

Lightweight RFID Authentication Protocols for Special Schemes

Xuefei Leng

Technical Report
RHUL-MA-2011-1
10 February 2011



Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England

<http://www.rhul.ac.uk/mathematics/techreports>

**LIGHTWEIGHT RFID
AUTHENTICATION PROTOCOLS
FOR SPECIAL SCHEMES**

Xuefei Leng

Royal Holloway and Bedford New college,
University of London

*Thesis submitted to
The University of London
for the degree of
Doctor of Philosophy
2010.*

Declaration

I, Xuefei Leng, hereby declare that these doctoral studies were conducted under the supervision of my supervisor Keith Mayes and my advisor Konstantinos Markantonakis.

The work presented in this thesis is the result of original research conducted by myself, in collaboration with others, whilst enrolled in the Department of Mathematics as a candidate for the degree of Doctor of Philosophy. This work has not been submitted for any other degree or award in any other university or educational establishment.

Signature:

Date:

Abstract

This thesis addresses the problem of providing secure protocols for the practical application of low cost Radio-Frequency Identification (RFID) tags. This is particularly challenging because the tags are limited in cryptographic functionality and general performance, but also when there is a need to read multiple tags in a range of practical environments whilst maintaining data privacy. Several protocols exist for reading of RFID tags, either individually or within a group; however some have design weaknesses and/or would be impractical for a real-world implementation. The contribution of this work is first to review and propose improvements to the security of existing protocols for single tags and then to groups of tags. It then goes on to propose optimised protocols, exploiting capabilities that are common in real RFID tag systems, but are overlooked in most theoretical protocols. For the reading of individual tags, reference is made to the HB family of RFID protocols and in particular the lightweight HB-MP protocol. This thesis contributes analysis revealing that the protocol is vulnerable to man-in-the-middle attacks and then goes on to propose an improved version HB-MP⁺. For multi-tag group reading an analysis is performed of protocols based on the yoking proof mechanism proposed by A. Juels, identifying problems with tag ordering and data privacy. This leads to original proposals for improved protocols offering tag order independence, privacy, select-response optimisations, binary tree anti-collision groupings and simultaneous tag authentication via bit collision processing. .

Acknowledgements

The completion of this thesis would simply not have been possible without the support, encouragement and involvement of a number of people and organisations that I would like to mention here. My huge thanks go to my supervisor, Dr. Keith Mayes, for giving me an opportunity to become a part of his exciting research team, for sharing his profound knowledge with me, for teaching me the research, academic, and professional skills. I also would like to thank my advisor Dr. Konstantinos Markantonakis for always being helpful and encouraging to me. Additionally I would like to thank Smart Card Centre for offering scholar and studentship in the periods of my doctoral study, without which I could not have pursued my studies. Both Smart Card Centre and the Information Security Group were forthcoming with financial support for all travel related to my studies.

A great thank must be given to the Smart Card Centre for the provision of this great study opportunity and all facility supports. I would also like to thank our industrial founders: G&D, Vodafone, and industry sponsors: Transport of London and ITSO. Their continuous technical supports give me a broad horizon in the area of smart card and RFID technology.

My special thanks to Gerhard Hancke for his notable help with this thesis and my study. Also I would like to thank Mr. Yuanhung Lien, He is my best co-author and give me great support on my research. I would also like to thank my parents; without their cares and supports, I will not finish my PhD study.

Publication

A number of papers resulting from my work in this thesis have been presented in referred conferences and journals. Here is the list of my publications:

1. Xuefei Leng, Gerhard Hancke, Keith Mayes and Konstantinos Markantonakis. "Group Authentication of RFID Tags Using Bit-Collision Patterns", Submitted to Proceedings of IEEE, RFID Special Issue, 2010.
2. Yuanhung Lien, Xuefei Leng, Keith Mayes, and Jung-Hui Chiu, "Select-Response Grouping Proof and Its Verification Protocol for RFID Tags", International Journal of Intelligent Information and Database Systems, 2010.
3. Xuefei Leng, Yuanhung Lien, Keith Mayes, Jung-Hui Chiu and Konstantinos Markantonakis. "Select-Response Grouping Proof for RFID Tags", 1st Asian Conference on Intelligent Information and Database Systems, Quang Binh University, Dong Hoi City, Quang Binh Province, Vietnam, 1-3 April 2009.
4. Xuefei Leng, Keith Mayes and Konstantinos Markantonakis. "HB-MP+ Protocol: An Improvement on the HB-MP Protocol", 2008 IEEE International Conference on RFID, April 16-17,2008, Las Vegas, Nevada.
5. Yuanhung Lien, Xuefei Leng, Keith Mayes, and Jung-Hui Chiu, "Reading Order Independent Grouping Proof for RFID Tags", 2008 IEEE International Conference on Intelligence and Security Informatics, June 17-20, 2008, Taipei, Taiwan.
6. Xuefei Leng, Yuanhung Lien, Keith Mayes and Konstantinos Markantonakis, "An RFID Grouping Proof Protocol Exploiting Anti-collision

Algorithm for Subgroup Dividing”, International Journal of Security and Networks (IJSN) Special Issue on ”Security and Privacy in RFID Systems”, 2010.

Contents

Declaration	2
Abstract	3
Acknowledgements	4
Publication	5
Contents	7
List of Figures	12
1 Introduction	14
1.1 RFID System	15
1.1.1 Architecture of RFID Systems	15
1.2 Selection Criteria for RFID Systems	17
1.2.1 Operating Frequency	17
1.2.2 Interference Sensitivity	18
1.2.3 Practical Reading Range	19
1.2.4 Security Requirements	20
1.2.5 Memory Capacity	21
1.3 Practical Applications	22
1.3.1 Transportation Payments	22
1.3.2 Aviation and Railway	23
1.3.3 Access Control	24
1.3.4 Library	25
1.3.5 E-Passports	26
1.3.6 Anti-Counterfeiting	26

1.4	Potential Application	27
1.4.1	Supply Chains	27
1.4.2	Pervasive Computation	31
1.5	Security and Privacy Issues on RFID Protocol Design	32
1.5.1	Security	32
1.5.2	Privacy	33
1.5.3	Computation Limits of RFID Tags	33
1.6	Storyline of Protocol Introduction	35
2	HB-MP⁺ Protocol	37
2.1	Introduction	37
2.2	Prerequisites	39
2.2.1	Binary Inner Product	39
2.2.2	LPN Problems	40
2.3	HB ⁺ Protocol	40
2.4	The HB-MP' Protocol and Its Weakness	43
2.4.1	The HB-MP' Protocol	43
2.4.2	A Man-in-the middle Attack on the HB-MP' Protocol	45
2.5	The HB-MP Protocol and Its Weakness	47
2.5.1	The HB-MP Protocol	47
2.5.2	A Man-in-the-middle Attack on the HB-MP Protocol	48
2.6	An Improved HB-MP Protocol	50
2.7	An Abstract Form of HB-MP ⁺ Protocol	51
2.8	Security and Performance Analysis of The HB-MP ⁺ Protocol	53
2.8.1	Security Analysis	53
2.8.2	Performance Analysis	56
2.9	Conclusion and Future Work	57
3	Introduction to Grouping Proof Protocols	58
3.1	Preliminaries	58

3.1.1	Protocol Goal	58
3.1.2	Environment Assumption	59
3.1.3	Notation	62
3.2	Yoking Proof by A. Juels	62
3.2.1	The Original Yoking Proof	62
3.2.2	Replay Attack Against Original Yoking Proof	63
3.2.3	Yoking Proof with Time Stamps	65
3.2.4	The Modified Yoking Proof	66
3.3	Grouping Proof	68
3.3.1	Grouping Proof by J. Saito and K. Sakura	68
3.3.2	Grouping Proof Inspired by Piramuthu	70
3.3.3	“Generalized Yoking-Proofs” and “Anonymous Yoking” for a Group of RFID Tags	71
3.4	Conclusion	72
4	Reading Order Independent Yoking Grouping Proof Protocol	77
4.1	Reading Order Penalty	78
4.2	Reading Order Independence Protocol	79
4.3	Comparison of Protocols	82
4.3.1	Reading Order Independent Operations	82
4.3.2	Length of Grouping Proof	82
4.3.3	Efficiency	83
4.3.4	Privacy	84
4.4	Conclusion	84
5	Select-response Grouping Proof Protocol	87
5.1	Introduction	87
5.1.1	Problems of “Yoking Proof” Protocols	88
5.2	Select Response Grouping Proof	90
5.2.1	New Idea of Grouping Proof	90

5.3	Protocol Design	94
5.3.1	Protocol Description: SRP with querying sequence by demand	94
5.3.2	Design Explanation	97
5.3.3	Practical Arrangement of the Group Identifier (GID) in the UID Format of RFID	98
5.4	Protocol Analysis	100
5.4.1	Security	100
5.5	Conclusion	101
6	Subgrouped Frameworks for Grouping Proof Protocols	103
6.1	Introduction	104
6.1.1	Communication Errors in RFID System	104
6.1.2	The Idea of Subgroups	104
6.2	Anti-collision Algorithms	105
6.2.1	Binary Tree Algorithms	107
6.2.2	Dynamic Binary Tree Algorithm	108
6.3	Subgrouping Model of Grouping Proof Protocol	110
6.3.1	Dividing Subgroups	110
6.3.2	Verification of the Whole Group	112
6.3.3	Verification in Subgroups	114
6.3.4	Quick Retrieval of the Identified Subgroups	116
6.4	Analysis of the Subgrouping Protocol	116
6.4.1	Analysis of the Grouping Proof Subgrouping Protocol	117
6.4.2	Security Analysis of the Quick Retrieval	118
6.5	Conclusion	118
7	Group Authentication of RFID Tags	
	Using Bit-Collision Patterns	121
7.1	Introduction	121

7.2	Pseudo-Simultaneousness of the Previous Grouping Proofs . . .	124
7.3	Bit Collision	126
7.3.1	Compatibility of Current Standard	126
7.3.2	Current Application of Bit-collision	127
7.4	Using Bit-Collision Patterns for Group Authentication	128
7.4.1	Group-Authentication Protocol	129
7.4.2	Constructing S and f_2	135
7.5	Security Analysis	139
7.6	Conclusion	148
8	Conclusion and Future Work	151
8.1	Security Analysis in this Thesis	151
8.1.1	Computer Aided Analysis	151
8.1.2	Theoretical Formal Analysis	153
8.1.3	Attack-Countermeasure Design Analysis	154
8.2	Future Works and Application of HB protocols	155
8.3	Future Works and Applications of Grouping Proof	155
8.4	Future of the RFID Protocols	157
	Abbreviation Glossary	158
	Bibliography	159

List of Figures

1.1	RFID System	15
2.1	A Round of HB ⁺ Protocol	42
2.2	A Round of HB-MP' Protocol	44
2.3	A Successful Active Attack against HB-MP' Protocol	46
2.4	The <i>i</i> th Round of HB-MP Protocol	48
2.5	The <i>i</i> th Round of the Improved HB-MP Protocol	50
2.6	The <i>i</i> th Round of the Abstract HB-MP ⁺ Protocol	52
2.7	Assumed Man-in-the-middle Attack on HB-MP ⁺ Protocol	55
3.1	Yoking Proof for RFID Tags	62
3.2	Replay Attack Against Original Yoking Proof	64
3.3	Yoking Proof Using Time Stamps	66
3.4	Yoking Proof Modified by S. Piramuthu	67
3.5	Grouping Proof by J. Saito and K. Sakura	69
3.6	Grouping Proof Based on S. Piramuthu	70
3.7	Generalized Yoking Proof Based on L. Bolotnny's Algorithm	75
3.8	Anonymous Yoking Proof based on L. Bolotnny's Algorithm	76
4.1	Reading Order Independent Grouping Proof for RFID Tags with Pallet Tag	80
4.2	Reading Order Independent Grouping Proof for RFID Tags: Reader with Computation Power	86
5.1	The Concept of SRP with Querying Sequence by Batch	92
5.2	The Concept of SRP with Querying Sequenceby demand	94
5.3	Select-Response Grouping Proof Scheme for RFID Tags	95

5.4	Unique Identifier of the ISO 18000	100
5.5	Unique Identifier of the ISO 15963	100
5.6	Modified the UID Format of ISO 18000-6	101
5.7	Modified the UID Format of ISO 15693	101
6.1	Collision Bits Identified According to Manchester Coding . . .	107
6.2	An Example of a Binary Tree Anti-collision Algorithm	108
6.3	Redundant Bits in the Binary Tree Anti-collision Algorithm .	108
6.4	An Example of a Dynamic Binary Tree Anti-collision Algorithm	109
6.5	The Architecture of the Subgrouping Grouping Proof Model .	112
6.6	Grouping Proof inside Subgroups	114
7.1	Constructing bit-collision patterns with Manchester code . . .	126
7.2	Description of the proposed group-authentication protocol . .	130
7.3	Comparative attack probability when the attacker replaces a single tag if the length of S and s_i is 96 (a realistic response length for both EPC (ISO 18000-6C) and ISO 15693/18000-3 tokens).	143
7.4	Comparative attack probability when the attacker replaces a single tag if the size of the groups stays constant and the number of collisions per tag increases ($n = 8$).	145
7.5	Comparative attack probability if the attacker replaces multiple tags in the same group ($n = 16, c = 2$).	148

Chapter 1

Introduction

The price of the Radio-Frequency Identification (RFID) tags promises to drop to the range of \$0.05 per unit in the next several years. The challenge in providing security for low-cost RFID tags is a big challenge because that the low cost limits the computation resource on the tags, so that the traditional cryptographic service is too heavy for those tags. The main contribution of this thesis is to propose several interesting protocols which deal with some specific applications of RFID systems. Before explaining these protocols, I would like to give a brief introduction on the RFID technology and its applications in this introductory chapter.

In section 1.1, I begin my introduction from the architecture of the RFID system, and then in section 1.2 I briefly explain the selection criterion of different RFID systems in different applications. In section 1.3, I introduce the current practical applications of RFID technology and I continue to introduce the potential applications in section 1.4. At the end of this chapter, I give a brief introduction of the security and privacy concerns in the protocol designs in RFID system, as the foundation for the coming chapters on the RFID protocols.

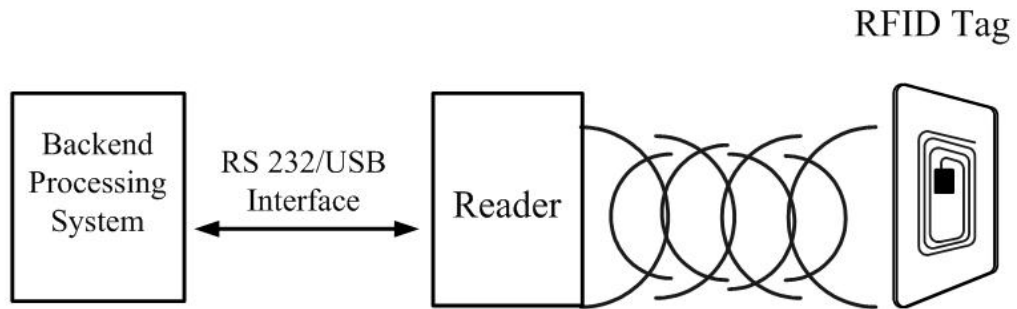


Figure 1.1: RFID System

1.1 RFID System

Radio Frequency Identification (RFID) is a technology that recently appears frequently on the newspaper, television, technical journals and even government white papers. As the name suggested, RFID is a general term used to describe a system that transmits the identity (in a secure way or insecure way) of an object using radio waves. Fundamentally, RFID is an enabling technology. It does not provide much value all by itself, but it does help in identifying and tracking goods and assets around the world. With the help of RFID, the whole supply chain, including the manufacturer, anti-counterfeiting supporter, the logistics providers and retail Point of Sale (POS) may all experience the fundamental changes of the way they operate their business. RFID also has numerous applications in transportation payments, library and access control, etc. There will be a detailed discussion on these current and potential applications of RFID in section 1.3.

1.1.1 Architecture of RFID Systems

A typical RFID system consists of a reader, tags, and a backend processing system(BPS), as shown in Fig.1.1.

An RFID tag is composed of an antenna, a wireless transceiver and an

encapsulating material [84]. These tags can be either active or passive. While the active tags have on-chip power, passive tags use the power induced by the magnetic field of the RFID reader. Thus passive tags are cheaper but with lower reading range (≤ 10 metres) and they are more sensitive to regulatory and environmental constraints, as compared to active tags. Via the embedded antenna, the tag receives power and exchanges messages with the reader. The radio frequency is modulated, demodulated and processed by the integrated circuit within the RFID tags.

An RFID reader consists of an antenna, transceiver and decoder, which sends periodic signals to inquire about any tag in its reading zone [84]. On receiving any signal from a tag it passes on that information to the backend processing system.

The backend processing system provides the means of processing and storing the data. Normally the backend processing system is a complicated enterprise-level system. backend processing system has interfaces to corporate business systems such as warehouse management systems, inventory systems, sales and billing systems, project systems, labour scheduling systems and enterprise resource planning systems. In most cases, a layer of software (often called middleware) will sit between the RFID system and the corporate systems, filtering and smoothing RFID data as well as passing relevant business data on to the corporate systems. Not all RFID data are relevant to enterprise applications. Middleware may be used to place RFID data in the context of the business setting and may be used to identify the relevant events and require passing on to a company's enterprise applications.

1.2 Selection Criteria for RFID Systems

Each of the RFID systems with the different frequencies and coupling methods has its own properties. The technical parameters of these systems are optimised for various fields of applications, such as ticketing, logistics, industrial automation and access control. The technical requirements of these applications often overlap, which means that the clear classification of suitable systems is not a simple matter. Besides of the frequencies and coupling methods, there are also other parameters like security, privacy and computing power, etc, need to be considered. It is difficult even for a specialist to retain an overview of the range of RFID systems currently on offer [26]. Therefore, it is not always easy for users to select the system best suited to their needs. In this section, I introduce some points for consideration when selecting RFID systems.

1.2.1 Operating Frequency

The specific absorption rate for water or non-conductive substances is lower by a factor of 100,000 at 100 kHz than it is at 1 GHz [20]. Therefore, comparing to the microwave, the low frequency waves are experiencing little absorption or damping when they go through liquid. Lower frequency systems are primarily used due to the better penetration of objects. An example of this is the bolus, a tag placed in the omasum of cattle [26], which can be read from outside at a reading frequency of less than 135 kHz. However, It also needs to be noticed that for some animal tracking systems, it is desirable to identify animals from a longer distance. Since the low frequency RFID systems do not offer enough reading range, so the higher frequency tags (some even with active power) are also used. Of course the tags are not inside animal body, they are

usually attached on the animal's fur. One example of such applications is that an agriculture technology company called TekVet works with IBM, utilizes the active 418 MHz RFID tags, sensors that monitor an animal's internal temperature and transceivers to transmit each tag's unique ID, as well as the animal's body temperature [4]. Microwave systems have a significantly higher range than inductive systems, typically 2-15 metres [26]. However, in contrast to inductive systems, microwave systems require an additional backup battery. The transmission power of the reader is generally insufficient to supply enough power for the operation of the tag. However, for the backscatter mechanism, typical ranges of 3 metres can now be achieved using passive (battery-free) backscatter tags, while ranges of 15 metres [73] and above can even be achieved using the active (battery-supported) backscatter tags. The battery of an active tag, however, never provides the power for data transmission between tag and reader, but serves exclusively to supply the microchip and for keeping the stored data. The power of the electromagnetic field received from the reader is the only used for the data transmission between the tags and readers [62].

1.2.2 Interference Sensitivity

Another important selection criterion is the sensitivity of the RFID signal to the electromagnetic interference fields. Metal will reflect the radio wave and affect the reading ability of inductive system. In the scenarios like the car assembly lines [20], Inductive tags have a significant disadvantage here because the reading rate will be greatly affected by the metal parts around. Microwave systems have therefore particularly established themselves in the production lines and painting systems of the automotive industry.

1.2.3 Practical Reading Range

The required range of an application is dependent upon several factors, for example, the positional accuracy of the tag; the minimum distance between tags nearby in practical operation; the speed of the tags in the reading zone of the reader. In the public transport tickets, the requirement of pass speed (number of tags passed in one minute) of the tag in the reading zone is low, comparing to the assembly line in the factory. Since the ticket is carried to the reading zone by hand. The minimum distance between several tags in this case should be smaller with the distance between two passengers entering the access control site. For such systems there is an optimal range of 0-10 cm [26]. A greater reading range would only give rise to problems in this case, since several passengers' tickets might be detected by the reader simultaneously. This would make it impossible to reliably allocate the ticket to the correct passenger.

Different vehicle models of varying dimensions are often constructed simultaneously on the production lines of the automotive industry. Thus great variations in the distance between the tag on the vehicle and the reader are pre-programmed. The write/read distance of the RFID system used must therefore be designed for the maximum required range. The distance between the tags must be such that only one tag is ever within the reading zone of the reader at a time. In this situation, microwave systems in which the field has a directional beam have clear advantages over the broad, non-directional fields of inductively coupled systems [20]. The speed of tags, relative to readers, together with the maximum write/read distance, determines the time spent in the reader's reading zone. For the identification of vehicles, the required range of the RFID system is designed such that at the maximum vehicle speed the

length of time spent in the reading zone is sufficient for the transmission of the required data [80].

1.2.4 Security Requirements

Security requirements should be analysed by security experts thoroughly before any planned RFID application, i.e. cryptography applied, authentication protocol and tamper-resist mechanism, should be assessed very precisely to prevent any horrible surprise in the implementation and usage phase [66]. For this purpose, the incentive that the system represents to a potential attacker as a means of stealing money or material goods by fraud manipulation should be evaluated. In order to be able to assess the attraction, the applications can be divided into two groups: Internal applications and public applications connected with money and material goods.

In the industrial internal applications, let us once again take an assembly line in the automotive industry as a typical example. Only the authorised or internal people have access to this RFID system, so the group of potential attackers remains reasonably small [72]. A malicious attack on the system by the alteration or tamper with the data on the tags or readers could cause a critical malfunction in the operation. However, as most of the companies have installed Closed-Circuit Television (CCTV), the personnel involved in such applications are under surveillance nearly all the time. Also such attack is rarely seen as beneficial to the attackers, unless for some market opponents. The probability of such an attack is very low. Thus a cheap low-end system without security logic can be used to save the operational cost.

For the public applications, a typical example is the e-ticket system used in public transport. In such a system, the tickets are in the form of RFID card

or fob, which are accessible to anyone and become an easy target of theft [21]. The group of potential attackers can be enormous. A successful attack on such a system could inflict large-scale financial loss to the public transport company and cause huge reputation damage. For such applications, a high-end tag with authentication and encryption procedures is necessary, thus the contactless smart card is used. For applications with maximum security requirements, such as debit cards, electronic purse, entrance tickets, only tags with microprocessors and security measures meet the minimum requirement of security.

1.2.5 Memory Capacity

Memory capacity plays an important role in certain applications. Since the chip size of the tag is primarily determined by its memory capacity, adding more memory would significantly increase the cost of tag. Therefore, permanently read-only tags are used in price-sensitive mass applications with a low local information requirement [23]. Only the identity of an object or some other crucial information can be put within such a tag. Further data is stored in the central database of the backend processing system. If data is to be written back to the tag, a tag with Electrically Erasable Programmable Read-Only Memory (EEPROM) or Random-Access Memory (RAM) technology is required [26]. EEPROM memories are primarily found in inductively coupled systems. Memory capacities of 16 bytes to 8 Kbytes [25] are available. Static Random Access Memory (SRAM) devices need a battery backup, are predominantly used in microwave systems. The memory capacities of such tags range from 256 bytes to 64 Kbytes [25]. The RFID tags with the EEPROM and SRAM are more expensive than the basic tags with the read-only memory,

so they are only used in certain applications with special requirement.

1.3 Practical Applications

There has been increasing popularity of RFID systems in recent years. In this section, I give a brief introduction to the current application of RFID technology.

1.3.1 Transportation Payments

The best example of the successful RFID applications is the contactless smart cards used as electronic tickets for public transport [19]. Not long ago it was inconceivable that tens of millions of contactless tickets would now be in use in the transportation payment system. The possible fields of application for contactless identification systems have also multiplied recently [21]. The public transportation has been using the conventional paper tickets for the last several decades. The shortcomings of the paper tickets are becoming more and more realised by the transportation companies and the policy makers. Here I only mention some but not all the drawbacks, such as disposed paper tickets are a huge waste of resource; the cost of producing fraud-proof tickets is expensive; paper tickets are not easy to be examined by the automatic machine and make complicated fare plans difficult to be shown and the paper ticket is weak to keep for a long time for long-term(monthly) use.

The RFID technology has the exact properties to overcome these shortcomings of the paper tickets. What is more, the passengers no longer need to bring some cash for the tickets, RFID tickets can be loaded with certain amounts of money, passengers no longer need to carry the correct change. There is no need to queue and waste time to buy the ticket every time. Weekly or monthly

tickets can begin on any day of the week or month and they are much stronger to be held for a long time[21]. The transport company can also enjoy the benefits from the RFID tickets such as the elimination of the daily cash calculation, the reduced operating and maintenance costs of sales dispensers. Improved security and easier implemented automatic ticket checking system. What is more, Because data is obtained automatically in every transaction in the electronic fare management systems, it is possible to calculate the travel statistics and help the company to make decisions according to the market.

For the reason stated, the transportation payment becomes the most successful area to the RFID applications.

1.3.2 Aviation and Railway

Aviation industry is facing big challenge from both the increasing passenger number and enhanced security controls which is required under the anti-terrorist demand [53]. To deal with huge numbers of passengers and luggage in an efficient way drives the aviation industry to the RFID technology. Comparing to the barcode labels currently used in the majority of airport, the RFID tags do not require the direct line of sight so that they are much easier to be read[57]. With this advantage , RFID technology has become a powerful and cost-effective resource for airport security and facilities management[19]. The RFID tags are implemented as passenger tickets and baggage labels. With the help of RFID, the people and baggage can be identified if the right bag is with the right plane at any point of transportation. The scanning of RFID tags inside the luggage label and boarding pass on the entrance and exit of the aircraft can greatly enhance control and improve security management.

In the railway industry, the railway operators always want a better monitoring system of their wagons, which are spreading across the railway network and dynamically changing their positions. With the RFID tagged on the goods and transportation units, the asset can be identified easily during the transportation process, which will greatly improve the asset management. In China, the biggest RFID company has its revenue mainly from the railway business [7].

Besides aviation and railway, the RFID technology also has wide application in other transportation systems. RFID tags effectively links the object and its related data together, and the asset management is facilitated by the fast and automatic reading process. The widespread deployment of RFID technology would therefore create a much smarter transportation management system.

1.3.3 Access Control

Nowadays, electronic access control applications are often based on RFID tokens, e.g, cards, fobs or badges. The wireless property of these tokens renders control more easily because people do not have to take out their tokens from the wallets or pockets. Comparing to the traditional keys and contact smart cards, damages to readers and vandalism are reduced [68]. Beyond the primary goal of access control, the other advantage of RFID systematic controls is to determine, in real time, the people who are present within a given area. It can be for safety reasons, e.g., when a building must be evacuated[68].

A particular example of access control can be found in the automobiles domain. Early applications were keyless entry: the driver opens and closes the car using a key fob that contains an active RFID tag. Passive entry

systems appeared recently, the driver, who is carrying a passive RFID tag, simply approaches the car and the door automatically unlocks itself [15]. Still further, today, many car keys have an RFID device integrated into them which activates the fuel injection system, thus preventing the vehicle from starting if the electronic tag is not present. The tag is activated when the key is inserted in the starting device and receives a challenge from the reader. Sending a correct response to this challenge is required to get access to the vehicle. In some cases, the (physical) ignition key is no longer required. The driver is just able to press a button “start” to ignite the car. The car has detected that the driver has an RFID tag embedded into a card that stays in his pocket[54].

1.3.4 Library

Management of library of increasing volume of publications and various format has give pressure to the libraries. Traditional library management uses the magnetic tripes and sensors to manage the loan service, which is slow in its nature and the magnetic stripe will not always last for a long time, which increase the book lost and reading failure. With the help of RFID technolgy, which simple attach the RFID tag to the asset in similar ways, depending on the material, paper, plastics, etc[44]. The RFID offers the faster and more accurate scanning of information. The advantages offers simple and easy way to manage the loan and return of the material. It also helps the ‘shelf reading’, which means that the self is installed with the reader and it can actively query the books on it. Thus the library can have a real-time inventory of all the books available on the shelves.

1.3.5 E-Passports

After 9/11 terrorist attacks, the major western countries begin to use RFID and biometric technologies in their boundary control. [33]. The United States government has mandated adoption of biometrically-enabled passports by the nations whose citizen can travel to the US within its Visa-Waiver Program by October 2006 [17]. The United States also issue its own citizen with the passport carrying biometric information by the end of 2005. The International Civil Aviation Organization (ICAO) has published the international standard for the e-passport implementation [41]. Their guidelines which are detailed in ICAO Document 9303 [1], call for incorporation of RFID chips into passports. Thus the passport becomes exactly an RFID tags which can conduct the authentication automatically. With the efforts of different countries, the e-passports embedded with RFID chips, will have a widespread in applications within a couple of years. [17].

1.3.6 Anti-Counterfeiting

Fake products are still rampant in many countries, which imposes huge loss to the companies that make the authenticate goods every year. A good news to them is that the RFID can help the anti-counterfeiting for them [75]. Because the RFID tag is unique to each item, so the RFID tags attached to the specific item can have all the information required to authenticate the item. Briefly, the authentication works like this: products are attached with the RFID tags and when it passes the reading area of readers, it is automatically queried and authentication protocol is run to verify its identity. For an authenticate tag we are sure that the item is also authentic. However, if an attacker can get the secret of the tag and store it in another chip, the attacker has effectively made a

clone of the original tag that can make a fake item appears authenticate. Thus the secret should be well protected to make sure it is impossible to derive the tag secrets by active or passive attacks [79]. Thus secure protocols are needed and the tags much be equipped with some hardware security mechanism to prevent the attackers probe the secrets from the tags.

1.4 Potential Application

1.4.1 Supply Chains

The biggest potential RFID application lies in the key role played by the ultra high-frequency (UHF) passive RFID tags in the supply chain [10] in the future. These tags are expected to offer the item tracking service in the global supply chain. Once all goods are attached with RFID tags, their status and locations can be tracked automatically by the RFID readers, which would offer complete inventory visibility and improved management efficiency. The current mature product identification technology in the supply chain is the bar code technology. Comparing the RFID technology with the bar code technology, one can find the following advantages and disadvantages, these contents are partly from the website [2].

Advantages of RFID tags

1. Easier to Read. RFID readers do not require a direct line of sight to read as barcode. Also RFID tags can also be read at much greater distances as far as 300 feet. The barcode has only no more than 15 feet reading distance. The RFID tags also have a much greater reading speed while reading barcodes is much more time-consuming due to the direct line of sight is needed[2].

2. Better Durability Direct line of sight requires the printed barcode must be exposed on the outside of the product, which makes it easy to worn-out. RFID has the electronic components inside a plastic cover, thus they are typically more rugged [58]. RFID tags can also be implanted within the product itself, which gives it even greater ruggedness.

3. Data Operation Barcodes is printed and thus have no write capability. If the application requires, RFID tags can be read/write devices. The RFID tags also offers greater data storage than the barcode, which explains why RFID can offer item-level product tracking but barcode can only offer catalogue-level tracking.

Disadvantages of RFID tags

RFID tags also have some disadvantages. First of all, RFID tags are expensive compared with bar code. Secondly the RFID tags are bulkier, it has a plastic cover and the embedded chip. However,now the embedding of electronic components can be even printed inside paper, which is desired in certain application. The last disadvantage is that an RFID tag is prone to electrical damage due to environmental conditions. The tags may require redesign and become more costly under extreme environmental conditions [72].

Challenge of Implementation

In June 2003, Walmart announced that it will demand the supplier have the RFID tags attached to their products[69]. In October of the same year, the EPCglobal organization was born from the previous Auto-ID center, which wants the leadership of the standard. As I have mentioned in the previous text, the RFID clearly has attractive advantages over barcode. However, until now,

supply chain application has been regarded as a disappointment in comparison with the expectations in 2003 [61]. What is the difficulty for the wide adoption of RFID technology in industry? Wu et al. have proposed these challenges in [82]:

Technology Challenges

To cut the cost of bulk amount of the RFID tags used in supply chain, these tags needs to be passive RFID, which relies on its antenna to receive radio waves emitted by the reader to power up and transmit data back. Therefore, the antenna design is crucial [82]. In the supply chain applications, radio waves will be reflected and refracted differently by the different materials. For example, the UHF radio waves will be affected heavily by the liquid refraction and mental reflection. In both cases, there will be signal strength degradation and interference of signal quality. Consumer products such as shampoo, juice and canned goods with give the challenge of the antenna design for the tags. [78].

Reading one tag and guarantee successful reading rate is not difficult, however in supply chain RFID applications, hundreds of tags may be read simultaneously, those radio signals would cause collision interference to the reader. Also when a large number of tags are being read simultaneously, it is difficult to identify those tags which have failed to be read. It is a serious problem in any tracking system if 100% accuracy cannot be achieved. Later in this thesis, I will introduce several ‘grouping proof’ RFID protocols to track on the multi-tags reading problems [82].

Standard challenges

There are three major advantages of developing international standards for RFID supply chain. First of all, one standard will ensure inter-operability

among tags and readers across national boundaries [26]. Secondly, with one standard, the demand for RFID components and equipment will be high, which can bring the cost down. Finally, an internationally accepted RFID standard will encourage the growth of the worldwide supply chain RFID system.

Currently, both EPCglobal and International Standards Organization (ISO) develop international standards for UHF RFID tags. EPCglobal released its EPC class 1 G2 standard [35] at the end of 2004, and the ISO released its 18000-6 [6] in August of 2004. Both standards are still evolving and not completely compatible with each other. The lack of a complete and unified RFID standard has caused many companies being afraid of making a commitment that might the whole investment worthless in the future.

To make things more complicated, radio spectrum allocation for RFID use are not unified among nations. Some UHF spectrum are auctioned to mobile service in certain countries [82]. For a unified spectrum in the international standard, some may need to buy that portion of spectrum back, if possible [67].

Patent challenges

The international standards will be embedded with certain intellectual properties (IPs) and technologies from all vendors in the industry. The EPCglobal's intellectual properties policy demands that the companies holding IPs that is embedded to EPC standards must declare the IPs and indicate whether they are available to other vendors on a royalty-free or a reasonable and non-discriminatory royalty-bearing basis [82]. Although the EPC Gen2 standard has already been formally released, the payments for IPs involved are still unclear. Potential vendors concerned about paying high cost is another obstacle holding back the development of RFID systems.

Infrastructure Required

To properly address the infrastructure factor in the RFID application, let us imagine that tags and readers are free now. What would the manufacturers do and what would the whole supply chain be now? Will there be a fundamental update to the whole industry soon? This often helps us evaluate the importance of the infrastructure which is behind scene but makes the tags working. Review in [78] pointed out that the number of companies that adopt RFID tags now, in case that RFID tags are free, would not be a lot bigger because of the lack of infrastructures.

Current industry supply chains are already quite long and cross international borders. Ideally, the flow of RFID attached goods should be tracked in every stage of the chain. But in order to achieve this goal, an entire international RFID infrastructure must first be established [48]. This will allow for the international collection of real-time information from anywhere in the global supply chain. The supply chain involves many manufacturing companies, transportation companies, and many sea ports and airports in different countries. Therefore, to build the entire RFID infrastructure available to track every tagged item from the beginning to the end of the supply chain is really a great challenge [82].

1.4.2 Pervasive Computation

Pervasive computing applications employs RFID as a basic part of infrastructure, making them giving information to the attached objects and conducts computation [60]. Thus the pervasive computation will have tremendous numbers of tagged entities, makes the tag number very high in a defined area. The interaction among the RFID tagged items should fully automatic without any invention of human beings and the information service is done. For example,

the refrigerator communicate with the milk bottles so it knows the milk is running out. Then the refrigerator automatically order a milk delivery next day without attention from the owner. Such service could convert simple identifications into higher-level information interactions that can be used for applications smarter than imagination [70].

Comparing to the traditional Internet client-server style computing? the pervasive computing network of RFID have put a great proportion of its functionality the network core to the edge [76]; Yet, at the present time, these fundamental change are still inadequately understood as neither deployments nor practical experience of their performance are learned well. [76]. Without doubt, further research in this area is required.

1.5 Security and Privacy Issues on RFID Protocol Design

As the I have described on the previous sections, the RFID tags are already widespread in many applications, and they will be the fundamental part of the pervasive computation in the future. It is vitally important that the security and privacy issues are thoroughly researched and well understood.

1.5.1 Security

For the security concerns, RFID tags are the easy target to various logical and physical attacks. The attackers want to reveal the secret of the tags and make a cloned one. The information transmitted between tags and readers should be well protected both on the confidentiality and integrity. The system also needs to be robust against the Denial of Service (DoS) attacks. However, the RFID protocol designers cannot depend on the traditional cryptographic

algorithms to protect the system since these algorithms are too heavy for the RFID tags. I will give a brief view of the limitation of the tag's resource at the end of this section.

1.5.2 Privacy

Privacy violations in the use of RFID can be broadly categorised in two classes: tracking, track the individuals using RFID tags associated with them; and inventorying, secretly read personal or intimate information stored in RFID tags without the authorisation of its owner [9]. Both violations are enabled by invisible reading RFID tags. The property of current RFID tags will automatically reply to the query signals it understands. Thus in the reading zone of certain readers, tags would broadcast their unique serial numbers to any party interested. Some tags, i.g. EPC tags, carries the information about the item to which they are attached. So the invisible query can leak some important information like the medicine carried with that person so the privacy of the health condition is leaked. In the future supply chain applications, if the tags are not removed or disabled at the point of sale, they can leak the private information with the consumer carries it. Such threats have been identified from early on in the use of RFID in the context of retail and have caused significant concerns among consumers [52].

1.5.3 Computation Limits of RFID Tags

As I have mentioned in the previous sections, the RFID is expected to replace the barcode in the near future. However, the security and privacy issues arouse for such pervasive implementation of RFID technology. Generally the RFID tags used are expected to be small and cheap. These tags are passive so they can only obtain very limited and often not reliable power from the radio

wave. What is more, In some applications, the pass speed must be guaranteed, which leave very limited processing time for the RFID tags to run a protocol session. All the facts mentioned above significantly limit the computing power of the RFID tags. The limited computing resource imposes great limitation for the cheap RFID tags to implement cryptographic functions. Since the security and privacy problems of RFID cannot be solved by conventional cryptographic services, the theoretical problems still exist and they are hot topics of research at this moment [12].

The RFID tags talked in this thesis, are supposed to be used in the future supply chain and pervasive computing schemes. These tags do not have adequate computation power for the conventional cryptographic computation. Current EPC tags suppose to have a couple of thousand gates, in which only a few hundred can be used as the security services [9]. Some people put the hope of moore's law that the computation ability of the RFID tags will be enhanced to an adequate level in the next couple of decays. But pricing pressure will still be a big counter force impeding the increase of computational ability. As long as cheap, insecure tags exists, that most retailers and manufacturers will choose the cheaper one instead of the expensive tags with cryptographic functions and good security features. Only price difference of several cents would impose a big cost change for the vendors, so the best way to cope with the situation is to develop appropriate low weight protocols for specific applications.

Considering the fact that the cryptography is the fundamental parts of the security services used in the current applications. The protocol designer for RFID applications are facing a big challenge to achieve the design goal under the low computation resource environment. On the other hand, the

lack of cryptography in basic tags poses intriguing research challenges. In recent years, researchers have contributed many publication of lightweight technical approaches to the problems of security and privacy. I will give a short introduction of these approaches as a good start to the protocols proposed in this thesis.

1.6 Storyline of Protocol Introduction

In this thesis, firstly I introduce the HB style protocols, which use a special security mechanism to counter the limited computing resource problems. The HB style protocols are a family of similar protocols which does not rely on any cryptographic algorithm and the authentication of a tag is based the chances of its correct replies in several rounds. The HB style protocols are invented very specifically to address the property of the RFID system. I make contribution based on one of these protocols. Then I introduce another security scheme called grouping proofs, which is also very specific to RFID system. Grouping proofs are designed to prove that a group of tagged item are present simultaneously. My research is firstly inspired by the previous work, found that the common weakness (reading order dependence) of the already proposed protocols. Then I have the innovation of the fundamental way of the grouping proof. Instead of using the ‘Yoking’ mechanism, I used the select-response mechanism to conduct the grouping proof. Then I proposed a framework to cope with the groups of large amount of tags. By slightly altering the anti-collision algorithm, the protocol divides the big group is divided into smaller subgroups without adding computation loads. Finally, inspired by the collisions in the anti-collision algorithm, I proposed the collision pattern as the grouping proof, which authenticates the whole group in a single

challenge-response round, gives the real simultaneousness and it is fast enough for practical applications nowadays.

Chapter 2

HB-MP⁺ Protocol

2.1 Introduction

For the reason we have stated in the previous chapter, the computing power and resource of RFID tags are expected to remain extremely limited to reduce the cost of RFID implementation. However, the security and privacy features needed in RFID systems are almost the same as needed in other computing systems.

Researchers have been struggling to find some mechanisms to meet the security demand without common cryptographic functions implemented on the tags. Some of them are very initiative and interesting, among them, HB RFID protocols and the later proposed HB protocol families are one typical example of these protocols. In this thesis, I call all these protocols HB style protocols for short.

In 2001, Hopper and Blum [63] proposed an authentication protocol to meet the demand of human authentication, known as the HB protocol. The HB protocol relies on the computation hardness of the Learning Parity with Noise (LPN) problem, and uses only dot products of binary vectors and a random noise bit, so it is very lightweight. In 2005, Juels and Weis [8] adopted this HB protocol into RFID systems because of the similarity between human

and tags. The authors also presented an active attack on the HB protocol, and they proposed an enhanced version called the HB⁺ protocol. Later in the same year, Katz and Shin [36] proved the parallel concurrent security property of the HB and HB⁺ protocols, but their proofs only imply meaningful security for $\varepsilon < 1/4$. Follow on work done by Katz and Smith [43] extend the proofs of security for the full HB and HB⁺ protocols for arbitrary $\varepsilon < 1/2$. Unfortunately these two protocols were shown by Gilbert et al. [31] to be vulnerable to certain man-in-the-middle attacks. In 2006, Bringer et al. [37] proposed an enhancement of HB⁺ called HB⁺⁺. Piramuthu [74] had a survey of the HB-family protocols and proposed another modification of the HB⁺ protocol, in which the bit-wise rotations are varied for each round and the message flow is simplified (saving one bit per round). In 2007, Duc and Kim presented a variation of HB⁺ protocol called HB*, which is resistant to Gilbert et al.’s attack [24], But Piramuthu [65] broke HB* and proposed his modified protocol. In 2008, Gilbert et al. [28] proposed their thorough analysis on the HB protocol families and proposed two new protocols called *RANDOM*-HB[#] and HB[#]: *RANDOM*-HB[#] avoids many practical drawbacks of HB⁺, remains provably resistant to attacks in the model of Juels and Weis [8], and is also provably resistant to a broader class of active attacks. However, *RANDOM*-HB[#] is required to store two random matrices, which make the storage costs insurmountable to the tags. HB[#] enhanced *RANDOM*-HB[#] by using Toeplitz matrices [46][47] to improve the performance. At the time when this paper is written, the latest paper on improvement of HB⁺ is the “Trusted-HB” protocol, which uses a LFSR-based Toeplitz hashing [46] to enhance the security of HB⁺.

In early 2007, Munilla and Peinado [38] proposed a prominent protocol

called HB-MP. The HB-MP protocol has a fresh way of exchanging messages and improved attack resistance whilst retaining the simplicity of the HB family. However, I have found this protocol is still vulnerable to a replay attack and the countermeasure is discussed within this chapter.

The remaining part of this chapter is organised as follows. After the introduction, the prerequisites, namely, the binary inner product and the LNP problem are introduced. In section 2.3, the HB protocol family is analysed in detail, this prepares the discussion in subsequent sections. In section 2.4, I introduce an early step of the HB-MP protocol and a man-in-the-middle attack that effectively breaks it. Section 2.5 introduces the HB-MP protocol which had some countermeasures added to defend against this attack. It furthermore describes the vulnerability of HB-MP protocol which could enable the attack mentioned. Section 2.6 proposes an improved HB-MP protocol to resist such attack. Section 2.7 proposes a general form of HB-MP⁺ protocol. Section 2.8 gives security and performance analysis of the HB-MP⁺ protocol.

2.2 Prerequisites

2.2.1 Binary Inner Product

The HB protocol makes use of binary inner product of two k -bit numbers. I briefly review the concept including its property. Given two k -bit number $a = (a_0a_1 \cdots a_{k-1})_2$ and $b = (b_0b_1 \cdots b_{k-1})_2$, the binary inner product of a and b , denoted as $a \cdot b$ is computed as follows:

$$a \cdot b = (a_0 \wedge b_0) \oplus (a_1 \wedge b_1) \oplus \cdots \oplus (a_{k-1} \wedge b_{k-1})$$

2.2.2 LPN Problems

All the protocols of the HB family are based on the conjectured hardness of the Learning Parity in the Presence of Noise, or LPN problem. Here I offer the definition of the LPN problem:

LPN Problem: The LPN problem with security parameters q, k, η with $\eta \in (0, 1/2)$ is defined as follows: given a random $q \times k$ binary matrix A , a random k -bit vector x , a vector v such that $|v| \leq \eta q$, and the product $z = A \cdot x \oplus v$, find a k -bit vector x' such that $|A \cdot x' \oplus z| \leq \eta q$, where $|v|$ denotes the Hamming weight of vector v .

The LPN problem is known to be NP-Hard [14]; currently no polynomial algorithm is known to solve the LPN problem. Hopper and Blum [63] and Juels and Weis [8] cited the BKW algorithm which is considered to be fastest in solving LPN problem. However in Gilbert et al. [28], the authors used the results from other researchers, conclude the former way of defining security parameters of LPN problem needs adjustment, as the BKW algorithm is improved significantly. For example, it was thought that a LPN problem has the length of the secret $k = 224$ and the noise level $\eta = 0.25$ could achieve around 80-bit security. Unfortunately Gilbert et al. [28] cited new research showing that, using the new BKW algorithm, $k = 224$ and $\eta = 0.25$ can only offer a security level no more than 52 bit. They conservatively proposed that $k = 512$ and $\eta = 0.25$ should provide a good security level.

2.3 HB⁺ Protocol

Since Juels and Weis [8] introduced the HB⁺ protocol, considerable research interests were generated and several protocols based on this protocols were

introduced in the previous section. The previous works, has been the works given various proofs on the properties of protocols, evolved in different directions. Some of them, for example, “Trusted-HB” protocol has gone so far from the original idea of HB protocol as to use the LFSR-based Toeplitz hashing to achieve provable security. Comparatively, HB-MP protocol is more HB⁺ protocol as the previous work. I also introduce the man-in-middle attack of the HB⁺ protocol, which also inspire the attack on the HB-MP protocol I found in this article. For reader’s convenience, I introduce the notation used in our article, which are consistent with that used in Munilla and Peinado’s paper [38].

k	length of the secret keys shared by the reader and the tag.
x, y	k bits secret keys shared by the reader and the tag.
a, b	random k -bits binary vectors.
v	noise bit; $v = 1$ with probability $\eta \in [0, 1/2]$.
\oplus	XOR operation.
$a \cdot x$	scalar product of vector a and x
q	Number of rounds in an authentication session

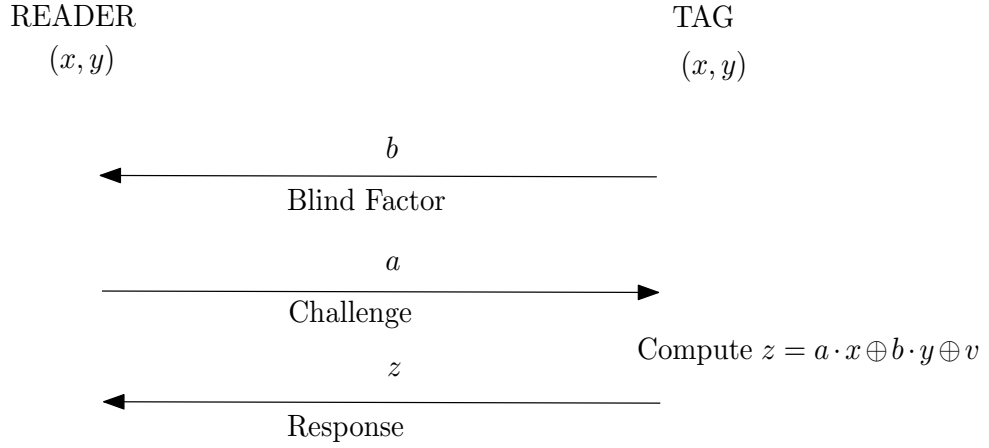
Table 2.1: Notations for HB⁺ Protocol

Step 1. The tag chooses at random a k -bit binary vector b , and sends it to the reader.

Step 2. The reader generates at random a challenge a and send it to the tag.

Step 3. The tag computes $z = a \cdot x \oplus b \cdot y \oplus v$ and sends z .

Step 4. The reader checks whether $z = a \cdot x \oplus b \cdot y$.



Check if $z = a \cdot x \oplus b \cdot y$

Figure 2.1: A Round of HB⁺ Protocol

The HB⁺ protocol runs q rounds and figure 2.1 shows one round. If the non-match rounds exceeds a threshold t , the tag is rejected and otherwise it will be accepted. People reading this protocol would naturally consider that the error reject rate can be fairly high and it is possible for a very lucky fake tag to be successfully authenticated. From probability theory, one can give the equations of false rejection rate P_{FR} , and false acceptance P_{FA} :

$$P_{FR} = \sum_{i=t+1}^q \binom{q}{i} \eta^i (1 - \eta)^{q-i} \quad (2.3.1)$$

$$P_{FA} = \sum_{i=0}^t \binom{q}{i} 2^{-q} \quad (2.3.2)$$

It is clear from the equations that both P_{FR} and P_{FA} are irrelevant to the lengths of the secret k , they are only relevant to q , t and η . In the original HB⁺ protocol, a threshold of $t = \eta q$ is suggested. However in Gilbert et al. [28], a table describing the relations of security parameters and error rates shows the default choice $t = \eta q$ gives an unacceptably high false rejection rate. For

example, when $q = 60$, $\eta = 0.25$, $k = 224$, P_{FR} can be as large as 0.43! It is hard to imagine any practical scenario where a probability higher than 1% of rejecting a legitimate tag could be tolerated.

Another obvious problem of implementing the HB^+ protocol is the transmission costs of the q rounds communication. Actually considering the high false rejection rate, a genuine tag might need to run over q rounds to get authenticated. In each round, a pack of three k -bit messages have to be transmitted. Gilbert et al. [28] gave some description of the transmission cost which shows the transmission cost of HB^+ protocol with previously proposed security parameters, $q = 60$, $\eta = 0.25$, $k = 224$, will have to transmit at least 26,984 bits of data just for a whole (q rounds) authentication. For the more secure HB^+ protocol with $k = 512$, there needs at least 61,500 bits of data to be transmitted. All this calculation is not including other necessary transmission overloads like the transmission time intervals and error-checking code attached. So HB^+ protocol is still impractical for current RFID systems.

2.4 The $HB-MP'$ Protocol and Its Weakness

2.4.1 The $HB-MP'$ Protocol

$HB-MP$ protocol is a variation of HB^+ protocol. There is an important step in developing the $HB-MP$ protocol from the HB^+ protocol called the $HB-MP'$ protocol. It is a direct modification of HB^+ but vulnerable to the man-in-middle attack proposed by Gilbert et al. [31]. The $HB-MP$ protocol is an enhanced version of $HB-MP'$ to resist such attack.

The protocol of $HB-MP'$ is composed of q rounds. One of which is depicted in Figure 2.2 and described as follows:

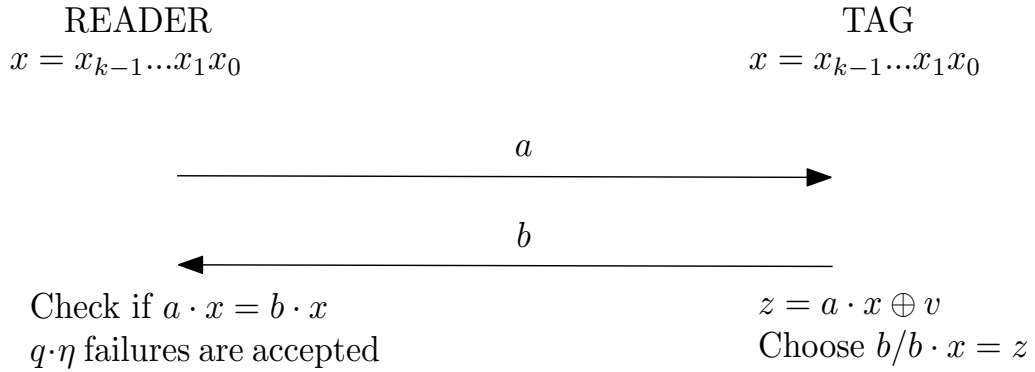


Figure 2.2: A Round of HB-MP' Protocol

Step 1. The reader chooses at random a k -bit binary vector a , and sends it to the tag.

Step 2. The tag computes z as follows: $z = a \cdot x \oplus v$ and looks for a k -bit binary vector b such that $b \cdot x = z$.

Step 3. The tag sends b to the reader.

Step 4. The reader checks whether $b \cdot x = a \cdot x$.

Munilla and Peinado [38] give a detailed explanation about finding x knowing the vectors a and b is at least as difficult as solving the LPN problem. The essential change made by the HB-MP' protocol is that the tag side takes the computation load of picking the response b . In their paper, they give a neat algorithm on picking b when $\eta = 0.25$:

Algorithm 1. Input: a, x . Output: b such that $b \cdot x = a \cdot x \oplus v$, where $v = 1$ with probability $1/4$.

Computes $z = a \cdot x$

Generates at random k -bit binary vector b

If $b \cdot x = z$

```

    Sends  $b$ 
else
    Generates and sends a new random  $k$ -bit vector  $b$ 
end

```

From the algorithm, one can know that the possibility that $b \cdot x \neq a \cdot x$ is $0.5 \times 0.5 = 0.25$, that means $\eta = 0.25$. If the b is checked n times before it is sent, then $\eta = 1/2^{n+1}$. In Algorithm 2, I give a general form of n times checking on b before sending it.

Algorithm 2. Input: a, x, n . Output: b such that $b \cdot x = a \cdot x \oplus v$, where $v = 1$ with probability $1/2^{n+1}$.

```

Computes  $z = a \cdot x$ 
While  $n \geq 1$ 
    Generates at random  $k$ -bit vector  $b$ 
    If  $b \cdot x = z$ 
        Break
     $n = n - 1$ 
end While
Sends  $b$ 
end

```

$n = 1$, $\eta = 0.25$ and $n = 2$, $\eta = 0.125$ are both practical and popular choices.

2.4.2 A Man-in-the middle Attack on the HB-MP' Protocol

HB-MP' is the prototype of HB-MP protocol but it is vulnerable to a man-in-the-middle attack similar to the one proposed by Gilbert et al. [31]. Munilla

and Peinado [38] mentioned this type of attack on the HB-MP' protocol, however, the attack was not described within their paper. Here I give an example of such an attack. It is reasonably assumed that the adversary is capable of

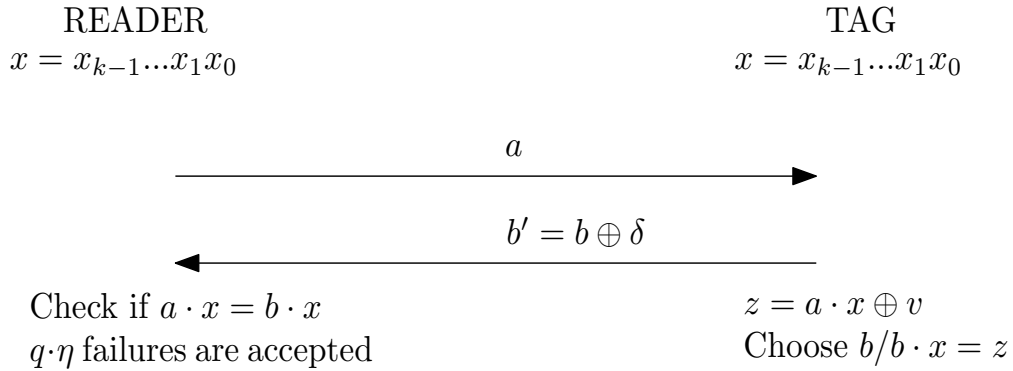


Figure 2.3: A Successful Active Attack against HB-MP' Protocol

manipulating challenges sent by a legitimate tag to a legitimate reader during the authentication procedure, checking whether this manipulation results (or not) in an authentication failure. The attack is illustrated in Figure 2.3 onto a single round of the HB⁺ protocol. The attacker chooses a constant k -bit vector δ and use it to perturb the response sent by a legitimate tag to the legitimate reader: $b' = b \oplus \delta$ for each of the q rounds of authentication. If the authentication process is successful, then it must be true that $\delta \cdot x = 0$ with overwhelming probability. If authentication doesn't succeed then $\delta \cdot x = 1$ with overwhelming probability.

I use the same δ in all q rounds of the protocol. Acceptance or rejection by the reader would reveal one bit of secret information x . To retrieve the k -bit secret x , it is enough to repeat the full protocol k times for linearly independent δ 's, and to solve the resulting system. Conveniently, the attacker can choose δ s with a single non-zero bit and this non-zero bit is different for each δ . Once

x has been derived, the attacker is able to impersonate the tag. Another side effect of the disclosure of x is that the privacy of the tag's identity is also compromised.

2.5 The HB-MP Protocol and Its Weakness

2.5.1 The HB-MP Protocol

The HB-MP protocol is an enhancement of HB-MP'. With the same notation of HB-MP', there are some more notations:

m	length of the message exchanged between the parties.
x, y	k bits secret keys shared by the reader and the tag.
xm	the m -bit binary vector consisting of the m least significant bits of x .
a, b	a, b are m -bit long in the following protocols
$Rot(p, u)$	the bitwise left rotate operator. The operand p is rotated u positions.

Table 2.2: More Notations for HB-MP Protocol

The protocol also runs q rounds to achieve one authentication, one of which, the i th round, is depicted in Figure 2.4 and described as follows:

Step 1. The reader chooses at random an m -bit binary vector a and sends it to the tag.

Step 2. The tag computes $x = Rot(x, y_i)$, where y_i is the i th bit of the key y .

Step 3. The tag computes z as follows:

$$z = a \cdot xm \oplus v \tag{2.5.1}$$

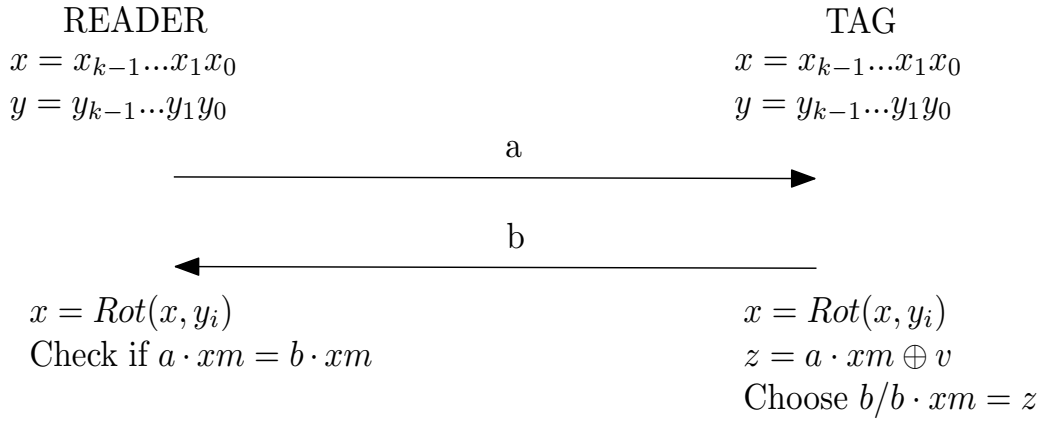


Figure 2.4: The i th Round of HB-MP Protocol

and looks for a k -bit binary vector b such that $b \cdot xm = z$.

Step 4. The tag sends b to the reader.

Step 5. The reader computes the x in the i th round as $x = Rot(x, y_i)$, where y_i is the i th bit of y .

Step 6. The reader checks if

$$a \cdot xm = b \cdot xm \tag{2.5.2}$$

After q rounds, the reader trusts the tag is legitimate if the failures are below $q \cdot \eta$ rounds.

2.5.2 A Man-in-the-middle Attack on the HB-MP Protocol

Defending against the man-in-the-middle attack as proposed by Gilbert et al. has been considered in the HB-MP protocol, hence the rotation of xm . But this rotation has its own weakness. In the design of the HB-MP protocol, for every new session, xm needs to be identical in the i th round. It is not stated clearly about when to start and end an authentication session. It is reasonable

to suppose that when the tag enters the electromagnetic field and starts to talk with the reader, an authentication session begins and when the q -round enquiry is finished or the tag departs from the electromagnetic field of the reader, the authentication session ends. Since $x = Rot(x, y_i)$, so xm in the first round of all the authentication sessions should be the same. If the attacker pretends to be a valid reader, he can initiate repetitive authentication sessions, initially restricted to the first round. The techniques used in last section can then be exploited to reveal the tag's first round xm . If the attacker observes the i th round, he is able to reveal the xm used in the i th round.

The reason why the protocol has to use the same xm between authentication sessions is the synchronisation problem. If the value of x is updated to the rotated value after every authentication session on both the reader and tag side, a new reader will not be able to verify the updated tag and a valid new tag can not be verified by the updated reader, since the values of x stored in the reader and tags are not the same. Unless all the readers and tags are updated at the same time after every authentication session, which is expensive and technically difficult, the synchronization problem forbids the HB-MP protocols to change the xm .

Even if the synchronization problem is solved, the x is updated in every authentication session. There is still a way to conduct the man-in-the-middle attack. The length of x and y is k , if in an authentication session, the protocol runs k rounds, the x will be rotated p bits, here p is the number of '1' in y , so if the attacker runs the protocol for k times, namely k^2 rounds, the x will be rotated $p \cdot k$ times and it is rotated back to its initial value. so a repeat of xm happens again. Since the proposed x is 512 bits, so 262,144 rounds will definitely generate a repeated xm . It is an affordable attack.

2.6 An Improved HB-MP Protocol

The vulnerability of this protocol stems from the predictable repetition of xm , if the rotation is random in each round, the repetition of xm is unpredictable, thus the attack is defended.

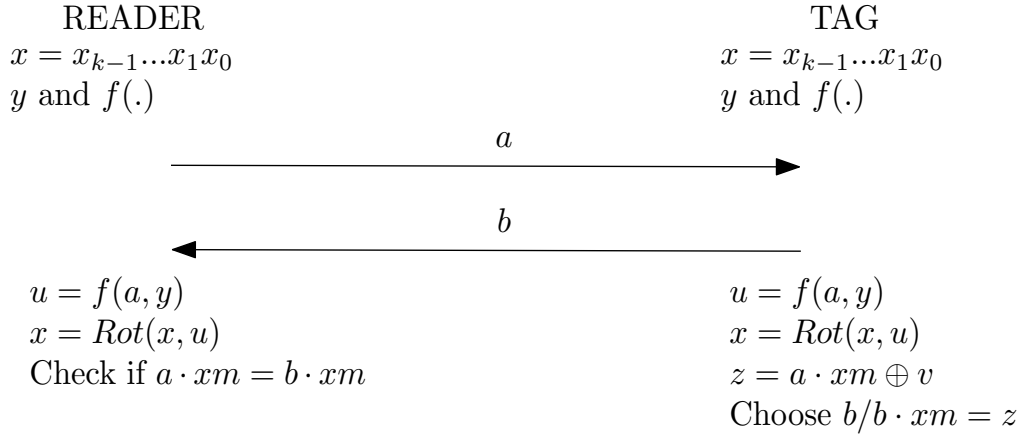


Figure 2.5: The i th Round of the Improved HB-MP Protocol

The notations are same with the original HB-MP protocol except a one way function $f(\cdot)$ and an intermediate value $u = f(a, y)$. Instead of using the default threshold ηq , I define the threshold t . Because Gilbert et al. [28] cites the result of other research, describing threshold ηq can cause unreasonably high false rejection rate. The threshold t can be adjusted to a value larger than ηq to reduce the false rejection rate, and retain a low false acceptance rate at the same time. This improved protocol also has q rounds and if the failures do not pass the threshold t , the tag is authenticated successfully. The i th round is depicted in Figure 2.5 and described as follows:

Step 1. The reader chooses at random a m -bit binary vector a and send it to the tag.

Step 2. The reader and tag compute $u = f(a, y)$ and $x = Rot(x, u)$. xm is

selected as the first m -bits of current x .

Step 3. The tag computes z as follows:

$$z = a \cdot xm \oplus v \tag{2.6.1}$$

and looks for a m -bit binary vector b such that $b \cdot xm = z$.

Step 4. The tag sends b to the reader.

Step 5. The reader computes the x in the i th round as $x = Rot(x, t)$ and selects xm from this x .

Step 6. The reader checks if

$$a \cdot xm = b \cdot xm \tag{2.6.2}$$

By using the random number a , The improved protocol makes the rotation of x unpredictable. An advantage of this improvement is that the original form of the HB-MP protocol is kept and the only change is the computation operated inside the tag and reader. This protocol improves the HB-MP protocol by making the rotation of the secret unpredictable to the attacker.

2.7 An Abstract Form of HB-MP⁺ Protocol

The core idea of the improved protocol is to use some additional random bits generated by the reader to randomize the rotation. The evolutionary design idea from HB-MP' to HB-MP was to rotate the secret key x in each round. The improved HB-MP protocol makes the rotation unpredictable by adding randomness. If I extend the design idea a step further, the essential part of defending against man-in-the-middle attack is to use a random secret in each round, namely a **Round Key**. The vulnerability of HB-MP comes from the predictability of the round key. So if the focus of protocol design points to

the generation of a round key by using the random bits and shared secrets, a new protocol can be proposed. The HB-MP⁺ protocol I proposed is called in abstract form because the one-way function $f(\cdot)$ is not concrete. HB-MP⁺ is also in accordance of the naming convention of HB-family protocols. The HB-MP⁺ protocol also has adjustable threshold t to improve the false rejection rate. Figure 2.6 shows one round of the HB-MP⁺ protocol. *Step 1.* The reader

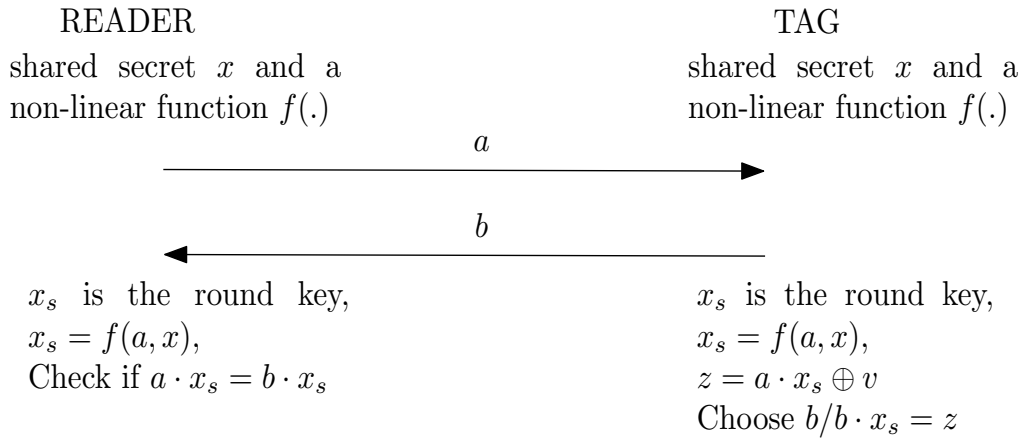


Figure 2.6: The i th Round of the Abstract HB-MP⁺ Protocol

chooses at random a m -bit binary vector a and send it to the tag.

Step 2. The reader and tag compute the round key $x_s = f(a, x)$. $f(\cdot)$ is the one-way function.

Step 3. The tag computes z as follows:

$$z = a \cdot x_s \oplus v \tag{2.7.1}$$

and looks for a m -bit binary vector b such that $b \cdot x_s = z$.

Step 4. The tag sends b to the reader.

Step 5. The reader computes the $x_s = f(a, x)$, using the secret x and random number a .

Step 6. The reader checks if

$$a \cdot x_s = b \cdot x_s \quad (2.7.2)$$

The notations are almost the same with the improved HB-MP protocol, except that the non-linear function $f(\cdot)$ is the abstraction from $Rot(\cdot)$. Since rotation is a linear operation, the output of $f(\cdot)$ should be less predictable. Using the bit operations, it is easy to implement a low-cost non-linear function $f(\cdot)$. As $f(\cdot)$ does not necessarily use rotation, the bits of x are not mentioned. The round key x_s is generated by a random number a and shared secret x . There is no need for another shared secret y and because the x is not changed, there are no synchronisation problems between the readers and tags.

2.8 Security and Performance Analysis of The HB-MP⁺ Protocol

2.8.1 Security Analysis

The improved HB-MP protocol and HB-MP⁺ protocol is based on the HB-MP protocol. Since the way of transmitting messages is the same with the HB-MP protocol. The analysis in Munilla and Peinado [38] also applies to the HB-MP⁺ protocol: A passive attacker has to solve the LPN problem to reveal the secret of tags(x , more specifically).

In the design of these two protocols, it is very important that the round key is calculated by random challenge a and the secret x . Since a is randomly generated by the reader and stored in the reader, the attacker cannot either predict a or modify a . If an attacker modifies a to a' to cheat the tag, the tag side will use a' to generate the round key x_s , which is not the same round key generated at the reader side since the reader still uses its own a . Thus the

attacker cannot get any valuable response from the tag.

It is necessary to define the attacker's aim when I start to talk about the attack. In the two protocols I propose, the tag does not authenticate the reader. In other word, the tag does not care who is challenging it, it just gives back responses according to the challenge. So the attacker's meaningful aim is to fool the reader to authenticate a fake tag or to reveal the secret x . Man-in-the-middle attack presented in Gilbert et al. [31] can reveal the secret of the target protocols. To illustrate how our protocols defend against the man-in-the-middle attack, I give a model describing the ability of an active attacker. Let \mathcal{A} be an adversary, who can intercept the communication between the reader and the tag. \mathcal{A} can also pretend a reader to challenge the tag. \mathcal{A} can block and modify the messages sent both by the reader and the tags. Finally \mathcal{A} can fool the reader to answer multiply responses for a single challenge in each round (which is unlikely to happen to a reader with the predefined procedure to handle the information received). I only illustrate the attack to the HB-MP⁺ protocol, which is the mature protocol in this paper. Fig.7 shows an assumed man-in-the-middle attack launched by the attacker \mathcal{A} against the HB-MP⁺ protocol.

If \mathcal{A} is the man-in-the-middle and attacking the HB-MP⁺ protocol, \mathcal{A} gets the challenge a sent by an authentic reader, then \mathcal{A} pretends to be a reader and challenge the tag with the same a q times. \mathcal{A} receives q responses b_i ($1 \leq i \leq q$) from the tag, \mathcal{A} modified the responses into $b'_i = b_i \oplus \delta$ ($1 \leq i \leq q$) and gives them back to the reader at one time. \mathcal{A} gets the answer(accepted or rejected) from the reader. Then \mathcal{A} knows that $x_s \cdot \delta = 1$ or $x_s \cdot \delta = 0$ with high probability, \mathcal{A} can retrieve a bit of the round-key x_s generated by a . If the attacker can fool the reader to answer the challenge a continuously,

assumed attack is unlikely to happen, the communication between RFID and the reader is predefined and sequenced. A normal reader will not tolerate more than one responses for a single challenge. State machine inside the reader will also start a fresh challenge after a q -round authentication session. To response one challenge so many times is the assumed condition to facilitate the attack. Another attack needs to be noticed is that the adversary can always response $b = a$. The reader needs to check that the response should be different from the challenge each round.

2.8.2 Performance Analysis

Because the HB-MP⁺ protocol uses round keys, the secret is updated in each round, so m (the size of the round key) does not need to be as large as suggested in Gilbert et al. [28]. In their paper the authors suggested that $k = 512$ and $\eta = 0.25$ is good enough. In HB-MP⁺ protocol, the secret x can be 512 bits while m can be significantly smaller. $m = 224$ offers a security level of 52-bit [28], which should be enough for the round key.

Another significant reduce of transmission cost comes from the fact that HB-MP⁺ protocol transmits two messages instead of three each round. This will cut 1/3 transmission cost comparing to the HB⁺ protocol.

Despite the improvement on the performance, HB-MP⁺ protocol still suffers the same performance penalties with the HB⁺ protocol. It still needs to run many rounds and there are still too much data needs to be transmitted for an authentication session. The transmission cost is too high for current RFID systems.

2.9 Conclusion and Future Work

In this chapter, a vulnerability of the HB-MP protocol that may enable a successful man-in-the-middle attack has been identified. An enhanced version of HB-MP protocol is proposed that eliminates the vulnerability and keeps the simplicity of the original protocol. In the HB protocol family, the HB-MP protocol and our enhanced protocol have maintained a simple form and they are closer to the original idea of HB protocols. At the end of chapter 2, an abstract form of the HB-MP⁺ protocol introduces the idea of random round keys is also proposed. This chapter also improves the algorithms of picking random responses and gives adjustable threshold to reduce the false rejection rates. The abstract form of the HB-MP⁺ protocol requires concrete the one-way function $f(\cdot)$, however this is thought within the capabilities of RFID devices and comparable with the $Rot(\cdot)$ function used in HB-MP.

Chapter 3

Introduction to Grouping Proof Protocols

In this chapter, I give the detailed description and analysis of the RFID grouping proofs protocols published in recently years, as the foundations of the next several chapters in which our contributions are introduced.

3.1 Preliminaries

A rather different security oriented problem for RFID tags is considered in this chapter. The aim is to enable a pair of or multiple RFID tags to generate a proof, which shows that they have been scanned simultaneously by a reading device. I refer to this as a grouping proof.

3.1.1 Protocol Goal

The first grouping proof protocol is proposed by A. Juels [40]. The original protocol was called yoking proof (applying “yoke” with its meaning to join things together), the word “grouping proof” is extended from yoking proof in the following researches. Considering multiple tags are involved in these publications, The word “grouping proof” is used instead of “yoking”. But essential mechanism in these publications is very close to yoking proof.

Readers might wonder the qualification of the simultaneously reading problem. When a reading device is reading several tags, it can recognise all the tags after anti-collision, why bother to produce such a proof? The key idea which qualifying the “yoking proof” is that, in certain circumstances, a proof is need to be verifiable by a trusted party in an offline setting, rather than requiring direct involvement by this party. So the result of the protocol should produce a proof which can be sent to another offline party to be verified.

3.1.2 Environment Assumption

In this section, I will brief describe the environment in which the protocols will work. To help the readers with the notation used in the following protocols described in this chapter, I also put the notation used in all in table 3.1. After introducing this necessary knowledge, I will show how these proofs are evolved.

Tags

Given their inexpensive nature, RFID tags can offer very little in the form of tamper-resistance. However, that basic tamper-resistant features will act as an obstacle to key retrieving attacks. Actually nearly all the current RFID protocols assume that the key of an uncorrupted tag is not revealed by other attacks. In other word, the attackers do not know the keys of the innocent tags. Additionally, in the current RFID system, basic RFID tags cannot communicate with one another directly. Rather they must rely on the reading device that is querying them as a communications channel.

Adversaries

Because the adversary exists to test the security of the protocol, as the normal assumption, I assume that the adversary actively controls the communication

channel between the tags and the reader. But his attack does not involve other side channel attacks to the tags itself. So an adversary in this system does not have the key of the innocent RFID tags. He will try to actively or passively attack the communication channels to break the security of this system. Also Juels' proposal relies on a timeout assumption, namely that the protocol will always terminate within a certain interval of time t . This is a feature of the basic RFID protocol itself, that is to say, the attacker should finish his attack in a limited time, the protocol cannot wait his attacks for a long time.

Minimalist MAC

The challenge in designing a yoking-proof protocol is that RFID tags have very rudimentary computational abilities. In this thesis, I assume that they can perform only very basic computational operations, which do not include standard cryptographic functions, but MACs are included. Along with the first “yoking proof” protocol, A. Juels proposed a minimalist MAC, which is a stripped-down, symmetric-key version of the Lamport digital-signature construction as described in [50]. The minimalist MAC assumes that the message m to be MACed is exactly d bits in length. A secret key SK is determined as collection of random, l -bit secret values $\{(s_i^0, s_i^1)\}_{i=1}^d$. This secret key SK is shared between the signer and verifier. A MAC on message m can be computed as $MAC(m) = b_1b_2 \cdots b_d$ simply consists of the collection of secret values $\{s_i^{(b_i)}\}_{i=1}^d$. To forge a MAC in this scheme, an adversary must successfully guess at least one unrevealed value $s_i^{(1-b_i)}$. Given sufficiently a large value l (e.g., $l = 128$), this is infeasible.

However, in Juels' paper, he proposed setting $l = 1$, i.e., making each $s^{(b_i)}$ only a single, random bit. The problem with this approach, of course, is that

Table 3.1: Notation of Yoking Proof by A. Juels

\oplus	XOR operator
T, T_A, T_B	Tags
A, B	IDs of tag A and tag B
A_i, A_R, A_{PT}	IDs of tag i , Reader and PT, respectively
r, r_i, r_A, r_B, r_P	Random Numbers
X, X_A, X_B	Symmetric Secret Keys
MAC	Message Authentication Code
$MAC_X(m)$	MAC of m using key X
V	Verifier
R	Reader
TS	Time Stamp
$SK_x[m]$	a ciphertext by using secret key x to message m .
P_n, P_{AB}	Grouping Proof)

given a $MAC_x[m]$, an adversary can forge $MAC_x[m']$ on a new message m' quite easily: If m' differs from m in a single bit, the adversary only need to guess a single bit to perform the forgery. So Juels hope that if the message space is sufficiently sparse and pairs of messages tend to have relatively large Hamming distances, then forgery will be more difficult. By choosing sufficiently large d , I can ensure that the Hamming distance between randomly selected bit strings is large.

It is possible to do somewhat better, however, by crafting the message space more carefully. Given the space limitations of RFID tags, this is important. In particular, one can select a message space with a good lower bound on the Hamming distance between any two messages. This is most easily achieved by defining the message space as the codebook for an error-correcting code. As an example, suppose that we set $d = 120$, and select a message space size of 2^{32} (enough for billions).

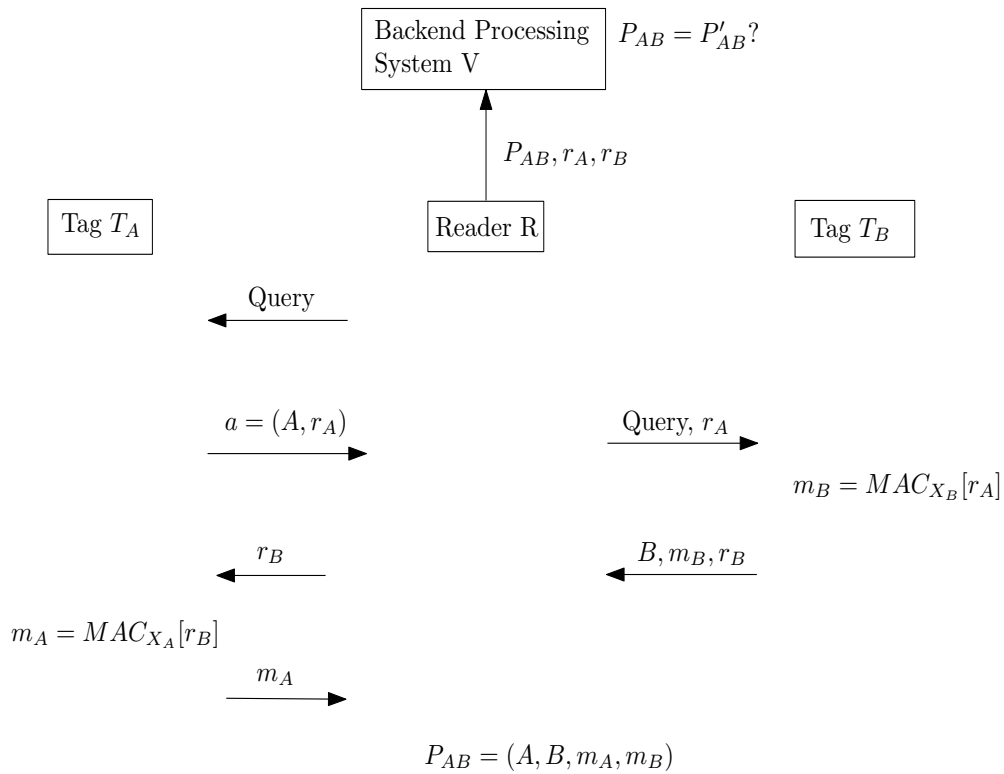


Figure 3.1: Yoking Proof for RFID Tags

3.1.3 Notation

3.2 Yoking Proof by A. Juels

The aim of yoking proof is “to enable a pair of RFID tags to generate a proof that they have been scanned simultaneously by a reading device.” [40].

3.2.1 The Original Yoking Proof

The original yoking proof was proposed by A. Juels, the procedure of this protocol is shown in Fig.3.1. The tags T_A and T_B share the secret keys X_A and X_B with the verifier V (Backend Processing System). The protocol takes the following steps:

1. The reader R sends the Query command to Tag T_A , On receiving the Query command, T_A generates a random number r_A and sends it back to R with its Identity (ID) A .
2. R forwards r_A to T_B .
3. T_B calculates $m_B = MAC_{X_B}[r_A]$, it can be adapted to the 'Minimalist MAC' protocol by Juels [39], using its secret key X_B and r_A received from R . T_B then generates a random number r_B and send it back to R along with m_B and its ID B .
4. R forwards r_B to T_A .
5. T_A calculates $m_A = MAC_{X_A}[r_B]$, using its secret key X_A and r_B just received, and send m_A back to R .
6. R sends the concatenation, i.e. proof, $P_{AB} = (A, B, m_A, m_B)$ and r_A, r_B to the Backend Processing System V .
7. Since V has everything needed to assembly the P_{AB} , V generates its proof P'_{AB} and compare it with P_{AB} received. If these two values are identical, V believes T_A and T_B are present simultaneously.

3.2.2 Replay Attack Against Original Yoking Proof

In 2005, J. Saito and K. Sakurai [71] pointed out the original “yoking proof” is vulnerable to the replay attack. The main vulnerability of the yoking proof comes from the fact that the verifier does not give any randomness to ensure the freshness of the proof generated. An attacker can reuse the messages and succeed a replay attack. This reply attack is described in The attackers can reuse any proofs produced by the tags. In Saito and Sakurai’s paper, they

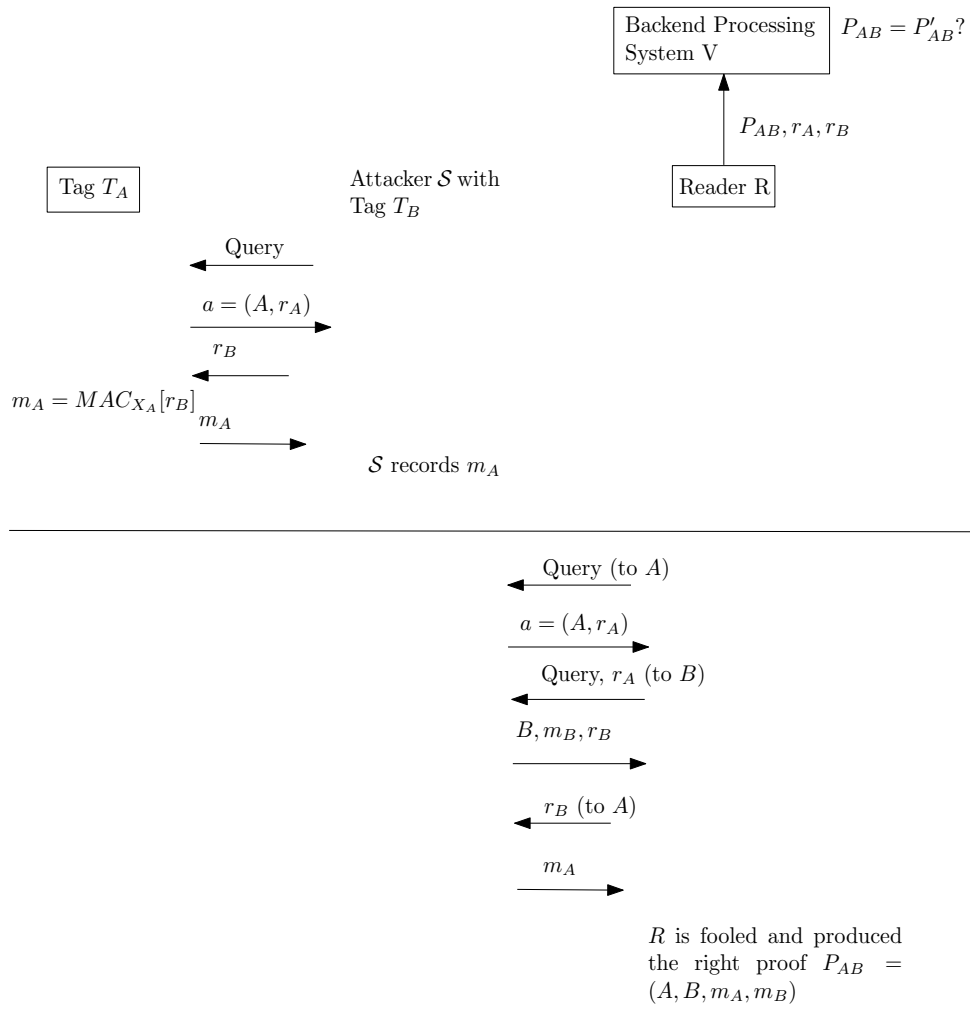


Figure 3.2: Replay Attack Against Original Yoking Proof

gave an attack scenario, in which the attacker produced the complete yoking proof with only tag B . Fig.3.2:

1. An attacker \mathcal{S} emits a query to A and gets r_A .
2. \mathcal{S} send A back with his own r_B
3. A calculate $m_A = MAC_{X_A}[r_B]$ and send it back. \mathcal{S} acquires m_A
4. Now \mathcal{S} can fool the reader that he has A and B simultaneously while he

only has B .

5. \mathcal{S} replays the message from A to answer the queries to A .
6. After receiving r_A and A from the reader, \mathcal{S} used B to calculate $m_B = MAC_{X_B}[r_A]$, and send the old r_B back with m_B .
7. After receiving r_B from the reader, \mathcal{S} use the same m_A , using its secret key X_A and r_B just received, and send m_A back to R .
8. Now R has the proof, $P_{AB} = (A, B, m_A, m_B)$. He is fooled by \mathcal{S} and thought \mathcal{S} has both A and B .

3.2.3 Yoking Proof with Time Stamps

To counter the replay attack, J. Saito and K. Sakurai have proposed a solution by adding the time stamps. In their proposal, RFID tags compute a MAC by applying a secret key to a time stamp from V . Therefore a verifier can verify the freshness of the MAC. Thus we can prevent the replay attack which reuses the MAC. Their scheme also relies on assumption of the timeout. Their protocol is illustrated in Fig.3.3.

1. Reader R gets a time stamp TS from V through a secure channel and sends TS to tags T_A and T_B .
2. After receiving TS , T_A calculate $m_A = MAC_{X_A}[TS]$ and send it back to R .
3. R forwards m_A to T_B .
4. T_B calculates $m_B = MAC_{X_B}[TS, m_A]$, using its secret key X_B and m_A received from R . T_B then sends m_B to R .

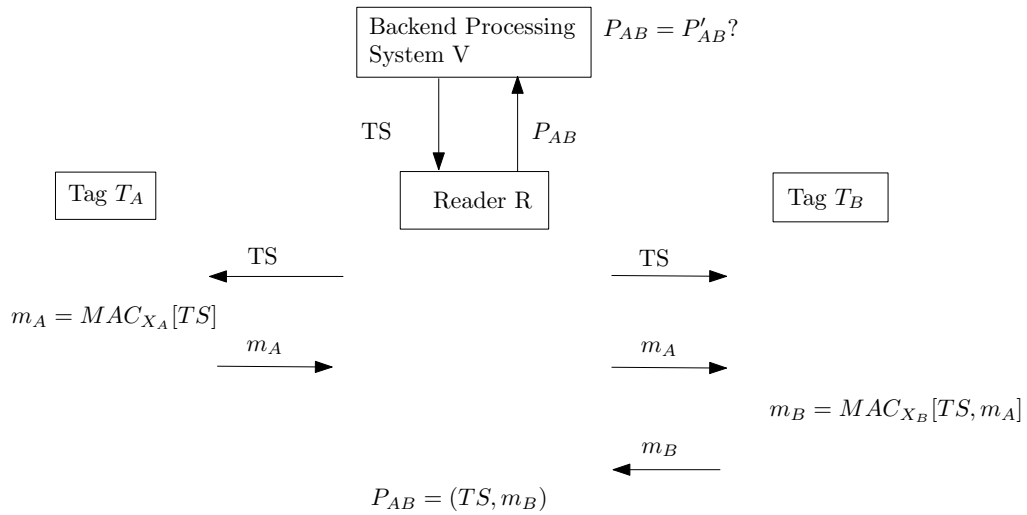


Figure 3.3: Yoking Proof Using Time Stamps

5. R send the proof $P_{AB}(TS, m_B)$ to the Backend Processing System V .
6. V generates its proof P'_{AB} and compare it with P_{AB} received. If these two values are identical, V believes T_A and T_B are present simultaneously.

3.2.4 The Modified Yoking Proof

S. Piramuthu [64] pointed out that J. Saito and K. Sakurai's yoking proof with Time Stamps is not completely resistant to the replay attack. S. Piramuthu pointed out that the attacker can begin by repeatedly query tag T_A , using several different time stamps from some later points in time. Various combinations of (TS, m_A) can be gathered in this manner. Then, at some later point in time, if any TS is used by the V , the replay attacks can be instantiated without the presence of T_A . So the protocols proposed by J. Saito and K. Sakurai are vulnerable to the replay attack.

His solution is illustrated in Fig.3.4 . In his solution, the Backend Processing System V gives R a random number r through a secure channel. This r

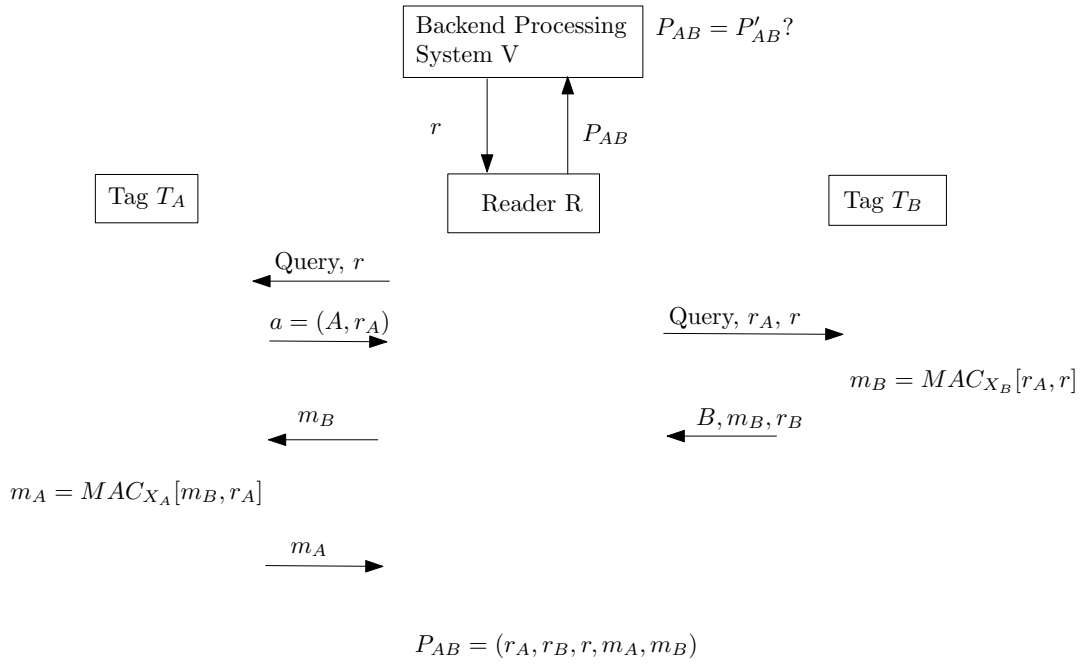


Figure 3.4: Yoking Proof Modified by S. Piramuthu

serves as the random seeds for T_A and T_B to generate their random numbers r_A and r_B . V starts its transaction timer as soon as it sends r . The reader must send back the proof P_{AB} in a specified time period otherwise the verification fails. The addition of a random variable r sent to both the tags from the verifier through the reader sending r instead of sending the time stamp eliminates the possibility that the time stamp is predicted by the attacker.

There are several properties in this modified protocol:

- Since r_A is generated and used internally in T_A for generating m_A , an adversary cannot run a replay attack on either of the tags. Also because m_A depends on m_B and r_A and m_B generated by T_B depends on r_A , another protection layer is added for the replay attack.
- The fifth transmission in Fig.3.4 is m_B instead of r_B in Fig.3.3 to improve

the security.

- Using m_B in generating m_A is crucial since T_A has to wait for T_B to generate m_B . Therefore, T_A 's part of the proof cannot occur before T_B 's part and T_B 's part cannot happen independently since it is also dependent on input from T_A (r_A).

3.3 Grouping Proof

Although the yoking proof for only a pair of tags is considered, A. Juels mentioned that he is considering the expansion to produce the proof for a group of tags. The problem is that as the number of tags increases, it would certainly be desirable to reduce this requirement and produce more efficient proofs. The concrete idea of grouping proof is proposed by J. Saito and K. Sakurai [71]. Later L. Bolotnyy and G. Robins [49] proposed the “Generalized yoking-proofs” by extending Juels’ methods and an “Anonymous yoking” method to improve privacy. Piramuthu’s [64] two tags yoking proof may also be extended to a grouping proof since it can be considered as an improved version of Saito and Sakurai’s protocol. We suppose in these schemes that all the verifiers know which tags they are expecting and their secret keys.

3.3.1 Grouping Proof by J. Saito and K. Sakura

J. Saito and K. Sakura give the concept of grouping proof in [71]. Their protocol is shown in Fig.3.5. A party called “Pallet Tag” (PT) is introduced in this protocol. PT can be a large metal plate or flat wooden pallet on which some products can be lifted or moved. PT has more computing resources than normal tags, i.e. enough to meet the demand of this protocol. Prior to running the protocol, the Verifier V is assumed to have securely shared the secret keys

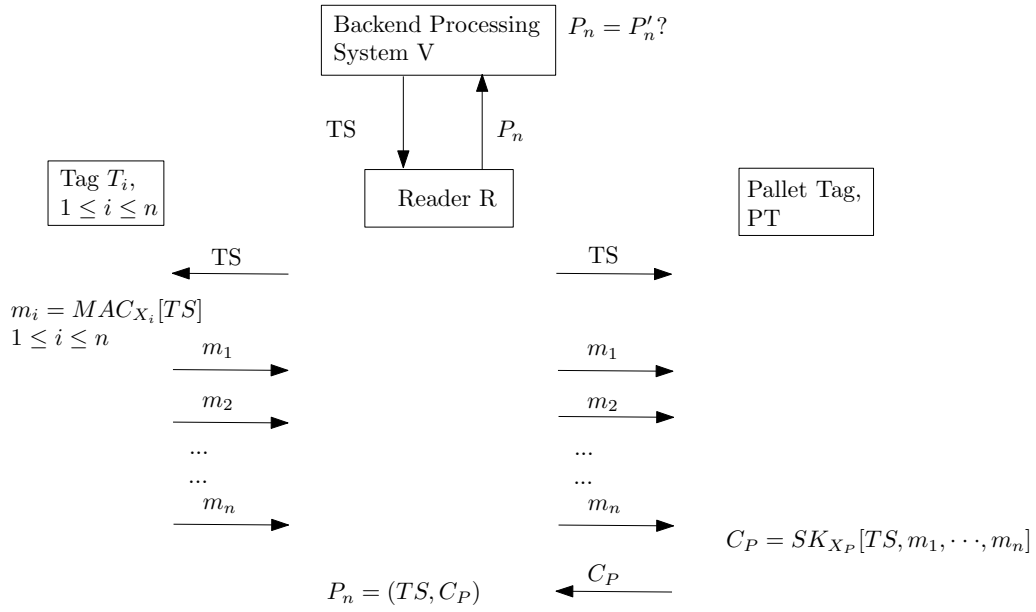


Figure 3.5: Grouping Proof by J. Saito and K. Sakura

with the tags and PT. The steps of their protocol are:

1. Reader R gets a time stamp TS from verifier and broadcasts it to all the tags and PT. The Verifier starts the timer as soon as it sends the time stamp.
2. After receiving TS , it can be adapted to the 'Minimalist MAC' protocol by Juels [39], each tag T_i calculates $m_i = MAC_{X_i}[TS]$ and sends m_i back to R .
3. R forwards m_i to PT, in the numerical order from 1 to n .
4. After receiving all the m_i , PT uses its X_p to calculate $C_P = SK_{X_P}[TS, m_1, \dots, m_n]$ and sent C_P to R .
5. R sends the concatenation $P_n = (TS, C_P)$ to the Backend Processing System V as the grouping proof.

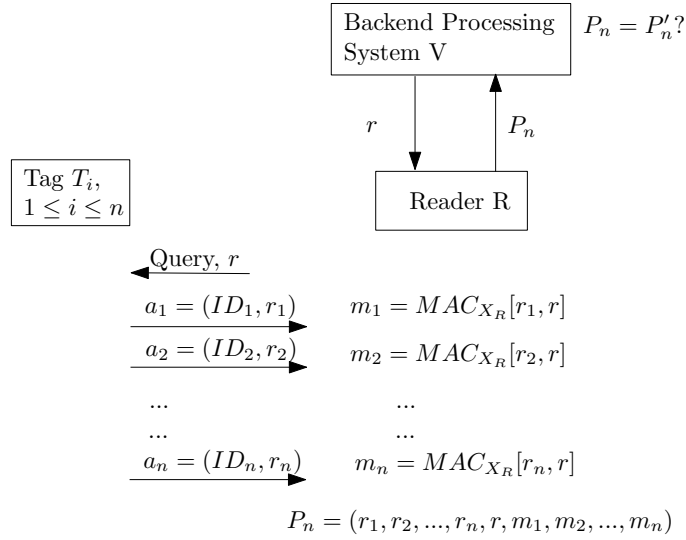


Figure 3.6: Grouping Proof Based on S. Piramuthu

6. V generates its proof P'_n and compare it with P_n received. If these two values are identical, V believes all the tags are present simultaneously.

3.3.2 Grouping Proof Inspired by Piramuthu

S. Piramuthu's also proposed the grouping proof based on the mechanism dealing with two tags [64]. The key idea of his protocol is to ensure that the inputs to a tag are based on parameters that are necessary for the other tag, and to create dependence of the tags on each other so that they cannot be processed separately in the proof without the presence of the other tag. As illustrated in Fig.3.6, The reader calculates $m_i = MAC_{X_R}[r_i, r]$ after receiving the responses $(a_i = ID_i, r_i)$ from each tag(T_i). Then R produces $P_n = (r_1, r_2, \dots, r_n, r, m_1, m_2, \dots, m_n)$ as the proof to the verifier.

3.3.3 “Generalized Yoking-Proofs” and “Anonymous Yoking” for a Group of RFID Tags

L. Bolotnyy and G. Robins [49] extended A. Juels’s work and proposed “Generalized yoking-proofs” and “Anonymous yoking proof”. In their works, they still use yoking proof instead of grouping proof to refer the protocols deal with more than two tags. yoking proof or grouping proof are just some word difference and does not matter at all. Although there is no graphic interpretation of these protocols, we draw the figures (Fig. 3.7 and Fig. 3.8 for their protocols according to the description in the article. Each tag T_i has a counter c_i , and the first tag T_1 uses its secret key x_1 and hash function to generate r_1 , i.e. $r_1 = f_{x_1}[c_1]$, then T_1 sends ID_1, c_1, r_1 to the reader. T_2 gets $a_1 = (ID_1, c_1, r_1)$ from the reader and computes $r_2 = MAC_{x_2}[c_2, a_1]$. Each tag performs the same process in series. Finally, the reader sends the a_n to the first tag, which computes $m = MAC_{x_1}[a_1, a_n]$, and sends m to the reader. The reader creates a proof $P_n = (ID_1, ID_2, \dots, ID_n, c_1, c_2, \dots, c_n, m)$ and send P_n back to the verifier. However, it can be seen that the extension of the original protocol does not hide the identities of individual tags. In practical scenarios, it is desirable that the privacy of objects associated with the tags is preserved. A new scheme called “anonymous yoking” is introduced, to preserve privacy of the tags. In Fig.3.8, upon receiving the request, each tag generates a random number r_i , and computes $a_i = f_{x_i}[r_i, a_{i-1}]$, where $a_0 = 0$. Tags send back their (r_i, a_i) pairs to reader, and the first tag links the chain by computing $m = f_{x_k}[a_1, a_n]$. The reader creates a proof $P_n = (r_1, r_2, \dots, r_n, m)$ for the verification. In their protocol, L. Bolotnyy and G. Robins consider many practical implementation details of the grouping proofs. Firstly, they consider the problem of privacy in the RFID protocols. Privacy is a serious concern in many RFID applications.

It is preferable to preserve the privacy of objects associated with the tags. They adopted the temporary IDs in the communications and called their new methods “anonymous yoking”. Secondly their paper considered the counters c_i inside the each tag. If the first tag does not update its counter right after it sends its first message, a possibly malicious reader can create a proof P that will successfully pass through the verifier, without reading all the tags within the specified time. In such a scenario, a proof can be forged as follows. The malicious reader can ask the first tag T_1 to compute a_1 , then wait for T_1 to timeout. Then the attacker sends a_1 to T_2 to obtain a_2 . Then, the attacker could send a_2 to T_1 to obtain m , and construct a valid proof. Juels original yoking protocol suffers from this problem unless the counter on the first tag is incremented on a timeout, but this is not specified in his paper. Thirdly, they pointed out that the protocol can be sped up by splitting the circular chain of dependent MACs into subgroups, where each subgroup consists of a sequence of dependent MACs, and where the adjacent subgroups are inter-dependent. Each subgroup has a single element that plays the role of the “first” and the “last” tag. In this thesis, this idea is also developed in the Chapter 6.

3.4 Conclusion

The “grouping proof” is a term generalised from “yoking proof”. yoking proof means two tags are bounded together by some kind of linkage, like a yoke binding two horses. Since yoking proof proves binding of only two tags, grouping proof is given as the term to describe more than three tags bound together. These extend the yoking-proofs to handle arbitrary number of tags in a group, where the group of tags generates a proof of having been scanned nearly simultaneously. From Juels’ original yoking proof protocol[40], the authors might

only want to show a new idea of ‘proof of actions’. Previously the proofs in the RFID system are ‘proof of identities’. The yoking proof protocol is designed to show a certain action of RFID tags: Being scanned together at a particular time. The original paper focuses more in introducing the new proof ideas than make it secure or efficient. The yoking proof draw much attention after its publications. That is because the future application of RFID tags are due to be in huge amount, and dealing with a bunk of tags is still a theoretical unsolved problem. In the remain part of this thesis, I am going to propose some grouping proof protocols with different properties to solve these problems.

Then with those expanded applications, new properties of the protocol have been required to fulfill the demand of these applications. To fix the vulnerability of Juels’ yoking proof, which enabled the replay attack successfully breaks the proof. This vulnerability comes from the fact that the verifier does not give any randomness to ensure the freshness of the proof generated. Saito and Sakurai[71] modified the original protocol to include a time-stamp, with a view to thwart replay attacks as described in Chapter 3. Piramuthu[64] adapted the Saito and Sakurai’s protocols[71] to include random values in lieu of time-stamps. This is important, because time-stamps can be predicted, allowing for attacks that collect prior responses and combine them to achieve false proofs of simultaneous interaction. L. Bolotnyy and G. Robins[49] also proposed that, to avoid replay attacks, the verifier stores some information about previous correct proofs. The verifier is not required to store this information, if replays of valid proofs are not considered to be attacks. This will be elaborated upon in the discussion following the protocol specification. L. Bolotnyy and G. Robins also first proposed the idea of anonymous grouping proof. For privacy issues of RFID, it is desired that real ID of the RFID tags

are not disclosed in the transmission. Their anonymous grouping proof uses the hashed result of tag's secret and its real ID. Their protocol gives the first idea of using temporary ID instead of real ID for privacy conservation.

In some of the above scenarios, the RFID reader may not enjoy continuous connectivity with the trusted Verifier, and delayed confirmation may be acceptable. For instance, this may indeed be the case with regard to supply chain applications, due to the increased fragmentation and outsourcing of manufacturing functions. A supplier of partially assembled kits may perform scanning activities that will be verified later when the kits are completed at a different site.

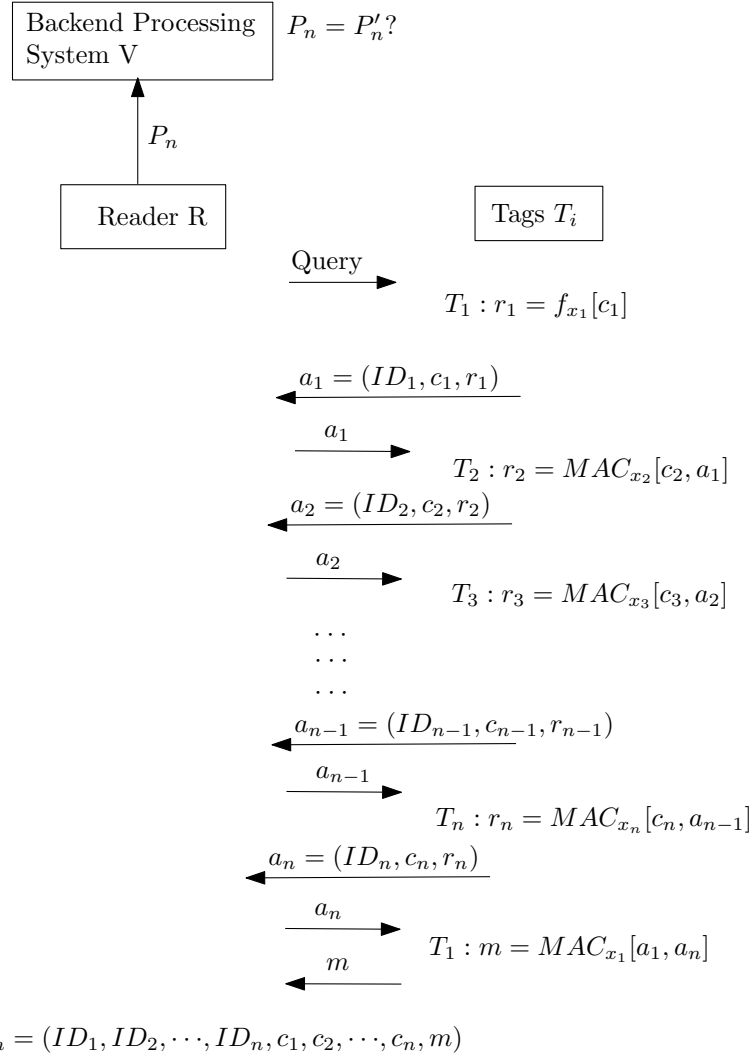


Figure 3.7: Generalized Yoking Proof Based on L. Bolotny's Algorithm

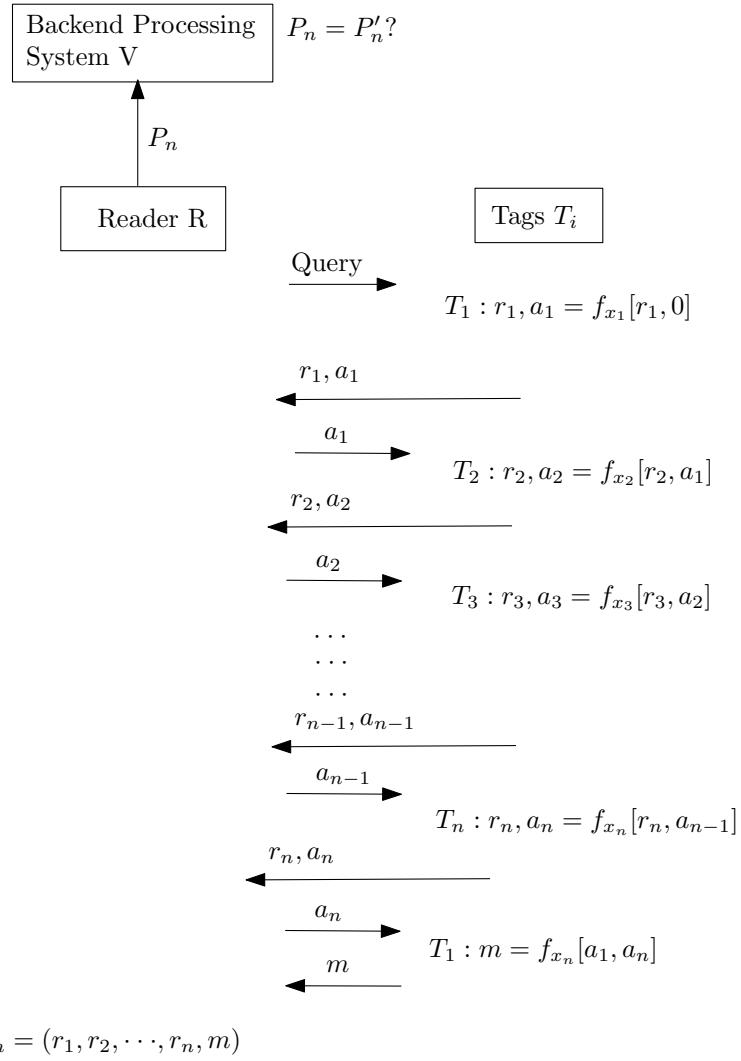


Figure 3.8: Anonymous Yoking Proof based on L. Bolotny's Algorithm

Chapter 4

Reading Order Independent Yoking Grouping Proof Protocol

In this chapter, we pinpoint the problems of using temporary ID in the previous grouping proofs which are reading order dependent. For a reading order dependent grouping proof protocol, the reader side cannot tell the correct reading order from the temporary IDs, plus the reading order dependent properties, the correct grouping proof is very difficult to produce. We introduce our reading order independent grouping protocol in this chapter. In the introduction section, there will be an analysis interpreting why the reading order independence is a demanded property in certain applications. Then we give our reading order independent protocol in the following section. We also have a brief analysis of this protocol at the end of this chapter. cooperated work with Mr. Yuanhung Lien, I have contributed on the protocol design and security analysis.

4.1 Reading Order Penalty

All the yoking proof and the group proofs mentioned above have their innovations and contribution in solving the multiply tags identification problems. The core idea of these protocols is “binding tags together”. However, when it comes to applications, the problems of ‘how to bind the tags efficiently’ is a viable difficulty. Piramuthu’s protocol and the other two protocols proposed by L. Bolotnyy and G. Robins are also running tag sequential order, i.e. the tags need to be called in some specific order. In all these protocols, besides the tags must be read in exactly the same order for every authentication session. In Fig.3.5, $P_n = (TS, C_p)$, in which $C_p = SK_{X_p}[TS, m_1, m_2, \dots, m_n]$ is the result of symmetric encryption, so if m_i is not in the expected order, the result C_p would not match the proof stored in the Verifier and the verification would fail. In Fig.3.6, Fig.3.7 and Fig.3.8, if the grouping proof P_n is not sent to the Verifier in the expected order, the Verifier can still identify the tags by verifying ID_i or r_i , but the verification time and computation load is increased. In a word, since the tags must wait the response from the previous tags before they can do anything. Order dependence is inefficient and raises the failure rate of verification in those protocols. We call it reading order penalty.

Besides the reading order penalty, the reading order dependence also makes the anonymity difficult to implement. According to L. Bolotnyy and G. Robins [49], they used temporary IDs to keep the real ID of the tags being disclosed. Temporary ID also serves the defence of tag tracking since the temporary ID changes every time, which make the attackers unable to track certain tag according to the ID previously gained. Temporary ID is a good method for these purposes. However, for a reading order dependent grouping proof protocol, since the reader side is not able to identify the real ID of

the tags, actually the reader is not suppose to identify the tags in this scenario. How can the reader query the tags in the pre-defined order? It is an impossible task for real implementation. Thus reading order independent is a desired property of the grouping protocols which adopt temporary ID for privacy protection.

Now we introduce the our Reading Order Independent (ROI) protocols first, and then examine how the reading order penalty affects the performance of the previous protocols by simulate all the grouping proof protocols proposed.

4.2 Reading Order Independence Protocol

To make the verification order independent, we propose a grouping proof protocol which modifies S. Piramuthu’s protocol and the idea of the “Pallet Tag” (PT). As illustrated in Fig.4.1, all the tags and backend processing system have the same random number generator structure and the verification steps are:

1. Reader R gets a random number r from the Verifier and broadcasts it to all the tags and PT. The Verifier starts the timer as soon as it sends r , i. e. the procedure must complete within a certain time period.
2. Using r as the seed, each tag T_i calculate its random number r_{A_i} and sends r_{A_i} back with its identification code A_i . PT also generates its random number r_p .
3. For efficiency, R sends $\{A_{PT}, A_i, r_{A_i}\}$ pairs to PT without regard for ordering. Adding A_{PT} is to indicate the destination of this message is PT.

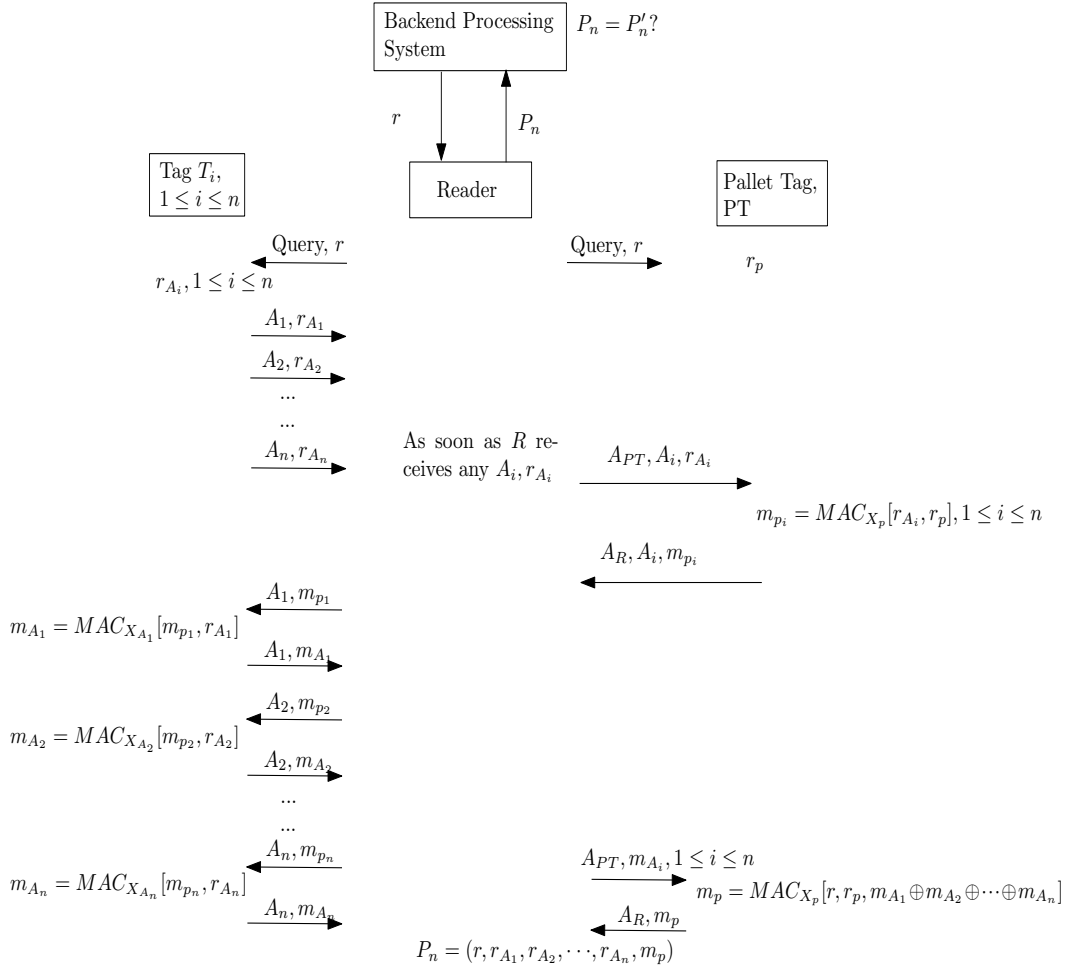


Figure 4.1: Reading Order Independent Grouping Proof for RFID Tags with Pallet Tag

4. After receiving n pairs of $\{A_{PT}, A_i, r_{A_i}\}$, PT uses its secret key X_p to calculate $m_{p_i} = MAC_{X_p}[r_{A_i}, r_p], 1 \leq i \leq n$ and $\{A_R, A_i, m_{p_i}\}$ back to R .
5. R sends A_i, m_{p_i} to the corresponding T_i .
6. T_i uses its secret key X_{A_i} to calculate $m_{A_i} = MAC_{X_{A_i}}[m_{p_i}, r_{A_i}]$ and sends $\{A_i, m_{A_i}\}$ back to R .

7. R sends any $\{A_{PT}, A_i, m_{A_i}\}$ it received to PT without regards for ordering.
8. After receiving n pairs of $\{A_{PT}, A_i, m_{A_i}\}$, PT uses its secret key X_p to compute $m_p = MAC_{X_p}[r, r_p, m_{A_1} \oplus m_{A_2} \oplus \dots \oplus m_{A_n}]$ and sends $\{A_R, m_p\}$ back to R .
9. R generates the grouping proof $P_n = (r, r_{A_1}, r_{A_2}, \dots, r_{A_n}, m_p)$ and sends P_n to the verifier.
10. V generates its proof P'_n and compares it with the received P_n . If these two values are identical, V believes that all the tags are simultaneously present.

The main properties of this protocol are:

- The order independent exclusive-or operation \oplus is adopted to calculate the grouping proof. So all the data transmissions are order independent, which saves time and reduces the failure rates.
- The random number r_p generated by PT is included in m_{p_i} instead of transferring it in plain text, thus the security is improved.
- As an added privacy measure, A_i can change every time after each successful authentication, making it difficult to trace the tags.
- A_R and A_{PT} are added in the messages between Reader and PT, which indicates the destination of this messages, thus the tags will not misunderstand these messages.
- The reader waits the response from all of the tags, and sends them together in a bundle to PT, which calculates all the m_{p_i} and sends them

back altogether. Hence the efficiency of this protocol is better than others which use serial steps, although the reader and PT require larger memories for temporary data storage.

If the reader has enough computation power, and the operations of PT are merged into the reader, the protocol can be further simplified as shown in Fig.4.2. Suppose this protocol is used in a goods checking system. There may be no need to unpack the good to check the quantities and the integrity can be confirmed by the backend processing system.

4.3 Comparison of Protocols

All the protocols use tags with similar restricted computing and a verified timer to guarantee the time period of verification, which are necessary to prove the simultaneous coexistence of the tags. The following table shows the comparison of the protocols mentioned in this paper:

4.3.1 Reading Order Independent Operations

The protocol proposed in this paper generates grouping proof $P_n(r, m_p)$, in which $m_p = MAC_{X_p}[r, r_p, m_{A1} \oplus m_{A2} \oplus \dots \oplus m_{A_n}]$. The main computation is the exclusive-or operation, which is independent of the reading order of T_i . The reading order independence is the main advantage of this protocol comparing to other grouping proof protocols.

4.3.2 Length of Grouping Proof

In our reading order independent protocol, the proof length is fixed, because the \oplus function is used to combine each result from the individual tags. No matter how many tags are included in the group, the result would be the

Table 4.1: Comparison of Protocols

	J. Saito's	S. Piramuthu	Generalised yoking proof	Anonymous yoking proof	ROI with PT (fig 4.1)	ROI with computation power (fig 4.2)
Reading order	dependent	dependent	dependent	dependent	independent	independent
Length of group	increase with tags number	increase with tags number	increase with tags number	increase with tags number	fixed	fixed
Efficiency	4	2	2	2	3	1
Privacy	No ID transmitted	Permanent ID transmitted	Permanent ID transmitted	No ID transmitted	Temp ID, hard to track	Temp ID, hard to track
PT necessary	Yes	No	No	No	Yes	No
Reader with Computation Power	No	Yes	No	No	No	Yes

same length. It is more efficient than the concatenation. A concatenation of individual proofs can be observed by the attackers and make a guess of the scale of the group.

4.3.3 Efficiency

Besides the reading order independency, our protocol provides batch response in both directions. Hence, the proposed protocol should offer better efficiency than others. In the table, I show the efficiency with numbers, number 1 means

the most efficient one, number 4 means the least efficient one. J' Saito's protocol [71] used time stamp and the authentication involves PT, every message needs to be transferred among these 3 parties, considering that the time used in data transmission between tags and readers are much longer than the computation time, I list J' Saito's protocol [71] the least efficient. The reading order independent protocol with PT is listed number 3 since it also requires communication among three parties. The other grouping proof is listed number 2 because the reading order independent protocol without PT is more efficient and listed as number 1.

4.3.4 Privacy

In Fig.3.6 and Fig.3.7, the ID of tags are sent in plaintext, which could compromise the privacy of the tags. Since Before the execution of these protocols, the reader is ignorant of the identities of tags, and running these protocols will reveal them to it. Fig.3.8, Fig.4.1 and Fig.4.2 do not transmit any ID of the tags. On concern of the privacy in our protocols, A_i changes every time after a successful verification. This adds the difficulties to the attacker to trace the tags.

4.4 Conclusion

As this chapter described, reading order will be a practical problem of previous RFID grouping proof. We compares the existing grouping proof mechanisms and proposes a reading order independent protocol to improve the security and efficiency. In our proposed protocol, all the previous discussion on the security and privacy requirements of the applications are considered. we uses MAC computation with secret keys X_p and X_{A_i} shared between the Verifier, Reader

and Tags to defend against the replay attack. We also use the temporary IDs to keep the privacy. In the comparison forms we made, it is clear that our protocol has certain advantages over the previous protocols.

The most possible future improvement of the reading order independent protocol is the ' \oplus ' operation we used in the proof. ' \oplus ' is an operation with weak security property. It is good to find another operation, which also have the reading order independent property, and this operation is more secure than the ' \oplus ' operation.

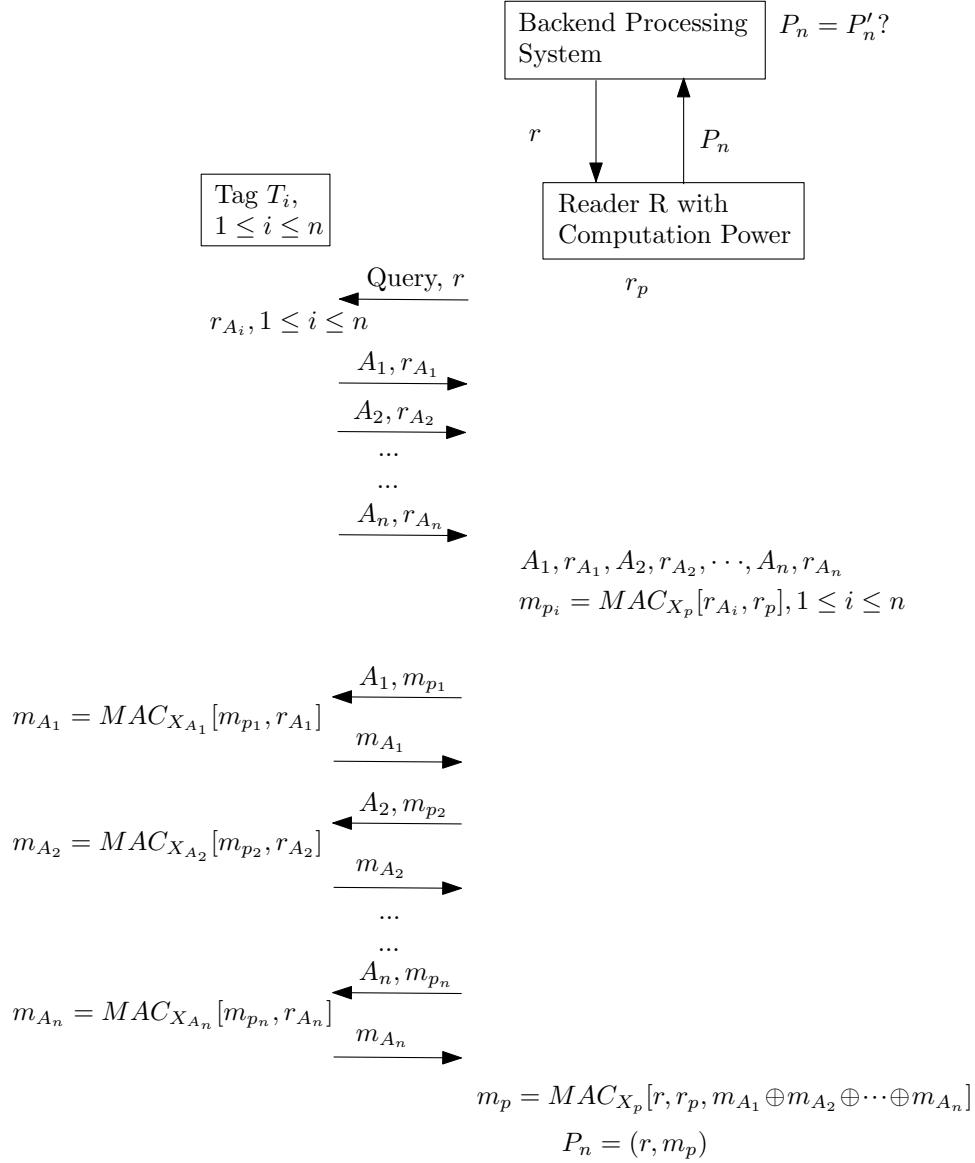


Figure 4.2: Reading Order Independent Grouping Proof for RFID Tags: Reader with Computation Power

Chapter 5

Select-response Grouping Proof Protocol

In this chapter, we discuss another scenario of grouping proof. In some application of grouping proof, real time verification of grouping proof is needed since the failure of the verification demands a real time action. However, the grouping proof which is produced by the yoked secrets of tags offers insufficient information. In this chapter, we propose a new mode of grouping proof, which is called select-response grouping proof, to meet the demands in the scenario concerned. The chapter is structured as follows: The first section introduces the supposed scenario and why the previous grouping proof scenario is inadequate. Then we introduce our new protocol from the prerequisite conditions to the step by step details. We also give analysis of our protocol at the end of this chapter. It is a cooperated work with Mr. Yuanhung Lien, I have contributed on the protocol design and analysis.

5.1 Introduction

Examining the previous grouping proof scenario, the verifier is supposed to be offline, which is the important reason why the grouping proof is necessary: the grouping proof offers the evidence of the completeness of a group of tags for

the verifier to check at a certain time in the future. However, in some practical applications, for example, on the assembly line, if the grouping proof is used to check the integrity of certain parts, failure of generating a grouping proof would desire an immediate action, otherwise the incomplete group of parts may cause unpredictable trouble in the next assembly steps. In the cases the incomplete groups need immediate actions, the online verifier is necessary, but the grouping proof has its own problems which limit its usage in the scenario mentioned. We give detailed analysis of those limits in the following section.

5.1.1 Problems of “Yoking Proof” Protocols

Yoking proof, i.e. the linkage of secrets, is a creative idea. Thus the grouping proof mentioned in the previous chapters all yoke the secret of the tags together. However, the yoking proof protocols also have suffered certain problems, which also comes from its property of linking proofs together.

1. In the scenario concerned in this chapter, the immediate action is desired when a failure happens. The main problem of the general grouping proof is that when the verification fails, it cannot find out the reasons of the failure. The verifier is not capable of identifying where the failure comes from. The reason can be missing an element in the group, transmission error, faulty tags or anything else. For the missing elements in the grouping proof, the verifier is also unable to point out which tag(s) caused the failure. Without this essential information, the choice of further actions coping with a grouping proof failure is very limited.
2. In the recent work of Burmester [59], it has been pointed out that the

tags in the yoking proof do not (and cannot) check each other's computation. It is not only an undesirable waste of resources in many practical applications, but also could be characterized as the Denial of Service vulnerability. If a malicious tag is involved in the yoking proof, the whole proof result will be contaminated and the whole group of tags cannot be verified.

3. In case of an incomplete group, not until the proof has been transmitted by the reader to the verifier, there is no way for the verifier to notice it in advance, which is not efficient. Also the grouping proofs mentioned in [49, 64, 71, 84] all generate the grouping proof dynamically, which is necessary for defending the replay attack. Thus the verifier needs to generate the grouping proof in real time for the verification. This added considerable computation load for the verifier.

In general, the procedures of grouping proof always terminate within a certain interval of time to avoid the replay attack. Simultaneous presence of tags is guaranteed only if a correct proof is completed before the timeout. Therefore, using response from one tag as an input for another tag is not a necessary condition in the group verification.

In the next two sections, we propose a new idea of the verification process of grouping proofs, in which the verification process completes under a time bound without adopting the mechanism of "yoking proof". Our proposed protocol possesses several benefits such as collision-free, reading order independent, and the missing tags identification properties. Hence, most of the three problems mentioned above will be solved. In addition, the overall security is improved.

5.2 Select Response Grouping Proof

5.2.1 New Idea of Grouping Proof

In the Bolotnyy and Robins protocol [49] mentioned in chapter 3, there is no explicitly discussion on how the reader knows which is the first and special tag T_1 , the querying sequence of the group tags, the total tags number in the group, or when to send the final response a_n to T_1 to close the chain. Apparently, the reader needs to get the related information of the group before performing the group verification. This will be a very inconvenient process for the verification if the information of a certain group has to be stored in readers in advance. Moreover, it is impossible to forecast when and where a particular reader will perform a verification process. Hence, the most reasonable way is that the reader will get the information of a certain group from the verifier when a verification proof of that particular group is required.

Aforementioned protocols about “grouping proof” for RFID tags all assume that the verifier had knowledge of each tag in the group before the verification process. In general, a reader queries each tag in a predefined sequence and computes a proof for the verifier. Hence there is no information for the reader to judge neither the completeness, nor the redundancy (outside tags) of the group in the proposed scheme. In fact, in some attack scenarios, the readers are not treated as trusted entities, so they are not supposed to acquire much knowledge of the tags. This problem can be solved if we use the reader as an interface between the tags and the online verifier. The readers actively call and find the tags. Since the verifier is assumed to possess knowledge of each tag in a group, the actual computation is done at the verifier’s end. During the process of select-response, the information of absent tags can be recorded

Table 5.1: Notations of Chapter 6

V	The Backend Processing System(BPS), Verifier
R	Reader
GID	Group identifier
ID	identifier of tag i
S_g	Group secret
A, B	Grouping random number
P_{AB}	Grouping Proof of tag A and B
xP_{AB}	Expected value of P_{AB}
P_n	Grouping Proof of n tags
xP_n	Expected value of P_n
$r_i, r_{i,j}, r_A, r_B$	Random numbers
ID_i	Identifier of Tag T_i
S_i	Secret of Tag T_i
$H(m)$	Hash function of message m .
$MAC_{x_i}(m)$	Hash function of message m .

and the immediate action can be taken even before the whole group has been called. For example, if an important part is missing, after several calls from the reader, the absence of this single response could trigger the fault-action mechanism since the whole group is pointless without this part. Of course we can make options that the group is traversed by reader's call anyway, when a report which fully covers all the missing parts is desired.

For reader's convenience, we put the table of notation before the description adopting them.

To achieve the goal we described, we propose a novel protocol. We named this kind of protocol as the "Select-Response Protocol (SRP)" and defined it as the tags of a grouping proof are required to respond for the reader/verifier in a predefined sequence. Under such verification process, instead of passively waiting responses from some tags, the verifier/reader actively sends command to a specific tag and asks for a response. On the other hand, some previously

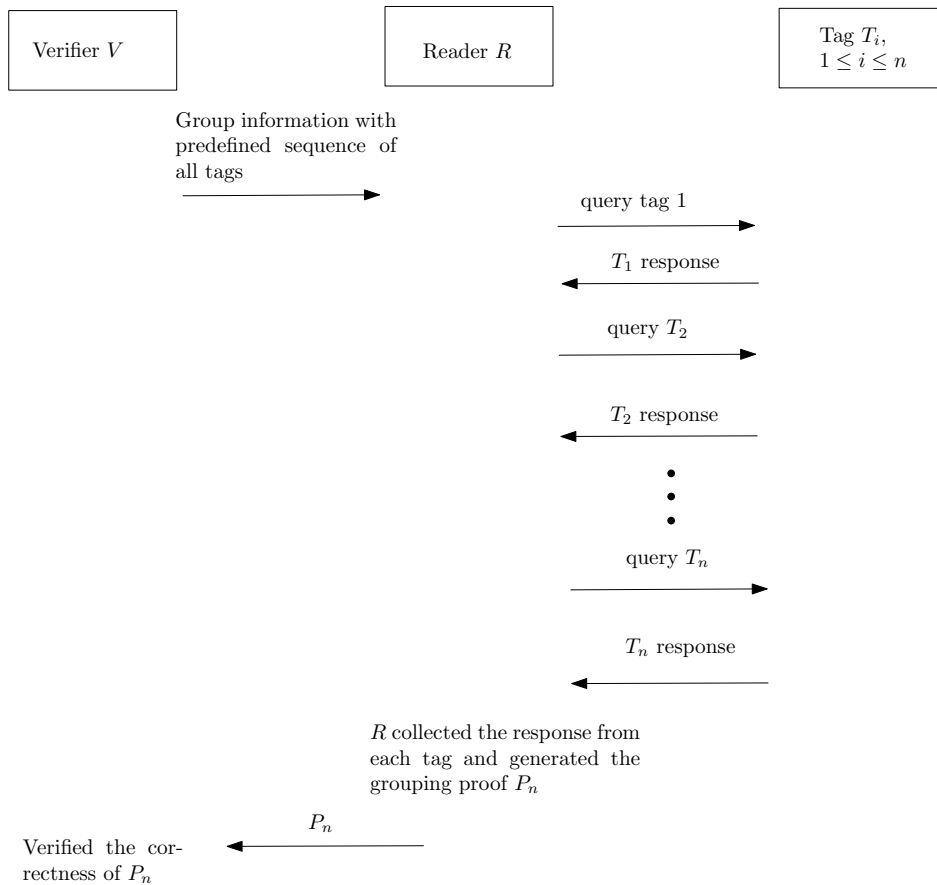


Figure 5.1: The Concept of SRP with Querying Sequence by Batch

proposed grouping proof schemes [49, 64, 71, 84] uses a broadcast grouping request to ask tags of the designated group to randomly respond to grouping proof. We distinguish such kind of protocols as “Random-Response Protocol (RRP)”.

We categorized the SRP in two classifications as in Figures 5.1 and 5.2, respectively. In Figure 5.1, the concept of “SRP with querying sequence by batch” is described. The RFID reader is assumed to be always honest, and get the information of verification with predefined sequence of all tags (by a batch file) from the verifier. The reader then collects each tag’s response

and generates the grouping proof P_n for the verifier to verify the correctness of a group. The verifier had knowledge of verification of the group including the query sequence of each tag when it receives the grouping proof P_n from the reader. The verifier only needs to acquire the information of all tags by following a predefined sequence from the database, calculates the expected grouping proof xP_n , and then verifies the xP_n with P_n . Hence, the speed of the verification process will be improved. In Figure 5.2, the concept of “SRP with querying sequence by demand”, the RFID reader is still assumed to be always honest. The verifier is kept online and responsible for every tag’s query command transmission. Under this protocol, the reader obtains the querying sequence from the verifier by demand and sends the response of each tag further to the verifier. Hence, the reader is not required to generate the grouping proof but just acts as a transparent interface between the verifier and tags. The verifier regenerates the response of each tag from the database and compares it with the tag’s response from the reader. The verification is passed only if the results of comparison are equal. Otherwise the computation overhead of the verification process will be declined evidently.

In the Bolotny and Robins protocol[49], there is no explicit discussion on how the reader gets the querying sequence of the group tags. This protocol can be classified as the “SRP with querying sequence by batch” if the reader got the querying sequence from the verifier before performing the verification process. In the following section, a “SRP with querying sequence by demand” is proposed and the concept of protocol design is explained.

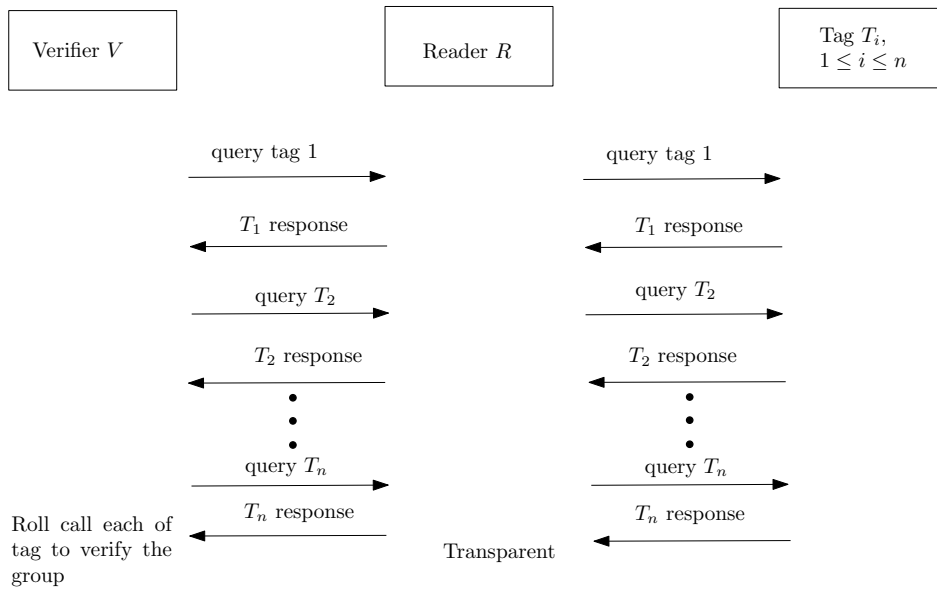


Figure 5.2: The Concept of SRP with Querying Sequence by demand

5.3 Protocol Design

In this section a “SRP with querying sequence by demand” is proposed and the explanation of the protocol design is provided. Finally, we introduce the practical arrangement of the Group Identifier (GID) in the Unique Identifier (UID) format of RFID.

5.3.1 Protocol Description: SRP with querying sequence by demand

Assuming that the group contains n tags denoted as T_1, T_2, \dots, T_n . Each tag T_i is assigned a unique identifier ID_i , a tag secret S_i and with a group secret S_g stored in its memory. Each tag has the ability to compute a hash function and respond to a specific roll call signal from the reader. The reader has to verify itself with the verifier V before obtaining the information of verification from V as well as when sending back the respond data from each tag. The

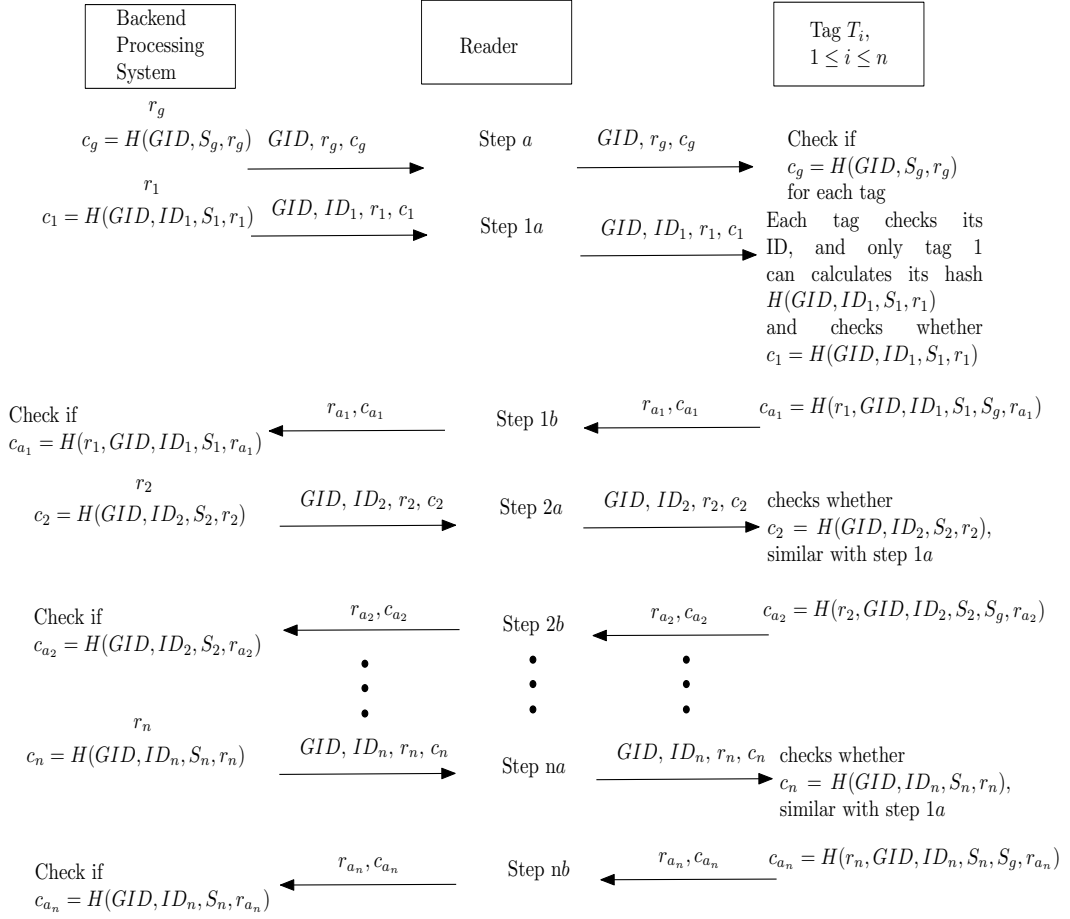


Figure 5.3: Select-Response Grouping Proof Scheme for RFID Tags

verifier V knows all secret information assigned to tags. We also assume that the reader R and the verifier V communicate through secure channels. The overall protocol is shown in Figure 5.3, and each step is described below.

1. Initial Setup

(1) User decides how many tags will be included in a group, and the reader will get the Group Identifier (GID) and the group secret parameter S_g used to prove membership in a group. The Backend Processing System (BPS, verifier) stores these values (for each tag) in a database entry (ID_i, S_g, S_i). We also assume that reader and the verifier communicate through the secure channels.

The GID and S_g will be written to each tag through the reader.

(2) Each tag has a individual tag secret S_i and a random number generator.

Each tag is also able to compute and compare the hash function $H(\cdot)$.

(3) The verifier V stores these values (ID_i, S_g, S_i for each tag) in a database entry.

2. Verification Procedure

During the verification of a group, the verifier is actively involved in the verification procedures namely, the verifier needs to be online during the procedure of verification. The proposed protocol will be described according to the sequence of messages exchanged. When the verifier V performs the verification of a group, it actively involves in the process by generating the querying sequence and requests each tag for a response. Since V knows all the information related to a group, it is not necessary for V to follow a specific order when it calls the tags. In fact, the verifier V can call the tags in any order it wants.

Step a: The verifier V generated a fresh random number r_g and calculated the hash value $c_g = H(GID, S_g, r_g)$, and then broadcasts GID, r_g, c_g to wake up the tags through the reader. Each tag will check whether $c_g = H(GID, S_g, r_g)$, to verify that the message actually came from a legal group entity.

Step 1a: The verifier V calculated the hash value $c_1 = H(GID, ID_1, S_1, r_1)$ and then sent GID, ID_1, r_1, c_1 to the T_1 through the reader. Each tag checks its ID, and only T_1 with correct match. Hence, T_1 calculates its hash $H(GID, ID_1, S_1, r_1)$ and checks it with the received data c_1 in order to verify that the message came from a legal verifier V .

Step 1b: If the checked result in step 1a is successful, then, T_1 generates a fresh random number r_{a_1} and calculates the hash value $c_{a_1} = H(r_1, GID, ID_1, S_1, S_g, r_{a_1})$ and then sends r_{a_1}, c_{a_1} to the verifier V through the reader R to authenticate the message actually comes from tag T_1 . The verifier V will check whether $c_{a_1} = H(r_1, GID, ID_1, S_1, S_g, r_{a_1})$.

From Step 2a: Repeat from **Step 1a** to **Step 1b** on each tag.

If all responses that V gets from the selected tags are correct, then the grouping proof verification is done successfully. Otherwise the verification process can be terminated immediately whenever any selected tags does not response correctly.

5.3.2 Design Explanation

In case that several groups in the same reading zone, the verifier/reader can employ GID to specify which group of tags it wants to talk to. Please note, other groups of tags are still in the sleep mode so that the interference problem can be greatly reduced. In other words, the group secret S_g is the evidence of a verified group member for a tag. A tag will respond only if the interrogating reader has the correct GID and S_g .

Hence, the efficiency of verification in multiple group application scenarios will be improved.

(1) In Figure 5.3, we introduced the Group Identifier (GID) to the RFID tags. Apparently, introduction of GID is not a necessary condition to perform the Select-Response protocol. In other words, all of the previous protocols

can still perform the Select-Response protocol only if the verifier keeps an online operation. However, the usage of GID and group secret S_g has some benefits. Firstly, in case of several groups are in the same reading zone, the verifier/reader can employ GID to specify which group of tags it wants to awake. Please note, the other groups of tags are still in the sleep mode so that the interference problem can be greatly reduced. Secondly the group secret is not only the evidence of a verified group member for a tag, but also its protection against a clandestine scan (i.e., the illegal reader secretly queries group's tags and collects corresponding responses). The tag only responds after it checks the interrogating reader has the S_g indeed. By this way the tags will only respond to the reader knows their GID and S_g and the privacy is preserved against inventorying attack.

(2) If it needs to perform the missing tags identification. For instance, if T_i is missing, the verifier will never receive the correct response (if any) from tag T_i until timeout. The verifier only records the ID_i of the missing tag, and then continues the verification.

(3) Random numbers r_g, r_i and r_{a_i} always refresh in each verification procedure. All the messages $c_g = H(GID, S_g, r_g)$, $c_i = H(GID, ID_i, S_i, r_i)$ and $c_{a_i} = H(r_i, GID, ID_i, S_i, S_g, r_{a_i})$, have included the random numbers. This feature is designed to defend the replay attack.

5.3.3 Practical Arrangement of the Group Identifier (GID) in the UID Format of RFID

As we mentioned in the introduction chapter, RFID tags can be distinguished based on either their frequency of operation (HF or UHF) or power techniques (active, passive or semi-active). The first generation EPCglobal standard for RFID tags (work on UHF band) was proposed in 2002. Later, the Generation

2 (Gen-2) standard was proposed to meet the requirements around the world [35]. The International Standards Organization (ISO) has approved the Gen-2 protocol as an amendment to its ISO/IEC 18000-6 standard in 2006 [6]. ISO 18000-6 requires that each tag should be uniquely identified by a 64-bit UID and be set permanently by the IC manufacturer. Figure 5.4 shows the UID format of the ISO 18000-6 and its compatibility with ISO 15693 working on the HF band is shown in Figure 5.7.

The UID comprises:

- The 8 Most Significant Bits(MSB) from bit-57 to bit-64, shall be E0
- The IC Manufacturer code, from bit-49 to bit-56, is assigned according to ISO/IEC 7816-6.
- A unique serial number bits, from bit-1 to bit-32 in ISO 18000-6 and from bit-1 to bit-48 in ISO 15693, is assigned by the IC manufacturers.
- From bit-33 to bit-48 in ISO 18000-6 are Reserved for Future Use (RFU) and set all to zero.

In order to be compatible with ISO 18000-6 or ISO 15693, the user can separate n bits from RFU field (from bit-33 to bit-48) or from IC manufacturer serial number, respectively, and named it the Group Identifier (GID). The GID field indicates which group the tag belongs to. The GID field has to be set all zero when the tag does not belong to any group. Tables 4 and 5 show the modified the UID format of the ISO 18000-6 and 15963 accordingly.

MSB		LSB	
b64-----b57	b56-----b49	b48-----b33	b32-----b1
'E0'	IC Mfg Code	RFU set all '0'	IC manufacturer serial number

Figure 5.4: Unique Identifier of the ISO 18000

MSB		LSB	
b64-----b57	b56-----b49	b48-----b1	
'E0'	IC Mfg Code	IC manufacturer serial number	

Figure 5.5: Unique Identifier of the ISO 15963

Introduce the Group Identifier to the RFID tags of ISO 15693

ISO 15693 requires that each tag should be uniquely identified by a 64 bits Unique Identifier (UID), which shall be set permanently by the IC manufacturer. Figure 5.5 shows the UID's format of the ISO 15963.

If the 48-bit serial number is required, the n -bit GID field can be added as Figure 5.7 and 5.6.

5.4 Protocol Analysis

In this section, we carry out the security and performance analysis of the proposed protocol, "SRP with querying sequence by demand", of grouping proof for RFID tags.

5.4.1 Security

Comparing to the previous grouping proof protocols, the proposed protocol improves security from the following prospective.

- (1) Mutual Authentication. As the protocol shows, both the tag's and

MSB				LSB
b64-----b57	b56-----b49	b48----48-n+1	b48-n-----41	b32-----b1
'E0'	IC Mfg Code	GID	RFU set all '0'	IC manufacturer serial number

Figure 5.6: Modified the UID Format of ISO 18000-6

MSB				LSB
b64-----b57	b56-----b49	b48 ---b48-n+1	b48-n-----b1	
'E0'	IC Mfg Code	GID	IC manufacturer serial number	

Figure 5.7: Modified the UID Format of ISO 15693

reader's side would authenticate each other.

(2) If it needs to perform the missing tags identification. For instance, if T_k is missing, the backend processing system will never receive the correct response (if any) from tag T_k until timeout. backend processing system only records the ID of the missing tag, and then continues the verification.

(3)The random numbers r_g, r_i, r_{a_i} always refreshes in each verification procedure. All the messages, $c_g = H(GID, S_g, r_g)$, $c_1 = H(GID, ID_1, S_1, r_1)$ and $c_{a_i} = H(r_i, GID, ID_i, S_i, S_g, r_{a_i})$, have included the random numbers used. This feature is designed to defend the replay attack.

5.5 Conclusion

Chapter 5 proposes a different grouping proof structure from yoking proof, namely "Select-Response" grouping proof protocol. Different from the previous grouping proofs, which generally have the RFID group of tags act like a relay team, each tag receives and passes the message one by one. The final proof has each of the tags' secret linked together in certain ways. The select-respond protocol is running in a way like this: the verifier behaves like a

captain calling his soldier from a name list. The tags are like soldiers, they are not answering until being asked. From its list and answers from the tags, the verifier knows exactly who is absent. It can choose to take immediate action in case of tag absence even before the whole verification process is finished. Because of this new mechanism, our protocol has improved the efficiency and flexibility of the previous yoking proof protocols. This protocol has practical value to many application which demand the immediate action in case of verification failure. However, the select-response protocol can only identify missing tags, it cannot identify the mixed outsiders. That means, for applications require the purity checks, like pharmacy production applications, they require the material to be highly pure. Select-Response protocol can not distinguish the outsider tags, this problem will be solved in chapter 7.

Chapter 6

Subgrouped Frameworks for Grouping Proof Protocols

This chapter gives a framework of dividing large groups of RFID tags into smaller groups. This framework can be used to any previously proposed protocols. The value of this framework lies on the fact that the subgroup division is done during the anti-collision process. After the subgroups have been divided, any grouping proof can be adopted inside the subgroup. To make our proposal a complete protocol, we give an example of grouping proof inside the subgroups, which is also inspired by the previous work of select-response protocols. In this chapter, the reason and necessity of dividing large groups into smaller subgroups are interpreted in the first section. Then we introduce the anti-collision algorithms and why we choose dynamic binary tree anti-collision algorithm to assist the subgroup division. In the third section we give detailed explanation of our subgroup division framework. Then we also give a grouping proof inside the subgroups to complete the protocol. It is a cooperated work with Mr. Yuanhung Lien, I have contributed on the original idea, the paper frames, protocol design and analysis.

6.1 Introduction

In this introduction section, we give the reason why the subgroup division is able to deal with the communication errors in the grouping proof protocols for group of large amount of RFID tags.

6.1.1 Communication Errors in RFID System

Although the aforementioned protocols enriched the knowledge and understanding of grouping proof, the feasibility to extend their applications to a large number of tags still needs more consideration. Given the fact that the RFID systems rely on a simple radio interface, there is a possibility that the communication is interrupted, e.g. by interference, noise, signal shielding or indeed loss of tag power as this is dependent on the received signal strength and the alignment and separation of the tag and reader antennas, which could pose a major challenge. As the core idea of the relevant “yoking proof” protocols is to link secrets of the tags together for the generation of the final proof, which brings heavy communication load between the tags and reader. Any communication interruption taking place between tags and the reader will ruin the whole verification process. The issue of communication interrupt is more serious in the grouping proof of a large number of tags.

6.1.2 The Idea of Subgroups

As described in the previous subsection, the probability of interruptions during the verification process is greater when there are a large number of tags in a group. If the large group can be divided into several subgroups, each with a small number of tags and the integrity of each subgroup is checked individually, the verification process will be stopped at an earlier stage in the case

of any error. In case of the radio communication errors, proofs of successfully verified subgroups can be reused by the reader. Therefore the reader does not need to perform the whole verification process again for all subgroups. An additional consideration is that, the proposed verification process must prevent the attacker providing fake tags in order to fool the reader. Moreover, since the algorithm is designed to overcome communication interruption during verification, the time interval between verifications of subgroups must be limited so that the overall proof completes in an acceptable time period. An optimization strategy is to count the number of tags in the subgroup in advance of the verification process. If the number of tags is different from the expected number, it is meaningless to start the verification. In our proposed protocol, the number of tags in each subgroup is counted just after the completion of the anti-collision algorithm performed in each subgroup.

Inspired by of the Select-Response grouping proof in Leng et al.'s work [83], a variant of the Select-Response “yoking proof” with subgrouping mechanism is proposed in this paper. Segmenting of the large group into smaller subgroups is aided by the anti-collision algorithm, specifically, the dynamic binary tree anti-collision algorithm. The introduction of the anti-collision algorithm is given in the section 2. Exploitation of the binary tree search anti-collision algorithm is interpreted in the section 3. In section 4, analysis of the subgrouping yoking proof is proposed, and the conclusions and future works are presented in the last section.

6.2 Anti-collision Algorithms

When multiple tags are present within the reading zone of a reader, the reader broadcasts the request command to all the tags. Upon receiving the command

message, all the tags send their responses back to the reader. Their responses will collide on the radio communication channel, thus the reader is unlikely to receive any meaningful signal from any of them. This is called the “Tag-collision Problem”. The ability to resolve this problem is crucial for the performance of the RFID systems which many tags so that the anti-collision algorithms are often treated in a proprietary manner by the companies. Basically the anti-collision algorithms can be categorised into Aloha based algorithms and tree-based algorithms [34]. Aloha based algorithms have a single tag response at a time by dividing the time into slot units. A tree-based algorithm creates a tree by splitting the group of colliding tags into two subgroups and traverses the tree, makes further subgroups until the reader finally recognises the identification of tags without collision. Although high performance can be achieved via Aloha-based methods, they may not function as well as binary tree searches in high tag density environments. Since the more the tags are present, the more collisions could happen in the small time slots which prevent the reader from reading any tag. In the scenario considered in this paper, the yoking proof is generated for groups with large numbers of tags in which the subgroup mechanism is advantageous, and the tree-based algorithm is utilised.

During the tree traverse process, the tags will be sub-grouped automatically and the tree based anti-collision algorithms are used to facilitate the subgrouping of the RFID tags. In our paper, we introduce one of the tree-based anti-collision algorithms, the Dynamic Binary Tree Algorithm [22], to illustrate the natural integration of the anti-collision algorithm and the subgrouping action for groups with large numbers of RFID tags.

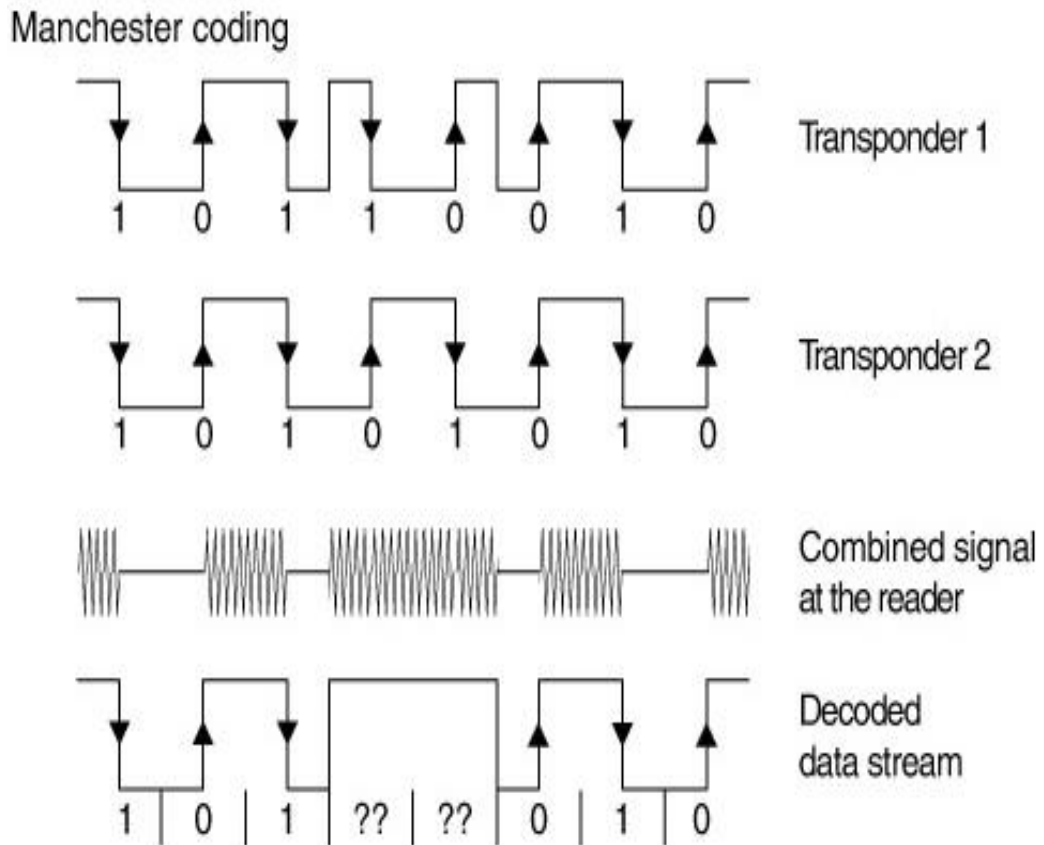


Figure 6.1: Collision Bits Identified According to Manchester Coding

6.2.1 Binary Tree Algorithms

In Manchester Coding, the value of a bit is defined by the change in level (negative or positive transition) within a bit window. Logic 0 is coded by a positive transition and logic 1 is coded by a negative transition. The transitions are also used for clock recovery, and so the non-transition state is not permissible during data transmission and is recognised as an error.

The reason why tree-based anti-collision algorithms use Manchester Coding is that the different bits of two signals will add up and produce a non-transition state. As shown in Figure 7.1, The 100% Amplitude Shift Keying modulation

In the binary search procedure described in Figure 6.2, both the search criterion and the identifiers of the tags are always transmitted at their full length. From a thorough analysis of the binary tree anti-collision algorithm, one can discover that a part of the transmitted data in the command and response is redundant. In Figure 6.3 the bits shown as shaded are all redundant. If the redundant bits are not to be transmitted, an optimised algorithm is produced, namely, the dynamic binary tree anti-collision algorithm. Instead of transmitting the identifiers with the full length, only the search criterion and the complementary bits are transmitted. The tags are informed of the number of subsequent bits by an additional parameter (NVB = number of valid bits) in the REQUEST command. Figure 6.4 illustrates more details on the request sequences of a dynamic binary tree anti-collision algorithm on the basis of a similar example given in Figure 6.2. The redundant bits are not transmitted and the over all efficiency is improved.

Downlink	REQUEST NVB=0	REQUEST NVB=2 10	REQUEST NVB=3 100	REQUEST NVB=4 1000	REQUEST NVB=6 100010	Select
Uplink	1XXXXX1	XX1X1	X1X1	1X1	1	
Tag 1	1001101	01101	1101			
Tag 2	1000111	00111	0111	111		
Tag 3	1000101	00101	0101	101	1	1000101
Tag 4	1101111					
Tag 5	1010101	10101				
Tag 6	1101101					

Downlink: Reader->Tag Uplink: Tag-> Reader

Figure 6.4: An Example of a Dynamic Binary Tree Anti-collision Algorithm

6.3 Subgrouping Model of Grouping Proof Protocol

We will consider an application scenario in which a specific group involves a large number of tags and that a verification proof is requested to prove that all the tags are simultaneously present. A reader R is connected with a host machine (verifier V) and communication between them is through a secure channel. The verifier V is a trusted backend server with a database that stores all the related information of the tags for the verification process. Each tag in a specific group has a fixed unique identifier (ID), a temporary identifier ($A_{j,i}$) and possesses enough computing resources to calculate a MAC [39] functions. The tags also have some writable memory reserved for data storage. The reader R and the tags are able to perform the dynamic binary tree anti-collision algorithm.

The notations used in our proposed protocol is given in Table 6.1.

6.3.1 Dividing Subgroups

According to the description of the dynamic binary tree anti-collision algorithm in section 2.2, each tag responses to its own identifier corresponding to the starting bits of the REQUEST command. A few prefix bits of the tag's ID are employed to facilitate the subgrouping process and are referred to as the subgroup identifier in this paper. For instance, if the first three prefix bits are chosen as the subgroup identifier, eight subgroup identifiers (with length=3, starting from 000 to 111) are obtained. We denote ID_{s_j} as the subgroup identifier of the subgroup S_j , and $T_{j,i}$ as the i th tag of the subgroup S_j . The architecture of the subgrouping and grouping proof is depicted in Figure 6.5. The procedure of subgrouping is described below.

Table 6.1: Notations

V	Verifier
R	Reader
\oplus	XOR operator
S_j	Subgroup j
ID_{S_j}	ID of subgroup S_j
n_j	the tag quantity in a subgroup S_j
$T_{j,i}$	Tag i of subgroup S_j
$A_{j,i}$	Temporary IDs of tag $T_{j,i}$
r	Random number generated by V
r_j	Random number for Subgroup S_j .
$r_{j,i}$	Random number generated by tag $T_{j,i}$
$X_{j,i}$	Symmetric secret keys of tag $T_{j,i}$
MAC	Message Authentication Code
$MAC_X(m)$	MAC of m using key X
$H(m)$	Hash function of message m .
P_j	Grouping proof of subgroup S_j
P	Grouping proof of the whole group

1. The user decides the number of tags in the specific group. The reader sends a query command to each tag of the group, collects their corresponding responses (identifiers), and then sends them back to the verifier.
2. According to the number of tags in the specific group, the verifier decides how many subgroups should be created dynamically. First, the verifier decides the length of the subgroup identifier according to the number of tags and starts to distribute each tag to its corresponding subgroup. If some subgroups contain more tags than others, the verifier may divide this subgroup further into smaller subgroups. For instance, if the verifier notes that the 6th subgroup (with the subgroup identifier $ID_{s_6} = 110$) contains a relatively large number of tags, it will extend the length of the subgroup identifier one more bit (i.e., using $ID_{s_{6-0}} = 1100$, $ID_{s_{6-1}} = 1101$) in order to further divide this subgroup to smaller subgroups and so on.

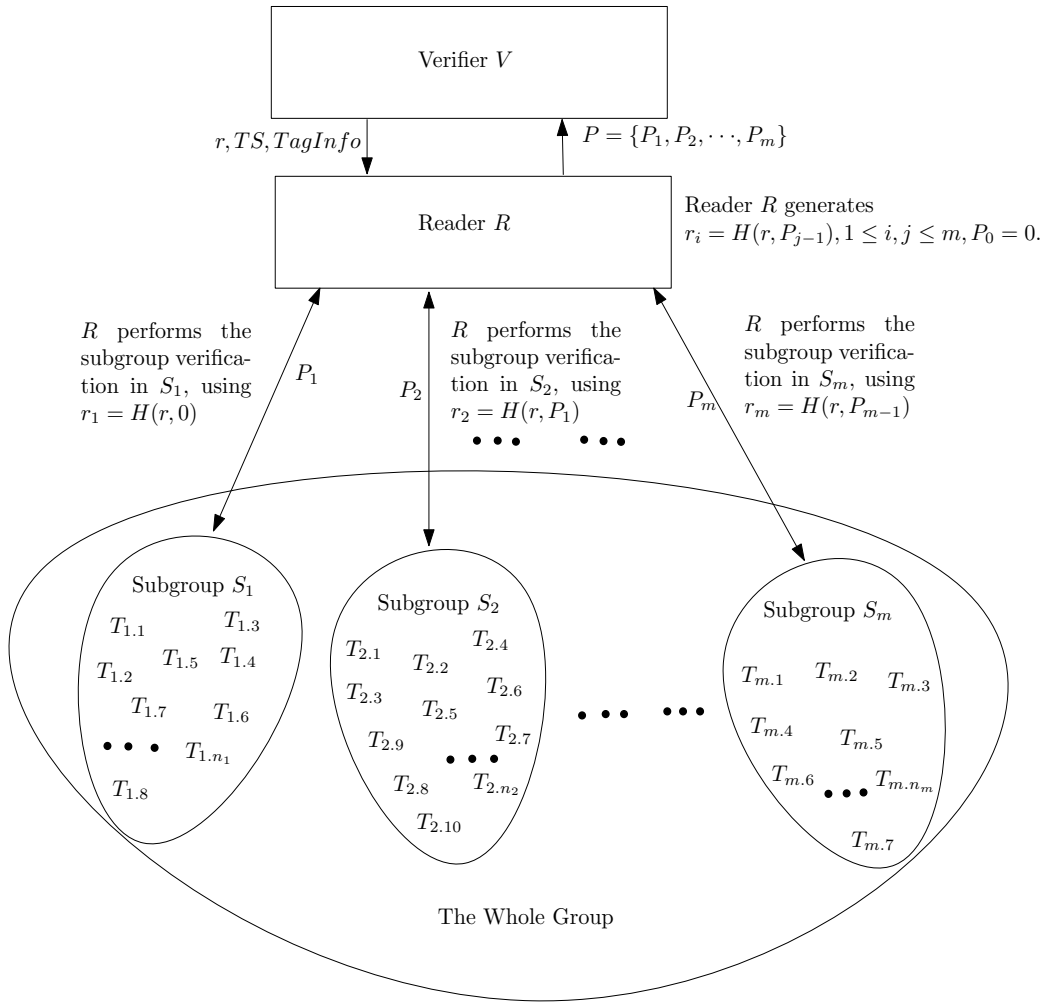


Figure 6.5: The Architecture of the Subgroup Grouping Proof Model

3. The verifier saves the subgroup information to the database. Such information will be sent to the reader when the verification of the subgrouping grouping proof is requested.

6.3.2 Verification of the Whole Group

The verifier sends the information back to the reader through a secure channel when verification of the whole group is required. The architecture of the subgrouping grouping proof is depicted in Figure 6.5 and the overall verification

procedures are described below.

1. The reader R gets the relevant information (random number r , time stamp TS , and all tag information TagInfo) from the verifier V to perform the verification procedures.
2. The reader R generates the random number $r_1 = H(r, P_0)$, with the initial proof $P_0 = 0$, and performs the verification procedures in the first subgroup S_1 . The reader R will produce a proof of the subgroup P_1 after the verification process is completed.
3. The reader R uses the P_1 as a seed to generate $r_1 = H(r, P_1)$ and then sends r_1 to the subgroup S_2 for subgroup verification.
4. Repeating steps (2) and (3), the reader R conducts the verification procedures in each subgroup in a sequential manner. In general, the reader R generates the $r_j = H(r, P_{j-1})$ to perform the subgroup verification in the j th subgroup S_j , where $1 \leq j \leq m$ and $P_0 = 0$. The P_{j-1} is the grouping proof of subgroup S_{j-1} and will be described in the next subsection. In order to guard against the replay attack, each r_j contains a fresh random number r from the verifier V for every verification request for the whole group. In addition, each r_j is also dependent on the proof from the previous subgroup P_{j-1} . Hence, a chain is established by linking all of the subgroup yoking proofs.
5. After harvesting all the subgrouping proofs $P_j, 1 \leq j \leq m$, the reader R concatenates them together to generate the final grouping proof $P = \{P_1, P_2, \dots, P_m\}$. P is then sent to the verifier V for the verification of the whole group.

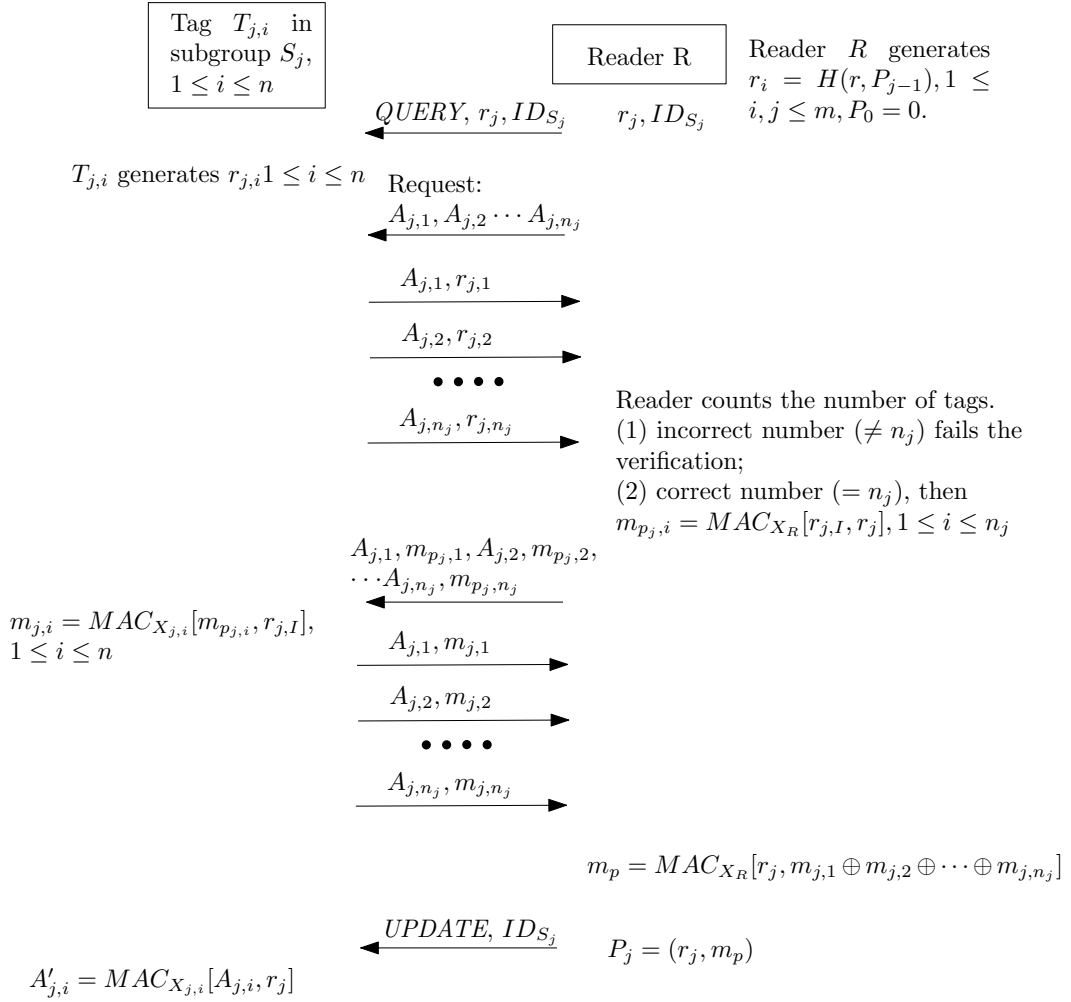


Figure 6.6: Grouping Proof inside Subgroups

6.3.3 Verification in Subgroups

Figure 6.6 shows the details of the yoking proof for subgroup S_j , which is assumed to have n_j tags from $T_{j,1}$ to T_{j,n_j} . This proposed protocol is modified from the order independent protocol provided by Lien et al. [84], and adopts the concept of the Select-Response protocol given by Leng et al. [83]. The detailed explanation is as follows.

1. The reader R broadcasts the *QUERY* command with r_j and ID_{s_j} to all tags.
2. Each tag compares the subgroup identifier ID_{s_j} with the first few prefix bits of its own ID . Only the tags in subgroup S_j use the r_j as a seed to generate the random number $r_{j,i}$, $1 \leq i \leq n_j$. After receiving the *RESRQ* (*RES*ponse *Re*quest) command with tag response sequence $\{A_{j,1}, A_{j,2}, \dots, A_{j,n_j}\}$ from the reader R , only the tags of subgroup S_j with the same temporary identifier $A_{j,i}$ will send back $A_{j,i}$ and $r_{j,i}$ to the reader in a predefined time interval to avoid collisions.
3. The reader R counts the number of tag response $(A_{j,i}, r_{j,i})$ from the subgroup S_j . If the number of responses is not equal to the expected value, the verification of the subgroup S_j fails. Otherwise, the reader R calculates the Message Authentication Code using secret key x_R , $m_{p_{j,i}} = MAC_{x_R}[r_{j,i}, r_j]$, $1 \leq i \leq n_j$.
4. The reader R concatenates the $\{A_{j,1}, m_{p_{j,1}}, A_{j,2}, m_{p_{j,2}}, \dots, A_{j,n_j}, m_{p_{j,n_j}}, n_j\}$ together and sends it back to all the tags of subgroup S_j .
5. The tags of subgroup S_j calculate the $m_{j,i} = MAC_{x_{j,i}}[m_{p_{j,i}}, r_{j,i}]$, $1 \leq i \leq n_j$ and sends $(A_{j,i}, m_{j,i})$ back to the reader R in a predefined time interval to avoid collisions.
6. The reader R calculates the $m_p = MAC_{x_R}[r_j, m_{j,1} \oplus m_{j,2} \oplus \dots \oplus m_{j,n_j}]$ and generates the subgrouping proof $P_j = (r_j, m_p)$ for subgroup S_j .
7. The reader sends the *UPDATE* command with the subgroup identifier ID_{s_j} to the tags of subgroup S_j .

8. The tags of subgroup S_j update their temporary identifier $A'_{j,i} = MAC_{x_{j,i}}[A_{j,i}, r_j]$ to prevent the replay attack.

6.3.4 Quick Retrieval of the Identified Subgroups

The primary aim of this paper is to find a way to reduce the effect of communication interruptions and errors on grouping proof calculations for large numbers of tags. With the help of subgrouping, this aim can be achieved in a simple way. This protocol has options to reuse the identified subgrouping proofs during the valid time defined by TS . The reader R can set the quick retrieval mode for the application. Under this mode, if an interruption occurs, instead of restarting another round of yoking group verification for all the subgroups, the reader R will only check the updated temporary identifiers $A'_{j,i}, 1 \leq i \leq n_j$ of the previously proved subgroups. If all the $A'_{j,i}$ are correct, the existing proofs of these subgroups will be re-used and the reader R just continues with the verification of the remaining subgroups. To guarantee that the temporary identifiers $A'_{j,i}$ of the subgroup S_j are used only once as the evidence, the reader R will send the *UPDATE* command again after counting the number of tags in the subgroup S_j . So new temporary identifiers, $A''_{j,i} = MAC_{x_{j,i}}[A'_{j,i}, r_j]$ will be generated. If the reader R does not receive the expected temporary identifiers, the quick retrieval mode will stop and the normal verification procedure will start again.

6.4 Analysis of the Subgrouping Protocol

In this section, we carry out the security analysis for the yoking proof of our protocol and the quick retrieval model.

6.4.1 Analysis of the Grouping Proof Subgrouping Protocol

The main properties of this protocol are:

- This protocol adopts both the “yoking proof” ideas from Juels [40] and the select-response idea from Leng et al. [83]. The “yoking proof” is used to link the subgroups together and for the verification of each subgroup, the Select-Response idea is employed. By actively selecting the subgroups and their tags, this protocol is safe from interference from irrelevant tags (i.e., tags that are not selected), and has the ability to identify a missing tag. In this hybrid design, the core spirit of “yoking proof” is maintained and its performance penalty is overcome by adopting the Select-Response mode in the subgroup verification.
- Using the Dynamic Binary Tree Anti-collision Algorithm to subgroup and select tags, the whole verification process is collision free without installation of any new algorithms because the pre-installed anti-collision algorithms will perform the subgrouping task.
- The number of subgroups is dynamically decided according to the tag density in the whole group, which is more efficient and flexible than predefining this statically.
- The exclusive-or operation (\oplus), which possesses the order independent property, is adopted in our protocol to calculate the grouping proof. All the data transmissions are order independent and the verification process is speeded up.
- As an added privacy measure, the temporary identifier $A_{j,i}$ changes every

time after each successful verification. making the tracing of tags more difficult.

6.4.2 Security Analysis of the Quick Retrieval

One security focus of the design is on the quick retrieval mode. A possible attack against the quick retrieval of the identified subgroups is the “Interrupt and Replace” attack in which an attacker interrupts the verification of subgroup S_j and then replaces an authentic tag $T_{j,i}$ in an earlier subgroups with a fake tag. Our proposed protocol requires that the reader R demands for the updated temporary identifier $A'_{j,i} = MAC_{x_{j,i}}[A_{j,i}, r_j]$. Because the fake tag does not have the correct secret key $x_{j,i}$, it cannot generate the updated temporary identifier $A'_{j,i}$. The attacker also cannot reuse the temporary identifier $A'_{j,i}$ since it will be updated again after the reader R sends an *UPDATE* command.

Another possible attack is that the attacker cuts off the *UPDATE* command. The reader R still supposes that the tags have been updated and demands for the updated temporary identifier. However a synchronisation error occurs and the quick retrieval mode will stop.

6.5 Conclusion

In this chapter, we introduced a framework which divides a big group into smaller subgroups through the anti-collision algorithm process. This frame can be applied to any grouping proof which uses the tree-based anti-collision algorithm, since the anti-collision algorithm is the fundamental algorithm which is implemented in every RFID system, so no new algorithm is required. With the help of anti-collision algorithm, the subgroups are divided dynamically

without adding much computing overload to the system. All the previously mentioned ‘grouping proof’ can be adopted inside the authentication of subgroups. Thus this framework can be applied to all the previously mentioned grouping proofs. The primitive motivation of subgroup division is to improve the performance of grouping proof and cope with the radio communication errors and interruptions for very large amount of tags. In case of communication failure, we proposed the quick retrieval mechanism to reduce the high failure rates. The quick retrieval mechanism checks the IDs of the tags which just fulfilled their parts of the grouping proof before last error occurs. Then the proof process can be continued instead of being restarted.

In that chapter 6, we also presents a complete grouping proof protocol which uses the framework to divide a large group of tags into several subgroups. For this complete protocol, we adopts all the previously discussed ideas in the grouping proofs. The ‘yoking proof’ idea is implemented to link the subgroups together and inside each subgroup, and the ‘Select-Response’ mode is used for subgroup verification, we can call the final grouping proof a hybrid grouping proof.

The subgroup ideas have many benefits in management and performance. Considering the number of the tags in the whole group and application demand, what is the best way to divide the subgroups to meet the balance of subgroup computation cost and the radio communication failure rate is an interesting topic waiting for more works. Finding the best way to divide the subgroups to meet the balance of subgroup computation cost and to decrease the radio communication failure rate is also an important issue. Moreover, the updated synchronisation problem between tags and the reader/verifier may cause a quick retrieval mode failure. The above problems will be considered

seriously in the future works.

Chapter 7

Group Authentication of RFID Tags Using Bit-Collision Patterns

This chapter propose an innovating idea of using collision pattern as the proof to authenticate the RFID groups. Collision pattern is a property of the RFID groups, by my particular design, it becomes gives a neat solution to authenticate the whole group in one challenge response session. It is a cooperated work with Gerhard Hancke, I have contributed on the original idea and protocol design.

7.1 Introduction

As we have introduced in the first chapter, the potential RFID applications like supply chain and pervasive computing are becoming practical, it is likely that high-volume tags will be required to provide identification and security services in certain applications. Methods to implement additional security functionality, while keeping tags cost low, are desirable in this scenario and as a result implementing ‘lightweight’, or minimalist, security mechanisms for the RFID environment has become a very active research area [11, 9, 51].

Alternatively, if tags are used to track security sensitive or valuable items, the tags' capability to limit damages resulting from possible theft could justify the cost of implementing more sophisticated security mechanisms. One of the primary security concerns in a tracking system would be that attackers could create clones of existing tokens. This could allow them to replace valuable items in shipments with forgeries, which would appear like genuine products to the system's readers. In this case, authenticating the tags could aid in detecting and discouraging the use of fake tags. Conventional authentication mechanisms would require the reader to challenge each individual tag and wait for a response from the tags. On the tag side, it needs certain cryptographic computation with its limited computing resources, which is heavily limited by the cost of the tag. Taking the time to run an authentication protocol with each tag might not be a practical option too, especially if there are a large number of individual tags that need to be read within a short period of time.

Basically, the transaction time, i.e. the message transmission and the time intervals between them, should also be taken into account when designing secure protocols or RFID system, but this limitation is often overlooked in favour of processing or power restrictions. What is more, the reader has to interact with every single tag, the transaction time of the whole group will be the single tag's transaction time multiplies the tags number. At last, we need to add the anti-collision time of the group of tags, because singling out every individual tags will take significant time for the groups with a large number of the tags. In systems where the transaction time is limited, it would be ideal if multiple tags could be authenticated simultaneously instead of sequentially.

In this chapter, we propose an authentication protocol that allows for multiple tags in a group to be authenticated using a single response, thus achieving an execution time comparable to that of a transaction with a single tag. Comparing to the grouping-proofs, which are introduced and explained in the previous chapters, the protocol we proposed in this chapter is similar to those protocols in certain ways. Those protocols verify a group of tags as a whole entity, emphasising the relation between tags inside the group. However, all those protocols deal with the tags individually, which would inevitably encounter the performance problems mentioned previously. The new protocol proposed in this chapter also authenticates the whole group in a different way, not anything similar with the previous proposed ‘Yoking Proof’ or ‘Select-Response’ mode. However, we still categorise our protocol as a kind of grouping proof, meaning that it authenticates a whole group of tags as a single entity.

The aim of the proposed protocol is that authenticate the group is both complete (no tag is missing) and pure (no tag has been replaced or stranger tags get mixed in). The proposed protocol is practically feasible as the physical communication properties required are already implemented in current RFID standards, such as ISO 14443 and ISO 15693 (ISO 18000-3), and could be implemented in other tag technologies with minor modifications to the communication logic. We also provide a security analysis of our protocol that describes the attack success probability for several attack strategies. RFID is widely used in Automatic Identification and Data Capture (AIDC) systems, which will lead to the potential applications of supply chain and pervasive computation mentioned in chapter 1. The identified items can often be grouped into logical groups, e.g. a set of spare parts for a specific car or different

medicines constituting a patient’s prescription. In secure tracking applications the system often has to verify that all the items in the group are still together and that the items are genuine, i.e. that none of the items have been lost or replaced. This involves the system authenticating each tag, which is essentially the verifier issuing a challenge to each tag and the tag replying with a response that only it should know. For a large number of items the total time needed to run an authentication protocol with each tag might become impractical as the transaction time in RFID systems are generally limited. For example, RFID-tagged items might be rapidly moving down a conveyor belt in the assembly line or contained in a truck driving past the reader, so the tags will only be in the reader’s field for a short period of time (a few seconds, at most). As a result the time the reader has to communicate with each tag is restricted. A more efficient way of authenticating a group of tags is therefore needed.

7.2 Pseudo-Simultaneousness of the Previous Grouping Proofs

Although the previous protocols give the evidence that some RFID tags are scanned simultaneously. None of these proposals solve the transaction time problem. Some grouping proofs are often described as methods for proving that multiple tags were read “simultaneously”, but in reality this only means that all the tags were queried sequentially within a specified time bound. We call it pseudo-simultaneousness. For example, in the original two-tag yoking proof, both tags are within the reader’s range at the same time, but the reader can only communicate separately with tag t_A and tag t_B , then the reader binds the secret information from two tags and claim that they are

read simultaneously. This separate communication channel is bound by the traditional way by which reader talks with tags. Although time bound in the grouping proof is absolutely necessary, otherwise no simultaneousness is guaranteed, but how to define the value can be intriguing. The time bound cannot be too loose to undermine the simultaneousness; it also cannot be too tight because each of the previous grouping proofs needs heavy traffic flows in a successful process. The traffic flow normally increases as the number of the tags increases. So these protocols need more time to finish in case of large groups. So the time bound should be an increasing function of the group scales. In the practical implementation, the verifier should calculate the time bound and tell the readers every time before launching a grouping proof. Not to mention how to deal the case in which a reader encounters unexpected groups. On the other hand, the reading failure is a common error in radio communication system, certain tags with undesirable position or weak signal zone might require more than one reading attempts, which would significantly increase the reading time.

Pseudo-randomness also causes the security vulnerability of the previous protocols. As shown in chapter 3, the replay attack, which successful breaks Juels' original yoking proof and Saito's grouping proof, actually take the advantage of breaking the sequential queries. Such attack will not be effective if all the tags could truly be authenticated simultaneously using only a single challenge-response sequence.

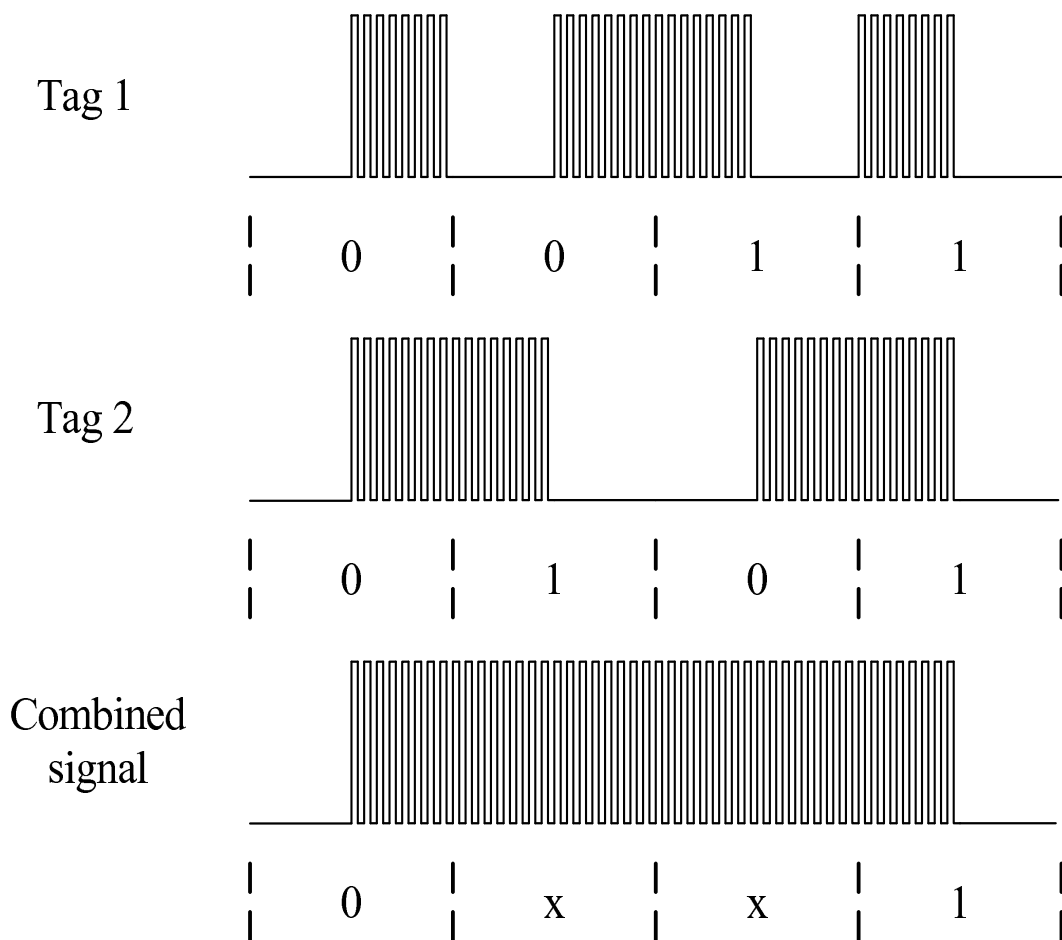


Figure 7.1: Constructing bit-collision patterns with Manchester code

7.3 Bit Collision

7.3.1 Compatibility of Current Standard

This would require that all tags transmit their responses at the same time, and fortunately some RFID technologies already allow this transmission for this scenario. For example, in systems adhering to the ISO 14443 [5] and ISO 15693 [3] (ISO 18000-3 [6]) standards multiple tags simultaneously transmit information to the reader during the anti-collision process. The tokens' responses are Manchester coded and they are synchronised to start transmitting

in the same bit period. This results in clearly defined bit collisions during certain bit periods, as shown in Figure 7.1. The reader therefore observes a bit pattern containing both collisions and non-collision bits. Not all RFID tags use Manchester encoding but some tags do, however still use communication symbols that could be used to construct bit-collision patterns. For example, EPC Class-1 Generation-2 (ISO 18000-6C) [35] tags encode their data as follows: a ‘1’ is represented by the signal staying high or low for the entire bit duration while a ‘0’ is represented by a signal transition edge after half the bit period has elapsed (either high-to-low or low-to-high). The two ‘0’ symbols are therefore identical to the symbols used in Manchester coding and can therefore be used in a similar way to construct a bit-collision pattern.

Readers interested in further details on RFID communication channels and different anti-collision methods are invited to read [26].

7.3.2 Current Application of Bit-collision

Bit-collisions have been incorporated into key exchange protocols [18, 30] and into security mechanisms providing confidentiality and privacy services by intentionally blocking tag responses to unauthorised readers [42] but have to date not been used in authentication schemes. In all these proposals it is assumed that if a collision is observed then the attacker cannot determine who has sent which bit symbol, e.g. which tag sent a ‘1’ and which tags sent a ‘0’. In 2007, Hancke [29] demonstrated that an attacker could in some cases deduce the responses from two tags contributing to a bit-collision pattern because of variations in the communication channel of passive tags. The chances of the attacker successfully using this method decreases, however, as the number of tags contributing to the collision pattern increases.

7.4 Using Bit-Collision Patterns for Group Authentication

Authentication protocols usually require a challenge-response exchange. Our scheme is based on the basic principle that if n tags in a group all transmit their authentication responses at the same time this will result in a verifiable bit collision pattern that represents the authentication response for the entire group. To explain this further, we need to define the bit collision operation formally. We denote the collision operation between two bit symbols, β and β' , $\beta, \beta' \in \{0, 1\}$ as $\beta \mathbb{M} \beta'$ and use x to indicate that a bit collision occurred. The results of $\beta \mathbb{M} \beta'$ is as follows:

β	β'	$\beta \mathbb{M} \beta'$
0	0	0
0	1	x
1	0	x
1	1	1

If two tags transmit 1 and 0 in the same bit slot, it will result in a bit collision x regardless of the values transmitted by any other tags, i.e. $1 \mathbb{M} x = x$ and $0 \mathbb{M} x = x$.

In our proposed scheme each tag t_i , $i = 1$ to n , that is part of the group contains an authentication state s_i . If all the tags transmit their authentication states simultaneously then, collision will happen in certain bit slot and this will result in a group authentication state S , i.e. $S = s_1 \mathbb{M} s_2 \mathbb{M} \dots \mathbb{M} s_n$. The group authentication state S can then be used to determine whether the group is both complete and pure.

Assuming that each tag in the group contributes at least one bit collision, and all tags cause an equal number of bit collisions, then a verifier can check for completeness of the group. The verifier simply counts the number of bit collisions that occur. If there are less than it expects, it knows that either a

tag is missing or a fraudulent tag has caused collisions at the same time as another tag that is present. If there are also a suitable number of bit positions where a collision does not occur than the verifier can check for purity. The verifier checks in which bit positions the collisions occur. If a bit collision is detected in a bit position where it was not expected the verifier knows that a fraudulent tag has failed to calculate in which bit position it should cause a collision or inadvertently caused an extra collision because it failed to guess a correct non-collision value. The verifier therefore knows that the group has been “contaminated”. For example, if we have four tags each contributing one collision to a group authentication state of length 8 the authentication process works as follows:

tags	correct	missing s_4	fake s_4
s_1	01000011	01000011	01000011
s_2	01001001	01001001	01001001
s_3	01100001	01100001	01100001
s_4	11000001	missing	01010001
S	$x1x0x0x1$	$01x0x0x1$	$01xxx0x1$

If the group is complete and pure, i.e. all the tags are present and have the correct state as shown in the left column, then the verifier would expect four collisions in bit positions 1, 3, 5 and 7 respectively. If the groups were not complete, for example s_4 is missing, there would only be three bit collisions, as showed in the middle column, and the verifier will detect that a tag is missing. Similarly, if one tag (s_4 in this example) was replaced with a forged tag transmitting the incorrect state it would be detected, as the verifier would notice that no collision occurred in bit position 1 but rather in bit position 4.

7.4.1 Group-Authentication Protocol

- **System operation:** The system consists of a number of nodes that track the progress of a group of items, i.e. a single package or shipment,

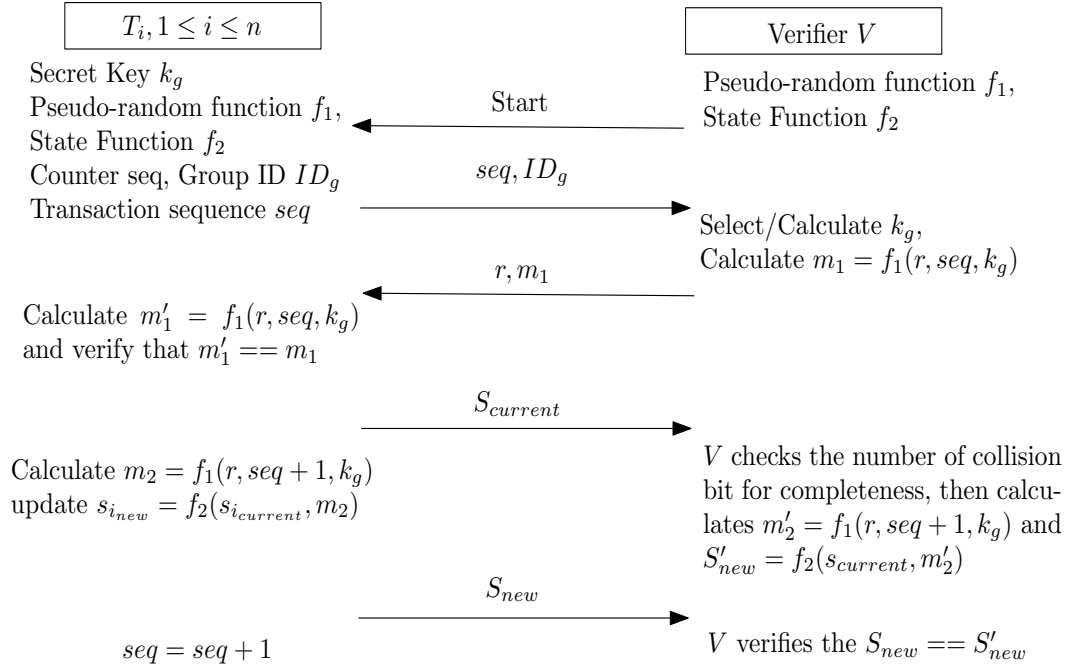


Figure 7.2: Description of the proposed group-authentication protocol

from its sender to the intended recipient. The sender, recipient and nodes could be controlled by different organisations, but a key management infrastructure is in place that allows the sender to distribute key material to the verifying nodes and the recipient, i.e. the sender can securely share information with the verifying nodes and the recipient. The group verifier, i.e. a node or the recipient, does not necessarily have the ability to share information with other verifiers and as a result it should not require knowledge of previous protocol runs between the group and other verifiers.

- **Security objective:** The purpose of our protocol is only to prove the completeness and purity of a chosen group to the recipient and intermediate verifying nodes, thus preventing a third-party attacker from

replacing or stealing items during shipping. The protocol will not help to prove these conditions to any third party, in other words, it does not provide non-repudiation of purity and completeness for anyone who does not trust the verifying node or recipient. The sender, recipient and verifying nodes are seen as trusted entities, who will not reveal key material or create fraudulent tags/groups.

- **Cryptographic primitives:** For the purpose of running the protocol the group of tags, sender, recipient and the verifying nodes share a dedicated secret group key k_g , a keyed public pseudo-random function f_1 and a public bit permutation function f_2 . In practice $f_1(m, k_g)$ would probably be based either on symmetric encryption, e.g. $\text{Enc}(m)_{k_g}$, or a hash function, e.g. $h(m, k_g)$.
- **Grouping process:** A group is made up of a number of related items labeled with RFID tags. A group would be a number of items physically packaged together as these must be interrogated together by a single reader. The sender is responsible for creating a legitimate group and initialising the tags. The sender will also send the group ID and a description of the items to the intended recipient, via a suitable communication channel, to enable the recipient to identify the contents. If required, tags can contain additional data, e.g. an item description or serial number, protected with a key shared between the sender and recipient, although this is beyond the scope of our protocol.

Our protocol proposal is shown in Figure 7.2. As stated previously, the tags and the verifier share a keyed public pseudo-random function f_1 . The tags and the verifier also share a state permutation function f_2 , which is described

in more detail in Section 7.4.2. Each tag t_i , with $i = 1$ to n where n is the number of tags, contains at least a stored current authentication state $s_{i_{\text{current}}}$, a common counter value seq , a common group identifier ID_G and a common group key k_g .

1. The verifier signals its intention to start the authentication protocol by transmitting a *Start* command. The *Start* command could also be used to narrow down the groups responding, by transmitting a value similar to the Application Family Identifier (AFI) used by EPC (ISO 18000-6C) and ISO 15693/18000-3 tags.
2. All the tags simultaneously respond with a group ID ID_G , counter value seq and the number of tags in the group n . Since all the tags are transmitting the same data no bit collisions should occur. If a bit collision occurs only in seq the verifier knows that a tag has become unsynchronised, while a bit collision in both seq and ID_G indicates that tags from another group have also responded. The verifier can use ID_G to select the correct group key from a pre-distributed database. It would also be possible to derive the group key using a master key and ID_G but doing so securely is beyond the scope of our protocol. If required, this step could be used for anti-collision, e.g. tag groups can reply in randomly chosen time slots (all tags in a group choose the same slot) and the verifier can retransmit the *Start* command until it learns a complete group identifier.
3. Next the verifier generates a random bit string r and calculates $m_1 = f_1(r, seq, k_g)$. It then transmits the group ID ID_g of the tags it wishes to authenticate, the bit string r and m_1 . All the tags in group ID_g now

calculate $m'_1 = f_1(r, seq, k_g)$ and verify that $m'_1 == m_1$, thereby essentially authenticating the verifier. The combination of the random bit string r and seq provides freshness for each protocol run, which prevents an attacker from replaying previous authentication responses.

4. If the verifier is shown to be legitimate the tags all transmit their current authentication state $s_{i_{\text{current}}}$, so that the verifier can learn the current composite group authentication state S_{current} . This is required for synchronisation between the group and the verifier as we assume that there is not necessarily a communication channel between verifiers that can be used to share the group's last known state. An exception is raised if no tags transmit their authentication state and the protocol information should be reported to the sender. This is to discourage the 'dummy' tag attack described within Case 4 in Section 7.5.
5. The tags then each update their authentication state using the state permutation function f_2 and $s_{i_{\text{new}}} = f_2(s_{i_{\text{current}}}, m_2)$, where $m_2 = f_1(r, seq + 1, k_g)$. Subsequently, each tag transmits $s_{i_{\text{new}}}$ resulting in a new group authentication state S_{new} .
6. In the meantime the verifier has calculated $S'_{\text{new}} = f_2(S_{\text{current}}, m'_2)$, where $m'_2 = f_1(r, seq + 1, k_g)$, so it can verify that all the tags updated their authentication state correctly by comparing $S_{\text{new}} == S'_{\text{new}}$, thereby effectively authenticating the group.
7. Finally, each tag increments seq and sets $s_{i_{\text{current}}} = s_{i_{\text{new}}}$. The verifier also reports the identifier ID_g , sequence number seq , the random bit string r and the new authentication state S_{new} to the sender. The sender learns the current location of the group, i.e. the group is in close proximity to

a known verifying node, and stores the additional information for audit purposes.

If $S_{\text{new}} \neq S'_{\text{new}}$ fails then an exception will be raised and the grouped items could be moved to a secure and controlled area where each one could be further investigated. Our proposal has the added advantage that it can authenticate any subset of the group or even individual tags using exactly the same protocol steps. For example, if the protocol is only run with one tag the verifier checks if $s_{i_{\text{new}}} = f_2(s_{i_{\text{current}}})$ to determine whether the tag is legitimate. In this case the two group states are conventional bit strings with no collisions. Similarly, the protocol could be run with a subgroup containing l tags each contributing c collisions. The verifier could use $f_2(S'_{\text{current}})$ to check whether S'_{new} contains lc collisions in the correct bit positions along with an additional $(2n - l)c$ correct non-collision values. The verifier can therefore run the protocol again with each item, or split the group into smaller subgroups, to look for malfunctioning or counterfeit tags. If during the check a tag is found that responds with $seq - 1$ after the *Start* command then the reason for the group authentication failing is likely to be technical, i.e. the tag failed to receive the ID_g, r, m_1 command and therefore failed to respond and update its state. In this case the protocol is run again with this tag, using the same r as used in the first group authentication run, to resynchronise the tag with the rest of the group.

If a verifier finds that a group has become desynchronised before it arrived the sender should be able to provide the verifier with all the previously used protocol information. The verifier then runs the protocol with the tags ‘left behind’, using the sequence numbers to determine the correct order in which to use previous bit strings r , until all the tags are again synchronised, i.e. until

all the tags have the same sequence number.

For our protocol we assumed that the recipient and nodes are trusted entities. If the protocol, however, is executed in an environment where the sender does not trust the verifying node it is possible for the tags to be verified directly by the sender. In this case the tags are initialised with a node key k_n and a group key k_g . The local verifying node uses k_n to authenticate itself to the tag in Step 3 but the tags use k_g to generate S_{new} . The node then sends seq , r , S_{current} and S_{new} to the sender who uses this information and k_g to verify that S_{new} is valid. In this variation the node cannot remove an item and create a replacement tag as it does not know the group key k_g .

7.4.2 Constructing S and f_2

The processes of initialising the group and updating the authentication state are essential to the correct operation of our protocol. We now examine in more detail how the initial authentication state for each tag and the state permutation function f_2 could be constructed.

Group initialisation: For the purpose of proving completeness and purity, at least one bit collision must be attributed to each of the n tags in the group. If this is not the case, the tags that cause no bit collisions could be removed from the group without affecting the group authentication state, thus the incomplete group might not be detected by the authentication. Each tag should contribute an equal number of bit collisions, which would allow the verifier to determine how many tags are missing, if the group is found to be incomplete. Consider a group containing two tags, A and B , contributing one collision each and two tags contributing two each, C and D . As the tags are not contributing the same number of bit collisions the verifier does not know

whether the group is missing two tags ($A + B$) or one tag (either C or D) if the group authentication state is found to be missing two bit collisions. A bit collision should also be paired with a bit value (non-collision) so that the bit swap operation in f_2 (see the following section) always changes the position of a bit collision, i.e. if a bit pair consisted of two collisions or non-collisions a bit swap would not cause a change in the collision pattern. If each tag contributes c bit collisions, which are each paired with a non-collision bit value, then the bit length of the group authentication state S and the individual tags' states s_i should be $2cn$. To meet all these criteria, we propose that the sender constructs the tag group authentication state as follows:

1. Choose the number of collisions c that each of the n tags in the group will contribute.
2. Create a state matrix M with n rows, each containing a tag authentication state vector of length $2cn$ and all values set to 0.
3. Randomly select c bit pairs, b_1, \dots, b_c , out of the set of cn possible pairs. In the first row set the first bit value of each chosen pair equal to 1. i.e. $M_{1,2b_i-1} = 1$ for $i = 1, \dots, c$. Remove the previously selected b_1, \dots, b_c from the set, choose another c pairs from the remaining $c(n-1)$ pairs and set the corresponding bit values in row 2 to 1 in a similar way as before. Repeat until all n rows contain c collisions. For example, if $n = 4$ and $c = 2$ the tags' authentication states are set as follows:

	Tag states	Choosing bit pairs
s_1	1000000000001000	1, 7 of (1, 2, 3, 4, 5, 6, 7, 8)
s_2	0010100000000000	2, 3 of (2, 3, 4, 5, 6, 8)
s_3	0000001000100000	4, 6 of (4, 5, 6, 8)
s_4	0000000010000010	5, 8 of (5, 8)
S	$x0x0x0x0x0x0x0$	

4. Load each tag with the group ID ID_G , group key k_g and set the sequence counter. It would be preferable if the initial value of the group's sequence number is randomly chosen as to not have the same sequence numbers repeat in authentication transactions with different groups.
5. Finally, the sender runs the group authentication protocol with the tags to randomise the values of the non-collision bits and the positions of the bit collisions before shipping.

State permutation function f_2 : f_2 should be chosen to satisfy the following property: $f_2(s_1) \text{ \& } f_2(s_2) \text{ \& } \dots \text{ \& } f_2(s_n) = f_2(S)$. The simplest way to do this is to create a permutation function using XOR, bit swap and shift operations. The bit swap operation switches the collision and non-collision value in a bit pair if the corresponding bit in the swap string is '1'. The shift operation bit rotates the entire authentication state (the bits wrap around). The bit swap and shift operations do not affect the number of bit collisions or the tag which is responsible for a specific bit collision, but these operations do change the positions of the bit collisions. Assuming that the initial state is created using the method described above we always shift the string in multiples of 2 (or in bit pairs) so that a pair always consists of one bit collision and one non-collision value. This ensures that a bit swap will always change a collision position. The XOR operation does not effect the number or positions

of the bit collisions even though it might change the underlying bit values contributing to the collision, i.e. if a, b, c are binary bits, then if $a \neq b$, $(a \oplus c) \wedge (b \oplus c) = x$. This property always holds because if $a \neq b$ then $(a \oplus c) \neq (b \oplus c)$, which results in $(a \oplus c) \wedge (b \oplus c) = x$. The XOR operation's primary purpose is to randomly alter the non-collision bit values.

In the protocol, f_2 will accept the results of a keyed pseudo-random function f_1 . If there are n tags, each contributing c collisions, then the length of the authentication states (s_i or S) is $2cn$ and there are cn bit pairs, which can be shifted between 1 and $cn - 1$ positions. The result of the pseudo random function can be parsed into three bit strings that define the state permutation, XOR string ($2cn$ bits), bit swap string (cn bits) and shift string ($\text{ceil}(\log_2(cn))$ bits). The pseudo-random function should therefore yield a result of length $3cn + \text{ceil}(\log_2(cn))$ bits. If a single output of f_1 is too short a longer pseudo-random number could possibly be created by running the function repeatedly using the previous result as input.

The following example illustrates how the state permutation function works. If we have four tags with authentication states equal to

s_1	01000011
s_2	01001001
s_3	01100001
s_4	11000001
S_{old}	$x1x0x0x1$

and the pseudo-random function yields a bit string

<u>0101</u>	<u>01</u>	<u>10100101</u>
Swap	Rotate	XOR

then we need to swap bit pairs 2 and 4, rotate right by one bit pair and XOR the result with 10100101.

	<i>Swap</i>	<i>Shift</i>	<i>XOR</i>
s_1	01000011	11010000	01110101
s_2	01001010	10010010	00110111
s_3	01010010	10010100	00110001
s_4	11000010	10110000	00010101
S_{new}	$x10xx01x$	$1xx10xx0$	$0xx10xx1$

From the example it can be seen that $S_{\text{new}} = f_2(S_{\text{old}}) = f_2(s_1) \text{ \& } f_2(s_2) \text{ \& } f_2(s_3) \text{ \& } f_2(s_4)$.

7.5 Security Analysis

The group authentication protocol proves that a group is complete and pure. After running the protocol the verifying node should be confident that none of the legitimate group tags have been replaced by an attacker or simply lost. The condition for the group being complete is as follows:

- the verifier must observe cn bit collisions in the tags' authentication states S_{old} and S_{new} , where n is the number of tags in the group and c is the number of collisions attributed to each tag.

If a tag is absent in the authentication, the bit collisions caused by that tag will not be present in the authentication state. If less than cn collisions are observed it therefore means that at least one tag must be missing from the group. An attacker wishing to remove an item, and by implication a tag, from the group might substitute legitimate tags with devices, which attempt to replicate the responses of the legitimate tags. For a group to be pure the following conditions must hold:

- the group must be complete.
- the bit positions of all bit collisions in S_{new} must be correct, i.e. these must occur in the expected bit periods.

- the bit values received during the bit periods where no bit collisions occur must be correct.

If an attacker tries to replace an authentic tag, it must ensure that his fake tag's bit responses still result in a valid S_{new} . In other words, without knowing the group key k_g the attacker must cause bit collisions in the same bit positions as the tag it replaced. The attacker must also guess the values of the remaining non-collision bits otherwise his substitute device would cause additional bit collisions in bit positions where none are expected.

We consider five attack cases where an attacker attempts to remove an item from a group of three or more items. In each case we state the probability p_a that the attack will not be detected. We assume that the attacker knows the value of the current group authentication state S_{current} , i.e. it observed S_{new} during the previous protocol run, but that it has no knowledge of the group key k_g .

Case 1 – Attacker uses a simple tag: The attacker's replacement tag functions as a normal tag, e.g. it has obtained empty tags of the same type. His replacement therefore adheres to the protocol rules and does not know what the other tags are transmitting. Although the attacker knows S_{current} this does not assist him a great deal in calculating S_{new} as it does not know which bit collisions are contributed by the tag it replaced, which bit values need to be transmitted to cause the bit collisions or what the updated non-collision values will be. The attacker would need to contribute bit collisions in the correct bit positions and transmit the correct non-collision values in order to not introduce extra collisions. To succeed the attacker essentially needs to guess the right bit values in all positions where other tags do not cause bit collisions. Let us consider an example where 4 tags each contribute 2 collisions.

The group authentication state therefore has a length of 16 bits and should contain a total of 8 collision bits. The attacker has a 1 in 2^{16} chance to guess the exact authentication state of the removed tag. However, if another tag causes a collision it does not matter what value the attacker guessed in that bit position as either a ‘0’ or ‘1’ will still result in a bit collision. So only the 2 bits that actually contribute bit collisions and the 8 bits that are non-collisions need to be correct as the other 6 bits would effectively be hidden by the bit collisions caused by the other tags. This means that of the 2^{16} possible bit permutations 2^6 would contribute to the right result. The attacker therefore has a 1 in 2^{10} chance of choosing a tag authentication state s_i that will result in the correct group authentication state S_{new} . The probability of an attack succeeding if the attacker removes $n_a \leq n - 2$ tags from a group of n tags, each contributing c collisions, can be calculated as follows:

$$p_a = \left(\frac{1}{2}\right)^{c(n+n_a)} \quad (7.5.1)$$

If the attacker replaces $n_a = n - 1$ tags the attack probability is given by

$$p_a = \left(\frac{1}{2}\right)^{2cn} \quad (7.5.2)$$

because the single legitimate tag left cannot contribute any collisions by itself and therefore the attacker would need to guess all the bit values correctly.

Case 2 – Attacker uses a quiet tag: The attacker’s replacement tag does not strictly adhere to the protocol but it does not know what the other tags are transmitting. In this case the attacker’s strategy is to only transmit bit values in the positions where it thinks the tag would need to cause a bit collision. The attacker’s tag stays quiet the rest of the time. The main benefit to the attacker is that it does not need to guess the correct values of the non-collision bits, as these are still transmitted by the other tags, and does therefore not

risk causing extra bit collisions. To succeed the attacker needs to select the correct bit positions to cause collisions and choose the right bit value that will result in a collision, i.e. the attacker needs to transmit a different symbol as the rest of the tags. The attacker first needs to guess which of the bit pairs contain a collision caused by the tag it replaced. Using the same example as in Case 1 ($n = 4, c = 2$), and assuming the group was constructed as explained in Section 7.4.2, the attacker has to choose one of $\binom{8}{2}$ possible position permutations for the two pairs of interest, where $\binom{8}{2}$ is the binomial function $\frac{8!}{(8-2)!(2)!} = \frac{8 \cdot 7}{2 \cdot 1}$. The attacker also has to guess in which bit position within the pair he needs to cause a collision and what the bit value must be to cause the collision, i.e. it has a $\left(\frac{1}{2}\right)^2$ chance to get the collision within each pair correct. In this example the attacker therefore succeeds with probability $\left(\frac{2 \cdot 1}{8 \cdot 7}\right) \cdot \left(\frac{1}{2}\right)^4$. In the case where the attacker removes $n_a \leq n - 2$ tags from a group of n tags that each contribute c collisions the probability of this attack succeeding can be calculated as follows:

$$p_a = \binom{cn}{cn_a}^{-1} \cdot \left(\frac{1}{2}\right)^{2cn_a} \quad (7.5.3)$$

If the attacker replaces $n_a = n - 1$ tags it must contribute a collision in every bit pair, since the single legitimate tag remaining cannot cause a bit collision by itself. The probability that the attack succeeds in this case is the same as given in Equation 7.5.2.

Case 3 – Attacker uses a smart tag: The attacker’s replacement tag does not adhere to the protocol and can observe what the other tags are transmitting. We assume that the tag can determine the bit value and prepare its response right at the start of the bit period, even though this might be unrealistic in practice as RF receivers usually integrate or sample over the entire bit period. The primary benefit is that if the attacker, i.e. the attacker’s

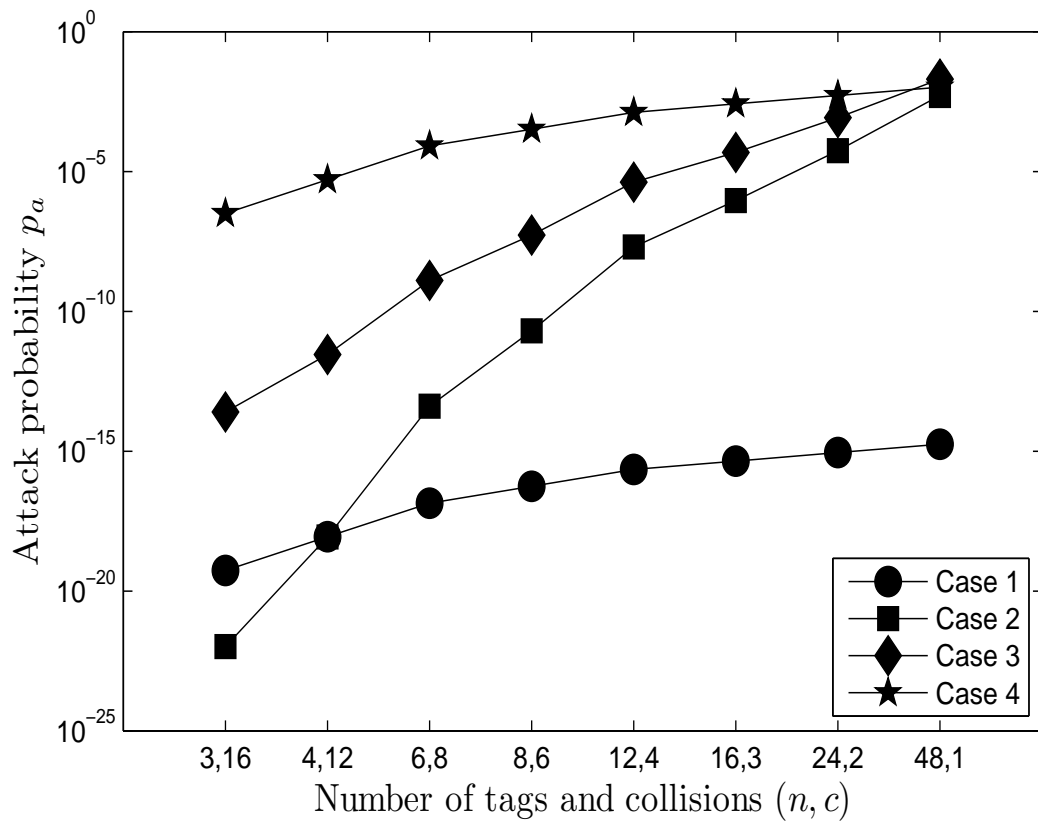


Figure 7.3: Comparative attack probability when the attacker replaces a single tag if the length of S and s_i is 96 (a realistic response length for both EPC (ISO 18000-6C) and ISO 15693/18000-3 tokens).

tag, wants to contribute a bit collision it knows the values the other tags are transmitting and therefore it simply needs to transmit the alternative value. The attacker can also observe bit collisions, which could help it to choose the bit positions in which it causes bit collisions. Obviously, the attacker would not need to guess the values of the non-collision bits as it could either choose not to transmit anything or simply learn the correct value from the other tags. Once again let us consider an example where a group, initialised as described in Section 7.4.2, contains 4 tags that contribute 2 collisions each. The attacker does not guess which bit pairs contain the relevant bit collisions

right away. Instead, it observes the first bit position of each of the bit pairs. If the first bit position is a collision it knows that it does not need to cause a bit collision in that bit pair as it is contributed by another tag. Taken that the bit swap is based on a random binary string it could be assumed that on average half the 6 collisions contributed by the other tags will be in the first bit position of the pair. As a result the attacker can eliminate 3 of the 6 pairs when choosing in which pair to cause a collision. The number of bit pairs the attacker has to consider therefore decreases from 8 to 5 and the number of possible permutations it has to choose from decreases to $\binom{5}{2} = \frac{5 \cdot 4}{2 \cdot 1}$. The attacker still needs to guess whether to cause the bit collision in the first or second position within these remaining pairs so it has a $\frac{1}{2}$ chance of contributing the collision correctly in each chosen bit pair. As a result the overall probability of the attack succeeding for this group example is $\left(\frac{2 \cdot 1}{5 \cdot 4}\right) \cdot \left(\frac{1}{2}\right)^2$. For a group with n tags, which each contribute c collisions, the probability of this attack succeeding if $n_a \leq n - 2$ tags are replaced can be approximated by:

$$p_a = \left(\frac{(cn + cn_a)/2}{cn_a} \right)^{-1} \cdot \left(\frac{1}{2} \right)^{cn_a} \quad (7.5.4)$$

If $cn + c$ is an odd number it can be rounded down, representing a better case for the attacker than if rounding up, to ensure that the binomial function input is an integer. In the case where the attacker removes $n_a = n - 1$ tags from the group the attacker has to contribute a bit collision in each bit pair but it can still observe the legitimate tag's response and therefore knows the correct bit value to transmit if it wishes to cause a bit collision. The attack success probability is therefore equal to

$$p_a = \left(\frac{1}{2} \right)^{cn} \quad (7.5.5)$$

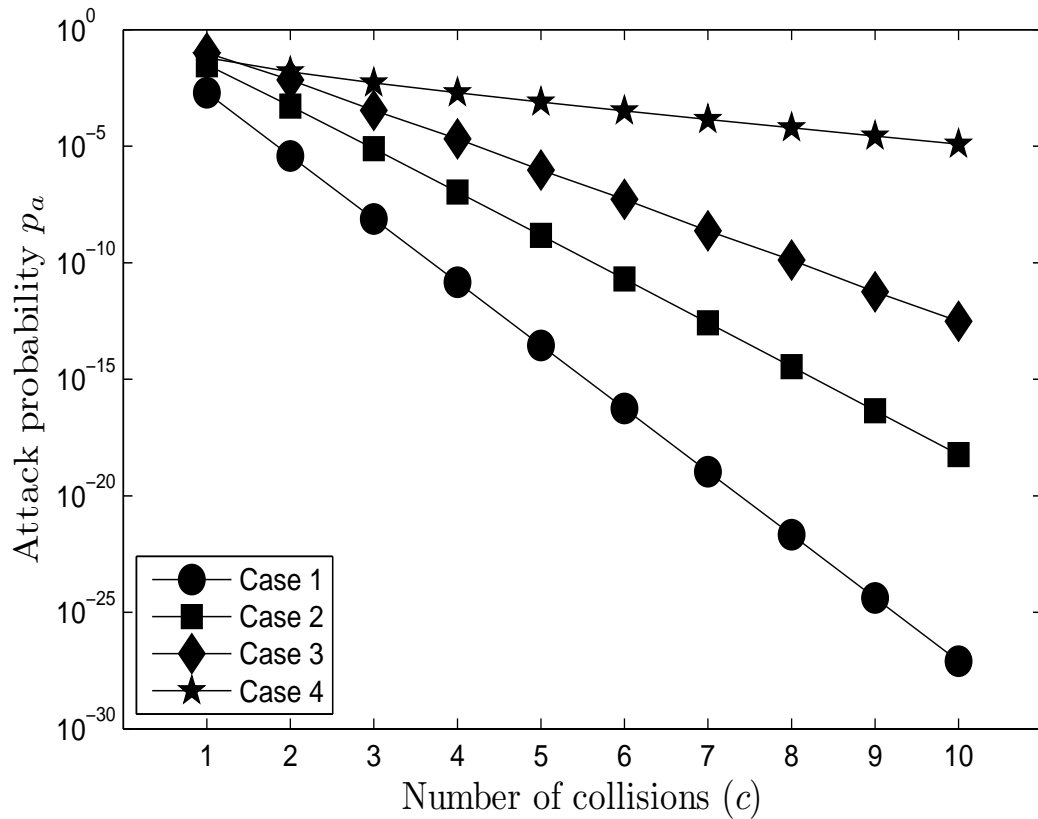


Figure 7.4: Comparative attack probability when the attacker replaces a single tag if the size of the groups stays constant and the number of collisions per tag increases ($n = 8$).

Case 4 – Attacker knows tag state and collision positions: In this case the attacker knows the current authentication states of the tag it wants to replace and at least two other tags. This allows the attacker to identify which bit collisions are contributed by the tag it replaces by comparing its state with the other tags' states and seeing what bit values differ, i.e. if the other two tags have the same bit value and the replaced tag's bit value differs it is contributing a collision at that bit position. This is truly a worst case scenario as this is unlikely to happen during normal operation of the system. The attacker can only eavesdrop the group state S , which as a whole does

not reveal much information about individual tag authentication states. As a result, an attacker would need to be in a position to observe $s_{i_{\text{new}}}$ in protocol runs between the node and individual tags during a checking routine following an error, although it was previously assumed that this process will be executed in a secure area. An attacker cannot read the state out of a tag as it would not be able to pass node authentication (step 3 of the protocol) without knowing the group key k_g . The attacker could execute a ‘dummy’ tag attack where the attacker’s tag responds with ID_g, seq, n during step 2 of the protocol and then records r, m_1 transmitted by the verifier during the next step. An attacker could then replay the r, m_1 data to the individual items in the real group, since the tags’ counter value is still equal to seq , and retrieve the individual tag authentication states. An attacker can also desynchronise the group by running the protocol with a single tag to increase its sequence number seq , but the verifying node would be able to resynchronise the tags as described in Section 7.4.1. Note that the attacker cannot replay the recovered responses to the node as the correct response in the next protocol run will be calculated with a new r and that the verifier should under normal circumstances raise an exception when no tags respond with their authentication states (thereby detecting the ‘dummy’ tag attack). For the simple tag scenario knowing the current tag state and position of the bit collisions does not benefit the attacker. It still needs to guess the bit pair rotation in addition to the non-collision values and the bit values needed to cause collisions as these are randomly changed by the XOR operation. The attacker’s best approach is still to randomly guess the tags’ individual authentication states and therefore the probability that the attack succeeds is still represented by Equation 7.5.1. In the quiet and smart tag scenarios the attack success probability increases. Since the

attacker knows the bit pairs in which his tag contributes bit collisions it only has to guess how many bit pairs the authentication state is going to rotate and whether the bit pair values will swap to know in which bit positions the collisions should be contributed in the updated group authentication state. For the general smart tag case, with $n_a \leq n - 2$, the probability of the attack succeeding becomes

$$p_a = \frac{1}{cn} \cdot \left(\frac{1}{2}\right)^{cn_a} \quad (7.5.6)$$

For the quiet tag scenario the attacker also needs to guess the bit value that will cause the bit collision, so the new attack success probability can be written as follows:

$$p_a = \frac{1}{cn} \cdot \left(\frac{1}{2}\right)^{2cn_a} \quad (7.5.7)$$

If the attacker removes $n_a = n - 1$ tags the attack probability is the same as previously described for this scenario in Case 2 and Case 3.

Case 5 – Attacker creates a new group: The attacker attempts to create a whole new group to hide the fact that it took an item or multiple items. In this case it has to guess all the bit collision positions and the values of the non-collisions within the updated group authentication state S . The advantages the attacker gains in Case 2 and Case 3 no longer applies as there are no legitimate tags to observe or rely upon to transmit the correct non-collision bit values. This means that the attacker needs to guess the bit position of the collision and the value of the non-collision bit position in each bit pair. It therefore has a $\left(\frac{1}{2}\right)^2$ chance of guessing each bit pair correctly. For a group of n tags, with c collisions per tags, the probability that the attacker successfully creates a new group is therefore given by

$$p_a = \left(\frac{1}{2}\right)^{2cn} \quad (7.5.8)$$

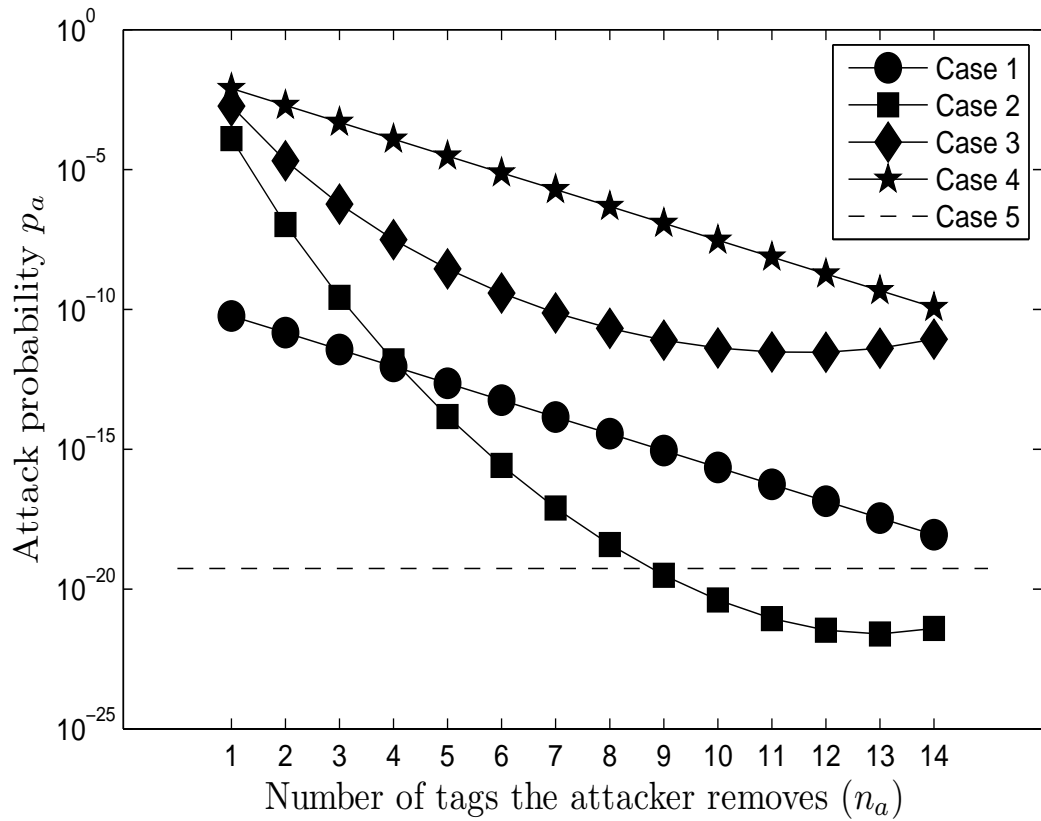


Figure 7.5: Comparative attack probability if the attacker replaces multiple tags in the same group ($n = 16, c = 2$).

The effects of the group size, the collisions per tag and the attacker removing multiple tags on the probability that an attack succeeds for the different cases are illustrated in Figures 7.3, 7.4 and 7.5 respectively. The probability for Case 4 in each figure was calculated using Equation 7.5.6 as it represents the best case for the attacker.

7.6 Conclusion

In this chapter, we have presented a group authentication protocol that verifies whether a group of RFID tags is both complete and pure. The protocol

allows for multiple tags to transmit their authentication responses simultaneously as it uses the resultant bit-collision pattern instead of the individual responses to authenticate a group of tags. The previous protocols query the tags sequentially, if the authentication process is finished successfully within a limited time period, it can be claimed that the tags are present simultaneously. However, the tags are not really authenticated simultaneously, so we call the previous grouping proofs to be pseudo-simultaneous grouping proofs. The collision pattern grouping proofs are called real-simultaneous grouping proof. Since the tags are sending signals together, the proof is the collision pattern of these signals. As a result, the verifier does not need to authenticate each tag and the whole group is authenticated as one entity, which is comparable to the time taken to perform a single challenge-response sequence. This reduces the transaction time when processing a high volume of items at any specific reader. The protocol uses only a keyed pseudo-random function and simple bit permutation operations, which is comparable to the cryptographic primitives required by most grouping proofs and lightweight authentication protocols proposed for the RFID environment.

Bit collision are already used in current RFID technology, e.g. in anti-collision methods for ISO 14443 and ISO 15693 (ISO 18000-3) systems, so this protocol could be implemented with tags compatible with current standards. In the last part of chapter 7, we provide a security analysis discussing the attack success probability for five prominent attack scenarios and show that it is possible to obtain a low attack probability when appropriate n and c are chosen.

In this thesis, we proposed a way of encrypting the collision pattern. There are numerical way to make the collision pattern unpredictable to the attackers.

Our way is only a start of further research on this authentication mechanism for a group of RFID tags.

Chapter 8

Conclusion and Future Work

8.1 Security Analysis in this Thesis

In this section, we talk about several ways of security analysis for protocols and why the attack-countermeasure design analysis is adopted in this thesis. The first way of conducting security analysis is to use the computed aided automatic analysis tools, the second way is the theoretical proof that prove the protocol is secure theoretically. Neither of these two methods are used in this thesis. The security analysis used in this thesis is the attack-countermeasure design analysis. We give a review of these analysis in this section.

8.1.1 Computer Aided Analysis

Security protocols have been designed, studied and attacked for over thirty years. Today, computer aided formal analysis tools are becoming popular for assisting in the design process. However, the assumptions that formal tools make and the restrictions they put on the description limit their application to harder protocol design problems of today. In particular, the design of RFID protocol as well as conventional protocol design in the computation resource constrained environments cannot benefit fully from the existing tools because of these impractical assumptions and restrictions [16]. To examine the

protocols described in this thesis, I have tried the Casper [55] tool to analyse the protocols.

Casper is a program that will take a description of a security protocol in a simple, abstract language, and produce a Communicating Sequential Processes (CSP) [32] description of the same protocol, suitable for checking using Failures-Divergence Refinement 2 (FDR2) [56]. The Casper input file must define not only the operation of the protocol, but also the system to be checked [55]. To describe the system, Casper defines cryptographic services used in the common protocols, like public/symmetric key encryption and hash functions. Unfortunately, the Casper does not give definition to the algorithms based in the LPN problem. The Casper is able to describe transmission of an encrypted message, the HB-MP⁺ protocols in Chapter 2 is based on the probability of correct guesses to judge the authentication result and the transmitted messages are in all unencrypted plan-text. Casper will not see any security of the transmitted message.

For the grouping proof protocols proposed in this thesis, they all involve a non-linear message sequences between multiple entities, which is also the major class of protocols Casper is currently unable to deal with[55]. Besides, for the security goals of these protocols such as ‘proof of action’ and anonymity, Casper has no ‘assertion’ to check them in its algebraic system. In the Casper manual[55], G. Lower admitted that the development of Casper has concentrated on protocols whose aims are establishing shared secrets or achieving entity authentication so far.

What is more, the computer aided formal proof itself cannot guarantee the security of the protocol [27]. It is trying to find the feasible attacks(limited in their libraries) on your protocols. If no attacks are found, the protocols cannot

be guaranteed to be secure. At most we can say it has been checked by many known attacks (which the computer can simulate in that tool). However, we cannot say that the protocol is safe against the attacks that the tools have not tried.

8.1.2 Theoretical Formal Analysis

A theoretical formal analysis normally proves the security of the cryptographic scheme by deducing the answer that the breaks of the scheme equals to solve a mathematically difficult problem [45]. The problem with the theoretical formal analysis is that it fails to anticipate most of the attacks on a cryptographic system that are likely to occur. The underlying one-way function is the essential part in the system. However, almost all of the effective attacks on the most popular systems have succeeded not by inverting the one-way function, but rather by finding a weakness in the protocol [13]. The security proofs of protocols need to have assumption of the environments, which contains the ability of the attackers, which might only cover some but not all the attacks. It is turned out that some of the theoretically proven protocols have been found vulnerable to certain attacks. In chapter 2, we have seen that Juels and Weis [8] have give a proof of the security of the HB^+ protocol. Later, Katz and Shin [36] and Katz and Smith [43] proved the parallel concurrent security property of the HB and HB^+ protocols. Unfortunately, the man-in-the-middle attack shown in Gilbert et al. [31] has effectively broken these two protocols. Of course nobody would deny the value of the theoretical proofs, but this example shows the theoretical proofs cannot always exclude all the vulnerability of some protocols.

The other problem is that the theoretical security proofs could have gaps

even under examinations. Comparing to the pure mathematics, the cryptographic society also having less scrutiny towards the theoretical proofs. A famous example is the security proof of Optimal Asymmetric Encryption Padding (OAEP) was accepted for 7 years before a fallacy was noticed. Stern et al. [77] has expressed that, if this proof went unexamined for 7 years, the researchers have no confidence whether the lengthy and often poorly written proofs of less famous security claims are ever examined carefully by anyone. We would feel a little more at ease with “provable security” results if the same tradition of careful examination of all important papers existed in theoretical cryptography.

8.1.3 Attack-Countermeasure Design Analysis

In this thesis, the attack-countermeasure design analysis is used across all the chapters. This analysis shows the design ideas and reveals the countermeasures to defend the known attacks to the previous similar protocols. It is an analysis used most often in the protocol research papers. By this analysis, the design ideas are very clearly explained, and the vulnerability of previous similar protocols are eliminated accordingly. In this thesis, both HB protocols and grouping proof protocols are already researched by many previous publications, most of the publications has the attack-countermeasure design analysis. Normally such publication would inspire new publications which improve the original one. In this way, the research in topic reaches greater depth with continuous publications.

8.2 Future Works and Application of HB protocols

The HB style protocol, including the protocol and abstract form I proposed in chapter 2, are still not practical in the current RFID systems. The first unavoidable performance penalty for the implementation is the repeated message transmission for a single tag's authentication. In current RFID system, most of the operation time is spent on the transmission and most of reading errors occur in the transmission. So in the implementation, the threshold of a successful authentication should be loosen to a certain degree to beat the affection of the transmission errors. However, it is not to say that the HB protocols are only invented to make a deliberate use case of the 'LPN Problems' in RFID. To achieve security, there must be some security mechanism implemented in the RFID tags, generally it is expected that certain very light symmetric cryptography or MAC functions be available for RFID tags. HB style protocols uses a different and creative mechanism. It depends on how the protocol family would evolve, it may have some practical value in the future.

8.3 Future Works and Applications of Grouping Proof

The yoking proof is firstly proposed by Ari Juels [40] to prove two tags are present together. Then the idea has been developed by several publications into the grouping proofs, which verify whether they are present simultaneously. After several works which my friend and I have conducted together, the understanding of grouping proofs goes much deeper. The reading order independence, as our first work in this field, is inspired directly by the reading order penalties in previous publications. Then we investigate the user case of

“yoking proof”, found the way of binding secret together makes identification of the failed tags impossible. If the final result goes wrong, it is impossible for the “yoking proof” to identify which tags are the causes of the failure. For the practical applications, usually it is desirable to find these causes and solve the problem. From the core idea of “grouping proof”, which requires the RFID tags to be verified in the short time limits and produce a group of their simultaneousness, we proposed the select-respond grouping proof. The subgrouping frameworks in Chapter 6 is inspired by the divide-and-conquer technique used in many computer algorithms. The tree-based anti-collision algorithm is able to provide the facility to divide the subgroups without introducing any new algorithm. Also the idea of the select-response mode previously published is adopted in the protocol demonstration example. By further exploiting the anti-collision algorithm adopted in the subgrouping proofs, I proposed the idea of the bit-collision patterns. The collision pattern itself becomes the grouping proof, which proves the real simultaneous presence of the tags. The initiation of each paper is inspired by the previous publication and my own work. By this way, the research on the grouping proof schemes of RFID tags goes deeper.

The application of yoking proof is quickly expanded to provide evidence of a complete group of RFID tags passing a certain check point, which has practical value in current applications such as assembly line, logistics and distribution centre. For example, manufacturers can use group proofing to reliably check the integrity of their products opening the packaging and thereby improve their efficiency. In medical practices, all RFID tagged medication (such as patient’s ID, injection, medicine and associated instruction) can be included in our proposed grouping proof scheme. Hence, the overall safety of medical

treatment can be enhanced. The case to deal with large amount of tags together, the bit-collision patterns are genuinely fast and deal the group as a whole. It can be adopted in any identification cases that needs to deal with large amount of tags.

8.4 Future of the RFID Protocols

RFID protocols have been a very interesting field of research in these years. The RFID systems are usually distributed and open systems. They are distributed because many entities (multiple tags and readers) are acting together. They are open systems because the unexpected new tags can add to the system at any time as the goods are traveling globally nowadays [81]. The new application schemes, coupled with the limited computing resource of the RFID tags, makes the RFID protocol design an active research area in these years. Many creative protocols has been proposed, although very few of them are practical in applications, even fewer of them can be proven to be secure, these researches and innovations have greatly enriched the protocol design methods and the knowledge of the RFID system.

Abbreviation Glossary

Abbreviation	Full Words
AFI	Application Family Identifier
AIDC	Automatic Identification and Data Capture
BPS	Backend Processing System
CCTV	Closed-Circuit Television
DoS	Denial of Service
EEPROM	Electrically Erasable Programmable Read-Only Memory
GID	Group Identifier
ICAO	International Civil Aviation Organization
ISO	International Standard Organization
IP	Intellectual Properties
ONS	Object Naming Service
LNP	Learning Parity in the presence of Noise
MAC	Message Authentication Code
NVB	Number of Valid Bits
OAEP	Optimal Asymmetric Encryption Padding
POS	Point Of Sale
PT	Pallet Tag
RAM	Random-Access Memory
RFID	Radio Frequency IDentification
RRP	Random-Response Protocol
SRAM	Static Random Access Memory
SRP	Select-Response Protocol
UHF	Ultra-High Frequency
UID	Unique IDentifier

Table 8.1: Abbreviation Glossary

Bibliography

- [1] *9303, Parts I, II, III, Machine Readable Travel Documents specifications.*
- [2] *Advantages of RFID versus barcodes*, 2006.
- [3] *ISO/IEC 15693. Identification cards – contactless integrated circuit cards – vicinity cards.*, 2006.
- [4] *Tekvet-IBM cattle tracker uses active RFID tags*, 2006.
- [5] *ISO/IEC 14443. identification cards – contactless integrated circuit cards – proximity cards*, 2008.
- [6] *ISO/IEC 18000 information technology aidc techniques-RFID for item management – air interface.*, 2008.
- [7] *Company profile of invengo ltd*, 2009.
- [8] S.Weis A. Juels, *Authenticating pervasive devices with human protocols*, Advances in Cryptology-Crypto2005, Lecture Notes in Computer Science, Springer **3621** (2005), 293–308.
- [9] A.Juels, *RFID security and privacy: A research survey*, IEEE Journal on Selected Areas in Communications **Vol. 24, No 2** (2006), 381–394.
- [10] R. Angeles, *RFID technologies: supply-chain applications and implementation issues*, Information Systems Management **22** (2005), no. 1, 51–65.
- [11] G. Avoine, *Bibliography on security and privacy in RFID systems.*
- [12] G. Avoine and P. Oechslin, *RFID traceability: A multilayer problem*, Lecture Notes in Computer Science **3570** (2005), 125.
- [13] M. Bellare, *Practice-oriented provable-security*, Lecture notes in computer science **1561** (1999), 1–15.
- [14] E. Berlekamp, R. McEliece, and H. Van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Transactions on Information Theory **24** (1978), no. 3, 384–386.
- [15] H.L. Blöcher, G. Rollmann, and S. Gärtner, *Trends in Automotive RF Wireless Applications and their Electromagnetic Spectrum Requirements*, Proc. of the German Microwave Conference 2005, pp. 148–151.

- [16] M. Bond and J. Clulow, *Extending security protocol analysis: New challenges*, Electronic Notes in Theoretical Computer Science **125** (2005), no. 1, 13–24.
- [17] D. Carluccio, K. Lemke-Rust, C. Paar, and A. Sadeghi, *E-Passport: The Global Traceability or How to Feel Like a UPS Package*, Lecture Notes in Computer Science **4298** (2007), 391.
- [18] C. Castelluccia and G. Avoine, *Noisy tags: A pretty good key exchange protocol for RFID tags*, Lecture Notes in Computer Science **3928** (2006), 289.
- [19] Jonathan Collins, *Hong kong’s airport to tag bags*, Tech. report, RFID Journal, 2004.
- [20] G. De Vita and G. Iannaccone, *Design criteria for the RF section of UHF and microwave passive RFID transponders*, IEEE Transactions on microwave theory and techniques **53** (2005), no. 9, 2978–2990.
- [21] M. Donath, *Future directions in RFID application and research in transportation*, Transportation Conference on Research Opportunities in RFID Transportation Applications, 2006.
- [22] David C. Yen Shi-Ming Huang Dong-Her Shih, Po-Ling Sun, *Taxonomy and survey of RFID anti-collision protocols*, Computer Communications **Volume 29, Issue 11** (July 2006), 2150–2166.
- [23] L. Dong-Sheng, Z. Xue-Cheng, Z. Fan, and D. Min, *Embedded EEPROM Memory Achieving Lower Power-New design of EEPROM memory for RFID tag IC*, IEEE Circuits and Devices Magazine **22** (2006), no. 6, 53–59.
- [24] D.N. Duc and K. Kim, *Securing HB⁺ against GRS man-in-the-middle attack*, Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security, 2007, pp. 23–26.
- [25] E. Dyson and E. Dean, *RFID: Logistics meets identity*, Release 1.0: Esther Dysons Monthly Report (2003).
- [26] Klaus Finkenzeller, *RFID handbook: Fundamentals and applications in contactless smart cards and identification, second edition*, John Wiley & Sons Ltd, 2003.
- [27] C. Fisher, *Advancing the study of programming with computer-aided protocol analysis*, Empirical studies of programmers: Second workshop, Ablex Publishing Corp., 1987, p. 216.

- [28] H. Gilbert, M.J.B. Robshaw, and Y. Seurin, *Increasing the Security and Efficiency of HB*, Advances in cryptology–EUROCRYPT 2008: 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008: proceedings, Springer-Verlag New York Inc, 2008, p. 361.
- [29] G. Hancke, *Modulating a noisy carrier signal for eavesdropping-resistant hf RFID*, e & i Elektrotechnik und Informationstechnik **124** (2007), 404.
- [30] E. Haselsteiner and K. Breitfuss, *Security in near field communication (NFC)*, InWorkshop on RFID Security (2006).
- [31] H.Silbert H.Gilbert, M.Robshaw, *An active attack against HB⁺, a provable secure lightweighted authentication protocol.*, Cryptology ePrint Archive **237** (2005).
- [32] CAR Hoare, *Communicating Sequential Processes*, Communications of the ACM (1985).
- [33] J.H. Hoepman, E. Hubbers, B. Jacobs, M. Oostdijk, and R.W. Schreur, *Crossing borders: Security and privacy issues of the European e-passport*, Lecture Notes in Computer Science **4266** (2006), 152.
- [34] D.R. Hush and C. Wood, *Analysis of tree algorithms for RFID arbitration*, IEEE International Symposium on Information Theory, Citeseer, 1998, pp. 107–107.
- [35] E.P.C. Inc, *Class 1 generation 2 UHF air interface protocol standard version 1.0.9*, Gen **2** (2005).
- [36] J.Shin J. Katz, *Parallel and concurrent security of the HB and HB⁺ protocols.*, (2005).
- [37] E Dottax J.Bringer, H.Chabanne, *HB⁺⁺: a lightweight authentication protocol secure against some attacks.*, IEEE International Conference on Pervasive Sevices, Workshop on Security,Privacy and Trust in Pervasive and Ubiquitous Computing - SecPerU. **1** (2006), 34.
- [38] A.Peinado J.Munilla, *HP-MP: A further step in the hb-family of lightweight authentication protocols*, Computer Networks **51** (2007), 2262–2267.
- [39] A. Juels., *Minimalist cryptography for low-cost RFID tags*, Security of Communication Networks (SCN) (2004), 149X164.
- [40] A. Juels, *"Yoking-proofs" for RFID tags*, Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (2004).

- [41] A. Juels, D. Molnar, and D. Wagner, *Security and privacy issues in e-passports*, IEEE SecureComm **5** (2005).
- [42] A. Juels, R.L. Rivest, and M. Szydlo, *The blocker tag: Selective blocking of RFID tags for consumer privacy*, Proceedings of the 10th ACM conference on Computer and communications security, ACM New York, NY, USA, 2003, pp. 103–111.
- [43] J. Katz and A. Smith, *Analyzing the HB and HB+ protocols in the large error case*, IACR ePrint report **326** (2006), 2006.
- [44] C. Kern, *Radio-Frequency-Identification for security and media circulation in libraries*, The Electronic Library **22** (2004), no. 4, 317–324.
- [45] N. Koblitz and A.J. Menezes, *Another Look at “Provable Security”*, Journal of Cryptology **20** (2007), no. 1, 3–37.
- [46] H. Krawczyk, *LFSR-based hashing and authentication*, Lecture Notes in Computer Science **839** (1994), 129–139.
- [47] H. Krawczyk, *New hash functions for message authentication*, Lecture Notes in Computer Science **921** (1995), 301–310.
- [48] P. Krishna and D. Husak, *RFID infrastructure*, IEEE Communications Magazine **45** (2007), no. 9, 4.
- [49] G. Robins L. Bolotnyy, *Generalized yoking-proofs for a group of RFID tags*, International Conference on Mobile and Ubiquitous Systems, July 2006.
- [50] L. Lamport, *Constructing digital signatures from a one-way function*, Tech. report, Technical Report CSL-98, SRI International, 1979.
- [51] M. Langheinrich and R. Marti, *Practical minimalist cryptography for RFID privacy*, IEEE Systems Journal, Special Issue on RFID Technology **1** (2007), no. 2, 115–128.
- [52] S.M. Lee, S. Park, S.N. Yoon, and S. Yeon, *RFID based ubiquitous commerce and consumer trust*, Industrial Management and Data Systems **107** (2007), no. 5, 605.
- [53] C. Legner and F. Thiesse, *RFID-based maintenance at Frankfurt airport*, IEEE Pervasive Computing **5** (2006), no. 1, 34–39.
- [54] K. Lemke, A.R. Sadeghi, and C. Stubble, *An open approach for designing secure electronic immobilizers*, Lecture notes in computer science **3439** (2005), 230.

- [55] G. Lowe, *Casper: A compiler for the analysis of security protocols*, Journal of computer security **6** (1998), no. 1, 53–84.
- [56] FSE Ltd, *Failures-divergence refinement: FDR 2 user manual*.
- [57] T. McCoy, RJ Bullock, and PV Brennan, *RFID for airport security and efficiency*, Signal Processing Solutions for Homeland Security, 2005. The IEE Seminar on (Ref. No. 2005/11108), 2005, p. 9.
- [58] K. Michael and L. McCathie, *The pros and cons of RFID in supply chain management*, Proceedings of the International Conference on Mobile Business, IEEE Computer Society, 2005, pp. 623–629.
- [59] Rossana Motta Mike Burmester, Breno de Medeiros, *Provably secure grouping-proofs for RFID tags*, International Conference, 8th Smart Card Research and Advanced Applications (CARDIS 2008), IFIP WG 8.8/11.2, September 2008.
- [60] B. Nath, F. Reynolds, and R. Want, *RFID Technology and Applications, Pervasive Computing*, IEEE **5** (2006), no. 1, 22–24.
- [61] F. Niederman, R.G. Mathieu, R. Morley, and I.W. Kwon, *Examining RFID applications in supply chain management*, Communications of the ACM **50** (2007), no. 7, 101.
- [62] P.V. Nikitin and KVS Rao, *Performance limitations of passive UHF RFID systems*, Proceedings of the IEEE Antennas and Propagation Symposium, 2006, pp. 1011–1014.
- [63] M.Blum. N.J.Hopper, *Secure human identification protocols.*, Advanced in Cryptology-ASIACRYPT’2001, Lecture Notes in Computer Science, Springer **2248** (2001), 52–66.
- [64] S. Piramuthu, *On existence proofs for multiple RFID tags*, Proc. ACS/IEEE International Conference on Pervasive Services, 26–29 June 2006, pp. 317–320.
- [65] S. Piramuthu and YJ Tu, *Modified HB authentication protocol*, Western European Workshop on Research in Cryptology, WEWoRC, 2007.
- [66] D. Ranasinghe, D. Engels, and P. Cole, *Low-cost RFID systems: Confronting security and privacy*, Auto-ID Labs Research Workshop, Cite-seer, 2004, pp. 54–77.
- [67] K.V.S. Rao, P.V. Nikitin, and S.F. Lam, *Antenna design for UHF RFID tags: a review and a practical application*, IEEE Transactions on antennas and propagation **53** (2005), no. 12, 3870–3876.

- [68] M.R. Rieback, B. Crispo, and A.S. Tanenbaum, *Keep on blockin'in the free world: Personal access control for low-cost RFID tags*, Lecture Notes in Computer Science **4631** (2007), 51–59.
- [69] M. Roberti, *Analysis: RFID wal-marts network effect*, Analysis (2003).
- [70] G. Roussos and V. Kostakos, *RFID in Pervasive Computing: State-of-the-art and Outlook*, Pervasive and Mobile Computing **5** (2009), no. 1, 110–131.
- [71] J. Saito and K.Sakurai, *Grouping proof for RFID tags*, Proceedings of the 19th International Conference on AINA, 2005.
- [72] S.E. Sarma, S.A. Weis, and D.W. Engels, *RFID systems and security and privacy implications*, Lecture notes in computer science (2003), 454–469.
- [73] B. Soller, M. Wolfe, and M. Froggatt, *Polarization resolved measurement of Rayleigh backscatter in fiber-optic components*, National Fiber Optic Engineers Conference, OSA Technical Digest Series.
- [74] S.Piramuthu, *HB and related lightweight authentication protocols for secure RFID tag/reader authentication*, COLLECTeR Europe Conference (June, 2006.).
- [75] T. Staake, F. Thiesse, and E. Fleisch, *Extending the EPC network: the potential of RFID in anti-counterfeiting*, Proceedings of the 2005 ACM symposium on Applied computing, ACM, 2005, p. 1612.
- [76] V. Stanford, *Pervasive computing goes the last hundred feet with RFID systems*, IEEE pervasive computing (2003), 9–14.
- [77] J. Stern, *Why provable security matters?*, Lecture notes in computer science (2003), 449–461.
- [78] F. Thiesse and F. Michahelles, *An overview of EPC technology*, Sensor review **26** (2006), no. 2, 101–105.
- [79] P. Tuyls and L. Batina, *RFID-tags for Anti-Counterfeiting*, Lecture Notes in Computer Science **3860** (2006), 115.
- [80] R. Weinstein, *RFID: a technical overview and its application to the enterprise*, IT professional (2005).
- [81] S.A. Weis, S.E. Sarma, R.L. Rivest, D.W. Engels, et al., *Security and privacy aspects of low-cost radio frequency identification systems*, Lecture notes in computer science (2004), 201–212.

- [82] NC Wu, MA Nystrom, TR Lin, and HC Yu, *Challenges to global RFID adoption*, Technovation **26** (2006), no. 12, 1317–1323.
- [83] Keith Mayes Jung-Hui Chiu Xuefei Leng, Yuanhung Lien, *Select-response grouping proof for RFID tags*, Proceeding of 1st Asian Conference on Intelligent Information and Database Systems Lecture Notes in Computer Science, April, 2009.
- [84] Jung-Hui Chiu Yuanhung Lien, Keith Mayes and Xuefei Leng, *Reading order independent grouping proof for RFID tags*, Proceeding of Intelligence and Security Informatics (ISI) IEEE ISI 2008 International Lecture Notes in Computer Science 5075 Springer 2008, 2008.