2010

Security and Performance Analysis for RFID Protocols

Bing LIANG
Singapore Management University, bing.liang.2007@smu.edu.sg

Security and Performance Analysis for RFID Protocols

by

Bing LIANG

Submitted to School of Information System in
partial fulfillment of the requirements for the Degree of
Master of Science in Information Systems

Supervisor: Assistant Professor Yingjiu Li

Singapore Management University

2010

Abstract

SECURITY AND PERFORMANCE ANALYSIS FOR RFID PROTOCOLS

Bing LIANG

Singapore Management University, 2010

Thesis Director: Assistant Professor Yingjiu Li

Radio Frequency Identification (RFID) is an advanced object identification technology that has already been applied in various industries. However, the insecure nature of the communication channel between readers and tags makes RFID systems vulnerable to various kinds of attacks. In recent years, many new methods have been proposed to improve the security of RFID systems, such as disabling tags, agent management and establishing cryptographic protocols. Among them, we focus on the last approach, which is more economic and convenient in certain cases.

The first part of our work is to categorize typical existing RFID protocols according to their security levels. The result is vitally important to RFID system administrators who need to find different protocols to be implemented in their systems. The trade-off to be made in decision is that higher security level typically implies worse performance.

We examine the performance in two aspects: the look-up cost in RFID reader's back-end database and the tag-related cost. The tag-related cost includes the cryptographic operation cost (cryptographic computation cost along with access operation cost in RFID tag's chip memory), and the communication cost between reader and tag. We perform traditional database complexity analysis to assess the database look-up cost, and conduct experiments to evaluate both the cryptographic operation cost and and communication cost, so as to have a thorough understanding of the overall time cost of each RFID protocol. This work is important because efficiency is a major concern in the design of RFID protocols, especially when high security level is achieved with complex cryptographic operations being performed on resource-limited RFID tags. An inefficient RFID protocol could be a bottleneck of the whole system in real applications.

Finally, we benchmark the performance of each RFID protocol, compare performance of protocols in different security levels and investigate the extra cost required to achieve certain security properties. We find a trade-off between look-up cost and tag-related cost. Based on the results of performance benchmarks, we revise some existing protocols and propose several design suggestions for creating new RFID protocols.

Table of Contents

Acknowledgments

Thanks to my advisor Assistant Professor Yingjiu Li for his guidance and advice.

Thanks to my committee members: Professor Robert Deng and Assistant Professor Xuhua Ding for their previous time and concerns.

Thanks to all the members of RFID security group: Tieyan Li, Changshe Ma, Kevin Chiew, Chunhua Su, Zongyang Zhang, Shaoying Cai, Yan Li and Qiang Yan for their useful discussions.

Dedication

I dedicate this thesis to my parents, my grandfather and my husband for their love, support and encouragement.

Chapter 1: Introduction

In this section, we briefly introduce the background of RFID systems, RFID system performance measurements and the major contributions of this thesis.

1.1 RFID Systems

An Radio Frequency IDentification (RFID) System is a wireless device originated from military systems. An RFID system comprises three components: a tag (transponder), a reader (transceiver) and the reader's back-end database. A tag usually has a microchip for storing basic information (ID, manufacture's info) and an antenna for transmitting signals to RFID reader. A reader can interrogate a tag by sending a signal via electro-magnetic fields. The tag receives the signal through its own antenna and responds with information stored on its microchip, which is verified by the reader against its back-end database.

According to different working frequencies, the tags can be categorized into three classes: low frequency (LF) tags, working frequency from 124 to 135 kHz; high frequency (HF) tags, working at around 13.56 MHZ and the ultra high frequency (UHF) tags, working frequency from 860MHz to 960MHz [RCT06]. In general, the higher the working frequency, the farther the read range of a tag. Typically, LF tags can be read within 30 cm, and HF ones can be interrogated up to 1 m. The UHF tags can even be read from 7 m away. According to the capability of RFID tags, RFID tags can be also classified into passive tags and active tags [RCT05]. Usually, active tags have their own batteries and extra memory storage while passive tags do not. Therefore, they have more energy, more powerful computational ability and larger memory. Their read range can be extended to more than 100 m but the tags cost around $20 [Jue06]. However, a most important goal

in industry is to minimize the cost (around $0.05 / p), so in this thesis, our topic only focuses on the passive tags. A passive tag would respond automatically when a reader interrogates with this tag. For the reason of low-cost features, RFID passive tag has its own disadvantages compared to other wireless device. First, as discussed above, the range between reader and tag is limited: from 0.1 m to 7 m. Second, due to the low cost, there are only less than 5000 gates on a tag so that the tag has only limited computational ability and storage. Thirdly, the chip on a tag is sensitive to the external environment: metal, liquid, radio-reflect and radio-absorb material all can affect the communication between a reader and a tag [Jue06,RCT06,WSRE03].

As an RFID system shown in Fig 1.1, it is generally assumed that the communication channel between a reader and its back-end database is secure while the channel between a reader and a tag is wireless and insecure. A reader interrogates a tag through forward channel and a tag responds to a reader through backward channel. The signals in a backward channel are much weaker than the ones in a forward channel, because the passive tag's power is obtained from a reflection of reader's signal. The messages in the wireless channels can be eavesdropped by any passive party with the receiver equipment. For a malicious party, it is more difficult to eavesdrop in the backward channel than in the forward channel. An active malicious party can even intercept, insert, block, modify the messages in the wireless channel.

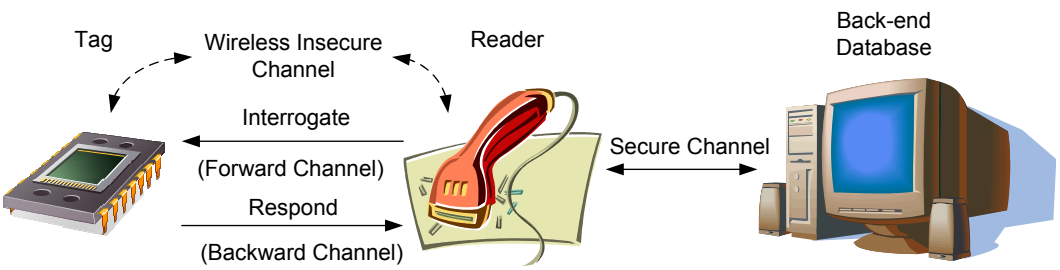


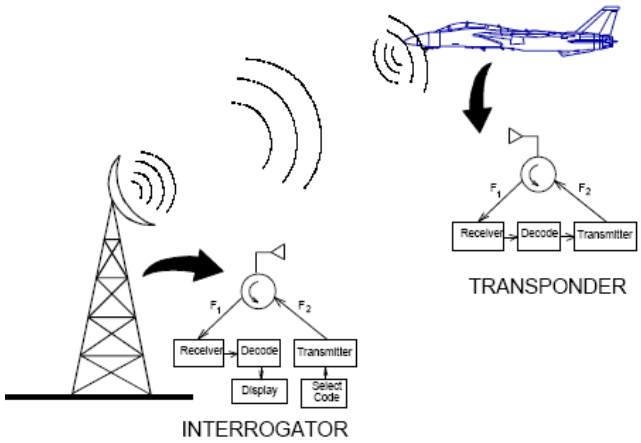
Figure 1.1: RFID Systems

The RFID idea ordinates from the military system "Identification Friend or Foe" (IFF) system shown in Fig 1.2 [IFF,RCT06], which is the radar system to identify friend's or enemy's planes by ways of authenticating signals reflected from planes.

IFF's transponder and interrogator share a secret key, and encrypted information is transferred between them. After receiving encrypted messages, the two party can decrypt them with the sharing key. Though RFID originates from IFF, the technical requirements are different between IFF military systems and RFID applications. The five main differences between RFID system and IFF military system are summarized in the Table 1.1, based on Rieback, Crispo and Tanenbaum's analysis in [RCT06].

First, there is a clear definition about attackers and defenders in military system, while the delineation becomes fluffy in RFID applications. For example, in RFID systems, it is possible the former legitimate user becomes an attacker as time changes. Second, IFF systems are physical secure, but in RFID systems, it is nearly impossible to guarantee this situation. In RFID systems, the former legitimate users, who once obtained RFID tags, can be potential attackers . Third, the security goals for two applications are not similar: the IFF's goal is to protect the confidentiality and even sacrify the privacy, yet RFID system should protect both security and privacy. Fourth, the IFF's back-end database is a stand-alone one, means one database for each plane. However, the RFID's database can be shared by various parties, and different parties in RFID systems can access the database online or offline. Especially in supply chains, the database is shared by manufactories, retailers, logistic parties, even single customers. The last but not the least, IFF system involves nation security, so it is enforced at all cost. On the contrary, RFID system is an industrial application, and the low cost is one of the most important concerns. Till now, the acceptable cost is $0.05 per tag.

In the industrial field, the most popular application in logistic and retailer is barcode nowadays. RFID is called next-generation barcode and considered as the most possible substitute of today's optical barcode. In certain cases, RFID systems are more convenient and more efficient than barcode. In addition, they can adapt to different environments. However, there exist potential privacy threats in RFID systems. The different features of RFID and barcode are shown in Table 1.2.



[IFF]

Figure 1.2: IFF Systems

Table 1.1: RFID Systems vs. IFF Systems
[RCT06]

	$RFID$	$IFF\ System$
Attack Model	Fuzzy delineation between attackers and defenders	Clear delineation between attackers and defenders
Physical Security	Physically insecure	Physically Secure
Security Goal	Security & Privacy	Confidentiality
Back-end Infrastructure	Shared Database	Stand-alone database
Security Cost	Low cost 5 cents, recycled	At all cost

Compared with traditional method for object identification—barcode, RFID systems have many different features. First of all, RFID tags can be identified automatically in bulks without light, while barcode needs a person to scan codes through line-of-sight one by one to identify the codes. These features improve efficiency of an identification system significantly. In addition, RFID tags can identify every single object with a unique ID, while barcode seems to have limited ability in this aspect. Barcode is usually used to identify one class of items. For example, the same type of Coca-Cola bottles has the same barcode in a supermarket nowadays. In future, it is possible that every bottle of them is attached an RFID tag with a unique ID. In case that a customer meets a problem with one bottle, not only does the supermarket's manager know exactly that there is a problem with Coca-Cola drinks, but also he knows exactly which bottle has the problem. Next, RFID system communication is through wireless channel and the range can be extended up to 7 m, while barcode needs a nearby scan and the effective range is limited. This

Table 1.2: RFID vs. Barcode
[Jue06][MW04]

$RFID$	$Barcode$
No need of Line-of-sight	Need Line-of-sight
No human-intervention	Need human-intervention
Proceed in Bulks	Proceed one by one
Identify objects in item-level	Identify objects in type-level
Wireless	closely scanned
Low cost 5 cents, recycled	Low cost, unrecycled
People unconsciously scanned	People consciously scanned

merit dramatically eliminates the range limitation of barcode. What's more, RFID tags can be recycled and reused while barcode cannot. This advantage is specially meaningful to a supply chain system that involves a large number of tags. The last feature of RFID tags is that they can be read without the taker's consciousness, which is a potential threat of sensitive information leakage. According to RFID systems' special features, they can be applied in a wide range of everyday life. On the contrary, RFID systems spring out a couple of security problems.

In lots of applications, especially supply chains, multiple tags need to communicate with a same reader. However, a reader can respond to only one tag at a time. Therefore, there exist some anti-collision algorithms according to tags' different working frequencies. The anti-collision algorithm for high frequency (HF) tags is the probabilistic "slotted aloha" algorithm [Bin00, MB83], while for ultra high frequency (UHF) tags is the deterministic "binary tree walking" algorithm [SWE02]. The reason for different choices of anti-collision algorithms is that the bandwidth of the UHF's communication channel is more than the one of HF's communication channel [SWE02]. These anti-collision methods help with the multi-response problem in RFID systems.

1.2 RFID Applications

Because of the beneficial features of RFID systems, a wide range of applications have been reported in industries. A few typical applications are given below:

1. Supply Chain [RCT06]:

   RFID technique's potentially widest application is the supply chain and logistics. RFID tags can be attached onto the objects in a supply chain so that every object has a unique ID. With the unique IDs, the transportation routine can be supervised in the whole supply chain from manufactory to retailers till stores. In an RFID supply chain, numerous RFID tags are needed as a number of different goods are involved. Therefore, the cost-effectiveness is an essential requirement in RFID supply chain. Low-cost and recyclable tags meet this requirement perfectly. In addition, the overall cost of the supply chain could be further reduced by adopting high efficiency bulk-processing and non-human-intervention method. Another benefit is that RFID's object-level identification can guarantee the goods safety.

2. Animal Tracking [GJP05] [RCT06]:

   RFID tags can either be used in animal scientific research or used in home pet tracing. Scientists can implant RFID chips with GPS, into the endangered animals to learn their habits and further protect them. For instance, some scientists apply this methods on a couple of dolphins [GJP05]. Other scientists implant tags into livestock to prevent, mitigate and control from a large scale diseases bursting out such as H1N1 Flu [RCT06]. Other than the scientific research, RFID tags can be implanted in pets to locate them in case that they are lost.

3. Access Control [GJP05] [MW04] [RCT06] [Rot08]:

   RFID tags can be a key and used in access control systems [MW04]. They can be a security device to launch a car [GJP05]. Proximity card with RFID chips can manage the entry of libraries, companies to guarantee these places' security [RCT06] [Rot08].

4. Medical System [GJP05] [Jue06] [Rot08]:

Once patients are attached or implanted by RFID tags in the future, doctors can immediately obtain a patient's detailed information such as the patient's ID, name, medical history, allergy history, and emergency contacts by scanning the patient's tag. This documentary mechanism is much more efficient and accurate than the traditional manual checking. However, the sensitive information such as the patient's medical history can be stolen by being illegally scanned.

5. Smart Appliance [Jue06]:

   In future, appliances with RFID systems can be intelligent and provide a big convenience in people's everyday life. For example, smart washing machine with an RFID reader may read information from the tags attached on the clothes. Through checking the back-end database, this washing machine can learn the knowledge of the clothes' material and choose a suitable washing mode (e.g economy, wool or fuzzy mode) for each clothes. Another example is that a smart refrigerator with RFID reader can warn people about an expired food by scanning the tag on package.

6. Automatic Payment [GJP05] [Jue06] [RCT06] :

   Today, credit card's serial-number-based payment is insecure and it is easy to counterfeit a card for anyone who obtains the card's number. To improve the security, RFID tags are involved in credit card systems as a token. In such credit card systems, only after the reader authenticates the token successfully, the payment is made. RFID token enhances credit card system's security a lot, because only attackers with both specific reader equipment and the card's serial number can take malicious actions. The other automatic payment is the public transportation ticket system. With RFID embedded card, bus systems and subway systems can reduce costs from selling tickets manually. The third RFID automatic payment system is automatic toll: these tolls can read the tags in a car and charge the toll fee even when the car is traveling up to

100 miles per hour [GJP05]. They are more efficient, accurate, convenient than the traditional toll stations which collect fees by person.

7. E-passport [Jue06] [Rot08]:

   RFID chips can be embedded in passports to record the holder's biometric information such as fingerprint and iris data. A key of 128 bits [Rot08] is also used to protect the data in a tag's chip. Compared to the traditional passport with only a photo, RFID embedded passports are more difficult to counterfeit. Therefore, e-passport could be of a great assistance in population monitoring and management as well as national security.

1.3 EPC Specification in RFID Systems

Among the popular benchmarks of RFID protocols, EPCglobal organization proposed EPC (Electronic Product Code) specification as RFID protocol standard. We take EPC Class I Generation II as an example to give readers a brief introduction of this standard. The working frequency of EPC Class I Gen II tags is from 860 MHz to 960 MHz. In this industry standard, there are four memory banks in each tag [Inc08]: user memory, reserved memory, EPC memory and TID memory. A users can store user-specific data in a tag's user memory. The 32-bit kill password and/or the 32-bit access passwords are stored in the reserved memory. The tag's manufactory information such as a tag's serial number is stored in the TID memory. Finally, the tag's unique EPC code is in the EPC memory. We take a 96-bit EPC code as an example. The first 8-bit EPC code is the header. The next 28-bit is the general manager number which identifies the organization owning the tag. The next 24-bit code illustrates which class the object attached to the tag belongs to. The last 36-bit serial number is the unique ID in the object class. The IP address of the database containing the object's detailed information such as delivery routine can be pointed by the whole 96-bit EPC code [OSK03]. A typical industrial application of EPC

Class I Gen II standard is supply chain and logistics.

1.4 Performance Measurements

It is important to measure the performance of an RFID system. On one hand, due to the countless number of tags in industries such as supply chain, the search overhead of a tag in the reader's back-end database may be overwhelmingly high. Therefore, the search cost should be a concern. On the other hand, in order to protect an RFID system's security, it is usually to implement an RFID protocol in the system. Thus, another cost of RFID security protocol is cryptographic operation cost as well as reader-tag communication cost. In [ADO05], Avoine claimed that a cryptographic operation on a single computer only takes 2^{-23} second to finish, while it would take a lot more time to finish a cryptographic operation on a tag due to a tag's limited memory and computational ability. Therefore, cryptographic operation cost in RFID reader side is ignorable. In summary, we measure the performance of an RFID system in two aspects:

1. Search cost in reader's database

   When there are a large number of tags in RFID system, the search cost of a specific tag cannot be ignorable. Usually, it is assumed the total number of tags is N, and we use functions with parameter N to represent the search cost.

2. Tag-related cost

   Tag-related cost comprises cryptographic operation cost in tags and communication cost between reader and tag. After simplifying the procedure, there are four basic operations used in cryptographic operations in tags: tag-tag read, tag-tag write, AES encryption and hash functions. The communication cost comprises reader-to-tag operations, including reader-tag read and reader-tag write, because RFID passive tags cannot actively launch communication

sessions.

In the following, we will analyze search cost and tag-related cost in detail in chapter 4 and 5, respectively.

1.5 Our Contribution and Thesis Organization

The main contribution of this thesis is three-fold:

1. First, existing RFID protocols are categorized into six classes in terms of different security levels. Compared with previous work, our categorization covers two categories of weak security protocols, which were not discussed in previous security and privacy models. Our work is thus closer to real applications, which usually implement solutions with weak security.

2. Second, the relationship between security and a reader's search cost is analyzed. We take BMM protocol as a case study to show the trade-off between security and search cost. In addition, the search costs of typical protocols in each category are summarized. We find that in certain cases, the search cost is higher with higher security levels. In other cases, the search cost is not related to security levels directly but more related to the data structure of a reader's back-end database.

3. Finally, the relationship between security and tag-related cost is investigated. A formula is set up on tag-related cost for any generic RFID protocols. Additionally, the tag-related time costs of typical protocols in six categories are calculated according to the formula. The best performance protocols in each category are selected as benchmarks to measure other protocols' performance. Through comparison, redundant operations of a couple of protocols are discovered for revision. To make a trade-off between search cost and tag-related cost, some protocol-design suggestions are proposed.

The rest of my thesis is organized as follow: in Chapter 2, security problems and existing solutions in RFID systems are introduced. In Chapter 3, existing RFID protocols are categorized according to different security levels. In Chapter 5, the relationship between security and tag-related cost is discussed. In Chapter 6, the thesis is summarized and future research direction is pointed out.

Chapter 2: Security Problems in RFID Systems and Existing Solutions

There are three parts in the whole RFID system: tag, reader and back-end database. The back-end database can be defended by traditional database security protection such as encrypted database, access control, etc.. Thus, we do not consider the back-end database security problem as an especial RFID system problem. In addition, the connection between reader and back-end database is assumed as a secure channel; therefore, we do not consider the attacks through this channel, either. In particular, we consider the following components for addressing the security problems in RFID systems: RFID reader, RFID tag and reader-tag communication channel.

1. RFID Reader:

   Adversary can mimic legitimate readers to interrogate authentic tags. Fake readers can stole the sensitive information of tags and violate their privacies. Several fake readers can even collude to trace a certain tag. If there are no write protections on tags, malicious readers can even modify a tag's content arbitrarily.

2. RFID Tag:

   Low-cost RFID tags could be easily cloned without enough protection. Reverse engineering can even clone tags through physical layer analysis. Adversaries can use cloned tags to replace the original ones. For example, they can replace the valuable goods with cheap ones. If an attacker cracks a tag successfully and obtains the tag's secret, he/she can trace the previous messages about the tag. In another aspect, adversaries can launch Denial of service

(DoS) attacks on legitimate RFID readers by querying the reader with a number of fake tags. As a result, the reader cannot achieve normal interrogations with real tags.

3. Communication Channel:

   Malicious parties can sit in the communication channel for eavesdropping the conversation between a legitimate reader and a tag. Active ones can even intercept, relay, replay, block, change and insert messages between reader and tag. An adversary can record and replay messages to launch man-in-the-middle attacks.

2.1 Adversary Model

In typical RFID security scenarios, adversaries with different levels of power are modeled to analyze different RFID authentication protocols [LLG08]. We simplify the categorizing methods in [LLG08] and consider adversaries in three levels:

- **Level-1 (Passive attack)**:

  Able to perform passive eavesdropping and intercept messages over legitimate protocol sessions.

- **Level-2 (Active attack with protocol participation and protocol disruption)**: *Able to communicate with a legitimate tag or reader by following the steps specified under the protocol and to replay, corrupt, block or inject (replace) messages.*

- **Level-3 (Active attack with secret compromise)**:

  Able to capture a legitimate tag and extract its secrets through physical layer attack and side channel attacks.

2.2 Existing Solutions

Till now, there are three major types of solutions to address the security threats in RFID systems: disabling tags, agent managements and cryptographic protocol design. Of existing solutions, cryptographic protocol solution is the most popular one because it is low-cost, without side-effect, convenient for users and it keeps post-purchase functions of RFID systems.

2.2.1 Disabling tags

In order to prevent tags from illegal scanning, people proposed "permanently disabling tags" and "temporarily disabling tags".

1. Permanent Disabling Tags

   In the EPC specification, a tag's owner can "kill" it by writing a kill password into the tag, so that a tag is destroyed and an attacker cannot trace the tag anymore. However, after being killed, a tag's post-purchase functions are all disabled [Jue06]. For instance, if a customer buys a sweater and disables its RFID tag at the shop's counter, the smart washing function cannot be achieved when he gets home. This strategy is also inapplicable in supply chains that require recycling. Therefore, permanently disabling RFID tags is not an excellent security solution.

2. Temporarily Disabling Tags [MRT09]

   As permanent disabling tags is not a good solution, researchers use the method of temporarily disabling tags instead. A tag's manufacture sets a button on the tag, and the tag's owner can press the button and input a password to choose either "sleeping" mode (tag temporarily inactive) or "wake" mode (tag active again). However, the tag's user suffers from heavy burdens of managing the passwords [Jue06]. A second method to temporarily disable a tag is to use a faraday cage [JRS03] to block the reader's signals from tag at the physical

layer. However, the faraday cage is inconvenient for mobile tags sticked to human beings such as watches.

2.2.2 Agent Management

The second type of solutions is to bring a third party into the RFID system, which takes up 11% of the publications in survey [SE09]. The third party agent can either be an RFID guardian [RCT05] or blocker tags [JRS03]. RFID guardian is a kind of battery-powered device, which can create scanning logs, achieve the access control, take charge of key infrastructure, and mediate between RFID readers and tags. Blocker tags can actively send jamming signals to disturb the communications when a reader tries to query a private tag or private zones of RFID tags illegally.

Although agent management strategy can reduce extra security functions implemented inside RFID tags, setting up an agent results in extra cost. In addition, actively jamming is illegal and may affect the nearby electronic equipment.

In [CKSK08], context-recognition technique is combined with RFID system and context-aware unit is added to the RFID system. Though this management is simple as well as efficient and it can achieve backward compatibility, it requires extra recognition unit. What's more, the recognition templates are limited, thus the recognition system is easy to be cracked by attackers.

2.2.3 Cryptographic Protocols

Compared with two previous solutions, cryptographic protocol strategy can keep tags recyclable, easy-to-use, as well as low-cost and it does not interfere other nearby electronic magnetic instruments. Therefore, my thesis focuses on cryptographic protocol solution. The basic RFID protocol is in Fig 2.1. The left protocol is a one-way authentication protocol, which means only a tag tells a reader who it is. The right part is a mutual authentication protocol, which means both reader and tag tell each other their own identities.

About 82% publications in survey [SE09] are related to cryptographic protocols. Though this strategy is the most feasible solution to RFID security problems in certain cases and does not need extra single unit like agent management, it needs extra memory and computational ability in RFID tags and extra search cost in RFID reader's database. In addition, some RFID protocol needs several communication runs between RFID reader and tags instead of basic challenge-and-respond policy.

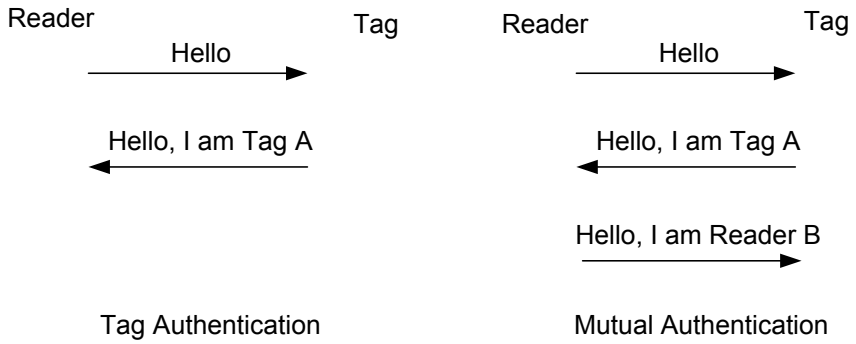


Figure 2.1: Basic Protocols

Numerous papers addressing RFID cryptographic protocols have been published recently (please refer to [Jue06] for a detailed literature survey). One concern in research realm is on tag authentication [Dim06, OSK03, Vau07] in the left part shown in Fig 2.1, and the other one is on RFID reader/tag (mutual) authentication in the right part shown in Fig 2.1, which has also been rigorously studied in the literature [CCB06] [Dim05, Dim06, JPP08, MSW05, OSK03, Tsu06].

A couple of RFID authentication protocols based on secure one-way hash functions have been proposed [WSRE03]. In one of the previous works, Ohkubo, Suzuki and Kinoshita (OSK) proposed using hash chain to update the internal states and responses to readers [OSK03]. The scheme needs to compute two different hash function values, one to update the tag's secret and the other one to compute the response that is transmitted to the reader during tag identification. This method incurs a large overhead at the reader's side due to the exhaustive search in the back-end database to identify the tag. To mitigate the high search cost, Avoine and Oechslin proposed an optimization of the scheme using a time-memory trade-off for the computation of OSK hash chains [AO05]. However, in the later works [Dim05]

and [Dim06], the authors pointed out that the optimized scheme is still vulnerable to tag impersonation attack and suffers from low scalability in the presence of attacks. Dimitriou in [Dim05] proposed a challenge-response protocol for tag-reader authentication. However, it is still possible for an adversary to de-synchronize tags, leading to a denial of service.

Pseudonym Random Function (PRF) has been used in the design of RFID protocols. In [Tsu06], Tsudik proposed YATRAP protocol for RFID authentication. It only needs a single key and a single pseudorandom function (PRF) in a tag, but it is vulnerable to de-synchronization and denial of service (DoS) attacks as the timestamps can be manipulated in this protocol. Then, Chatmon, van Le and Burmester's YATRAP+ and OTRAP [CCB06] were proposed to address the problem of YATRAP. Their schemes were essentially designed mainly for privacy-preserving identification of tags without providing reader authentication.

To reduce reader's search overhead, people proposed to use tree-structure in RFID protocols. Dimitriou proposed a tree-based privacy-preserving RFID identification scheme [Dim06]. In [MSW05], Molnar, Soppera, and Wagner proposed a tree based scheme with a high scalability of identifying tags. Under these schemes, each tag stores a group of secret keys that lie along the path of a key tree from root to leaf layer maintained by the back-end database. During RFID identification, a tag responds a group of values computed using the group of secret keys over a random challenge and the reader will use the group of responses to identify a tag. However, it is difficult to implement key updating because some keys are shared by different tags. Even worse, compromising attack exists, that is, if one tag's secret is compromised, it may affect other tags and leak their secrets.

Other than the search cost, another non-ignorable cost in RFID protocols is tag-related cost, including cryptographic computational cost in tags and reader-tag communication cost. Feldhofer achieved AES algorithm in HF tags [FDW04]. Later on, he added on SHA-1. SHA-256, MD5, MD4 in his hardware implementation of RFID system [FR06]. Chiew et al. implemented all the AES, MD5, SHA functions

as well as achieved write/read operations on the IAIK UHF tag platform. In addition, this work measured time cost of each operation. This work made tag-related cost evaluation possible [CLL+10]. In addition, more efficient pseudo-random generator and universal hash function instead of one-way hash function was proposed in [BBEG09], which could be used to reduce tag-related cost by replacing traditional hash functions such as MD5. Other than symmetric key cryptographic operations, public key cryptography was also implemented in tags, such as that ECC was achieved in HF tag [KP06].

In Chapter 3, existing RFID protocols are categorized into six groups according to their different security levels. The performance analysis of these protocols comprises two parts: one is the search cost in readers' back-end database and the other one (i.e. tag-related cost) covers the cryptographic operation and communication cost in RFID tags, which are analyzed in Chapter 4 and Chapter 5, respectively. Suggestions for balancing security and performance in RFID protocol design is given in Chapter 5.

Chapter 3: Categorization of RFID Protocols

In this chapter, we categorize the existing RFID protocols in terms of their security levels. Compared with previous work, our work is less complex and closer to real applications. Among several previous categorizations, Vaudenay's work is one of the most systematic one and widely cited by academic researchers. We will mainly focus on comparing our categorization model with Vaudenay's model.

3.1 Previous Work on Categorization

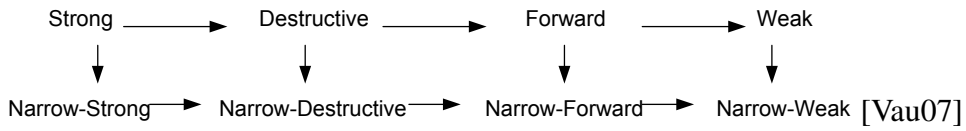


Figure 3.1: Vaudenay's Model

There exist several theoretic categorization works till now. In CCS'09, Ma et.al categorized RFID models into two groups: INP (Indistinguishability-Based Privacy) and UNP (Unpredictability-Based Privacy). INP means an attacker cannot distinguish two tags with a probability higher than a random guess. UNP means an attacker cannot predict whether a message is an output of an protocol or a random one with a probability higher than a random guess. However, this work only roughly categorized protocols into two groups and thus was not fine-grained. Another theoretic categorization work was Vaudenay's model. In [Vau07], Vaudenay categorized the RFID tag authentication privacy models into eight different categories according to adversaries' different abilities. In his model, an adversary can simulate seven oracles: $Createtag^b(ID)$, Draw(distr), Free(vtag), Launch, Sendreader(m, π), Result(π), Corrupt(vtag). **Strong** adversary is defined as who can access all the oracles. **Destructive** adversary is defined as who never uses the

target tag again after corrupting it. **Forward** adversary is defined as who can only access corrupt oracle after the target tag being corrupted. **Weak** adversary can not access corrupt oracle. **Narrow** adversary in the second line shown in Fig 3.1 can not access result(π) oracle, which returns 1 if the protocol was complete, otherwise 0. Therefore, there are eight privacy models: strong, destructive, forward, weak; narrow-strong, narrow-destructive, narrow-forward, narrow-weak. Their relationship is illustrated in Fig 3.1. Later in ASIACCS'08 [PV08], the eight model was extended into mutual authentication models. In ESORICS'08, Ng et al. further simplified the eight RFID privacy models into three categories [NSMSN08]. Till now, there still exist arguments about classification standards. Among previous works, Vaudenay's work is one of the most systematic one and widely cited. Nevertheless, one category in his model cannot be achieved, which is the strong model. Yet another category (narrow-strong) needs to be carried out by public-key cryptography (PKC) [PV08], which is not a light-weighted solution. PKC costs much more to achieve the same security level than symmetric key cryptography. In Table 3.1, Zhu et al. in [LZL08] illustrated that the key size of Elliptic Curve Cryptography(ECC) is at least twice that of the symmetric key cryptography to achieve the same security level and the key size of RSA is between 16 and 60 time as long as symmetric key. Therefore, till now, PKC is not suitable for low cost RFID applications because PKC requires more memory cost and transmission cost than symmetric key security solution and it is not a light-weighted solution for RFID system.

As PKC is infeasible for low cost RFID systems, the previous RFID protocol categorization is not appropriate in many applications. In addition, all the three categorization methods proposed by Ma, Vaudenay and Ng are based on theoretic models. Unfortunately, they did not count in the feasibility. They did not consider such protocols, which have weak security properties though, are quite relevant to practice. We re-categorize the existing RFID protocols in terms of previous references but our work mainly focuses on the symmetric cryptography protocols. To our best knowledge, this categorization can cover most existing RFID symmetric

Table 3.1: ECC Support for PKINIT, Comparable key sizes between symmetric key and public cryptography (in bits)
[LZL08]

$Symmetric$	ECC	RSA
80	160-223	1024
112	224-255	2048
128	256-383	3072
192	384-511	7680
256	512+	15360

key protocols.

3.2 Our Categorization and Typical Protocols

Similar to Vaudenay's scheme [Vau07], we categorize RFID protocols based on **anti-tracing** property and **forward secrecy** property . To measure anti-tracing property, we need to see whether a tag's response to reader changes. To measure forward secrecy property, we need to see whether a tag's internal state updates. From the an attacker's point of view, we categorize the existing RFID security protocols into six classes in terms of whether a tag's response changes and whether the tag's internal state updates once the tag is queried. The tag's response includes three situations: unchanging plain ID, unchanging Meta ID (i.e. pseudoname) and changing response, which are shown in Table 3.2 (a). It is required that based on changing responses, it is impossible for an attacker to distinguish two tags with a probability higher than a random guess. If an RFID system is implemented in the first two response strategies, the tags in this system can be traced with unchanging responses. As shown in Table 3.2 (b), the tag's internal state also comprises three ways: never updating internal state, only legitimate party updating the internal state, the tag automatically updating its internal state every time a reader queries it. After a tag's internal state updates, it is impossible for an adversary who can corrupt

a tag to obtain the tag's internal states before the corruption. In other words, the adversary cannot distinguish between two tags before corruption with a probability higher than 50% even if the adversary can corrupt one or two of the tags after updates.

(a)

	Response
Unchanging	ID
	MetaID
Changing	

(b)

	Internal State
No updating	
Updating	Legitimate Party Updating
	Automatically Updating

Table 3.2: Response and Internal State

Based on combination of tag's response and internal state, existing RFID protocols can be categorized into $3 \times 3 = 9$ categories. However, in EPC protocol, a tag sends plain ID only as a response to reader. Therefore, two situations: plain ID response with legitimate party updating internal state and plain ID response with automatically updating internal state do not exist. In addition, the case that a tag responds unchanging MetaID with automatically updating internal state is meaningless, because it is easily transferred into the case that a tag changes its response with automatically updating internal state by means of updating internal state first and responding to a reader later according to its internal state. Therefore, existing RFID protocols can be categorized into six classes:

1. EPC protocol:

   In EPC protocol, a tag does not update its internal state. The tag responds its EPC code in plain text every time it is interrogated by a reader, so EPC protocol can guarantee neither forward secrecy nor anti-tracing. It is the weakest protocol. EPC protocol only uses two passwords to protect RFID tags: one is the kill password which is used to destroy the tag and the other one is the access password to prevent malicious readers from writing into a tag arbitrarily. Of our concern, EPC protocol is the baseline of all RFID security protocols.

2. Tracing Protocols:

Better than the EPC protocol, a tag implemented with a tracing protocol does not respond its EPC code directly to a reader. Instead, the tag responds with its unchanging MetaID back to reader. Therefore, to some extent, it protects the tag's privacy. However, through the same MetaID (i.e. pseudoname), attackers can still trace the same tag. Additionally, in tracing protocols, a tag does not update its internal state. Thus, tracing protocols cannot achieve forward secrecy, which means if a tag is corrupted and its internal state is extracted by a malicious reader, the tag's previous internal states can be retrieved by the malicious reader successfully. Hash-lock protocol [WSRE03] and Key-Sharing protocol [JPP08] belong to this category.

3. Strong Anti-tracing Protocols:

   Compared with previous two categories, strong anti-tracing property is added to Category III protocols. Thanks to functions such as PRF (Pseudo random function) in tags, tags can respond with different, unlinkable values every time a reader interrogates them. As a result, with changing responses, an adversary cannot distinguish two tags with a probability higher than a random guess. Nevertheless, without updating internal state, strong anti-tracing protocols still cannot guarantee forward secrecy. Strong anti-tracing protocols comprise Random Hash-lock [WSRE03], Big Brother [Dim06], MW Tree [MW04], Dual Mode protocol [CLLD09a], improved BMM protocol [LLM$^+$09], UNP protocol [MLDL09], zk-privacy protocol [DLYZ10] etc..

4. Weak Anti-tracing, Weak Forward Secrecy Protocols:

   The next three categories achieve forward secrecy on different levers. Weak forward secrecy is added in Category IV protocols. In Category IV, only legitimate parties can update tags' internal states and change tags' responses. Thus, a tag's response and internal state keep unchanging between two legitimate parties. As a consequence, this kind of protocol is still vulnerable to

partial tracing attack and it can achieve only partial forward secrecy. For instance, in a factory, the legitimate readers only interrogate tags at the entrance and exit spots. Therefore, inside the factory, an adversary can easily trace the same tag and if the tag is corrupted in the factory, the tag's previous internal states as early as the entrance spot can be restored. Update Key-sharing [CLM+09], LD supply chain protocol [LD07], lightweight RFID protocol [Dim05] and HM hash-based protocol [HM04] are examples of category IV protocols.

5. Strong Anti-Tracing, Weak Forward Secrecy Protocols:

   In Category V protocols, with the help of PRFs in tags, tags respond with changing, unlinkable messages every time and achieve strong anti-tracing all the time rather than partial tracing in Category IV. However, as same as in Category IV protocols, a tag updates its internal states only after successfully authenticating legitimate parties in Category V protocols. Thus, between two honest parties, forward secrecy still cannot be achieved in this category. A couple of protocols belong to this category, including: SM protocol [SM08], revised SM [CLLD09b].

6. Strong Anti-Tracing, Strong Forward Secrecy Protocols:

   The last category owns the strongest security properties in our categorization. In Category VI, a tag automatically updates not only its response but also the internal state every time, regardless of whether the querying reader is legitimate or not. Therefore, they can achieve both strong anti-tracing and strong forward security. OSK [OSK03], RFIDDOT [Dim08], narrow-destructive protocol [PV08] are three typical Category VI protocols.

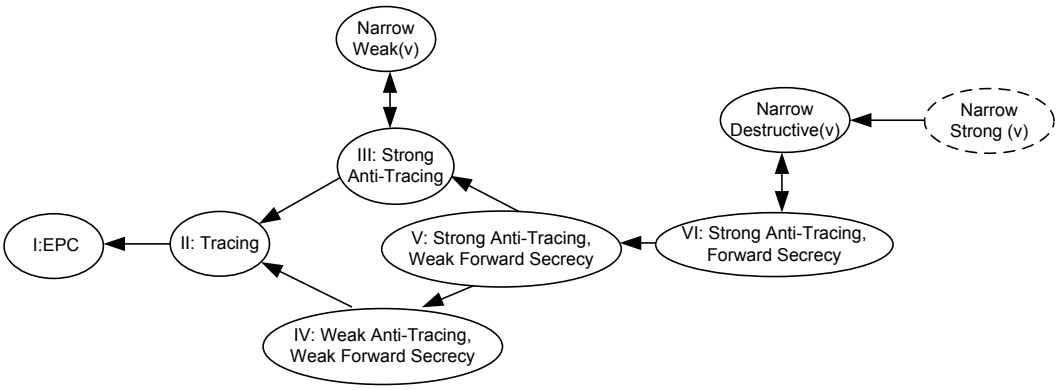


Figure 3.2: Comparison With Vaudenay's Model

3.3 Comparison with Vaudenay's model

Similar to Vaudenay's model, our categorization is based on protocols' different security levels. There are two main differences between our work and Vaudenay's work. The first difference is that we do not consider the side-channel information obtained by attackers, whereas the $result(\pi)$ oracle is inaccessible by adversaries in [Vau07]. Therefore, our categorization is located within narrow models in Vaudenay's categorization. Second, we add two weak categories and eliminate Vaudenay's strong model, which has to be achieved by PKC. Our Category I (EPC Protocol) and II (Tracing Protocols) are weaker than Vaudenay's weakest model. However, they are most practical in reality. Category III (Strong anti-tracing Protocols) is nearly equal to narrow-weak model in Vaudenay's categorization. Category IV and category III are parallel with each other, because Category IV is weaker in aspect of anti-tracing than category III, while stronger in aspect of forward secrecy. Category V protocols is stronger than both category III and category IV. Our strongest category protocols are similar to Vaudenay's narrow-destructive models. Note that our categorization does not cover Vaudenay's narrow-strong model, which demands PKC in implementation. The relationship between my categorization and Vaudenay's model is shown in Fig 3.2. In summary, my categorization is close to industrial applications and focuses on symmetric cryptography.

Chapter 4: RFID Security and Search Cost Analysis

One of the performance overhead for RFID security is the search cost in readers' back-end database. Usually, there exist a large number of tags in RFID applications and each tag has a unique ID. The look-up cost is non-ignorable on the reader side. If a tag responds its their EPC code or MetaID directly to a reader, the search cost of the corresponding tag is constant in terms of exact match. However, in order to protect the tag's privacy, the response is usually calculated by a function F based on a secret k and a random number r, like $(F(k, r))$. Hence, a reader may need an exhaustive search in its database to find the tag's information. Accordingly, the search cost is relatively high.

4.1 Existing Approaches

Assume there are N tags in an RFID system. The random hash-lock protocol in [WSRE03] requires an exhaustive search in the reader's database to identify a tag, so the overhead of this protocol is $O(N)$. In the OSK protocol [OSK03], the reader has to calculate hash values with $O(N)$ complexity. Molnar and Wagner's method manages the keys of tags in a tree structure [MW04] with a search cost of $O(log(N))$. Although the cost is already much better than the exhaustive search in other protocols, it is still non-ignorable when the number of tags increases to a large one. In such cases, the scalability of tag search may become a performance bottleneck.

In [BdMM08], Burmester, Medeiros and Motta (BMM) proposed an RFID authentication protocol with constant key-lookup to balance the privacy requirement and scalability. To the best of our knowledge, this protocol is one of the most scalable solutions that preserve privacy as claimed. We take BMM protocol as an

example to analyze the search cost in the reader's database. However, this protocol has security vulnerabilities. We will analyze the vulnerabilities and give our improvement.

4.2 Case Study on BMM Protocol

Now, we take BMM protocol as a case study to analyze the security and a reader's search cost.

4.2.1 Notation

First, a lot of notations of BMM protocol is introduced. If $A(\cdot, \cdot, ...)$ is a randomized algorithm, then $y \leftarrow A(x_1, x_2, ...; cn)$ means that y is assigned the unique output of the algorithm A on inputs $x_1, x_2, ...$ and coins cn. Let g be a pseudorandom function (PRF) [GGM86]. If S is a set, then $s \in_R S$ indicates that s is chosen uniformly at random from S. If $x_1, x_2, ...$ are strings, then $x_1||x_2||\cdots$ denotes the concatenation of them. If x is a string, then $|x|$ denotes its bit length in binary code. Let ε denote the empty string. If S is a set, then $|S|$ denotes its cardinality (i.e. the number of elements of S). If ctr is a counter which starts from n_1 and ends with n_ℓ, then $ctr(j)$ denotes its jth value, i.e. $ctr(j) = n_j$, where $1 \leqslant j \leqslant \ell$. Let IV be an initial vector for the PRF g.

4.2.2 BMM Protocol

In the RFID system constructed by BMM protocol in Fig 4.1, there is a set-up procedure which initializes the reader and every tag. Then, they will engage in a protocol to identify the tag. The whole RFID system is described as follows.

Setup: When creating a new tag T, the system generates a secret key k, a pseudonym seed q, a one-time pseudonym r, a counter $ctr = 1$, and a flag $mode = 0$. Then it sets up the initial state information of the tag T as the tuple

Tag		Reader
$(k, r, q, mode, ctr)$		$D = \{k, r_i, q, q_i^1, \cdots, q_i^\ell, i \in \{old, cur\},\}$
	$\xleftarrow{c}$	$c \in_R \{0,1\}^n$
If $mode = 0$ then $ps \leftarrow r$		
Else		
$ps \leftarrow g(k; q\|\|IV\|\|ctr)$,		
Update ctr		
$\nu_0\|\|\nu_1\|\|\nu_2 \leftarrow g(k; ps\|\|c)$		
$auth \leftarrow \nu_1$	$\xrightarrow{ps\|\|auth}$	If $(k, ps) \notin D$ then $REJECT$
		Else $\nu_0'\|\|\nu_1'\|\|\nu_2' \leftarrow g(k; ps\|\|c)$
		If $\nu_1' \neq auth$ then $REJECT$
		Else $conf \leftarrow \nu_2'$
	$\xleftarrow{conf}$	
If $conf = \nu_2$ then		If $ps = r_{cur}$ then
If $mode = 0$ then $r \leftarrow \nu_0$		$r_{old} \leftarrow r_{cur}$ and $r_{cur} \leftarrow \nu_0'$
Else		Else if $ps = r_{old}$ then $r_{cur} \leftarrow \nu_0'$
$mode \leftarrow 0$ and $q \leftarrow \nu_0$		Else if $ps = q_{cur}^j$ then $q \leftarrow \nu_0'$ and
Else $mode \leftarrow 1$		$\{q_{old}^i \leftarrow q_{cur}^i\}_{i=1}^\ell$ and
		$\{q_{cur}^i \leftarrow g(k; q\|\|IV\|\|ctr(i))\}_{i=1}^\ell$
		Else if $ps = q_{old}^j$ then $q \leftarrow \nu_0'$ and
		$\{q_{cur}^i \leftarrow g(k; q\|\|IV\|\|ctr(i))\}_{i=1}^\ell$
		Output $ACCEPT$

Figure 4.1: BMM Protocol

$(k, q, r, ctr, mode)$. The system also associates the tag T with its identity ID_T in the reader's database by initiating a tuple $(r_{old}, r_{cur}, q^1_{old}, \cdots q^\ell_{old}, q^1_{cur}, \cdots q^\ell_{cur}, k, q, ID_T)$, where $r_{old} = r_{cur} = r$ and $q^j_i = g(k; ||q||IV||ctr(j)$, for $i = \{old, cur\}$, and $j = 1, \cdots \ell$.

The BMM Protocol: It runs in three rounds:

Round 1. First, the reader starts the protocol by sending a challenge c to the tag. Upon receiving c, the tag first checks its $mode$ state: if $mode = 0$, it sets the pseudonym $ps = r$; otherwise, it computes $ps = g(k; q||IV||ctr)$ and updates the counter $ctr = ctr+1$. Then, the tag calculates $\nu_0||\nu_1||\nu_2 = g(k; ps||c)$. Here, ν_0 is used to replace the pseudonym r; $auth = \nu_1$ is used to authenticate itself to the reader, and ν_2 is used to authenticate the reader.

Round 2. The tag sends the message $ps||auth$ to the reader. Upon receiving $ps||auth$, the reader requests to its back-end database to look up the tuple $(r_{old}, r_{cur}, q^1_{old}, \cdots\ q^\ell_{old}, q^1_{cur}, \cdots q^\ell_{cur}, k, q_0, ID_T)$ such that $r_i = ps$ or $q^j_i = ps$, where $i = \{old, cur\}$ and $j = 1, \cdots \ell$, through using ps as an index. If the tag is de-synchronized within ℓ times, we can find the tuple in constant time by $2\ell + 2$ indexes. If the tuple is found, the reader calculates $\nu'_0||\nu'_1||\nu'_2 \leftarrow g(k; ps||c)$ and accepts the tag if $auth = \nu'_1$. Otherwise, the tag is rejected. If a tag is accepted, the reader prepares a confirmation message $conf \leftarrow \nu'_2$.

Round 3. The reader sends the confirmation message $conf$ to the tag. The tag authenticates the reader by checking whether $conf = \nu_2$. If the reader is successfully authenticated, the tag then updates its pseudonym: if $mode = 0$, it updates the pseudonym $r = \nu_0$; if $mode = 1$, it updates pseudonym seed $q = \nu_0$ and keep the pseudonym r unchanged. If the reader is not authenticated, the tag sets $mode = 1$ and does nothing else. On the reader side, it updates the tuple $(r_{old}, r_{cur}, q^1_{old}, \cdots, q^\ell_{old}, q^1_{cur}, \cdots$

$q^{\ell}_{cur}, k, q_0, ID_T)$ associated with the tag as follows. If $ps = r_{cur}$, it updates $r_{old} = r_{cur}$ and $r_{cur} = \nu'_0$. If $ps = r_{old}$, it only updates $r_{cur} = \nu'_0$. If $ps = q^j_{old}$ for some j between 1 and ℓ, it updates $q = \nu'_0$ and $q^j_{cur} = g(k; ||q||IV||ctr(j)$ for $j = 1, \cdots \ell$. If $ps = q^j_{cur}$ for some j between 1 and ℓ, it updates $q = \nu'_0$, $q^j_{old} = q^j_{cur}$ and $q^j_{cur} = g(k; ||q||IV||ctr(j)$ for $j = 1, \cdots \ell$.

4.2.3 Attacks on BMM Protocol

In following analysis, we identify the shortcoming in BMM protocol [BdMM08] and propose an improved protocol accordingly. We argue that the improved protocol provides stronger privacy than the BMM protocol, while the performance of the improved protocol is the same as the BMM protocol. Our contributions are summarized below:

1. We analyze the BMM-protocol and find a subtle flaw, by which we can break the privacy property, namely untraceability. Exploiting this flaw, we design an easy-to-launch attack under a weak adversary model. Under our attack, an adversary can easily trace a tag in a supply chain party. Thus, one by one, we can trace such a tag in a whole supply chain if the BMM protocol is implemented.

2. To improve the protocol, we propose an anonymous RFID authentication protocol that can fulfill all privacy claims of [BdMM08], including defense against eavesdropping attack, spoofing attack, replay attack, de-synchronization attack, tracing attack and compromising attack.

Burmester, Medeiros and Motta claimed that this protocol can *"support anonymity with constant key-lookup cost; however, it suffers from entrapment attacks"* [BdMM08]. To preserve the privacy of a queried tag, an adversary that eavesdrops over the protocol should not be able to figure out the identifier of the tag with higher

likelihood than a pure random guess. The same should also apply to an unauthorized reader that attempts to query the tag. In other words, the protocol should ensure "tag anonymity", in terms of session unlinkability: an adversary should not be able to link together two or more protocol sessions involving the same tag (regardless whether the identity of the tag is known or not) to track the activities of the tag. To achieve this, any two protocol exchanges involving the same tag must appear reasonably random such that the adversary cannot differentiate it with non-negligible probability from two protocol exchanges involving two different tags.

Unfortunately, there exist some flaws in the updating procedures in the design of BMM protocol. The flaws can be subsequently exploited to launch a simple attack to trace a tag in a series of protocol runs.

Adversary Model

We use the adversary model in Chapter 2.1. It is reasonable to assume that a higher level adversary also possesses the abilities of all levels preceding it, i.e. a level-3 adversary has the abilities of level-1 and level-2 adversaries, as well as the set of additional abilities of physical layer attacks and side channel attacks. As we will be showing in next subsection, our attack requires a relatively weak adversary model (*w.r.t.*, a level-2 adversary), where an adversary has limited ability to communicate with a legitimate tag following protocol steps.

Different kinds of attacks can achieve variable goals. Eavesdropping attacks can track a tag successfully if the tag's responses keep same. Attackers can communicate with trusted readers and trusted tags through spoofing and replay attack. De-synchronization attacks can interrupt regular communications between trusted readers and tags through blocking, modifying and injecting messages. Denial of Service (DoS) attacks mean that a legitimate reader is flooded with useless messages so that it cannot communicate with legitimate tags normally.

Three-Run Interleave Attack

We first give the intuition behind our attack. We observe that the state information (index) 'r' in the tag always keeps unchanged in the protocol executions when $mode = 1$ and $conf = \nu_2$ (see Figure 4.1). It means that the tag will reply with the same response in the next interrogation. Our attack follows this observation and uses a 'three-run interleave' technique to push the tag into the state of $mode = 1$ and $conf = \nu_2$.

As mentioned in Chapter 2.1, we assume a level-2 adversary as the malicious reader, denoted by $\mathcal{R}^M$. We denote a legitimate tag by $\mathcal{T}$ and a trusted reader by $\mathcal{R}^T$. The attack consists of three runs, during which $\mathcal{T}$ is interrogated by $\mathcal{R}^M$ twice and by $\mathcal{R}^T$ once. We present the attack in detail as follows.

Tag		Malicious Reader
$(k, r, q, mode = 0, ctr)$		$c \in_R \{0,1\}^n$
	$\xleftarrow{c}$	
$ps \leftarrow r$ and Update ctr		
$\nu_0 \| \nu_1 \| \nu_2 \leftarrow g(k; ps \| c)$		
$auth \leftarrow \nu_1$	$\xrightarrow{ps \| auth}$	Receive and Store
		Send another
$\nu_2 \neq \overline{c}$		Random Number $\overline{c}$
and $mode \leftarrow 1$	$\xleftarrow{\overline{c}}$	

Figure 4.2: First Run of the Attack

- **First Run: $\mathcal{R}^M$ interrogates $\mathcal{T}$**

  This first run of our attack is illustrated in Figure 4.2. During the first protocol run, $\mathcal{R}^M$ interrogates $\mathcal{T}$ with an incomplete protocol execution. We assume that $\mathcal{R}^M$ can launch attacks after several legitimate communications between $\mathcal{R}^T$s and $\mathcal{T}$, so we can consider the initial status of $\mathcal{T}$ as $mode = 0$. After sending a challenge c, $\mathcal{R}^M$ receives the reply message $ps||auth = r||\nu_1$ from

$\mathcal{T}$. As $\mathcal{R}^M$ does **not** share any secret with $\mathcal{T}$, it cannot compose the correct confirmation message for $\mathcal{T}$. Instead, $\mathcal{R}^M$ sends a random value $\overline{c}$ to $\mathcal{T}$. At the tag's side, $\overline{c}$ cannot be verified against $conf$, so $\mathcal{T}$ changes its status into an attacked state with $mode = 1$. To this end, $\mathcal{R}^M$ stores the reply '$r||\nu_1$' and continues to the next step.

Note that if $\mathcal{R}^M$ sends queries to a tag continuously, he/she can only obtain the unlinkable information $ps||auth$. Therefore, to get useful information, which can link the same tag by comparing 'r', the adversary intentionally involves a trusted reader $\mathcal{R}^T$ in the second run.

- **Second Run: $\mathcal{R}^T$ interrogates $\mathcal{T}$**

  The second run of our attack is shown in Figure 4.3. During the second protocol run, $\mathcal{T}$ is put forward and interrogated by a trusted reader $\mathcal{R}^T$ with a complete protocol execution, while $\mathcal{R}^M$ does nothing. Note that in the first run of our attack, $\mathcal{T}$ toggles its $mode$ in $\mathcal{T}$ to '1'; therefore, after $\mathcal{T}$ receives the confirmation message from the legitimate reader, its $mode$ is changed into '0'. As now, $\mathcal{T}$ only updates q into ν_0 but keeps r unchanged.

- **Third Run: $\mathcal{R}^M$ interrogates $\mathcal{T}$**

  During the third protocol run, $\mathcal{R}^M$ interrogates with $\mathcal{T}$ again as in the first run for tracing the same tag $\mathcal{T}$ that has been interrogated in the first run. To achieve this, $\mathcal{R}^M$ sends the same challenge c to the tag and expects a repeated reply by $\mathcal{T}$. Recall that in the second run, a successful protocol run between $\mathcal{R}^T$ and $\mathcal{T}$ toggles $\mathcal{T}$ to a secure status $mode = 0$. Following the protocol, $\mathcal{T}$ shall reply with $ps||auth = r||\nu_1$, which is the same authentication information as that in the first run. It is thus easy for the attacker to trace the tag $\mathcal{T}$ by comparing the $ps||auth$ values.

Tag		Legitimate Reader
$(k, r, q, mode = 1, ctr)$		$D = \{k, r_i, q, q_i^1, \cdots, q_i^\ell,$ $i \in \{old, cur\}\}$,
$ps \leftarrow g(k; q\|\|IV\|\|ctr)$,	$\xleftarrow{c'}$	$c' \in_R \{0,1\}^n$
Update ctr		
$\nu_0\|\|\nu_1\|\|\nu_2 \leftarrow g(k; ps\|\|c')$		
$auth \leftarrow \nu_1$	$\xrightarrow{ps\|\|auth}$	If $(k, ps) \notin D$ then $REJECT$
		Else $\nu_0'\|\|\nu_1'\|\|\nu_2' \leftarrow (k; ps\|\|c')$
		If $\nu_1' \neq auth$ then $REJECT$
		Else $conf \leftarrow \nu_2'$
	$\xleftarrow{conf}$	
If $conf = \nu_2$		If $ps = q_{cur}^j$ then $q \leftarrow \nu_0'$ and
$mode \leftarrow 0$ and $q \leftarrow \nu_0$		$\{q_{old}^i \leftarrow q_{cur}^i\}_{i=1}^\ell$ and
		$\{q_{cur}^i \leftarrow g(k; q\|\|IV\|\|ctr(i))\}_{i=1}^\ell$
		Else if $ps = q_{old}^j$ then
		$q \leftarrow \nu_0'$ and
		$\{q_{cur}^i \leftarrow g(k; q\|\|IV\|\|ctr(i))\}_{i=1}^\ell$
		Output $ACCEPT$

Figure 4.3: Second Run of the Attack

Discussions

We stress that our attack is practical. There could be a number of ways to launch such an attack.

Recall that in the first protocol run of our attack, a malicious reader interrogates with a legitimate tag. We can further reduce this requirement if the adversary has minimum eavesdropping and blocking capabilities: in the first run, the adversary eavesdrops the first two protocol messages and blocks the third messages to make the protocol incomplete. Thereafter, the tag is triggered into an insecure state and the reader updates the status for the record of this tag. The attack continues with a successful second run and an incomplete third run (same as that of the first run). By comparing the eavesdropped messages in the first run and the third run, the adversary can trace the tag. Such an adversary is more stealthy as no active interrogation between a malicious reader and a legitimate tag is needed [1].

In summary, the attack can be extended, but not limited to the following forms:

- $\diamond \dashrightarrow \mathcal{R}^M \dashrightarrow \mathcal{R}^T \dashrightarrow \mathcal{R}^M \dashrightarrow$
- $\diamond \dashrightarrow \mathcal{R}^T_{\mathcal{A}} \dashrightarrow \mathcal{R}^T \dashrightarrow \mathcal{R}^T_{\mathcal{A}} \dashrightarrow$
- $\diamond \dashrightarrow \mathcal{R}^M \dashrightarrow \mathcal{R}^T \dashrightarrow \mathcal{R}^T_{\mathcal{A}} \dashrightarrow$
- $\diamond \dashrightarrow \mathcal{R}^T_{\mathcal{A}} \dashrightarrow \mathcal{R}^T \dashrightarrow \mathcal{R}^M \dashrightarrow$

Where $\mathcal{R}^T_{\mathcal{A}}$ denotes an adversary's presence in an interrogation between a trusted reader and a legitimate tag.

4.2.4 Cracking a Whole Supply Chain by Using Basic Attack

Based on the basic three-run interleave attack, more advanced attacking strategies are designed to crack an RFID-enabled supply chain that implements the BMM protocol.

[1] Note that in the third run, a different challenge c'' could be used by a trusted reader to challenge the tag. As long as the r value is not updated in the second run, the ps value is still the same as the one in the first run.

Assumptions

We need to make several reasonable assumptions about an RFID-enabled supply chain before we elaborate on our attacking strategies.

1. ***Trusted Zone:***

   We consider a geographically distributed supply chain, in which each party in the supply chain may receive tagged articles, process these articles, and ship them out. For simplicity, we consider the area as a trusted zone inside a supply chain party, and public zone outside. An adversary is not able to interact with a legitimate tag in a trusted zone, but can interrogate with a tag in the public zone.

2. ***One-time Authentication:***

   While tagged articles are being processed by a supply chain party, the authentication is performed only once (*e.g.*, typically at the entry point of the trusted zone). This is reasonable as authentication procedure is much more expensive and time-consuming than identifier scanning procedure. As the area inside a supply chain party is considered as a trusted domain, indeed no additional authentication is necessary. While multiple scanning for identifying the tags is still allowed to facilitate other operations (which are not security related). This is to guarantee that only one successful session of authentication protocol is conducted in a trusted zone so that once the articles are shipped out to the public zone, the adversary can launch the tracing attack.

3. ***Sticky Adversary:***

   We assume that an adversary may possess multiple readers at multiple locations or equivalently possess one reader at multiple instant locations. In other words, we assume an ubiquitous adversary who is able to stick on the targeted articles in the public zone along a supply chain.

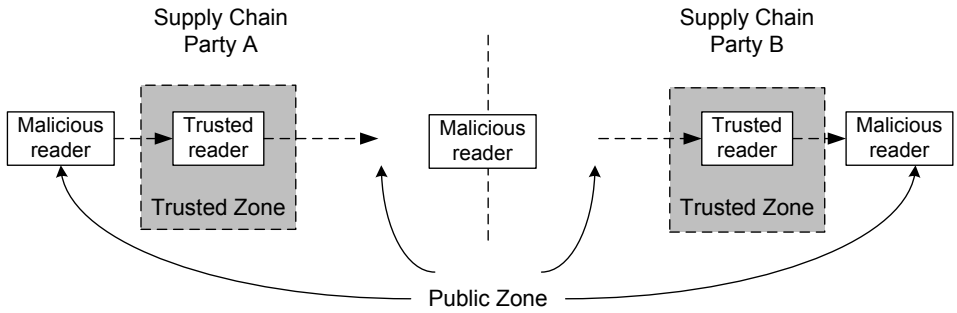


Figure 4.4: An Example for Cracking Supply Chain System

With these assumptions, we illustrate how to crack a supply chain system as in Figure 4.4, where two supply chain parties are involved. In an attack, the adversary can setup malicious readers in the public zones near each supply chain party. Furthermore, two attacking strategies are given below.

Attacking Strategies

There are two potential strategies to attack a supply chain. The first focuses on a special goods, for instance, a valuable jewelry or a secret weapon. The second one can totally crack the supply chain and its threat is much more severer.

Case 1: Tracing a Single Tag along Supply Chain

Suppose an adversary targets on a particular article with an RFID tag $\mathcal{T}$. Before it arrives at supply chain party A, a malicious reader can launch its attack by interrogating with $\mathcal{T}$ and obtaining a ps value ($ps = r$) specific to this tag. Inside the domain of party A, $\mathcal{T}$ is authenticated once and processed in some other ways. At last, the article attached with $\mathcal{T}$ is shipped out. Once again, a malicious reader scans all outbound articles and find this particular tag with the pseudonym ps. Following on, the adversary repeats the attacks at various transportation locations visited by this article. Eventually, a list of visited sites of the article, [$\dashrightarrow A \Rightarrow B \Rightarrow C \Rightarrow D \Rightarrow E \dashrightarrow$], are recorded, which enables the total visibility of this article (in the supply chain, which is serious breach of its privacy). The tracing attack is illustrated in Figure 4.5.

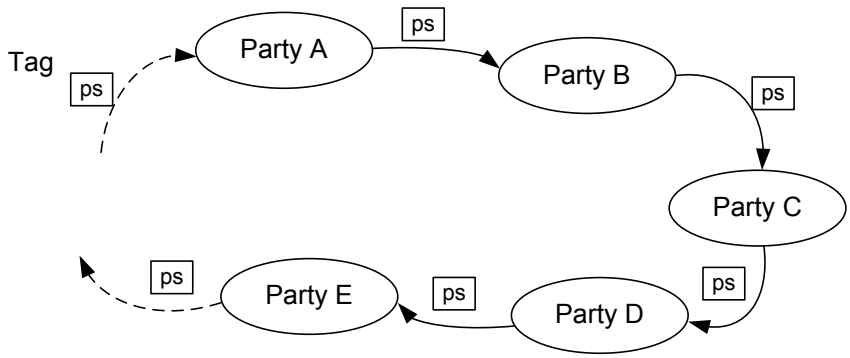


Figure 4.5: Tracing a Single Tag along its Supply Chain

Tag	ps	Location 1	Location 2	Location 3	Location 4	Location 5	$\cdots$
Tag 1	09310A78	$\surd$	$\surd$	$\times$	$\times$	$\surd$	$\cdots$
Tag 2	38901D43	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\cdots$
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
Tag 100	9A7B2811	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\cdots$

Figure 4.6: The Adversary's Database

Case 2: Tracing Multiple Tags and Constructing Supply Chain Map

Suppose an adversary, for the purpose of obtaining commercial secret, targets on a manufacture who supplies its goods to various distributors, retailers, *etc.*, via complex supply chain paths. To construct such a map, he/she needs to trace all the goods attached with tags along their supply chains. As such, the adversary first builds a database for all the tags scanned immediately after the goods are shipped out. Suppose 100 tags are being scanned and recorded in the database, as shown in Figure 4.6. For each record of the database, $\surd$ (or $\times$) represents whether the tag is scanned at certain locations or not. 'ps' denote the pseudonyms of a tag, for simplicity, $|ps| = 32$. As long as the adversary has enough resources to monitor all potential locations via a number of supply chains, it will finally draw a complete map for all delivery paths.

We assume that there are L possible locations for each tag, and the number of total tags is N. An attacker only needs to set up a database with size of $O(L \times N)$. He/she can efficiently query the information of a tag in polynomial time.

4.2.5 Improving BMM Protocol

Tag		Reader
$(k, r, q, mode, ctr)$		$D = \{k, r_i, q, q_i^1, \cdots, q_i^\ell, i \in \{old, cur\}\}$
	$\xleftarrow{c}$	$c \in_R \{0,1\}^n$
If $mode = 0$ then $ps \leftarrow r$		
Else		
$ps \leftarrow g(k; q\|\|IV\|\|ctr)$,		
Update ctr		
$\boxed{\nu_0\|\|\nu_1\|\|\nu_2\|\|\nu_3 \leftarrow g(k; ps\|\|c)}$		
$auth \leftarrow \nu_1$	$\xrightarrow{ps\|\|auth}$	If$(k, ps) \notin D$ then $REJECT$
		Else $\boxed{\nu_0'\|\|\nu_1'\|\|\nu_2'\|\|\nu_3' \leftarrow g(k; ps\|\|c)}$
		If $\nu_1' \neq auth$ then $REJECT$
		Else $conf \leftarrow \nu_2'$
	$\xleftarrow{conf}$	
If $conf = \nu_2$ then		If $ps = r$ then $r \leftarrow \nu_0'$
If $mode = 0$ then $r \leftarrow \nu_0$		Else if $ps = q_{cur}^j$ then $q \leftarrow \nu_0'$
Else		$\{q_{old}^i \leftarrow q_{cur}^i\}_{i=1}^\ell$ and $\boxed{r \leftarrow \nu_3'}$
$mode \leftarrow 0$ and		$\{q_{cur}^i \leftarrow g(k; q\|\|IV\|\|ctr(i))\}_{i=1}^\ell$
$q \leftarrow \nu_0$ and $\boxed{r \leftarrow \nu_3}$		Else if $ps = q_{old}^j$ then
Else $mode \leftarrow 1$		$q \leftarrow \nu_0'$ and $\boxed{r \leftarrow \nu_3'}$
		$\{q_{cur}^i \leftarrow g(k; q\|\|IV\|\|ctr(i))\}_{i=1}^\ell$
		Output $ACCEPT$

Figure 4.7: Improved BMM Protocol

We observe that the main reason that the BMM protocol is vulnerable to our three-run interleave attack is that the pseudonym 'r' shared between the legitimate tag and the trusted reader is not properly updated. Intuitively, we solve the problem by updating the pseudonym r at both side after the third protocol message is sent

even if the $mode$ is 1 for the tag.

Improved Protocol

Our improved protocol is shown Figure 4.7. In the first round, our protocol is the same as the BMM protocol except that we separate the result $g(k; ps||c)$ into four parts ν_0, ν_1, ν_2 and ν_3. The new part ν_3 is used to update r when the tag's $mode = 1$, and other parts are kept the same as those of the original BMM protocol. In the second round, the reader also needs to divide the result of $g(k; ps||c)$ into four parts ν'_0, ν'_1, ν'_2 and ν'_3. Here, ν'_3 is used to update the reader when the received $ps = q_i^j, i \in \{old, cur\}, j = 1, 2 \cdots \ell$, and the reader keeps other operations the same as BMM protocol. In the third round, after receiving the confirmation message in the protocol, we update the status of r at the tag's side with $r \leftarrow \nu_3$ when '$mode = 1$' holds in the tag. In this round, we also update the status as described in the boxed parts at the reader in Figure 4.7. Since the pseudonym 'r' is updated whenever the $mode$ is 0 or 1, the response of the tag behaves randomly at every interrogation. Therefore, our three-run interleave attack is no longer feasible.

Security Analysis

We analyze the improved protocol regarding some important security properties. The essential objective of the protocol is to achieve mutual authentication between a reader and a tag without disclosing the tag's identity to a third party, and it is based on a classic challenge-response mechanism. Without the shared secret, no polynomial probabilistic time (PPT) adversary can generate the authentication messages transferred between the two parties.

Our improved protocol's main purpose is to protect the tags' privacy, which means to keep tags' anonymity and untraceablity. Our improved protocol prevents tags from *tracing attack*. The meaning of untraceability contains two aspects: 1) The outputs of a tag in any two sessions are unlinkable, and 2) The outputs of readers are independent from those of tags. First of all, we analyze the outputs of any

two sessions of a tag. For any two session $i\ and\ j,\ i \neq j$ of a tag, let $ps(i)||auth(i)$ and $ps(j)||auth(j)$ denote the output of the session i and j, respectively.

$$ps = \left\{ \begin{array}{ll} r, & mode = 0 \\ g(k; q||IV||ctr), & mode = 1 \end{array} \right\}$$

If $mode = 0$, then $ps = r$, and r is updated by a PRF $g(\cdot)$ in the tag after every successful protocol; otherwise, $ps = g(k; q||IV||ctr)$, the output of PRF $g(\cdot)$. Therefore, whether $ps = r$ or $ps = g(k; q||IV||ctr)$, $ps(i)$ and $ps(j)$ are independent as the output of a PRF are pairwise independent. The latter part of the tags' output is $auth = \nu_1$ which is a part of $g(k; ps||c)$ ($g(\cdot)$ is a PRF). Therefore, $auth(i)$ and $auth(j)$ are independent and unlinkable. As a result, $ps(i)||auth(i)$ is independent from $ps(j)||auth(j)$.

Second, we illustrate the output of the reader is independent from the output of a tag. We consider the output of tag is $ps||auth$ and the output of the reader is $conf$. ps is the input of the PRF $g(k; ps||c)$, and $conf$ is the output of PRF $g(\cdot)$. As the input and output of a PRF are independent, ps is independent from $conf$. The $auth = \nu_1$ is the second part of the output $g(k; ps||c)$, and $conf = \nu_2'$ is the third part of the output of the PRF $g(k; ps||c)$. Therefore, $auth$ is independent from $conf$. In all, the output of tag $ps||auth$ is independent from the output of the reader $conf$. Thus, the independence of outputs between different sessions of a tag and the independence of outputs between a reader and a tag guarantee the privacy of tags, and attackers cannot trace a tag by eavesdropping or active interrogations.

Based on challenge-and-respond technique, mutual authentication, PRF in both tag and reader, and update processes, Level-2 attacks cannot be applied here, for instance, *de-synchronization attack*. Because the trusted reader keeps not only the newly updated values, but also the old values corresponding to a former corrupted protocol run, if a tag is pushed de-synchronized with the legitimate reader by a malicious adversary, it can still be recognized by referring to the older record $q_{old}^i, i = 1, 2, \cdots, N$ in the database. By successful mutual authentication, the

reader and tag can be re-synchronized again. As we argue in Chapter 2.1, our improved protocol can prevent level-2 attack, so it can possess the ability of counteracting weaker attacks. To counteract Level-1 attacks, for example, *eavesdropping attack*, an adversary can only obtain the challenge c and pseudonyms $ps||auth$ and ν_2', which are generated by PRF, but nothing else. Level-1 adversaries cannot link the information together to trace a tag, either. To prevent Level-2 attackers, the challenge-and-respond technique protects the reader from Denial-of-Service (DoS) attack. In addition, since fresh random numbers are generated by both the reader and the tag for mutual authentication and both the tag and the reader update their states after a successful protocol run, simple *spoofing* and *replay attacks* have negligible success rate. In addition, unlike some tree-based RFID protocol [MSW05], if some tags are compromised unfortunately, released information will not affect other tags' secrecy due to that tags do not share secrets in our protocol.

Nevertheless, the improved protocol does not incur any additional high cost with respect to storage and computation. Therefore, the lightweightness of the BMM protocol is maintained. As stated in [BdMM08], the database stores limited numbers of q_i^j, when these numbers are used up, the BMM protocol suffers from an "entrapment attack". The "entrapment attack" means *"the tag is prevented from communicating with authorized readers and can only be interrogated by the adversary"* [BdMM08]. In conclusion, as mentioned in Chapter 4.2.3, the security analysis we conducted is limited to level-1 to level-2 adversaries, while level-3 adversary is more powerful and may bring more harmful attacks to the existing protocol.

4.2.6 Case Study Summary

So far we have taken BMM protocol [BdMM08] as an example to investigate the security and scalability. We found a subtle flaw in this protocol. Under a weak adversary model, an attacker can launch a three-run interleave attack to trace and

identify a tag. Further on, complex attacking strategies can be constructed on cracking the whole supply chain using such an authentication protocol. We improve this protocol by eliminating the flaw in BMM protocol. We provide a security analysis on the improved protocol and claim that it meets its security requirements and that it is as efficient as the original protocol in each invocation. Though the readers' search cost of this protocol is constant by pre-computing all the indexes, it is still vulnerable to "entrapment attack" when the indexes are used up. To our best knowledge, the improved BMM protocol is an outstanding one to balance security properties and search cost.

4.3 Search Costs of RFID Protocols by Category

Based on categorization in Chapter 3, the search costs of typical protocols in each category are summarized in Table 4.1. In Category I and Category II, tags respond with EPC code or unchanging MetaID, so the search costs are constant. In Category III, tags change responses by calculating a function $F(k, r)$ based on a secret k and a random number r in the random hash-lock protocol and the UNP protocol, which result in exhaustive searches. In the the big brother protocol and the MWTree protocol, the back-end database is set up with a tree structure of tags' keys and each tag has a series of keys from the tree's root layer till the leaf layer. Thus, the search costs are reduced from O(N) to O(log(N)).The BMM protocol further reduces the search cost to constant by taking pre-computing strategy. In the LD protocol of Category IV, the reader needs an exhaustive search due to the use of a random number in generating a tag's response, while in the lightweight and HM hash-based protocols, a reader only needs a constant search because a tag can be searched based on the tag's MetaID. In Category V and VI, because tags change responses every time, the readers need exhaustive search. From the table, it is obvious that search costs of Category V and VI are no lower than those of other four categories. In Category III and IV, the search cost is not directly related to the security level of

Table 4.1: Search Costs of Typical Protocols in each Category

I	II	III	IV	V	VI
EPC Baseline O(1)	Hash-Lock O(1)	Random Hash-Lock O(N)	LD O(N)	SM O(N)	OSK O(N)
	Key-Sharing O(1)	Big Brother O(log(N))	Lightweight O(1)	Revised SM O(N)	RFIDDOT O(N)
		UNP O(N)	HM Hash-based O(1)		Narrow-Destructive O(N)
		MWTree O(log(N))			
		BMM O(1)			

a protocol, but more related to the data structure of a reader's back-end database. For instance, in Category III, though the five protocols have same security level, their search costs are different from each other. By using tree structures, exhaustive search can be reduced to log-linear O(log(N)). In addition, some pre-computing operations can help further reduce the search cost to constant.

Chapter 5: RFID Security and Tag-Related Cost Analysis

In the previous chapter, we analyzed the performance overhead of RFID security on reader's search cost. Now we analyze performance overhead of RFID security on tag-related cost, including:

1. Cryptographic Operation in Tags

   After simplifying and combining the operations, there are four basic operations in cryptographic operations in tags: tag-tag read, tag-tag write, AES encryption, hash functions.

2. Communication Cost

   Because RFID tags cannot actively launch communication sessions, communication costs are reader-to-tag operations, including reader-tag read and reader-tag write.

Based on the two costs mentioned above, we want to evaluate the performance of protocols in different security levels and benchmark the optimal protocols.

5.1 Basic Operations in RFID Protocols

The existing RFID protocols can be decomposed into the following eleven basic operations:

1. Hash
2. AES
3. XOR

4. Concatenation (||)

5. Left/Right Shift $(<<, >>)$

6. PRF (pseudo random function)

7. HMAC

8. Tag-Tag Read:

   The reads from a tag to itself, noted as $R_{\ell}^{T->T}$, ℓ is the length of read operation.

9. Reader-Tag Read:

   The reads from a reader to a tag, noted as $R_{\ell}^{R->T}$, ℓ is the length of read operation.

10. Tag-Tag Write:

    The writes from a tag to itself, noted as $W_{\ell}^{T->T}$, ℓ is the length of write operation.

11. Reader-Tag Write:

    The writes from a reader to a tag, noted as $W_{\ell}^{R->T}$, ℓ is the length of write operation.

Note that AES algorithm and hash function includes numerous logic calculation such as XOR, left shift $(<<, >>)$. Therefore, compared to AES and hash calculation, the costs of XOR, concatenation (| |), left/right shift $(<<, >>)$ are ignorable. Additionally, in [FDW04, CHT09, KO10], authors pointed out that Pseudo Random Function (PRF) can be achieved by AES or MAC, so PRF can be removed from basic operations. HMAC can be implemented by two hash functions instead, as in equation 5.1 in [GC97].

$$HMAC(k, m) = H((k \bigoplus opad)||H((k \bigoplus ipad)||m)) \tag{5.1}$$

After simplifying and combining the operations, we keep six operations as basic operations: Hash, AES, $R_\ell^{R->T}$, $R_\ell^{T->T}$, $W_\ell^{R->T}$, and $W_\ell^{T->T}$.

5.2 Experimental Settings and Testing Results

To calculate the protocol's performance, the cryptographic operations in tags and the communication cost between reader and tag must be measured. Feldhofer [FDW04] implemented AES on HF tags. In that work, AES module was finished within 1016 clock cycles at 100kHz working frequency. Later on, hash functions like MD5, SHA-1 were added into the experiments [FR06]. Besides symmetric key cryptographic operations, some public key cryptographic functions were also implemented in RFID tags. ECC (Elliptic Curve Cryptosystems) was implemented in HF tags [KP06]. When the field size 113 was chosen, ECC calculation took 14.4ms at 13.56 MHz working frequency, which was close to 2×10^5 clock cycles. The number of clock cycles in this case was nearly 200 times more than that of AES in [FDW04]. If larger field size was chosen, more clock cycles were needed. The experimental results further illustrate that public key solution is not light-weighted and not suitable for RFID security systems. In later performance analysis, we will use the experiment results from [CLL$^+$10] because this work is the most complete one. This work measured not only the symmetric key operations such as AES, MD5, but also the read/write operations in a tag. The experiments were conducted on CAEN A828 Reader with back-end database on an IBM T43 laptop with Windows XP operating system. The RFID tag was simulated by IAIK UHF Demotag, whose optimal working frequency was set on 868MHz. The Demotag was implemented with ISO18000-6C standard by default [CLL$^+$10].

The time cost of Read/Write operations are given in Table 5.1 and 5.2 [CLL$^+$10]. It is obvious that the write operations are more expensive than the read operations. Table 5.3 shows the time cost of three AES operations: AES-128, AES-192 and AES-256. In AES algorithm, the input size, key size and output size are the same.

Table 5.1: Time Costs of Reader-Tag Operations (ms) [CLL+10]

Word	1	2	3	4	5	6	7	8
Read($R^{R->T}$)	46.9	46.9	46.9	47.0	47.1	47.1	47.3	47.1
Write ($W^{R->T}$)	63.3	82.9	103.2	124.1	142.3	162.3	183.5	204.0
Word	9	10	11	12	13	14	15	16
Read($R^{R->T}$)	47.2	47.2	47.2	47.3	47.2	47.2	56.8	57.3
Write ($W^{R->T}$)	222.6	243.3	263.6	283.3	303.2	324.5	344.6	364.1

Table 5.2: Time Costs of Tag-Tag Operations Per 16 bits (ms) [CLL+10]

Read($R^{T->T}$)	0.007
Write ($W^{T->T}$)	16.7

Different from AES operations, the input size of hash operations can be various, but their outputs are of fixed length. Time costs of two hash functions: MD5 and SHA (SHA-160, SHA-256, SHA-512) are given in Table 5.4. The reader is refered to [CLL+10] for detailed calculations.

Table 5.3: Time Costs of AES Operations in Tags (ms) [CLL+10]

	AES-128	AES-192	AES-256
Encryption	2.8	3.3	4.3
Decryption	3.1	3.6	4.8

Table 5.4: Time Costs of Hash Operations in Tags (ms) [CLL+10]

	One Block Input Size(Byte)			Two Block Input size(Byte)		
	6	30	54	56	90	118
MD5	1.8	1.8	1.8	3.1	3.2	3.1
SHA-1	5.0	5.1	5.0	10.1	10.2	10.1
SHA-256	11.7	11.7	11.6	23.1	23.2	23.1
SHA-512	6	56	111	112	176	239
	41.0	41.1	41.1	81.7	81.8	81.8

5.3 Systematic Evaluation of Typical RFID Protocols

In this section, we use the mathematic formula 5.2 to calculate protocols' tag-related time costs. This formula is generic for calculating the tag-related time cost of almost any symmetric key based RFID protocols. As long as the time costs of operations $R_{i'}^{T->T}, R_{i''}^{R->T}, W_{j'}^{T->T}, W_{j''}^{R->T}$, $AES^j, Hash^j$ are given in experiments, the tag-related time costs of any protocol can be calculated accordingly.

$$Total\ cost = \sum_i^j T_{i-j} \cdot X_i^j + \sum_{i'} T_{i'}^{T->T} \cdot XR_{i'}^{T->T} + \sum_{i''} T_{i''}^{R->T} \cdot XR_{i''}^{R->T} + \sum_{j'} T_{j'}^{T->T} \cdot XW_{j'}^{T->T} + \sum_{j''} T_{j''}^{R->T} \cdot XW_{j''}^{R->T}, \tag{5.2}$$

Where,

$i \in$ *Cryptographic operation*

$(AES-128, AES-192, AES-256, MD5, SHA\ functions)$

j : *Input length for each cryptographic operation*

i' : *Read length from tag to tag (bits)*

i'' : *Read length from reader to tag (bits)*

j' : *Write length from tag to tag (bits)*

j'' : *Write length from reader to tag (bits)*

X_i^j : *Time cost of operation i with input j bits in tag*

$R_{i'}^{T->T}$: *Read i' bits from tag to tag*

$R_{i''}^{R->T}$: *Read i'' bits from reader to tag*

$W_{j'}^{T->T}$: *Write j' bits from tag to tag*

$W_{j''}^{R->T}$: *Write j'' bits from reader to tag*

$XR_{i'}^{T->T}$: *Time cost of read i' bits from tag to tag*

$XR_{i''}^{R->T}$: *Time cost of read i'' bits from reader to tag*

$XW_{j'}^{T->T}$: *Time cost of write j' bits from tag to tag*

$XW_{j''}^{R->T}$: *Time cost of write j'' bits from reader to tag*

$T_{i-j}, T_{i'}^{T->T}, T_{i''}^{R->T}, T_{j'}^{T->T}, T_{j''}^{R->T}$:

The counts of cryptographic operation/read/write from reader/tag to tag

In order to calculate the tag-related time costs of protocols, first, we need to decompose protocols into basic operations as shown in Table 5.5. Then, based on the time costs of basic operations, the tag-related time costs of protocols can be calculated according to formula 5.2.

To unify the criteria for the security measurement, the lengths of all the parameters are set as ℓ bits (e.g. $|key| = |ID| = \ell$). Considering the low cost requirements in RFID systems, we set $|\ell|$ = 128 bits by default. MD5 and AES-128 are used as cryptographic operations in later analysis. In addition, as previous discussed, we use two kinds of PRFs. One kind of PRFs is generated by AES-128, called AES-128 based PRFs. The other kind of PRFs is generated by HMAC, called HMAC-based

Table 5.5: Protocol→Basic Operations

Message from Reader to Tag	Write Operation from Reader to Tag
Message from Tag to Reader	Read Operation from Reader to Tag
Security Operation	1) Obtain input parameter from the tag's own memory bank through Tag-Tag Read Operation 2) Perform basic Security Operation such as AES, HASH 3) Write the output into the tag's memory bank through Tag-Tag Write Operation

PRFs. We use two different kinds of PRFs, because their parameter requirements are different. The input size, key size and output size of AES-128 based PRFs are the same while input size and key size of HMAC based PRFs can be different. Their output sizes are both of fixed length, 128 bits. To generate an AES-128 based PRF, a tag first reads two 128-bit pads (ipad, opad) from its memory as an input as well as a key, and it performs an AES-128 encryption to get a 128-bit output as the random number; finally the tag writes the random number back to its memory. Next time, the tag changes 1 bit of ipad/opad to generate another random number. Different from AES-128 based PRFs, HMAC based PRFs are usually used as an authentication message. The input and the key of HMAC will be particularly provided, and a tag performs two MD5 calculations with the input and the key to generate an output, which the tag writes into its memory. The HMAC based PRFs cost more than AES-128 based PRFs because HMAC calculation contains two MD5 functions, which cost at least $2 \times 1.8 = 3.6(ms)$, while the time cost of AES-128 encryption is 2.8 (ms). In later analysis, we also deduct redundant $R^{T->T}$ and $W^{T->T}$ operations to minimize the total tag-related time costs of each protocol. In the next subsections, the tag-related time costs of protocols in each category are calculated and compared.

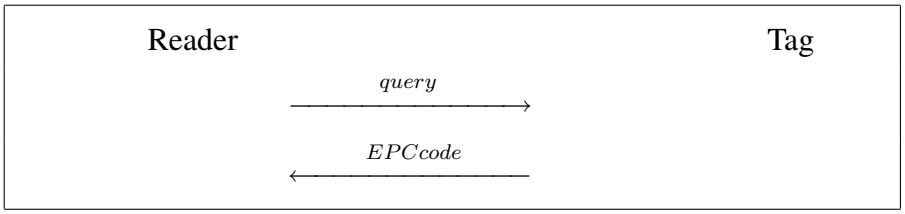


Figure 5.1: EPC Protocol
[Inc08]

Table 5.6: Time Cost of EPC Protocol (ms)

Security Operation	Nil
Tag-Tag Read	Nil
Tag-Tag Write	Nil
Reader-Tag Read	$R_{128}^{R->T}$
Reader-Tag Write	Nil
Search Cost in Reader	O(1)
EPC Cost= $XR_{128}^{R->T}$	
EPC Cost= $47.1ms$	

5.3.1 Category I: EPC Protocol

EPC protocol (Fig. 5.1) contains only one read operations. As claimed before, there are no security protections in a tag except two passwords: kill password and access password. The kill password is to destroy the tag and the access passwords is used to prevent from writing a tag arbitrarily. A reader obtains a tag's EPC ID through a special command "inventory" of EPC standard which is similar to the $R^{R->T}$ operation. Overall, the tag-related time cost of EPC protocol is $R_{128}^{R->T}$ as shown in Table 5.6. As a tag's ID is transmitted in plain text in EPC protocol, the reader can pre-compute and sort the EPC ID of each tag so that the search cost of a tag in a reader's database is constant.

5.3.2 Category II: Tracing Protocol

Two typical cryptographic operations are needed in tracing protocols:

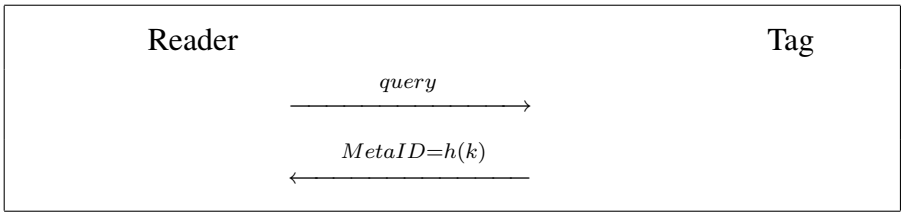


Figure 5.2: Hash-lock Protocol
[WSRE03]

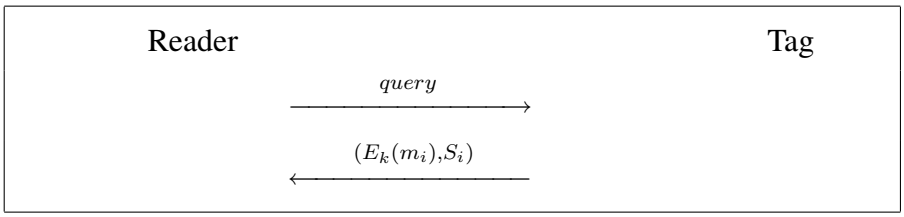


Figure 5.3: Key-Sharing Protocol
[JPP08]

1. Hash lock (Fig. 5.2)

   In the hash lock protocol, a tag responds its MetaID ($h(k)$) instead of EPC code, so the tag has to calculate a hash value. First, a tag reads its secret k from its memory through $R_{128}^{T->T}$ operation. Then the tag performs an MD5 calculation through $MD5^{128}$ operation. Finally, the tag writes the hash value back to its memory through $W_{128}^{T->T}$. A reader reads the MetaID from the tag's memory through $R_{128}^{R->T}$ operation.

2. Key-sharing (Fig.5.3)

   The key-sharing protocol is designed for supply chain application. In this protocol, the first supply chain party writes the encrypted information $E_k(m_i)$ and a key-sharing S_i into tag i's memory. The following supply chain parties need to collect enough key-sharings from a bunch of tags to recover the key k. With the recovered k, the supply chain parties can decrypt the $E_k(m_i)$ to get m_i. In this scenario, the tag does not need to do any cryptographic calculation and there are no write operations in this protocol except the first supply chain party. The following supply chain parties only need to perform an $R^{R->T}$ operation to obtain $(E_k(m_i), S_i)$. In paper [JPP08], the author claimed the

(a)

Security Operation	$MD5^{128}$
Tag-Tag Read	$R_{128}^{T->T}$
Tag-Tag Write	$W_{128}^{T->T}$
Reader-Tag Read	$R_{128}^{R->T}$
Reader-Tag Write	Nil
Search Cost in Reader	O(1)
Hash Lock Cost= $X_{MD5}^{128} + XR_{128}^{T->T} + XW_{128}^{T->T} + XR_{128}^{R->T}$ Hash Lock Cost= $1.8 + 0.007 \times 8 + 16.7 \times 8 + 47.1 = 182.556ms$	

(b)

Security Operation	Nil
Tag-Tag Read	Nil
Tag-Tag Write	Nil
Reader-Tag Read	$R_{144}^{R->T}$
Reader-Tag Write	Nil
Search Cost in Reader	O(1)
Key-Sharing Cost= $XR_{144}^{R->T}$	
Key-Sharing Cost= $47.2ms$	

Table 5.7: Time Costs of Category II Protocols

length of S_i was 16 bits, so only 16+128=144 bits data is transmitted in total.

All of the above two protocols respond a MetaID ($h(ID)/E_k(m_i)$) to a reader. The advantage is that, MetaID can be an index to help a reader find the corresponding tag's information conveniently, and that the readers' search costs of both protocols are constant. However, a MetaID releases a tag's privacy and gives a hint to an attacker for tracing a certain tag. The tag-related time costs of the two protocols are shown in Table 5.7. It is obvious that the key-sharing protocol is more efficient than the hash-lock protocol because key-sharing protocol does not need a tag to do any cryptographic calculations and write operations. Though the hash-lock protocol requires a tag to conduct MD5 calculation and write hash value back to the tag's memory, the performance of the hash-lock protocol can be improved by pre-computing the $h(k)$ values in tags, in which case the tag-related time cost is nearly the same as key-sharing protocol.

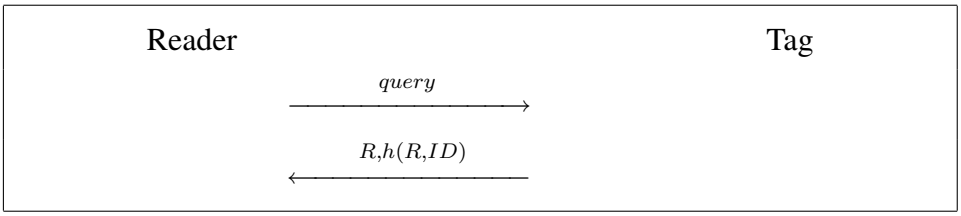


Figure 5.4: Random Hash-Lock Protocol
[WSRE03]

5.3.3 Category III: Strong Anti-Tracing Protocols

Because random numbers are brought into strong anti-tracing RFID protocols, the tracing problem is solved. After being queries by a reader, a tag would reply a message containing a random number. However, without updating tags' internal states, there is still no forward secrecy in strong anti-tracing protocols. There are five typical protocols in this category:

1. Random hash-lock Protocol (Fig 5.4) [WSRE03]:

   In the random hash-lock protocol, a tag generates a random number R by an AES-128 based PRF after a reader queries it. Therefore, there is an AES-128 encryption. Then, the tag calculates a hash value $h(R, ID)$ based on its ID and the random number R through $MD5^{256}$ operation. The reader obtains the random number as well as the hash value through $R_{256}^{R->T}$ operation. However, without a reader's challenge, the random hash-lock protocol is vulnerable to "replay attack", which means a malicious party can fake a real tag by intercepting and replaying the message $(R, h(R, ID))$ to a reader. After the real tag is stolen, the malicious party can still replay the message to deceive the reader and make the reader believe that the real tag is still there.

2. Big brother Protocol (Fig 5.5)) [Dim06]:

   In the big brother protocol, to avoid exhaustive search in a reader, a tree structure of keys is set up in reader's database, so every tag owns a series of keys from the root layer till the leaf layer. As claimed in [MW04], it is assumed that there are total of 2^{20} tags in the RFID system, so the key tree contain 20

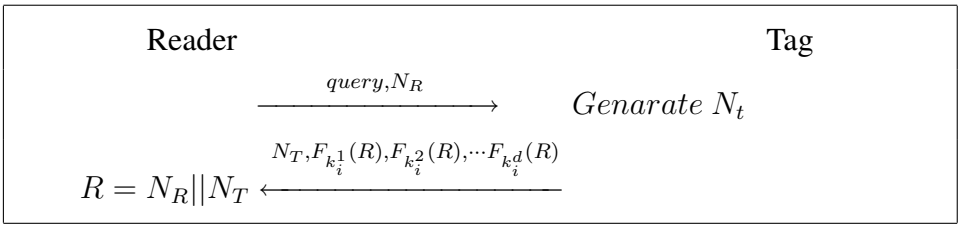


Figure 5.5: Big Brother Protocol
[Dim06]

layers in total and every tag owns 20 keys. In the first round, a reader challenges a tag 'i' with a random number N_R through $W_{128}^{R->T}$ operation. Tag 'i' generates another random number N_T by an AES-128 based PRF operation. Based on the two random numbers N_R, N_T and 20 keys $k_i^1, k_i^2, \cdots, k_i^{20}$, tag 'i' calculates 20 HMAC-based PRF$(F_k(\cdot))$ values through MD5 operations. In the second run, the reader gets the random number N_T as well as 20 PRF values through $R^{R->T}$ operation.

3. UNP protocol (Fig 5.6) [MLDL09]:

   In the UNP protocol, there exists a counter in a tag. Every time a tag is queried, the counter will increase by 1 bit. The time cost of counter operation is ignorable. In the first run, a reader challenges a tag with a random number c. Based on c and the tag's ctr, the tag performs two HMAC-based PRFs $(F_k(\cdot))$ to generate a response to reader. In the HMAC-based PRF calculation, $pad1$ and $pad2$ are two fixed-length pads to guarantee $|ctr||pad1| = 256$ bits and $|ctr||pad2| = 128$ bits. In the second run, the reader gets the concatenation value of two PRFs' outputs through $R^{R->T}$ operation. This $R^{R->T}$ operation will not be mentioned in the following analysis.

4. MWTree Protocol (Fig 5.7) [MW04]:

   Similar to the big brother protocol, there also exists a tree structure in the MWTree protocol. The differences between them is that the big brother is a tag-authentication protocol, while the MWTree protocol is a mutual authentication protocol. Therefore, MWTree protocol needs a tag to calculate

Reader		Tag
$\{I, k, ctr, ID\}$		(k, ctr)
	$\xrightarrow{c \in_R \{0,1\}^{\kappa_1}}$	
		$I = F_k(ctr\|\|pad_1)$
	$\xleftarrow{r = r_1 \|\| I}$	$r_1 = F_k(c\|\|I) \bigoplus (ctr\|\|pad_2)$
$If\ find\ tuple\ (I, k, ctr', ID), then$		$ctr = ctr + 1$
$If\ ctr'\|\|pad_2 = F_k(c\|\|I) \bigoplus r_1, then$		
$Update\ ctr' = ctr' + 1\ and\ I = F_k(ctr'\|\|pad_1), Accept$		
$Else\ Reject$		
$Else\ If \exists (I', k, ctr', ID)$		
$such\ that\ ctr\|\|pad_2 = F_k(c\|\|I) \bigoplus r_1\ and\ F_k(ctr\|\|pad_1) = I, then$		
$Update\ ctr' = ctr + 1\ and\ I' = F_k(ctr'\|\|pad_1)\ and\ Accept$		
$Else\ Reject$		

Figure 5.6: UNP Protocol
[MLDL09]

(a)

Security Operation	$MD5^{256} + AES^{128}$
Tag-Tag Read	$4R_{128}^{T->T}$
Tag-Tag Write	$2W_{128}^{T->T}$
Reader-Tag Read	$R_{256}^{R->T}$
Reader-Tag Write	Nil
Search Cost in Reader	O(N)
Random Hash lock Cost= $X_{MD5}^{256} + X_{AES}^{128} + 4XR_{128}^{T->T} + 2XW_{128}^{T->T} + XR_{256}^{R->T}$	
Random Hash lock Cost= $1.8 + 2.8 + 4 \times 0.007 \times 8 + 2 \times 16.7 \times 8 + 57.3 = 329.324ms$	

(b)

Security Operation	$AES^{128} + 20MD5^{128} + 20MD5^{512}$
Tag-Tag Read	$22R_{128}^{T->T} + R_{256}^{T->T}$
Tag-Tag Write	$21W_{128}^{T->T}$
Reader-Tag Read	$10R_{256}^{R->T} + R_{128}^{R->T}$
Reader-Tag Write	$W_{128}^{R->T}$
Search Cost in Reader	O(log(N))
Big Brother Cost= $X_{AES}^{128} + 20X_{MD5}^{128} + 20X_{MD5}^{512} + 24XR_{128}^{T->T} + 21XW_{128}^{T->T} + 10XR_{256}^{R->T} + XR_{128}^{R->T} + XW_{128}^{R->T}$	
Big Brother Cost= $2.8 + 1.8 \times 20 + 20 \times 3.2 + 24 \times 0.007 \times 8 + 21 \times 16.7 \times 8 + 10 \times 57.3 + 47.1 + 204.0 = 3733.844ms$	

(c)

Security Operation	$2MD5^{512} + 2MD5^{128}$
Tag-Tag Read	$5R_{128}^{T->T}$
Tag-Tag Write	$2W_{128}^{T->T}$
Reader-Tag Read	$R_{256}^{R->T}$
Reader-Tag Write	$W_{128}^{R->T}$
Search Cost in Reader	O(N)
UNP Cost= $2X_{MD5}^{256} + 2X_{MD5}^{512} + 5XR_{128}^{T->T} + 2XW_{128}^{T->T} + XR_{256}^{R->T} + XW_{128}^{R->T}$	
UNP Cost= $1.8 \times 2 + 3.2 \times 2 + 5 \times 0.007 \times 8 + 2 \times 16.7 \times 8 + 57.3 + 204.0 = 538.78ms$	

Table 5.8: Time Cost of Two-Run Category III Protocols

Reader	$s \in \{0,1\}^n$	Tag
$r_1 \in R\{0,1\}^n$	$\xrightarrow{query, r_1}$	
$Find(s, ID) \in Ds.t$	$\xleftarrow{r_2, \sigma = ID \bigoplus f_s(0, r_1, r_2)}$	$r_2 \in_R \{0,1\}^n$,
$ID = \sigma \bigoplus f_s(0, r_1, r_2)$	$\xrightarrow{\tau = ID \bigoplus f_s(1, r_1, r_2)}$	
		$Check\ ID = \tau \bigoplus f_s(0, r_1, r_2)$

Figure 5.7: MWTree Protocol
[MW04]

20 more HMAC-based PRFs $(f_s(\cdot))$ for verifying a reader by comparing 20 PRF's outputs with reader's authentication messages, which are obtained by a tag through $W^{R->T}$ operations in the third run.

5. Improved BMM protocol (Fig 4.7) [LLM+09]:.

   In the improved BMM protocol, there exists a flag $mode$ in a tag. When $mode = 0$, the tag is queried by a legitimate reader; otherwise, if $mode = 1$, the tag is queried by a malicious one. As same as the UNP protocol, there is a counter ctr in a tag. After being queried, firstly, a tag's $mode$ is checked. Based on different values of $mode$, different ps is chosen as parts of inputs to generate HMAC-based PRFs four times. Among the four time PRF calculations, each time the input $ps||c$ increases by 1 bit. The four outputs are assigned to four parameters $(\nu_0, \nu_1, \nu_2, \nu_3)$. In the last run, the state $mode$ is checked and updated. In addition, the index such as r is also updated by $W_{128}^{T->T}$ operation in this run. The lengths of ctr is set as 40 bits [MLDL09], and other parameters $(ps, \nu_0, \nu_1, \nu_2, \nu_3, q, q_i^j, i \in \{old, cur\}, j \in \{1, \ell\}, IV, r)$ are all set as 128 bits. The updating of $ps||c$, updating of ctr and updating of $mode$ are all 1 bit operations, so the time costs of them are ignorable. As the reader party pre-computes and stores the indexes q_i^j, the search cost of the improved BMM protocol is constant. However, when the indexes are used up, the tag is vulnerable to "entrapment attack".

(a)

Security Operation	$AES^{128} + 40MD5^{128} + 40MD5^{513}$
Tag-Tag Read	$68R_{128}^{T->T}$
Tag-Tag Write	$21W_{128}^{T->T}$
Reader-Tag Read	$10R_{256}^{R->T} + R_{128}^{R->T}$
Reader-Tag Write	$10W_{256}^{R->T} + W_{128}^{R->T}$
Search Cost in Reader	O(log(N))
MWTree Cost= $X_{AES}^{128} + 40X_{MD5}^{128} + 40X_{MD5}^{513} + 68XR_{128}^{T->T} + 21XW_{128}^{T->T} + 10XR_{256}^{R->T} + XR_{128}^{R->T} + 10XW_{256}^{R->T} + XW_{128}^{R->T}$ MWTree Cost= $2.8 + (1.8 + 3.2) \times 40 + 68 \times 0.007 \times 8 + 21 \times 16.7 \times 8 + 10 \times 57.3 + 47.1 + 10 \times 364.1 + 204.0 = 7477.308ms$	

(b)

Security Operation	$5MD5^{128} + MD5^{552} + 4MD5^{512}$
Tag-Tag Read	$8R_{128}^{T->T} + R_{40}^{T->T}$
Tag-Tag Write	$8W_{128}^{T->T}$
Reader-Tag Read	$R_{256}^{R->T}$
Reader-Tag Write	$2W_{128}^{R->T}$
Search Cost in Reader	O(1)
Improved BMM Maximum Cost= $5X_{MD5}^{128} + X_{MD5}^{552} + 4X_{MD5}^{512} + 8XR_{128}^{T->T} + + XR_{40}^{T->T} + 8XW_{128}^{T->T} + XR_{256}^{R->T} + 2XW_{128}^{R->T}$	
Improved BMM Maximum Cost= $5 \times 1.8 + 3.2 \times 5 + 8 \times 0.007 \times 8 + 0.007 \times 5 + 8 \times 16.7 \times 8 + 57.3 + 2 \times 204.0 = 1559.583ms$	

Table 5.9: Time Costs of Three-Run Category III Protocols

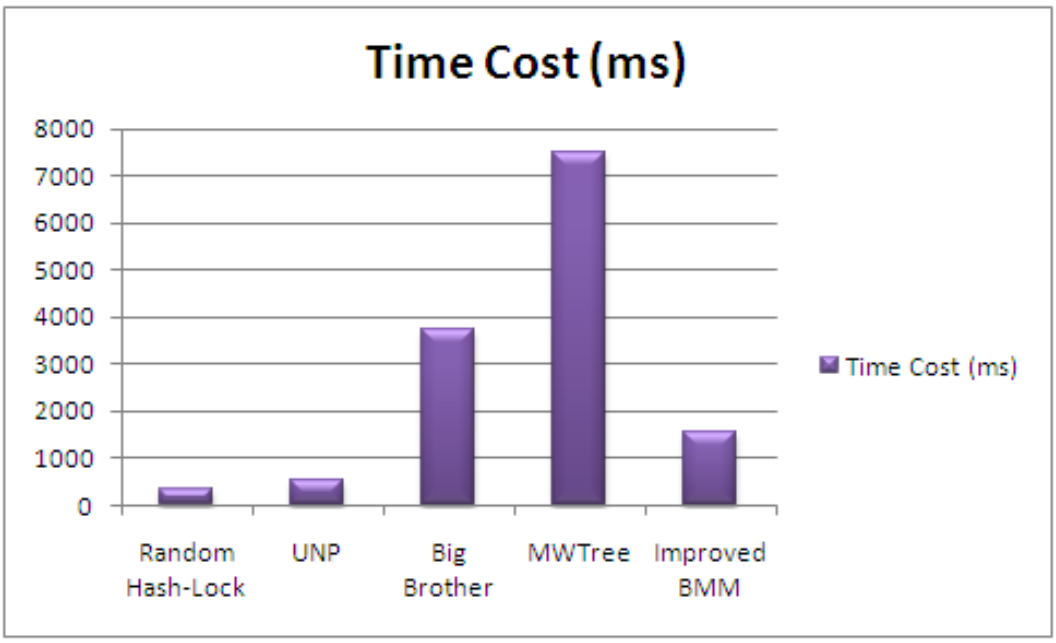


Figure 5.8: Time Cost Comparison in Category III

The tag-related time costs of two-run protocols are shown in Table 5.8 and the tag-related time costs of three-run protocols are shown in Table 5.9. We also compare them directly in Fig 5.8. It is obvious that the random hash-lock protocol, UNP protocol and improved BMM protocol's tag-related time costs are less than those of two tree structure protocols. From the tables, the random hash-lock protocol is most efficient, but it is vulnerable to replay attack due to the lack of a random number from a reader. The UNP protocol costs more than random hash-lock, because a reader challenges a tag by writing a random number in it. The third most efficient protocol is the improved BMM protocol. Although the improved BMM is a three-run protocol, its tag-related time cost is less than half that of the big brother protocol, because of the tree structure in the big brother protocol. On one hand, a tag has to generate and reply 20 HMAC-based PRF values in the big brother protocol, which increases the tag-related time cost a lot. On the other hand, if we do not consider the pre-computing strategy, extra tree structure decreases the search cost from O(N) to O(log(N)). Thus, it is a trade-off between tag-related time costs in tags and search costs in readers. The MWTree is the least efficient protocol, because not only it has a tree structure, but also it is a three-run mutual authentication protocol.

5.3.4 Category IV: Weak Anti-Tracing, Weak Forward Secrecy Protocols

Forward secrecy exists in next three categories to various extents. Category IV protocols can only achieve weak anti-tracing and weak forward secrecy. In these protocols, only after being queried by a legitimate reader, a tag will respond changing, unlinkable values and only after a reader is authenticated successfully, a tag will update its internal state. Malicious readers can repeat querying a certain tag with the same challenge to trace it. If a tag is cracked and a malicious party obtains its internal state between two honest parties A and B, all the tag's internal

Reader		Tag
(ID, k_i, k_{i+1})		$(\alpha = k_i \bigoplus ID)$
	$\xrightarrow{query,r}$	
	$\xleftarrow{t=H(r \bigoplus \alpha)}$	$t = H(r \bigoplus \alpha)$
$Check \exists t = H(r \bigoplus ID \bigoplus k_i)$		
$IF so, Update a = k_i \bigoplus k_{i+1},$		
$b = H(k_{i+1} \bigoplus r \bigoplus ID)$		
$Otherwise, \perp$		
	$\xrightarrow{a=k_i \bigoplus k_{i+1}, b=H(k_{i+1} \bigoplus r \bigoplus ID)}$	
		$Check\ if\ b = H(k_{i+1} \bigoplus r \bigoplus ID)$
		$If\ so, update\ \alpha = k_{i+1} \bigoplus ID$
		$Otherwise, reject$

Figure 5.9: LD Protocol
[LD07]

states from spot A till cracked spot can be totally retrieved. The LD protocol shown in Fig 5.9 [LD07], the update key-sharing shown in Fig 5.10 [CLM+09] and the lightweight protocol shown in Fig 5.11 [Dim05] belong to category IV. Note that in the LD protocol, the tag's updated content is written by a legitimate reader, while in update key-sharing protocol, the tag's content is updated by hash function. In the lightweight protocol, the author used ID as the secret key to generate HMAC-based PRF values as authentication messages in the second run and the third run. After the reader is authenticated successfully in the third run, we assume that the new ID is updated by an HMAC ($ID_{i+1} = h_{ID_i}(N_R, N_T)$) operation. HM hash-based protocol shown in Fig. 5.12 has some redundant information which can be deleted from the protocol to save tag-related time cost. For example, the $DB - ID$ in the second round is to help a reader find the back-end database which stores the corresponding tag's information. This information is better stored in the reader' side. In addition, the $\Delta TID = TID - LST$ (TID is the transaction ID and LST is the last

Reader		Tag
k, k', S_i		$S_i, m_i, k, c_i = h(k\|\|S_i)$
	$\xrightarrow{query}$	
$c_i = h(k\|\|S_i),$	$\xleftarrow{(E_k(m_i),S_i)}$	
$c'_i = h(k'\|\|S'_i)$		
$A = (S'_i\|\|m'_i) \bigoplus h('0'\|\|c_i)$		
$B = c'_i \bigoplus h('1'\|\|c_i)$		
$C = h(c_i\|\|S'_i\|\|m'_i\|\|c'_i)$	$\xrightarrow{A,B,C}$	
		$S'_i\|\|m'_i = A \bigoplus h('0'\|\|c_i)$
		$c'_i = B \bigoplus h('1'\|\|c_i)$
		$If\ C = h(c_i\|\|S'_i\|\|m'_i\|\|c'_i)$
		$Update\ S_i \leftarrow S'_i$
		$m_i \leftarrow m'_i$
		$c_i \leftarrow c'_i$

Figure 5.10: Update Key-Sharing RFID Protocol
[CLM+09]

Reader		Tag
	$\xrightarrow{query,N_R}$	
	$\xleftarrow{h(ID_i),N_T,h_{ID_i}(N_T,N_R)}$	
$ID_{i+1} = h_{ID_i}(N_R, N_T)$	$\xrightarrow{h_{ID_{i+1}}(N_T,N_R)}$	$ID_{i+1} = h_{ID_i}(N_R, N_T)$
	If verify successfully, Update$ID_i = ID_{i+1}$	

Figure 5.11: Lightweight RFID Protocol
[Dim05]

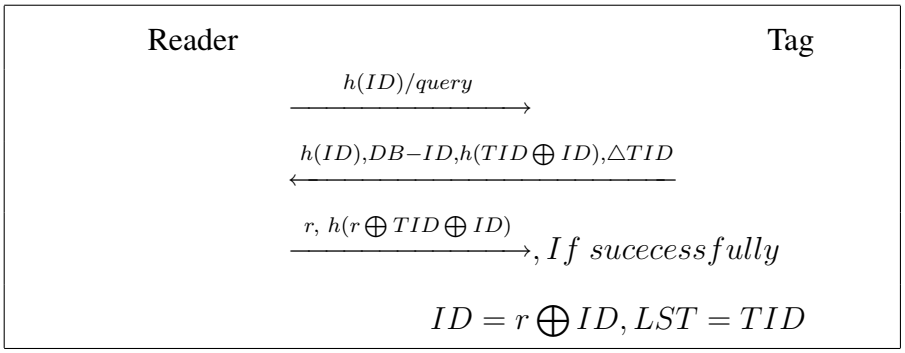


Figure 5.12: HM Hash-Based Protocol
[HM04]

successful traction ID), which is transferred in plain text and can be easily modified by an attacker, is also unnecessary. The time costs of the four protocols are given in Table 5.10.

5.3.5 Category V: Strong Anti-Tracing, Weak Forward Secrecy Protocols

Compared to category IV protocols, the tag automatically changes its response every time in category V. Therefore, strong anti-tracing is achieved. Nevertheless, forward secrecy is still weak because only after authenticating a reader successfully, a tag will update internal states. The revised SM protocol shown in Fig. 5.13 is a typical category V protocol.

The original SM protocol in [SM08] suffers from server impersonation attack, tag impersonation attack and de-synchronization attack [CLLD09b] due to the insecure XOR operations. The revised SM protocol replaced XOR operations by concatenations to eliminate these security flaws. A tag generates a random number r_2 by an AES-128 based PRF after a reader interrogates it, so every response from the tag is different from each other and unlinkable. Thus, strong anti-tracing property is carried out. In the third round, only after reader-authentication is successful, the tag updates its internal state by hash function calculation. In summary, the revised SM protocol is strong anti-tracing and weak forward secrecy.

(a)

Security Operation	$2MD5^{128}$
Tag-Tag Read	$5R_{128}^{T->T}$
Tag-Tag Write	$2W_{128}^{T->T}$
Reader-Tag Read	$R_{128}^{R->T}$
Reader-Tag Write	$W_{256}^{R->T} + W_{128}^{R->T}$
Search Cost in Reader	O(N)
LD Cost= $2X_{MD5}^{128} + 5XR_{128}^{T->T} + 2XW_{128}^{T->T} + XR_{128}^{R->T} + XW_{256}^{R->T} + XW_{128}^{R->T}$	
LD Cost= $2 \times 1.8 + 5 \times 0.007 \times 8 + 2 \times 16.7 \times 8 + 47.1 + 364.1 + 204.0 = 886.28ms$	

(b)

Security Operation	$AES^{128} + 2MD5^{129} + MD5^{384}$
Tag-Tag Read	$6R_{128}^{T->T}$
Tag-Tag Write	$3W_{128}^{T->T}$
Reader-Tag Read	$R_{144}^{R->T}$
Reader-Tag Write	$W_{128}^{R->T} + W_{256}^{R->T}$
Search Cost in Reader	O(1)
Update Key-sharing Cost= $X_{AES}^{128} + 3X_{MD5}^{384} + 6XR_{128}^{T->T} + 3XW_{128}^{T->T} + XR_{144}^{R->T} + XW_{128}^{R->T} + XW_{256}^{R->T}$	
Update Key-sharing Cost= $2.8 + 1.8 \times 3 + 6 \times 0.007 \times 8 + 3 \times 16.7 \times 8 + 47.2 + 204.0 + 364.1 = 1024.636ms$	

(c)

Security Operation	$3MD5^{512} + 4MD5^{128} + AES^{128}$
Tag-Tag Read	$9R_{128}^{T->T}$
Tag-Tag Write	$4W_{128}^{T->T}$
Reader-Tag Read	$R_{128}^{R->T} + R_{256}^{R->T}$
Reader-Tag Write	$2W_{128}^{R->T}$
Search Cost in Reader	O(1)
Lightweight Cost= $4X_{MD5}^{128} + 3X_{MD5}^{512} + X_{AES}^{128} + 9XR_{128}^{T->T} + 4XW_{128}^{T->T} + XR_{256}^{R->T} + XR_{128}^{R->T} + 2XW_{128}^{R->T}$	
Lightweight Cost= $1.8 \times 4 + 3 \times 3.2 + 2.8 + 9 \times 0.007 \times 8 + 4 \times 16.7 \times 8 + 57.3 + 47.1 + 2 \times 204.0 = 1066.904ms$	

(d)

Security Operation	$3MD5^{128}$
Tag-Tag Read	$6R_{128}^{T->T}$
Tag-Tag Write	$4W_{128}^{T->T}$
Reader-Tag Read	$R_{256}^{R->T}$
Reader-Tag Write	$W_{256}^{R->T}$
Search Cost in Reader	O(1)
HM Cost= $3X_{MD5}^{128} + 6XR_{128}^{T->T} + 4XW_{128}^{T->T} + XR_{256}^{R->T} + XW_{256}^{R->T}$	
HM Cost= $3 \times 1.8 + 6 \times 0.007 \times 8 + 4 \times 16.7 \times 8 + 57.3 + 364.1 = 961.536ms$	

Table 5.10: Time Cost of Category IV Protocols

Reader		Tag
$(k_i, t_i)_{new}, (k_i, t_i)_{old}, D_i$		$[t_i]$
$r_1 \in_R \{0,1\}^l$	$\xrightarrow{r_1}$	$r_2 \in_R \{0,1\}^l$
		$M_1 = t_i \bigoplus r_2$
		$M_2 = f_{t_i}(r_1 \| r_2)$
Check whether$\exists t_i, s.t.$	$\xleftarrow{M_1, M_2,}$	
$r_2 \leftarrow M_1 \bigoplus t_i$ *and* $M_2 = f_{t_i}(r_1 \| r_2)$		
$M_3 = h(r_2) \bigoplus k_i$	$\xrightarrow{M_3}$	
		$k_i \leftarrow M_3 \bigoplus h(r_2)$
$k_{i(old)} \leftarrow k_i$		Check if $h(k_i) = t_i$
$t_{i(old)} \leftarrow t_i$		$t_i \leftarrow h((k_i \ll l/4) \bigoplus (t_i \gg l/4) \bigoplus r_1 \bigoplus r_2)$
$k_{i(new)} \leftarrow (k_i \ll l/4) \bigoplus (t_i \gg l/4) \bigoplus r_1 \bigoplus r_2$		
$t_{i(new)} \leftarrow h(k_{i(new)})$		

Figure 5.13: Revised SM Protocol
[SM08] [CLLD09b]

From Table 5.11, we can see the tag-related time cost of the revised SM protocol is more than 1 second, because a tag needs too many $W^{T->T}$ operations.

5.3.6 Category VI: Strong Anti-Tracing, Strong Forward Secrecy Protocols

Tags implemented with strong anti-tracing and strong forward secrecy protocols can automatically update responses and internal states every time being queried. This category is the strongest one in our categorization. Three typical protocols introduced here are OSK protocol [OSK03], RFIDDOT protocol [Dim08] and Narrow-destructive protocol [PV08].

1. OSK Protocol

Security Operation	$AES^{128} + 4MD5^{128} + MD5^{512}$
Tag-Tag Read	$9R_{128}^{T->T}$
Tag-Tag Write	$4W_{128}^{T->T}$
Reader-Tag Read	$R_{256}^{R->T}$
Reader-Tag Write	$2W_{128}^{R->T}$
Search Cost in Reader	O(N)
Revised SM Cost= $X_{AES}^{128} + 4X_{MD5}^{128} + X_{MD5}^{512} + 9XR_{128}^{T->T} + 4XW_{128}^{T->T} + XR_{256}^{R->T} + 2XW_{128}^{R->T}$	
Revised SM Cost= $2.8 + 4 \times 1.8 + 3.2 + 9 \times 0.007 \times 8 + 4 \times 16.7 \times 8 + 57.3 + 2 \times 204.0 = 1013.404ms$	

Table 5.11: Time Cost of Category V Protocols

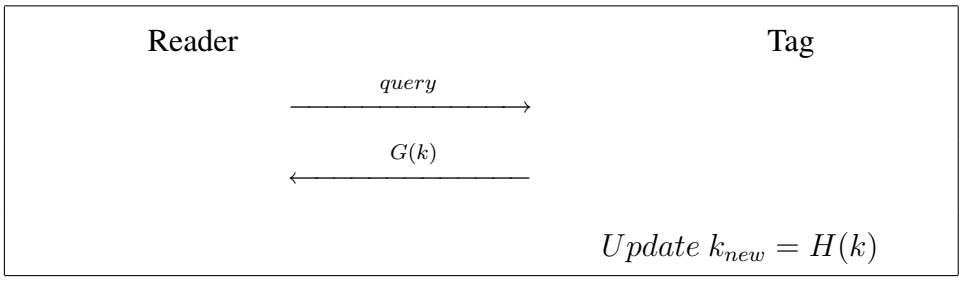


Figure 5.14: OSK Protocol
[OSK03]

The OSK protocol [OSK03] shown in Fig. 5.14 is a tag-authentication only protocol. Every time a tag is queried, it performs two hash functions based on its secret k. One value is sent back to a reader as a reply and the other one is used to update its secret key. Thus, the OSK protocol needs a tag to calculate two hash functions. However, without a reader's challenge, replay attack can be successful in the OSK protocol. In addition, the OSK protocol is vulnerable to Denial of Service (DoS) attack, which means a malicious reader can keep querying a certain tag and push the tag updating its secret constantly. As a result, the malicious party de-synchronize the legitimate reader and the tag. When a legitimate reader communicates with a tag, it cannot authenticate the tag successfully or it needs to do an exhausive search and compute many hash functions to find the corresponding tag because of the de-synchronization.

2. RFIDDOT Protocol

Reader		Tag
	$\xrightarrow{query, N_R}$	
	$\xleftarrow{N_T, F_k(N_T, N_R)}$	
	$\xrightarrow{R \bigoplus F_k(N_R, N_T), F_k(R, N_R, N_T)}$	
$K_{new} = F_k(N_R, R, N_T, 1)$		$K_{new} = F_k(N_R, R, N_T, 1)$

Figure 5.15: RFIDDOT Protocol
[Dim08]

Reader	$(S = K)$	Tag
	$\xrightarrow{query, a}$	
	$\xleftarrow{c = F_S(0,a)}$	$c = F_S(0, a), d' = F_S(1, a)$
		$S \leftarrow G_S(S)$
$Find\ (ID, K)\ and\ i$	$\xrightarrow{d = F_K(1,a)}$	$Check\ if\ d = d'$
$s.t\ c = F_K(0, G^i(K), a)\ and\ i < t$		$Output\ OK, iff\ check$
$K \leftarrow G^i(K), Otherwise\ set\ k\ to\ random$		
$d = F_K(1, a), Output\ ID\ or \perp\ if\ not\ found$		

Figure 5.16: Narrow-Destructive Privacy Protocol
[PV08]

Different from the OSK protocol, the RFIDDOT protocol shown in Fig 5.15 is a mutual authentication protocol. More importantly, it prevents replay attack with the reader's challenge N_R. Unfortunately, as same as the OSK protocol, it is vulnerable to DoS attack too.

3. Narrow-destructive Protocol

   The narrow-destructive protocol shown in Fig 5.16 is more efficient than the RFIDDOT protocol, because a tag does not need to generate its own random number through AES-128 based PRF, and in the third run the tag only sends 128 bit HMAC based PRF value back to a reader instead of 256 bit message in the RFIDDOT protocol.

(a)

Security Operation	$2MD5^{128}$
Tag-Tag Read	$R_{128}^{T->T}$
Tag-Tag Write	$2W_{128}^{T->T}$
Reader-Tag Read	$R_{128}^{R->T}$
Reader-Tag Write	Nil
Search Cost in Reader	O(N)
OSK Cost= $2X_{MD5}^{128} + XR_{128}^{T->T} + 2XW_{128}^{T->T} + XR_{128}^{R->T}$ OSK Cost= $2 \times 1.8 + 0.007 \times 8 + 2 \times 16.7 \times 8 + 47.1 = 317.956ms$	

(b)

Security Operation	$4MD5^{128} + 2MD5^{640} + 2MD5^{512} + AES^{128}$
Tag-Tag Read	$10R_{128}^{T->T}$
Tag-Tag Write	$3W_{128}^{T->T}$
Reader-Tag Read	$R_{256}^{R->T}$
Reader-Tag Write	$W_{128}^{R->T} + W_{256}^{R->T}$
Search Cost in Reader	O(N)
RFIDDOT Cost= $4X_{MD5}^{128} + 2X_{MD5}^{640} + 2X_{MD5}^{512} + X_{AES}^{128} + 10XR_{128}^{T->T} + 3XW_{128}^{T->T} + XR_{256}^{R->T} + XW_{128}^{R->T} + XW_{256}^{R->T}$	
RFIDDOT Cost= $4 \times 1.8 + 4 \times 3.2 + 2.8 + 10 \times 0.007 \times 8 + 3 \times 16.7 \times 8 + 57.3 + 204.0 + 364.1 = 1049.56ms$	

(c)

Security Operation	$3MD5^{128} + 2MD5^{385} + MD5^{384}$
Tag-Tag Read	$4R_{128}^{T->T}$
Tag-Tag Write	$3W_{128}^{T->T}$
Reader-Tag Read	$R_{128}^{R->T}$
Reader-Tag Write	$2W_{128}^{R->T}$
Search Cost in Reader	O(N)
Narrow-Destructive Cost= $3X_{MD5}^{128} + 2X_{MD5}^{385} + X_{MD5}^{384} + 4XR_{128}^{T->T} + 3XW_{128}^{T->T} + XR_{128}^{R->T} + 2XW_{128}^{R->T}$	
Narrow-Destructive Cost= $6 \times 1.8 + 4 \times 0.007 \times 8 + 3 \times 16.7 \times 8 + 47.1 + 2 \times 204.0 = 866.924ms$	

Table 5.12: Time Cost of Category VI Protocols

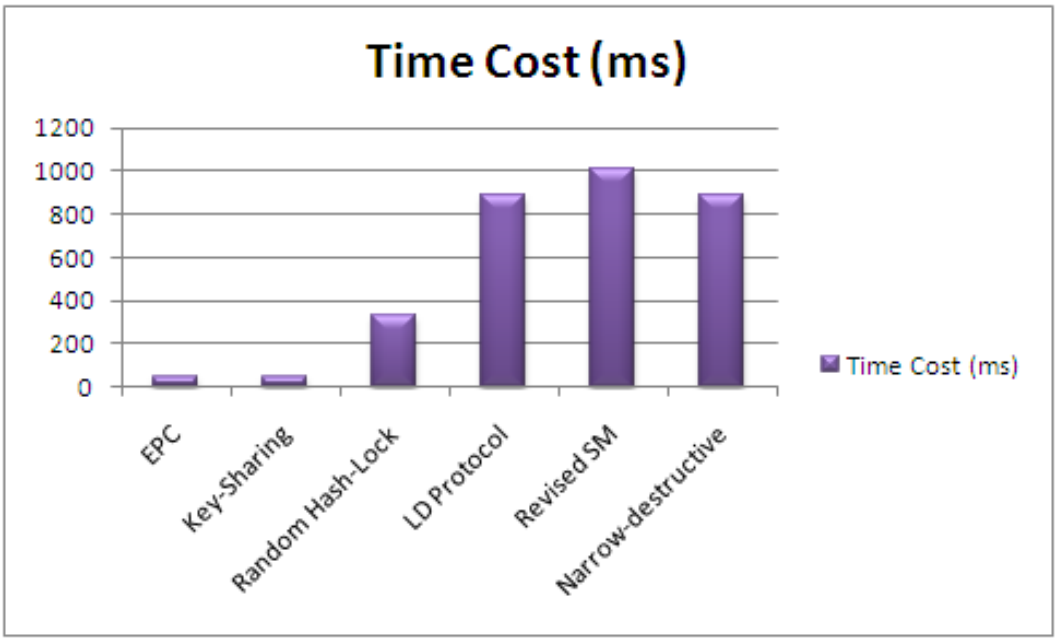


Figure 5.17: Time Cost Comparison of Best-Performance Protocols in Each Category

The tag-related time cost of the OSK, the RFIDDOT and the narrow-destructive protocols are given in Table 5.12. The RFIDDOT's tag-related time cost is more than three times that of the OSK protocol and the narrow-destructive protocol's tag-related time cost is twice more than that of the OSK protocol because they each contain a reader's challenge and add a reader authentication process. Though the OSK protocol is the most efficient one, the OSK protocol is vulnerable to replay attack without reader's challenge. Note that though three protocols achieve strong anti-tracing and strong forward secrecy, all of them are vulnerable to DoS attacks.

5.3.7 Comparison of RFID Protocols' Performance

After analyzing tag-related time cost of every protocol in each category, we compare them by category. The protocols with best performance in each category are selected as benchmarks. Therefore, six protocols represent their own category and their time efficiency is compared in Fig. 5.17. Note that the random hash-lock protocol of category III is vulnerable to replay attack, so we use the UNP protocol instead. The new comparison figure is given in Fig 5.18. In the first three categories,

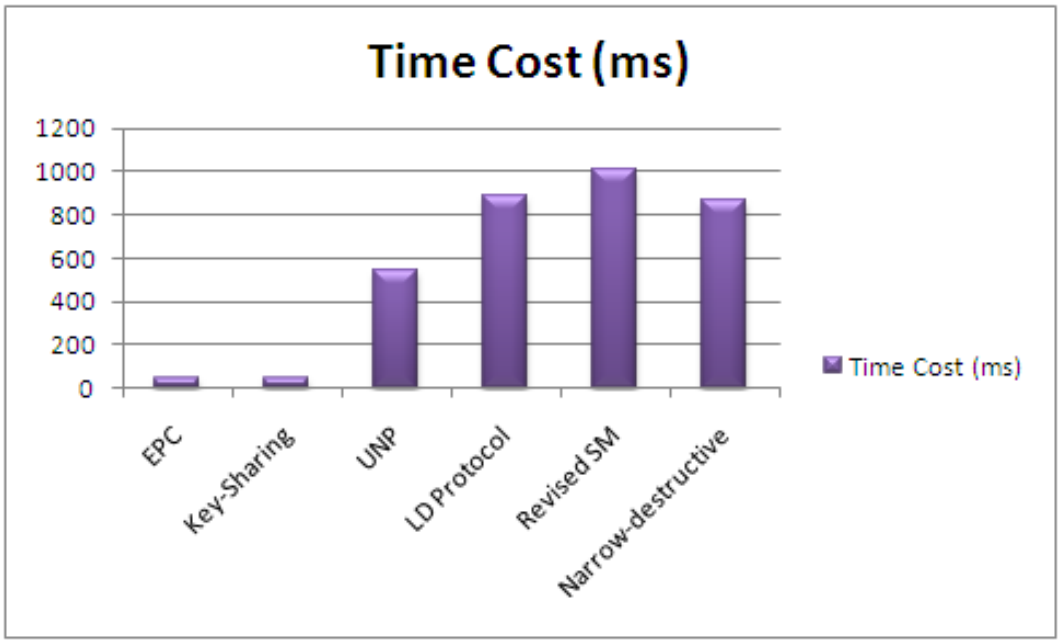


Figure 5.18: Time Cost Comparison of Best-Performance Protocols without Replay Attacks

two run protocols (EPC protocol, the key sharing protocol and the UNP protocol) can achieve their security properties, respectively. Category VI, V protocols need reader-authentication, so they need three runs. To fairly compare the performance with category IV, V, we use the three-run mutual authentication protocol in category VI. The tag-related time costs of the protocols in the first two categories are nearly the same and both protocols are efficient because there is only one $R^{R->T}$ operation being used. The main reason for the big gap of time costs between the UNP protocol in category III and the first two protocols is that the UNP protocol needs high-cost write operations, including $W^{R->T}$ and $W^{T->T}$. The protocols in the first three categories only have two runs, while there are three runs in last three categories. This explains the extra time cost of the LD protocol in category IV, compared with the UNP protocol in category III. It is reasonable that as security properties gets stronger and the number of communication runs gets larger, the tag-related time cost of a protocol increases.[1] However, it is noted that although

[1]The strict sequence of security property from weak to strong is Category I, II, III, V, VI or Category I, II, IV, V, VI. The security level of category III protocols is parallel with that of category IV protocols. Because category III protocols are stronger in anti-tracing aspect than category IV ones, while weaker in forward secrecy aspect. We just put them into the Fig 5.17 to show the trend.

Reader		Tag
(ID, k_i)		(k_i)
	$\xrightarrow{query,\ r}$	
	$\xleftarrow{t=H(r \bigoplus k_i)}$	$t = H(r \bigoplus k_i)$
$Check \exists t = H(r \bigoplus k_i)$		
$IF so, Update\ k_{i+1} = h_{k_i}(r),$		
$b = H(k_{i+1} \bigoplus r)$		
$Otherwise, \perp$		
	$\xrightarrow{b=H(k_{i+1} \bigoplus r)}$	
		$Check\ if\ b = H(k_{i+1} \bigoplus r)$
		$If\ so, update\ k_{i+1} = h_{k_i}(r)$
		$Otherwise, reject$

Figure 5.19: Revised LD Protocol
[LD07]

narrow-destructive protocol belongs to the strongest category, it costs less than the revised SM protocol in category V and its cost is nearly the same as the LD protocol in category IV. This discovery surprises us. The reason of this phenomenon is that the LD protocol and the revised SM protocol have a couple of redundant operations. In the next subsection, these two protocols are analyzed and refined. The performance of them are re-evaluated.

5.3.8 Revising Protocol and Re-evaluating Protocols' Performance

The main reason of the high tag-related time cost of the LD protocol is that a tag's internal state is updated by $W^{R->T}$ operations in the third run, which is neither secure nor efficient. On one hand, the XOR value of old key k_i and new key k_{i+1} is transferred in plain text. Once an adversary corrupts a tag and obtains k_{i+1}, he can recover the previous keys $k_i, k_{i-1}, \cdots$ with the previous intercepted messages $k_i \bigoplus k_{i+1}, k_{i-1} \bigoplus k_{i+1}, \cdots$. Thus, it is not secure to update a tag's internal state by

Table 5.13: Revised LD Protocol Performance

Security Operation	$3MD5^{128} + MD5^{384}$
Tag-Tag Read	$5R_{128}^{T->T}$
Tag-Tag Write	$2W_{128}^{T->T}$
Read-Tag Read	$R_{128}^{R->T}$
Read-Tag Write	$2W_{128}^{R->T}$
Search Cost in Reader	O(N)
Revised LD Cost= $3X_{MD5}^{128} + X_{MD5}^{384} + 5XR_{128}^{T->T} + 2XW_{128}^{T->T} + XR_{128}^{R->T} + 2XW_{128}^{R->T}$ Revised LD Cost= $3 \times 1.8 + 1.8 + 5 \times 0.007 \times 8 + 2 \times 16.7 \times 8 + 47.1 + 2 \times 204.0 = 729.78ms$	

Reader	$(S = K)$	Tag
	$\xrightarrow{query,a}$	
$Find\ (ID, K)\ and$	$\xleftarrow{b,c=F_S(a,b)}$	$b, c = F_S(a, b)$
$i.s.t\ c = F_K(a, b)$		
$d = F_K(b, a)$	$\xrightarrow{d=F_K(b,a)}$	$Check\ if\ d = d' = F_S(b, a)$
		$Update\ S \leftarrow G_S(a, b, 1), iff\ check$
$K_{new} \leftarrow G_K(a, b, 1),$		
$Store\ old\ k\ prevent\ de - synchronized\ attacks$		
$Output\ ID\ or \perp if\ not\ found$		

Figure 5.20: Category V Protocol
[PV08]

writing keys' XOR values into a tag. On the other hand, the write operation $W^{R->T}$ is quite costly and the update process in the original LD protocol is finished by this high-cost operation. The write operation can be replaced by tag automatic update with an HMAC calculation $k_{i+1} = h_{k_i}(r)$. The revised protocol is shown in Fig 5.19 and the performance is given in Table 5.13.

The other protocol that can be improved is the revised SM protocol in category V. The revised SM protocol is not efficient because it has redundant $W^{T->T}$ operation. We design a new protocol shown in Fig 5.20 to improve it. This protocol is based on Vaudenay's narrow-destructive model, and the difference between them is

Table 5.14: Category V Protocol Performance

Security Operation	$3MD5^{128} + 2MD5^{512} + MD5^{513} + AES^{128}$
Tag-Tag Read	$9R_{128}^{T->T}$
Tag-Tag Write	$3W_{128}^{T->T}$
Read-Tag Read	$R_{256}^{R->T}$
Read-Tag Write	$2W_{128}^{R->T}$
Search Cost in Reader	O(N)
Category V Cost= $3X_{MD5}^{128} + 2X_{MD5}^{512} + X_{MD5}^{513} + X_{AES}^{128} + 9XR_{128}^{T->T} + 3XW_{128}^{T->T} + XR_{256}^{R->T} + 2XW_{128}^{R->T}$ Category V Cost= $3 \times 1.8 + 3 \times 3.2 + 2.8 + 9 \times 0.007 \times 8 + 3 \times 16.7 \times 8 + 57.3 + 2 \times 204.0 = 884.404ms$	

that a tag in this protocol does not automatically update its internal state. Only after a tag authenticates a reader successfully, the tag updates its internal state. Therefore, this protocol can prevent a malicious reader from illegally de-synchronizing a legitimate reader and a tag. In addition, strong anti-tracing is achieved in this protocol by a tag generating a random number. The performance analysis of this protocol is given in Table 5.14 and its tag-related time cost is nearly as same as the narrow-destructive protocol's.

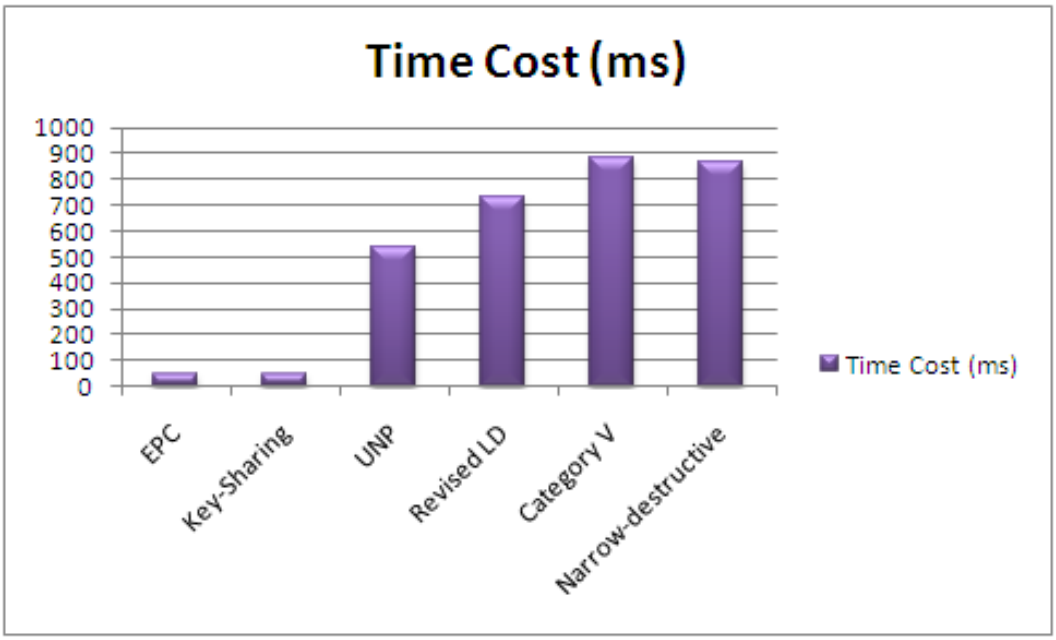


Figure 5.21: Revising of Time Cost Comparison of Best-Performance Protocols

After we use two refined protocols instead of the LD protocol and the revised SM protocol in category IV and V, the revision of tag-related time cost comparison

in each category is shown in Fig 5.21. After the adjustment, it is obvious that tag-related time cost is positively correlated with security properties and the number of communication runs, which means as security becomes stronger and the number of communication runs become larger, the tag-related time cost of corresponding protocol is more. As claimed in [BBEG09], it is impossible to achieve strong forward secrecy and prevent infinite DoS attacks at the same time. Therefore, only symmetric key cryptography cannot eliminate DoS attack in the narrow-destructive and the RFIDDOT in category VI. Public key cryptography (PKC) [PV08, Su10], is needed to get rid of DoS, which costs more than symmetric key cryptography. The ideal model of tag-related time cost comparison is shown in Fig 5.22.

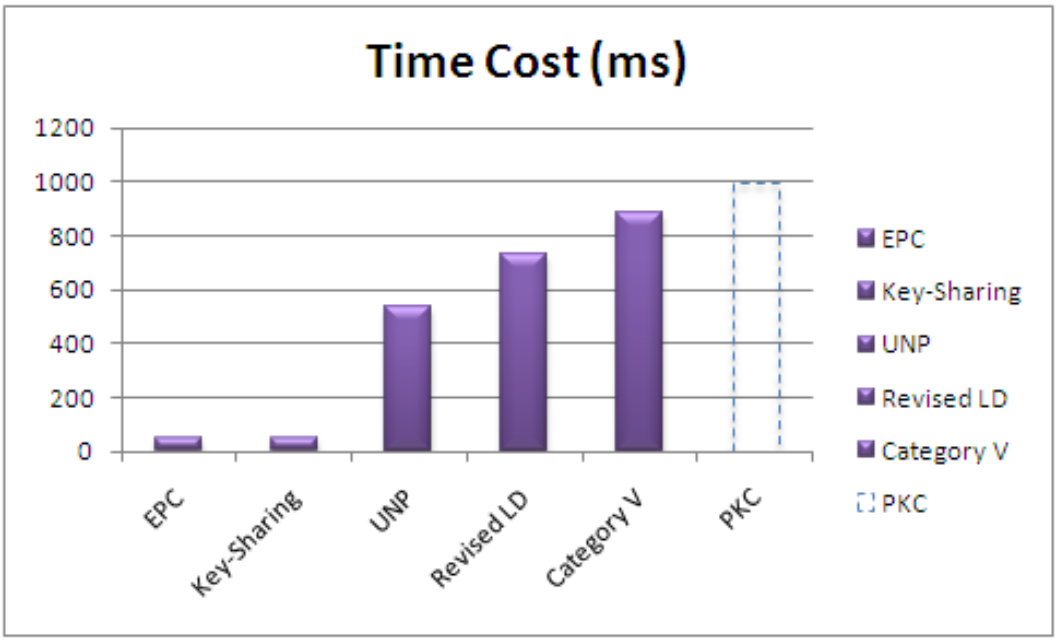


Figure 5.22: Ideal Model of Time Cost Comparison of Best-Performance Protocols

5.3.9 Suggestions for Protocol Designs

RFID tags have only limited memory and computational ability, so the tag-related time cost may be the bottleneck of the whole process. It is important to reduce the high-cost write operations as few as possible. In Fig. 5.23, the percentages of write operations in typical protocols are shown. Except the first two protocols, the time cost of write operations, including reader-tag write and tag-tag write, takes up more

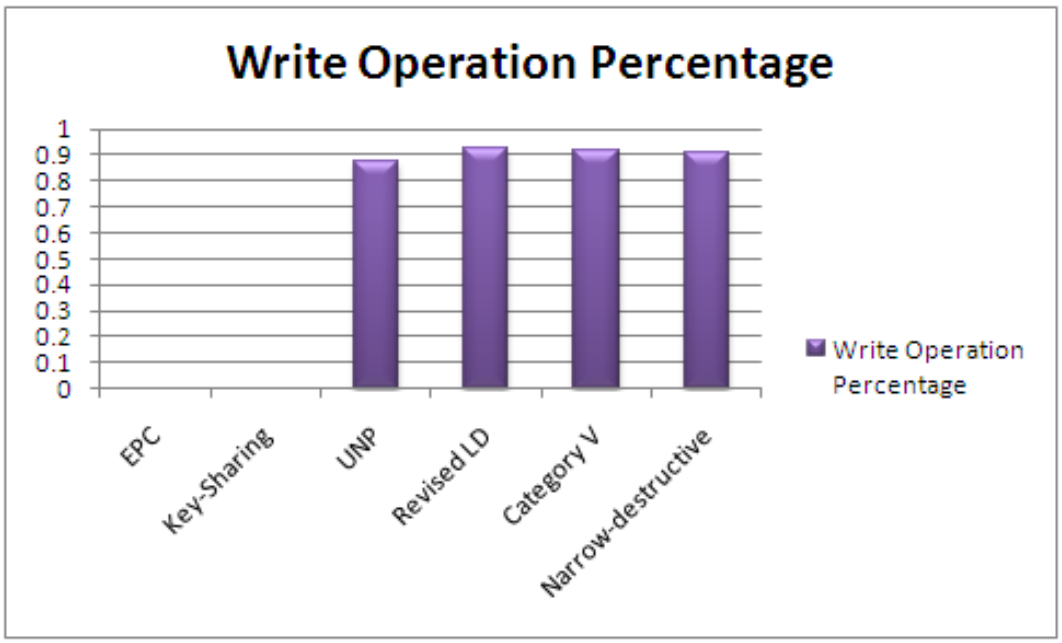


Figure 5.23: Write Operation Percentage

than 80% of total time cost. The other high-cost operation is $R^{R->T}$. We take 128 bit operation as an example, $XW_{128}^{R->T} > XW_{128}^{T->T} > XR_{128}^{R->T} > XR_{128}^{T->T}$. In summary, one suggestion for future protocol design is to reduce write operation as few as possible.

In certain situations, there exists a trade-off between the search cost in reader's database and the tag-related cost. For example, the tree structure in the big brother protocol and the MWTree protocol can be used to reduce the search cost from O(N) to O(log(N)), at the same time, the tag-related costs increase dramatically. This is because the special data structure requires extra tag-related operations such as HMAC-based PRFs and $R^{R->T}$. Therefore, if the number of tags is not too large, RFID system's administrator should avoid adopting such data structure in reader's database, which may result in high tag-related cost. On the contrary, if there are a large number of tags in an RFID system, the administrator should compare the two types of costs in advance and make an optimal decision.

Finally, we note that our categorization does not cover all security properties in RFID system, but mainly focuses on anti-tracing and forward secrecy. In Table 5.15, some other potential attacks and their corresponding possible solutions are

Table 5.15: Attacks and Defenses in RFID Systems

Attack	**Defense**
Tag Cloning	Tag Authentication
Eavesdropping	PRF, Hash
Tag Tracing	PRF, Hash
Unauthorized Reading	Reader Authentication
Tag Impersonation	Store Secret
Replay Attack	Challenge and Response Authentication
Forward Traceavility	Update key
Backward Traceability	Cannot Intercept all Messages
De-synchronization	Store Old Secrets
Denial of Service	Store Old Secrets

illustrated.

Chapter 6: Conclusions and Future Research

The thesis focused on security and performance analysis for RFID protocols. First, existing RFID protocols were classified into six categories by their anti-tracing and forward secrecy properties. The six categories included: EPC protocol, tracing protocols, strong anti-tracing protocols, weak anti-tracing and weak forward secrecy protocols, strong anti-tracing and weak forward secrecy protocols, strong anti-tracing and strong forward secrecy protocols. Among them, EPC protocol and tracing protocols were merely discussed in previous categorization works. However, they were particularly relevant to practice. In this sense, our cateforization model is more relevant to practice than other categorization models.

The administrator of an RFID system can choose protocols in different categories according to the system's security requirements. The trade-off is that higher security usually implies worse performance. We analyzed the performance in two perspectives: search cost of a tag in a reader's database and tag-related cost. The tag-related costs include: cryptographic operations cost in tags and communication cost between reader and tag.

On one hand, we investigated security and search cost by category using traditional database complexity analysis. We found that in some situations, higher security levels resulted in higher search cost. In other situations, search cost was not directly affected by security properties, but it was more related to the data structure of a reader's back-end database.

On the other hand, we examined security and tag-related cost based on some experimental results. A generic formula was set up to calculate the tag-related time cost of any RFID protocol. The tag-related time costs of protocols in each category were calculated using this formula. The best performance protocols in each category were selected as benchmarks to evaluate other protocols' performance. By

comparison, redundant operations in a couple of existing RFID protocols are discovered for revision. Finally, suggestions are proposed for future protocol design so as to make a better trade-off between search cost and tag-related cost. This work is significant because both high-cost operations on resource-limited RFID tags and overwhelming search overhead in a reader's back-end database might incur a long time delay in an RFID system.

In the future, we intent to evaluate more protocols' performance in each category. In addition, aside from time costs evaluation in this thesis, other costs such as gate count and energy consumption can be tested and calculated.

Bibliography

[ADO05] Gildas Avoine, Etienne Dysli, and Philippe Oechslin. Reducing Time Complexity in RFID Systems. In Bart Preneel and Stafford Tavares, editors, *Selected Areas in Cryptography – SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 291–306, Kingston, Canada, August 2005. Springer.

[AO05] G. Avoine and P. Oechslin. A scalable and provably secure hash-based RFID protocol. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 110–114, 2005.

[BBEG09] Côme Berbain, Olivier Billet, Jonathan Etrog, and Henri Gilbert. An Efficient Forward Private RFID Protocol. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *Conference on Computer and Communications Security – ACM CCS'09*, pages 43–53, Chicago, Illinois, USA, November 2009. ACM, ACM Press.

[BdMM08] Mike Burmester, Breno de Medeiros, and Rossana Motta. Robust, anonymous RFID authentication with constant key-lookup. In *ASIACCS '08: Proceedings of the 2008 ACM symposium on Information, computer and communications security*, pages 283–291. ACM, 2008.

[Bin00] Benny Bing. Broadband Wireless Access. Norwell, MA, USA, 2000. Kluwer Academic Publishers.

[CCB06] T. van Le C. Chatmon and M. Burmester. Secure anonymous RFID authentication protocols. *Technical Report TR-060112*, 2006.

[CHT09] Jung Hee Cheon, Jeongdae Hong, and Gene Tsudik. Reducing RFID Reader Load with the Meet-in-the-Middle Strategy. Cryptology ePrint Archive, Report 2009/092, 2009.

[CKSK08] Alexei Czeskis, Karl Koscher, Joshua R. Smith, and Tadayoshi Kohno. Rfids and secret handshakes: defending against ghost-and-leech attacks and unauthorized reads with context-aware communications. In *ACM Conference on Computer and Communications Security*, pages 479–490, 2008.

[CLL+10] Kevin Chiew, Yingjiu Li, Tieyan Li, Robert Deng, and Manfred Aigner. Time Cost Evaluation for Executing RFID Authentication Protocols. In *Internet of Things 2010 Conference (IOT)*, Tokyo, Japan, Nov 29-Dec 1 2010.

[CLLD09a] Shaoying Cai, Tieyan Li, Yingjiu Li, and Robert H. Deng. Ensuring dual security modes in RFID-enabled supply chain systems. In *ISPEC '09: Proceedings of the 5th International Conference on Information Security Practice and Experience*, pages 372–383, Berlin, Heidelberg, 2009. Springer-Verlag.

[CLLD09b] Shaoying Cai, Yingjiu Li, Tieyan Li, and Robert Deng. Attacks and Improvements to an RFID Mutual Authentication Protocol and its Extensions. In David A. Basin, Srdjan Capkun, and Wenke Lee, editors, *Proceedings of the 2nd ACM Conference on Wireless Network Security – WiSec'09*, pages 51–58, Zurich, Switzerland, March 2009. ACM, ACM Press.

[CLM+09] Shaoying Cai, Tieyan Li, Changshe Ma, Yingjiu Li, and Robert H. Deng. Enabling secure secret updating for unidirectional key distribution in rfid-enabled supply chains. In *ICICS*, pages 150–164, 2009.

[Dim05] Tassos Dimitriou. A Lightweight RFID Protocol to protect against Traceability and Cloning attacks. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm*, Athens, Greece, September 2005. IEEE.

[Dim06] Tassos Dimitriou. A secure and efficient RFID protocol that could make big brother (partially) obsolete. *Pervasive Computing and Communications, IEEE International Conference on*, 0:269–275, 2006.

[Dim08] Tassos Dimitriou. RFID-DOT: RFID Delegation and Ownership Transfer made simple. In *4th International Conference on Security and Privacy for Communication Networks – SecureComm 2008*, Istanbul, Turkey, September 2008.

[DLYZ10] Robert H. Deng, Yingjiu Li, Moti Yung, and Yunlei Zhao. A new framework for rfid privacy. In *15th European Symposium on Research in Computer Security (ESORICS)*, September 2010.

[FDW04] Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong Authentication for RFID Systems using the AES Algorithm. In Marc Joye and Jean-Jacques Quisquater, editors, *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370, Boston, Massachusetts, USA, August 2004. IACR, Springer.

[FR06] Martin Feldhofer and Christian Rechberger. A Case Against Currently Used Hash Functions in RFID Protocols. In Robert Meersman,

Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems 2006 – OTM 2006*, volume 4277 of *Lecture Notes in Computer Science*, pages 372–381, Montpellier, France, November 2006. Springer.

[GC97] Network Working Group and R. Canetti. Rfc2104 rfc, hmac: Keyed-hashing for message authentication. *RFC*, 2104:2104, 1997.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GJP05] Simson Garfinkel, Ari Juels, and Ravi Pappu. RFID Privacy: An Overview of Problems and Proposed Solutions. *IEEE Security and Privacy*, 3(3):34–43, May-June 2005.

[HM04] Dirk Henrici and Paul Müller. Hash-Based Enhancement of Location Privacy for Radio-Frequency Identification Devices Using Varying Identifiers. In Ravi Sandhu and Roshan Thomas, editors, *International Workshop on Pervasive Computing and Communication Security – PerSec 2004*, pages 149–153, Orlando, Florida, USA, March 2004. IEEE, IEEE Computer Society.

[IFF] IFF Identification Friend or Foe. http://www.tscm.com/iff.pdf, December.

[Inc08] EPCglobal Inc. EPC^{TM} Radio-Frequency Identity Protocols. Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz 960 MHz Version 1.2.0. 2008.

[JPP08] Ari Juels, Ravikanth Pappu, and Bryan Parno. Unidirectional Key Distribution Across Time and Space with Applications to RFID Security. In *17th USENIX Security Symposium*, pages 75–90, San Jose, California, USA, July 2008. USENIX.

[JRS03] Ari Juels, Ronald Rivest, and Michael Szydlo. The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. In Vijay Atluri, editor, *Conference on Computer and Communications Security – ACM CCS*, pages 103–111, Washington, DC, USA, October 2003. ACM, ACM Press.

[Jue06] Ari Juels. RFID Security and Privacy: A Research Survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, February 2006.

[KO10] Florian Kerschbaum and Nina Oertel. Privacy-Preserving Pattern Matching for Anomaly Detection in RFID Anti-Counterfeiting. In *Workshop on RFID Security – RFIDSec'10*, Istanbul, Turkey, June 2010.

[KP06] Sandeep Kumar and Christof Paar. Are Standards Compliant Elliptic Curve Cryptosystems Feasible on RFID? In *Workshop on RFID Security – RFIDSec'06*, Graz, Austria, July 2006. Ecrypt.

[LD07] Yingjiu Li and Xuhua Ding. Protecting RFID Communications in Supply Chains. In Feng Bao and Steven Miller, editors, *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security – ASIACCS '07*, pages 234–241, Singapore, Republic of Singapore, March 2007. ACM, ACM Press.

[LLG08] Tong-Lee Lim, Tieyan Li, and Tao Gu. Secure RFID identification and authentication with triggered hash chain variants. In *ICPADS '08: Proceedings of the 2008 14th IEEE International Conference on Parallel and Distributed Systems*, pages 583–590, Washington, DC, USA, 2008. IEEE Computer Society.

[LLM+09] Bing Liang, Yingjiu Li, Changshe Ma, Tieyan Li, and Robert Deng. On the untraceability of anonymous RFID authentication protocol with constant key-lookup. In *ICISS '09: Proceedings of the 5th International Conference on Information Systems Security*, pages 71–85, Berlin, Heidelberg, 2009. Springer-Verlag.

[LZL08] K. Jaganathan L. Zhu and K. Lauter. Elliptic Curve Cryptography (ECC) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT). Microsoft Corporation, September 2008.

[MB83] Robert M. Metcalfe and David R. Boggs. Ethernet: distributed packet switching for local computer networks. *Commun. ACM*, 26(1):90–95, 1983.

[MLDL09] Changshe Ma, Yingjiu Li, Robert Deng, and Tieyan Li. RFID Privacy: Relation Between Two Notions, Minimal Condition, and Efficient Construction. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *Conference on Computer and Communications Security – ACM CCS'09*, pages 54–65, Chicago, Illinois, USA, November 2009. ACM, ACM Press.

[MRT09] Aikaterini Mitrokotsa, Melanie R. Rieback, and Andrew S. Tanenbaum. Classifying RFID Attacks and Defenses. *Information Systems Frontiers*, July 2009.

[MSW05] David Molnar, Andrea Soppera, and David Wagner. A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags. In Bart Preneel and Stafford Tavares, editors, *Selected Areas in Cryptography – SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 276–290, Kingston, Canada, August 2005. Springer-Verlag.

[MW04] David Molnar and David Wagner. Privacy and Security in Library RFID: Issues, Practices, and Architectures. In Birgit Pfitzmann and

Peng Liu, editors, *Conference on Computer and Communications Security – ACM CCS*, pages 210–219, Washington, DC, USA, October 2004. ACM, ACM Press.

[NSMSN08] Ching Yu Ng, Willy Susilo, Yi Mu, and Reihaneh Safavi-Naini. Rfid privacy models revisited. In *ESORICS*, pages 251–266, 2008.

[OSK03] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic Approach to "Privacy-Friendly" Tags. In *RFID Privacy Workshop*, MIT, Massachusetts, USA, November 2003.

[PV08] Radu-Ioan Paise and Serge Vaudenay. Mutual Authentication in RFID: Security and Privacy. In Masayuki Abe and Virgil D. Gligor, editors, *Proceedings of the 3rd ACM Symposium on Information, Computer and Communications Security – ASIACCS'08*, pages 292–299, Tokyo, Japan, March 2008. ACM, ACM Press.

[RCT05] Melanie Rieback, Bruno Crispo, and Andrew Tanenbaum. RFID Guardian: A Battery-Powered Mobile Device for RFID Privacy Management. In Colin Boyd and Juan Manuel González Nieto, editors, *Australasian Conference on Information Security and Privacy – ACISP'05*, volume 3574 of *Lecture Notes in Computer Science*, pages 184–194, Brisbane, Australia, July 2005. Springer-Verlag.

[RCT06] Melanie Rieback, Bruno Crispo, and Andrew Tanenbaum. The Evolution of RFID Security. *IEEE Pervasive Computing*, 5(1):62–69, January–March 2006.

[Rot08] Pawel Rotter. A Framework for Assessing RFID System Security and Privacy Risks. *IEEE Pervasive Computing*, 7(2):70–77, June 2008.

[SE09] Sarah Spiekermann and Sergei Evdokimov. Privacy Enhancing Technologies for RFID - A Critical Investigation of State of the Art Research. In *IEEE Privacy and Security*. IEEE, IEEE Computer Society, 2009.

[SM08] Boyeon Song and Chris J. Mitchell. RFID Authentication Protocol for Low-cost Tags. In V. D. Gligor, J. Hubaux, and R. Poovendran, editors, *ACM Conference on Wireless Network Security, WiSec'08*, pages 140–147, Alexandria, Virginia, USA, April 2008. ACM Press.

[Su10] Chunhua Su. RFID Mutual Authentication Protocols with Universally Composable Security. 2010.

[SWE02] Sanjay Sarma, Stephen Weis, and Daniel Engels. RFID Systems and Security and Privacy Implications. In Burton Kaliski, Çetin Kaya ço, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 454–469, Redwood Shores, California, USA, August 2002. Springer-Verlag.

[Tsu06] Gene Tsudik. YA-TRAP: Yet Another Trivial RFID Authentication Protocol. In *International Conference on Pervasive Computing and Communications – PerCom 2006*, Pisa, Italy, March 2006. IEEE, IEEE Computer Society Press.

[Vau07] Serge Vaudenay. On Privacy Models for RFID. In *Advances in Cryptology - Asiacrypt 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 68–87, Kuching, Malaysia, December 2007. Springer-Verlag.

[WSRE03] Stephen Weis, Sanjay Sarma, Ronald Rivest, and Daniel Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In Dieter Hutter, Günter Müller, Werner Stephan, and Markus Ullmann, editors, *International Conference on Security in Pervasive Computing – SPC 2003*, volume 2802 of *Lecture Notes in Computer Science*, pages 454–469, Boppard, Germany, March 2003. Springer-Verlag.

Curriculum Vitae

Bing LIANG was born on January 8, 1983, in Beijing, China, and is a China citizen. She received her bachelor degree of Electronic Engineering from Tsinghua University, Beijing, China, 2006. Her research results have been published in two conferences:

Bing Liang, Yingjiu Li, Changshe Ma, Tieyan Li, Robert Deng: "On Untraceability of Anonymous RFID Authentication Protocol with Constant Key-Lookup. 5th International Conference on Information Systems Security (ICISS), pages 71-85, Kolkata, India, December 14-18, 2009." and

Bing Liang, Kevin Chiew, Yingjiu Li, Yanjiang Yang: "Privacy Disclosure Analysis and Control for 2D Contingency Tables Containing Inaccurate Data. Privacy in Statistical Databases (PSD), pages 1-16, Corfu, Greece, September 22-24, 2010."

Her research interests include RFID security and database privacy.