2012년 4월
박사학위 논문

년 월 박사학위논문 강인한 프라이버시를 제공하는 효율적인 RFID 인증 프로토콜 심 검

A Secure RFID Authentication Protocol in the Random Oracle Model with Low Computational Cost

조선대학교 대학원

컴퓨터공학과

심 검

A Secure RFID Authentication Protocol in the Random Oracle Model with Low Computational Cost

낮은 연산 비용을 갖는 랜덤 오라클 모델의 안전한 RFID 인증 프로토콜

2012년 8월 24일

조선대학교 대학원

컴퓨터공학과

심 검

A Secure RFID Authentication Protocol in the Random Oracle Model with Low Computational Cost

지도교수 정 일 용

이 논문을 공학박사학위신청 논문으로 제출함

2012년 04월

조선대학교 대학원

컴 퓨 터 공 학 과

심 검

심검의 박사학위논문을 인준함

위원장	조선대학교	교수	모 상 만	(인)
위 원	조선대학교	교수	신 석 주	(인)
위 원	조선대학교	교수	강 문 수	(인)
위 원	조선대학교	교수	문 인 규	(인)
위 원	조선대학교	교수	정 일 용	(인)

2012년 06월

조선대학교 대학원

Contents

List of Figures

List of Tables

초록

낮은 연산 비용을 갖는 랜덤 오라클 모델의 안전한 RFID 인증 프로토콜

심 검
지도교수 : 정 일 용
조선대학교 대학원 컴퓨터공학

무선 주파수 식별 (RFID)기술이 지속적으로 발전하고 성숙해 가는 것처럼, RFID 태그도 광범위한 응용에서 구현할 수 있다. 이러한 응용들에서는 RFID 태그와 리더가 무선 매체를 공유하기 때문에 공격자는 RFID 시스템에 대해 다양한 공격을 시도할 수 있다. 이런 다양한 공격들을 저지하기 위해 우리는 참여 태그들의 실제 아이디를 노출하지 않고도 인증작업이 가능하며 RFID 사용자에게 강인한 프라이버시와 보안 보호를 제공하는 랜덤 오라클 모델의 안전한 RFID 인증 프로토콜(SRAP)을 제안한다. SRAP는 태그의 불추적성 뿐만 아니라 익명성도 제공한다. 이 프로토콜은 또한 오늘 전송된 데이터가 미래에 발생하는 조작에 의해 비밀 태그의 정보가 누출되었을지라도 안전함을 보장하는 전방향 보안(전방향 프라이버시)를 제공한다. 본 논문에서 우리는 RFID 시스템의 인증과 프라이버시를 위한 정식 보안 모델을 정의한다. 이 모델 하에 우리는 인증과 프라이버시 속성을 대부분 만족하는 프로토콜인 SRAP를 설명한다. 여기에 더해, 제안된 SRAP는 인증을 수행하는 데 있어 극도로 자원이 제한된 저비용 RFID 태그에 적합한 매우 적은 자원을 사용한다. 특히, 연산 비용에 있어서, SRAP의 연산 비용과 요구 저장공간은 이전의 연구에서 보이는 것보다 적다.

ABSTRACT

A Secure RFID Authentication Protocol in the Random Oracle Model with Low Computational Cost

Jian Shen
Advisor : Prof. Il Yong Chung, Ph.D.
Department of Computer Engineering,
Graduate School of Chosun University

As the radio frequency identification (RFID) technology continues to evolve and mature, RFID tags can be implemented in a wide range of applications. Due to the shared wireless medium between the RFID reader and the RFID tags, adversaries can launch various attacks on the RFID system. To thwart different types of attacks, we propose a Secure RFID Authentication Protocol in Random Oracle Models (*SRAP*), which can accomplish the authentication without disclosing real IDs of the participating tags and provide strong privacy and security protection with RFID users. SRAP offers the anonymity of tags in addition to tag untraceability. It also provides forward security (forward privacy) which ensures that data transmitted today will still be secure even if secret tag information is revealed by tampering in the future. In this paper, we define a formal security model for authentication and privacy in RFID systems. Under this model, we describe the protocol that provably achieves the properties of authentication and privacy. In addition, the proposed SRAP requires only little resources to perform the authentication, which is suitable for highly resource-constrained low-cost RFID tags. Particularly, the computational cost, communication cost and storage requirement in SRAP are all less than those in the previous researches.

1. Introduction

Radio Frequency Identification (RFID) is a rapid growing technology for automated identification of objects and people. RFID devices are fast becoming one of the most popular technologies ever to enter the consumer marketplace. These so-called RFID devices (better known as RFID tags) are small microchips designed for wireless data transmission. There are three kinds of RFID tags: passive tags, semi-passive tags and active tags [1]. Our focus in this paper is on the passive tags due to their low cost and promising future. Passive RFID tags are expected to be a next-generation successor to barcodes, leading to new markets in various fields. However, these tags have no on-board power source and they only derive their transmission power from the signal of an interrogating reader. That is, these tags are more challenging because of the resource-constrained environment. These resource constrains arise primarily due to cost consideration and their size. Low processing power and low memory necessitates the use of lightweight cryptography for dealing with privacy and security issues.

Nowadays, RFID is used in a wide variety of applications, from remote keyless entry for automobiles to highway toll collection, supply-chain and inventory management, theft prevention, security and access control, and anti-counterfeiting protection [2], [3]. Fig. 1 to Fig. 4 show some examples of RFID application. As we know, RFID tags are gradually used to replace barcodes. Compared with barcodes, RFID tags are able to store significantly more data and do not require line-of-sight contact. Although RFID tags are purported to supplant the ubiquitous barcode on almost every grocery product in the very near future, privacy and security issues associated with their use hamper their smooth implementation. Note here that RFID privacy concerns the problem of misbehaving readers harvesting information from well-behaving tags, while RFID authentication concerns the problem of well-behaving readers harvesting information from misbehaving tags [1]. Lots of complicated and powerful cryptographic algorithms cannot be implemented in RFID tags due to the constrained resource. Hence, designing a lightweight cryptosystem in RFID to protect the privacy and security of RFID system is one of the most challenging tasks.

SMART ENTRY & START SYSTEM
A complete aftermarket smart keyless entry and ignition solution
Advanced_Keys
NOW WITH
360° RF COVERAGE
New
Push-Start
Ignition
BUILT IN
PASSIVE BYPASS
GENUINE OEM LEXUS
PUSH START BUTTON INCLUDED
ENGINE
START
STOP

Fig.1 Remote keyless entry for automobiles.

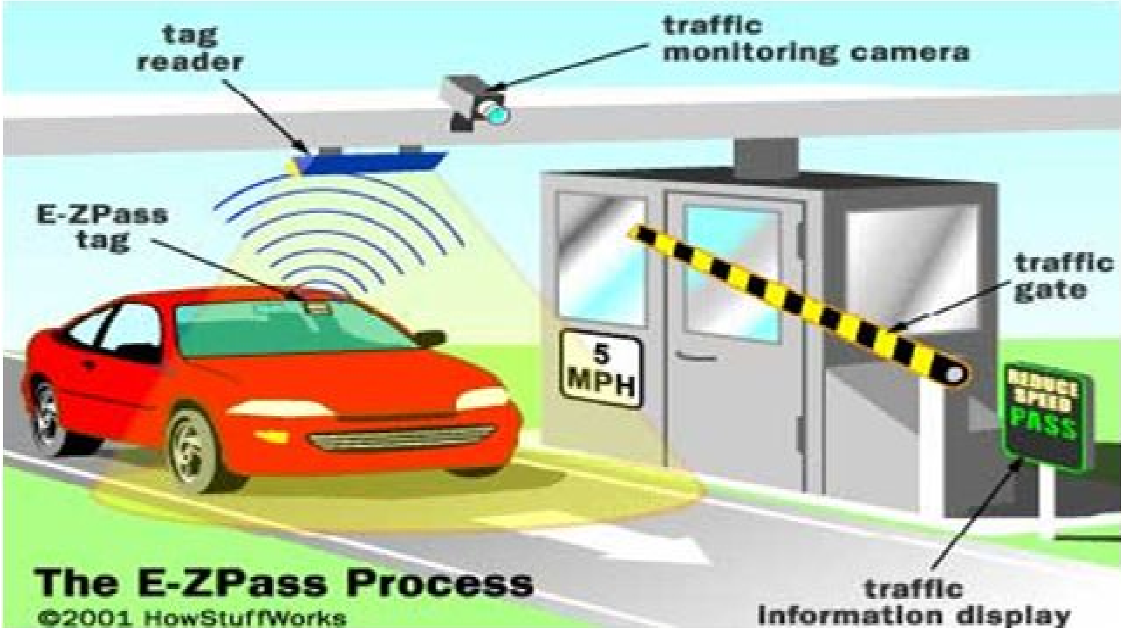

Fig. 2 Highway toll collection.

Fig.3 Supply-chain and inventory management.

Fig. 4 RFID anti-counterfeiting protection.

RFID system has many benefits [13], [14], for example, in libraries where putting an electronic tag in each book simplifies the borrowing and returning procedures and facilitates the staff's job. However, to receive the benefits of RFID, we must overcome the security and privacy problems. The former assesses the soundness of authentication and the latter assesses the ability to resist the tracking (tracing) attacks. Tag authentication and reader authentication are basic requirements, which ensure that only correct tags or readers can be accepted. Tag untraceability is another important security requirement. Due to the basic functionality of RFID [1], [15], the responses from tags are transmitted indiscriminately. This property can be utilized by adversaries to track a specific user or object. The traceability of tags, and by extension of people, is a difficulty that RFID technology must surmount if it is to be widely used. In addition, forward-security is also a significant requirement. RFID tags are not tamper-proof devices, so adversaries can easily obtain the secret data stored in the tags. Forward-security is required to protect the tag's previous data. Other security requirements such as self-synchronization and replay attack resistance are also important and need to be satisfied.

Recent works in RFID attempt to solve the RFID privacy and security problems, however they usually use real IDs of the participating tags to do the authentication during the communications between the reader and the tag. In general, the RFID reader or the back-end server stores all the IDs and secret passwords of RFID tags. After the reader queries the tag, the tag can transmit a reply to the reader. Based on the reply, the reader or the back-end server can verify whether the tag is genuine. Unfortunately, the procedures of authentication with real IDs of the tags are very dangerous. Once the real IDs of the tags are exposed, the adversary can easily obtain the privacy information of the tags as well as the privacy information of their related owners. It would further result in the fact that the adversary is able to track the possessor of the products attached with the tags. Pseudonyms can be used to substitute the real ID of the tag to protect the privacy, however only static pseudonyms are not enough to defend against tracking attacks. The reason is the static pseudonyms will be analyzed the same way as its real ID. Therefore, the tag should use dynamic pseudonyms instead and update its pseudonym after each successful authentication protocol session. Lots of researches [4]-[12], [30], [31], [42], [43], [45]-[48] focus on designing authentication protocols in RFID-tagged systems to protect the privacy and security of the use of RFID tags, however some of them either violate the privacy and security requirements or satisfy the requirements but with high cost.

In this paper, we first define a formal security model for authentication and privacy in RFID system. Under this model, we then propose an Secure RFID Authentication Protocol (*SRAP*) that provides strong privacy and security protection of the RFID users. Note that SRAP can perform the authentication without disclosing real IDs of the participating tags. Except the basic mutual authentication, our protocol offers tag untraceability to resist the tracking attack and forward-security to protect tags' history. In addition, other security requirements such as self-synchronization, replay attack resistance and disclosure attack resistance are also satisfied in SRAP. Finally, we analyze the privacy and security of the proposed protocol and show the proofs. Compared with the previous researches, the major advantage of SRAP is that SRAP can withstand different types of attacks with low cost, which satisfies the requirement of highly resource-constrained RFID tags. In particular, the proposed SRAP requires only 8 hash operations and 18 bit-wise operations for computational cost.

The rest of this paper is organized as follows: In the following section, some related works are briefly introduced. Security model and definitions are presented in Section 3. Under this model, a secure RFID authentication protocol (SRAP) is described in detail in Section 4. Security analysis and performance analysis are shown in Section 5 and Section 6, respectively. Finally, the conclusions of this paper are covered in Section 7.

2. Related Work

The success of RFID tag implementations depends on addressing privacy and security issues surrounding the use of RFID tags. People always hope that their privacy and security are able to be protected. However, a majority of existing RFID tag implementations are not secure, even though the RFID technology increases the safety of food and drugs through proper monitoring and counterfeit prevention. These tags can broadcast information about their presence so that an adversary can silently track and monitor the presence of an RFID tag from a distance without the knowledge of the person holding the tagged object [13]. Therefore, a robust RFID authentication protocol is needed to provide strong privacy and security protection of the RFID users.

In order to design a robust RFID authentication protocol, we need first define a proper security model, under which we describe the protocol and achieve the provable security properties [33]. Avoine [34], Juels and Weis [35], Vaudenay [36], and Ouafi and C. W. Phan [37] made the notable work in designing privacy models in RFID system. Their models differ in the treatment of the adversary's ability to corrupt tags. Our model in this paper is mainly based on [35], [37].

Recent researches in RFID authentication protocols focus on offering adequate privacy and security protection of the use of RFID tags with low cost. We categorize existing protocols into two classes: single tag authentication protocols and multiple tags authentication protocols. Interested readers can refer to recent survey papers [1], [3], [13], [14] for more details.

Several authentication protocols for single tag have been proposed in the literature [4]-[12], [15]-[17], [25], [28], [42], [43], [45]-[48]. Early work by Weis et al. [4] required cryptographic hash functions to perform RFID authentication such that an adversary cannot track the movement of a tag by repeatedly querying and comparing values received from the tag reply. However, the procedures of RFID authentication use the real IDs of RFID tags. An adversary may compromise a tag and derive all the important information by physical attack. Furthermore, an adversary can counterfeit RFID tags or track the objects attached with RFID tags. Molnar and Wagner [18] pointed out that the randomized hash functions do not defend against an eavesdropper. An adversary can impersonate the RFID tag to fool a reader by intercepting and learning the tag reply during the communication between reader and tag. The authors suggested that both the reader and tag need to contribute a random number. However, it does not consider the case of a compromised reader. An adversary is able to obtain the secret information of each tag stored in the reader. Our protocol addresses this vulnerability.

Henrici and Muller [16] proposed a protocol by using the hashed session number. It prevents replay attacks since the session number is incremented every time the tag is read. But this protocol can still be compromised because of the attacks based on the non-randomness of transmitted information, refreshment avoidance, and database de-synchronization. The protocol proposed by Tsudik [5] uses monotonically increasing timestamps to provide tag authentication. As we know, only using timestamps is inadequate for authentication. It is vulnerable to replay attacks since an adversary can send a series of future timestamps to the tag and record its responses. When the times in these timestamps eventually become true, the adversary can respond to requests from the reader without the tag presence. Protocol of Lee et al. [6] uses both XOR and hash chains to authenticate tags and readers. Protocol of Yang et al. [7] has its own freshly generated random bit vectors (r_A, r_B). They can prevent lots of kinds of attacks such as replay attack, man-in-the-middle attack, and so on. However, the real IDs are still operated in the procedures of authentication.

Some commercially available RFID tags can perform cryptographic challenge-response protocols [19]. Such tags offer resistance to attacks involving skimming and cloning. They cost significantly more than the basic and passive tags. Therefore, they are viable only for niche applications like consumer payments.

Lopez et al. proposed a series of lightweight authentication protocols [9]-[11], where the tags involve only simple bit-wise operations like XOR, AND, OR, and addition mod 2^m. These protocols are very efficient and utilize the pseudonyms instead of the real IDs of the tags to perform the authentication. Unfortunately, these schemes are not very robust. They cannot provide strong privacy and security. In particular, Li and Wang [23] and Li and Deng [24] pointed out Lopez's protocols cannot withstand the de-synchronization attack and the disclosure attack. Chien [25] also found that the previous schemes [9]-[11] only provide weak authentication and weak integrity protection, which make them vulnerable to various attacks. Hence, Chien proposed a new RFID authentication protocol termed SASI [25], which can provide strong authentication and strong integrity by using only bit-wise operations. However, Cao et al. [26], Phan [27], and D'Arco et al. [44] pointed out SASI is insecure. It may be quite dangerous using only simple bit-wise operations to achieve RFID authentication under powerful adversarial model. Their works reported the de-synchronization attack, man-in-the-middle attack, tracing attack, and disclosure attack on SASI. Cao et al. [26] also claimed that SASI does not support forward security and further emphasized that the assertion of the SASI protocol that it provides mutual authentication is incorrect. An attacker can easily replay old messages and impersonates a reader, since the tag does not support random number generator to generate a challenge nonce.

A serverless RFID authentication protocol (SLRAP) is proposed by Tan et al. [28]. It can provide mutual authentication between the RFID reader and RFID tag without the need for a persistent central database. However, the reader needs to achieve an access list of tags from certificate authority (CA) before the mutual authentication. It constrains the authentication flexibility. Moreover, for each authentication, the server needs to traverse its database to perform hash operation. It is really a heavy computation burden for the server especially when the number of tag is large. In addition, SLRAP does not provide de-synchronization resistance and forward security. Recently, some enhanced RFID authentication protocols are proposed in [42], [43], [45]-[48]. However, the security and privacy protections of these protocols are weak.

Another research aspect of RFID authentication is multiple tags authentication, which is briefly discussed in this paragraph. We'd like to emphasize that our paper focuses only on a novel single tag authentication protocol, which provides strong privacy and security. However, it is worth studying multiple tags authentication protocols in order to better understand RFID technology in the applications of privacy and security. The earliest work of multiple tags authentication is "Yoking-Proofs" proposed by Juels [20], which gives a proof that a pair of authentic RFID tags has been scanned simultaneously. Saito and Sakurai [21] pointed out that "Yoking-Proofs" cannot resist replay attacks and presented a new multiple tags authentication called grouping proof. In order to avoid the replay attack, they suggested performing the authentication by using timestamps. Later, Piramuthu [22] showed that grouping proof can still not withstand the replay attack. Piramuthu improved "Yoking-Proof" and grouping proof, and mentioned that his modified proof is able to prevent the replay attack. The idea is to ensure that the inputs to a tag are based on parameters that are necessary for the other tag, and to create dependence of the tags on each other. In this case, they cannot be processed separately in the proof without the presence of the other tag.

3. Security Model and Definitions

3.1. Modeling the System

An RFID system is made up of entities (back-end server, readers, and tags) as well as communication channels. Note here that the information contained by the back-end server and the readers is secure as these devices do not have particular restrictions and can therefore make use of appropriate cryptographic techniques. The communication channel between the readers and the back-end server is assumed to be secure. As a consequence, the readers and the backend server are often considered as a single and unique entity in the security analysis (denoted by R). However, the communication channel between the readers and the tags is susceptible to all possible attacks. Adversaries can totally control the communications between the readers and the tags. In this paper, we assume that passive RFID tags have poor electronic power provided by interrogating readers and only can perform light calculations. The memory in the tag is not resilient against tampering attacks. The notations used in this paper are summarized in Table 1.

Table 1 Notations

Symbol	Description
$\mathcal{R}$	RFID reader
r_R	Random number generated by $\mathcal{R}$
$\mathcal{T}$	RFID tag
r_T	Random number generated by $\mathcal{T}$
ID	Static identity of $\mathcal{T}$
P	Pseudonym of $\mathcal{T}$
x	Secret key of $\mathcal{T}$
$\mathcal{A}$	Adversary
i	Protocol session identifier
$\mathcal{P}$	An RFID authentication protocol
BS	Back-end server
$\epsilon(k)$	Negligible function of k
H	Hash functions: $\{0,1\}^* \rightarrow \{0,1\}^l$
$\oplus$	Exclusive-OR (XOR) operation

3.2. Adversary Model

In this subsection, we present the formalization of the adversarial model in order to analyze the security of the proposed protocol and show the security proof. Our model is mainly based on [35], [37].

In our model, a reader is denoted by $\mathcal{R}$, a tag is denoted by $\mathcal{T}$, and an adversary is denoted by $\mathcal{A}$. We let i be protocol session identifier. Note here that a reader can concurrently run several instances of the RFID protocol $\mathcal{P}$ while a tag can run only one instance of $\mathcal{P}$ at a time. Adversary $\mathcal{A}$ controls the communications between a $\mathcal{T}$ and a $\mathcal{R}$ by interacting with them as defined by the protocol, formally captured by $\mathcal{A}$'s ability to issue queries of the following form:

Execute($\mathcal{R}$, $\mathcal{T}$, i):

This query models passive attacks, where adversary $\mathcal{A}$ gets access to an honest execution of the protocol session i between $\mathcal{R}$ and $\mathcal{T}$ by eavesdropping.

Send^1($\mathcal{R}$, $\mathcal{T}$, i, m_1):

This query models active attacks by allowing the adversary $\mathcal{A}$ to impersonate reader $\mathcal{R}$ in the first data-flow of some protocol session i and send a message m_1 of its choice to tag $\mathcal{T}$.

Send^2($\mathcal{T}$, $\mathcal{R}$, i, m_2):

This query models active attacks by allowing the adversary $\mathcal{A}$ to impersonate tag $\mathcal{T}$ in the second data-flow of some protocol session i and send a message m_2 of its choice to reader $\mathcal{R}$.

Send^3($\mathcal{R}$, $\mathcal{T}$, i, m_3):

This query models active attacks by allowing the adversary $\mathcal{A}$ to impersonate reader $\mathcal{R}$ in the third data-flow of some protocol session i and send a message m_3 of its choice to tag $\mathcal{T}$.

Corrupt($\mathcal{T}$, i):

This query allows the adversary $\mathcal{A}$ to learn the content of the tag $\mathcal{T}$'s memory in some protocol session i. This query can be used only once such that Execute, Send^1, Send^2 and Send^3 can no longer be used after. This kind of attack is possible in view that RFID tags are typically not designed to be tamper-resistant, thus once tags are deployed it is possible for an adversary to tamper with the tag to read from its memory in which stored secrets are kept. We assume that tampering with a tag destroys it so that it no longer circulates in nature. This attack is an invasive one that is much stronger than active attacks captured by the Send* ($* \in \{1, 2, 3\}$) queries, because Corrupt queries mean that the adversary has physical access to the tag, compared to Send* ($* \in \{1, 2, 3\}$) queries where the adversary has access only to the communication channel between the reader and the tag. Indeed, in the event that Corrupt queries are possible, i.e., the adversary can read and tamper with the tag's memory, the most that can still be offered is that security of previously completed sessions are not compromised. This notion is known as forward security.

Test(i, $\mathcal{T}_0$, $\mathcal{T}_1$):

This query is the only query that does not correspond to any of $\mathcal{A}$'s abilities or any real-world event. This query allows to define the indistinguishability-based [38], [39] notion of untraceability (UNT). Upon the issuance of a Test query for session i, then depending on a randomly chosen bit $b \in \{0,1\}$, $\mathcal{A}$ is given $\mathcal{T}_b$ from the set $\{\mathcal{T}_0, \mathcal{T}_1\}$. Informally, $\mathcal{A}$ succeeds if it can guess the bit b.

For ease of legibility, we will use E, S^1, S^2, S^3 and C to represent respectively the queries Execute, Send^1, Send^2, Send^3 and Corrupt.

3.3. Security Definitions

There are two characterizations in a RFID authentication protocol: security and privacy. The former assesses the soundness of authentication. The latter property is for the ability to resist to adversaries aiming at tracing or linking tags.

In this paper, we are concerned with mutual authentication and untraceability. Moreover, the extended security notions of forward-security and self-synchronization are also considered. The RFID tag does not have tamper-resilient memory, so the adversary can easily obtain the secret data stored in the tag. Forward-secure property is required to protect the tag's previous privacy and security. In addition, communication through wireless channel can be easily blocked by adversaries' attacks or disturbed due to transmission failures. Hence, self-synchronization is required to resist the de-synchronization attack or recover the status in the case the communication between the reader and the tag fails. Note here that the attacks like de-synchronization are very difficult to prevent. In this paper, we assume that the adversary cannot block the two same messages transmitted between the reader and the tag in two consecutive sessions.

Definition 1 (Tag Unforgeability).

Our definition of tag unforgeability (TUF) for the proposed protocol characterizes the ability of adversary $\mathcal{A}_{\mathrm{TUF}}$ to clone valid-looking tags in an RFID system. TUF is defined using the game $\mathcal{G}_{\mathrm{TUF}}$ played between a malicious adversary $\mathcal{A}_{\mathrm{TUF}}$ and a collection of reader and tag instances, in which $\mathcal{A}_{\mathrm{TUF}}$ interacts with the reader and with tags for an arbitrary period of time determined by $\mathcal{A}_{\mathrm{TUF}}$. $\mathcal{A}_{\mathrm{TUF}}$ runs the game $\mathcal{G}_{\mathrm{TUF}}$ whose setting is as follows.

Phase 1 (Learning):

$\mathcal{A}_{\mathrm{TUF}}$ is able to send any Execute, Send^1, Send^2, and Send^3 queries at will.

Phase 2 (Challenge):

In the challenge phase, the adversary $\mathcal{A}_{\mathrm{TUF}}$ has no oracle access to tags. $\mathcal{A}_{\mathrm{TUF}}$ outputs a message to query $\mathcal{R}$ until $\mathcal{R}$ yields some output γ.

```
if $\mathcal{R}$ accept then
    output '1' ;
else
    output '0' ;
```

Then, $\mathcal{A}_{\mathrm{TUF}}$ terminates the game simulation.

The goal of $\mathcal{A}_{\mathrm{TUF}}$ is to cause $\mathcal{R}$ to accept. Acceptance in the "challenge" phase implies a successfully mounted adversarial attack against the tag unforgeability. In particular, a successful adversary is capable of creating a freestanding tag that can cause the reader to accept at least once. The success of $\mathcal{A}_{\mathrm{TUF}}$ in winning $\mathcal{G}_{\mathrm{TUF}}$ and thus breaking the notion of TUF is quantified in terms of $\mathcal{A}_{\mathrm{TUF}}$'s advantage in causing $\mathcal{R}$ to output "γ = 1". This is denoted by $Adv_{A_{TUF}}^{TUF}(k)$ where k is the security parameter. Hence, our concrete definition of TUF is given as follows:

A protocol in an RFID system with security parameter k is tag unforgeable if:

$$\begin{aligned} Adv_{A_{TUF}}^{TUF}(k) &= \Pr\left[A_{TUF}^{E,S^1,S^2,S^3}\ win\ G_{TUF}\right], \\ &= \Pr[\gamma = 1] \le \epsilon(k) \end{aligned}$$

where $\epsilon(k)$ is negligible function of k[1].

Definition 2 (Reader Unforgeability).

Reader unforgeability (RUF) is defined using the game $\mathcal{G}_{\mathrm{RUF}}$ played between a malicious adversary $\mathcal{A}_{\mathrm{RUF}}$ and a collection of reader and tag instances, in which $\mathcal{A}_{\mathrm{RUF}}$ interacts with the reader and with tags for an arbitrary period of time determined by $\mathcal{A}_{\mathrm{RUF}}$. $\mathcal{A}_{\mathrm{RUF}}$ runs the game $\mathcal{G}_{\mathrm{RUF}}$ whose setting is as follows.

Phase 1 (Learning):

$\mathcal{A}_{\mathrm{RUF}}$ is able to send any Execute, Send^1, Send^2, and Send^3 queries at will.

[1] A function is negligible if it approaches zero faster than the reciprocal of any polynomial $p(k)$. More formally, $\epsilon: N \rightarrow R$ is negligible if for any nonzero polynomial $p(\bullet)$ there exists an m such that $\forall n > m, |\epsilon(n)| < 1/p(n)$.

Phase 2 (Challenge):

In the challenge phase, the adversary $\mathcal{A}_{\mathrm{RUF}}$ has no oracle access to readers. $\mathcal{A}_{\mathrm{RUF}}$ outputs a message to query $\mathcal{T}$ until $\mathcal{T}$ yields some output η.

```
if $\mathcal{T}$ accept then
    output '1' ;
else
    output '0' ;
```

Then, $\mathcal{A}_{\mathrm{RUF}}$ terminates the game simulation.

The goal of $\mathcal{A}_{\mathrm{RUF}}$ is to cause $\mathcal{T}$ to accept. Acceptance in the "challenge" phase implies a successfully mounted adversarial attack against the reader unforgeability. In particular, a successful adversary is capable of creating a freestanding reader that can cause the tag to accept at least once. The success of $\mathcal{A}_{\mathrm{RUF}}$ in winning $\mathcal{G}_{\mathrm{RUF}}$ and thus breaking the notion of RUF is quantified in terms of $\mathcal{A}_{\mathrm{RUF}}$'s advantage in causing $\mathcal{T}$ to output "η = 1". This is denoted by $Adv_{A_{RUF}}^{RUF}(k)$ where k is the security parameter. Hence, our concrete definition of RUF is given as follows:

A protocol in an RFID system with security parameter k is reader unforgeable if:

$$\begin{aligned} Adv_{A_{RUF}}^{RUF}(k) &= \Pr\left[A_{RUF}^{E,S^1,S^2,S^3}\ win\ G_{RUF}\right], \\ &= \Pr\left[\eta = 1\right] \le \epsilon(k) \end{aligned}$$

where $\epsilon(k)$ is negligible function of k.

Definition 3 (Tag Untraceability).

Our definition of tag untraceability (UNT) for the proposed protocol characterizes the ability of adversary $\mathcal{A}_{\mathrm{UNT}}$ to trace or link tags in an RFID system. UNT is defined using the game $\mathcal{G}_{\mathrm{UNT}}$ played between a malicious adversary $\mathcal{A}_{\mathrm{UNT}}$ and a collection of reader and tag instances, in which $\mathcal{A}_{\mathrm{UNT}}$ interacts with the reader and with tags for an arbitrary period of time determined by $\mathcal{A}_{\mathrm{UNT}}$. $\mathcal{A}_{\mathrm{UNT}}$ runs the game $\mathcal{G}_{\mathrm{UNT}}$ whose setting is as follows.

Phase 1 (Learning):

$\mathcal{A}_{\mathrm{UNT}}$ is able to send any Execute, Send^1, Send^2, Send^3, and Corrupt queries at will.

Phase 2 (Challenge):

1. At some point during $\mathcal{G}_{\mathrm{UNT}}$, $\mathcal{A}_{\mathrm{UNT}}$ will choose two fresh tags ($\mathcal{T}_0$, $\mathcal{T}_1$) to be tested and send a Test query corresponding to this. Fresh means that the tags have not been issued any Corrupt query. Depending on a randomly chosen bit $b \in \{0, 1\}$, $\mathcal{A}_{\mathrm{UNT}}$ is given a tag $\mathcal{T}_b$ from the set $\{\mathcal{T}_0, \mathcal{T}_1\}$.
2. $\mathcal{A}_{\mathrm{UNT}}$ continues making any Execute, Send^1, Send^2, Send^3, and Corrupt queries, subjected to the restriction that the tag $\mathcal{T}_0$ and $\mathcal{T}_1$ are not issued any Corrupt query.

Phase 3 (Guessing):

Eventually, $\mathcal{A}_{\mathrm{UNT}}$ terminates the game simulation and outputs a bit b', which is its guess of the value of b.

The goal of $\mathcal{A}_{\mathrm{UNT}}$ is to guess the correct value of b. The correct output of b' in the "guessing" phase implies a successfully mounted adversarial attack against the tag untraceability. The success of $\mathcal{A}_{\mathrm{UNT}}$ in winning $\mathcal{G}_{\mathrm{UNT}}$ and thus breaking the notion of UNT is quantified in terms of $\mathcal{A}_{\mathrm{UNT}}$'s advantage in distinguishing whether $\mathcal{A}_{\mathrm{UNT}}$ received $\mathcal{T}_0$ or $\mathcal{T}_1$, i.e., it correctly guesses b. This is denoted by $Adv_{A_{UNT}}^{UNT}(k)$ where k is the security parameter. Hence, our concrete definition of UNT is given as follows:

A protocol in an RFID system with security parameter k is tag untraceable if:

$$\begin{aligned} Adv_{A_{UNT}}^{UNT}(k) &= \left|\Pr\left[A_{UNT}^{E, S^1, S^2, S^3, C} \ win \ G_{UNT}\right] - 1/2\right| \ , \\ &= |\Pr[b' = b] - 1/2| \leq \epsilon(k) \end{aligned}$$

where $\epsilon(k)$ is negligible function of k. We subtract 1/2 here as the adversary can trivially guess bit b successfully with probability 1/2.

Definition 4 (Forward Security).

Our definition of forward security (FS) for the proposed protocol characterizes the ability of adversary $\mathcal{A}_{\mathrm{FS}}$ to obtain the tag's previous secret keys when $\mathcal{A}_{\mathrm{FS}}$ corrupts the tag. FS is defined using the game $\mathcal{G}_{\mathrm{FS}}$ played between a malicious adversary $\mathcal{A}_{\mathrm{FS}}$ and a collection of reader and tag instances, in which $\mathcal{A}_{\mathrm{FS}}$ interacts with the reader and with tags for an arbitrary period of time determined by $\mathcal{A}_{\mathrm{FS}}$. $\mathcal{A}_{\mathrm{FS}}$ runs the game $\mathcal{G}_{\mathrm{FS}}$ whose setting is as follows.

Phase 1 (Learning):

$\mathcal{A}_{\mathrm{FS}}$ is able to send any Execute, Send^1, Send^2, Send^3, and Corrupt queries at will.

Phase 2 (Challenge):

$\mathcal{A}_{\mathrm{FS}}$ starts a new session i, during which $\mathcal{A}_{\mathrm{FS}}$ chooses a fresh tag $\mathcal{T}'$ to be challenged and sends it a Corrupt query to obtain its secret key x^i. Fresh means that the tag has not been issued any Corrupt query before.

Phase 3 (Guessing):

Eventually, $\mathcal{A}_{\mathrm{FS}}$ terminates the game simulation and outputs $x^{t'}\ (t < i)$, which is its guess of the value of the previous key x^t of $\mathcal{T}'$. Note here that $x^i = Update^{i-t}(x^t, t, \bullet)$2).

The goal of $\mathcal{A}_{\mathrm{FS}}$ is to guess the correct value of x^t. The correct output of $x^{t'}$ in the "guessing" phase implies a successfully mounted adversarial attack against the forward security. The success of $\mathcal{A}_{\mathrm{FS}}$ in winning $\mathcal{G}_{\mathrm{FS}}$ and thus breaking the notion of FS is quantified in terms of $\mathcal{A}_{\mathrm{FS}}$'s advantage in guessing the correct value of the previous key x^t of $\mathcal{T}$. This is denoted by $Adv_{A_{FS}}^{FS}(k)$ where k is the security parameter. Hence, our concrete definition of FS is given as follows:

A protocol in an RFID system with security parameter k is forward secure if:

$$\begin{aligned} Adv_{A_{FS}}^{FS}(k) &= \Pr\left[A_{FS}^{E,S^1,S^2,S^3,C}\ win\ G_{FS}\right] \\ &= \Pr\left[x^{t'} = x^t \mid x^i = Update^{i-t}(x^t, t, \bullet\,)\right] \\ &\le \epsilon(k) \end{aligned} ,$$

where $\epsilon(k)$ is negligible function of k.

2) *Update* denotes some key updating function. ' • ' denotes some related input of function *Update*.

4. SRAP: A Secure RFID Authentication Protocol

In this section, we elaborate the novel RFID authentication protocol-SRAP. Our protocol works under the model defined in Section 3. We suppose that the RFID tag has a random number generator, XOR gates and a re-writeable memory like EEPROM. We let $H: \{0,1\}^* \rightarrow \{0,1\}^l$ be a random oracle. Note that the random oracle model is used as a mathematical model of a perfect hash function. Intuitively, it captures the intuition that we should not be able to extract any information from how a hash function computes its hash. As we know, a hash function is a powerful and computational efficient cryptographic tool. According to [29], a hash function can be implemented with only about 1.7*K* gates, which satisfies the requirement of highly resource-constrained low-cost RFID tags.

In SRAP, RFID tags substitute pseudonyms for real IDs in communications. If a tag uses one pseudonym all the time, it will not help to defend against passive attacks, because the pseudonym will be analyzed the same way as its real ID. Therefore, each tag should use dynamic pseudonyms instead. For this purpose, each tag has to update its pseudonym after each successful authentication. Moreover, it is worth noting that tag memories are not assumed to be tamper-proof. RFID tags cannot be trusted to securely store long-term share secrets [4]. Hence, it is necessary for the tag to execute secret key update after each successful authentication, which will be explained in detail below.

In SRAP, each RFID tag pre-shares a static identity (*ID*), a pseudonym (P), and a secret x with the backend server. We assume that the length of each of *ID*, P and x is l bits. That is, the memory size of a RIFD tag is only $3l$ bits. In accordance with [4], a 96-bit *ID* including the identifying data of the manufacturer, brand, model and a unique serial number would suffice for most applications. Hence, in practice, a RFID tag with only 288-bit re-writeable memory size can be utilized to perform the authentication. In order to withstand the replay attack, the random numbers r_R and r_T will be changed each session. Also, the length of a random number is l bits. Meanwhile, in order to resist the possible de-synchronization attack, the back-end server will actually keep two entries of $(P,\ x)$. One is for the old values and the other is for the potential next values. Hence, the back-end server stores five values for each tag: P^{old}, x^{old}, P^{new}, x^{new} and the static *ID*. This arrangement will become obvious when we introduce the protocol and analyze the possible attacks. We'd like to emphasize that the attacks like de-synchronization are very difficult to prevent. In this paper, we assume that the adversary cannot block the two same messages transmitted between the reader and the tag in two consecutive sessions.

4.1. Efficient RFID Authentication Protocol

The protocol can be divided into three phase: initial phase, tag authentication phase, and reader authentication phase. The details of the protocol are shown in Table 2.

Table 2 SRAP Protocol

Back-end Sever/Reader		RFID Tag
Protocol session i		Protocol session i
$r_R \xleftarrow{R} \{0,1\}^l$		Random number generated by $\mathcal{R}$
Send r_R	$\xrightarrow{r_R}$	$r_T \xleftarrow{R} \{0,1\}^l$
		Compute
		$M = H(r_T \oplus r_R \oplus P \oplus x)$
	$\xleftarrow{r_T\,, P, M}$	Send r_T, P, M
Tag authentication		
Match P and find out x		
Compute		
$M^i = H(r_T \oplus r_R \oplus P^i \oplus x^i)$		
Verify $M^i = ?M$		
if $M^i = M$ accept and set $\gamma \leftarrow 1$		
else compute		
$M^{i-1} = H(r_T \oplus r_R \oplus P^{i-1} \oplus x^{i-1})$		
if $M^{i-1} = M$ accept and set $\gamma \leftarrow 1$		
else set $\gamma \leftarrow 0$		
if $\gamma = 1$ and $M^i = M$		
compute $N = H(M^i \oplus P^i \oplus x^i)$		
if $\gamma = 1$ and $M^{i-1} = M$		
compute $N = H(M^{i-1} \oplus P^{i-1} \oplus x^{i-1})$		
if $\gamma = 0$		
Pick $rnd \xleftarrow{R} \{0,1\}^*$ and set $N = H(rnd)$		
Send N	$\xrightarrow{N}$	

Table 2 SRAP Protocol (Continue)

Back-end Sever/Reader		RFID Tag
	$\xrightarrow{N}$	
		Reader authentication
		Compute $N' = H(M \oplus P \oplus x)$
		if $N' = N$ accept and set $\eta \leftarrow 1$
		else set $\eta \leftarrow 0$ and reject
Pseudonym and key update		Pseudonym and key update
if $M^i = M$		if $\eta = 1$
Compute $P = H(P^i \oplus x^i \oplus ID)$		Compute $P' = H(P \oplus x \oplus ID)$
$x = H(x^i \oplus M^i \oplus ID)$		$x' = H(x \oplus M \oplus ID)$
and set $P^{i-1} \leftarrow P^i, P^i \leftarrow P$		and set $P \leftarrow P'$
$x^{i-1} \leftarrow x^i, x^i \leftarrow x$		$x \leftarrow x'$
$i \leftarrow i+1$		$i \leftarrow i+1$
else keep the status		else keep the status

Initial Phase:

Initially, the reader (the back-end server)[3] starts a protocol session by sending a "request" with a random number r_R to the tag. After receiving the message from the reader, the tag also generates a random number r_T , and further calculates the value of M by the following equation: $M = H(r_T \oplus r_R \oplus P \oplus x)$.

Tag Authentication Phase:

The tag sends the message $\{r_T, P, M\}$ to the reader, but keeps x secret. The reader keeps five values for each tag: P^{old}, x^{old}, P^{new}, x^{new} and *ID*. The entry of (P^{old}, x^{old}) is for the old values while the entry of (P^{new}, x^{new}) is for the potential next values. For example, in protocol session i, P^i and x^i delegate the potential next values while P^{i-1} and x^{i-1} denote the old values. After receiving the message $\{r_T, P, M\}$, the reader checks its memory to match the tag's response P using the potential next value and find out the corresponding secret key. And then, the reader computes $M^i = H(r_T \oplus r_R \oplus P^i \oplus x^i)$ to check whether M^i equals M. If $M^i = M$, the reader authenticates the tag and sets $\gamma = '1'$. If $M^i \neq M$, the reader further calculates $M^{i-1} = H(r_T \oplus r_R \oplus P^{i-1} \oplus x^{i-1})$ using the old values and checks whether M^{i-1} equals M. Under this condition, if $M^{i-1} = M$, the reader still can authenticate the tag and set $\gamma = '1'$. Otherwise, the reader sets $\gamma = '0'$. Based on the value of γ, the reader computes N and sends it to the tag. After that, the reader performs pseudonym and key update[4], where the pseudonym P and the secret x are both updated. The whole process is specified in Table 2.

[3] Note that the reader and the back-end server are often considered as a single entity. We assume that the reader can securely access to the data base in the server. For ease of legibility, we will use "reader" to represent "reader and server".

[4] We do not consider the process of pseudonym and key update as an individual phase. In fact, the processes of pseudonym and key update are performed in tag authentication phase and reader authentication phase, respectively.

Reader Authentication Phase:

The reader sends N to the tag. To authenticate the reader, the tag needs to compute $N' = H(M \oplus P \oplus x)$ and verifies whether the equation $N' = N$ can hold. If the tag confirms $N' = N$, it makes sure that the reader is the authentic RFID reader and sets $\eta = '1'$. Otherwise, the tag rejects and sets $\eta = '0'$. After the authentication, the tag performs pseudonym and key update as specified in Table 2.

5. Security Analysis

5.1. Security Proof

In this subsection, we analyze the security of the proposed protocol and show the security proofs according to the security requirements defined in Section 3.

As we see, the security of our system depends most critically on the key-length parameter l. This parameter determines the probability with which an adversary may guess unknown keys in the system. To simplify our proofs, we assume that successful guessing of any key by the adversary results in defeat of the security properties of our protocol.

For the proofs of the theorems below, we begin by defining a special adversary $\mathcal{A}^*$ (benign adversary) with restricted capabilities. This adversary must deliver all reader-to-tag and tag-to-reader messages faithfully, that is, to the correct recipient and without any modification. $\mathcal{A}^*$ must deliver messages corresponding to a specific tag in their correct order. Like $\mathcal{A}$, the adversary $\mathcal{A}^*$ may cause a reader to initiate a session, i.e., yield a first-flow output at any time that the reader is not already engaged in an active session. Thus, $\mathcal{A}^*$ may be regarded essentially as an honest-but-curious adversary. Our proof strategy is to show that a real-world adversary A can effectively do little more than the special adversary $\mathcal{A}^*$ [40], [41]. In other words, we will show that the probability that $\mathcal{A}$'s behavior is not simulable by $\mathcal{A}^*$ is negligible. As a result, the following theorems are shown.

Theorem 1. (Tag Unforgeability)

The proposed protocol is tag unforgeable if hash function H is a random oracle.

Proof of Theorem 1.

We are given a real world adversary $\mathcal{A}_{\mathrm{TUF}}$ and a benign adversary $\mathcal{A}^{*}{}_{\mathrm{TUF}}$. In order to demonstrate that $\mathcal{A}_{\mathrm{TUF}}$ gains no knowledge from its interaction in the real RFID system, we will show that the probability that $\mathcal{A}_{\mathrm{TUF}}$'s behavior is not simulable by $\mathcal{A}^{*}{}_{\mathrm{TUF}}$ in game $\mathcal{G}_{\mathrm{TUF}}$ is negligible.

If $\mathcal{A}_{\mathrm{TUF}}$'s behavior is not simulable by $\mathcal{A}^{*}{}_{\mathrm{TUF}}$ in game $\mathcal{G}_{\mathrm{TUF}}$, then one of the following conditions must occur at some point in the course of the game:

1). $\mathcal{A}_{\mathrm{TUF}}$ sends Send^{2} query to $\mathcal{R}$ and passes the tag verification: Suppose that $\mathcal{A}_{\mathrm{TUF}}$ can make at most q Send^{2} queries to $\mathcal{R}$. Given that M is a l-bit value, then the probability that $\mathcal{A}_{\mathrm{TUF}}$ can successfully guess the correct M and pass the tag verification is at most $q/2^{l}$.

2). $\mathcal{A}_{\mathrm{TUF}}$ sends Send^{3} query to $\mathcal{T}$ and passes the reader verification: Suppose that $\mathcal{A}_{\mathrm{TUF}}$ can make at most t Send^{3} queries to $\mathcal{T}$. Given that N is a l-bit value, then the probability that $\mathcal{A}_{\mathrm{TUF}}$ can successfully guess the correct N and pass the reader verification is at most $t/2^{l}$.

3). $\mathcal{A}_{\mathrm{TUF}}$ submits to H a query of the form $\{\cdot,\ x\}$: Suppose that $\mathcal{A}_{\mathrm{TUF}}$ can make at most w queries to H. Given that H is a random oracle, its outputs reveal no information about secret keys. Hence, the probability that $\mathcal{A}_{\mathrm{TUF}}$ can successfully submit a query of the form $\{\cdot,\ x\}$ is at most $w/2^{l}$.

Thus $\mathcal{A}_{\mathrm{TUF}}$'s behavior is not simulable by $\mathcal{A}^{*}{}_{\mathrm{TUF}}$ with probability at most $(q+t+w)/2^{l}$, which is negligible for polynomially bounded $\mathcal{A}_{\mathrm{TUF}}$. □

Theorem 2. (Reader Unforgeability)

The proposed protocol is reader unforgeable if hash function H is a random oracle.

Proof of Theorem 2.

We are given a real world adversary $\mathcal{A}_{\mathrm{RUF}}$ and a benign adversary $\mathcal{A}^*_{\mathrm{RUF}}$. In order to demonstrate that $\mathcal{A}_{\mathrm{RUF}}$ gains no knowledge from its interaction in the real RFID system, we will show that the probability that $\mathcal{A}_{\mathrm{RUF}}$'s behavior is not simulable by $\mathcal{A}^*_{\mathrm{RUF}}$ in game $\mathcal{G}_{\mathrm{RUF}}$ is negligible.

If $\mathcal{A}_{\mathrm{RUF}}$'s behavior is not simulable by $\mathcal{A}^*_{\mathrm{RUF}}$ in game $\mathcal{G}_{\mathrm{RUF}}$, then one of the following conditions must occur at some point in the course of the game:

1). $\mathcal{A}_{\mathrm{RUF}}$ sends Send^2 query to $\mathcal{R}$ and passes the tag verification: Suppose that $\mathcal{A}_{\mathrm{RUF}}$ can make at most q Send^2 queries to $\mathcal{R}$. Given that M is a l-bit value, then the probability that $\mathcal{A}_{\mathrm{RUF}}$ can successfully guess the correct M and pass the tag verification is at most $q/2^l$.

2). $\mathcal{A}_{\mathrm{RUF}}$ sends Send^3 query to $\mathcal{T}$ and passes the reader verification: Suppose that $\mathcal{A}_{\mathrm{RUF}}$ can make at most t Send^3 queries to $\mathcal{T}$. Given that N is a l-bit value, then the probability that $\mathcal{A}_{\mathrm{RUF}}$ can successfully guess the correct N and pass the reader verification is at most $t/2^l$.

3). $\mathcal{A}_{\mathrm{TUF}}$ submits to H a query of the form $\{\cdot,\ x\}$: Suppose that $\mathcal{A}_{\mathrm{RUF}}$ can make at most w queries to H. Given that H is a random oracle, its outputs reveal no information about secret keys. Hence, the probability that $\mathcal{A}_{\mathrm{RUF}}$ can successfully submit a query of the form $\{\cdot,\ x\}$ is at most $w/2^l$.

Thus $\mathcal{A}_{\mathrm{RUF}}$'s behavior is not simulable by $\mathcal{A}^*_{\mathrm{RUF}}$ with probability at most $(q+t+w)/2^l$, which is negligible for polynomially bounded $\mathcal{A}_{\mathrm{RUF}}$. □

Theorem 3. (Tag Untraceability)

The proposed protocol is tag untraceable if hash function H is a random oracle.

□

Proof of Theorem 3.

We are given a real world adversary $\mathcal{A}_{\mathrm{UNT}}$ and a benign adversary $\mathcal{A}^{*}_{\mathrm{UNT}}$. In the challenge phase, in order to demonstrate that $\mathcal{A}_{\mathrm{UNT}}$ gains no knowledge from its interaction with $\mathcal{T}_b$ (b $\in$ {0, 1}) in the real RFID system, we will show that the probability that $\mathcal{A}_{\mathrm{UNT}}$'s behavior is not simulable by $\mathcal{A}^{*}_{\mathrm{UNT}}$ in game $\mathcal{G}_{\mathrm{UNT}}$ is negligible.

If $\mathcal{A}_{\mathrm{UNT}}$'s behavior is not simulable by $\mathcal{A}^{*}_{\mathrm{UNT}}$ in game $\mathcal{G}_{\mathrm{UNT}}$, then one of the following conditions must occur at some point in the course of the game:

1). $\mathcal{A}_{\mathrm{UNT}}$ sends Send^2 query to $\mathcal{R}$ and passes the tag $\mathcal{T}_0$ or $\mathcal{T}_1$ verification: Suppose that $\mathcal{A}_{\mathrm{UNT}}$ can make at most q Send^2 queries to $\mathcal{R}$. Given that M is a l-bit value, then the probability that $\mathcal{A}_{\mathrm{UNT}}$ can successfully guess the correct M and pass the tag $\mathcal{T}_0$ or $\mathcal{T}_1$ verification is at most $2q/2^l$.

2). $\mathcal{A}_{\mathrm{UNT}}$ sends Send^3 query to $\mathcal{T}$ and passes the reader verification: Suppose that $\mathcal{A}_{\mathrm{UNT}}$ can make at most t Send^3 queries to $\mathcal{T}$. Given that N is a l-bit value, then the probability that $\mathcal{A}_{\mathrm{UNT}}$ can successfully guess the correct N and pass the reader verification is at most $t/2^l$.

3). $\mathcal{A}_{\mathrm{UNT}}$ submits to H a query of the form $\{\cdot,\ x_0\}$ or $\{\cdot,\ x_1\}$[5]: Suppose that $\mathcal{A}_{\mathrm{UNT}}$ can make at most w queries to H. Given that H is a random oracle, its outputs reveal no information about secret keys. Hence, the probability that $\mathcal{A}_{\mathrm{UNT}}$ can successfully submit a query of the form $\{\cdot,\ x_0\}$ or $\{\cdot,\ x_1\}$ is at most $2w/2^l$.

Thus $\mathcal{A}_{\mathrm{UNT}}$'s behavior is not simulable by $\mathcal{A}^{*}_{\mathrm{UNT}}$ with probability at most

[5] x_0 and x_1 denote the secret keys of $\mathcal{T}_0$ and $\mathcal{T}_1$.

$(2q+t+2w)/2^l$, which is negligible for polynomially bounded $\mathcal{A}_{\mathsf{UNT}}$.

Theorem 4. (Forward Security)

The proposed protocol is forward secure if hash function H is a random oracle.

Proof of Theorem 4.

We are given a real world adversary $\mathcal{A}_{\mathsf{FS}}$ and a benign adversary $\mathcal{A}^*_{\mathsf{FS}}$. In order to demonstrate that $\mathcal{A}_{\mathsf{FS}}$ gains no knowledge from its interaction in the real RFID system, we will show that the probability that $\mathcal{A}_{\mathsf{FS}}$'s behavior is not simulable by $\mathcal{A}^*_{\mathsf{FS}}$ in game $\mathcal{G}_{\mathsf{FS}}$ is negligible.

If $\mathcal{A}_{\mathsf{FS}}$'s behavior is not simulable by $\mathcal{A}^*_{\mathsf{FS}}$ in game $\mathcal{G}_{\mathsf{FS}}$, then one of the following conditions must occur at some point in the course of the game:

1). $\mathcal{A}_{\mathsf{FS}}$ sends Send^2 query to $\mathcal{R}$ and passes the tag verification: Suppose that $\mathcal{A}_{\mathsf{FS}}$ can make at most q Send^2 queries to $\mathcal{R}$. Given that M is a l-bit value, then the probability that $\mathcal{A}_{\mathsf{FS}}$ can successfully guess the correct M and pass the tag verification is at most $q/2^l$.

2). $\mathcal{A}_{\mathsf{FS}}$ sends Send^3 query to $\mathcal{T}$ and passes the reader verification: Suppose that $\mathcal{A}_{\mathsf{FS}}$ can make at most t Send^3 queries to $\mathcal{T}$. Given that N is a l-bit value, then the probability that $\mathcal{A}_{\mathsf{FS}}$ can successfully guess the correct N and pass the reader verification is at most $t/2^l$.

3). $\mathcal{A}_{\mathsf{FS}}$ submits to H a query of the form $\{\cdot,\ x\}$: Suppose that $\mathcal{A}_{\mathsf{FS}}$ can make at most w queries to H. Given that H is a random oracle, its outputs reveal no information about secret keys. Hence, the probability that $\mathcal{A}_{\mathsf{FS}}$ can successfully submit a query of the form $\{\cdot,\ x\}$ is at most $w/2^l$.

4). $\mathcal{A}_{\mathsf{FS}}$ guesses the correct value of x^t: Given that H is a random oracle, the output of H is random distribution in the view of $\mathcal{A}_{\mathsf{FS}}$. $\mathcal{A}_{\mathsf{FS}}$ can obtain x^i by sending a Corrupt query in protocol session i, but the probability that $\mathcal{A}_{\mathsf{FS}}$ can successfully guess the correct x^t is at most $(i-1)/2^l$.

Thus $\mathcal{A}_{\mathsf{FS}}$'s behavior is not simulable by $\mathcal{A}^{\star}{}_{\mathsf{FS}}$ with probability at most $(q+t+w+i-1)/2^l$, which is negligible for polynomially bounded $\mathcal{A}_{\mathsf{FS}}$.

□

5.2. Security Discussion

Except the basic mutual authentication, tag untraceability and forward-security, our protocol satisfies other security properties such as de-synchronization attack resistance, replay attack resistance and disclosure attack resistance. We give a simple discussion below.

5.2.1. De-synchronization Attack Resistance

In de-synchronization attacks, attacker can modify the shared data to make the server and the tag out of synchronization without being noticed. However, in our protocol, the attacker cannot change the data without being noticed, since the calculations of M and N explicitly involve the random numbers (r_R, r_T), the pseudonym (P) and the secret (x), which makes the calculation authenticity and integrity. Any slight modification of the data will result in the failure of the authentication. In addition, in our protocol, the adversary can intercept and block the data N sent by the reader to make the server updates its local data while the tag does not. Fortunately, this cannot cause trouble to our protocol, because the server keeps two entries of (P, x). One is for the old values (P^{old}, x^{old}) and the other is for the potential next values (P^{new}, x^{new}). The back-end server will first verify the tag using the potential next values. If $\gamma=1$, the server will continue the next steps of the tag authentication phase. If not, the server will try to verify the tag using the old values. Therefore, in our protocol, even though the attacker makes the server and the tag out of synchronization, the server can still authenticate the tag using the old values. Note that if the adversary blocks all response from server to tag through several consecutive sessions, and then corrupts the tag, the adversary can de-synchronize the update and trace the history of the tag during these sessions. In this paper, we assume that the adversary cannot block the two same messages transmitted between the reader and the tag in two consecutive sessions.

5.2.2. Replay Attack Resistance

In our protocol, the random numbers r_R and r_T are used to resist the replay attack. An eavesdropper could store all the messages interchanged between the reader and the tag. After that, the attacker may replay the response M from a tag. However, the back-end server will find the invalidity of the replay value, because the random numbers generated from the reader and the tag are updated each session.

5.2.3. Disclosure Attack Resistance

Under the disclosure attack, an adversary can deduce partial information of the response from the tag through slightly modifying the challenge from the reader [23]. In SRAP, any slight modification on the transmission will be detected. Therefore, our protocol can resist the disclosure attack.

A simple comparison of the privacy and security properties among the proposed SRAP and the previous works [28], [42], [43] is given in Table 3. SLRAP proposed by Tan [28] is a serverless RFID authentication protocol, which provides mutual authentication without the need for a persistent central database. However, the reader needs to achieve an access list of tags from certificate authority (CA) before the mutual authentication. It constrains the authentication flexibility. In addition, SLRAP does not provide de-synchronization resistance and forward security. Recently, some enhanced RFID authentication protocols are proposed in [42], [43]. However, the security and privacy protections of these protocols are still weak. From the comparison Table 3, we conclude that our proposed protocol can provide the strongest privacy and security protections. We'd like to emphasize that using only simple bit-wise operations to achieve RFID authentication may be quite dangerous [26]. Early works proposed by Lopez [9]-[11] are not robust. [23]-[25] pointed out Lopez's series is vulnerable to various attacks. SASI proposed by Chien [25] is also weak. Many researches [26], [27], [44] pointed out SASI is insecure, even though it is quite lightweight. Their works reported the de-synchronization attack, man-in-the-middle attack, tracking attack, and disclosure attack on SASI. In addition, SASI does not support forward security, and the mutual authentication in SASI is incorrect.

Table 3 A Simple Comparison of the Privacy and Security Properties

	SLRAP[28]	He's[42]	Rahman's[43]	SRAP
Tracking attack resistance	Yes[6)]	No	No	Yes
Forward security	No[7)]	Yes	Yes	Yes
De-synchronization attack resistance	No	No	No	Yes
Cloning attack resistance	Yes	Yes	Yes	Yes
Disclosure attack resistance	Yes	No	No	Yes

[6)] "Yes" denotes that the property is satisfied.

[7)] "No" denotes that the property is not satisfied.

6. Performance Analysis

In this section, we analyze the performance of the proposed scheme in terms of computational cost, communication cost and storage requirement. In particular, the proposed protocol is able to provide strong privacy and security with low computational cost. Tables 4-6 summarize the comparison results of the proposed SRAP and the previous works [28], [42], [43]. From Tables 4-6, we can find that our protocol SRAP requires only little resources to perform the authentication.

6.1. Computational Cost

Passive RFID tags are very limited devices with only a small amount of memory and very constrained computational capability. As we know, the computational cost in RFID system consists of the hash operation cost and the bit-wise operation cost. However, the hash operation cost is much higher than the bit-wise operation cost. Compared with the hash operation cost, the bit-wise operation cost is negligible. Therefore, in the comparison of computational cost, we ignore the bit-wise operation cost since only the hash operation cost dominates the computational cost. In our protocol, for each session, the RFID tag requires only 4 hash operations and 9 bit-wise operations. Similarly, the back-end server needs the same number of hash operation and bit-wise operation. Hence, it is easy to infer that the RFID system in our scheme only needs 8 hash operations and 18 bit-wise operations in all. Observed from Table 4, our scheme dramatically reduces the computational cost compared with [28], [42] and [43]. In particular, in [28], [42] and [43], the

RFID system requires $n+3$, $n+5$ and $n+6$ hash operations, respectively. In their protocols, in order to authenticate the tag, the server needs to perform hash operation for each shared key material to match the message transmitted from the tag. When the number of tag is large, our scheme will be more efficient. Note that the computational cost of our scheme is higher than that of Lopez's series [9]-[11] and SASI [25] where only bit-wise operations exist, however the privacy and security of our scheme is much more robust than that of Lopez's series and SASI. We insist that it is quite dangerous using only simple bit-wise operations to achieve RFID authentication. In a word, our protocol SRAP can be efficiently implemented into the RFID applications and provide strong privacy and security.

Table 4 A Comparison of the Computational Cost

Computational cost	SLRAP [28]	He's [42]	Rahman's [43]	SRAP
No. of hash operation	$n+3$[8]	$n+5$	$n+6$	8
No. of bit-wise operation	2	0	$2n+1$	18

6.2. Communication Cost

Regarding the communication cost, we only need to count the messages transmitted between the tag and the reader, which contributes most of the communication cost. It is easy to find that only five messages are transmitted between the tag and the reader, which are r_R , r_T , P, M and N. Each of them is l-bit length, so the total communication demand is $5l$ bits. Seen from the Table 5, our protocol needs low communication cost.

[8] n is the number of tags. n is assumed to be larger than 10.

Table 5 A Comparison of the Communication Cost

	SLRAP [28]	He's [42]	Rahman's [43]	SRAP
Communication cost	$5l$[9]	$7l$	$8l$	$5l$

6.3. Storage Requirement

The storage requirement for each tag in RFID system is composed of the memory size on tag and the memory size for each tag on server. In our protocol, for each session, the tag needs to store P, x and *ID*, which are $3l$ bits in all. Meanwhile, the server needs to keep P^{old}, P^{new}, x^{old}, x^{new} and *ID*, which are $5l$ bits in total. Therefore, the storage requirement for each tag in our protocol is only $8l$. However, the storage requirement for each tag in [28], [42] and [43] is $9l$, $10l$ and $12l$ respectively, where the memory size on tag is $3l$, $5l$ and $6l$ respectively, and the memory size for each tag on server is $6l$, $5l$ and $6l$ respectively. Observed from Table 6, we can know that our protocol requires the least storage requirement for each tag.

Table 6 A Comparison of the Storage Requirement

	SLRAP [28]	He's [42]	Rahman's [43]	SRAP
Storage requirement for each tag	$9l$	$10l$	$12l$	$8l$

[9] l is the bit length of one pseudonym, one random number, one secret key, or one static *ID*.

7. Conclusion

Radio frequency identification (RFID) is the latest technology for object identification, which plays an important role in manufacturing, supply chain management and retail inventory control. Given the myriad essential application areas where RFID can be beneficially used, it also leaves ample opportunities for adversaries to trick the system. Due to the limited computational capabilities of RFID tags, the privacy and security issues of RFID are important and challenging. Lots of authentication protocols have been proposed to enhance the privacy and security of RFID. We believe, in the near future, RFID tags will become ubiquitous just like barcodes.

In this paper, we proposed SRAP, an secure RFID authentication protocol providing strong privacy and security with low cost. Our protocol utilizes the pseudonym rather than the real ID of the tag to perform the authentication. It is impossible for an adversary to identify the tag and track the tag. The major advantage of SRAP is that SRAP can withstand different types of attacks with low cost, which satisfies the requirement of highly resource-constrained RFID tags. It is worth noting that SRAP requires only 8 hash operations and 18 bit-wise operations for computational cost. Meanwhile the communication cost and storage requirement in SRAP are all less than that in the previous researches. We conclude that out protocol is more robust and efficient. These excellent features make it very attractive to low-cost RFID applications.

References

[1] A. Juels, "RFID Security and Privacy: A Research Survey," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381-394, Feb. 2006.

[2] S. L. Garfinkel, A. Juels, and R. Pappu, "RFID Privacy: An Overview of Problems and Proposed Solutions," *IEEE Security & Privacy*, vol. 3, no. 3, pp. 34-43, 2005.

[3] S. Piramuthu, "Lightweight Cryptographic Authentication in Passive RFID-Tagged Systems," *IEEE Trans. on System, Man, and Cybernetics-Pact C: Applications and Reviews*, vol. 38, no. 3, pp. 360-376, May, 2008.

[4] S. A. Weis, S. E. Sarma, R. Rivest, and D. W. Engels, "Security and Privacy Aspects of Low-cost Radio Frequency identification Systems," *Security in Pervasive Comp.*, vol. 2802, pp. 201-212, 2004.

[5] G. Tsudik, "YA-TRAP: Yet Another Trivial RFID Authentication Protocol," *Proc. of the 4th Annu. IEEE Int. Conf. Pervasive Comp. Comm. Workshop (PERCOMW 2006)*, pp. 640-643, 2006.

[6] S. Lee, T. Asano, and K. Kim, "RFID Mutual Authentication Scheme Based on Synchronized Secret Information," *Proc. of Symp. Crptography Info. Security (SCIS)*, Hiroshima, Japan, Jan. 2006.

[7] J. Yang, J. Park, H. Lee, K. Ren, and K. Kim, "Mutual Authentication Protocol for Low-cost RFID," *Proc. of Workshop RFID Lightweight Cryptography*, pp. 17-24, 2005.

[8] S. Karthikeyan and M. Nesterenko, "RFID Security without Extensive Cryptography," *Proc. of the 3rd ACM Workshop Sec. Ad Hoc Sensor Network*, pp. 63-67, 2005.

[9] P. P. Lopez, J. C. H. Castro, J. M. E. Tapiador, and A. Ribagorda, "EMAP: An Efficient Mutual Authentication Protocol for Low-cost RFID Tags," *Proc. of OTM Federated Conf. and Workshop: IS Workshop*, LNCS, vol. 4277, pp. 352-361, 2006.

[10] P. P. Lopez, J. C. H. Castro, J. M. E. Tapiador, and A. Ribagorda, "LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID Tags," presented at *the Workshop RFID Sec.*, Graz, Austria, 2006.

[11] P. P. Lopez, J. C. H. Castro, J. M. E. Tapiador, and A. Ribagorda, "M2AP: A Minimalist Mutual-authentication Protocol for Low-cost RFID Tags," *Proc. of Int. Conf. Ubiquitous Intell. Comp. (UIC2006)*, LNCS, vol. 4159, pp. 912-923, 2006.

[12] C. C. Tan, B. Sheng, and Q. Li, "Serverless Search and Authentication Protocols for RFID," *Proc. of Int. Conf. Pervasive Comp. Commu. (PerCom 2007)*, pp. 3-12, New York, 2007.

[13] M. R. Rieback, B. Crispo, and A. S. Tanenbaum, "The Evolution of RFID Security," *IEEE Pervasive Comp.*, vol. 5, no. 1, pp. 62-69, Jan.-Mar. 2006.

[14] R. Want, ''An Introduction to RFID Technology,'' *IEEE Pervasive Comp.*, pp. 25-33, Jan.-Mar. 2006.

[15] M. Ohkubo, K. Suzuki, and S. Kinoshita, ''A Cryptographic Approach to 'Privacy-Friendly' Tags," presented at the RFID Privacy Workshop, MIT, Cambridge, MA, 2003.

[16] D. Henrici and P. Muller, "Hash-based Enhancement of Location Privacy for Radio Frequency Identification Devices using Varying Identifiers," *Proc. of the 1st Int. Workshop Pervasive Comp. Comm. Sec. (PerSec2004)*, pp. 149-153, 2004.

[17] A. Juels, "Strengthening EPC Tags Against Cloning," *Proc. of ACM Workshop on Wireless Security (WiSe)*, pp. 67-76, 2005.

[18] D. Molnar and D. Wagner, "Privacy and Security in Library RFID: Issues, Practices, and Architectures," *Proc. of the 11th ACM Conf. on Comp. and Comm. Security (CCS'04)*, pp. 210-219, 2004.

[19] A. Juels and S. A. Weis, "Authentication Pervasive Devices with Human Protocols," *Proc. of Crypto'05*, LNCS, vol. 3621, pp. 293-308, 2005.

[20] A. Juels, "'Yoking-Proofs' for RFID tags," *Proc. of the 2nd IEEE Annual Conference on Pervasive Comp. and Comm. Workshops (PERCOMW'04)*, pp. 138-143, 2004.

[21] J. Saito and K. Sakurai, "Grouping Proof for RFID tags," *Proc. of the 19th Int. Conf. on Advanced Info. Networking and Applications (AINA'05)*, pp. 621-624, Mar. 2005.

[22] S. Piramuthu, "On Existence Proofs for Multiple RFID tags," *Proc. of IEEE Int. Conf. on Pervasive Services*, pp. 317-320, 2006.

[23] T. Li and G. Wang, "Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols," *Proc. of IFIP Int. Federation for Info. Processing*, vol. 232, pp. 109-120, May 2007.

[24] T. Li and R. H. Deng, "Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol," *Proc. of the Second Int. Conf. Availability, Reliability, and Security (AReS '07)*, pp. 238-245, 2007.

[25] H. Y. Chien, "SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity," *IEEE Trans. on Dependable and Secure Computing*, Vol. 4, No. 4, pp. 337-340, 2007.

[26] T. Cao, E. Bertino, and H. Lei, "Security Analysis of the SASI Protocol," *IEEE Trans. on Dependable and Secure Computing*, Vol. 6, No. 1, pp. 73-77, 2009.

[27] R. C.-W. Phan, "Cryptanalysis of a New Ultralightweight RFID Authentication Protocol-SASI," *IEEE Trans. on Dependable and Secure Computing*, Vol. 6, No. 4, pp. 316-320, 2009.

[28] C. C. Tan, B. Sheng, and Q. Li, "Secure and Severless RFID Authentication and Search Protocols," *IEEE Trans. on Wireless Comm.*, Vol. 7, No. 4, pp. 1400-1407, Apr. 2008.

[29] K. Yuksel, "Universal Hashing for Ultra-Low-Power Cryptographic Hardware Applications," *Master's Thesis*, Dept. of Electronical Engineering, WPI, 2004.

[30] Y. Liu, "An Efficient RFID Authentication Protocol for Low-Cost Tags," *Proc. of IEEE/IFIP Int. Conf. on Embedded and Ubiquitous Computing (EUC'08)*, pp. 180-185, 2008.

[31] Y. Lee, Y. Hsieh, P. You, and T. Chen, "An Improvement on RFID Authentication Protocol with Privacy Protection," *Proc. of Int. Conf. on Convergence and Hybrid Information Technology (ICCIT'08)*, pp. 569-573,

2008.

[32] J. Shen, D. Choi, S. Moh, and I. Chung, "A Novel Anonymous RFID Authentication Protocol Providing Strong Privacy and Security," *Proc. of the 2nd Int. Conf. on Multimedia Info. Networking & Security (MINES 2010)*, pp. 584-588, 2010.

[33] M. Bellare and P. Rogaway, "Entity Authentication and Key Distribution," *Proc. of Advances in Cryptology-Crypto'93*, LNCS, vol. 773, pp. 232-249, 1993.

[34] G. Avoine, "Adversarial Model for Radio Frequency Identification," Cryptology ePrint Archive, report 2005/049, 20 Feb. 2005. Available at IACR ePrint Archive, http://eprint.iacr.org/2005/049

[35] A. Juels and S. A. Weis, "Defining Strong Privacy for RFID," *Proc. of PerCom '07*, pp. 342-347, 2007.

[36] S. Vaudenay, "On Privacy Models for RFID," *Proc. of Advances in Cryptology - Asiacrypt '07*, LNCS vol. 4833, pp. 68-87, 2007.

[37] K. Ouafi and R. C. W. Phan, "Privacy of Recent RFID Authentication Protocols," *Proc. of Information Security Practice and Experience*, LNCS vol. 4991, pp. 263-277, 2008.

[38] M. Naor and M. Yung, "Public-Key Cryptosystems Provably Secure against Chosen Cipertext Attacks," *Proc. of ACM Symposium Theory of Computing (STOC'90)*, pp. 427-437, 1990.

[39] C. Rackoff and D. Simon, "Non-interactive Zero-Knowledge Proof of Knowledge and Chosen Cipertext Attacks," *Proc. of the 11th Ann. Int. Cryptology Conference (CRYPTO'91)*, pp. 433-444, 1991.

[40] M. Bellare, R. Canetti, and H. Krawczyk, "A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols," *Proc. of the 30th Annual ACM Symposium on the Theory of Computing*, pp. 419-428, 1998.

[41] V. Shoup, "On formal Models for Secure Key Exchange," *IBM Research Report RZ 3120 (version 4)*, Nov. 1999.

[42] L. He, S. Jin, T. Zhang, and N. Li, "An Enhanced 2-Pass Optimistic

Anonymous RFID Authentication Protocol with Forward Security," *Proc. of the 5th Int. Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1-4, 2009.

[43] M. S. Rahman, M. Soshi, and A. Miyaji, "A Secure RFID Authentication Protocol with Low Communication cost," *Proc. of Int. Conference on Complex, Intelligent and Software Intensive Systems (CISIS'09)*, pp. 559-564, 2009.

[44] P. D'Arco and A. D. Santis, "On Ultralightweight RFID Authentication Protocol," *IEEE Trans. on Dependable and Secure Computing*, Vol. 8, No. 4, pp. 548-563, 2011.

[45] H. Mobahat, "Authentication and Lightweight Cryptography in Low Cost RFID," *Proc. of the 2nd International Conference on Software Technology and Engineering (ICSTE)*, Vol. 2, pp. 123-129, Oct. 2010.

[46] G. Bianchi, "Revisiting an RFID Identification-Free Batch Authentication Approach," *IEEE Communications Letters*, Vol. 2, No. 6, pp. 632-634, 2011.

[47] L. H. Chan and Y. J. Hyun, "Development of Privacy-Preserving RFID Authentication System using Mobile Devices," *Proc. of International Conference on ICT Convergence (ICTC)*, pp. 760-765, 2011.

[48] H. Y. Jung, J. C. Huan, W. H. Hsun, H. Y. Hao, and L. K. Jen, "Mutual Authentication Protocol for RFID System," *Proc. of IEEE 14th International Conference Computational Science and Engineering (CSE)*, pp. 73-80, 2011.

감사의 글

이 논문이 완성되기까지의 모든 과정 속에서, 많이 부족한 저를 항상 지키고 돌보아 주시는 하나님께 먼저 감사를 드립니다. 학위과정 내내 물심양면으로 지원을 아끼지 않으시고 많은 지도와 가르침을 베풀어 주신 정일용 교수님께 감사를 드립니다. 바쁘신 와중에도 논문의 부족한 부분을 지적해 주시며 완성할 수 있도록 도와주신 모상만 교수님과 신석주 교수님, 강문수 교수님께 깊은 감사를 드립니다. 또한 지도 교수님 이상으로 논문의 부족한 점을 알려주시고 완성도를 높게 해 주신 국가보안기술연구소의 이철원 본부장님께도 깊은 감사를 드립니다.

연구실에서 학문에 전념할 수 있도록 신경 써 주시며 도와주신 조영주 박사님, 연구실의 마스코트이시며 곧 결혼하실 쟈스민 탁동길 박사님, 소소한 부분까지 신경써 주시며 도와주신 강정희 선생님, 바쁜 업무중에도 묵묵히 도와주신 조윤옥 선생님, 석사때부터 함께 연구실을 지켜온 최동민과 지금은 없지만 함께 하면서 정이 들었던 조충용 그리고 소신에게도 감사를 드립니다. 또한 늘 가족과 같은 분위기로 지내왔던 컴퓨터공학과 대학원 선후배님들에게도 감사를 드립니다.

제 삶의 절반인 교회에서 큰 어른으로 타인의 모범이 되어주시는 차채원, 김정수, 윤병섭 장로님과 신정님 전도사님, 지금 자리에는 없지만 타지에서 묵묵히 주님 일에 힘쓰시는 이복음 김소망 선교사님께 깊은 감사의 마음을 전합니다. 또한 젊은 세대 신앙의 모범으로서 자기 자리를 지키시는 황도웅 선생님에게 깊은 감사를 드립니다. 귀여운 동생이며 이제 학문의 초입에 들어서는 박경호, 끊임없는 성실함이 장점인 조영창, 조영만에게 감사를 드립니다. 누구보다 가장 친한 친우로서 모든 일에 조언을 아끼지 않고 자기 일처럼 함께 해준 윤호철에게 감사의 마음을 전합니다. 깊은 사랑으로 도와주며 신앙의 길을 함께 가는 자매인 서평에게 진심으로 감사를 드립니다.

저를 세상에 있게 하여 주시며 큰 힘이 되어 주신 사랑하는 부모님과 동생 유난이와 고명 형제 가정에게도 감사의 마음을 전하며, 끝으로 이 모든 기쁨과 감사의 마음을 담아 주님께 올려드립니다.

저작물 이용 허락서

학 과	컴퓨터공학과	학 번	20097764	과 정	박사
성 명	한글: 심 검 　 한문: 沈 劍 　 영문: Jian Shen				
주 소	광주광역시 동구 서석동 조선대학교 전자정보공과대학 정보통신실험실 8103 호				
연락처	e-mail : s_shenjian@126.com				
논문제목	한글: 낮은 연산 비용을 갖는 랜덤 오라클 모델의 안전한 RFID 인증 프로토콜				
	영문: A Secure RFID Authentication Protocol in Random Oracle Models with Low Computational Cost				

본인이 저작한 위의 저작물에 대하여 다음과 같은 조건 아래 조선대학교가 저작물을 이용할 수 있도록 허락하고 동의합니다.

- 다 음 -

1. 저작물의 DB구축 및 인터넷을 포함한 정보통신망에의 공개를 위한 저작물의 복제, 기억장치에의 저장, 전송 등을 허락함.
2. 위의 목적을 위하여 필요한 범위 내에서의 편집과 형식상의 변경을 허락함. 다만, 저작물의 내용변경은 금지함.
3. 배포・전송된 저작물의 영리적 목적을 위한 복제, 저장, 전송 등은 금지함.
4. 저작물에 대한 이용기간은 5년으로 하고, 기간종료 3개월 이내에 별도의 의사 표시가 없을 경우에는 저작물의 이용기간을 계속 연장함.
5. 해당 저작물의 저작권을 타인에게 양도하거나 출판을 허락을 하였을 경우에는 1개월 이내에 대학에 이를 통보함.
6. 조선대학교는 저작물 이용의 허락 이후 해당 저작물로 인하여 발생하는 타인에 의한 권리 침해에 대하여 일체의 법적 책임을 지지 않음.
7. 소속 대학의 협정기관에 저작물의 제공 및 인터넷 등 정보통신망을 이용한 저작물의 전송・출력을 허락함.

동의여부 : 동의(○) 반대( )

2012 년 4 월

저작자 : 심 검 (인)

조선대학교 총장 귀하