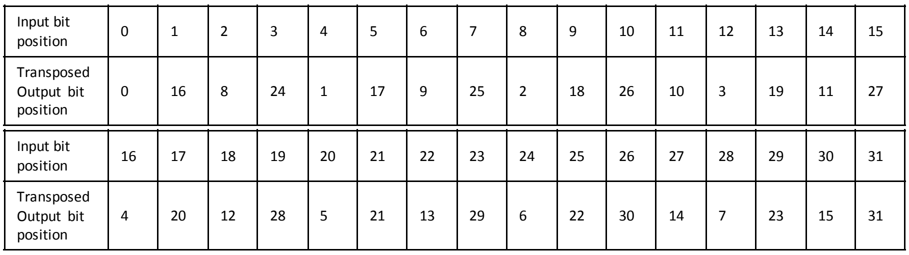
Example S-box based on the one used in PRESENT Cipher [4] We could use the similar S-box as shown in Fig.3.5 proposed in the PRESENT block cipher, which satisfies the above properties of S-boxes.



Input 4 Bits	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U Addr.	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303	300 ~ 303
U Key: Output 4 bits	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Figure 3.5: Example S-box based on the one used in PRESENT block cipher [4]

Bit Transpose The *Bit Transpose* used in our Diffusion & Confusion Procedure is a 32 bit narrowed down version of 64 bit permutation table used in PRESENT block cipher [4] as shown in Fig.3.6.



Input bit position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Transposed Output bit position	0	16	8	24	1	17	9	25	2	18	26	10	3	19	11	27
Input bit position	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Transposed Output bit position	4	20	12	28	5	21	13	29	6	22	30	14	7	23	15	31

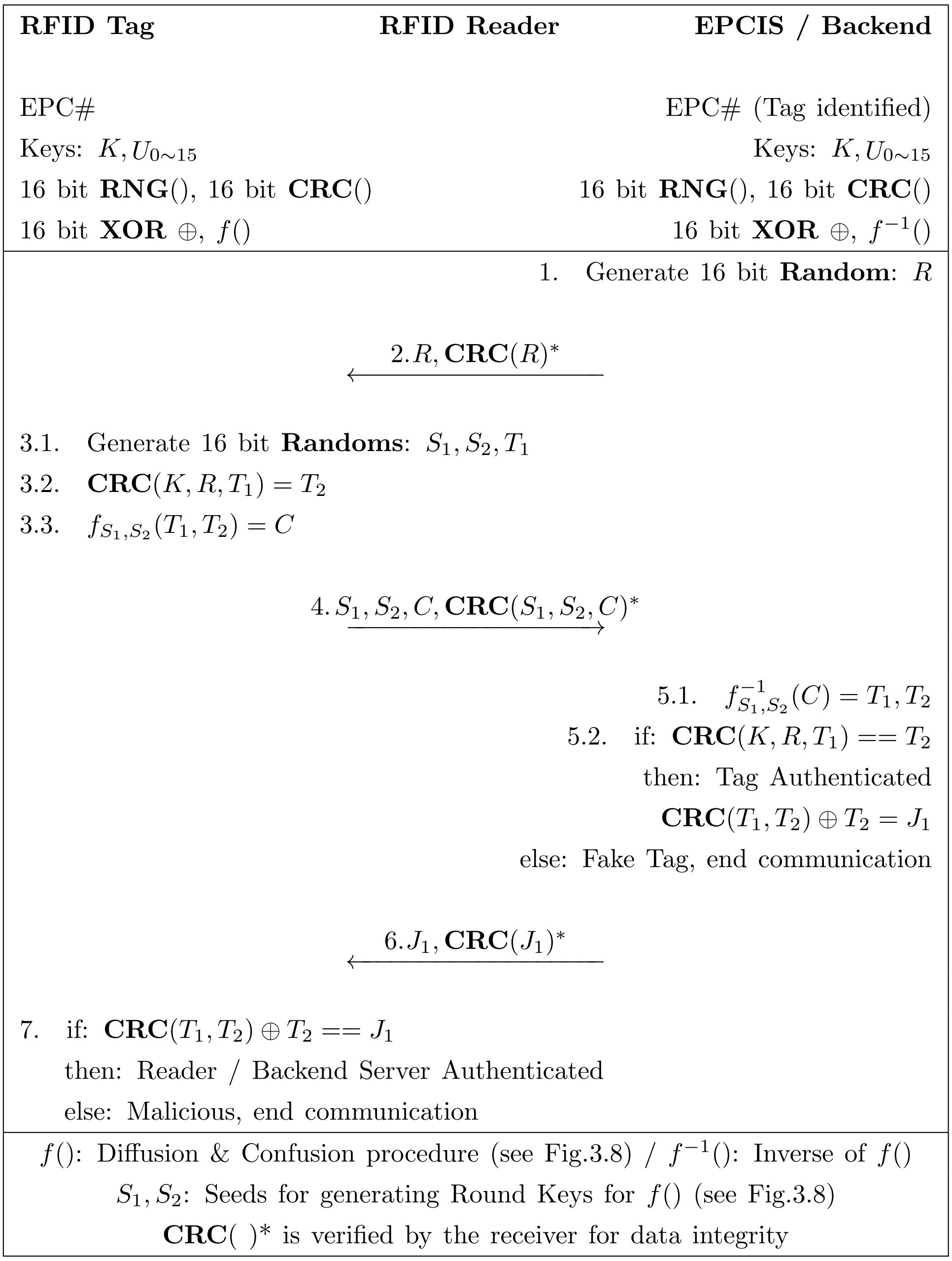
Figure 3.6: The 32 bit version of permutation table used in PRESENT block cipher [4]

The Inverse of Diffusion & Confusion Procedure $f^{-1}_{S_1,S_2}(C)$: In our DCSTaR protocol (Fig.3.7) the inverse of Diffusion & Confusion Procedure is done only by the server/EPCIS. Referring to Fig.3.8 it is fairly easy to compute this inverse. First we generate all the $(Y_{0\sim6})$ and proceed with the $f_{S_1,S_2}()$ procedure bottom-up, until T_1 and T_2 are recovered. Here, $MU^{-1}(U_{\mathbf{3}}[52{:}54]) = (\mathbf{330}_h{\sim}\mathbf{333}_h) = \mathbf{3}_h = \mathbf{0011}$.

3.5.4 Analysis of Proposed *DCSTaR* Protocol

Tag←reader→EPCIS mutual authentication

Readers and EPCIS authenticate and secure their communications via TLS/SSL protocol. An adversary can't randomly pick T_1 and T_2, as T_2 and C can only be computed by a tag possessing R, T_1, and the keys $(K, U_{0\sim15})$. Only the EPCIS possessing these keys can recover T_1, T_2 and verify $CRC(K, R, T_1) == T_2$ and then compute J_1. The R, T_1, S_1, and S_2 are unique per transaction and are all linked together through out the protocol to thwart any reply attack.



RFID Tag **RFID Reader** **EPCIS / Backend**

EPC#
Keys: $K, U_{0\sim15}$
16 bit **RNG**(), 16 bit **CRC**()
16 bit **XOR** $\oplus$, $f()$

EPC# (Tag identified)
Keys: $K, U_{0\sim15}$
16 bit **RNG**(), 16 bit **CRC**()
16 bit **XOR** $\oplus$, $f^{-1}()$

1. Generate 16 bit **Random**: R

$\xleftarrow{2. R, \mathbf{CRC}(R)^*}$

3.1. Generate 16 bit **Randoms**: S_1, S_2, T_1
3.2. $\mathbf{CRC}(K, R, T_1) = T_2$
3.3. $f_{S_1,S_2}(T_1, T_2) = C$

$\xrightarrow{4. S_1, S_2, C, \mathbf{CRC}(S_1, S_2, C)^*}$

5.1. $f^{-1}_{S_1,S_2}(C) = T_1, T_2$
5.2. if: $\mathbf{CRC}(K, R, T_1) == T_2$
then: Tag Authenticated
$\mathbf{CRC}(T_1, T_2) \oplus T_2 = J_1$
else: Fake Tag, end communication

$\xleftarrow{6. J_1, \mathbf{CRC}(J_1)^*}$

7. if: $\mathbf{CRC}(T_1, T_2) \oplus T_2 == J_1$
then: Reader / Backend Server Authenticated
else: Malicious, end communication

$f()$: Diffusion & Confusion procedure (see Fig.3.8) / $f^{-1}()$: Inverse of $f()$
S_1, S_2: Seeds for generating Round Keys for $f()$ (see Fig.3.8)
CRC( )* is verified by the receiver for data integrity

Figure 3.7: Proposed *DCSTaR* Protocol

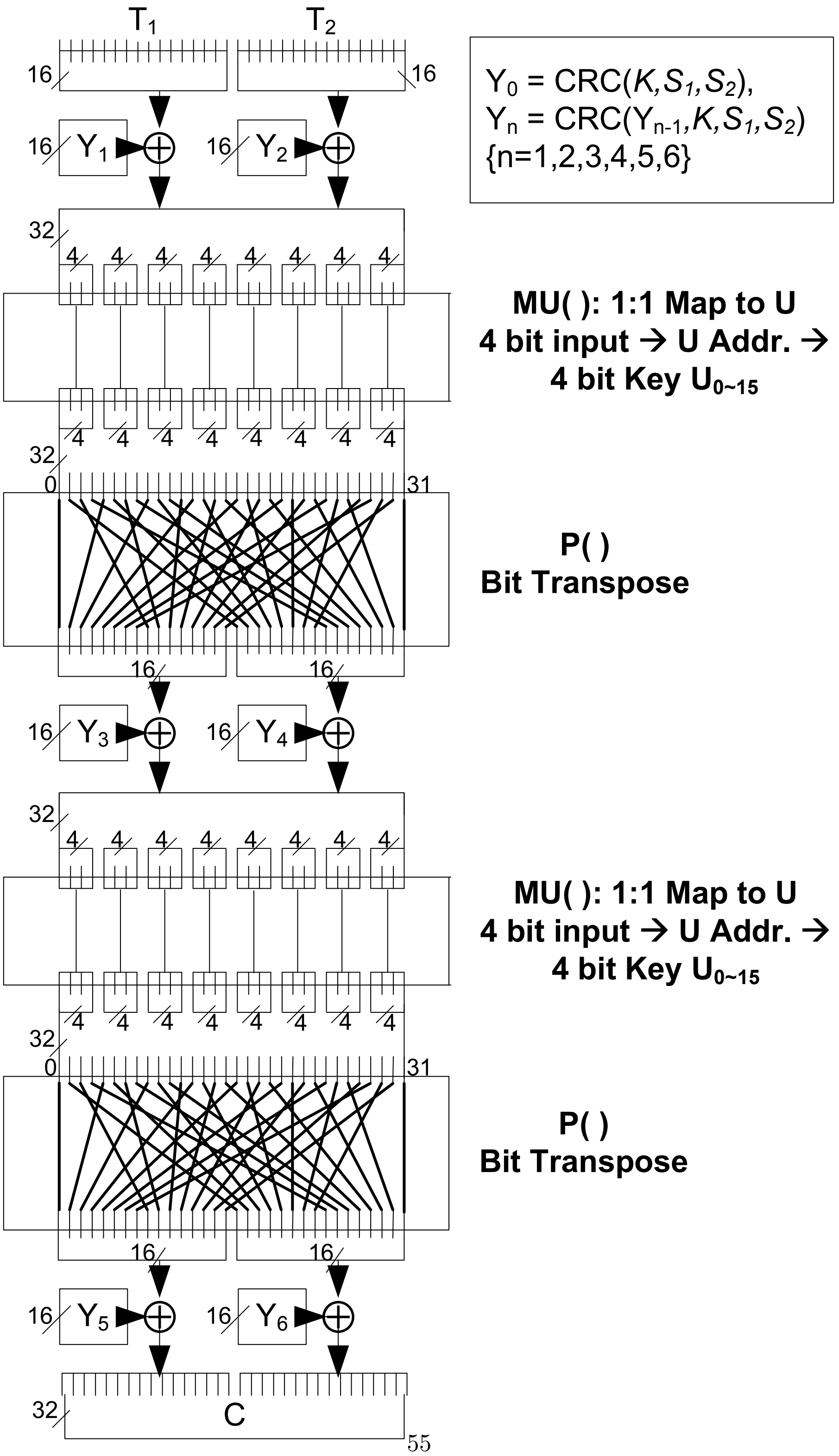


Figure 3.8: Diffusion & Confusion procedure: $f()$

Data Integrity

Though CRC()* provides data integrity check, any modification to steps 2, 4 & 6 would fail the authentication process. If Z represents a command or encrypted-user-data from the EPCIS, then EPCIS computes $CRC(Z, T_1, T_2) \oplus T_2 = J_1$. An adversary can intercept, modify Z to Z', and send $\{J_1, Z', CRC(J_1, Z')\}$ to the tag. But the tag can detect this because $CRC(Z', T_1, T_2) \oplus T_2 \neq J_1$.

Secure key-distribution and key-protection

DCSTaR protocol can be done online with manufacturer's EPCIS. Else, the manufacturer can remotely access, monitor, and manage a server at a large-scaled stakeholder's supply-chain processing facility and update this server with relevant tags' keys. The steps 2, 4, & 6 do not expose any of the keys.

Reader relaying only obscured data

The steps 4 & 6 do not expose any sensitive data like the keys $K, U_{0\sim15}$ and the challenges T_1, T_2 to the readers.

Tag Verification and Tag Anonymity for Consumers

A consumer can use his/her RFID reader-enabled portable device (e.g., mobile phone) to Query and send R to the tag (as in Step 1-Fig.3.7). This RFID reader-enabled portable device obtains the EPC,S_1, S_2, and C from the tag, and send this data along with the R to the EPCIS via 3G/4G network or Wi-Fi connection. EPCIS would then verify C and replies to the device whether the item is genuine or fake. In here neither the tag's keys nor tag's sensitive data are exposed to the customer.

After purchasing an item the consumer would obtain the tag keys: $K, U_{0\sim15}$ from the store and store them into his/her device. Using these keys the consumer can execute the DCSTaR protocol and read-lock the EPC memory bank using the Lock command. As a result the tag no longer emits its EPC number, thus protecting the privacy of the consumer from eavesdropping malicious readers.

Since the tag no longer emits its EPC number, the consumer executes DCSTaR protocol by just sending R to the tag. The tag responds with its 64 bit C, which now becomes the tag's Pseudo-ID (PID). The consumer uses this PID to do a brute force search of all the tags in his/her possession that give out the same C and thus arrives at the correct

EPC number. A consumer would not have that many items/tags; therefore we can assume that there would be no PID collisions or computationally intensive database searches.

Data Confidentiality

Though R is exposed, its just one among two other secrets K and T_1 needed to compute T_2. The 32 bit C obscures T_1 and T_2. The 16 bit J_1 are neither guessable nor exposing any sensitive data.

Additional feature: A tag may store few bytes of stakeholder's (user) data. We suggest that the reader Writes already encrypted user data it received from EPCIS. At a later stage, the reader can retrieve the stored encrypted user data from the tag and relay it to the EPCIS to be decrypted. Thus the data is secured in the tag and also while writing/reading to/from the tag.

Differential Cryptanalysis

The resistance of a block cipher against differential cryptanalysis [3] depends on the maximum probability of differential characteristics, which are paths from the plaintext difference to the ciphertext difference. based on the similar analysis provided for PRESENT block cipher [4], assuming that we have at least 4-active S-boxes in our 2-round diffusion-confusion procedure. The maximum differential probability of a our S-box is 2^{-2} and so the probability of a 2-round differential characteristic with 4-active S-boxes is bounded by 2^{-8}. Once again this is very weak in terms of a strong secure encryption cipher, but for in an extremely fast paced, RFID supply chain processing environment, it is might be sufficient. However if we assume more than 4-active S-boxes (among the 16 S-boxes in the 2-rounds) as shown in the table 3.2 below we can achieve better results. This was evident when we carried out statistical analysis of our diffusion-confusion procedure, which passed all the tests.

Linear Cryptanalysis

We describe the case of linear cryptanalysis of our diffusion-confusion procedure based on the similar analysis provided for PRESENT block cipher [4]. Here we analyze the best linear approximation of our two round diffusion-confusion procedure.

Matsui's piling-up lemma [31] estimates the bias of a linear approximation involving n S-boxes to be

$$2^{n-1}\prod_{i=1}^{n}\varepsilon_i,$$

Table 3.2: Differential Cryptanalysis Data

No. of Active S-boxes	Differential Characteristic is Bounded By	Value
5	2^{-10}	0.0009765625
6	2^{-12}	0.000244140625
7	2^{-14}	0.00006103515625
8	2^{-16}	0.0000152587890625
9	2^{-18}	0.000003814697265625
10	2^{-20}	0.00000095367431640625

where the values ε_i are the individual bias of each (independent) S-box. According to the design principles of our S-box, the bias of all linear approximations is less than 2^{-2}. Let $\varepsilon_{1\mathrm{R}}^{(j)}$ denote the bias of a linear approximation over 1-round involving j active S-boxes. Therefore the overall bias for a 1-round and 4-active S-boxes approximation can be bounded as follows:

$$\varepsilon_{1\mathrm{R}}^{(4)} \leq 2^3 \times (2^{-2})^4 = 2^{-5}$$

Therefore the maximal bias of a linear approximation of 1-round and and 4-active S-boxes of confusion-diffusion procedure is:

$$\varepsilon_{1\mathrm{R}}^{(4)} \leq \tfrac{1}{2^5}$$

we can now bound the maximal bias of a 2-round and 4-active boxes linear approximation $\varepsilon_{2\mathrm{R}}^{(4)}$ by

$$\varepsilon_{1\mathrm{R}}^{(4)} \leq \tfrac{1}{2^5}$$

$$2^1 \times (\varepsilon_{1\mathrm{R}}^{(4)})^2 = 2^1 \times (2^{-5})^2 = 2^{-9}$$

$$\varepsilon_{2\mathrm{R}}^{(4)} = 2^{-9}$$

Therefore to mount a key recovery attack, linear cryptanalysis of our cipher would require of the order of $(2^9)^2 = 2^{18} = 262144$ known plaintext/ciphertexts. This is very weak in terms of a strong secure encryption cipher, but for in an extremely fast paced, RFID supply chain processing environment, it is not feasible to continuously eavesdrop on one particular tag-reader communication channel for a time long enough time to obtain such number of known plaintext/ciphertexts. Several bulks of items pass through the readers with in a very short interval of time. However if we assume more than 4-active S-boxes (among the 16 S-boxes in the 2-rounds) as shown in the table 3.3 below we can achieve better results. This was evident when we carried out statistical analysis of our diffusion-confusion procedure, which passed all the tests.

Table 3.3: Linear Cryptanalysis Data

No. of Active S-boxes (j)	$\varepsilon_{2R}^{(j)}$	Reqd. No. of Known Plaintext/Ciphertexts
5	2^{-11}	4194304
6	2^{-13}	67108864
7	2^{-15}	1073741824
8	2^{-17}	17179869184
9	2^{-19}	274877906944
10	2^{-21}	4398046511104

Empirical Statistical Testing

To justify the use of only two round diffusion-confusion and the strength of $f_{S_1,S_2}(T_1,T_2) = C$ procedure, we used TestU01 - a software library of 'utilities for empirical statistical testing of RNGs' implemented in the C language [29]. It is comprehensive, frequently updated, and encompasses most of the other test-suites. Different quality criteria are used for RNGs in cryptology-related applications and for gambling machines in casinos. In these settings, an additional concern is unpredictability of the forthcoming numbers. The theoretical analysis of RNGs in cryptology is usually asymptotic, in the framework of computational complexity theory. Nonlinear recurrences and/or output functions are used, which prevents one from measuring the uniformity of the set Ψ_t of all t-dimensional vectors of t successive values that can be produced by the generator, from all its possible initial states (or seeds). As a result, empirical testing is even more necessary.

We subjected several 150 megabytes of C values obtained under multiple trails and different keys to the following batteries of test: SmallCrush, PseudoDIEHARD, Alphabit, BlockAlphabit, Rabbit, and FIPS-140-2 (NIST std.: security requirements for cryptographic modules). The batteries Rabbit, Alphabit and BlockAlphabit are for binary sequences from a cryptographic pseudorandom generator. Most of these batteries return p-values

Summary results of FIPS-140-2 Number of bits: 20000			
Test	s-value	p-value	FIPS Decision
Monobit	9979	0.61	Pass
Poker	18.87	0.22	Pass
0 Runs, length 1	2508		Pass
0 Runs, length 2	1233		Pass
0 Runs, length 3	634		Pass
0 Runs, length 4	306		Pass
0 Runs, length 5	168		Pass
0 Runs, length 6+	152		Pass
1 Runs, length 1	2450		Pass
1 Runs, length 2	1300		Pass
1 Runs, length 3	653		Pass
1 Runs, length 4	307		Pass
1 Runs, length 5	152		Pass
1 Runs, length 6+	139		Pass
Longest run of 0	13	0.5	Pass
Longest run of 1	13	0.5	Pass
All values are within the required intervals of FIPS-140-2			

Table 3.4: NIST (FIPS_140_2) package: testing & certification of RNGs for cryptographic applications

for all its tests, and those that are within the [0.001 0.9990] range are passed. To speed-up these tests, we utilized cluster computing and implemented DCSTaR as a parallel C program. DCSTaR passed all these batteries of tests.

FIPS_140_2 Test Suite As shown in the Table.3.4, the NIST package contains 15 tests, oriented primarily toward the testing and certification of RNGs used in cryptographic applications [45].

The TestU01 [29] provides the batteries of tests like the Rabbit, Alphabit and Block-Alphabit to test the random bits of a cipher output. These batteries of tests encompass several statistical tests such as: entropy, linear complexity, distinct bit patterns, fourier coefficients, autocorrelations, run and gap tests, serial tests, rank of a binary matrix, longest

Test Name	Avg. p-value
smultin_MultinomialBitsOver	0.70
snpair_ClosePairsBitMatch	0.62
svaria_AppearanceSpacings	0.51
scomp_LinearComp	0.61
scomp_LempelZiv	0.59
sspectral_Fourier1	0.52
sspectral_Fourier3	0.56
sstring_LongestHeadRun	0.66
sstring_PeriodsInStrings	0.69
sstring_HammingWeight	0.61
sstring_HammingCorr	0.53
sstring_HammingIndep	0.75
sstring_AutoCor	0.50
sstring_Run	0.60
smarsa_MatrixRank	0.71
swalk_RandomWalk1	0.80

Table 3.5: Rabbit, Alphabit and BlockAlphabit Batteries of Tests

run of 1's, hamming weights, and random walk tests, these tests determine if a cipher output is resistant to linear and differential cryptanalysis. We describe each of these tests and the table.3.5 presents the average p-values of these tests, which prove that our diffusion-confusion procedure function $f()$ is indeed resistant against Linear and Differential Cryptanalysis.

Let us consider the diffusion-confusion procedure function $f()$. We want to test the null hypothesis that the individual bits of the cipher output $C = C_0, C_1, C_2, \cdots$, are independent and take the values 0 or 1 with equal probability. The standard procedure for doing this in TestU01 [29] is to take bits $r+1, ..., r+s$ from each cipher output, i.e., skip the first r bits and take the s bits that follow, for some integers $r \geq 0$ and $s \geq 1$, and concatenate all these bits in a long string. The values of r and s are parameters of the tests. Tests on binary sequences have been designed primarily in the area of cryptology, where high entropy and complexity are key requirements. The multinomial tests are essentially entropy tests.

Linear Complexity One way of testing the complexity is to examine how the *linear complexity* L_ℓ of the first ℓ bits of the sequence increases as a function of ℓ. The linear complexity L_ℓ is defined as the smallest degree of a linear recurrence obeyed by the sequence. It is nondecreasing in ℓ and increases by integer-sized jumps at certain values of ℓ.

Jump Complexity The jump complexity test counts the number J of jumps in the linear complexity, J is approximately normally distributed with mean and variance. The jump size test counts how many jumps of each size there are and compares these frequencies to the theoretical distribution (a geometric with parameter 1/2) by a chi-square test.

A different type of test, used in the NIST suite, uses a two-level procedure with a large N and relatively smaller n. It computes N realizations of L_n, counts how many times each value has occurred, and uses a chi-square test to compare these counts to their theoretical expectations.

Distinct Bit Patterns Complexity can also be tested by counting the number W of distinct bit patterns that occur in the string. This number measures the compressibility of the sequence. W is approximately normally distributed with mean $n/\log_2 n$ and variance $0.266n/(\log_2 n)^3$.

Fourier Coefficients Spectral tests on a binary sequence of $n = 2^k$ bits compute (some of) the discrete Fourier coefficients, which are complex numbers defined by

$$f_\ell = \sum_{j=0}^{n-1} (2b_j - 1)e^{2\pi\iota j\ell/n}, \ell = 0, 1, \cdots, n-1,$$

where $\iota = \sqrt{-1}$. Let $|f_\ell|$ be the modulus of f_ℓ. A first test, counts the number O_h of $|f_\ell|$'s that are smaller than some constant h, for $\ell \leq n/2$. For large n and $h = \sqrt{2.995732274\text{n}}$, O_h is approximately normal with mean $\mu = 0.95n/2$ and variance $\sigma^2 = 0.05\mu$.

Autocorrelations The sample autocorrelation of *lagℓ* in a bit sequence, defined as

$$A_\ell = \sum_{i=0}^{n-\ell-1} C_i \oplus C_{i+\ell}$$

A_ℓ has the binomial distribution with parameters $(n-\ell, 1/2)$, which is approximately normal when $n - \ell$ is large.

Run and Gap Tests Every binary sequence has a run of 1's, followed by a run of 0's, followed by a run of 1's, and so on, or vice-versa if it starts with a 0. Suppose we collect the lengths of all runs of 1's and all runs of 0's until we have a total of $2n$ runs (n of each type). We count the number of runs of 1's of length j and the number of runs of 0's of length j, for $j = 1, \cdots, k$ for some integer k (regrouping the runs of length larger than k with those of length k) and apply a chi-square test on these $2k$ counts. Since any given run has length j with probability 2^{-j} , we readily know the expected number of runs of each length. Note that each run of 0's is a gap between the occurrence of 1's, and vice-versa, so this test can also be seen as a gap test for binary sequences.

We now consider tests that try to detect "dependencies" in bit strings of length m, in the sense that some of the 2^m possibilities are much more likely than others. All these tests use essentially (indirectly) the following pattern: regroup the 2^m possibilities for the bit string into, say, k categories, count how many of the n strings fall in each category, and compare the results with the expectations.

Serial Tests Number the possible m-bit strings from 0 to $k - 1 = 2^m - 1$ and let X_j be the number of occurrences of string j in the n strings. The same set of test statistics like chi-square, entropy, collisions, *etc.*, can be used.

Rank of a Binary Matrix A powerful test to detect linear dependencies between blocks of bits is the matrix rank test. Fill up a $k \times \ell$ binary matrix row by row with a bit string of length $m = k \times \ell$, and compute the rank R of this binary matrix (the number of linearly independent rows). Repeat this n times and compare the empirical distribution of the n realizations of R with its theoretical distribution with a chi-square test.

Longest Run of 1's The longest head run test is a variant of the run test that looks at the length Y of the longest substring of successive 1's in a string of length m. This is repeated n times and the empirical distribution of the n realizations of Y is compared with its theoretical distribution by a chi-square test.

Hamming Weights . To measure the clustering of 0's or of 1's in a bit sequence, we can examine the distribution of the Hamming weights of disjoint subsequences of length m. More clustering should lead to higher variance of the Hamming weights and to positive dependence between the Hamming weights of successive subsequences.

Random walk tests From a bit sequence of length ℓ, we can define a random walk over the integers as follows: the walk starts at 0 and at step j, it moves by 1 to the left if $C_j = 0$, and by 1 to the right if $C_j = 1$. If we define $S_0 = 0$ and $S_k = \sum_{j=1}^{k} (2b_j - 1)$ for $k > 0$, then the process $\{S_k, k \geq 0\}$ is this random walk.

3.5.5 Performance Aspects of the Proposed *DCSTaR* Protocol

- *DCSTaR* achieves tag←reader→EPCIS mutual authentication in just three communication steps 2, 4, & 6 (Fig.3.7).

- DCSTaR strictly utilizes only the RNG, XOR, and CRC light-weight primitives/operations.

- The mapping function $MU()$ is implemented in a way that the input bits to this function are used as a memory address to replace them with the KEY stored in that address. This simple approach requires no additional hardware implementation like the substitution and inverse tables.

- The tag needs to execute only $f()$ procedure but not $f^{-1}()$.

- DCSTaR protocol does require an additional memory space to accommodate the keys and to execute the diffusion-confusion procedure. However we have to assume that low-cost passive item-tags can have a memory capacity of several bytes e.g., 512 bytes, therefore DCSTaR's additional memory requirement can be easily incorporated.

Comparison between the *DCSTaR* and C1G2 Access Procedure Air-Interface Commands

In this section we compared the *DCSTaR* Air-Interface Commands with the actual C1G2 Access Procedure Air-Interface Commands. The fig. 3.9 depicts the C1G2 Access Procedure's air-interface commands. It can be noticed that it takes 14 steps to achieve only just a "one-way" reader authentication. Whereas the fig. 3.10, which depicts our *DCSTaR* protocol achieves a tag-reader mutual authentication in just 10 steps and there is no need for additional commands. The *DCSTaR* can thus be easily incorporated into the C1G2 protocol.

3.5.6 Comparison with Related Work

Comparison with Optimized/Light-Weight Block Ciphers

Some of the previously proposed solutions are based on hash functions,[2, 6, 46, 19, 44] and optimized implementations of block (AES, DES) [4], [43], [21], [15], and stream ciphers, but passive low cost tags are not capable of executing such computationally intensive functions due to their constrained resources.

Optimized/light-weight ciphers such as PRESENT [4] or HIGHT [21] require only 1,570 or 3,048 gate equivalents respectively [42]. However, this number of gates may still cause considerable burden on low-cost passive tags. We compare our *DCSTaR* protocol's diffusion-confusion procedure $f()$ with the PRESENT block cipher [4] .

The PRESENT cipher is an SP-network and consists of 31 rounds. The block length is 64 bits and two key lengths of 80 and 128 bits are supported. Whereas our diffusion-confusion procedure $f()$ has a 96 bit key and just two rounds of very basic and fundamental diffusion and confusion procedure. Our diffusion-confusion procedure $f()$ is used only to obscure random 32 bit challenge and response pair from the tag. With just a 32 bit cipher output our diffusion-confusion procedure $f()$ is not designed to be an extremely secure encryption, instead to provide just enough obscuring for the tag's challenge and response pair so that it can be easily implemented on a low-cost passive tag. We also emphasize that we proposed a novel DCSTaR protocol (Fig.3.7) and the diffusion and confusion procedure (Fig.3.8) is just a part of the DCSTaR protocol, therefore we claim that any other efficient 32 bit cipher can be easily incorporated into our DCSTaR protocol in place of our diffusion-confusion procedure $f()$. It is evident from our DCSTaR protocol that the purpose of obscuring $T1, T2$ can be done by any cipher and the rest of the protocol still holds good.

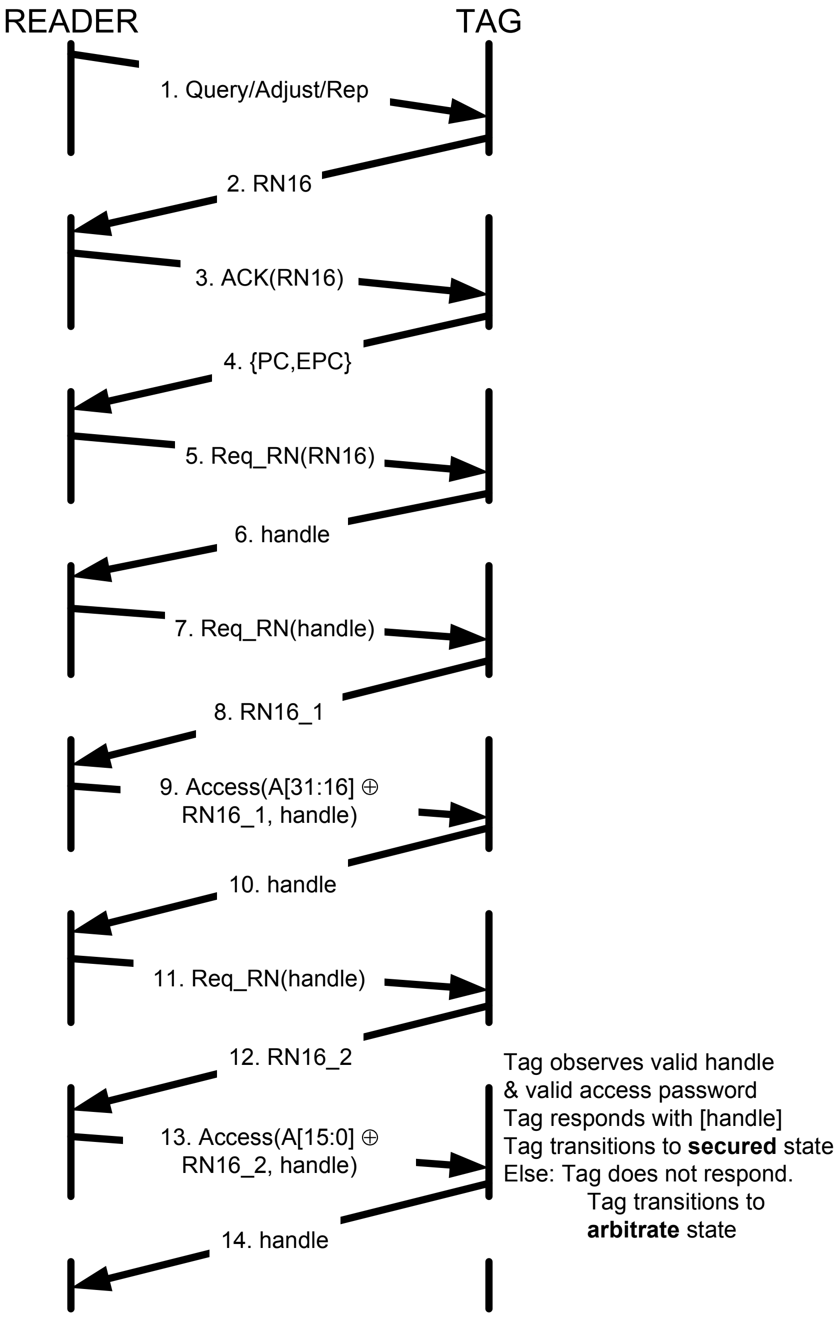


Figure 3.9: C1G2 Access Procedure Air-Interface Commands

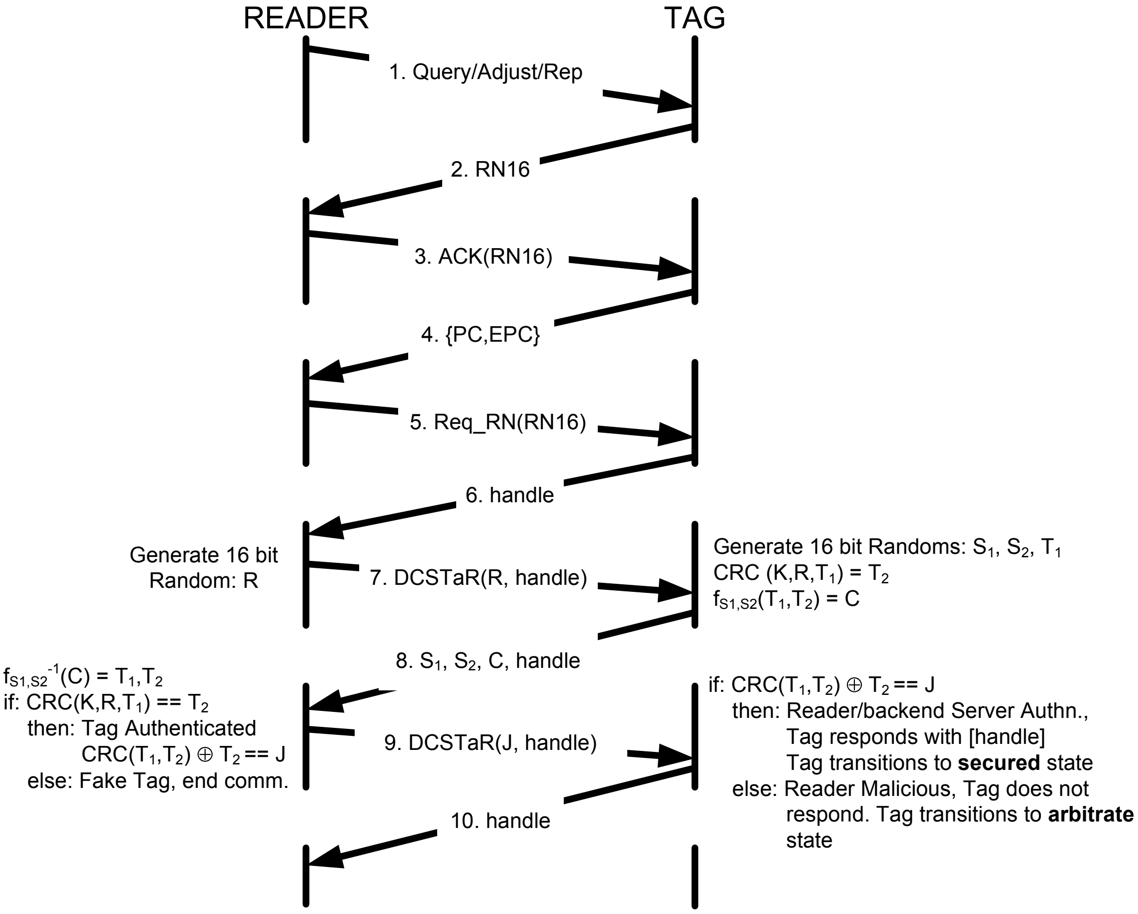

READER
TAG
1. Query/Adjust/Rep
2. RN16
3. ACK(RN16)
4. {PC,EPC}
5. Req_RN(RN16)
6. handle
Generate 16 bit Random: R
7. DCSTaR(R, handle)
Generate 16 bit Randoms: S_1, S_2, T_1
CRC (K,R,T_1) = T_2
$f_{S1,S2}(T_1,T_2) = C$
8. S_1, S_2, C, handle
$f_{S1,S2}^{-1}(C) = T_1,T_2$
if: CRC(K,R,T_1) == T_2
then: Tag Authenticated
CRC(T_1,T_2) $\oplus$ T_2 == J
else: Fake Tag, end comm.
9. DCSTaR(J, handle)
if: CRC(T_1,T_2) $\oplus$ T_2 == J
then: Reader/backend Server Authn.,
Tag responds with [handle]
Tag transitions to secured state
else: Reader Malicious, Tag does not respond. Tag transitions to arbitrate state
10. handle

Figure 3.10: DCSTaR Air-Interface Commands

Table 3.6: Comparing DCSTaR with PRESENT and HIGHT Block Ciphers

Cipher	Gates Equivalent	Cycles per block
DCSTaR's Diffusion and Confusion Procedure	620	2
PRESENT [4]	1,570	32
HIGHT [21]	3,048	34

To the best of our knowledge, we roughly calculated the Gate Equivalents (GE) of our diffusion-confusion procedure $f()$: Key XOR: 200 GE; Mapping Function: 113 GE; Bit Transpose: 0 GE (due to hardware wiring); Registers: 307 GE, which turns out to be 620 gate equivalents and our diffusion-confusion procedure $f()$ requires 2 clock cycles to produce 32 bit block. The table 3.6 does provide a comparison, but it is not to show that our diffusion-confusion procedure is much efficient than the PRESENT or HIGHT block ciphers. As mentioned above our diffusion-confusion procedure $f()$ must still be thoroughly verified and precisely analyzed with respect to its security and hardware performance.

Comparison with Light-Weight Protocols

In here let us discuss only light-weight protocols [36] that utilize light-weight primitives like the Random Number Generator (RNG), Cyclic Redundancy Check (CRC), modular addition and bit-wise operators such as XOR, AND, OR, rotate, *etc.* However [34], [5], and [33] attacked the most recent of such light-weight protocols and proved that they are susceptible to: key disclosure, man-in-the-middle, de-synchronization, replay, and impersonation attacks. On the other hand the most of these protocols failed to consider the insider attack, thereby allowing the readers to possess the secret keys.

Juels *et al.* [22] first proposed HB+ protocol,which is based on 'inner dot product' and satisfying NPhard - 'Learning Parity with Noise' problem. HB+ and its later improvements have all been proved insecure against the man-in-the-middle attack [18], [33], exposing the tag's secret keys and these protocols consider only tag (not reader) authentication. They also require a minimum of: 500 bit keys, many 250 bit challenge strings, and a noise parameter of 0.25 [50], all of which may not be practical for low-cost passive-tags.

Karthikeyan *et al.* [23] first proposed a protocol that utilizes matrix-multiplication and XOR, but Chien *et al.* [7] showed it suffers from de-synchronization of session keys and replay (impersonation) attacks and proposed an improvement that uses RNG, CRC, and XOR. However, Peris-Lopez *et al.* [37] proved that [7] is still not secure from the very same attacks and later proposed three novel protocols that use XOR, AND, OR, and addition mod 2^m: LMAP [38], M2AP [40], and EMAP [39], but Li *et al.* [27], [28] proved that these protocols again suffer from de-synchronization and full-disclosure of tag's secret information. Then again, Chien *et al.* [8] pointed out the weakness of [27] and like-wise Arco *et al.* [10] proved that SASI protocol [9] that additionally used rotate operation is also prone to the above mentioned weaknesses. The table 3.7 provides a comparison of our *DCSTaR* protocol with these related light-weight protocols.

Drawbacks of providing tag anonymity at supply chain

To achieve tag anonymity, previous protocols prevent the tag from emitting its EPC, instead use 'per-transaction-updatable' tag Pseudo-IDs (PIDs). The innovative measures proposed by [5] and [35] to: minimize exhaustive computation and DB search for a particular PID, restore PID synchronization between the tag and EPCIS, resolve PID collisions in the DB, and session unlinkability; can still be a bit overkill/impractical, causing overhead, delay, and uncertainty at a large-scaled and fast-paced supply-chain processing. The speed, accuracy, and atomicity achieved with EPC is lost and as per the EPCglobal, it is the EPC that is used as an URL along with Object Naming Server to locate the appropriate EPCIS. Therefore, using PIDs at the supply-chain level defeats the very purpose of RFID. We consider that though the EPC is exposed at supply-chain level, we can alleviate the threats that demand the need for tag anonymity at the supply-chain level by simply allowing only authorized (stakeholders) readers to access EPCIS. This prevents malicious readers from obtaining critical detailed information about items from the EPCIS.

Table 3.7: Comparison with Related Work

	Atk	R-UR	R-RA	R-MMA	DC	DI	R-DA	R-DSA	OPER
Original: *Access* (Fig.3.1) [12]	-	X	X	X	X	X	X	-	RNG, CRC, $\oplus$
Karthikeyan *et al.* [23]	[7]	X	X	X	✓	X	✓	X	Matrix-multi, $\oplus$
Chien *et al.* [7]	[37]	✓	✓	X	X	X	X	X	RNG, CRC, $\oplus$
LMAP, M2AP [38, 40] *& EMAP* [39]	[27] [28]	X	X	X	X	X	X	X	$\oplus, \vee, \wedge, +$ (mod 2^m) $\oplus, \vee, \wedge$
SASI [9]	[1]	X	✓	X	X	X	X	X	$\oplus, \vee, \wedge, +$, Rotate Left
Proposed: *TRMA* (Fig.3.2)	[41]	✓	✓	X	X	X	X	-	RNG, CRC, $\oplus$
Proposed: *DCSTaR* (Fig.3.7)	-	✓	✓	✓	✓	✓	✓	✓	RNG, CRC, $\oplus$

Atk: Attack Paper — R-UR: Resist Untrusted Readers — R-RA: Resist Reply Atk — R-MMA: Resist Man-in-the-Middle Atk

R-KDA: Resist Key Disclosure Atk — DC: Data Confidentiality — DI: Data Integrity — R-DSA: Resist De-Synchronization Atk

3.6 Discussion

Our *DCSTaR* protocol requires a precise practical design and implementation on a CMOS and evaluate its die size, clock cycles, throughput, and power consumption. The *DCSTaR* protocol must be thoroughly analyzed in terms of air-interface such as interacting with the physical layer, collision arbitration algorithm, and data-coding methodology. Though the *DCSTaR* protocol is simple and efficient it does require additional memory space to store the 96 bit key and execute the diffusion-confusion procedure. We are confident that *DCSTaR* would encourage further research especially on item-tags implementing simple ciphers and meeting the minimum security requirements as suggested in this paper. Designing a cipher is a difficult and an continuously evolving process to make it more efficient. We encourage through analysis and improvements of our DCSTaR protocol before being deployed.

4. Efficient and Privacy-Preserving mRFID/NFC Payment Model

Mobile payment is a payment method, where a mobile phone or a smartphone is used to pay for merchandize and services. Mobile payment is gaining popularity especially in Asia and Europe. The research firm Gartner Inc. predicts that the number of mobile payment users will reach more than 190 million in 2012 [16].

In this paper we focus on two emerging, promising and related technologies namely: the Radio Frequency Identification (RFID) and Near Field Communication (NFC). The 'Mobile NFC Payment' is one of the applications of NFC that is drawing a great deal of attention. Currently efforts are being put to deploy a mobile NFC Payment model that precisely mimics the Contactless (RFID) Card Payment model, where a NFC-enabled smartphone behaves as a contactless credit/debit card. But through this paper we emphasize that since credit/debit card payment transactions do not protect the customer's privacy and are also prone to card fraud and identity theft, the Mobile NFC Payment application must also have an 'alternate' payment model to support those customers who give high priority to privacy and want to be in complete control of their payments.

Previously proposed anonymous (untraceable) electronic cash models were not viable to be deployed as real world applications. However, in this paper, we take advantage of smartphones, RFID and NFC technologies to resuscitate anonymous (untraceable) electronic cash model and propose a simple, efficient, and privacy-preserving mobile payment model, which could be an alternative to the Mobile NFC Payment.

4.1 Motivation and Related Work

4.1.1 Drawbacks of Credit/Debit Card Payments

In recent years the number of credit/debit card payment transactions have substantially increased. The credit/debit cards offer great convenience to customers, eliminating the need to carry cash (banknotes and coins) for most of the payments and the payments are accomplished much faster. The credit cards also allow customers to obtain instant loans (based on their credit limit), which they can repay at a later time. However, it is well-known that the credit/debit card payment transactions have the following critical drawbacks.

Privacy Violation and Card Fraud

Credit/debit card payment transactions do not protect the customer's privacy. The customer's card details, the payment amount, and when and where the payment was made, are of course exposed to the merchant, but also to the merchant's bank (acquirer), the card companies (*e.g.,* Visa, MasterCard), the customer's bank (card issuer), and multiple intermediate third-party payment processor companies (Independent Sales Organizations), and internet payment gateway companies. Exposing the customer's card details to so many entities leads to serious card frauds, skimming, identity theft and customer profiling [65].

Many times the systems at the intermediate third-party payment processor/gateway companies are breached and a huge number of credit/debit card details are exposed [66]. In the year 2009, the payment processor 'Heartland Payment Systems Inc.' disclosed that it became a victim of a massive data breach [68], details of more than 130 million credit/debit cards were believed to be stolen making it the biggest payment card breach to date. Similarly some of the stores that belong to 7-Eleven, Hannaford Brothers, and TJX Companies Inc., have also been breached by hackers.

Customer Not in Control of Payment

Most of the customers are not confident, instead extremely cautious, while using their credit/debit cards, because once they handover the card to the cashier, they are "no longer in control" on how their card details and money are handled, the card security is potentially compromised from here on. Since customers use their cards extensively, they always fear becoming victims of a card fraud and/or an identity theft. Customers can identify irregularities only after the fraud has been committed, and that too by thoroughly verifying their monthly credit/debit card statements.

Temptation to Overspend & Bankruptcy

Especially with credit cards, customers are quickly tempted to spend more than they can afford. The credit card debt with high interest rate can lead to financial crisis and bankruptcy [60].

Expensive 'Per Transaction' Fees for Retailers & Inflated Pricing for Customers

The card companies charge an expensive per transaction 'interchange fee' [56] from the merchants and the intermediate third-party payment processor/gateway companies also charge several other fees. To compensate for these transaction fees, merchants may charge customers extra for card payments or inflate the prices of their merchandize, effecting even those customers who do not use credit/debit cards for payments [67]. On the other hand merchants actually prefer cash payments in order to reduce their 'per transaction' credit/debit card processing fees.

4.1.2 Drawbacks of Contactless Credit/Debit Card & Mobile NFC Payment

The card companies, MasterCard and Visa have introduced contactless (RFID) credit/debit cards, PayPass [61] and payWave [69] respectively. These cards are based on the standard for Radio Frequency (RF) cards - ISO 14443 type:A/B [59]. MasterCard and Visa have also integrated their contactless credit/debit card payment model into NFC-enabled mobile phone as a mobile NFC payment model [62] and [70] and are now conducting trails.

Customers instead of handing their card to a cashier, bring their contactless card or NFC-enabled mobile phone within one-two inches of a reader at point-of-sale. The stored card details in the mobile phone are sent to the reader, and the rest of the payment transaction procedure is the same as normal credit/debit card payments. Therefore, even though the contactless card or NFC-enabled mobile phone is in the possession of the customer, the above mentioned credit/debit card payment drawbacks are not alleviated. For the card companies, contactless RIFD cards or mobile NFC payments are intended for getting more faster credit transactions from customers.

On the other hand, these contactless cards can be scanned from a distance without the knowledge of the customer. Heydt-Benjamin et al., [57] have shown various vulnerabilities in several contactless RFID credit cards. [Verbatim from [57]] Their study observed that the cardholder's name and often credit card number and expiration are leaked in plaintext to unauthenticated readers, some cards may be skimmed once and replayed at will, and they are susceptible in various degrees to a range of other traditional RFID attacks such as skimming and relaying.

4.1.3 Drawbacks of Prepaid Contactless Card Payment

We use prepaid contactless cards for micro-payments at the subway, bus, vending machines, *etc.*, but such cards do have a unique ID and the customer can be tracked and traced based on this unique ID. The customer can protect his/her privacy by frequently canceling these cards and re-issuing new ones. But canceling and re-issuing these cards require the customer to personally visit the authorized entity and it often involves non-refundable registration fees.

4.1.4 Drawbacks of Anonymous Electronic Cash Payment

Anonymous Electronic Cash Payment models [52], [54], [55] are based on (partially) blind-signature schemes and they adopt coin/token-based approach. These payment models are a bit complicated, where the customer has to withdraw a big coin from the bank, divide the big coin to smaller coins, pay the merchant with the smaller coins, the merchant submits the coins to the bank, the bank verifies the coins for illegal double spending and if it detects double spending it should trace the customer who committed the fraud. The bank also have to verify if the merchant has not re-submitted already cleared coins, the bank has a huge burden. These payment models have to be implemented and deployed precisely to prevent fraud, therefore they haven't been viable for large-scale deployment.

4.2 Goals: Offering a Choice - Alternate Payment Model

We do not want to replace credit/debit card payments. There will be, services/merchandize sold solely based on these cards, and customers and merchants who love the convenience and benefits got from these cards. But due to the above mentioned drawbacks, we also believe that there is a large number of customers and merchants very reluctantly using and accepting credit/debit card payments as they don't have a choice, they can of course deal in cash payments, but it is so inconvenient and unsafe to carry cash at all times. Therefore, in this paper we offer both customers and merchants another choice to choose from - an alternate payment model that satisfies the following goals:

- Neither a 'credit/debit card' nor a 'contactless card' payment model
- Simple (is the best), efficient, faster, convenient, and secure payment model
- Protect customer privacy from the banks and merchants
- Provide customers complete control on their payment transactions

- Prevent customers from waiting in long check-out lines
- Reduce transaction fees for merchants and provide instant payment

4.3 The Big Picture of the Proposed Mobile Payment Model

We propose a "Pre-Paid Mobile HTTPS-based Payment model". For ease of describing our idea, let us consider one particular customer called Alice. Let us also assume that: both Alice and the merchant have a bank account in the same bank, the items (in the store) chosen by Alice are all tagged with RFID tags, and Alice's smartphone is NFC-enabled. We consider 3 entities: (i) Customer Alice with NFC-enabled smartphone (NSP), (ii) Bank, and (iii) Store. Our proposed payment model involves four procedures as described below:

Anonymous Pre-Paid Digital Cash Certificate Issuing Procedure

Alice can use her computer or her smartphone's 3G/4G network to establish a HTTPS (Hypertext Transfer Protocol Secure) connection [58] with the bank and request for a digital cash certificate of a certain amount. The bank deducts the amount from Alice's account and returns a digitally signed cash certificate. This pre-paid cash certificate is anonymous, i.e., it does not contain any details of Alice's true identity, and the bank itself cannot link this certificate to Alice at a later stage. Alice stores this *Anonymous Pre-Paid Digital Cash Certificate* in her smartphone.

Obtaining Digital Invoice Certificate Procedure

Alice visits a department store and chooses some items. She approaches one of the several RFID-kiosks in the store. The kiosk instantaneously scans the RFID-tagged items in the shopping cart and generates a digital *Invoice Certificate.* This *invoice certificate* can contain several details such as store's name and address, list of items chosen and their prices, *etc.*, but most importantly it contains: a unique *invoice ID*, the *invoice amount* and the *merchant's bank account details* (account name, and account number). Alice must deposit the *invoice amount* into the merchant's bank account in order to complete the payment. Alice's NFC-enabled smartphone, which is brought closer to the kiosk's NFC module, can download this *invoice certificate.*

Alice can now walk away from the kiosk, but cannot leave the store, since her chosen items haven't been flagged as 'sold' and they would trigger the alarm at the store's exit.

However, Alice can complete the remaining payment procedure at her own comfort anywhere within the store, *e.g.,* least crowded area or while sitting at the store's food court, thus preventing customers from waiting in long check-out lines.

The Payment Procedure

Alice uses her smartphone's 3G/4G network to establish a HTTPS connection with the bank and submits a digital *Cheque Certificate*. This *Cheque* contains: Alice's *Anonymous Pre-Paid Digital Cash Certificate*, the *invoice ID*, the *invoice amount* and the *merchant's bank account details* (account name, and account number). As mentioned above, the bank cannot link the *Anonymous Pre-Paid Digital Cash Certificate* to Alice, but to prevent unauthorized use of this cash certificate, the *Cheque Certificate* provides an 'anonymous proof' to the bank, that the owner of the cash certificate is indeed involved in this payment procedure.

If this is the first time that bank is receiving this *Anonymous Pre-Paid Digital Cash Certificate*, it registers this cash certificate into its database, along with a *certificate balance* parameter. Initially, the *certificate balance* value is equal to the amount value of the cash certificate. The bank deducts the *invoice amount* from the *certificate balance* value and deposits the *invoice amount* into the merchant's bank account. The bank then updates the *certificate balance* value in its database. From here on, whenever the same *Anonymous Pre-Paid Digital Cash Certificate* is received for other payments, the bank first checks the *certificate balance* in its database and then proceeds with the deposit, else it would respond 'insufficient cash'.

The bank sends a digital *Invoice Paid Receipt* to the merchant. This receipt confirms that the *invoice ID* has been paid. The merchant flags the items listed under the *invoice ID* as 'sold' and also acknowledges this to the bank. The bank can now send another *Invoice Paid Receipt* to Alice, confirming that the *invoice ID* payment has been successful. Alice can now leave the store with her purchased items.

Reclaiming Unspent Amount Procedure

Alice can choose to cancel her *Anonymous Pre-Paid Digital Cash Certificate* whenever she wants or when the smartphone alerts Alice, that her cash certificate is soon expiring and it has some unspent balance amount. Alice has two options to cancel her certificate and reclaim the unspent amount.

(1) Alice connects her smartphone to the bank's ATM and provides an 'anonymous proof' that she is indeed the owner of the cash certificate. The bank refunds the unspent

amount via the ATM's cash dispenser and cancels the cash certificate.

(2) Let us assume that Alice obtains another new cash certificate. Alice can use her computer or her smartphone's 3G/4G network to establish a HTTPS connection with the bank and provide an 'anonymous proof' that she is indeed the owner of both the cash certificate that is to be canceled and the new cash certificate. The bank updates the database by adding the unspent amount value on the to-be-canceled cash certificate to the balance amount value of the new cash certificate and cancels the to-be-canceled cash certificate.

4.3.1 Benefits/Economic Motives for the Entities

Customer Alice in complete control knows the merchant's bank account and the invoice amount. Since the cash certificate is a digital form of physical cash, she keeps a check on her payments and resists overspending. Unlike credit cards there are no interest fees in this model. Customer can easily cancel and request new cash certificates anytime. Customer's privacy is protected from both the merchant and the bank.

Bank charges merchant a fee. Our proposed model can be provided by any bank, unlike the credit/debit cards that are monopolized by few card companies. Merchants pay a small fee to the bank; compared to expensive fees to multiple parties associated with card transactions; thus avoid inflating commodity prices. Customers needn't wait in checkout lines. Mobile Operator charges customers for using 3G/4G Internet.

4.3.2 Requirements

- Partially Blind Signature [51] needed for customer privacy protection: *Anonymous Pre-Paid Digital Cash Certificate* and 'Anonymous Proof'. We assume that the NFC-enabled smartphone and the bank are capable of executing partially blind signature procedure. Abe *et al.,* [51] proposed the idea of partially blind signature with security proofs. A partially blind signature scheme is an extension of an ordinary blind signature scheme [55]. It has two portions, one portion consists of the message that is blinded by the user from the signer and in the other portion, the signer can explicitly embed some mutually agreed information such as amount, issuing date, and expiry date, *etc.* In this paper we implemented the randomized RSA-based partially blind signature scheme proposed by Cao *et al.,* [53].

- Digital Signature Algorithm (DSA) [64] needed for entity authentication and data integrity

- HTTPS (Hypertext Transfer Protocol Secure) [58] connection needed for bank (server) authentication and mobile phone (client)-bank (server) data confidentiality

4.4 Technical Details of the Proposed Mobile Payment Model

We consider 3 entities: (i) Customer Alice with NFC-enabled smartphone (NSP), (ii) Bank, and (iii) Store. Table 4.1 provides the list of notations we used in this paper. The communication channel between the entities is secured via the standard HTTPS protocol. The four procedures are well depicted in the Figures 4.1 to 4.4. Due to the space constraint, we describe the important steps of these phases below, skipping some of the trivial ones.

Table 4.1: Notations for mRFID/NFC Mobile Payment Model

NOTATION	DESCRIPTION
A	Customer Alice's NFC smart phone (NSP)
B	Bank
S	Store / Service Provider (S)
X	An Entity: A, B, S
idX	Identity of X
eX	Public-Key of X
dX	Secret-Key of X
$drtX$	Digital Certificate of X
$sigX\{\}$	Digital Signature using dX
$expC$	Certificate Expiry Date
$amtC$	Cash Value of Cash Certificate
$crtC$	Alice's Cash Certificate
$balC$	Available balance amount on $crtC$
$date, time$	Date & Time of generating certificate
$acctS$	Bank Account details of S
$amtV$	Total Invoice Amount
idV	Unique ID of the Invoice
$crtV$	Digital Invoice Certificate
$crtQ$	Digital Cheque Certificate
$crtR$	Digital Paid Receipt Certificate

4.4.1 Anonymous Pre-Paid Digital Cash Certificate Issuing Procedure

Fig.4.1: 1.0 to 3.2:

The eA is just a public-key and not a certificate-authority issued digital (public-key)

Alice (A) NSP / NSP$\leftrightsquigarrow$PC/ATM	Secure Channel SSL/TLS	Bank (B) / ATM
1.0. Generate: eA, dA		
1.1. $blind(eA) = m$		
1.2. Choose: $amtC, expC$		
	2. $\xrightarrow{idA/pwdA, amtC, expC, m}$	
		3.0. Verify: $idA/pwdA$ is customer Alice
		3.1. Check: $amtC$ is deposited & $expC$ is valid
		3.2. $blindsign(m, amtC, expC)$ $= sigB\{m, amtC, expC\}$
	4. $\xleftarrow{sigB\{m, amtC, expC\}}$	
5.0. Verify: $sigB\{m, amtC, expC\}$ using $drtB$		
5.1. $unblind(sigB\{m, amtC, expC\})$ $sigB\{eA, amtC, expC\} = crtC$		
5.2. $amtC = balC_A$		
5.3. Create data: $[eA : dA, crtC, balC_A]$		

Figure 4.1: Anonymous Pre-Paid Digital Cash Certificate *crtC* Issuing Procedure

certificate (containing a public-key and also its owner's identity) and it is blinded as m. Alice uses her NSP's 3G/4G network to connect to her bank's Internet banking facility and authenticates herself with ID/Pwd. Alice types in her desired *amtC* value of the certificate and its expiry duration (*expC*), *e.g.,* number of days. The bank deducts the *amtC* from Alice's account. It verifies if the *expC* value satisfy the bank rules. By utilizing the partially blind signature scheme, the bank embeds *amtC* and *expC* values while signing m, to generate:
$sigB\{m, amtC, expC\}$.

Fig.4.1: 5.1 to 5.3: The NSP un-blinds m to reveal the eA in the *cash certificate*: $crtC = sigB\{eA, amtC, expC\}$. The $balC_A$ indicates the up-to-date balance amount on the *crtC* after every payment transaction, therefore it is initially assigned the value of *amtC*. The *eA* becomes the pseudo-ID of the *crtC*, therefore the NSP creates a data table in its memory as $[eA : dA, crtC, balC_A]$.

Other Options: Anonymous Pre-Paid Digital Cash Certificate can be obtained using the smartphone, but using an ATM or PC will reduce communication and computational burden on the phone. Alice can plug the NSP to her PC and let the PC execute this phase and transfer *eA*, *dA*, *crtC* and *drtB* to the NSP. Alice can also connect her NSP to the bank's ATM. The NSP sends m and Alice enters *amtC* and *expC* into the ATM, which then returns $sigB\{m, amtC, expC\}$ and *drtB* to the NSP. If Alice doesn't have a bank account, she can still connect her NSP to the ATM, but Alice must deposit the cash of *amtC* into the ATM.

4.4.2 Obtaining Digital Invoice Certificate Procedure

Alice chooses some tagged items at a department store, and approaches one of the several RFID-kiosks in the store. The kiosk instantaneously scans the tagged items in the shopping cart and generates a digital *invoice certificate* $crtV = sigS\{idV, amtV, acctS\}$. Alice must deposit *amtV* into the *acctS* to complete the payment. The *crtV* can contain other details such as store's name and address, list of items chosen and their prices, *etc.* The NSP is brought closer to the kiosk's NFC module, and can thus download the *crtV* and the store's *drtS*, which is used to verify the signature on *crtV*. The NSP adds a data record in its memory as $[idV : drtS, crtV]$.

Alice can now walk away from the kiosk, but cannot leave the store, since her chosen items haven't been flagged as 'sold' and they would trigger the alarm at the store's exit. However, Alice can complete the remaining payment procedure at her own comfort anywhere within the store, *e.g.,* least crowded area or while sitting at the store's food court, reducing and even preventing customers waiting at long check-out lines.

4.4.3 The Payment Procedure

Fig. 4.2: 1.0 to 2: To complete the remaining payment procedure, the NSP generates a digital *cheque* ($crtQ$), authorizing the bank to deduct the $amtV$ from the $crtC$ and deposit the $amtV$ into the $acctS$. The NSP connects to the bank via the 3G/4G network, sending 'only' the $crtQ$.

Fig. 4.2: 3.0 to 3.2: Since the bank issued the $crtC$, it verifies its signature on the $crtC$. If the $crtC$ is being used for the first time, then the bank adds a data record with eA as the primary reference and the $balC_B$ initially assigned the value of $amtC$ specified in the $crtC$. From here on, whenever the $crtC$ is received for other payments, the bank first checks and then updates this data $[eA : crtC, balC_B]$.

To protect privacy, this procedure 'does not' require Alice to authenticate herself to the bank and also the eA was blinded from the bank during the $crtC$ issuing procedure, therefore the bank cannot link this $crtC$ to Alice, but to prevent unauthorized use of the $crtC$, the bank needs an 'anonymous proof' that the owner of the $crtC$ is indeed involved in this payment procedure. Therefore, the eA included in the $crtC$ must verify the signature on the $crtQ$, proving to the bank that a owner possessing the dA has signed the $crtQ$.

If the $amtV > balC_B$, the bank responds "insufficient balance on the $crtC$" and ends the payment procedure. Its not shown in the Fig. 4.2, but the bank sends a digital *paid receipt*:
$crtR_S = sigB\{idVpaid, amtV, acctS, dateR_S, timeR_S\}$ to the store confirming the payment for idV. The store flags the items listed under idV as 'sold' and sends an acknowledgement to the bank. Now the bank sends a digital *paid receipt* $crtR_A$ to the NSP, proving the successful completion of the payment. The bank updates its data with the new $balC_B$ value and also adds the $crtQ$ and $crtR_A$ as a proof of this payment.

Fig. 4.2: 5.1 to 5.3: The NSP calculates $balC_A - amtV = balC_A$ and updates its data with the new $balC_A$ value and also adds the $crtR_A$ as the proof of this payment. Alice can now leave the store with her purchased items.

4.4.4 Reclaiming Unspent Amount Procedure

The NSP alerts Alice that her $crtC$ is soon expiring and it has some unspent balance amount. Alice has two options to reclaim this amount. Again, to protect privacy, this procedure 'does not' require Alice to authenticate herself to the bank.

Alice (A) $\longleftarrow$ Secure Channel $\longrightarrow$ Bank (B)

NSP SSL/TLS

1.0. Retrieve data:

$[eA : dA, crtC, balC_A]$ and $[idV : drtS, crtV]$

1.1. Generate:

$sigA\{idV, amtV, acctS, dateQ, timeQ, crtC\} = \underline{crtQ}$

1.2. Update data: $[idV : drtS, crtV, \underline{crtQ}]$

$$\xrightarrow{\quad 2.\ crtQ \quad}$$

3.0. Verify: $crtC$ using eB & $expC$ not expired

3.1. If: $crtC$ is being used for the first time

Create data: $[eA : crtC, balC_B]$ where $amtC = balC_B$

3.2. Else:

Retrieve already created data: $[eA : crtC, balC_B]$

Verify: $\underline{crtQ}$ using eA in $crtC$ & $dateQ, timeQ$

If $amtV < balC_B$: deposit $amtV$ into $acctS$

$balC_B - amtV = \underline{balC_B}$

$sigB\{idVpaid, dateR_A, timeR_A, balC_B\} = \underline{crtR_A}$

Update data: $[eA : crtC, \underline{balC_B}, \underline{crtQ}, \underline{crtR_A}]$

$$\xleftarrow{\quad 4.\ crtR_A \quad}$$

5.0. Verify: $\underline{crtR_A}$ using $drtB$ & $dateR_A, timeR_A$

5.1. $balC_A - amtV = \underline{balC_A}$

5.2. Update data $[eA : dA, crtC, \underline{balC_A}]$

5.3. Update data $[idV : crtV, crtQ, \underline{crtR_A}]$

Figure 4.2: The Payment Procedure

Option 1 (Fig.4.3): Alice connects her NSP to the bank's ATM and like in the payment procedure, the NSP anonymously proves that she is the owner of the $crtC$. The bank refunds the unspent amount via the ATM's cash dispenser and cancels $crtC$.

Option 2 (Fig.4.4): Alice can obtain another cash certificate $crtC'$. Later, Alice can either use her NSP, NSP$\leftrightsquigarrow$PC to connect to the bank, or NSP$\leftrightsquigarrow$ATM and anonymously prove that she is the owner of both the $crtC$ and $crtC'$. The bank/ATM then adds the unspent amount value on $crtC$ to the balance amount value of the new $crtC'$ and cancels $crtC$.

Alice (A)	$\longleftarrow$ Secure Channel $\longrightarrow$	Bank (B) / ATM
NSP$\leftrightsquigarrow$ATM	SSL/TLS	

1.0. Retrieve data: $[eA : dA, crtC, balC_A]$

1.1. $sigA\{dateP, timeP, crtC\} = P$

$$\xrightarrow{\quad 2.\ P \quad}$$

3.0. Verify: $crtC$ using eB & $expC$ not expired

3.1. Retrieve data: $[eA : crtC, balC_B]$

3.2. Verify: P using eA in $crtC$ & $dateP, timeP$

3.3. Refund $balC_B$ via ATM's cash dispenser

3.4. Update data: $[eA : crtC, balC_B, cancelled]$

Figure 4.3: Reclaiming Unspent Amount Procedure: Option 1

4.5 Analysis

Our proposed mobile payment model utilizes Partially Blind Signature Scheme for customer privacy: Anonymous Pre-Paid Digital Cash Certificate and Anonymous Proof. Digital Signature Algorithm for entity authentication and data integrity. HTTPS (Hypertext Transfer Protocol Secure) communication for entity authentication and data confidentiality. Therefore, our solution can easily adhere to and deployable (as smartphone application) based on the secure "Electronic Data Interchange (EDI) via the Internet" [63] model. Currently the Internet payment transactions are carried via HTTPS and EDI model.

Alice (A) NSP / NSP⇜⇝PC/ATM	$\longleftarrow$ Secure Channel $\longrightarrow$ SSL/TLS	Bank (B) / ATM

1.0. Retrieve data: $[eA : dA, crtC, balC_A]$

1.1. Retrieve data: $[eA' : dA', crtC', balC'_A]$

1.2. $sigA'\{dateP', timeP', crtC'\} = P'$

1.3. $sigA\{dateP, timeP, crtC, P'\} = P$

$$\xrightarrow{\text{2. } P}$$

5.0. Verify: $crtC, crtC'$ using eB & $expC, expC'$ not expired

5.1. Retrieve data: $[eA : crtC, balC_B]$ and $[eA' : crtC', balC'_B]$

5.2. Verify: P using eA in $crtC$ & $dateP, timeP$

5.3. Verify: P' using eA' in $crtC'$ & $dateP', timeP'$

5.4. $balC_B + balC'_B = \underline{balC'_B}$

5.5. Update data: $[eA : crtC, balC_B, cancelled]$

5.6. Update data: $[eA' : crtC', \underline{balC'_B}]$

Figure 4.4: Reclaiming (Unspent Money) Phase: Option 2

4.5.1 Customer Privacy Protection

In our proposed model the customer is in-charge of the payment and the bank pays the store, therefore the customer remains anonymous to the store at all times. Our scheme provides restricted customer privacy when dealing with the bank. The bank receives customer's public-key value eA, therefore the customer can still be tracked with his/her eA usage, until the customer's cash certificate $crtC$ is either expired or canceled. But the real identity of the customer is never revealed, because eA acts as a pseudonym for the customer. Also, there is no match between the real ID of the customer and his/her eA, because during the cash certificate issuing phase the eA is blinded, and the bank knows the value of eA only during the payment phase. Our proposed model allows customer to easily cancel and request new cash certificates using smartphone periodically, further protecting his/her privacy.

4.5.2 Customer Public Key Collisions

In our proposed payment model we allow the customers to generate their own public-key pair (eA, dA), therefore the question of another customer having the same public-key may arise, thus causing a public-key collision. Such a rare scenario could be prevented by the bank strictly accepting only well-proven public-key algorithms like the RSA algorithm with a larger key size *e.g.,* 1024 bits. Public key is always derived from a private key and not vice-versa and the private-key of a particular customer is never exposed in our model, therefore an adversary cannot generate the same public-key as Alice. Lastly, during every transaction between the customer and the bank, the customer has to make use of his private-key to anonymously prove that the public-key in the pre-paid cash certificate is indeed generated from the private-key he/she possesses.

4.5.3 Man-in-the-Middle Attack

Our proposed mobile payment model utilizes HTTPS as in the case of EDI model, therefore communication channel is well secured from man-in-the-middle attacks. Even if the phone's network is lost, but once the network's re-established, customer will use the unique invoice ID and cash certificate to request the payment status or the paid receipt from the bank. Since we use digital signatures and HTTPS communication, all transactions are atomic and can be easily verified.

4.5.4 Prevent Replay Attack

The current date and time are included in every signature to detect re-played messages.

4.5.5 Prevent Double Spending

In our payment architecture the bank keeps track of eA and it's corresponding $balC_B$ value to prevent any double spending.

4.5.6 Non-Repudiation

In our payment architecture, non-repudiation is satisfied because both the customer and the merchant trust the bank. The digital receipts, generated by the bank prove that the transaction between the customer and merchant has been successful. Also the digital Cheque and invoice are proofs of the transaction.

4.5.7 Least Amount of Overhead

Our scheme poses little overhead both on the NSP and the bank. It can be noticed that the NSP has to just pass on the $crtQ$ and authenticate itself to the bank. The bank takes over the task of paying the merchant. The bank needs to keep track of just the eA and update it's corresponding $balC_B$ value. The cash certificate's expiry date prevents bank from keeping track of cash certificates for an infinitely long time. Expired certificate cannot be used for payments but can be submitted (grace period) for reclaiming unspent money. The merchant receives the payment immediately.

We assumed that the customer and the merchant have bank accounts at the same bank. But this assumption is purely for clarity and ease of explaining our model. Merchants can have accounts in several banks, and the RFID-kiosk can offer the customers a list of banks to choose from, so that the customer can pick a particular bank which has issued his/her Anonymous Pre-Paid Digital Cash Certificate. As a result the RFID-kiosk can generate the invoice certificate containing the merchant's account details at the bank chosen by the customer.

Limitation: On the other hand, if the merchant has only one account at a bank that is different from the customer's bank, then the customer's bank has to communicate with the merchant's bank and wire transfer the invoice amount. This would delay our payment model's processing time as the customer's bank have to await the confirmation from the merchant's bank.

4.5.8 Against Stolen Smartphone & Money Laundering

Our proposed model can be implemented as a smartphone application that is password protected (including the keys), this prevents un-authorized usage if the smartphone is stolen. The customer can keep a copy of the cash certificates in his/her computer and in case the phone is lost or broken, he /she can cancel the lost certificates and reclaim the money using the stored cash certificates in the computer. Our model also keeps a check on money laundering; the bank is always involved in transactions. Though the bank cannot identify the customer, it knows the payee's (store) identity and the amount of money being deposited.

4.6 Proof-of-Concept Implementation

Environment

Client: Customer Alice

- Android SDK 2.1 platform using the emulator: Android Eclair - SDK 2.1
- Java language on the eclipse IDE environment with java.security package built in on JDK 1.6.
- Ksoap2 for SOAP to contact with web server
- Database Management: SQLite

Server: Bank, Store

- PHP based web service (PHP/5.2.9)
- Nu-Soap for building web service and SOAP communication
- Server Specification: Processor : Intel Core 2 Duo E6750 @2.66GHz, Memory : 2GB, Microsoft Windows XP Pro, Web server : Apache HTTPD 2.2.11
- Database Management: MySQL 5.1.33

Secure communication b/w client and server

- HTTPS: OPENSSL for SSL connection
(OpenSSL/0.9.8i)

Three Procedures

- Cash Certificate (crtC) issuing procedure (Fig.4.5, Fig.4.6))
- Obtaining Invoice Certificate (crtV) procedure (Fig.4.7)
- Payment procedure (Fig.4.8)

Security Modules

- RSA Key Generator 1024 bits length of key pair
- Partially blind signature scheme
- RSA based digital signature

Results

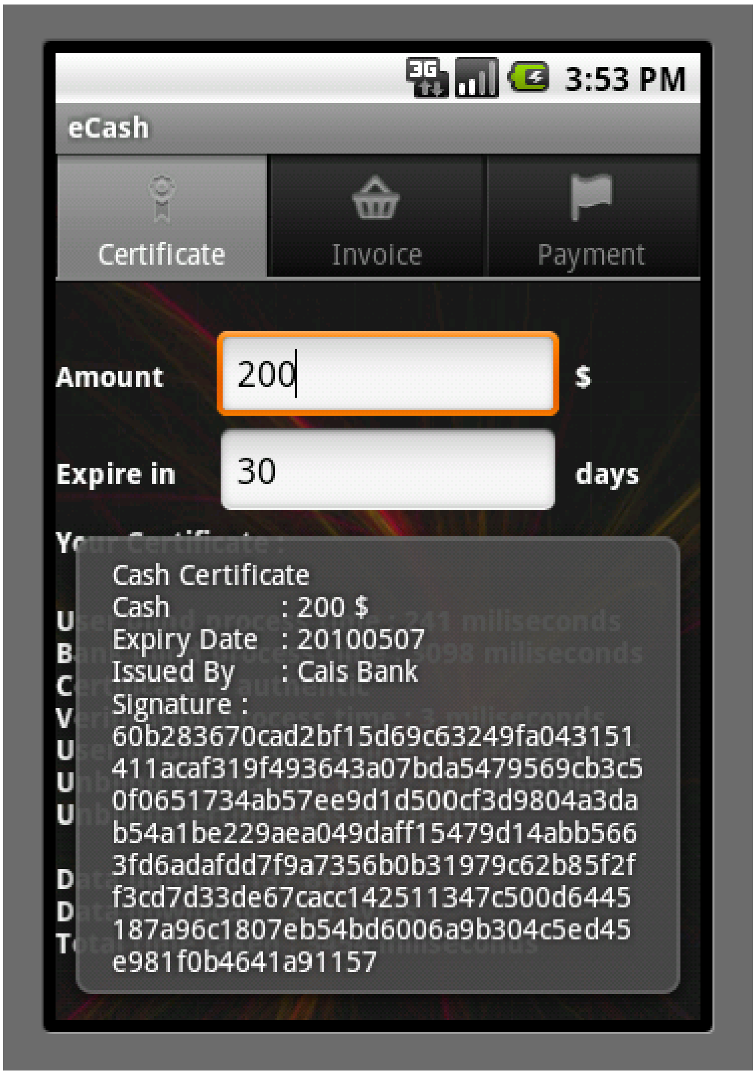
Fig.4.5 shows the Anonymous Pre-Paid Digital Cash Certificate with the amount value, expiry date, bank name, and all the above details are included in the cash certificate's digital signature generated by the bank.

Fig.4.6 shows the behind the scene processing of Anonymous Pre-Paid Digital Cash Certificate Issuing Procedure. It displays the time taken by the customer's NSP to blind eA and establish a HTTPS connection with the bank. It displays the time taken by the bank to generate the signed Anonymous Pre-Paid Digital Cash Certificate, time taken by the customer to verify this certificate, un-blind the certificate, and verify the un-blinded certificate. It also displays the NSP's upload and download data sizes.

Fig.4.7 shows the digital *invoice certificate* generated by the merchant's RFID-kiosk, which is in turn downloaded by the NSP. It displays the unique *invoice ID*, the *invoice amount* and the *merchant's bank account details*, and all the above details are included in the invoice certificate's digital signature generated by the merchant's RFID-kiosk. It also displays the NSP's upload and download data sizes.

Fig.4.8 shows the time taken by the NSP to generate and send the digital *cheque* to the bank, and the total time it takes for the NSP to receive the digital *paid receipt* from the bank. It also displays the NSP's upload and download data sizes.

We tried the emulation test 20 times, over wired internet as a proof-of-concept. Through our implementation we calculated the time it takes to execute the (i) Cash Certificate (crtC) issuing procedure (Fig.4.5, Fig.4.6), (ii) Obtaining Invoice Certificate (crtV) procedure (Fig.4.7) and (iii) Payment procedure (Fig.4.8). The results are shown in the Table.4.2, an average execution time: 3.6sec and a maximum pay load: 500 bytes, wouldn't

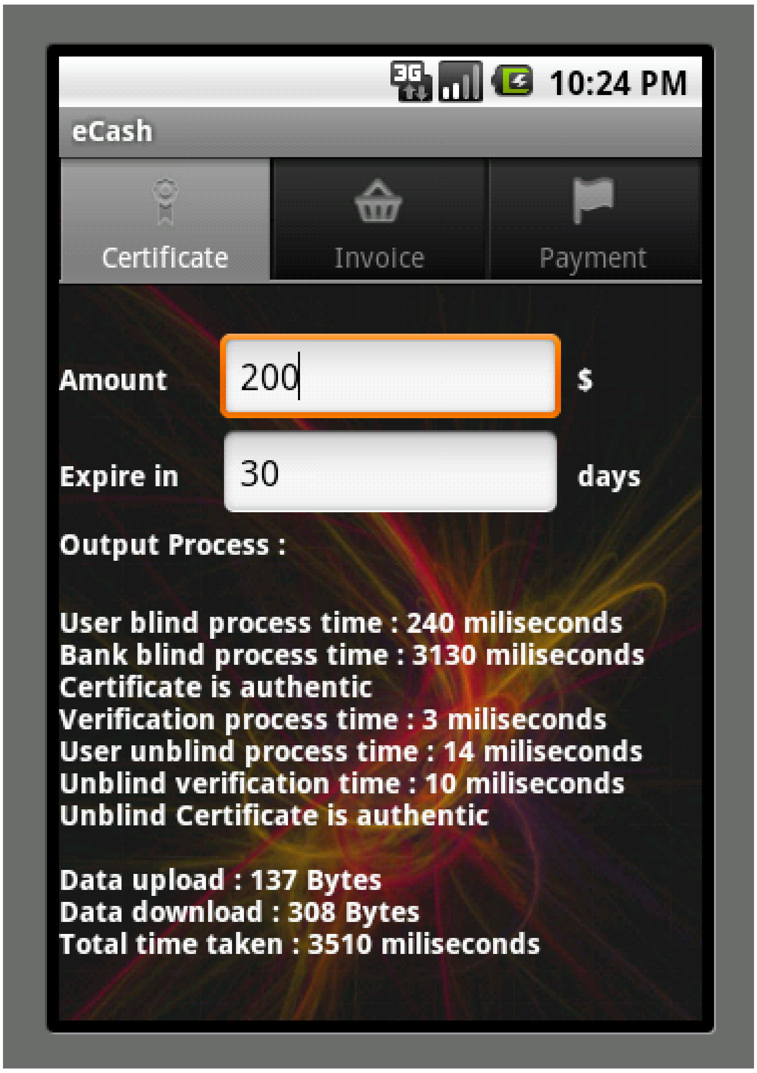


Amount 200 $

Expire in 30 days

```
Cash Certificate
Cash          : 200 $
Expiry Date   : 20100507
Issued By     : Cais Bank
Signature :
60b283670cad2bf15d69c63249fa043151
411acaf319f493643a07bda5479569cb3c5
0f0651734ab57ee9d1d500cf3d9804a3da
b54a1be229aea049daff15479d14abb566
3fd6adafdd7f9a7356b0b31979c62b85f2f
f3cd7d33de67cacc142511347c500d6445
187a96c1807eb54bd6006a9b304c5ed45
e981f0b4641a91157
```

Figure 4.5: Requesting and Obtaining Cash Certificate (*crtC*) from Bank



Amount 200 $

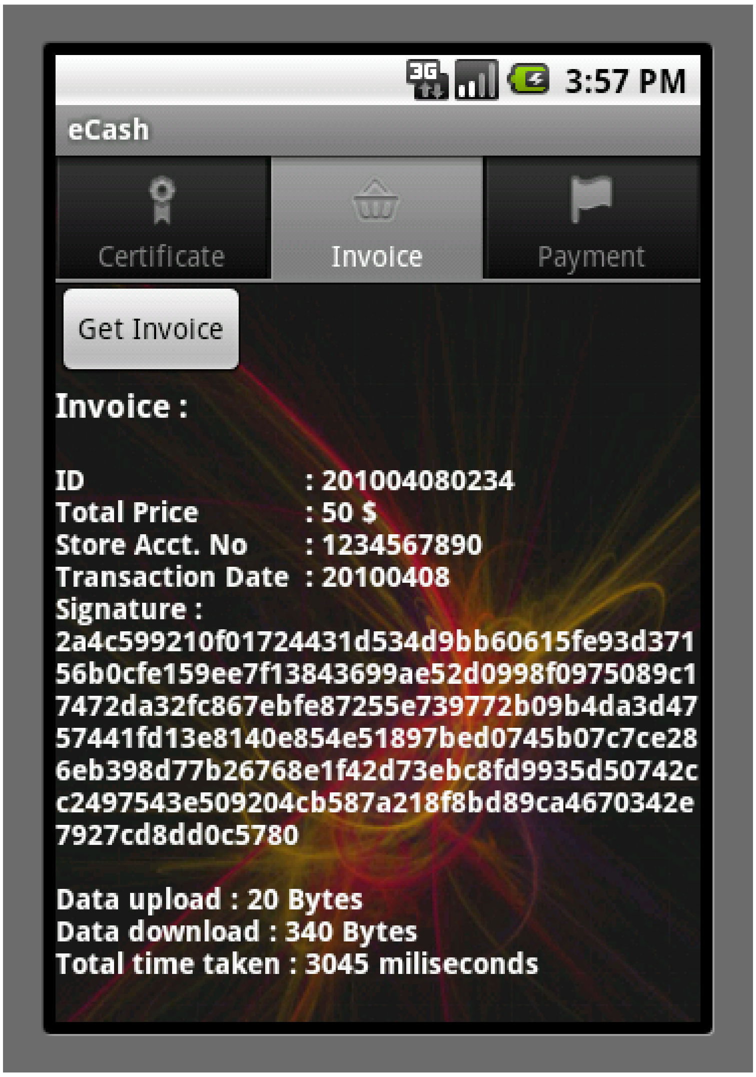
Expire in 30 days

Output Process :

User blind process time : 240 miliseconds
Bank blind process time : 3130 miliseconds
Certificate is authentic
Verification process time : 3 miliseconds
User unblind process time : 14 miliseconds
Unblind verification time : 10 miliseconds
Unblind Certificate is authentic

Data upload : 137 Bytes
Data download : 308 Bytes
Total time taken : 3510 miliseconds

Figure 4.6: Execution time - Cash Certificate (*crtC*)

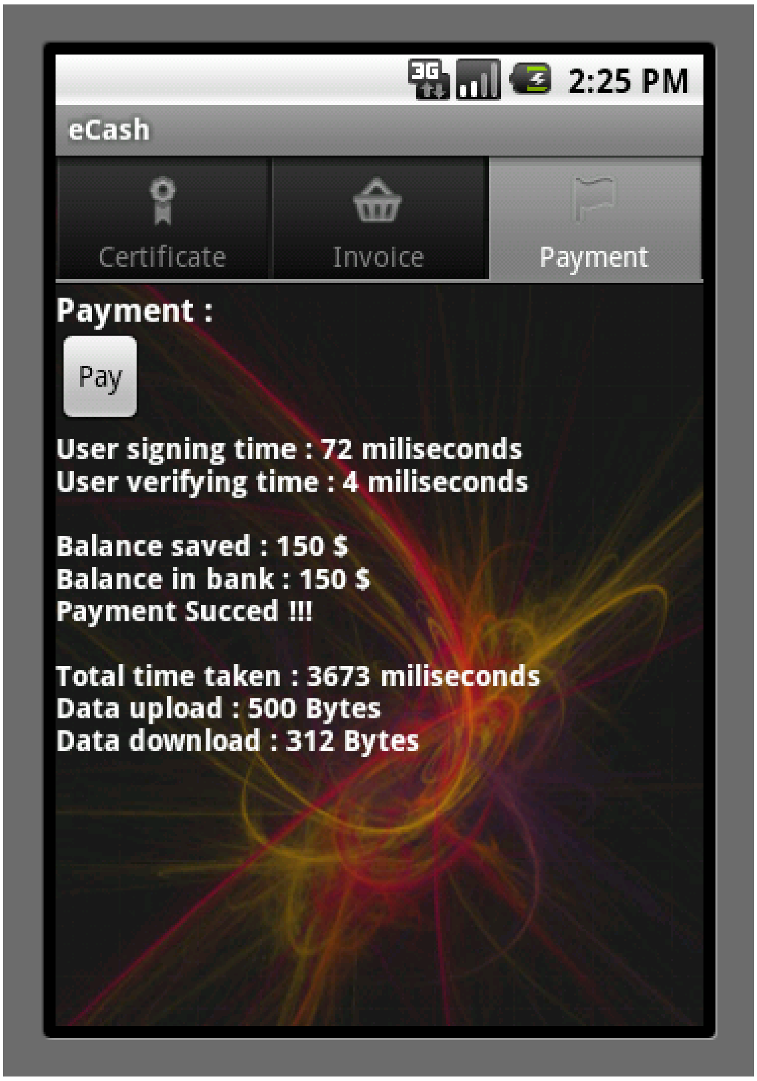


```
Invoice :

ID                : 201004080234
Total Price       : 50 $
Store Acct. No    : 1234567890
Transaction Date  : 20100408
Signature :
2a4c599210f01724431d534d9bb60615fe93d371
56b0cfe159ee7f13843699ae52d0998f0975089c1
7472da32fc867ebfe87255e739772b09b4da3d47
57441fd13e8140e854e51897bed0745b07c7ce28
6eb398d77b26768e1f42d73ebc8fd9935d50742c
c2497543e509204cb587a218f8bd89ca4670342e
7927cd8dd0c5780

Data upload : 20 Bytes
Data download : 340 Bytes
Total time taken : 3045 miliseconds
```

Figure 4.7: Obtaining Invoice Certificate (*crtV*) from Store



Payment :

User signing time : 72 miliseconds
User verifying time : 4 miliseconds

Balance saved : 150 $
Balance in bank : 150 $
Payment Succed !!!

Total time taken : 3673 miliseconds
Data upload : 500 Bytes
Data download : 312 Bytes

Figure 4.8: Pay and Receive Receipt Procedure

Table 4.2: Execution Time & Data Size

Procedure	Avg. Execution Time (ms)	Payload Sent (bytes)	Payload Received (bytes)
Certificate Issuing	3610	137	308
Invoice	3394	20	340
Payment	3601	312	500

be a burden on any 3G/4G smartphones and their battery life. The procedures take very little time but when deployed on a real commercial servers, they would be even faster.

4.7 Summary and Discussion

In this chapter we have proposed a privacy-preserving Pre-Paid Mobile HTTPS-based Payment model, which allows the customer to have complete control on his/her payments. Our proposed mobile payment model makes use emerging technologies like the smartphone, RFID and NFC. The proposed payment model provides the customer with complete control on his/her payments and privacy protection from both the bank and the merchant. The consumer can cancel and obtain new anonymous pre-paid cash certificates whenever and wherever he/she wants using the smartphone's 3G/4G/Wi-Fi networks. Our proof-of-concept implementation using Android SDK shows that our payment model can carried out by a smartphone within 4 seconds. Our future work would include a real practical implementation of our payment model using a NFC-enabled smartphone and developing a smartphone application that can communicate with the bank server via 3G/4G/Wi-Fi network and also communicate with a real NFC-module to download the merchant's bank account information and invoice. We would certainly focus on secure generation of customer's public and private key pairs in order to prevent public key collision with other customers.

5. mRFID↔Smart Home Security and Privacy

With Mobile-RFID (mRFID) technology, handheld portable devices like mobile/smartphones and PDAs, also behave as RFID readers and RFID tags. As RFID readers, mobile phones provide an user-friendly approach to quickly and efficiently scan, access and view information about RFID tagged items. As RFID tags, mobile phones can quickly identify themselves in order to communicate with other tagged devices, which provide essential services.

5.1 Contributions of this Chapter

The following are the contributions of this chapter:

- Introduce some of the practical example scenarios pertaining to mRFID applications in an public environment and RFID applications in smart home environment. Based on these example scenarios we derived their corresponding threats and needed security and privacy requirements.
- Propose a security framework that alleviates these derived threats. Our framework is just an outline of possible solutions to the security and privacy threats. The framework is built from a set of concepts that are linked to existing cryptographic methods and primitives.
- Our proposed framework is based on the following two standards: (i) EPCglobal Architecture Framework [11], (ii) EPCglobal C1G2 UHF RFID Protocol [12]. We assume that all the items are tagged with EPCglobal C1G2 UHF tags.

With the advent of RFID and mRFID technology we can find a public environment (*e.g.,* street, shopping mall), where Service Providers (SP) *e.g.*, shopping malls, department stores, cinema halls, and food courts, *etc.*, deploy RFID tagged items such as consumer goods, posters, sign boards, maps, and shopping catalogs, *etc.*, all around us. The intention of service providers is to provide services that are “related to” and “available at” customer’s current location. Therefore the coverage of this environment is very large, which includes all public places, roads, shopping malls, cinema halls, and food courts, *etc.* This will enable consumers to “Download and View Information represented by RFID tags”.

5.2 Download and View Information Represented by RFID Tags

Scenario I:*Alice visits a department store. She uses her mRFID-enabled mobile/smart phone to scan RFID tags attached to various items that are being sold. After scanning a particular RFID tag, the smartphone utilizes its 3G/4G/Wi-FI network to access store's "EPCIS", which contains a detailed database about the scanned RFID tag. As a result, the smartphone can download and store the price, picture, features, and manufacturer details of that item. Before purchasing an item, Alice would like to use her smartphone to verify if an item is genuine or a counterfeit.*

From the above scenario, we identified the following threats and security and privacy requirements:

5.2.1 Is that Tag Genuine or Counterfeit?: Tag Authentication

In a public environment, most of the RFID tags respond to every mRFID-enabled smartphone, otherwise the main purpose of these tags to provide "location-based instant information" would be defeated. Therefore, we do not consider a tag-reader mutual authentication and strong secure communication between RFID tag and smartphone. But there is one problem, these publicly available tags can be fake or must have been illegally modified (cloned) and no longer truly represent the information of the item in question. Also, before purchasing a product, Alice must be able use her smartphone to *"authenticate the tag"*, in order to prove beyond doubt that the tag attached to a product is indeed from the original manufacturer/SP of the product. As a result, we at least need a one-way authentication mechanism, which authenticates the RFID tag.

As mentioned in our proposed *DCSTaR: Diffusion-Confusion based Light-weight Security for RFID Tag-Reader Communication* protocol, Alice can use of her mRFID-enabled smartphone to Query and send R to the tag (as in Step 1-Fig.3.7). This smartphone obtains the EPC,S_1, S_2, and C from the tag, and send this data along with the R to the EPCIS via 3G/4G network or Wi-Fi connection. EPCIS would then verify C and replies to the device whether the item is genuine or fake. In here neither the tag's keys nor tag's sensitive data are exposed to Alice.

5.2.2 Can this EPCIS be trusted?: Secure Job Delegation & Trust Model

After scanning a tag, Alice's mRFID-enabled smartphone must be protected from being directed to, accessing, and downloading information, from malicious EPCIS, which can either induce virus code into to the smartphone or extract sensitive data off the smartphone. Also the true identity of Alice must never be revealed to the service provider, otherwise the service provider can generate detailed profile of Alice, her buying interests and know her current location. Alice should be close to the tag, in order to query it, and EPCIS already knows the location of that tag, this will implicitly give away the current location of Alice. Therefore, *"mRFID-enabled smartphone must anonymously communicate with only genuine service provider's EPCIS"*.

There would be many competitive service providers selling location-based services to users. Alice's mRFID-enabled smartphone may need to communicate with many service provider's EPCIS (Information Server). The smartphone should identify and authenticate genuine EPCIS and be able to secure the entire communication and also protect the Alice's privacy. But these tasks could create a huge burden on the smartphone and is certainly not user friendly. Therefore it would be lot easier for the smartphone to securely delegate its work to a trusted high-computing and resource-rich entity, such as a mobile operator. This approach helps in reducing the communication and computational burden on the smartphone. Therefore, establishing an efficient and a convincing trust model is very much required to ensure secure transactions, key distribution, and job delegation. With existence of a trust model, it would be lot easier for the smartphone to delegate its work to the mobile operator.

5.2.3 Security Framework for mRFID Applications in a Public Environment

Entities

The building blocks of mRFID infrastructure in a public environment zone could be similar to EPCglobal's RFID infrastructure, expect that we introduced mobile operator.

- Mobile RFID (mRFID): Mobile Phone with both RFID Reader and Tag functionalities, is used to scan tagged items available everywhere.
- RFID Tags: Every RFID tag contains its unique EPC number. EPC is a globally unique serial number that identifies an item in the supply chain. EPC data/number

contains: EPC Manager number (identifies the company), Object class (similar to a stock-keeping unit, also called product number), Serial number (specific instance of the object class being tagged, objects own unique identifier). EPCglobal allocates manufacturers specific blocks of EPC numbers, and manufacturers then add their own product codes and serial numbers to their assigned manufacturer numbers to create unique identifiers - EPCs.

Further information about the product is stored on a network of servers and databases called EPCIS/EPC Network. Therefore, unique EPC number acts like a pointer directing the RFID reader to the right entity on the EPC Network from where the reader can download additional related data about the product it scanned.

- Mobile Operator (MO): In the current mobile communications paradigm we have already put in a great deal of trust in MO, as it handles all our voice and data communications. It maintains a record of each subscriber's call details, contact information, and credit card details, *etc.* It even has the capability to easily determine our current location and tap into our communications. But what protects us from MO turning hostile is that it has to very strictly adhere to and follow legal, security and privacy policies imposed by the law. Our architecture extends this trust in MO to secure and provide privacy protection for mRFID transactions. This approach is very practical and easily deployable, as the current mobile communications infrastructure is widely spread and highly stable. MO takes responsibility on behalf of mRFID-enabled smartphone to select, identify, and authenticate genuine ECP-IS. MO behaving like a "Trusted Proxy" processes the request on behalf of the mRFID-enabled smartphone, greatly reducing the communication and computational burden on the user's smartphone and also provides users privacy protection. MO also takes responsibility on behalf of mRFID-enabled smartphone to select, identify, and authenticate only the genuine service providers and their EPS-IS.

- EPC Network: Just like the global look-up system such as the Domain Name Service (DNS), it becomes very necessary to look up each EPC number on a central data repository like we do with a Web page or other system using DNS. Keeping EPC data as an unique reference or primary ID, further information about the respective product is stored on databases and servers of EPC Network. This network assists local company staff and geographically distributed supply chain partners to easily and efficiently access information on any product they are handling from any location. The EPC Network [48] consists of three main components: Object Naming Service (ONS), the EPC-Information Services (EPCIS), and the EPC-Discovery Services (EPC-DS).

Brief Description of the Security Framework

- Step 1: The mRFID-enabled smartphone scans a RFID tag.
- Step 2: The RFID tag responds with EPC number.
- Step 3: The smartphone utilizes its 3G/4G/Wi-FI network to establish a HTTPS connection and authenticates itself to MO via login ID/pwd and sends the EPC number to MO.
- Step 4: The MO sends EPC number to the ONS.
- Step 5: The ONS responds with a URL of the EPCIS that related to the EPC number in question.
- Step 6: The MO sends the EPC number to the URL of EPCIS.
- Step 7: The EPCIS responds to the MO with related data about the EPC number in question via the same HTTPS connection.
- Step 8: The MO sends the EPC related information to the smartphone.

5.3 Carrying Tagged Consumer Items

Scenario II: *Alice visits a department store and purchases items that are having RFID tags attached to them. She wants to utilize the RFID tags attached to these purchased items in her smart home environment. But while carrying these items to her home, she might be snooped upon by a thief, Charlie, who has a powerful RFID reader, using which, from a distance he can scan the RFID tagged items inside Alice's bag, to check if she is carrying any items that are worth stealing. On the other hand, Alice may be carrying a RFID tagged MP3 player with her at all times and this tag has a unique EPC number. If Charlie happens to be a stalker, he can track and trace Alice at different locations based on this unique EPC number. Therefore consumers carrying RFID tagged items have to be protected from both Information and Location privacy violation.*

5.3.1 Protecting Consumer Privacy

Killing the Tag

As per the EPCglobal C1G2 UHF RFID Protocol standard [12], the manufacturer of the items can embed C1G2 UHF Tags with a Kill Password. Whenever a RFID reader sends this kill password to the tag, the tag is killed and rendered permanently unusable and unreadable. Therefore, once a tagged item is purchased by Alice, the trustable clerk at the point-of-sale (cashier) can obtain the tag's kill password from the store's EPCIS and kill the tag permanently. But with this approach Alice cannot make use of the tag capabilities at her smart home environment, e.g., RFID enabled refrigerator or book shelf.

Locking the Tag

As per the EPCglobal C1G2 UHF RFID Protocol standard [12], the manufacturer of the items can also embed C1G2 UHF Tags with a unique 32-bit value Access Password. A RFID reader submits the access password to the tag and the tag verifies if this access password is the same with the one embedded within itself. If the access passwords tally, the tag allows the reader to perform on it, the mandatory commands such as Read, Write, and Lock. A tag's chip has four memory banks: Reserved, EPC, TID, and User. The Reserved memory bank is used to store the kill password and access password. The reserved memory bank is permanently locked by the manufacturer; as a result the access password can neither be read nor modified by any reader.

As mentioned above, most of the tags contain only its unique EPC number and all the data associated with that EPC number is stored with the EPCIS. Access to EPCIS is secure, and restricted to only authorized supply chain stakeholders. Generally, the EPC memory bank is never locked, because the EPC number is used to retrieve the data associated with that item and also to retrieve its corresponding access password (from EPCIS). The tag's access password is thus used for "reader to tag" authentication and also allows the reader to access the locked memory banks within the tag, permission to change the lock status of the memory banks (except the reserved memory bank), and write data into the tag, etc.

Based on the above-mentioned access password and locking features available with C1G2 UHF tags, we propose the following approach, where the tag need not be killed permanently in order to protect consumer privacy. Once a tagged item is purchased by Alice, the trustable clerk at the point-of-sale can retrieve the tag's access password from the store's EPCIS and using this access password, the clerk can lock all the memory banks of the tag including the EPC memory bank. Alice can download and store the EPC numbers and their corresponding access passwords into her mobile/smart phone. This can be made possible via the mRFID-enabled mobile/smart phone communicating with the mRFID-module at the point-of-sale. With this proposed approach, adversary Charlie can no longer get any information (including the EPC number) from the RFID tags that are in the possession of Alice, as all the memory banks of the tags are locked and Charlie does not have the access passwords.

However in our chapter "RFID: Tag←Reader→Server Security" we showed that the C1G2 protocol's *Access* procedure using Access Password is not secure and we proposed our *DCSTaR: Diffusion-Confusion based Light-weight Security for RFID Tag-Reader Communication* protocol. Therefore if our *DCSTaR* protocol is deployed, the clerk can instead use our 96bit Key K to lock the EPC memory bank. Just by locking the EPC memory bank we protect the consumer privacy.

5.4 Interacting with Smart Home Environment

Scenario III: *After purchasing the RFID tagged items from the store, the point-of-sale terminal allows Alice to download and store the EPC numbers and their corresponding access passwords into her mRFID-enabled mobile/smart phone. Alice uses her smartphone's 3G/4G/Wi-FI network to establish a HTTPS (Hypertext Transfer Protocol Secure) connection [58] with her home server, in order to send the EPC numbers and their access passwords/keys. Based on the EPC numbers, the home server identifies the appropriate EPCIS and using the access passwords/keys as proof of purchase, downloads the related information (product description, size, weight, manufacturing date, expiry date, directions to use, ingredients, warranty certificate, etc.) associated with the EPC numbers. The EPCIS must provide only the information, which is relevant to the consumer who purchased the items. Therefore, by the time Alice reaches her home with the purchased tagged items, the home server is ready with all the information about the items.*

5.4.1 Secure Communication between mRFID-Smartphone and Home Server

Alice's mRFID-enabled smartphone can establish a HTTPS connection with the home server, before sending the EPC numbers and their corresponding access passwords/keys as shown in Fig.5.1. Otherwise the communication channel between the smartphone and the home server can be easily eavesdropped, and prone to man-in-the-middle attacks, replay attacks, data manipulation and corruption. A HTTPS communication uses cryptographic tunneling protocols to provide the intended confidentiality (preventing snooping and Packet sniffing), sender authentication (preventing identity spoofing), and message integrity (preventing message alteration) to achieve privacy. When properly chosen, implemented, and used, such techniques can provide secure communications over unsecured networks.

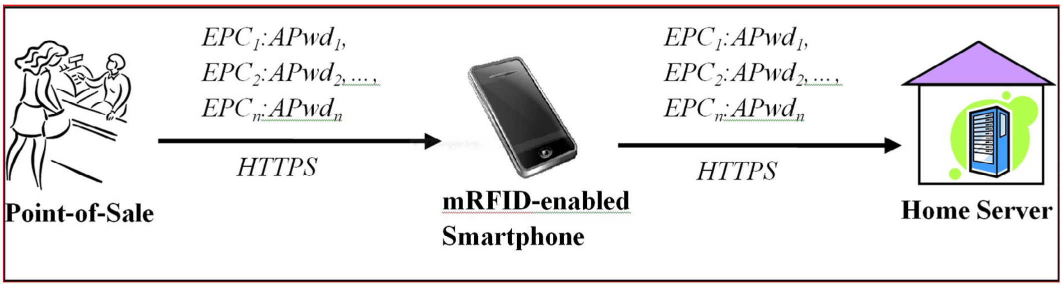


Figure 5.1: Secure Communication between mRFID-Smartphone and Home Server

5.4.2 Secure Communication between Home Server and EPCIS

After obtaining the EPC numbers from Alice's smartphone, home server now needs to contact the appropriate EPCIS to download the related information associated with the EPC numbers. As per the EPCglobal Architecture Specification [11], there exists an Object Naming Service (ONS), which can assist the home server to locate the EPCIS. The ONS provides a global lookup service to translate an EPC number into one or more Internet Uniform Reference Locators (URLs) where further information on the item may be found. The Root ONS provides the initial point of contact for ONS lookups. In most cases, the Root ONS delegates the remainder of the lookup operation to a Local ONS, which is within the control of the enterprise. The home server establishes a HTTPS connection with the EPCIS, before sending the EPC numbers and their corresponding access passwords as shown in Fig.5.2.

The clerk at the point-of-sale gives away the access passwords to only those consumers who purchased the tagged items. The EPCIS already has the list of EPC numbers and their corresponding access passwords, therefore when the home server sends the access passwords to EPCIS it proves that Alice/home server indeed purchased the tagged items.

However if our *DCSTaR* protocol is deployed, for better security the EPCIS would send a challenge R to the home sever (as in Step 1-Fig.3.7). Since the home server already has the respective keys of each tag, it can execute the *DCSTaR* protocol and send the EPC,S_1, S_2, and C to the EPCIS via the HTTPS connection. The EPCIS would then verify C corresponding to each EPC number and if verified it replies with the EPC related information.

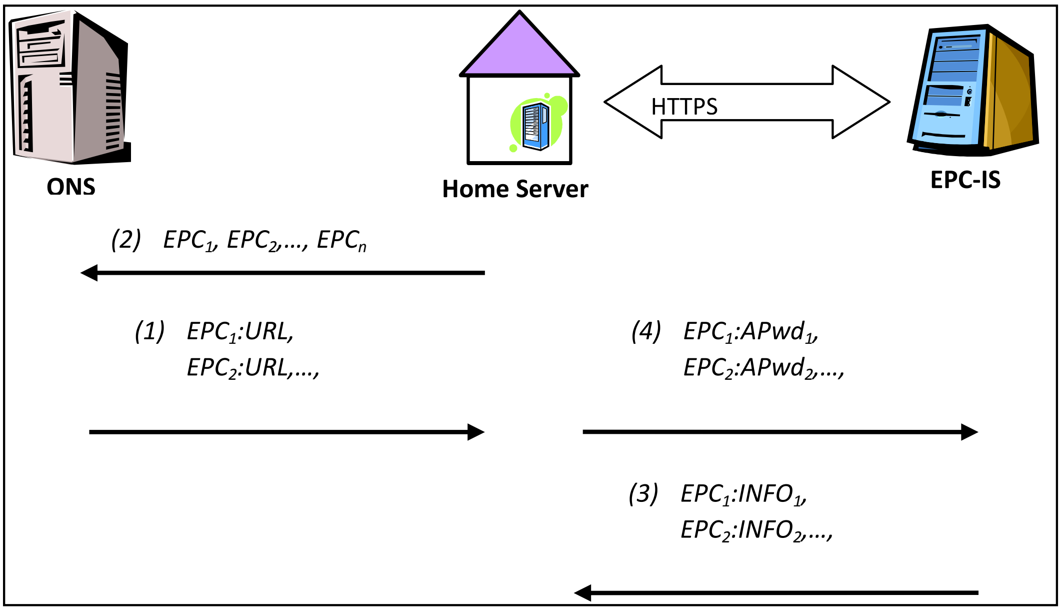

ONS
Home Server
HTTPS
EPC-IS
(2) EPC1, EPC2,…, EPCn
(1) EPC1:URL, EPC2:URL,…,
(4) EPC1:APwd1, EPC2:APwd2,…,
(3) EPC1:INFO1, EPC2:INFO2,…,

Figure 5.2: Secure Communication between Home Server and EPCIS

Scenario IV: *Alice stacks the tagged items in the RFID Reader-enabled refrigerator. The RFID reader in the refrigerator realizes that some of the tagged items do not respond with their EPC numbers, which means that these are newly added items and their memory banks are all locked. The RFID reader securely communicates with the home server to retrieve the access passwords/keys and unlocks the tags' memory banks. Whenever Alice requests for listing the items in the refrigerator, the RFID reader in the refrigerator collects all the EPC numbers from the tags and sends them to the home server. Home server would retrieve the information associated with these EPC numbers and displays the same on the refrigerator's display screen.*

5.4.3 Secure Communication: Reader-enabled Appliance & Home Server

RFID Reader-enabled appliance (e.g., refrigerator) must identify, authenticate and establish a HTTPS connection with the home server. We should also consider a threat where a malicious powerful RFID reader positioned outside the smart home, may impersonate as a genuine RFID reader-enabled appliance inside the home. Therefore, whenever a new RFID reader-enabled appliance/device is brought into the house, the home server would generate a private key and a public-key certificate for that appliance. Alice would then configure the appliance with its private key and public-key certificate and also installs the public-key certificate of the home server, in order for the appliance and the home server to successfully establish the HTTPS communication. This approach is depicted in Fig.5.3.

RFID reader in the refrigerator does not get any EPC number from the newly added items in the refrigerator as their memory banks are all locked. In such a situation, RFID reader communicates with the home server and requests for all the RFID tag access passwords/keys that have been downloaded by the server (from EPCIS) but not yet activated in the smart home. Home server sends all those access passwords/keys (probably few in number) to the RFID reader in the refrigerator and the reader checks each of these passwords/keys with every locked tag until a particular tag responds with its EPC number. With this approach a tag can be unlocked without knowing its EPC number initially. This approach can be easily understood by looking at the Fig.5.4.

Scenario V: *Alice has a RFID reader-enabled refrigerator, which stores many tagged items. All these tagged items emit their EPC number when queried by the RFID reader inside the refrigerator. But this poses a threat, where a malicious powerful RFID reader positioned outside the smart home, may be able to query the tagged items in the refrigerator and retrieve their EPC numbers. Then the malicious reader may communicate with EPCIS and retrieve information associated with these EPC numbers. This leads to privacy violation.*

5.4.4 Protecting Smart Home Residents Privacy

To alleviate the above mentioned problem, we propose the following approach: Once the RFID reader in the refrigerator unlocks the tags, it can assign a different unique tag ID (pseudonym) and write this pseudo ID into the User memory bank of the tag. After which, except the user memory bank, the RFID reader must also lock all the other memory banks including the EPC memory bank. The reader notifies the new pseudo ID to the home server, which maintains the reference between the EPC number and its new pseudo ID

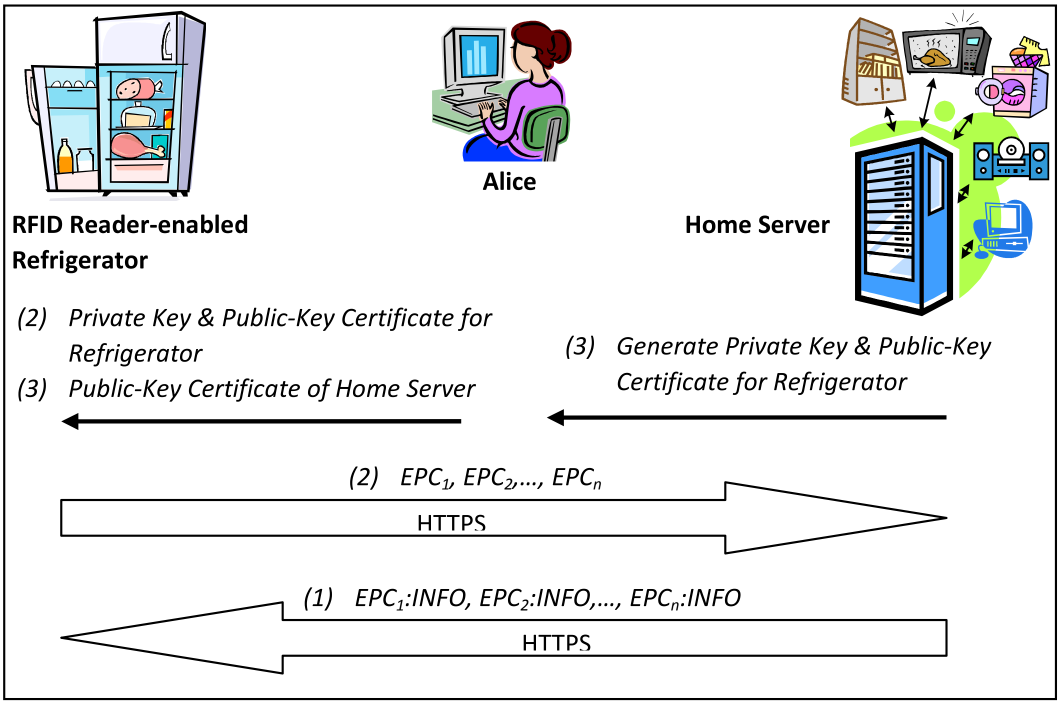

RFID Reader-enabled Refrigerator
Alice
Home Server
(2) Private Key & Public-Key Certificate for Refrigerator
(3) Public-Key Certificate of Home Server
(3) Generate Private Key & Public-Key Certificate for Refrigerator
(2) $EPC_1, EPC_2, \ldots, EPC_n$
HTTPS
(1) EPC_1:INFO, EPC_2:INFO,…, EPC_n:INFO
HTTPS

Figure 5.3: RFID Reader-enabled Appliance Configuring Process

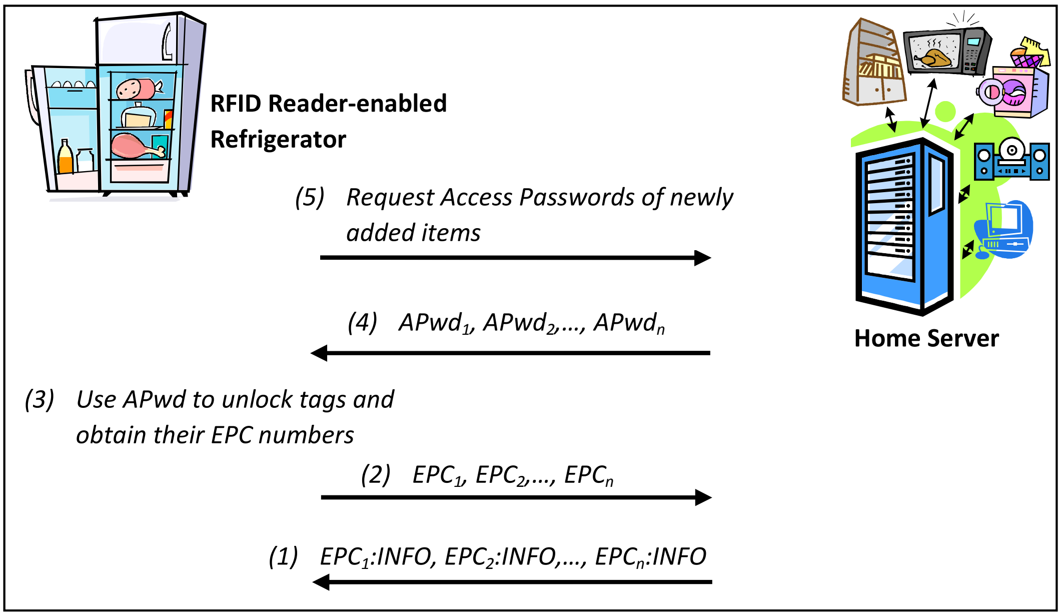

RFID Reader-enabled Refrigerator
(5) Request Access Passwords of newly added items
(4) $APwd_1, APwd_2, \ldots, APwd_n$
(3) Use APwd to unlock tags and obtain their EPC numbers
(2) $EPC_1, EPC_2, \ldots, EPC_n$
(1) EPC_1:INFO, EPC_2:INFO,…, EPC_n:INFO
Home Server

Figure 5.4: Unlocking RFID tags by RFID Reader-enabled Device

number. From now on whenever the RFID reader inside the refrigerator queries the tags in the refrigerator, they all respond with their new pseudo IDs completely different from their original EPC numbers. And only this new pseudo ID will be used in the smart home environment. Even if a malicious RFID reader gets these unique pseudo IDs he cannot obtain any information by sending pseudo IDs to EPCIS, as the EPCIS has no knowledge about these new pseudo IDs.

On the other hand if our *DCSTaR* protocol is deployed, since the tag no longer emits its EPC number, the RFID reader in the refrigerator executes *DCSTaR* protocol by just sending R to the tag. The tag responds with its 64 bit C, which in fact becomes the tag's Pseudo ID. The RFID reader in the refrigerator sends C to the home server, which uses this Pseudo ID to do a brute force search of all the tags in database that give out the same C and thus arrives at the correct EPC number. A consumer would not have that many items/tags; therefore we can assume that there would be no Pseudo ID collisions or computationally intensive database searches. Therefore since the adversary outside the home does not have the keys he cannot make sens of the Pseudo ID C.

5.5 Summary and Discussion

In this chapter we considered various RFID-based application scenarios that are suitable for Smart Home environment. Based on these scenarios we identified some of the security and privacy threats. We identified the need for protecting the consumer privacy and proposed ”Locking the Tag” approach. We also proposed security measures to provide authentication, data confidentiality, and data integrity between the following communicating entities: consumer’s mobile RFID-enabled smartphone, home server, Electronic Product Code – Information Services, RFID Reader-enabled household appliances and devices. We are confident that this conceptual idea would become a seed for further research and efficient modifications or improvements. Our future work includes practical implementation and thorough performance analysis.

6. Conclusion and Future Work

In this thesis we identified security and privacy threats pertaining to few specific RFID Applications that are beneficial to businesses and consumers. We carried out a security assessment of these RFID applications and proposed cryptographic protocols that achieve mutual authentication, secure communication and consumer privacy protection. Our major contributions in this thesis are as below:

Security Assessment of RFID: Supply Chain Management System This work has given us the opportunity to study and understand the fundamentals of RFID technology and it's benefits to supply chain management system. It allowed us to study the EPCglobal Inc.'s standard specification called the "EPCglobal Architecture Framework" for RFID-based supply chain management system. Having understood the functioning of various building blocks/entities in the EPCglobal architecture framework we analyzed their security aspects and identified certain threats. We proposed several security measures, which can be adopted to secure this architecture framework. We then narrowed down our research focus to RFID: Tag←Reader→Server Security. This work has also helped us to come up with some creative RFID application scenarios that are beneficial to consumers and address their related security and privacy issues. As a future work in this area we would like to focus on EPCIS Repository and EPCIS Accessing Application and address the security aspects of these two entities in greater detail.

RFID: Tag←Reader→Server Security In this chapter we considered low-cost passive RFID tags as they are expected to be widely used to tag items in the supply chain. Therefore, we analyzed the security aspects of the EPCglobal Standard "Class-1 Generation-2 (C1G2) UHF (Ultra High Frequency) RFID Protocol for Communications at 860MHz - 960MHz Version 1.2.0" also known as the ISO 18000 Part 6C standard. This standard specifies the Physical interactions (the signaling layer of the communication link) between readers and tags, and Reader and tag operating procedures and commands. We identified its security loopholes, which are prone to threats like the man-in-the-middle attacks, cloned fake tags, malicious readers, insider attacks, and consumer privacy violation. Since these low-cost passive tags have constrained resources, we first proposed a light-weight "Tag Reader Mutual Authentication (TRMA)" protocol that utilizes only light-weight primitives such as 16 bit: random number generator, CRC, and XOR. But TRMA protocol has its weaknesses, which allow the tag's access and kill passwords to be exposed. Based on the lessons learnt from the TRMA protocol, we proposed another protocol called the "Diffusion-Confusion based Light-weight Security for RFID Tag-Reader Communication (DC-STaR)". The DCSTaR protocol utilizes the very same light-weight primitives to provide a simple 32 bit diffusion-confusion procedure which obscures the challenge and response form the tag to the reader. We proved the strength of our cipher by subjecting several mega bytes of our cipher output to batteries of statistical random tests. We used C language parallel programming to execute our cipher in a cluster computing environment in order to speed up these batteries of tests. Our future work would include precise practical design and implementation of DCSTaR protocol and evaluate its die size, clock cycles, throughput, and power consumption.

Efficient and Privacy-Preserving mRFID/NFC Payment Model In this chapter we present the fact that the current mobile payment systems are based on credit/debit card payment models. We emphasize that credit/debit card payment models are prone to consumer privacy violation, card fraud, and expensive interchange fees for merchants. Therefore we proposed an alternate mobile payment model called the "Pre-Paid Mobile HTTPS-based Payment model" where the customer obtains the merchant's bank account information into his/her mRFID-enabled smartphone via the NFC protocol, then the customer using the smartphone instructs his/her bank to transfer the money to the merchant's bank account. The proposed payment model utilizes partially blind signature scheme to hide the customers' identity from the bank. Therefore the proposed payment model provides the customer with complete control on his/her payments and privacy protection from both the bank and the merchant. In our proposed payment model we converged Smartphones, mRFID/NFC technology, RFID technology, and HTTPS communication. The consumer can cancel and obtain new anonymous pre-paid cash certificates whenever and wherever he/she wants using the smartphone's 3G/4G/Wi-Fi networks. We gained good experience in developing a proof-of-concept implementation using Android SDK. Our future work would include a real practical implementation of our payment model using a NFC-enabled smartphone and developing a smartphone application that can communicate with the bank server via 3G/4G/Wi-Fi network and also communicate with a real NFC-module to download the merchant's bank account information and invoice.

mRFID↔Smart Home Security and Privacy In this chapter we considered various

RFID-based application scenarios that are suitable for Smart Home environment. Based on these scenarios we identified some of the security and privacy threats. We identified the need for protecting the consumer privacy and proposed ”Locking the Tag” approach. We proposed security measures to provide authentication, data confidentiality, and data integrity between the following communicating entities: consumer’s mobile RFID-enabled smartphone, home server, Electronic Product Code – Information Services, RFID Reader-enabled household appliances and devices. We are confident that this conceptual idea would become a seed for further research and efficient modifications or improvements. Our future work includes practical implementation and thorough performance analysis.

요 약 문

RFID 응용을 위한 프라이버시 보호 및 보안 프로토콜

RFID는 사람, 동물, 재산 등과 같은 개체를 효과적이며 빠르게 자동 식별하기 위한 수단이기에 비즈니스 분야에서 공급망 관리는 대표적인 응용분야이다. 조만간 RFID는 모든 상품에 부착될 것으로 기대되며, 스마트폰과 같은 대다수의 전자기기들은 RFID 리더 혹은 태그가 부착되어 나오기 때문에 사용자들의 다양한 일상생활에 응용될 수 있을 것으로 예측된다. 본 졸업논문에서는 비즈니스 및 사용자들에게 유용한 세가지RFID 응용분야를 중점적으로 살펴보고자 한다.

먼저, EPCGlobal 표준화 구조 프레임에 적한한 RFID 기반의 공급망 관리에서의 보안에 대해 살펴보고자 한다. 이를 위해 표준화 프레임워크의 각 개체 별 통신 인터페이스 보안 위협 및 미치는 영향을 분석하였으며, 위조된 RFID 태그 복제, 악의적인 RFID 리더, 사용자 프라이버시 침해가 RFID 기반의 공급망 관리 시스템의 주요 위협으로 파악되었다. 상호인증, 기밀성, 무결성, 안전한 키 분배 및 보호, 태그 익명성과 같은 보안 요구사항을 도출하였다. CRC 검사, XOR 함수, 난수 생성기를 활용해 두 가지 경량화 암호화 프로토콜을 제안하였다.

앞으로 RFID가 모바일 폰 혹은 스마트폰에 추가됨에 따라 새로운 결제 방법이 가능하다. 기존에 신용카드 기반의 결제 시스템에서 발생가능 한 신용카드 사기, 도난, 사용자 프로파일링과 같은 문제가 발생하기에 본 졸업논문에서는 RFID 기반의 모바일 결제 시스템을 새롭게 제안해 기존 기법에서의 문제를 해결할 수 있는 하나의 대안을 제안하였다. 제안된 기법은 모바일 HTTPS를 활용해 선결제 시스템이며, Android SDK를 활용해 제안 기법을 구현해 제안 기법의 효율성을 보여주었다.

RFID 태그 기반의 전자기기가 활성화 됨에 따라 사용자들이 RFID 태그가 부착된 제품을 자신들의 집에서 활용할 수 있기에 본 졸업논문에서는 새로운 보안 프레임워크를 제안하였다. 이를 위해 발생 가능한 각종 위협 파악 및 보안 요구사항을 도출하였다. 제안된 보안 프레임워크를 통해 사용자들은 RFID 태그가 부착된 제품을 자신들의 집에서 재활용할 수 있기에 RFID 태그 기반의 전자기기의 활용성을 증가시킬 수 있다.

References

[1] Arco, P. D., and Santis, A. D. , “From Weaknesses to Secret Disclosure in a Recent Ultra-Lightweight RFID Authentication Protocol”, Cryptology ePrint Archive, 2008. http://eprint.iacr.org/2008/470

[2] Avoine, G., and Oechslin, P., “A Scalable and Provably Secure Hash-Based RFID Protocol”, Proceedings of Workshop on Pervasive Computing and Communications Security, PerSec’05, pp. 110-114, IEEE Press, 2005.

[3] Biham, E., and Shamir, A., “Differential cryptanalysis of DES-like cryptosystems”, Advances in Cryptology - CRYPTO ’90 , Lecture Notes in Computer Science 537, pp. 2-21. Springer-Verlag, 1991.

[4] Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., and Vikkelsoe, C., “PRESENT: an ultra-lightweight block cipher”, in Proc. of CHES’07, vol. 4727 of LNCS, pp. 450–466, 2007.

[5] Burmester, M., and Munilla, J., “A Flyweight RFID Authentication Protocol”, RFIDSec‘09, http://www.cosic.esat.kuleuven.be/rfidsec09/Papers/paper-burmester-munilla.pdf, 2009.

[6] Burmester, M., De Medeiros, B., and Motta, R., “Anonymous RFID authentication supporting constant-cost key-lookup against active adversaries¡±, International Journal of Applied Cryptography, vol. 1, no. 2, pp. 79-90, 2008.

[7] Chien, H.Y., Chen, C.H., “Mutual authentication protocol for RFID conforming to EPC class-1 generation-2 standards”, Computer Standards & Interfaces, vol. 29, pp.254-259, 2007.

[8] Chien, H.Y., and Huang, C.W., “Security of Ultra-Lightweight RFID Authentication Protocols and Its Improvements”, ACM Operating System Rev., vol. 41, no. 2, pp. 83-86, July 2007.

[9] Chien, H.Y., “SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity”, IEEE Transactions on Dependable & Secure Computing, pp.337-340, 2007.

[10] D'Arco, P., and De Santis, A., "Weaknesses in a Recent Ultra-Lightweight RFID Authentication Protocol", AFRICACRYPT 2008, LNCS 5023, pp. 27-9, 2008.

[11] EPCglobal Ratified Specification, "The EPCglobal Architecture Framework", `http://www.epcglobalinc.org/standards/`

[12] EPCglobal Ratified Standard, "EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz - 960MHz Version 1.2.0", `http://www.epcglobalinc.org`.

[13] EPCglobal Ratified Standard, "EPCglobal Certificate Profile Version 2.0", `http://www.epcglobalinc.org/standards/cert`

[14] EPCglobal Web site, 2005, `http://www.EPCglobalinc.org`

[15] Feldhofer, M., Wolkerstorfer, J., and Rijmen, V., "AES implementation on a grain of sand", IEEE Proceedings In Information Security, vol. 152, no.1, pp. 13-20, 2005.

[16] Gartner Inc., "Dataquest Insight: Mobile Payment, 2007-2012", `http://www.gartner.com/it/page.jsp?id=995812`, 2009.

[17] H. Gilbert, M. Robshaw and H. Silbert, "An Active Attack Against HB+ - A Provably Secure Lightweight Authentication Protocol", Available at `eprint.iacr.org/2005/237.pdf`

[18] Gilbert, H., Robshaw, M.J.B., and Seurin, Y., "Good variants of HB+ are hard to find", Financial Cryptography'08, LNCS 5143, pp. 156-170, 2008.

[19] Gosset, F., Standaert, F.-X., and Quisquater, J.-J. "FPGA Implementation of SQUASH", 29th Symposium on Information Theory in the Benelux, pp 231-238, 2008.

[20] Hernandez-Castro, J. C., Tapiador, J.M.E., Peris-Lopez, P., Quisquater, J.-J., "Cryptanalysis of the SASI Ultralightweight RFID Authentication Protocol", `http://arxiv.org/abs/0811.4257`

[21] Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B.-S., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., and Chee, S., "HIGHT: A New Block Cipher Suitable for Low-Resource Device". In L. Goubin and M. Matsui, editors, Proceedings of CHES 2006, LNCS, volume 4249, pages 46–59, Springer-Verlag, 2006.

[22] A. Juels and S. Weis, "Authenticating Pervasive Devices with Human Protocols", CRYPTO'05, LNCS 3261, pp. 293-308, 2005.

[23] S. Karthikeyan, M. Nesterenko, "RFID security without extensive cryptography", SASN'05, ACM, pp.63-67, 2005.

[24] K. Kim, "Construction of DES-like S-boxes based on Boolean Functions Satisfying the SAC", Advances in Cryptology - Proc. of Asiacrypt'91, LNCS. 739, pp.59-72, 1991.

[25] F. Klaus, "RFID Handbook - Fundamentals and Applications in Contactless Smart Cards and Identification", 2nd Edition, John Wiley & Sons, 2003.

[26] G. Leander and A. Poschmann, "On the Classification of 4 Bit S-Boxes", Arithmetic of Finite Fields - WAIFI'07, LNCS 4547, pp.159-176, 2007.

[27] T. Li and G. Wang, "Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols", 22nd IFIP TC-11, 2007.

[28] T. Li and R.H. Deng, "Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol", AReS'07, 2007.

[29] P. L'Ecuyer, and R. Simard,"TestU01: a C library for empirical testing of random number generators", ACM Transactions on Mathematical Software, vol.33, no.4, article 22, 2007. http://www.iro.umontreal.ca/~simardr/testu01/tu01.html

[30] ISO/IEC 18092, "Near Field Communication Interface and Protocol (NFCIP-1)", http://www.iso.org/iso/catalogue_detail.htm?csnumber=38578

[31] M. Matsui, "Linear Cryptanalysis Method for DES Cipher", Advances in Cryptology - Eurocrypt '93, Lecture Notes in Computer Science 765, Springer-Verlag, pp. 386-397, 1993.

[32] NFC Forum website, http://www.nfc-forum.org/home

[33] K. Ouafi, R. Overbeck, and S. Vaudenay, "On the security of HB# against a man-in-the-middle attack", ASIACRYPT'08, LNCS 5350, pp. 108-124, 2008.

[34] P. Peris-Lopez, J.C. Hernandez-Castro, J.M.E. Tapiador, and J.C.A. van der Lubbe, "Security Flaws in a Recent Ultralightweight RFID Protocol", RFIDSec'10 Asia, http://arxiv.org/PS_cache/arxiv/pdf/0910/0910.2115v1.pdf, 2010.

[35] P. Peris-Lopez, J.C. Hernandez-Castro, J.M.E. Tapiador, and A. Ribagorda,“Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol”, Workshop on Information Security Applications, LNCS 5379, pp. 56-68, 2008.

[36] P. Peris-Lopez, “Lightweight Cryptography in Radio Frequency Identification (RFID) Systems”, PhD. thesis, Computer Science Dept., Carlos III University of Madrid, `http://www.lightweightcryptography.com/`, 2008.

[37] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador and A. Ribagorda, “Cryptanalysis of a Novel Authentication Protocol Conforming to EPC-C1G2 standard”, Elsevier-Computer Standards & Interfaces, 31(2), pp.372-380, 2009.

[38] 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”, RFIDSec’06, 2006.

[39] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador and A. Ribagorda, “EMAP: An efficient mutual authentication protocol for low-cost RFID tags”, IS’06, LNCS 4277, pp.352-361, 2006.

[40] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador and A. Ribagorda, “M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags”, UIC’06, LNCS 4159, pp.912-923, 2006.

[41] P. Peris-Lopez et al., “Vulnerability Analysis of a Mutual Authentication Scheme under the EPC Class-1 Generation-2 Standard”, RFIDSec’08.

[42] P. Peris-Lopez, T. Li, and J.C. Hernandez-Castro, “Lightweight Props on the Weak Security of EPC Class-1 Generation-2 Standard”, IEICE, Vol. E93-D, No.3, pp. 518-527, Mar. 2010.

[43] A. Poschmann, G. Leander, K. Schramm, and C. Paar, “New light-weight crypto algorithms for RFID”, in Proc. of IEEE International Symposium on Circuits and Systems, ISCAS’07, pp. 1843–1846, 2007.

[44] P. Rizomiliotis, E. Rekleitis, and S. Gritzalis, “Security analysis of the Song-Mitchell authentication protocol for low-cost RFID tags”, IEEE Communications Letters, vol 13, no. 4, pp. 274-276, 2009.

[45] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, “A statistical test suite for random and pseudorandom number generators for cryptographic applications”, NIST special publication 800-22, National Institute of Standards and Technology (NIST), 2001. http://csrc.nist.gov/rng/

[46] B. Song and C. J. Mitchell, “RFID authentication protocol for low-cost tags,¡± First ACM Conference on Wireless Network Security”, pp. 140-147, 2008.

[47] C.E. Shannon, “Communication Theory of Secrecy Systems”, Bell System Technical Journal, vol.28-4, pp. 656-715, 1949.

[48] VeriSign, “The EPCglobal Network: Enhancing the Supply Chain”, White Paper 2005, http://www.verisign.com/stellent/groups/public/documents/white_paper/002109.pdf

[49] L. Xiao and H.M. Heys, “Hardware Performance Characterization of Block Cipher Structures”, Topics in Cryptology — CT-RSA 2003, The Cryptographers’ Track at the RSA Conference 2003, San Francisco, CA, USA, April 13–17, 2003.

[50] E. Zenner, “Authentication for RFID Tags: Observations on the HB Protocols”, 4^{th} Interdisciplinary Seminar on Applied Mathematics, http://www.erikzenner.name/docs/2009_Aalborg_Talk.pdf, 2009.

[51] M. Abe and T. Okamato,“Provably Secure Partially Blind Signature”, In Proceedings of Annual International Cryptology Conference, LNCS 1880, pp. 271-286, 2000.

[52] S. Brands, “Untraceable off-line cash in wallets with observers”, In Proceedings of Annual International Cryptology Conference, pp. 302-318, 1993, ISBN 3-540-57766-1.

[53] T. Cao, D. Lin and R. Xue, ”A randomized RSA-based partially blind signature scheme for electronic cash”, Elsevier-Computers & Security, vol. 24-1, pp 44-49, 2005.

[54] D. Chaum, A. Fiat, and M. Naor, “Untraceable electronic cash”, In Proceedings of Annual International Cryptology Conference, pp. 319-327, 1988, ISBN 3-540-97196-3.

[55] D. Chaum, “Blind Signatures for Untraceable Payments”, In Proceedings of Annual International Cryptology Conference, pp. 199-203, 1982.

[56] F. Hayashi, “Do US consumers really benefit from payment card rewards?”, Economic Review, First Quarter, Federal Reserve Bank of Kansas City, https://www.kansascityfed.org/Publicat/ECONREV/PDF/09q1Hayashi.pdf, 2009.

[57] T.S. Heydt-Benjamin, D.V. Bailey, K. Fu, A. Juels, and T. O'Hare, "Vulnerabilities in First-Generation RFID-enabled Credit Cards", In Proceedings of Eleventh International Conference on Financial Cryptography and Data Security, LNCS 4886, pp. 2-14, 2007.

[58] Internet Engineering Task Force (IETF), Network Working Group, and E. Rescorla, "HTTP Over TLS", RFC2818, `http://tools.ietf.org/html/rfc2818`, 2000.

[59] ISO/IEC 14443-1∼4, "Identification cards – Contactless integrated circuit cards – Proximity cards", `http://www.iso.org/iso/iso_catalogue/ catalogue_tc/catalogue_detail.htm?csnumber=39693`, 2008.

[60] N. Massouda, A. Saundersb, and B. Scholnickc, "The cost of being late? The case of credit card penalty fees", Elsevier-Journal of Financial Stability, DOI: doi:10.1016/j.jfs.2009.12.001 , 2010.

[61] MasterCard Worldwide, "Tap & Go with MasterCard PayPass", `http://www.paypass.com/`.

[62] MasterCard Worldwide, "MasterCard Pioneers Innovation in Payments with NFC Enabled Mobile Phones", `http://www.mastercard.com/hk/personal/en/wce/pdf/19755_Microsoft_Word_-_0411_-_HK-_NFC_release_-_Eng_-FINAL.pdf`

[63] K. Michael and J.H. Burrows, "ELECTRONIC DATA INTERCHANGE (EDI)", National Institute of Standards and Technology, 1996/04/29. `http://www.itl.nist.gov/fipspubs/fip161-2.htm`

[64] National Institute of Standards and Technology (NIST), "Digital Signature Standard (DSS)", The Federal Information Processing Standards (FIPS) Publication 186-3, `http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf`,2009.

[65] S. Pritchard, "Data lost, not found ", Elsevier-Infosecurity, vol. 6-4, pp. 22-24, 2009.

[66] W. Roberds and S.L. Schreft, "Data breaches and identity theft", Elsevier-Journal of Monetary Economics, vol. 56-7, pp. 918-929, 2009.

[67] S. Schuhy, O. Shyz, and J. Stavins, "Who Gains and Who Loses from Credit Card Payments? Theory and Calibrations", The Economics of Payments IV - Federal Reserve Bank of New York, `http://newyorkfed.org/research/conference/2010/econ/reward28.pdf`, 2010.

[68] J. Vijayan, “Heartland data breach sparks security concerns in payment industry”, News article at Computerworld, http://www.computerworld.com/s/article/9126608/Heartland_data_breach_sparks_security_concerns_in_payment_industry, 2009.

[69] Visa USA, “VISA PAYWAVE”, http://usa.visa.com/personal/cards/paywave/index.html

[70] Visa Europe, “Visa Contactless - the wave and pay alternative to cash for low value transactions”, http://www.visaeurope.com/pressandmedia/factsheets/visacontactless.jsp

Acknowledgement

First of all, I thank my advisor, Prof. Kim, Kwangjo for his guidance and support throughout the course of my study in KAIST. I would like to thank the other members of my doctoral advisory committee: Prof. Kim, Daeyoung of KAIST, Prof. Kim, Soontae of KAIST, Prof. Lee, Byoungcheon of Joongbu University and Dr. Choi, Dooho of ETRI. Their insightful comments and suggestions have greatly helped me in my research work.

I also had the privilege of working with many kind and enthusiastic people at the Cryptology and Information Security lab at KAIST. I am very thankful for their research advice, encouragement, and support. I would like to personally thank Jangseong Kim, Kyusuk Han, Dang Nguyen Duc, Hyunrok Lee, Zeen Kim, Made Harta Dwijaksara, Junhyun Yim, Yi Jae Park, Doyoung Chung, and Ji Yeon Hong.

Most importantly, I thank God for his grace and for leading me through at all times. I thank my friends, elders, leaders, deacons and pastors at the "Saeronam Community Church" in Daejeon for their prayers and love.

I want to express my deepest gratitude to all the members of my family; My father: K.P. Munirathnam, my mother: K. Kiranmayee, my sister: Geetika, my brother-in-law: Vijay, my father-in-law: Oh Pil Jin, my mother-in-law: Choi Myung Kyu, and most of all, my beloved wife, Oh Jayoung, who is just awesome!. Their prayers, love, support and constant encouragement is the source of my inspiration and immense happiness and keeps me going on. I therefore dedicate this thesis to all my family members.

Curriculum Vitae

Name	:	Konidala, Divyan Munirathnam
Date of Birth	:	May 30, 1978
Birthplace	:	Rajahmundry, Andhra Pradesh, INDIA
Address	:	291 Daehak-ro, Yuseong-gu, Daejeon 305-701, REPUBLIC OF KOREA
E-mail	:	divyan@kaist.ac.kr

Educations

1995. 09. – 2000. 05. Computer Science and Engineering, Bangalore University, INDIA (B.S).

2002. 02. – 2004. 02. Information Security Track, Information and Communications University (merged into KAIST), School of Engineering, KAIST, REPUBLIC OF KOREA (M.S).

2004. 02. – 2010. 12. Information Security Track, Department of Information and Communications Engineering, KAIST, REPUBLIC OF KOREA (PhD).

Career

1999. 09. – 1999. 12. Assistant Computer Lab Coordinator at Seshi Computers (P) Ltd., Chennai, India.

1999. 12. – 2000. 03. Computer Lab Coordinator at El Net-3L Ltd., Chennai, India.

2000. 03. – 2000. 12. Programmer in the Web Development team, Eonour Software Limited, Chennai, India.

2000. 12. – 2001. 04. Freelancing Web Developer, Embassy of India, Seoul, Republic of Korea.

2001. 04. – 2001. 11. Freelancing Web Application Developer, ABIS Systems Limited, Seoul, Republic of Korea.

2003. 02. – 2003. 06. Teaching Assistant for ICE0605 (*Modern Cryptography*) by Prof. C. Pandu Rangan at Information and Communications University (merged into KAIST).

2003. 06. – 2003. 09. Research Internship at the *Dept. of Cryptography and Security, Institute for Infocomm Research (I^2R)*, Singapore.

2004. 03. – 2005. 02. Graduate Research Assistant for the project *A Group-Aware Middleware Infrastructure for Active Surroundings* Funded by: Ministry of Information and Communications at Information and Communications University (merged into KAIST).

2004. 02. – 2004. 06. Teaching Assistant for ICE1200 (*Data Structures*) by Prof. Myaeng, Sung Hyon at Information and Communications University (merged into KAIST).

2005. 02. – 2005. 06. Teaching Assistant for ICE1200 (*Data Structures*) by Prof. Myaeng, Sung Hyon at Information and Communications University (merged into KAIST).

2008. 01. – 2008. 12. Graduate Research Assistant for the project *Mobile and Next Generation RFID Technology Standards Development.* Funded by ICT Standardization program of the Republic of Korea's MKE (The Ministry of Knowledge Economy).

2010. 01. – 2010. 12. Graduate Research Assistant for the project *Mobile and Next Generation RFID Technology Standards Development.* Funded by ICT Standardization program of the Republic of Korea's MKE (The Ministry of Knowledge Economy).

2006. 01. – 2011. 02. Graduate Research Assistant in the *Auto-ID Labs, Korea* at Information and Communications University/KAIST.

Publications

1. **Divyan M. Konidala**, Kwangjo Kim, and Daeyoung Kim, "Diffusion-Confusion based Light-weight Security for Item-RFID Tag-Reader Communication", Journal of Internet Technology (JIT), Accepted on 2010/09/08, to appear 2011. [SCIE]

2. **Divyan M. Konidala**, Made H. Dwijaksara, Kwangjo Kim, Dongman Lee, Daeyoung Kim, Byoungcheon Lee, and Soontae Kim, "Resuscitating Privacy-Preserving Mobile Payment with Customer in Complete Control", Springer - Personal and Ubiquitous Computing (PUC), Accepted on 2010/10/06, to appear 2011. [SCIE, IF=1.554]

3. **Divyan M. Konidala** and Kwangjo Kim, "Mobile RFID Applications and Security Challenges", The 9th Annual International Conference on Information Security and Cryptology (ICISC 2006), LNCS 4296, pp.194-205, Nov. 30 Dec. 1, 2006, Busan, Korea. [SCIE, IF=0.402]

4. Xiaofeng Chen, Fangguo Zhang, **Divyan M. Konidala**, and Kwangjo Kim, "New ID-based Threshold Signature Scheme from Bilinear Pairings", The 5th International Conference on Cryptology in India (INDOCRYPT 2004), LNCS 3348, pp. 371-383, Dec.20 22, 2004 Chennai(Madras), India. [SCIE, IF=0.513]

5. **Divyan M Konidala**, Daeyoung Kim, Chan Yeob Yeun, and Byoungcheon Lee, "Security Framework for RFID-based Applications in Smart Home Environment", Journal of Information Processing Systems (JIPS), Accepted on 2010/10/14, to appear 2011.

6. Dang N. Duc, **Divyan M. Konidala**, Hyunrok Lee, and Kwangjo Kim, "RFID Security: A Research Survey and Solution to Some Open Problems", International Journal of Internet Technology and Secured Transactions (IJITST), Vol. 2, Nos. 3/4, pp. 222 - 249, 2010.

7. **Divyan M. Konidala**, Zeen Kim, and Kwangjo Kim, "A Simple and Cost-effective RFID Tag-Reader Mutual Authentication Scheme", International Conference on RFID Security 2007 (RFIDSec 2007), pp.141-152, Jul. 11-13, 2007, Malaga, Spain.

8. **Divyan M. Konidala**, Dang N. Duc, and Kwangjo Kim, "A Capability-based Privacy-preserving Scheme for Pervasive Computing Environments", 2nd IEEE International Workshop on Pervasive Computing and Communications Security (PerSec 2005) in conjunction with PerCom 2005, pp.136-140, Mar. 8 12, 2005, Hawaii, USA.

9. **Divyan M. Konidala**, Chan Yeob Yeun, and Kwangjo Kim, "A Secure and Privacy Enhanced Protocol for Location-based Services in Ubiquitous Society", 47th annual IEEE Global Telecommunications Conference 2004 (Globecom 2004), 2004 IEEE Computer Society, Volume 4, pp. 2164 - 2168, Nov.29 Dec.3, 2004, Dallas, Texas, USA.

10. **Divyan M. Konidala**, Hyunrok Lee, Dang Nguyen Duc, and Kwangjo Kim, "Security and User Privacy for Mobile-RFID Applications in Public Zone", in the pro-

ceedings of Triangle Symposium on Advanced ICT 2008 (TriSAI 2008), pp: 207-212, 2008.

11. **Divyan M. Konidala** and Kwangjo Kim, "Security for RFID-based Applications in Smart Home Environment", 2007 Symposium on Cryptography and Information Security (SCIS 2007), Abstracts pp.120, Jan. 23 26, 2007, Sasebo, Japan.

12. **Divyan M. Konidala** and Kwangjo Kim, "Mobile RFID Security Issues", 2006 Symposium on Cryptography and Information Security (SCIS 2006), Abstracts pp.166, Jan. 17 20, 2006, Hiroshima, Japan.

13. **Konidala M. Divyan**, Robert H. Deng, Jianying Zhou, and Kwangjo Kim, "A Secure and Privacy Enhanced Location-based Service Transaction Protocol in Ubiquitous Computing Environment", 2004 Symposium on Cryptography and Information Security (SCIS 2004), vol. 1/2, pp. 931 936, Jan.27 30, 2004, Sendai, Japan.

14. **Divyan M. Konidala**, Kwangjo Kim, and Woan-Sik Kim, "Security Assessment of RFID-based Supply Chain Management System", 2007 Anti-Counterfeiting & Secure Supply Chain Whitepaper Series by Auto-ID Labs. Published on 2007/01/15

15. **Divyan M. Konidala** and Kwangjo Kim, "RFID Tag-Reader Mutual Authentication Scheme Using Tag's Access Password", 2007 Anti-Counterfeiting & Secure Supply Chain Whitepaper Series by Auto-ID Labs. Published on 2007/01/15

16. **Divyan M. Konidala** and Kwangjo Kim, "Light-weight Security for RFID Tag-Reader Communication", CISC-S'10, 2010년도 한국정보보호학회 하계정보보호학술대회, pp.212-215, June 18, 2010, POSTECH, 포항.

17. 신승목, 이현록, **Divyan M Konidala**, 김광조, "비트 스크램블을 통해 개선된 RFID-sec07의 상호 인증 프로토콜", 2008년도 한국정보보호학회 영남지부 학술발표회 논문집, pp.89-93, 2008.2.20, 동서대학교, 부산.

18. **Divyan M Konidala**, 김진, 윤찬엽, 김광조, "Secure Approach to Deploy RFID-based Applications in Smart Home Environment", CISC-W'07, 2007년도 한국정보보호학회 동계학술대회 논문모집 및 정기총회, vol.17, no.2, pp.717-720, 2007.12.1, 상명대학교, 서울.

19. **Divyan M. Konidala** and Kwangjo Kim, "Light-weight RFID Tag-Reader Mutual Authentication Scheme", 2006년도 정보보호학술발표회논문집, pp. 179-194, 2006. 9.29-30, 목원대학교, 대전.

20. **Divyan M. Konidala**, Dang N. Duc, Dongman Lee and Kwangjo Kim, "A Capability-based Privacy-preserving Scheme for Pervasive Computing Environments", CISC-S'04,

2004년도 한국정보보호학회 하계정보보호학술대회, Vol.14, No.1, pp. 300 – 312, 2004.6.24 26, 경동대학교, 속초.

21. **Konidala M. Divyan** and Kwangjo Kim, “A Secure Location-Based Service Reservation Protocol in Pervasive Computing Environment”, CISC-W’04, 2003년도 한국정보보호학회 동계학술대회, pp.669-685, 2003.12.6 한양대학교, 서울.