2010

A privacy-preserving authentication protocol for smart tags

Michael Chih Huong Fong
Iowa State University

A privacy-preserving authentication protocol for smart tags

by

Michael Chih Huong Fong

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Wensheng Zhang, Major Professor
Yong Guan
Ting Zhang

Iowa State University

Ames, Iowa

2010

DEDICATION

This thesis is dedicated to my lovely sister Melissa and to my wonderful parents, Joseph and Lilian, who have raised me to be the person I am today. You have been with me every step of the way, through good and bad times. Thank you for all the unconditional love and support that you have always given me and instilling into me that I am capable of doing anything I put my mind into.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my appreciation to those who helped me with various aspects of conducting this research and writing of this thesis.

First and foremost, I would like to thank my major professor Prof. Wenshang Zhang for providing me with an opportunity to do research in this topic, and his invaluable insights, patience and guidence that have often inspired me to complete my graduate study.

I would also like to thank Dr. Ting Zhang and Dr. Yong Guan for serving on my thesis committee. Thank you for your support and guidance.

Lastly, I would like to thank Chuang Wang for many insightful discussion on Elliptic-Curve computations and uses of WM-ECC library.

ABSTRACT

RFID system has become a technology that many companies would like to adopt as it provides convenience to our society. Increasing competitiveness in the industry has reduced the cost of the technology, but it also raises considerable privacy concerns. Among many RFID applications that have made major impacts on various industries, the RFID-based authentication system has become a solution to resolve a number of issues, for instances, theft, compromised brand, lack of inventory control, supply chain inefficiencies, and etc. Many authentication protocols based on symmetric challenge-response scheme have been developed in order to ensure preservation of privacy. However, many of the these schemes cannot fully protect privacy in the presence of malicious readers or insider attacks.

In this thesis, we first investigated possible security and privacy threats that security engineers face from an RFID system. We then presented a new protocol to authenticate smart tags without exposing their private identities and activity patterns with resource-limited devices, such as RFID smart tags or wireless sensor nodes. We further analyzed the RFID system's security strength against various attacking scenarios, such as eavesdropping, collusive attack or tag-compromise situation, based on extensive experiments to validate the feasibility and security of our proposed solution.

CHAPTER 1. Introduction

Radio Frequency Identification (RFID) is a system that wirelessly enables massive identification and accelerated tracking of items. The first RFID tag developed in 1973, was implemented as a passive transponder to unlock a door without a key. However, this device did not open up the commercial market, until the rise of live stock tracking in 1990s(2). Without a standard setter in the 90s, incompatible types of identifiers and protocols were created by application developers and device vendors. In 1999, the AutoID center at the Massachusetts Institute of Technology standardized the development of the RFID technology(22), and resulted in the adoption of an international standard called Electronic Product Code (EPC). Afterward, the wireless capability with no-line-in-sight requirement makes RFID ideal for effective inventory control and fast check out. Thus, each tag embedded with a unique identifier follows a standard EPC, which is anticipated to serve on consumer product as a successor to the ubiquitous Universal Product Code (UPC) on consumer products in the near future. The making of EPC standard has benefited the RFID technology to adopt in a broad range of application nowadays, such as electronic toll collection for highways, inventory management, employee badges, and credit cards.

1.1 Composition of A RFID System

Two components involved in the RFID systems are tags and readers. A tag, usually attached to an item, contains a radio frequency transponder and a read-only (sometimes rewritable) memory chip that, preloaded with a unique identifier. The tags can be distinguished as passive and active tags by their power sources. Active tags include small batteries to transmit information directly to a reader; whereas passive tags take the energy received from

the reader through an antenna and use that energy to transmit their data back to the reader. The read range of passive tags is very short due to lack of power supply on the devices; however, passive tags are more welcomed by the mass market due to their low cost production.

When tags get queried by readers, they respond their unique identifiers. After the readers interrogates the tags, they sometimes transmit the data back to a back-end system, i.e. a database. Therefore, such an RFID reader should equip with an antenna for wireless communication and an RS-232 port or Ethernet jack for LAN communication toward the database system. Once the back-end system receives data transmitted by the RFID reader, the system runs applications or queries in the local database for further processing. In reality, the readers are designed to query multiple tagged items at once and are capable to distinguish between each one of them.

1.2 RFID Applications

Beginning from the 90s, RFID technology has launched in a variety of commercial products. For example, Wal-Mart has started to require their suppliers to adopt RFID technology in their own smart cards after year 2000. This initiative has been the biggest push for commercial usage of this technology in the recent years. Beside, the massive commercial adoption in the marketplace, such technology has also been applied in military in the United States. In the following section, we presented some common applications for RFID tags and highlighting their security concerns.

1. The most well known application of the RFID tags is in the world of supply chain management. Manufacturers, retailers, and logistics providers make exceptional uses of the RFID technology to track, secure and manage goods throughout the entire production cycle. For instance, pharmaceutical industries in the United States, capturing ten percent of the global market, makes $32 billion dollars. The recent increase of counterfeit or diluted drugs has caught the attention of Food and Drug Administration (FDA). FDA considered this as a threat to public health(7). Implementation of the RFID technology could immediately improve pharmaceutical supply chain safety through real-time, offline,

and item-level authentication, from the initial point of manufacturing to the final stage of dispensing drugs to consumers in the pharmacies.

2. Many RFID-based payment systems are widely used in our daily life, including RFID-capable credit cards from major credit card association and companies and payment cards in mass transit systems. This touch-free payment system speeds up transactions, when customers only need to place their credit cards in close proximity to an RFID reader. In addition, many public transportation systems are also RFID-ready in major cities around the world, such as Massachusetts Bay Transportation Authority system in Boston, the Easy Card for Taipei Metro system, and Octopus Card in Hong Kong. The adoption of the RFID technology in these markets has made the mass transit systems transforming from a slow cash collection process to a speedy fare scan-and-go process. Vending machines or many marketplaces in the cities can dispense the transit cards providing riders convenient sale methods and locations.

3. Another common RFID application is the RFID-enabled vehicle immobilizer in the automotive industry. A vehicle immobilizer is a system that prevents a car from being driven, when the car is started with the wrong RFID-equipped key. There is an embedded chip in each key that sends out out an 32-bit encrypted radio-frequency signal forming a particular code. With this code, the driver is able to start the car and activate the fuel pump with ease. As a result, this technology has increased anti-theft capability in the automotive industry.

4. RFID tags can also store personal information for security check-ins. For example, an employee carries an ID card, embedded with a RFID chip, could authenticate his or her identity at the security entry in a facility within a very short period of time. Another real-life example is the United States government issued the first passports containing RFID chips in October 2006. The embedded chips in the new passports store the same personal information as those in the old printed document, including names, nationalities, sex, dates of birth, places of birth, fingerprints, and photos of the passport holders. According

to government officials, the use of the RFID chips allow passports to be scanned and cross-referenced with security databases more easily, reducing the wait time at security check points. However, due to the wireless nature of communication in RFID, identity theft can be achieve more easily without proper security measurement(16). Personal information is exposed for hackers, who would break into the devices, snap personal information, and then walk away with it. Unauthorized duplication of passports not only jeopardizes millions of Americans' privacy, it also threatens national security.

1.3 Motivation

Among many RFID applications that have made major impacts on various industries, the RFID production authentication has become a solution to resolve a number of issues, such as, property theft, brand privacy, inventory mismanagement, supply chain inefficiencies, and other privacy concerns. In particular, the smart-card-based authentication system guarantees authenticity of a personnel to access certain resources, and keeps important personal and activity patterns information from unauthorized individuals. For instance, each employee carries a badge embedded with a smart tag with his or her personal and privileged information. The door in each room has installed an RFID reader, for validating any incoming smart-card carriers and determining if he or she has thee proper security clearance to enter the room according to the results. In addition to that, this function could potentially allow the system to track movement of employees in the building in case of emergency. Upon termination of employment or adjustment to employee's responsibility, a company can modify the security clearance within a short period of time.

In reality, RFID systems, similar to any wireless technology, face numerous security challenges due to their constant exposures in the untrusted environment. Apart from the conventional security threats, such as unencrypted transmissions, lack of data integrity or lack of mutual authentication(9), the most daunting problem in an RFID system is privacy violation. Many of the existing solutions are proposed at a low-cost setting(25; 10; 27); however, they do not prevent hackers attacking to a compromised tag, or from insiders attempting to steal

private information of their colleagues. Thus, we present a protocol with a state-of-the-art elliptic curve cryptographic approach to secure an individual's authenticate without revealing any sensitive information of his/hers while maintaining at an reasonable resource consumption.

Our proposed approach is prototyped using resource-constraint TelosB motes, which simulate the role of smart cards; whereas the PC emulates an RFID reader. We have evaluated the performance of this set up through multiple experiments, and analyzed the security and privacy preservation strengths are analyzed. According to the results of evaluations, the communication and storage overheads meet our resource constraints and successfully authenticate a smart card carrier in less than two seconds.

1.4 Thesis Outline

This thesis presents an authentication protocol for smart cards, which preserves the privacy of smart card carriers. We has started our quest with an introduction to their background, general concerns and motivation for the project in Chapter 1, followed by a collection of related works in the field of RFID security in Chapter 2. After that, the system model and the assumptions of our design as well as an in depth discussion in security and privacy problems including various suggested countermeasure against such threats are presented in Chapter 3, Next, we described our proposed scheme and discussed its security strengths against various forms of attacks in Chapter 4. In Chapter 5, we focus on software architecture and physical implementation on resource-limited TelosB sensors. Experimental results and evaluations are demonstrated in the following Chapter 6. Finally, Chapter 7 closes this thesis with our conclusions and recommendations for future research.

CHAPTER 2. Literature Review

Many papers have proposed solutions to improve RFID security, and we can summarized them into four categories based on different layers of the system and the physical implementations.

2.1 Suppressive Approach

The first category to enhance RFID security is to use a suppressive method. There are multiple ways to suppressive information transmitted between the smart tags and readers. We have included four examples of approaches in the following section. The first suppressive approach(5) applies a unique short, typically one byte, KILL pin to a smart tag. When the tag receives the KILL pin, it erases data stored in its memory, and deactivates itself permanently. This scheme kills the tag, and effectively ensures end user's privacy in the future. One major drawback is that it also eliminates future applications on this tag. In EPC Global Class 1 Generation 2, a kill command can be triggered by sending a 32-bit KILL pin to the tag.

Another proposed scheme(5) is to shield an RFID tag with a metal or foil container, known as a Faraday Cage. Such container is made of metal mesh or foil that is impenetrable to radio signals of certain frequencies. By covering the RFID tag, with those material, all communications between the tag and the reader are blocked. Hence, the user's privacy and security is achieved in this way. A good example of this protection is the new generation of United States passports. They are issued with covers that protect the privacy and security of passport holders. This approach, however, is not foolproof in practice. The security concern is temporarily relieved when the shields is shut. When the cover is partially open, the RFID passport is continuously exposed to external threats(11).

The third approach is active jamming(5) in which a radio frequency device is used to actively broadcast radio signals to block or disrupt the operations of any nearby RFID readers. Unfortunately, such powerful radio signal interfere all RF devices. Hence, this double-edged approach blocks signals from both illegitimate and legitimate readers in surrounding area.

Last, blocking approach, advocated by Rivest and et al(12)., requires no major modifications on tags, but rather incorporates the tags with one modifiable bit, called privacy bit. When the privacy bit in tags is set to 0, tags are available for public scanning. Nevertheless, if the privacy bit is set to 1, tags are in the privacy zone, and they are under protection of the specially made tags, called blocker tags. Blocker tags are responsible to permit authorized readers to proceed with normal activities, while preventing unwanted reading of tags from fraudulent readers. The blocker tags simply emit both a 0 and an 1 signals in response to all reader queries. An unauthorized reader would believe all possible tag identifiers are present and attempt to traverse the entire identifier tree. The tree size is too big to be fully scanned, thus, this hopeless attempt eventually stalls the reading device. This approach provides privacy enhancement at an affordable cost in the industry. However, the authors do not provide foolproof protection, to avoid a denial of service (DoS) threat as the fail attempt as the blocker tags fails to provide the intended service to the legitimate tags.

2.2 Solutions Based on Hardware Implementation

Another approach to improve RFID security is to use more sophisticated hardware. Molnar, Soppera and Wagner(24) proposed an alternative for readers to equip with trusted platform modules (TPMS). Such secured hardware implementation can maintain privacy policies within the tag, and readers can validate the tags corresponding to these policies. In their paper, they does not address the problem of counterfeit or compromised reader situations.

Rieback, Crispo and Tanenbaum(21) implemented a device called RFID Guardian, which acts as a personal RFID firewall. This devices intervenes requests between readers and tags, so that the Guardian can control tags' actions. Any outsider cannot intervene activities under Guardian's control. In addition, a Guardian can implement sophisticated privacy policies to

enhance its security measurements.

2.3 Cryptographic Approach

The third avenue to heighten security measurement in RFID devices is to utilize a cryptographic approach. One can apply such approach by using a couple of methods. Weis(28) firstly proposed a hash-based scheme, called Hash Lock. Combined with a back-end database to perform RFID authentications, each tag uses a hash of a random key as its metaID. When locked, a tag responds to all queries with its metaID to the reader. Then, the reader obtains the real ID of the tag from the database based on the metaID received. However, this scheme does not provide security in the future because the same metaID may be used repeatedly. Therefore, Weis extends the original scheme with another randomized method, called Randomized Access Control, which employs a random number generator to prevent from tracking of the metaID. With this improved protocol, tags are still susceptible to tag impersonation attack, since an intercepted response can be replayed.

Shamir(23) showcased a construction of an one-way function based on Rabin cryptosystem, called SQUASH. SQUASH - for SQUaring hASH - is provably as secure as factoring, and fully amenable to implementation on RFID tags. However some counter-attacks are proposed against the weakness of Rabin cryptosystem, and ultimately effects the security of SQUASH(19).

Jung and Lerch(15) presented a state of the art AES (Advanced Encryption Standard) suitable for RFID tags with 8-bit micro-controller. This AES unit can perform encryption and decryption of a 128-bit block size. In addition, Feldhofer et al.(4) presented an challenge-response authentication protocol based on RFID tags implemented with AES standard.

2.4 Challenge-Response Protocol

The last method to increase the security level is to use challenge-response protocols. We explore many recommendations in this realm in the following section. Mulnar and Wagner(18) suggested a server-less authentication system that both parties use shared secrets and individ-

ually contribute a random number to protection the messages communicated in the channel. Since the reader knows the shared secret, its own nonce, and previously tag-generated nonce, thus, the reader can obtain the tag ID in a secured channel. They have also built a tree-based protocol to provide scalable authentication with search complexity $O(logn)$. Each tag represents a leaf nodes in the tree, and each edge is associated with a secret between two nodes. In addition, a tag may be loaded with multiple secrets corresponding to the path from the tag to the root. However, this protocol does not guarantee backward untraceability, especially when a reader is compromised. In that case, the adversaries who attacked the reader can learn the secret keys to the very tag.

Similarly, Dimitriou(3) used a secret-sharing authentication protocol, in which both reader and tags employs their own random numbers. In this challenge-response scheme, when a reader queries, the tag response with a hash of its identifier. Then, the reader gives this hash to the secured server. After confirming the message, the secured server sends back a valid message to the reader, and the reader redirects this received message back to the tag. The tag will then verify the message sent by the reader. If the value matches, then the tag knows the reader has been authenticated by the server, and updated its secret ID. Otherwise, the tag remains the old ID. Moreover, the scheme is also prone to tag impersonation attack, because the same hashed tag identifier could be reused between valid sessions.

Ohkubo, Suzuki and Kinoshita(25) advocated another challenge-response scheme based on a sequence of hashed operations. During reader's interrogation, the tag sends its hashed identifier and renews its identifier by using a second hash function. Only the legitimate readers can link all the hashed values sent by the tag, as opposed to an attacker who cannot figure out the linkability.

Juel and Weis(13) introduced an symmetric-key authentication approach for low-cost RFIDs based on HB protocol from Hopper and Blum. This light-weight HB^+ protocol only requires bit-wise AND and XOR operation plus one random noise bit. The security of HB^+ protocols is based on the Learning Parity with Noise Problem, whose hardness over random instances still remains as an open question. However, Piramuthu has found its weakness against a realistic

active attack in his paper.

Tsudik(27) described a simple RFID authentication protocol, called YA-TRAP. The author aimed this novel approach at presumptions that tag information is processed in batches, and additionally RFID tags have their own power source to keep track of time. In this scheme, the reader sends a time-stamped message to a tag, which authenticates the reader's identity via evaluating the received time-stamp. If the time-stamp is invalid, the tag will output a random reply; otherwise, it will return an encrypted reply based on the received time-stamp and its own internal time-stamp. Last, the reader sends this reply back to a back-bend server to obtain the real tag data. Although Tsudik has not formally analyzed the security properties of YA-TRAP, other experiments have proved that this scheme is prone to several security threats, such as "future-time" attack in which the adversary queries the tag off line with several valid time references in the near future. Then, the adversary can captures the tag's responses and use their responses for online authentication during these future time periods. Therefore, this protocol does not provide future untraceable privacy.

In a recent paper proposed by Dr. Guan and Kulseng(17), the authors presented a lightweight solution to solve ownership transfer problem. In order to keep the cost of the RFID tags low, the proposed solution is implemented by the minimal cryptographic circuits, including Linear Feedback Shift Registers and Physical Unclonable Functions. In their design, the authors utilized their own mutual authentication protocol, equipped with minimalistic hardware on field-programmable gate array (FPGA). This set up can transfer the ownership of an RFID tag from one tag holder to another without revealing information from the old owner of the tag. Furthermore, they studied the secured search problem prevalent in low-cost RFID systems. They proposed several low-cost solutions based on LFSRs and PUFs, so that an legitimate reader securely searches for a particular tag. The solutions prevent adversaries from learning tag identity or cloning responding messages. This hardware implementation in the FPGA environment provides great efficiencies, lowers hardware consumptions (less 1,500 logic gates) and defends a variety of attacks.

2.5 Comparison Summary

Some approaches pointed above focus to block attacks creating barriers; however, these are not foolproof as the barriers can be penetrated in practice. For instances, radio frequency (RF) eavesdropping listens to the RF communication by simply using an antenna; whereas cloning is achieved by duplicating tag signals using reverse engineering or signal simulation techniques. KILL pin approach deactivates a working tag after a single usage, which can never be re-activated. Although the design of cryptographic protocol promises better confidentiality; however, their capabilities to preserve users' privacy are still questionable. (14).

Challenge-response protocol sometimes carries out a time-memory trade-off on a larger memory space. We can observe the trade-off when comparing Mulnar and Wagner's scheme (MW) (18) and Ohkubo, Suzuki and Kinoshita's approach (OSK) (25). OSK requires pre-computation to store for the hashed sequence on the server side. On their other hand, MW's scheme does not require pre-computation, however, its security strength rests on a large branching factor δ, of which the branching tree eventually consumes large amounts of run-time memory.

To overcome the aforementioned limitations in the existing solutions, we need a new application-specific protocol to protect users' security as well as privacy. We propose a different privacy-preserving authentication scheme that is based on computational hardness of well known as elliptic curve discrete logarithmic problem (ECDLP).

CHAPTER 3. Framework

3.1 Security and Privacy Concerns

Due to the hardware obstacle, security and privacy in RFID network work quite differently from the conventional network. Security is an act to prevent sensitive data from being stolen or altered; whereas privacy focuses on people, in a broader sense, protecting secret information about themselves without unsanctioned intrusions. Any information leakage related to the individual - for example, activity patterns - is under the protection of privacy preservation. Without a doubt, security and privacy are definitely critical issues for this uprising technology.

3.1.1 Information Security

This section introduces four aspects of current security concerns in the RFID environment: confidentiality, integrity availability and authenticity. We also discussed how these factors have influenced users' privacy, as well as how the hackers attack an RFID system in the real world.

Confidentiality

Confidentiality refers to concealment of information in a way that each party should not leak data to an unauthorized person. Examples of this kind of attacks are eavesdropping, spoofing, reader/tag comprised attacks. In general, eavesdropping could happen when an unauthorized party has been listening to an unencrypted channel. The standard solution for keeping sensitive data secretive is to encrypt the data with a private key, known only to the sender and receiver. This is an extremely important concept in a commercial application since the data stored in the tag is sometimes highly confidential. Spoofing attacks, on the other hand, focus to clone an identity of a tag by replaying the intercepted messages. A common

solution to this kind of attacks is to add a pseudo-random variable or time reference into the transmitting packet to prevent packet duplications.

Integrity

In the wireless environment, the information exchanged between two parties needs to be confidential when sensitive data, such as personal bio-matrix, must not be collected by an eavesdropper. Fortunately, with the implementation of confidentiality, the attacker may not steal the information. However, an active eavesdropper may modify the message in transit without knowing the contents of the message. The well-known example Diffie-Hellman protocol is a great example of a message-exchange protocol with no authentication, and thus, it is vulnerable to man-in-the-middle attacks. Message authentication codes, hash functions and digital signatures can protect any received data from being altered in transit.

Availability

Availability is an important aspect of reliability, especially when a reader needs to be ready to authenticate every smart tag that may enter its communication range at certain time intervals. However, realization of availability is probably the toughest feature to implement in the real world scenario. An adversary could simply cover the tags or the readers with metal cages or actively jam the chancel with an RF-device to prevent the smart tags from establishing communications with the readers. In such case, the denial-of-service attacks (DoS) are achieved. In addition, component theft is also inevitable to avoid because tags and readers are exposed in the open-environment at all time.

Authenticity

In a network system, authentication proves the claimed identity of the other parties and it is an important security measure for preventing counterfeit identity. Both the readers and users' tags need to confirm the identity of each party involved in the communication, even without revealing the secrets from the other party.

3.1.2 Threats against Privacy

Privacy preservation, in general, refers to the ability of an entity to stop information from being known to unauthorized people. In a system that lacks confidentiality and authenticity, there is a great chance for information to leak to unauthorized interrogators, leading to a violation of user privacy. For example, a person holds an RFID-embedded passport may reveal his or her private information to unauthorized attacker inadvertently without proper privacy measurement.

Another correlated privacy concern is tracking of an individual's activities by tracing the wireless RFID tags. Unfortunately, even with the presence of encryption and security protection, an experienced attacker could still obtain a history of visited locations, social interaction of an individual. For instance, attackers could determine user's movement - or even geographic position- by correlating data from placing multiple readers at several fixed locations.

In this section, we will discuss some common adversarial acts that violates the privacy of an individual.

Eavesdropping

By listening to the data, the adversary could easily discover the contents of communication. The discovery of sensitive data, such as an identification number, could be used to track a person. In consequence, this act seriously violates the privacy of an individual.

Traffic Analysis

This is an advanced attack from eavesdropping. An attacker could potentially monitor the transmitted packets between certain readers and RFID tags. Through analysis of the network traffic, the adversary can take advantages of seemly irrelevant data, analyze them, and derive much more sensitive information and behavior patterns. For example, he or she could effectively identify users with previous activity patterns, and even predict future users' movement. For this reason, this act violates personal and location privacy.

3.2 Attack Model

In this section below, we described some of the common attack strategies to steal private information from smart tags:

1. An unauthorized attacker attempts to peculate private information from a smart tag holder with following possible means:

   (a) Eavesdropping activities via fraudulent readers:

   An outside eavesdropper may attempt to decode hidden information in the communication message. Consequently, the user privacy is no longer under protection.

   (b) Impersonating smart tags:

   An adversary may attempt to clone a tag with information fetched from a breached smart tags, either via physically compromising the tags or eavesdropping on a legal channel. The attacker may conduct illegal activities under the wrong name of the stolen smart tag.

   (c) Replaying messages:

   An adversary engaging a replaying attack can monitor on a legitimate communication channel between smart tags and readers and replay the intercepted messages at some later time. Since the replayed messages originated from the authorized smart tags, the same receiver will accept them again. In that situation, the attacker could pass the security checkpoint.

2. A unethical smart card holder misuses his or her privileges to private information:

   (a) Passing private information to unauthorized individuals:

   Malicious owners of smart tags may release authentication to outsiders without impersonating any existing smart tags. The outsider can pass the authentication, and does illegal activities. Even their misbehavior is detected, there is no smart tag can be identified to link with the real violator.

(b) Colluding efforts among smart tag owners:

Several misbehaving owners of smart tags may attempt to break into the system by sharing their private information, decode the security system and ultimately pass the security checkpoint. We will see more details about this attack in the next chapter.

3. The operator of the readers attempts to guess the private information from an incoming smart card holder. Nevertheless, we assume that the legitimate reader cannot be physically compromised by an outsider.

All of these attacks could be prevented in our protocol as discussed in Chapter 4 under the assumption that the readers are difficult to compromised as they are usually belong to an organization, e.g. security department of a company.

3.3 Assumptions and Design Goals

The following assumptions are made regarding the system:

1. The open nature of the wireless communication between smart tags and RFID readers enables outside attackers to monitor the communication.

2. Smart tags can be physically stolen, and attackers can use the stolen tags to impersonate identity to access resources. However, the RFID readers in our scheme cannot be stolen.

3. The trusted third-party authority is always trustworthy and will not deviate from its fiduciary responsibility.

4. Each smart tag has a rewritable memory chip, and is computationally capable to perform a light weight cryptographic function, such as SHA-1.

Based on these assumptions, we identify the following design goals in our scheme to counterwork the attack scenarios we mentioned in the previous section:

- Defensibility against attacks:

It is required to be computationally impossible for the adversary, either an unauthorized

outsider, a legitimate but misbehaving owner, or an unethical operator of RFID readers, to violate the privacy of an innocent smart card holders, including impersonating his or her identity from a customized smart card by spoofing or replaying attacks.

- Untraceability of users' behaviors:

  After RFID readers verify the legitimacy of a smart tag, the private information from the owner should remain confidential during subsequent communication involved with the tag. Consequently, the reader cannot learn the identity information from the owner.

- Accountability

  The trusted third-party authority should be able to trace back the identity of a smart tags when necessary. For example, if a smart tag has been reported loss or stolen, the stolen tag should be revoked, such that the adversary should not be able to use this stolen tag to impersonate the identity of original owner to gain access.

- Affordability on resources

  The proposed scheme will run on resource-limited devices, in which computation, communication and storage overheads need to be carefully measured.

CHAPTER 4. Proposed Scheme

In this section, we proposed a scheme to authenticate smart cards without revealing private information from the users.

4.1 Preliminaries

We introduced the cryptographic concept, such as one-way hash function and elliptic curve discrete logarithm problem, and established a few notations used throughout this paper in the following sections:

4.1.1 Hash

A hash function h is an one-way function that maps an arbitrary length input to a k-bit output, i.e. $h : \{0,1\}^* \rightarrow \{0,1\}^k$. The typical requirement for this cryptographic check-sum functions is described as follows:

- Pre-image resistance: For any given input x, it is computationally efficient to compute h(x). However, given an arbitrary output y, it is computationally infeasible to find an input such that h(x) = y.
- 2nd pre-image resistance: Given x, it is computationally infeasible to find $x' \neq x$, such that $h(x) = h(x')$
- Collision resistance: It is computationally infeasible to find any pair of distinct inputs x and x', such that $h(x) = h(x')$

We assume the hash function in the design is unbreakable, however, we adopt SHA-1 hash function in the real implementation.

4.1.2 Elliptic Curve Discrete Logarithm Problem

The cryptographic system in our proposed protocol uses the elliptic curve group $\mathcal{E}(\mathbb{F}_q)$ of rational points on an elliptic curve, defined over some finite field F_q. Due to the cyclic property of this additive group, it makes multiplication and exponentiation easy. The problem is defined as follows:

Let $\mathcal{E}$ be an elliptic curve over some finite field, $\mathbb{F}_q$ and *ord* denote the order of the group $\mathcal{E}(\mathbb{F}_q)$. Let P denote an element of $\mathcal{E}(\mathbb{F}_q)$ and a point, Q, within subgroup of P, such that $Q \in < P >$. The goal is to find an integer, m, such that

$$Q = m \cdot P \tag{4.1}$$

m is also called the discrete logarithm of Q to the base, P. In the security point of view, this is also known as ECDLP, where the integer, m, is selected uniformly as a private key and the point Q is its corresponding public key.

4.1.3 Symbols and Notations

There are symbols with different format in this paper, and each format has a unique meaning and property in the system.

- The bolded symbols are distinctly chosen in random for each session, e.g., $\boldsymbol{\gamma_2}, \boldsymbol{\lambda}$.
- The underlined parameter indicates those transmitted from the tag side, for instance, $\underline{A(x,u)}$. A good combination of these two formats would be $\underline{A(\boldsymbol{\gamma_2},u)}$, which represents a tag function that needs to feed an input at different session.
- A function with a hat means it is an elliptic curve point multiplied with the output of a function over some finite field. An example would be $\hat{C}(x) = \alpha \times C(x)$, where α is an elliptic curve point.

4.2 Privacy-Preserving Authentication Protocol

System initialization is conducted by trustworthy authority, and is responsible for three tasks: selecting polynomial coefficients, initializing RFID tags and initializing smart tags.

4.2.1 Parameters Selection

Two groups of parameters need to be pre-selected from the trusted third-party. They include points from an elliptic curve, E, and unique coefficients for readers and each smart tags by solving a system of linear equations.

Elliptic Curve Parameters

The trusted authority will determine an elliptic curve E over some finite field, $\mathbb{F}_q$, where *ord* denotes the order of the group $\mathcal{E}(\mathbb{F}_q) = \{(x, y)|x, y \in \mathbb{F}_q\}$, P denotes as an element of $\mathcal{E}(\mathbb{F}_q)$, and a point, Q, within the cyclic subgroup of P, $Q \in < P >$. Finally, a secret elliptic curve point α is chosen from the pool of $< P >$.

Reader/User Tag Parameters

The trusted authority determines three polynomials $C(x), D(x)$ and $E(x)$ over prime finite field, Z_p, for the readers, where

$$C(x) = c_1 x + c_0 \tag{4.2}$$

$$D(x) = d_1 x + d_0 \tag{4.3}$$

$$E(x) = e_2 x^2 + e_1 x + e_0 \tag{4.4}$$

Then, the trusted authority chooses universal polynomials $A(x, y)$ and $B(x, y)$ over the same prime finite field, Z_p, for every tags, where

$$A(x, y) = a_{1,1}xy + a_{1,0}x + a_{0,1}y + a_{0,0} \tag{4.5}$$

$$B(x, y) = b_{1,1}xy + b_{1,0}x + b_{0,1}y + b_{0,0} \tag{4.6}$$

such that

$$A(x, y) \cdot C(x) - B(x, y) \cdot D(x) + E(x) = 0 \tag{4.7}$$

In addition, it is crucial to have either of the following cases to guarantee both x and y affect the result of the revocation process:

$$b_{1,1} \neq b_{1,0}$$

$$b_{0,1} \neq b_{0,0}$$

Combined the above observations to satisfy Equations 4.7, it is necessary to find out a proper set of coefficients in each functions, e.g., $A(x,y), B(x,y), C(x), D(x)$ and $E(x)$, in order to hold the relations in the following system of linear equations:

$$a_{1,1}c_1 - b_{1,1}d_1 = 0 \tag{4.8}$$

$$a_{1,1}c_0 + a_{0,1}c_1 - b_{1,1}d_0 - b_{0,1}d_1 = 0 \tag{4.9}$$

$$a_{0,1}c_0 - b_{0,1}d_0 = 0 \tag{4.10}$$

$$a_{1,0}c_1 - b_{1,0}d_1 + e_2 = 0 \tag{4.11}$$

$$a_{1,0}c_0 + a_{0,0}c_1 - b_{1,0}d_0 - b_{0,0}d_1 + e_1 = 0 \tag{4.12}$$

$$a_{0,0}c_0 - b_{0,0}d_0 + e_0 = 0 \tag{4.13}$$

$$b_{1,1} \neq b_{1,0} \tag{4.14}$$

$$b_{0,1} \neq b_{0,0} \tag{4.15}$$

$C(x), D(x)$ and $E(x)$ remain the same value when they are preloaded onto a reader. For example, $c_1, c_0, d_1, d_0, e_2, e_1$ and e_0 are predetermined prior to the authentication process. Therefore, the authenticator is responsible to verify results from an incoming requests, which carries results of $A(x,y)$ and $B(x,y)$ to satisfy Equation 4.7. Since the unknowns in function $C(x), D(x)$, and $E(x)$ do not play a factor in the system, they have now become a simpler system regarding for coefficients of $A(x,y)$ and $B(x,y)$. The number of unknowns from $A(x,y)$ and $B(x,y)$, i.e., eight unknown variables, is greater than the number of equations, i.e., seven equations. Thus, it is not possible for a naive attacker to break this system of linear equations.

In addition, in our recent discovery, the way to prevent linear collusive attacks is to add an additional equation to check tag's validity. We will see more detail about such attack in the section about security and privacy analysis. Combined with all of the information described above, we have considered additional variables in our systems. The reader has two more items:

$$H(x) = h_2x^2 + h_1x + h_0$$

$$J \in \mathbb{F}_q$$

Each tag is preloaded with two more pieces of information,

$$G(\boldsymbol{x}, u) = g_{2,2}x^2y^2 + g_{2,1}x^2y + g_{1,2}xy^2 + g_{1,1}xy + g_{0,2}y^2 + g_{0,1}y$$

$$I_u \in \mathbb{F}_q$$

To verify the validity of an tag, we combine all information with A(x,y) and B(x,y), such that the following equality holds:

$$A(x,y) \cdot B(x,y) + G(x,y) \cdot J - H(x) \cdot I_u = 0 \tag{4.16}$$

Similarly, to satisfy Equation 4.16, it is necessary to find out a proper set of coefficients in each function, e.g., $A(x,y), B(x,y), G(x,y)$ and $H(x)$, in order to hold the relations in the following system of linear equations:

$$a_{1,1}b_{1,1} + g_{2,2} \cdot J = 0 \tag{4.17}$$

$$a_{1,1}b_{1,0} + a_{1,0}b_{1,1} + g_{2,1} \cdot J = 0 \tag{4.18}$$

$$a_{1,1}b_{0,1} + a_{0,1}b_{1,1} + g_{1,2} \cdot J = 0 \tag{4.19}$$

$$a_{1,1}b_{0,0} + a_{1,0}b_{0,1} + a_{0,1}b_{1,0} + a_{0,0}b_{1,1} + g_{1,1} \cdot J = 0 \tag{4.20}$$

$$a_{0,1}b_{0,1} + g_{0,2} \cdot J = 0 \tag{4.21}$$

$$a_{0,1}b_{0,0} + a_{0,0}b_{0,1} + g_{0,1} \cdot J = 0 \tag{4.22}$$

$$a_{1,0}b_{1,0} - h_2 \cdot I_u = 0 \tag{4.23}$$

$$a_{1,0}b_{0,0} + a_{0,0}b_{1,0} - h_1 \cdot I_u = 0 \tag{4.24}$$

$$a_{0,0}b_{0,0} - h_0 \cdot I_u = 0 \tag{4.25}$$

Here is the list of coefficients selected in the most simplified form in our implementation:

- User's smart tag
  - $A(x,y) : a_{1,1} = 1, a_{1,0} = 1, a_{0,1} = 1, a_{0,0} = 1$
  - $B(x,y) : b_{1,1} = 1, b_{1,0} = 2, b_{0,1} = 1, b_{0,0} = 2$
  - $G(x,y) : g_{2,2} = -1, g_{2,1} = -3, g_{1,2} = -2, g_{1,1} = -6, g_{0,2} = -1, g_{0,1} = -3$

- $I_u = 2$

- Reader

  - $C(x) : c_1 = 1, c_0 = 1$
  - $D(x) : d_1 = 1, d_0 = 1$
  - $E(x) : e_2 = 1, e_1 = 2, e_0 = 1$
  - $H(x) : h_2 = 1, h_1 = 2, h_0 = 1$

With these coefficients, Equations 4.8 through 4.15 and Equations 4.17 through 4.25 are satisfied, which guarantee the equality of both phases of authentication for Equation 4.7 and Equation 4.16.

4.2.2 Reader Initialization

In addition to the parameters selected via a trusted third-party, a reader requires following initialization steps:

1. $\hat{C}(x) = \alpha \times C(x) = \alpha \times (c_1 \cdot x + c_0)$, with pre-selected c's coefficients and an point, α
2. polynomial function $D(x) = d_1 \cdot x + d_0$, with pre-selected d's coefficients.
3. polynomial function $E(x) = e_2 \cdot x^2 + e_1 \cdot x + e_0$, with pre-selected e's coefficients.
4. polynomial function $H(x) = h_2 \cdot x^2 + h_1 \cdot x + h_0$, with pre-selected h's coefficients.
5. an unique big integer, $J \in \mathbb{F}_q$

To sum up, eleven big numbers are chosen from $\mathbb{F}_q$ (i.e. $c_0, c_1, d_0, d_1, e_0, e_1, e_2, h_0, h_1, h_2$ and J) in addition with one elliptic curve point, α.

4.2.3 Smart Tag Initialization

When each tag is manufactured by a trusted authority, it is associated with several hidden identifiers (i.e. $u, s_u, \beta_u, I_u \in \mathbb{F}_q$), which are kept secret (even to tag itself) and known only to the trustworthy creator of the tag.

In addition to the secret tag-specific number, several polynomial functions are loaded:

$$\begin{aligned} A_u(\boldsymbol{x}) &= s_u \cdot A(\boldsymbol{x}, u) \\ &= A_{u,1} \cdot \boldsymbol{x} + A_{u,0} \\ B_u(\boldsymbol{x}) &= b_u \cdot x \\ &= \beta_u^{-1} \cdot (B_{u,1} \cdot \boldsymbol{x}) \\ &= \beta_u^{-1} \cdot \left[(b_{0,1} \cdot u + b_{0,0}) \cdot \boldsymbol{x} \right] \\ G_u(\boldsymbol{x}) &= \beta_u^{-1} \cdot s_u \cdot G(\boldsymbol{x}, u) \\ I'_u &= \beta_u^{-1} \cdot s_u \cdot I_u \end{aligned}$$

Last, the following secret points are also loaded to the tag:

$$\begin{aligned} \alpha_{u,0} &= \alpha \times s_u \\ \alpha_{u,1} &= \alpha \times (s_u \cdot \beta_u) \\ \alpha_{u,2} &= \alpha \times (s_u \cdot B_{u,0}) \\ &= \alpha \times \left[s_u \cdot (b_{0,1} \cdot u + b_{0,0}) \right] \end{aligned}$$

4.2.4 Communication Phase

The reader initiates the communication by sending a random session nonce, γ_0 to the tag. After receiving data from reader, the tag responds a random session nonce of its own, called γ_1, and hashes the concatenations of these two random session variables, which is, $\gamma_2 = H(\gamma_1 | \gamma_0)$. Following by that, the tag computes its secret function given γ_2. Then it

computes the following:

$$R_1 = \lambda \cdot A_u(\boldsymbol{\gamma_2}) \tag{4.26}$$

$$R_2 = \lambda \cdot \lambda'^{-1} \cdot B_u(\boldsymbol{\gamma_2}) \tag{4.27}$$

$$\begin{aligned} R_2' &= \alpha_{u,1} \times \lambda' \\ &= \alpha \times (s_u \cdot \beta_u \cdot \lambda') \end{aligned} \tag{4.28}$$

$$\begin{aligned} R_2'' &= \alpha_{u,2} \times \lambda \\ &= \alpha \times \left[s_u \cdot \lambda \cdot (b_{0,1} \cdot u + b_{0,0}) \right] \end{aligned} \tag{4.29}$$

$$\begin{aligned} R_3 &= \alpha_{u,0} \times \lambda \\ &= \alpha \times (s_u \cdot \lambda) \end{aligned} \tag{4.30}$$

$$\begin{aligned} R_4 &= \lambda^2 \cdot \lambda''^{-1} \cdot G_u(\boldsymbol{\gamma_2}) \\ &= \lambda^2 \cdot \lambda''^{-1} \cdot \beta_u^{-1} \cdot s_u \cdot G(\boldsymbol{\gamma_2}, u) \end{aligned} \tag{4.31}$$

$$\begin{aligned} R_4' &= \alpha_{u,1} \times \lambda'' \\ &= \alpha \times (s_u \cdot \beta_u \cdot \lambda'') \end{aligned} \tag{4.32}$$

$$\begin{aligned} R_5 &= \lambda^2 \cdot \lambda'''^{-1} \cdot I_u' \\ &= \lambda^2 \cdot \lambda'''^{-1} \cdot \beta_u^{-1} \cdot s_u \cdot I_u \end{aligned} \tag{4.33}$$

$$\begin{aligned} R_5' &= \alpha_{u,1} \times \lambda''' \\ &= \alpha \times (s_u \cdot \beta_u \cdot \lambda''') \end{aligned} \tag{4.34}$$

Last, the tag sends back tw packets of data as follows: $< \gamma_1, R_1, R_2, R_2', R_2'', R_3 >$ and $< R_4, R_4', R_5, R_5' >$ to the reader side.

4.2.5 Authentication Phase

With carefully selected coefficient, the authentication process could be achieved by checking the following equality, as shown in the next Figure 4.1:

$$\begin{aligned} R_2' \times R_2 + R_2'' &= (\alpha_{u,1} \times \lambda') \times (\lambda \cdot \lambda'^{-1} \cdot B_u(\boldsymbol{\gamma_2})) + (\alpha_{u,2} \times \lambda) \\ &= \left[\alpha \times (s_u \cdot \beta_u \cdot \lambda')\right] \times \left[\lambda \cdot \lambda'^{-1} \cdot \beta_u^{-1} \cdot (b_{1,1}u + b_{1,0}) \cdot \boldsymbol{\gamma_2}\right] \\ &+ \left\{\alpha \times \left[s_u \cdot (b_{0,1}u + b_{0,0}) \cdot \lambda\right]\right\} \\ &= \alpha \times \left[s_u \cdot \lambda \cdot B(\boldsymbol{\gamma_2}, u)\right] \end{aligned} \tag{4.35}$$

$$\begin{aligned} (R_2' \times R_2 + R_2'') \times R_1 &= \left\{\alpha \times \left[s_u \cdot \lambda \cdot B(\boldsymbol{\gamma_2}, u)\right]\right\} \times \lambda \cdot A_u(\boldsymbol{\gamma_2}) \\ &= \left\{\alpha \times \left[s_u \cdot \lambda \cdot B(\boldsymbol{\gamma_2}, u)\right]\right\} \times \lambda \cdot s_u \cdot A(\boldsymbol{\gamma_2}, u) \\ &= \alpha \times \left[(s_u^2 \cdot \lambda^2) \cdot A(\boldsymbol{\gamma_2}, u) \cdot B(\boldsymbol{\gamma_2}, u)\right] \end{aligned} \tag{4.36}$$

$$\begin{aligned} R_4' \times R_4 \times J &= \left[\alpha_{u,1} \times \lambda''\right] \times \left[\lambda^2 \cdot \lambda''^{-1} \cdot G_u(\boldsymbol{\gamma_2})\right] \times J \\ &= \left[\alpha \times (s_u \cdot \beta_u \cdot \lambda'')\right] \times \left[\lambda^2 \cdot \lambda''^{-1} \cdot \beta_u^{-1} \cdot s_u \cdot G(\boldsymbol{\gamma_2}, u)\right] \times J \\ &= \alpha \times \left[(s_u^2 \cdot \lambda^2) \cdot G(\boldsymbol{\gamma_2}, u) \cdot J\right] \end{aligned} \tag{4.37}$$

$$\begin{aligned} R_5' \times R_5 \times H(\boldsymbol{\gamma_2}, u) &= \left[\alpha_{u,1} \times \lambda'''\right] \times \left[\lambda^2 \cdot \lambda''^{-1} \cdot I_u'\right] \times H(\boldsymbol{\gamma_2}) \\ &= \left[\alpha \times (s_u \cdot \beta_u \cdot \lambda''')\right] \times \left[\lambda^2 \cdot \lambda'''^{-1} \cdot \beta_u^{-1} \cdot s_u \cdot I_u\right] \times H(\boldsymbol{\gamma_2}) \\ &= \alpha \times \left[(s_u^2 \cdot \lambda^2) \cdot I_u \cdot H(\boldsymbol{\gamma_2})\right] \end{aligned} \tag{4.38}$$

The authentication takes two-folds in our design. The reader decides the result of each phase based on whether the secret-sharing polynomials have established two pre-determined equalities, one for each phase. If the smart tag passes two phases, then the reader will grant it access; otherwise, reject its request. The two-fold operation is described as the following:

The First Phase of Authentication

The first phase in authentication will check if the smart card provides correct data in order to hold this equality[1]:

$$\begin{aligned}
&\hat{C}(\boldsymbol{\gamma_2}) \times \underline{R_1} - \underline{(R_2'' \times R_2 + R_2')} \times D(\boldsymbol{\gamma_2}) + \underline{R_3} \times E(\boldsymbol{\gamma_2}) \\
=&\alpha \times \left[C(\boldsymbol{\gamma_2}) \cdot \underline{\lambda \cdot s_u \cdot A(\boldsymbol{\gamma_2}, u)}\right] - \underline{\alpha} \times \left[\underline{\lambda \cdot s_u \cdot B(\boldsymbol{\gamma_2}, \boldsymbol{u})} \cdot D(\boldsymbol{\gamma_2})\right] + \underline{\alpha} \times \left[\underline{\lambda \cdot s_u} \cdot E(\boldsymbol{\gamma_2})\right] \\
=&\alpha \times \left\{\underline{\lambda \cdot s_u} \cdot \left[\underline{A(\boldsymbol{\gamma_2}, u)} \cdot C(\boldsymbol{\gamma_2}) - \underline{B(\boldsymbol{\gamma_2}, u)} \cdot D(\boldsymbol{\gamma_2}) + \underline{E(\boldsymbol{\gamma_2})}\right]\right\} \\
=&\alpha \times (\lambda \cdot s_u \cdot 0) \qquad \text{(as of Equation 4.7)} \\
=&0
\end{aligned}$$

The Second Phase of Authentication

In order to defend against linear collusive attacks, we will need an extra process to confirm the validity of the tag owners. A valid tag owns correct data and will pass the second phase by making the following equality:

$$\begin{aligned}
&\underline{(R_2'' \times R_2 + R_2') \times R_1} + \underline{(R_4' \times R_4)} \times J - \underline{(R_5' \times R_5)} \times H(\gamma_2) \\
=&\alpha \times \underline{(s_u^2 \cdot \lambda^2)} \cdot \left[\underline{A(\boldsymbol{\gamma_2}, u) \cdot B(\boldsymbol{\gamma_2}, u)} + \underline{G(\boldsymbol{\gamma_2}, u)} \cdot J + H(\boldsymbol{\gamma_2}) \cdot \underline{I_u}\right] \\
=&\alpha \times (s_u^2 \cdot \lambda^2) \cdot 0 \qquad \text{(as of Equation 4.16)} \\
=&0
\end{aligned}$$

4.3 Security and Privacy Analysis

We now analyzed the security effectiveness of the proposed scheme according to the attacks presented in Section 3.2. The analysis shows that the system is secured against many types of impersonation attacks mainly due to the hardness of ECDLP.

[1]The underlined parameters are those transmitted from the tag side, and the bolded symbols are the variables distinctly chosen at random for each session.

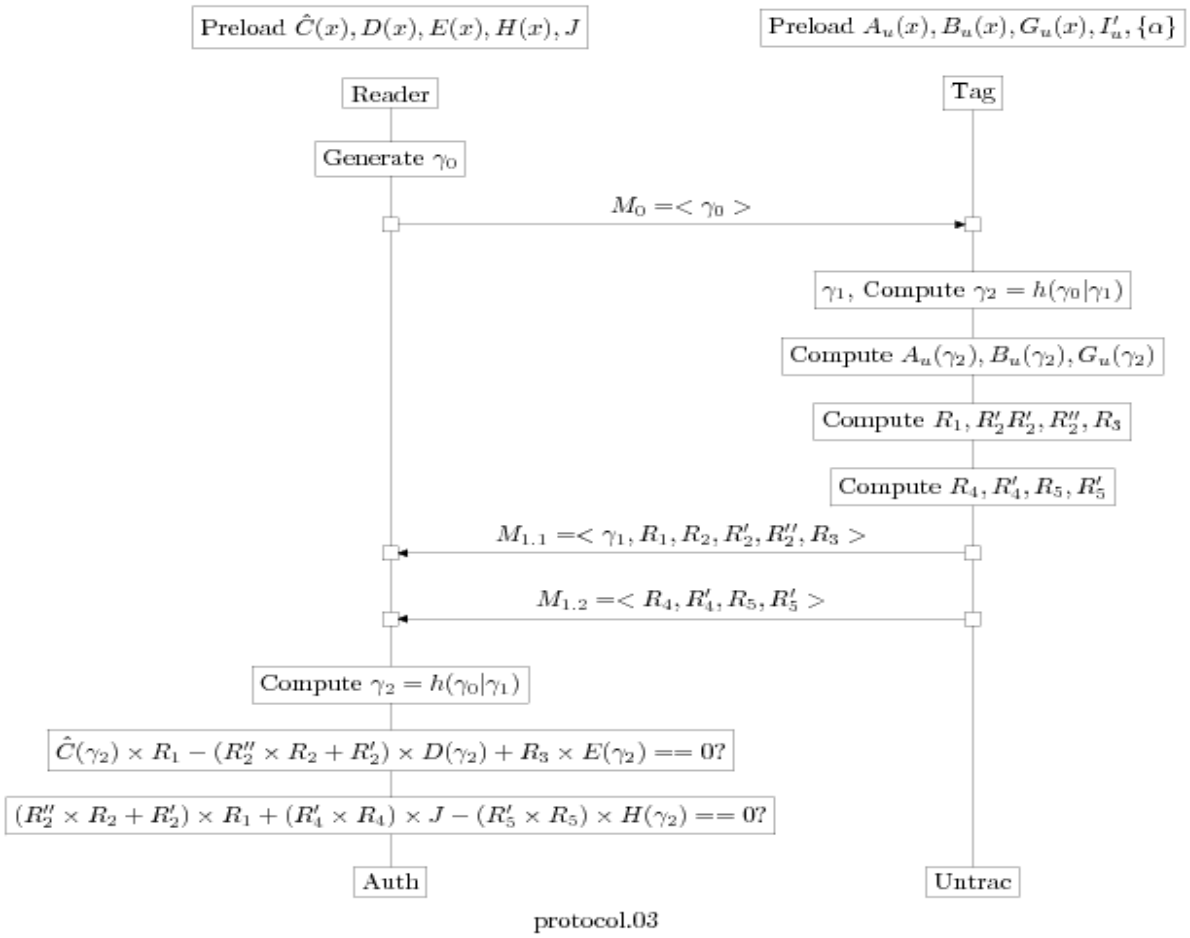


Figure 4.1 Authentication Protocol

4.3.1 Resistance against Tag-Compromised Attack

Imagine an attack scenario in which the attacker has compromised k tags as well as 3 pieces of information extracted from secret equations in each tag. The attacker may attempt to impersonate as the owner of the tags. This problem is preventable by our revocation function described in next Section 4.4. On the other hands, this attacker can attempt to bypass authentication by imitating some other valid smart tags. Nevertheless, this privacy-preserving protocol can successfully defend against k-tag compromised attack.

Let the compromised tags be labeled as $T_1, \ldots T_k$. The adversary is able to extract $<$

$R_1, R_2', R_2'', R_3 >$ from each compromised tag. Due to the computational impossibility of ECDLP, it is computationally difficult to infer the data protected by the elliptic curve point, α, as long as the order of the elliptic curve is large enough. Based on the remaining information, the compromised equations are presented as follows:

$$\text{Tag 1} \begin{cases} A_{u_1,1} = & s_{u_1} \cdot (a_{1,1}u_1 + a_{1,0}) \\ A_{u_1,0} = & s_{u_1} \cdot (a_{0,1}u_1 + a_{0,0}) \\ b_{u_1} = & s_{u_1} \cdot (b_{1,1}u_1 + b_{1,0}) \cdot (\beta_{u_1})^{-1} \end{cases}$$

$$\vdots$$

$$\text{Tag k} \begin{cases} A_{u_k,1} = & s_{u_k} \cdot (a_{1,1}u_k + a_{1,0}) \\ A_{u_k,0} = & s_{u_k} \cdot (a_{0,1}u_k + a_{0,0}) \\ b_{u_k} = & s_{u_k} \cdot (b_{1,1}u_k + b_{1,0}) \cdot (\beta_{u_k})^{-1} \end{cases}$$

There are 3k equations obtained from k tags; however, the unknown variables are $\underbrace{\{u_1, \ldots, u_k, s_{u_1}, \ldots, s_{u_k}, \beta_{u_1}, \ldots, \beta_{u_k}, a_{00}, a_{01}, a_{10}, a_{11}, b_{00}, b_{01}, b_{10}, b_{11}\}}_{(3k+8) \text{ unknowns}}$. Further, it is impossible to solve a system of linear equations when the number of unknown variables is greater than the number of the equation. Hence, it is unachievable to obtain a unique solution for the secret coefficients $a_{1,1}, a_{1,0}, a_{0,1}, a_{0,0}, b_{1,1}, b_{1,0}, b_{0,1}$ and $b_{0,0}$, which need to mimic a valid smart tag and successfully pass the security checkpoint.

The analysis shows that the attacker cannot obtain any secret function from the compromised smart tags. Therefore, the attacker cannot possible launch impersonation on a valid smart tags. For this reason, the system is secure.

4.3.2 Resistance against Eavesdropping Attack

Let us consider another form of practical attack. If an eavesdropper carries a malicious reader that could choose the system settings of his or her own flavor. In this attack scenario, the attacker could have been tailing and monitoring the same tag for k successful sessions, denoted as $Session^{(1)}, \ldots, Session^{(k)}$. Consequently, the number of the session-based variables involved in communication has now transformed into the target to break the system. Let us assume

that the overheard communication is associated with smart tags of a user u. In that case,

$$\text{Session 1} \begin{cases} R_1^{(1)} = \lambda^{(1)} \cdot A_u(x) \\ R_2^{(1)} = \lambda^{(1)} \cdot (\lambda'^{(1)})^{-1} \cdot B_u(x) \end{cases}$$

$$\vdots$$

$$\text{Session k} \begin{cases} R_1^{(k)} = \lambda^{(k)} \cdot A_u(x) \\ R_2^{(k)} = \lambda^{(k)} \cdot (\lambda'^{(k)})^{-1} \cdot B_u(x) \end{cases}$$

Due to the hardness of ECDLP, the attacker could not computationally possible acquire information from R_2', R_2'', R_3, R_4' and R_5' assuming the elliptic curve is large enough. Thus, the only useful piece of information is R_1, R_2, so that it makes 2k equations, and the unknowns now change to: $\underbrace{\{\lambda^{(1)}, \ldots, \lambda^{(k)}, \lambda'^{(1)} \ldots \lambda'^{(k)}, A_{u,1}, A_{u,0}, B_{u,1}, B_{u,0}, u, \}}_{2k+5 \text{ unknowns}}$. From the analysis above, we could also observe that it is unlikely for eavesdropper to receive private information given finite amount of time. Thus, the attacker is incapable of cloning a true identity. This concludes that the system has adapted to defend against eavesdropping attacks.

4.3.3 Resistance against Linear Collusive Attack

In this attack scenario, an attacker may have compromised two tags (T_1 and T_2) or two misbehaving employees who own tags, T_1 and T_2, may have agreed on a complicity. As a result, their hidden secrets, u_1 and u_2, are exposed. If the attacker finds two integers, w_1 and w_2 , such that $w_1 + w_2 = 1$ (by the mean of linearity from the system of linear equations), he or she is able to break into the system successfully by colluding information from stolen secrets. The reason is shown as the following:

Assume the attack creates $R_1 = [w_1 \cdot A(\boldsymbol{\gamma_2}, u_1) + w_2 \cdot A(\boldsymbol{\gamma_2}, u_2)]$, $R_2 = [w_1 \cdot B(\boldsymbol{\gamma_2}, u_1) + w_2 \cdot B(\boldsymbol{\gamma_2}, u_2)]$ by colluding the private polynomials from tags T_1 and T_2. At the first phase of authentication, the reader will evaluate the following formula to determine the validity of this

suspicious user.

$$
\begin{aligned}
&[w_1 \cdot A(\boldsymbol{\gamma_2}, u_1) + w_2 \cdot A(\boldsymbol{\gamma_2}, u_2)]C(\boldsymbol{\gamma_2}) - [w_1 \cdot B(\boldsymbol{\gamma_2}, u_1) + w_2 \cdot B(\boldsymbol{\gamma_2}, u_2)] \times D(\boldsymbol{\gamma_2}) + E(\boldsymbol{\gamma_2}) \\
=&w_1 \times [A(\boldsymbol{\gamma_2}, u_1) \times C(\boldsymbol{\gamma_2}) - B(\boldsymbol{\gamma_2}, u_1) \times D(\boldsymbol{\gamma_2})] \\
+&w_2 \times [A(\boldsymbol{\gamma_2}, u_2) \times C(\boldsymbol{\gamma_2}) - B(\boldsymbol{\gamma_2}, u_2) \times D(\boldsymbol{\gamma_2})] + E(\boldsymbol{\gamma_2}) \\
=&w_1 \times (-E(\boldsymbol{\gamma_2})) + w_2 \times (-E(\boldsymbol{\gamma_2})) + E(\boldsymbol{\gamma_2}) \\
=&(w_1 + w_2) \times (-E(\boldsymbol{\gamma_2})) + E(\boldsymbol{\gamma_2}) \\
=&- E(\boldsymbol{\gamma_2}) + E(\boldsymbol{\gamma_2}) \\
=&0
\end{aligned}
$$

This strategy can only pass the first phase of authentication process but it cannot survive second phase in the authentication process as the equality in the second phase would not hold. More precisely, if the attacker attempts to collude the private information from both tags, the reader will check the following formula and determine the result of authentication:

$$
\begin{aligned}
&[w_1 \cdot A(\boldsymbol{\gamma_2}, u_1) + w_2 \cdot A(\boldsymbol{\gamma_2}, u_2)] \cdot [w_1 \cdot B(\boldsymbol{\gamma_2}, u_1) + w_2 \cdot B(\boldsymbol{\gamma_2}, u_2)] \\
+&[w_1 \cdot G(\boldsymbol{\gamma_2}, u_1) + w_2 \cdot G(\boldsymbol{\gamma_2}, u_2)] \cdot J - H(\boldsymbol{\gamma_2}) \cdot (w_1 \cdot I_1 + w_2 \cdot I_2) \\
=&[w_1 \cdot w_1 \cdot A(\boldsymbol{\gamma_2}, u_1) \cdot B(\boldsymbol{\gamma_2}, u_1) + w_1 \cdot w_2 \cdot A(\boldsymbol{\gamma_2}, u_1) \cdot B'(\boldsymbol{\gamma_2}, u_2) \\
+&w_2 \cdot w_1 \cdot A(\boldsymbol{\gamma_2}, u_2) \cdot B(\boldsymbol{\gamma_2}, u_1) + w_2 \cdot w_2 \cdot A(\boldsymbol{\gamma_2}, u_2) \cdot B'(\boldsymbol{\gamma_2}, u_2)] \\
+&w_1 \cdot [G(\boldsymbol{\gamma_2}, u_1) \cdot J - H(\boldsymbol{\gamma_2}) \cdot I_1] + w_2 \cdot [G(\boldsymbol{\gamma_2}, u_2) \cdot J - H(\boldsymbol{\gamma_2}) \cdot I_2] \\
\neq&w_1 \cdot [A(\boldsymbol{\gamma_2}, u_1) \cdot B(\boldsymbol{\gamma_2}, u_1) + G(\boldsymbol{\gamma_2}, u_1) \cdot J - H(\boldsymbol{\gamma_2}) \cdot I_1] \\
+&w_2 \cdot [A(\boldsymbol{\gamma_2}, u_2) \cdot B(\boldsymbol{\gamma_2}, u_2) + G(\boldsymbol{\gamma_2}, u_2) \cdot J - H(\boldsymbol{\gamma_2}) \cdot I_2] \\
=&w_1 \cdot 0 + w_2 \cdot 0 \\
=&0
\end{aligned}
$$

The analysis shows that this attempt will fail at the second phase procedure resulting an invalid authentication. Hence, the system is able to defend attacks when two smart tags collude.

4.3.4 Resistance against Brute Force Attack

If an adversary attempts to exhaustively brute-force search the data $R_1, R_2, R_2', R_2'', R_3, R_4$, R_4', R_5 and R_5' which transmitted from a user's smart tag, he or her may find himself or herself to be in a desperate situation. First, the data protected by an elliptic curve point is known to be computationally irreversible, and it would take a brute-force attacker to search in an enormous set (as large as $O(2^{ord})$) of points on the curve. Furthermore, the attack attempts to guess the secret function but would not succeed. Each secret function is given an input , γ_2, and a number of tag-unique coefficients, which are all 160-bit data. More desperately, the bit length of all these numbers depends on the size of the finite field, which is around $O(2^{ord}) = O(2^{160})$ in our implementation.

4.3.5 Resistance against Replay Attack

Another common approach from an attacker is to overhear the communication, and replay the authentication data, which he or she has obtained in the past sessions. Nevertheless, this attack will not be effective, because γ_2 is defined to be session-based variable that keeps changing overtime. Also, the session nonce is protected via irreversible hash function, $\gamma_2 = h(\gamma_0|\gamma_1)$. Hence, the attacker would found it extremely hard to locate two duplicate sessions during a period of time.

4.3.6 Intraceability of User Privacy

In the proposed protocol, before the information was exchanged with the reader, the private information of the smart tag is obfuscated via randomization technique and under protection of the elliptic curve points. Hence, it would be impossible for either malicious outside attackers or unethical insiders to infer the identity of a tag from the authentication procedure.

In particular, even if the tag is interrogated via a fraudulent reader, each variable $\gamma_2, \lambda, \lambda', \lambda''$ and λ''' are randomly chosen in each session so that each response from the tag is changing overtime. Without the acknowledgment of user's secrets, the attacker is unable to gain user's identity since the tag's responses are indistinguishable in different authentication sessions.

Similarly, the user's movement pattern is also well protected due to to indistinguishability of the same smart card authenticated in different sessions. Thereupon, the proposed authentication scheme promises the protection not only for the privacy of the smart tag carrier (i.e. user's identity) and their location privacy (i.e. movement patterns).

4.4 Revocation Protocol

From the previous section, we have seen the system prevents unauthorized access; however, we still need a mechanism to trace out the IDs of the smart tags by trusted enforcements (i.e. police) when necessary.

Consider the following scenario: When a smart card is stolen, the owner reports the theft to the trusted law enforcement, who owns a complete record of secrets installed on every smart card. Thus, it is easy for the trusted party to find out the ID of stolen smart tag, and disseminate corresponding secrets (Identity is NOT included.) to RFID readers. Thus, the RFID reader is able to deny access, when the carrier of the stolen tag comes in.

In the following paragraphs, we described the method for the readers to identify the revoked smart cards in the beginning of the authentication procedure. Specifically, the reader has a list of revoked tag records and each of which has fields of its secret pair $< \hat{A}_i(x), \alpha_{i,0} >$. By intuition, each tuple $< \hat{A}_i(x), \quad \alpha_{i,0} >$ does not reveal its associated identity.

By observation, the easiest way to obtain λ is to derive it from R_1 in each tuple of the revocation records and test if the derived λ holds equality for R_3. If a matching record, i, exists in the revocation list, the following equality would hold:

$$
\begin{aligned}
R_1 &\equiv \lambda \cdot A_u(\gamma_2) \ (\mod ord) \\
R_3 &\equiv \lambda \times \alpha_{u,0} \\
\Rightarrow \lambda &\equiv R_1 \cdot \frac{1}{s_i \cdot A(\gamma_2, i)} \ (\mod ord) \\
\Rightarrow R_3 &\equiv \lambda \times \alpha_{i,0} \ (\mod ord)
\end{aligned}
$$

, where ord is the order of finite fields.

Given the information γ_2, R_1 and R_3, the trusted authority is able to find out the secret ID u associated with the smart tag. In other words, we are able to identify the owner of the smart card.

IsRevoked(γ_2, R_1, R_3)

1 **for** each tuple $< \bar{A}_i(\gamma_2), \alpha_{i,0} >$ in the revocation list

2 **do** $\lambda \equiv R_1 / \big[s_i \cdot A(\gamma_2, i)\big]$(mod ord)

3 **if** $R3 == \lambda \times \alpha_{i,0}$

4 **then return** TRUE

5 **return** FALSE

4.4.1 Traceability by the Trusted Third Party

When a revoked smart tag is reported stolen or suspicious for other illegal activities, a reader is able to deny access to any subsequent user for the tag. At the same time, the law enforcement is able to trace the identity of a particular smart card from the content of previously exchanged communications. For this reason, the reported misused smart tag can be managed properly and immediately reducing potential privacy damages to a minimum.

CHAPTER 5. Software Architecture

We implemented the TelosB sensor with NesC(8) code under TinyOS operating system(26)[1]. The cryptographic scheme is based on WM-ECC Library, contributed by Wang at Williarm-Mary College (29), as well as the Java-based ECC package of FlexiProvider(6). The authentication procedure and graphic user interface on base station (PC) are developed in Java. In our software implementation, we split the reader functions into two components: one processed the authenticated data for each incoming smart tag, and the other acted as a gateway sensor communicating between the RFID tag and the base station.

When a person enters a door, the reader will establish an connection with the user's smart tag, and attempt to verify user's access. There are two phases for authentication to take place, as the scheme is described in Chapter 4.2. Each time, the transmitted data is required to pass both phases of authentication in order to pass the security door. If it failed at either phase of the authentication procedure, his or her access will be denied. Figure 5.4 illustrates the step-by-step flowchart for the authentication process.

In order to help the audience to comprehend our design, we will decomposed the system in different modules in the following sections:

5.1 Module Decomposition

There are three modules in this system: user, gateway and base station. In our protocol, the user's smart tag needs to transmit a large amount of its sensitive data protected with elliptic curve points. In order to achieve this communication in real settings, not only we

[1]TinyOS is an open-source operating system designed for wireless embedded sensor networks. It features a component-based architecture with minimal code size as required by the severe memory constraints inherent in sensor networks

have adopted the optimized techniques, as explained later, but we transmitted the data in two consecutive packets. With all those techniques combined, the communication payload is reduced to a reasonable size.

5.1.1 User Module

This module, as shown in Figure 5.1, is installed in each smart tag carried by a user. When each tag is created, trusted party will install some tag-unique secrets in addition to confidential data, such as biometric data, personal private information, and etc. This module is also responsible to communicate with a gateway sensor on a channel. In the real world situation, when the user enters the security entrance, he or she will be authenticated based on the valid parameters that have been preloaded at installation. During the time when the message is exchanged wirelessly, out protocol guarantees not to leak out any private information to unknown third parties.

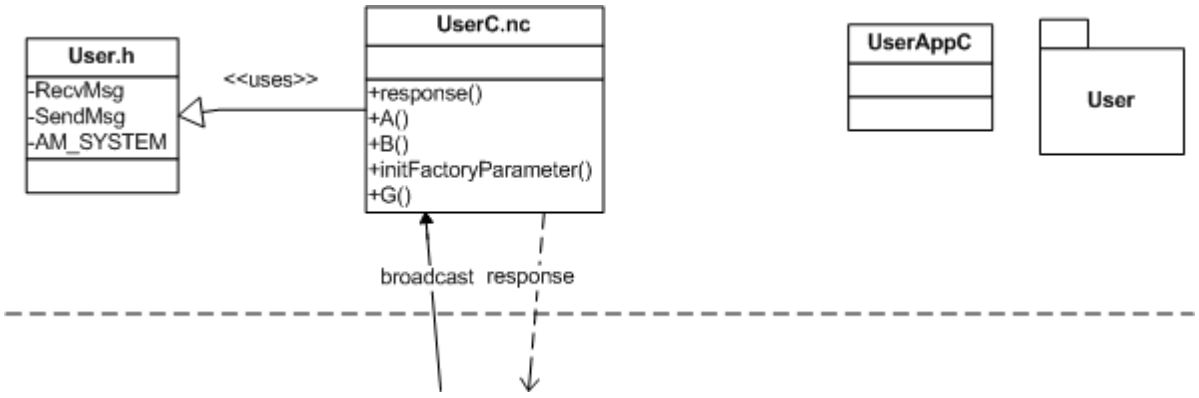


Figure 5.1 User Tag Module

5.1.2 Gateway Module

This module answers to all communications from user's smart tags to base station. When the base station first initiates a request, the gateway delivers this requests to an incoming smart tag. Soon after the tag responded back, the gateway passes its response back to the base station for authentication. In order to reduce the payload size at each session, this module

is also responsible to concatenate two random number, γ_0 and γ_1, generated by base station and user's smart tag respectively. Figure 5.2 illustrates this module.

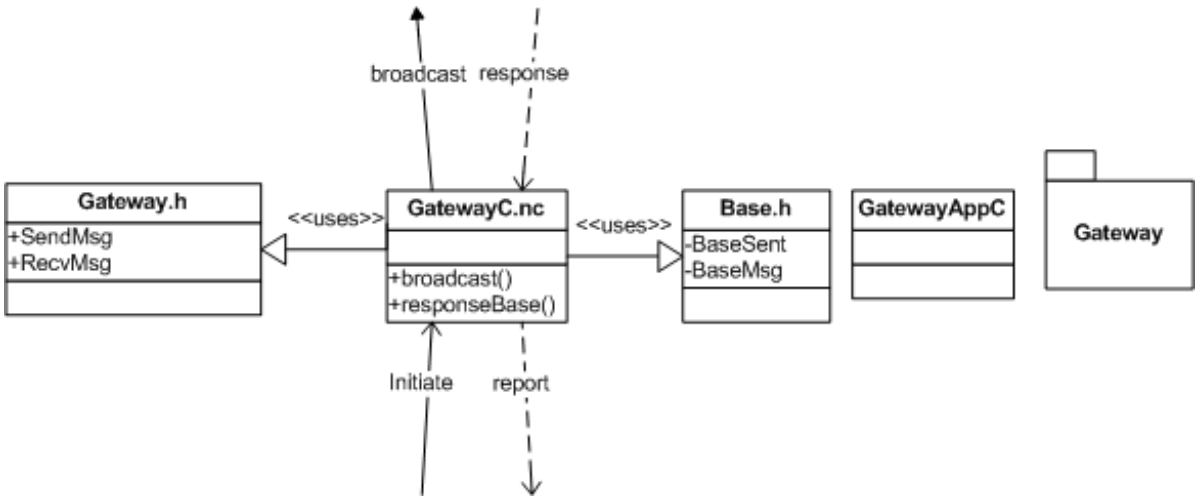


Figure 5.2 Gateway Module

5.1.3 Base Station Module

This module requires to run on a more computationally powerful device, e.g. a personal computer, and is required to be compatible with a Java compiler to run the graphic interface. Conceptually, this module can be divided in three parts, as illustrated in Figure 5.3. Some of which serve as core of communications for converting raw byte data into its corresponding Java objects. The authenticator module generates results of authentication. An interrogated smart tag and the other modules are toolkit modules that help the base station to perform statistical analysis and maintain packet synchronization.

- The sub-modules, Base and BaseMsg handle all communication tasks to the gateway module, including the generation of the first half of session key, γ_0, to initiate a communication and marshal the incoming NesC-compilable data from client to a Java-specific format.

- Authenticator module is responsible to determine the validity of an incoming user's tag and displayed the result onto the graphic user interface as the the second sub-module.

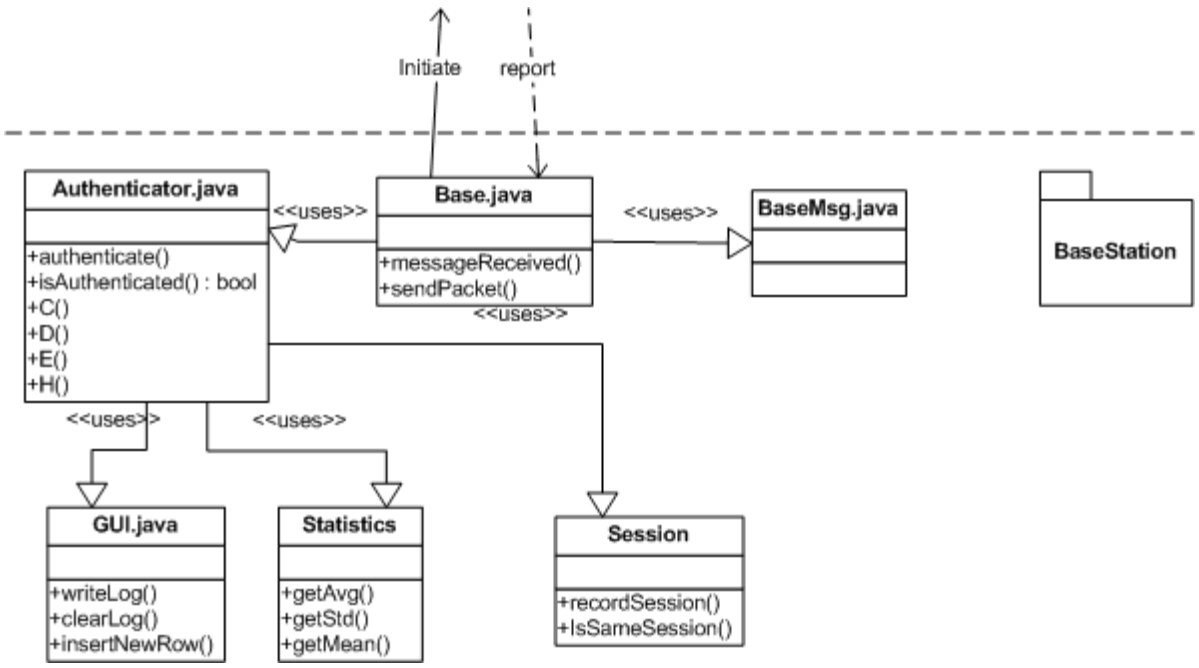


Figure 5.3 Base Station Module

- Statistics module is able to produce the average, median and standard deviation value for giving set of samples.
- Session module is able to provide a buffer for the base station and maintain the synchronization of incoming packets as well as a packet history.

5.1.4 Required Package

Here is the list of libraries that we have used in this project,

- WM-ECC

  We adapted the WM-ECC package [2] to perform operation on elliptic curve or big integer computation on user's smart card. In addition, hash function implemented in WM-ECC is SHA-1 algorithm from FIPS standards of NIST, and elliptic curve setup is is based on secp160r1 (see Appendix B) standards from SECG.

[2] WM-ECC is an Elliptic Curve Cryptography suite developed exclusively for wireless sensor motes. Official website: http://www.cs.wm.edu/∼wanghd/

- FlexiProvider (v. 1.6)

  FlexiProvider is a java-based cryptographic toolkit that provides full set of elliptic curve computation on personal computer as base station (part of reader competent).

- CoDec ASN.1 En/Decoder Library

  CoDec is a Java package for encoding and decoding ASN.1 data structures which are required to run FlexiProvider.

5.2 Optimization

Due to the resource limitation in smart tags, it is important to carefully adjust the strengths of the security in a way that provides reasonable protection while limiting overhead consumptions. There are two major concerns in the proposed protocol. The first problem is that the limited computational power on smart cards results in longer authentication time. The second concern is the amount of network traffic involved in data transmissions while preserving personal privacy. To solve the first problems, we reduced the amount of transmitted data via point compression technique and then we preloaded an enumeration of points in batches at tag-creating time to resolve the second problem.

5.2.1 Point Compression

In our authentication protocol, each smart tag needs to transmit several pieces of sensitive information wrapped with elliptic curve points. In reality, wireless sensor does not carry rich resources such as premium network bandwidths. Thus, it is desirable to represent those points in a minimum form when possible, which is usually referred as point compression.

Mathematically, in a full representation of point compressions, an affine point(x_a, y_a) requires 2n bits, where $n = \lceil log_2(p) \rceil$, assuming the prime field is $\mathbb{F}_p$. The compressed data is trivially reduced to n+1 bits by given the x-coordinate of a point plus an additional bit that is used to distinguish two different solutions ($\pm y$) of recovering the correct y-coordinate. Precisely, we needed to check for the least significant bit of the least significant coefficient of

y coordinate. In our proposed implementation, we used WM-ECC package, which adapted SECG

[3] secp160r1(20) recommended parameters to define our elliptic curve. In that standard, the size of an elliptic curve point was reduced from 320 bits to 160-bit when each elliptic curve point (P_x, P_y) consumes 160 bits for each coordinate. We will see more details on communication cost of our experiment in Section 6.1.3.

5.2.2 Point Pre-computation

In order to efficiently improve computation on tag side, we stored some pre-computed elliptic curve points in batch mode. These several elliptic curve points are served as guardians to protect sensitive data, as mentioned in our scheme. Our technical implementation is as follow:

We pre-installed an enumeration of points $< R'_2, R''_2, R_3, R'_4, R'_5 >$ in tables with respect to the values of $\lambda, \lambda', \lambda'', \lambda'''$. As a result, we have also observed that the RAM usage elevated while the number of points used increased. We can observe the result on storage consumption in Section 6.1.2.

[3] SECG is a consortium of companies formed to address potential interoperability problems with cryptographic standards. Current SECG2 policies specifies 15 NIST elliptic curve for several elliptic curve based standards, including ECDSA, ECIES, ECDH and etc.

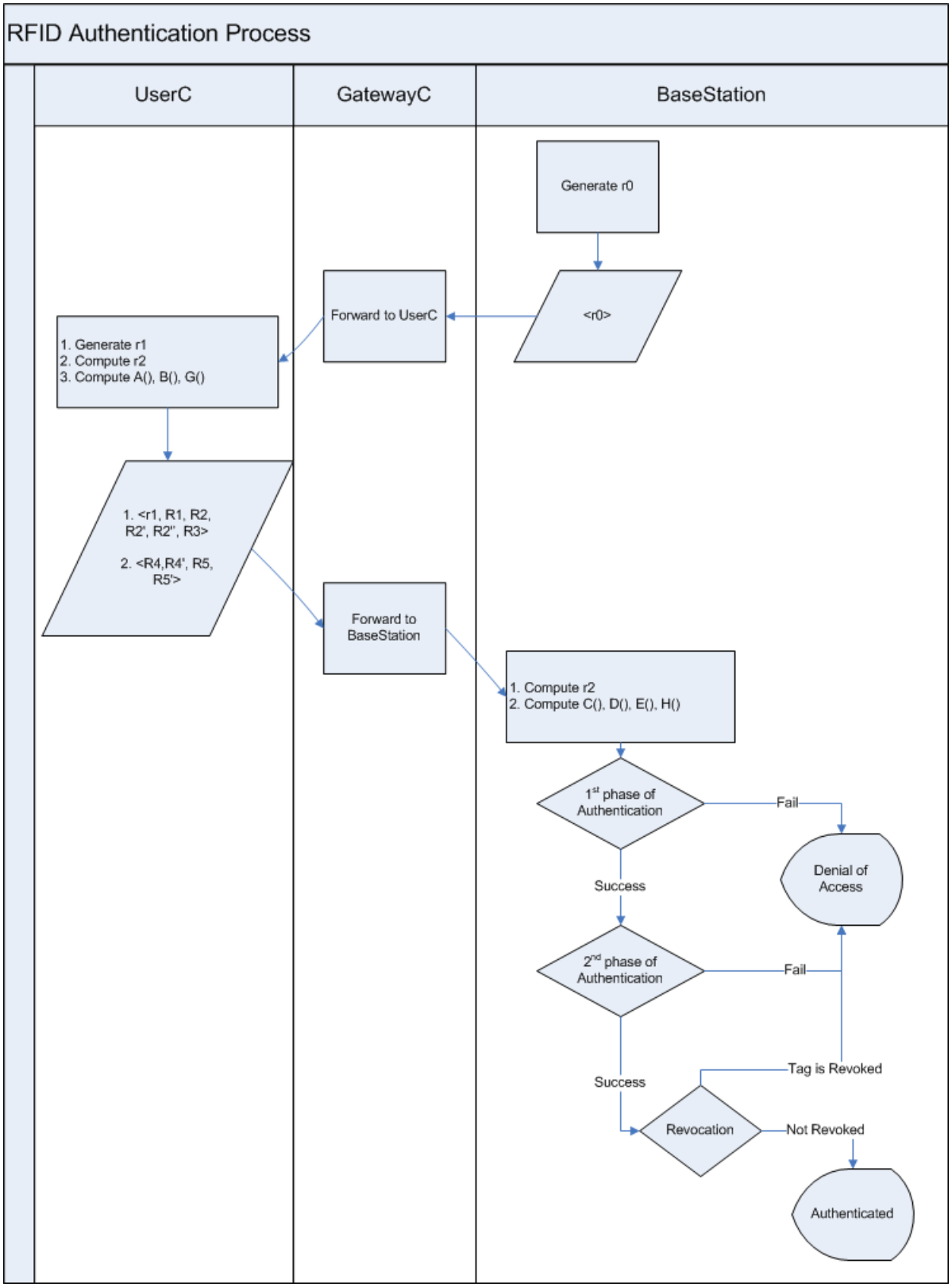


Figure 5.4 Flowchart of Authentication Process

CHAPTER 6. Experiments

As demonstrated in Chapter 5, we emulated an RFID reader as a gateway sensor attached to a base station. We performed our experiments on a duo-core processor with 1.6GHz Pentium(R) CPU and 1 gigabytes of RAM as base station, to which it was attached with a TI TelosB mote as a gateway. Each gateway sensor was responsible for communicating with a user's smart tag, also simulated by TelosB smart dusts. Each TI TelosB smart dust is equipped with a microprocessor running at 4 MHz, a memory of size 10 kilobytes and a flash storage of 1 megabytes.

6.1 Experimental Result

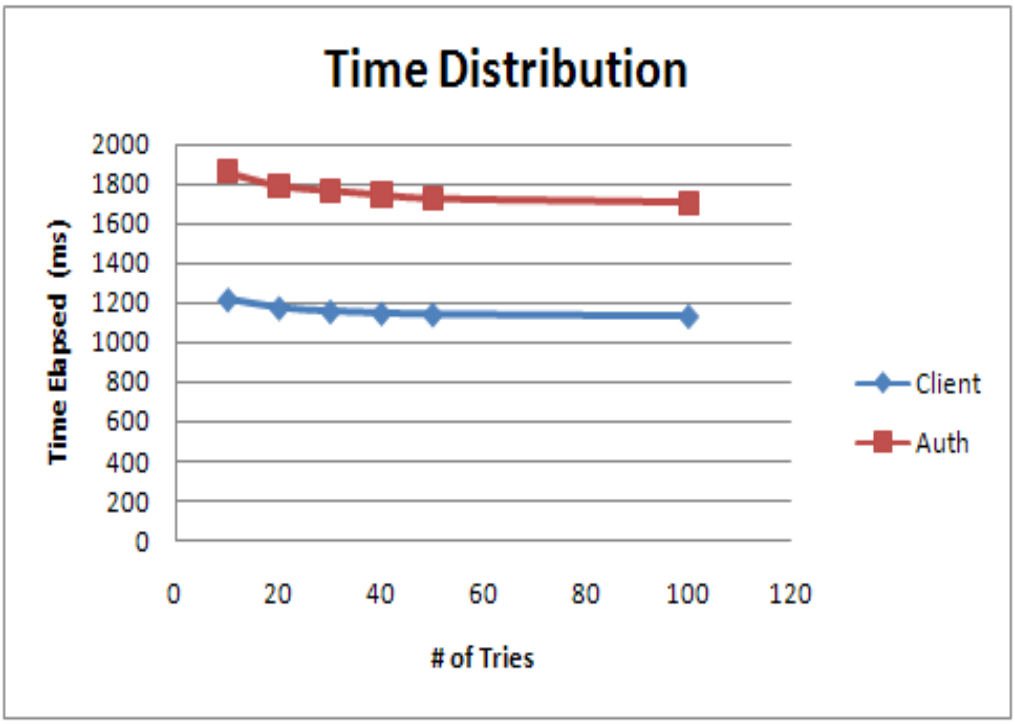
In this section, we evaluated the protocol proposed in three different aspects: how much time it takes to execute an authentication; how much traffic one transaction produces; and how much storage resource is required. Since smart cards and the smart dust on which we conducted our experiment are resource-limited devices, it is suffice to demonstrate that this protocol can survive in difficult resource constraints situations.

6.1.1 Computational Overhead

The most computationally expensive operations are the point addition and point multiplication - each takes around 100 milliseconds and 1450 milliseconds, respectively. In addition, taking an inverse of a big integer is also an expensive operation - roughly 50 millisecond for each inversion. In order to reduce the computation cost, we pre-computed an enumerated list of multiplication on point α and pre-loaded this list in the smart card at manufacturing time. When the RFID reader initiated a request, the smart tags picked a point in the list correspond-

ing to the parameters s_u and λ and responded back to the base station. In our experiment, we have measured that authentication took less than 2 seconds in general. As results shown in Figure 6.1, the total authentication time and client process time are 1.8 seconds and 1.2 seconds, respectively, as an average of up to 50 tries. Clearly, we showed a great reduction of client process time with adoption of the point pre-computation technique, as compared with the expensive time cost for each multiplication and inversion involved in the equation.

Figure 6.1 Time Distribution



Further, we analyzed the effect on authentication time based on different set of coefficients selected. As Figure 6.2 has shown, we found out that there was only a small time increase between the set of the most simplified coefficients (as specified in Section 4.2.1) and the set of huge numbers (as large as 9743344248872686121357898884775220131039550286 5) that were chosen as coefficients in each function. This observation told us the tag manufacturer has a higher scale of coefficient selection for each distinct tag. However, for security reasons, larger numbers are highly preferred because a set of small coefficients could make it easier for an attacker to break the system (i.e. brute force attack).

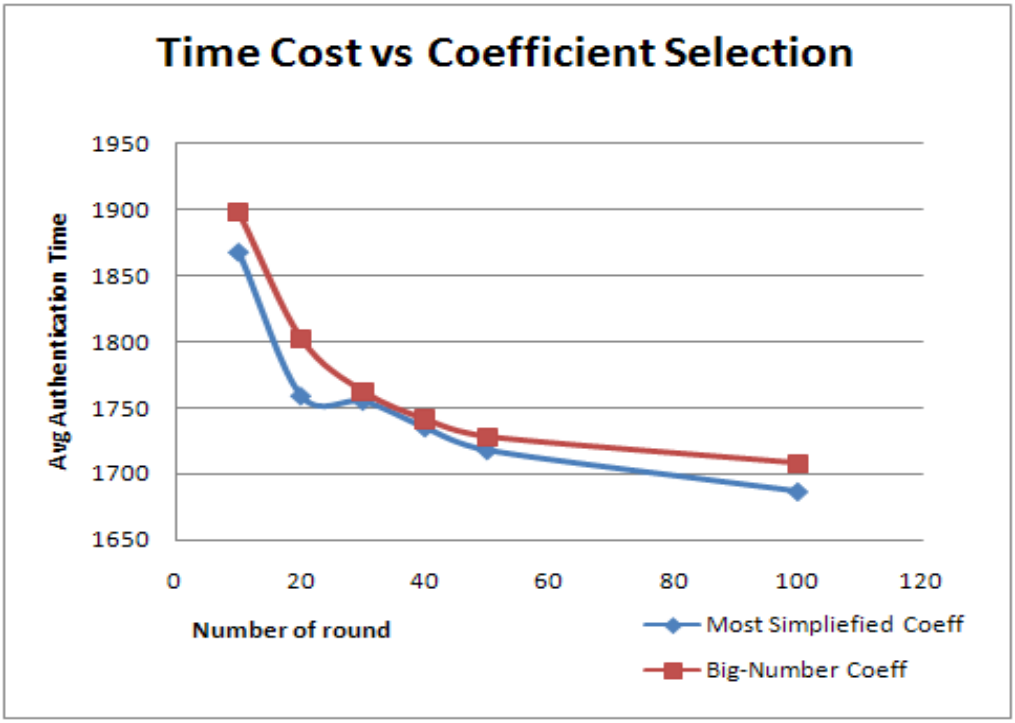


Figure 6.2 Time Cost vs Coefficients Selection

As the number of preloaded points increased, smart card computation would have a broader range for point selections with respect to each session-generated λ values. This phenomenon indirectly increases the security strengths, simply because the attacker needs to spend much more time in guessing the variables. Thus, the security of the system grows stronger when number of λ choices increases. However, there is a potential leak of λ values when an attacker has compromised a tag as well as the preloaded λ list in the memory stack. We analyzed security strength of the system based on the probability that an attacker could compromise all λ's in each session. For different sessions, we picked 4 λ's at random, which were selected out of the preloaded pool of size of 32. Mathematically, there are $\binom{32}{4} = 24800$ combinations of λ selections. Under the assumption that the attacker has no acknowledgment with regard to the preloaded information, the probability for an attacker to compromise (or guess) all 4 λ's is $\frac{1}{24800} \cdot 100\% = 0.004\%$. In other words, the probability that the system is able to defend such intrusion is 99.996%. In reality, an attacker still can brute-force search for the correct combination of λ's, if he or she has powerful computational device as well as sufficient

amount of time. For this reason, we perform the following analysis to observe the influence on the probability for an attacker to compromise λ's with respect to the increasing number of λ-combinations chosen from a fixed-number of preloaded list. The optimal solution is to maximize the number of λ's involved in the system. By intuition, the number is $\binom{32}{16} = 601080390$ of combinations. However, due to the memory limitation, we were not able to perform full-scale analysis for the list of size of 32, but instead we showed such influences for size of 16. The analysis is presented with respect to the number of λ-combination and probability of an attacker to break into the system, as shown in Figure 6.3. The figure shows that when more λs are involved in the system, the probability of an attacker to compromise all λs dramatically decreases due to a radical progression in the number of λ-combinations. Despite the rapid increase in security strength, Figure 6.4 shows that the growth of time cost still stays in the order of linear dependency. This implies that a small time increase as trade-off leads to a stronger security for the proposed protocol.

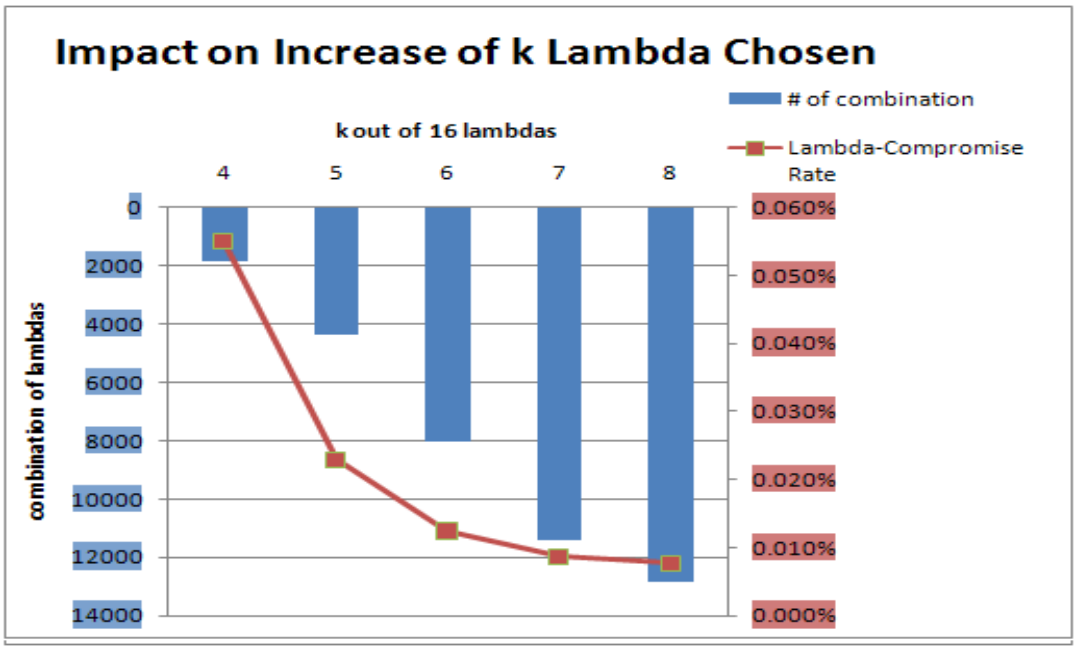


Figure 6.3 Number of Combinations, λ-compromised rate vs $\binom{16}{k}$ chosen λ

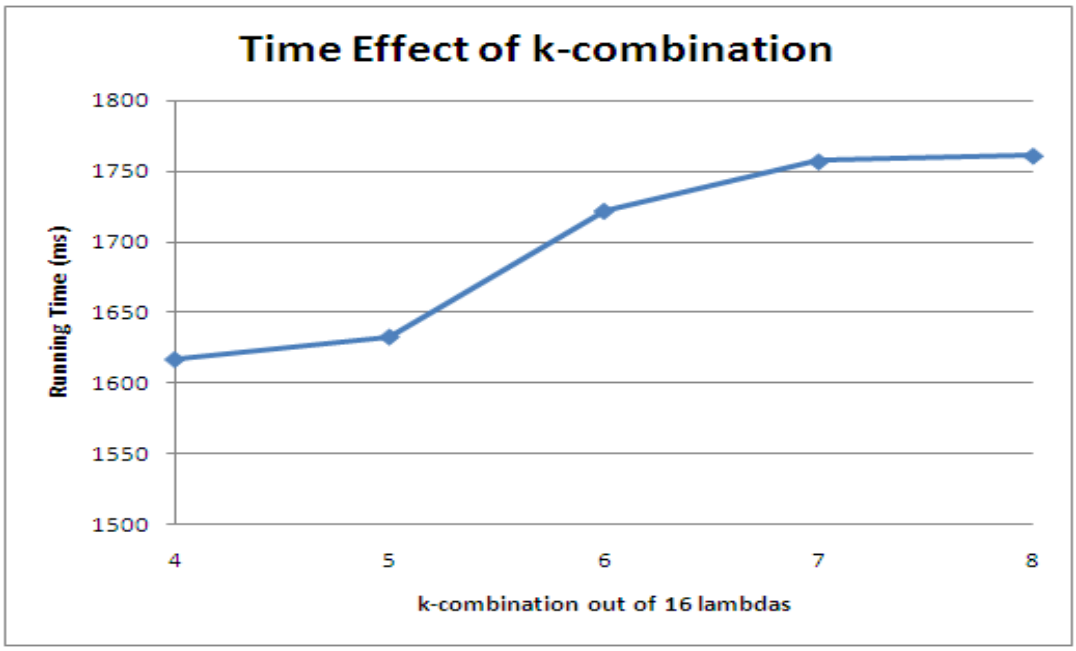


Figure 6.4 Time Cost vs Chosen Lambda

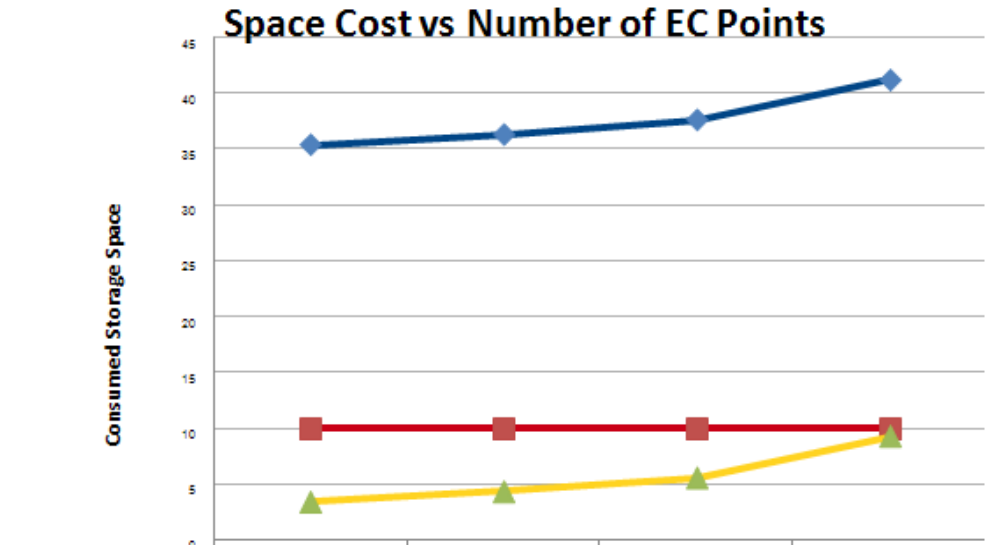
6.1.2 Storage Consumption

Like we discussed in Section 5.2.2, we preloaded some lists of points for each tag at the time when the tag is made. We compared the storage space needed when the size of the lists increased. In the Figure 6.5 below, we showed that memory usage in RAM and ROM required increases with the respect to the growth of the list of points. By observation, telosB mote is capable of holding roughly 100 pre-computated EC points. Furthermore, the space complexity indicates an approximately linear state of the execution time on the parameters chosen as expected. The red line indicates the maximum RAM size on telosbB sensor mote - approximately 10 kilobytes.

Type	RAM	ROM
Gateway	2506	26158
User's Tag	9262	41238

Table 6.1 Memory Usage for Gateway Mote and Smart Tag Mote

Lastly, the Table 6.1 above measures the code size of a gateway sensor mote and a smart card sensor (around 100 points preloaded):



	6	24	48	96
ROM (Byte)	35386	36310	37642	41238
RAM Limit (B)	10000	10000	10000	10000
RAM (B)	3410	4334	5566	9262

Figure 6.5 Progression on ROM/RAM Consumption

6.1.3 Communication Overhead

Under WM-ECC specification, each element relies on the size of finite field on the elliptic curve, which occupies 160 bits or 20 bytes. Moreover, each elliptic curve point consisted of two elements of x and y coordinates consumes 320 bits or 40 bytes of memory in total. In order to reduce the bandwidth, we applied the point compression technique for each transmission packet that involves elliptic curve points in communication. Once the compressed data is retrieved on the reader side, it is immediately decompressed back to the original form for further authentication evaluation.

In our experiment, each session requires significantly large amount of data including five EC points $< R_2', R_2'', R_3, R_4', R_5' >$ and four 20-bit big integer$< R_1, R_2, R_4, R_5 >$. The total size of 180 bytes exceeds the maximum payload size of 128, as specified by IEEE 802.15.4 standard. Therefore, in our communication scheme, we transmit the encrypted data in two consecutive packet with each packet size reduced to 112 bytes atop. Once the base station receives two

fractions of data, it is able to assemble the accurate data, while maintaining synchronicity of packets of the same session. Therefore, the proposed scheme satisfied the constraint of low communication expense.

CHAPTER 7. Future Work and Conclusion

The debate around technology and privacy has been going on for many years. As technologies have become increasingly sophisticated for intercepting messages, the ability of other people to see what we are doing has endangered an individual's privacy in our society. Inevitably, users place high priority on privacy and security in every RFID application.

In this work, we have shown that our proposed approach protects tags' identity from numerous types of attacks. Thereafter, extensive experiments have been conducted to evaluate its impact on bandwidths, storage and authentication time while preserving the strength of security and privacy. Finally, the results show that its low resource consumption is indeed affordable for the devices with extreme resource limitations.

From a security standpoint, we presented the scheme protected by elliptic curve points while maintaining the authentication service. The security of an elliptic curve is one of the most promising methods for security professionals. The Certicom challenge[1] began in 1997. The most current record is 109-bit solved in 2002 and 2004. At the present time, it is believed that current computer equipment is infeasible to break an elliptic-curve-based system with key size over 160 bits.

There are several potential research directions that could emerge from this project. The proposed scheme requires smart tags to provide some storage capacity to enumerate a list of elliptic curve points $R_2', R_2'', R_3, R_4', R_5'$ and associated λ values at the time when they were created. The pre-computation technique has shown its speedup to process time on user tag; however, the storage capacity is still limited on current technology. An algorithm that saves storage space would make improvements to the current scheme. In the experiment, 100 points

[1]The Certicom ECC challenge: http://www.certicom.com/index.php/the-certicom-ecc-challenge

are roughly the maximum number of points that could be pre-loaded on a smart tag, simulated by a TI TelosB sensor mote. Further, our scheme could be applied to other applications such as mobile services, or vehicular network security.

APPENDIX A. Algebraic Theory

A.1 Group

An abelian group (G,*) consists of a set G with a binary operation $* : G \times G \rightarrow G$ satisfying the following properties:

- (Associativity): x*(y*z) = (x*y)*z, for all $x, y, z \in G$
- (Commutativity): x* y = y * x, for all $x, y \in G$
- (Identity): There exists an element $e \in G$, such that x*e = e*x = x, for all $x \in G$
- (Inverse): For each $x \in G$, there exists an element $y \in G$ to be the inverse of x, such that x * y = y * x = e

A group is called *additive group*, if the identity is usually denoted by 0, and the inverse of a is denoted by -a. A group is called *multiplicative group*, if the identity is usually denoted by 1, and the inverse of a is denoted by a^{-1}.

Subgroup

Let (G,*) be a group, a non-empty subset H of G is a subgroup, if $a, b \in G \Rightarrow a * b \in H$. This implies that $H \subseteq G$.

Cyclic Subgroup

Let $(\mathbb{Z}_n^*, *)$ be the group, and for each element $a \in G$, there always exists a cyclic subgroup, denoted as $< a >= \{e, a^1, a^2, \ldots, \}$ under a finite field $\mathbb{F}_n$. For example, $< 7 >= \{1, 7, 24, 18\} \subseteq 25^*$

Order of a Group

The order of a group G refers to the smallest positive integer k, for any element in G, such that $a^k = e$. The group is finite refers to the order of the of the system, in which there re finite number of elements in G.

In addition, the order of a group G is also equivalent to the number of element in the subgroup $ord_G(a) = | < a > |$. For example, $ord_G(7) = | < 7 > | = |\{1, 7, 24, 18\}| = 4$

APPENDIX B. Elliptic Curve in a Nutshell

We began this chapter with an introduction to elliptic curve on a finite field to other elliptic-curve-related cryptographic topics.

An elliptic curve $\mathcal{E}$ is a set of solutions to the equation of the form $y^2 = x^3 + ax + b$ (also known as Wierstrass equation), where $4a^3+27b^2 \neq 0$ (If $4a^3+27b^2 = 0$, then the corresponding elliptic curve is called a singular elliptic curve). All points $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ on the curve are associated with a finite field $\mathbb{F}_q$, including an extra point, $\mathcal{O}$, called a point of infinity. Similar to the idea of modular arithmetic, both elements in the coordinate are integers between 0 and p - 1, where p is the arithmetic modulo of $\mathbb{Z}_p$ (also known as a set of residues).

For example, an elliptic curve $\mathcal{E} : y^2 = x^3 - x - 3$ over a finite field $\mathbb{F}_{13}$ contains the points $\mathcal{E}(\mathbb{F}_{13}) = \{\mathcal{O}, (0, \pm 6), (1, \pm 6), (2, \pm 4), (5, 0), (6, \pm 5), (10, \pm 5), (11, \pm 2), (12, \pm 6)\}$

In order to ensure the security of a system, the arithmetic modulo q (usually a prime number) is chosen at large so that there is sufficient number of points on the elliptic curve to resist exhaustive attacks.

B.1 Properties of Elliptic Curve

Here are some algebraic properties of this elliptic curve group

- The order of a point P is the smallest integer ord, such that $ord \cdot P = \mathcal{O}$.
- In addition, the notation of $< P >$ refers to the subgroup of P, where $< P >= \{\mathcal{O}, P, 2P, 3P, \ldots, (ord - 1) \cdot P\}$

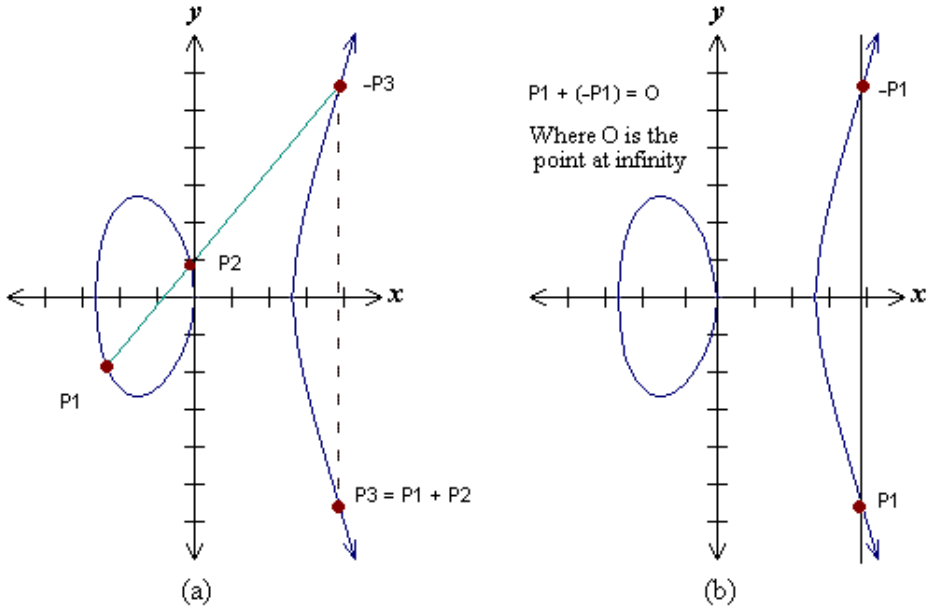


Figure B.1 Point Addition

Point Addition

Point addition is the addition of two points P_1 and P_2 on an elliptic curve $\mathcal{E}$ to obtain another point P_3 on the same elliptic curve. Literally speaking, f $P_1 \neq P_2$, then a line drawn through two points will intersect the curve $\mathcal{E}$ at exactly one more point, $-P_3$. The reflection of the point P_3 with respect to x-axis gives the point P_3, which is the result of $P_1 + P_2 = P_3$, as shown in Figure B.1(a). Considering the case that $P_1 = P_2$, as shown Figure B.1(b), by definition, we included one more point, point at infinity, denoted as $\mathcal{O}$. $\mathcal{O}$ is sitting at the top of the y-axis. Hence, $P_1 + (-P_1) = P_2 + (-P_2) = \mathcal{O}$

Mathematically, to calculate $P_3 = (x_3, y_3)$ by adding two points, $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$. Then

$$\begin{cases} x_3 &= m^2 - x_1 - x_2 \\ y_3 &= m \cdot (x_1 - x_3) - y_1 \end{cases}$$

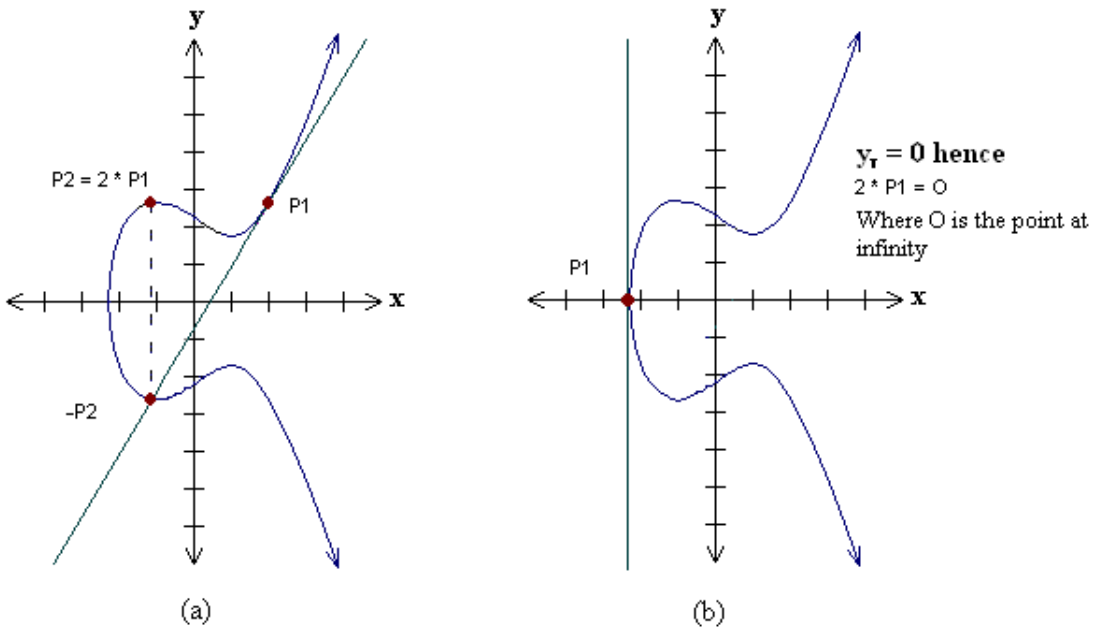


Figure B.2 Point Doubling

, where the slope m is defined as following:

$$m = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } P_1 \neq P_2 \\ \frac{3x_1^2 + a}{2y_1} & \text{if } P1 = P2 \end{cases}$$

Please note that if the slope m is infinite, then $P_1 + P_2 = \mathcal{O}$

Point Doubling

Point doubling is the addition of a point P_1 on the elliptic curve to itself to obtain another point P_2 on the same elliptic curve. To double a point P_1 to get P_2, i.e. $P_2 = P_1 + P_1$, consider a point P_1 on an elliptic curve as shown in Figure B.2. If y-coordinate of the point J is not zero, then the tangent line at J will intersect the elliptic curve at exactly one more point $-P_2$. The reflection of the point $-P_2$ with respect to x-axis gives the point P_2, which is the result of doubling the point P_1. Thus $P_2 = 2 \cdot P_1$.

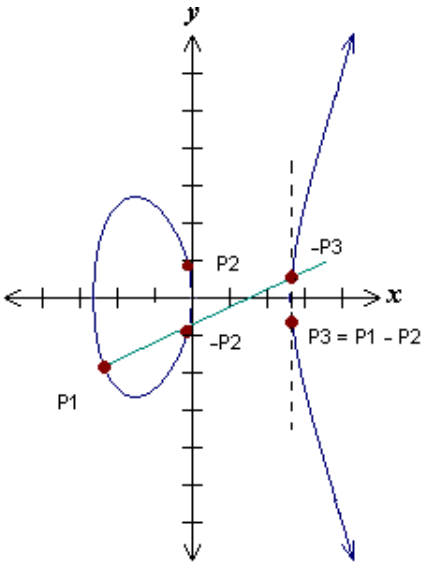


Figure B.3 Point Subtraction

Point Subtraction

Similar to point addition, except we take the reflection of second point to perform same addition operation. $P_1 - P_2 = P_1 + (-P_2) = P_3$. Figure B.3 shows an example of a point subtraction.

Point Multiplication

In point multiplication, a point P_1 on the elliptic curve is multiplied with a scalar k using an elliptic curve equation to obtain another point P_2 on the same elliptic curve. Point multiplication is achieved by two basic elliptic curve operations:

- Point addition: adding two points P_1 and P_2 to obtain another point P_3, i.e., $P_3 = P_1 + P_2$.
- Point doubling: adding a point P_1 to itself to obtain another point P_2, i.e., $P_2 = 2 \cdot P_1$.

, and follows the rule,

$$k \cdot P_1 = \begin{cases} P_1 & \text{if k} = 1 \\ \frac{k}{2} \cdot (P_1 + P_1) & \text{if k is even} \\ P_1 + (k-1) \cdot P_1 & \text{otherwise} \end{cases}$$

Therefore, to perform point multiplication over a point P with a scalar k is exactly completing point addition for k times.

B.1.1 SECG secp160r1 Elliptic Curve Parameter Setup

Here is the secp160r1 standard configuration for the elliptic curve $\mathcal{E}$, implemented in WM-ECC:

- Curve equation is defined as $y^2(\mod q) \equiv x^3 + a \cdot x + b(\mod q)$
- Finite field, $\mathbb{F}_q$, is defined as $q = 2^{160} - 2^{31}$.
- Curve order is $ord = q-1 = 2^{160}-2^{31}-1 = 1461501637330902918203684832716283019653785059327$.
- $\mathbb{Z}_p$ is predefined as large as $p = 1461501637330902918203687197606826779884643492439$
- Coefficient $a \in \mathbb{F}_q, a = 1461501637330902918203684832716283019653785059324$
- Coefficient $b \in \mathbb{F}_q, b = 163235791306168110546604919403271579530548345413$
- A base point P on the curve is selected:
  - x-coordinate $P_x = 425826231723888350446541592701409065913635568770$
  - y-coordinate $P_y = 203520114162904107873991457957346892027982641970$

Here is the time consumption for each elliptic curve and big integer operations performed by WM-ECC:

EC operation	Time (ms)	Big Integer Operation	Time (ms)
Addition (Ecc.padd)	107	Addiction	2
Fixed-Point Subtraction (Ecc.bsub)	1	Subtraction	2
Random-Point Multiplication (Ecc.gmul)	1450	Multiplication	3
Fixed-Point Multiplication (Ecc.bmul)	1420	Modular Reduction	6
Doubling (Ecc.pdbl)	107	Inversion	53

Table B.1 Time Comparison between Elliptic-Curve Operations vs Big Integer Operations

B.2 Elliptic-Curve Based Cryptography

B.2.1 Elliptic Curve Discrete Logarithm Problem

The well-known discrete logarithm problem (DLP) is a problem to determine the least positive integer, x (the discrete logarithm of a to base b), which would satisfy the equation:

$$a = b^x \tag{B.1}$$

for given two elements a and b in a multiplicative group G. In general reality, there is no efficient algorithm to solve a discrete logarithm in finite time. There are many other cryptographic protocols that are based on the hardness of DLP, such as the Diffie-Hellman key exchange protocol, ElGamal cryptosystem, and etc.

The cryptographic system in our proposed protocol uses the elliptic curve group $\mathcal{E}(\mathbb{F}_q)$ of rational points on an elliptic curve defined over some finite field F_q. Due to cyclic property of this additive group, it makes multiplication and exponentiation easy. The problem is defined as the following:

Let E be an elliptic curve over some finite field, $\mathbb{F}_q$ and *ord* denote the order of the group $\mathcal{E}(\mathbb{F}_q)$. Let P denote an element of $\mathcal{E}(\mathbb{F}_q)$ and a point Q within subgroup of P, $Q \in < P >$. The goal is to find an integer m, such that

$$Q = m \cdot P \tag{B.2}$$

m is also called as the discrete logarithm of Q to the base P. In the security point of view, this is also known as the elliptic curve discrete logarithm problem (ECDLP), where the integer m is selected uniformly as a private key and the point Q is its corresponding public key.

Security Strength

DLP in elliptic curve group is, in terms of order of magnitude, harder than any conventional problem in the group of finite field of a similar size. The conventional problem include public key cryptosystem cased on factorization, like RSA. Here is the plotted Figure B.4 that compares security strength, in term of key size in bits, between an elliptic curve cryptosystem and conventional public key system.

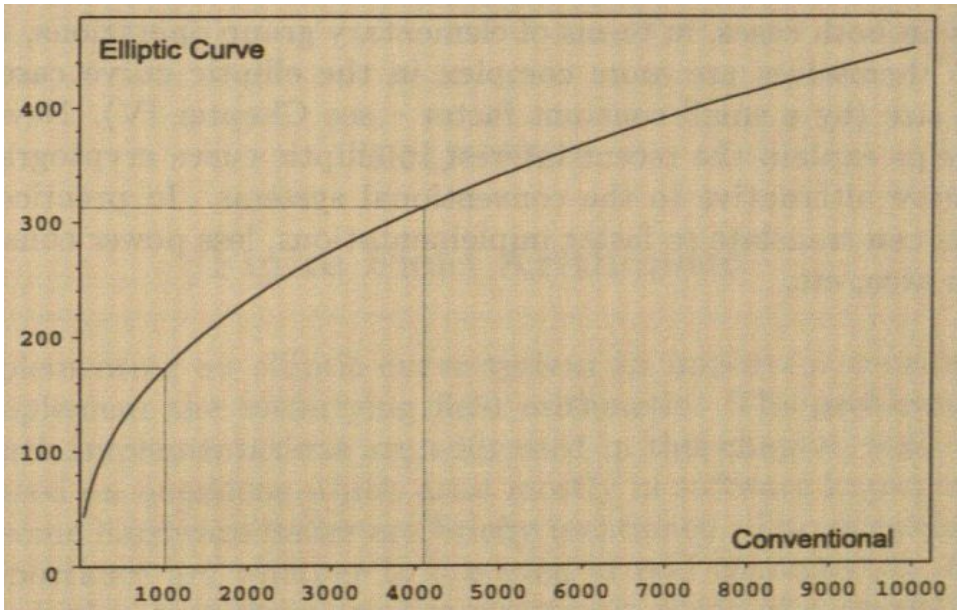


Figure B.4 Elliptic curve vs. conventional cryptosystem key sizes (in bits) for similar strength(1)

By observation, for the same level of security, an elliptic curve cryptographic system requires relatively smaller key in bits, compared to a conventional key size. For example, the dotted line shows that common RSA key size of 1024 and 4096 bits corresponds to the key sizes for an elliptic curve cryptography of 173 and 313 bits(1). In addition, the growth speed of key size in an elliptic curve cryptosystem increases a lot slower than the any traditional public key system for similar security strength. All of these information shows a solid evidence that, at present, the cryptosystem based hardness of ECDLP probably outperforms the old-school cryptographic problems, such as RSA (reduced from factorization problem), ElGamal (reduced from DLP), and etc.

Breaking the ECDLP

There are several known approaches to the solution of DLP. The biggest advantage of elliptic curve cryptosystem over those based on the DLP in finite fields is that for the former no general-case sub-exponential algorithms are known.

- **Brute-force** approach computes the sequence of points, P, 2P, 3P, ..., etc, until Q is hit.

The running time $O(ord)$ in the worst case, and $O(\frac{ord}{2})$ in the average case.

- **Pohlig-Hellman** method implies that solving any type of DLP in an abelian group is equivalent to solve the same DLP problem in subgroups of prime power order by appealing the Chinese Remainder Theorem. Therefore, an elliptic curve based security system requires the order of E to contain a larger prime divisor, so that finding the power of primes is hard.

- **Baby-Step Giant-Step** or **Shank's** method solves the ECDLP in any finite abelian group. This algorithm is also known to be an example of time-memory trade off, with time and storage complexity at $O(\sqrt{ord})$

- **Polard Rho** algorithm is also an efficient approach to solve an ordinary DLP at a time complexity of $O(\sqrt{ord})$, and is also the best known method to solve the ECDLP with an expected running time of $\frac{\sqrt{\pi \cdot ord}}{2} = 1.2533\sqrt{ord}$.

BIBLIOGRAPHY

[1] I. Blake, G. Seroussi, and N. P. Smart. *Elliptic Curves in Cryptography*. Cambridge University Press, 1999.

[2] T. I. Corp. Ti celebrates 10 year anniversary of rfid, http://www.ti.com/rfid/docs/manuals/rfidnews/tiris_nl20.pdf. RFID News, Issue 20, Texas Instruments., 2000.

[3] T. Dimitriou. A lightweight rfid protocol to protect against traceability and cloning attacks. September, 2005.

[4] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Wolkerstorfer. strong authentication for rfid systems using the aes algorithm. *CHES*, 2004.

[5] K. Finkenzeller. *Rfid handbook*. john wiley and sons, 1999.

[6] FlexiProvider. http://www.cdc.informatik.tu-darmstadt.de/flexiprovider/.

[7] Food and D. Administration. Combating counterfeit drugs: A report of the food and drug. Technical report, Administration Annual Update, May 18, 2005.

[8] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesc language: A holistic approach to networked embedded systems. *In Programming Language Design and Implementation (PLDI)*, June 2003.

[9] A. Juels. Rfid security and privacy: a research survey. *Selected Areas in Communications, IEEE Journal on*, Volume 24:p. 381–p. 394, 2006.

[10] A. Juels and T. Llc. Squealing euros: Privacy protection in rfid-enabled banknotes. In *Financial Cryptography 03*, pages 103–121. Springer-Verlag, 2003.

[11] A. Juels, D. Molnar, and D. Wagner. Security and privacy issues in e-passports. *IACR Cryptology ePrint Archive*, 2005.

[12] A. Juels, R. L. Rivest, and M. Szydlo. The blocker tag: selective blocking of rfid tags for consumer privacy. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 103–111, New York, NY, USA, 2003. ACM Press.

[13] A. Juels and S. Weis. Authenticating pervasive device with human protocols. *Crypto*, 2005.

[14] A. Juels and S. A. Weis. Defining strong privacy for rfid. Technical report, 2006.

[15] M. Jung, H. Fiedler, and R. Lerch. 8-bit microcontroller system with area efficient aes coprocessor for transponder applications. *Ecrypt Workshop on RFID and Lightweight Crypto*, 2005.

[16] K. Koscher, A. Juels, and T. Kohno. Epc rfid tags in security applications: Passport cards, enhanced drivers licenses, and beyond. 2008.

[17] L. S. Kulseng. Lightweight mutual authentication, owner transfer, and secure search protocols for rfid systems. Master's thesis, Iowa State University, 2009.

[18] D. Molnar and D. Wagner. Privacy and security in library rfid: Issues, practices, and architectures. In *Conference on Computer and Communications Security CCS '04,*, page pages 210219, 2004.

[19] K. Oua and S. Vaudenay. Smashing squash. 2007.

[20] C. Research. Recommended elliptic curve domain parameter. Technical report, Certicom Corp., September 20, 2005.

[21] M. R. Rieback, B. Crispo, and A. S. Tanenbaum. Rfid guardian: A battery-powered mobile device for rfid privacy management. pages 184–194. Springer-Verlag, 2005.

[22] S. Sarma, D. Brock, and K. Ashton. The networked physical world - proposals for engineering the next generation of computing, commerce & automatic identification. white paper, mit: Auto-id center, Oct 2000.

[23] A. Shamir. Squash - a new mac with provable security properties for highly constrained devices such as rfid tags. *Crypto*, 2007.

[24] A. Soppera, D. Molnar, and D. Wagner. Privacy for rfid through trusted computing. *Workshop on Privacy in the Electronic Society*, 2005.

[25] K. Suzuki, M. Ohkubo, and S. Kinoshita. Cryptographic approach to privacy-friendly tags. *RFID Privacy Workshop, MIT, USA*, 2003.

[26] TinyOS. http://www.tinyos.net.

[27] G. Tsudik. Ya-trap: yet another trivial rfid authentication protocol. 2006.

[28] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Security in Pervasive Computing*, volume 2802 of *Lecture Notes in Computer Science*, pages 201–212, 2004.

[29] WMECC. http://www.cs.wm.edu/∼wanghd/.