A New Formal Proof Model for RFID Location Privacy

JungHoon Ha[1], SangJae Moon[1], Jianying Zhou[2], and JaeCheol Ha[3]

[1] Kyungpook National University, Korea
{short98, sjmoon}@ee.knu.ac.kr
[2] Institute for Infocomm Research, Singapore
jyzhou@i2r.a-star.edu.sg
[3] Hoseo University, Korea
jcha@hoseo.edu

Abstract. The privacy and security problems in RFID systems have been extensively studied. However, less research has been done on formal analysis of RFID security. The existing adversarial models proposed in the literature have limitations for analyzing RFID location privacy. In this paper, we propose a new formal proof model based on random oracle and indistinguishability. It not only considers passive/active attacks to the message flows between RFID reader and tag, but also takes into account physical attacks for disclosing tag's internal state, thus making it more suitable for real RFID systems. We further apply our model to analyze location privacy of an existing RFID protocol.

Keywords: RFID security, location privacy, formal proof model.

1 Introduction

Radio Frequency Identification (RFID) systems are a new form of automatic identification technology involving the use of small devices called RFID tags. They are expected to replace optical barcodes due to several important advantages including small size, quick identification, and invisible implementation within objects. An RFID system consists of RFID tags, an RFID reader, and a back-end database. As the RFID reader communicates with the tags using RF signals, RFID protocols may face various security threats such as location privacy, authentication, and resynchronization between two entities. Much attention has been devoted to RFID security, and various schemes have been proposed. Nevertheless, most of RFID security research lacks formal analysis, therefore existing work mainly offers ad hoc notions of security [8].

Location privacy is one of the most important security requirements in an RFID system. The existing adversarial models proposed in the literature [1,8,19] have limitations in the analysis of RFID location privacy. In fact, Avoine's model [1] only captures a range of adversarial ability using some queries. Juels-Weis's model [8] is more specific and practical regarding the adversarial computation boundary. However, when analyzing the randomized hash-lock protocol with

S. Jajodia, and J. Lopez (Eds.): ESORICS 2008, LNCS 5283, pp. 267–281, 2008.

their model, it confirmed location privacy, whereas the protocol is known to be vulnerable to location tracking as a tag's *ID* is sent to the tag through an insecure wireless channel [16]. In addition, they did not define a concrete attack game for forward secrecy in RFID location privacy. More recently, Vaudenay has presented the classification of privacy in RFID [19] and shown narrow-destructive privacy for Ohkubo-Suzuki-Kinoshita (OSK) protocols [14, 15] in the random oracle model, so that the strong privacy is indeed not achievable in RFID.

In this paper, we present a formal definition of provable location privacy for an RFID system. Our adversarial model is more suitable for a real RFID system as it not only considers passive/active attacks to the message flows between RFID reader and tag, but also takes into account physical attacks for disclosing tag's internal state. It is based on the random oracle model and indistinguishability that is reminiscent of the classic indistinguishability under chosen-plaintext and chosen-ciphertext attacks in a cryptosystem's security game.

The rest of this paper is structured as follows. Section 2 explains the adversarial types and security requirements in an RFID system. Section 3 defines the security model for satisfying those requirements. Section 4 presents our formal definition for location privacy of an RFID system. Section 5 analyzes location privacy of an RFID protocol LRMAP [4] with our formal proof model. Final conclusions are given in Section 6.

2 Adversarial Types

We consider two types of adversaries in RFID systems.

- **Passive Adversary** $\mathcal{A}_{\mathcal{P}}$: $\mathcal{A}_{\mathcal{P}}$ eavesdrops all communications among a tag, a reader and a database. $\mathcal{A}_{\mathcal{P}}$ tries to find out a secret key or useful information of the targeted tag. However, $\mathcal{A}_{\mathcal{P}}$ cannot insert or alter any message in communication.
- **Active Adversary** $\mathcal{A}_{\mathcal{A}}$: $\mathcal{A}_{\mathcal{A}}$ can insert or modify any message in addition to eavesdropping. That is, $\mathcal{A}_{\mathcal{A}}$ impersonates a legal reader or tag by replay attack or spoofing attack, and causes de-synchronization between back-end database and a tag by message interruption or jamming. Moreover, $\mathcal{A}_{\mathcal{A}}$ also tries to find out a secret key or useful information like $\mathcal{A}_{\mathcal{P}}$.

Since the communication between a reader and a tag is performed using a wireless interface, the communicated data can be easily tapped by an attacker $\mathcal{A}$. Therefore, RFID protocols need to satisfy various security requirements as identified in the literature [9,12,18]. In particular, they should be designed secure against the following attacks.

- **Eavesdropping:** An adversary $\mathcal{A}_{\mathcal{P}}$ or $\mathcal{A}_{\mathcal{A}}$ can eavesdrop messages transmitted between the reader and tags via wireless communication, and tries to find out the secret key or other information like tag *ID*. With those information, An active adversary $\mathcal{A}_{\mathcal{A}}$ can further perform other enhanced attacks, such as replay attack or spoofing attack.

- **Impersonation:** An active adversary $\mathcal{A}_{\mathcal{A}}$ impersonates a legal reader or a legal tag by replay attack or spoofing attack so that it passes the authentication protocol/phase between the back-end database and a tag.
- **Message Interruption and Loss:** Since the communication between a reader and tags is performed wirelessly, the possibility of message loss is higher than with wired communication due to system malfunction or communication error. When an attacker $\mathcal{A}_{\mathcal{A}}$ tries to block the service by jamming, the communicated message between the reader and tags can be interrupted, and a message interruption or loss will cause a state of de-synchronization between the tag and the back-end database. We call this message interruption by malicious $\mathcal{A}_{\mathcal{A}}$ a *de-synchronizaton attack*.
- **Location Tracking:** An adversary $\mathcal{A}_{\mathcal{A}}$ may try to trace the location of a tag based on the interactions with it. For perfect untraceability, RFID protocols must satisfy *indistinguishability* [14] and *forward secrecy* [1, 14]. Indistinguishability means the values emitted by one tag should not be distinguishable from the values emitted by other tags. Forward secrecy means even if the adversary acquires the secret data stored in a tag, the tag's location cannot be traced back using previously known messages. Here we define *weak location privacy* that only satisfies indistinguishability while *strong location privacy* meets both indistinguishability and forward secrecy.

3 Security Model

For simplicity, we assume a fixed, polynomial-size tag set $\mathcal{TS} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$, a reader $\mathcal{R}$ and a back-end database $\mathcal{DB}$ as the elements for an RFID system: $\mathcal{S} = \{\mathcal{TS}, \mathcal{R}, \mathcal{DB}\}$. We do not assume that these subsets always have the same size or always include the same elements. A back-end database $\mathcal{DB}$ has information for $\mathcal{TS}$'s authentication such as tag's *ID*, state value and session id, etc. Before the protocol is run for the first time, an initialization phase occurs in both $\mathcal{T}_l$ and $\mathcal{DB}$, where $l = 1, \ldots, n$. That is, each $\mathcal{T}_l \in \mathcal{TS}$ runs an algorithm $\mathcal{G}(1^k)$ to generate the secret key k_l or identity ID_l, and $\mathcal{DB}$ also saves these values in a database field.

The research for secure RFID systems can be mainly categorized into *physical technologies* and *protocol-based techniques*. The first category includes 'Kill command' [21], 'Active jamming' [7] and 'Blocker tag' [7] approaches. The second category is further classified into three types, i.e., *hash-based protocol* [5, 15, 16, 18, 20, 21], *re-encryption protocol* [3, 6, 17] and *partial identity based protocol* [10, 11]. We do not consider the physical approaches but treat the weakness of protocol-based techniques in this paper.

Fig 1 represents the general structure of RFID protocols with 3 rounds and based on challenge-response.

$\mathcal{R}$ and $\mathcal{DB}$ can execute the protocol multiple times with different tags, which is modeled by allowing each principal an unlimited number of *instances* to execute the protocol. We denote instance i of entity $\mathcal{E}$ as $\mathcal{E}^i$ to represent a flow originating

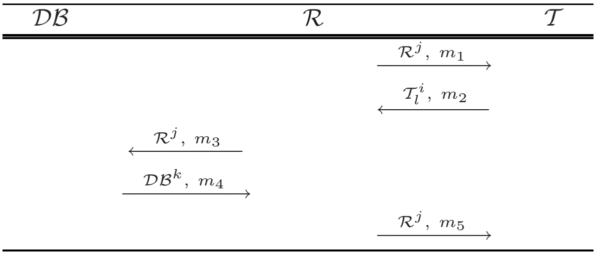

Fig. 1. General model of 3-round RFID protocols

from entity $\mathcal{E}$, where $\mathcal{E} \in \{\mathcal{T}_l, \mathcal{R}, \mathcal{DB}\}$ and $\mathcal{T}_l \in \mathcal{TS}$. Note, a given instance may only be used once.

The adversary $\mathcal{A}$ is assumed to have complete control over all communications in the protocol. In Fig 1, the flows for each round of a protocol are sent and controlled by the adversary in the adversarial model. $\mathcal{A}$'s interaction with the RFID entities in the network is modeled by sending the following queries to a oracle $\mathcal{O}$ and receiving the result from $\mathcal{O}$.

- $\mathsf{Query}(\mathcal{R}^j, m_1)$: It calls instance $\mathcal{R}^j$, and outputs m_1.
- $\mathsf{Reply}(\mathcal{T}_l^i, m_1', m_2)/\mathsf{Reply}^*(\mathcal{T}_l^i, m_1', m_2)$: It calls instance $\mathcal{T}_l^i$ with input m_1', and outputs m_2. We consider two cases, query for normal state and query for abnormal state, according to the state of previous session of RFID protocol. In fact, according to the authentication result of RFID protocol [4, 12], different messages can be sent to $\mathcal{R}$ in response to the query from the reader, which can influence the ways, effort, and abilities of an adversary for attacking an RFID protocol. $\mathsf{Reply}()$ means RFID protocol finished successfully at the previous $\mathcal{T}_l^{i-1}$, while $\mathsf{Reply}^*()$ considers RFID protocol failed in the previous $\mathcal{T}_l^{i-1}$ [1]. If a scheme sends response regardless of a tag's authentication result of the previous session, we only consider Reply since $\mathsf{Reply} = \mathsf{Reply}^*$.
- $\mathsf{Forward}_1(\mathcal{R}^j, m_2', m_3)$: It calls instance $\mathcal{R}^j$ with input m_2', and outputs m_3. This oracle models that $\mathcal{R}$ transmits the message received from a tag in response of $\mathcal{R}$'s query to $\mathcal{DB}$ in real RFID protocol.
- $\mathsf{Auth}(\mathcal{DB}^k, m_3', m_4)$: When receiving this call with input m_3', it outputs m_4. This oracle models that $\mathcal{DB}$ sends the authentication result of a tag to $\mathcal{R}$ in real RFID protocol.
- $\mathsf{Forward}_2(\mathcal{R}^j, m_4', m_5)$: It calls instance $\mathcal{R}^j$ with input m_4', and outputs m_5. This oracle considers that $\mathcal{R}$ forwards the authentication result received from $\mathcal{DB}$ to $\mathcal{T}$ in real RFID protocol.
- $\mathsf{Execute}(\mathcal{T}_l^i, \mathcal{R}^j, \mathcal{DB}^k)/\mathsf{Execute}^*(\mathcal{T}_l^i, \mathcal{R}^j, \mathcal{DB}^k)$: This oracle is defined to model $\mathcal{A}$'s eavesdropping of communicated messages. It executes RFID protocol

[1] Even though several instances can arise in the same session, for simplicity, we assume ith instance is for current session, while $(i-1)$th instance is for the previous session throughout this paper. That is, it is assumed only one instance is allowed for each session.

among unused instances of entities $\mathcal{T}_l^i \in \mathcal{TS}, \mathcal{R}^j$, and $\mathcal{DB}^k$. Execute and Execute* have the following relation with the previously defined oracles.

- Execute$(\mathcal{T}_l^i, \mathcal{R}^j, \mathcal{DB}^k)$
=Query$(\mathcal{R}^j, m_1)$ $\wedge$ Reply$(\mathcal{T}_l^i, m_1', m_2)$ $\wedge$ Forward$_1(\mathcal{R}^j, m_2', m_3)$ $\wedge$ Auth$(\mathcal{D}B^k, m_3', m_4)$ $\wedge$ Forward$_2(\mathcal{R}^j, m_4', m_5)$,
where $m_1 = m_1', m_2 = m_2', m_3 = m_3'$ and $m_4 = m_4'$.

- Execute*$(\mathcal{T}_l^i, \mathcal{R}^j, \mathcal{D}B^k)$
=Query$(\mathcal{R}^j, m_1)$ $\wedge$ Reply*$(\mathcal{T}_l^i, m_1', m_2)$ $\wedge$ Forward$_1(\mathcal{R}^j, m_2', m_3)$ $\wedge$ Auth$(\mathcal{D}B^k, m_3', m_4)$ $\wedge$ Forward$_2(\mathcal{R}^j, m_4', m_5)$,
where $m_1 = m_1', m_2 = m_2', m_3 = m_3'$ and $m_4 = m_4'$.

While Execute() considers the RFID protocol in the normal state, Execute*() executes the RFID protocol for the abnormal state.

– Reveal$(\mathcal{T}_l, i)$: It outputs all internal state of $\mathcal{T}_l$'s ith instance $\mathcal{T}_l^i$, such as tag's *ID*, secret key, and session id, etc. In real RFID systems, the useful internal information for $\mathcal{A}$ can be revealed by a physical attack.

– Test$(\mathcal{T}_l, i)$: This query is allowed only once at any time during $\mathcal{A}$'s execution. A random bit b is generated; if $b = 1$ $\mathcal{A}$ is given a message m corresponding to $\mathcal{T}_l^i$, and if $b = 0$ $\mathcal{A}$ receives a random value [2].

A *passive adversary* $\mathcal{A}_{\mathcal{P}}$ is given access to Execute$(\mathcal{T}_l^i, \mathcal{R}^j, \mathcal{D}B^k)$, Execute*$(\mathcal{T}_l^i, \mathcal{R}^j, \mathcal{D}B^k)$, Reveal$(\mathcal{T}_l, i)$ and Test$(\mathcal{T}_l, i)$ queries, while an *active adversary* $\mathcal{A}_{\mathcal{A}}$ is additionally given access to Query$(\mathcal{R}^j, m_1)$, Reply$(\mathcal{T}_l^i, m_1', m_2)$, Reply*$(\mathcal{T}_l^i, m_1', m_2)$, Forward$_1(\mathcal{R}^j, m_2', m_3)$, Auth$(\mathcal{D}B^k, m_3', m_4)$ and Forward$_2(\mathcal{R}^j, m_4', m_5)$ queries.

4 Definition of Location Privacy

Now we give the formal definitions of location privacy for RFID systems using the queries defined in the previous section. Note, we only consider Execute, Execute*, Query, Reply, Reply*, and Forward$_2$. Forward$_1$ and Auth are not needed for location privacy, as the communication between $\mathcal{R}$ and $\mathcal{T}$ is performed with an insecure air interface, while the communication between $\mathcal{DB}$ and $\mathcal{R}$ is assumed to be a secure channel. Therefore, only the queries modeling an insecure channel are considered for the case of location privacy [3]. Hereinafter, for simplicity, it is also assumed that $m_1 = m_1'$, $m_2 = m_2'$, $m_3 = m_3'$ and $m_4 = m_4'$ in the defined oracles.

[2] In this paper, the random value means an arbitrary value unrelated to the message outputted from an attack-target tag in real-world RFID system. It follows a uniform distribution [13] and its bit length depends on RFID protocols.

[3] In fact, Forward$_1$ and Auth could be used to define an authentication model for RFID systems when inducing the notion of matching conversation proposed by Bellare and Rogaway [2], but this will be left for future work.

Attack Game $\mathcal{AG}^{\mathsf{Indis}}$ for RFID Location Privacy: Indistinguishability

Attack Game $\mathcal{AG}^{\mathsf{Indis}}_{\mathcal{A},S}[e, r, f, k]$

Phase 1: Initialization

(1) Run algorithm $\mathcal{G}(1^k) \rightarrow (k_1, \ldots, k_n)$.
(2) Set each $\mathcal{T}_l$'s secret key as k_l, where $\mathcal{T}_l \in \mathcal{TS} = \{\mathcal{T}_1, \ldots \mathcal{T}_n\}$.
(3) Save each $\mathcal{T}_l$'s k_l generated in step (1) in $\mathcal{DB}$'s field.

Phase 2: Learning

(1) $\mathcal{A}$ executes oracles for all $n-1$ tags, except only one $\mathcal{T}_c \in \mathcal{TS}$ used in challenge phase.
 i. $\mathcal{A}_{\mathcal{P}}$ calls $\mathsf{Execute}(\mathcal{T}_l^i, \mathcal{R}^j, \mathcal{DB}^k)$ and $\mathsf{Execute^*}(\mathcal{T}_l^i, \mathcal{R}^j, \mathcal{DB}^k)$.
 ii. $\mathcal{A}_{\mathcal{A}}$ calls $\mathsf{Query}(\mathcal{R}^j, m_1)$, $\mathsf{Reply}(\mathcal{T}_l^i, m_1, m_2)$, $\mathsf{Reply^*}(\mathcal{T}_l^i, m_1, m_2)$ and $\mathsf{Forward}_2(\mathcal{R}^j, m_4, m_5)$ additionally.

Phase 3: Challenge

(1) $\mathcal{A}$ selects a challenge tag $\mathcal{T}_c$ from $\mathcal{TS}$.
(2) $\mathcal{A}$ executes oracles except $\mathsf{Reveal}(\mathcal{T}_c, i)$ for $\mathcal{T}_c$, where $i = 1, \ldots, q-1$.
 i. $\mathcal{A}_{\mathcal{P}}$ calls $\mathsf{Execute}(\mathcal{T}_c^i, \mathcal{R}^j, \mathcal{DB}^k)$ and $\mathsf{Execute^*}(\mathcal{T}_c^i, \mathcal{R}^j, \mathcal{DB}^k)$.
 ii. $\mathcal{A}_{\mathcal{A}}$ calls $\mathsf{Query}(\mathcal{R}^j, m_1)$, $\mathsf{Reply}(\mathcal{T}_c^i, m_1, m_2)$, $\mathsf{Reply^*}(\mathcal{T}_c^i, m_1, m_2)$ and $\mathsf{Forward}_2(\mathcal{R}^j, m_4, m_5)$ additionally.
(3) $\mathcal{A}$ calls the oracle $\mathsf{Test}(\mathcal{T}_c, q)$.
(4) For the $\mathcal{A}$'s Test, Oracle $\mathcal{O}$ tosses a fair coin $b \in \{0, 1\}$; let $b \overset{R}{\longleftarrow} \{0, 1\}$.
 i. If $b = 1$, $\mathcal{A}$ is given the message corresponding to $\mathcal{T}_c$'s qth instance.
 ii. If $b = 0$, $\mathcal{A}$ is given a random value.
(5) $\mathcal{A}$ outputs a guess bit b'.

$\mathcal{A}$ wins $\mathcal{AG}^{\mathsf{Indis}}_{\mathcal{A},S}$ if $b = b'$.

Fig. 2. Attack game between an adversary and oracles for indistinguishability

We now present an "Attack Game $\mathcal{AG}$ for Provable Location Privacy in an RFID System", reminiscent of the classic indistinguishability under a chosen-plaintext attack (IND-CPA) and chosen-ciphertext attack (IND-CCA) in a cryptosystem security game.

The goal of the adversary $\mathcal{A}$ in this game is to distinguish two different values within the limits of $\mathcal{A}$'s computational boundary. In other words, the success of $\mathcal{A}$ in $\mathcal{AG}$ is quantified in terms of $\mathcal{A}$'s advantage in distinguishing whether $\mathcal{A}$ receives an RFID tag's real response or a random value.

Considering both weak location privacy and strong location privacy of RFID systems described in Section 2, we define two different attack games between an adversary and oracles: $\mathcal{AG}^{\mathsf{Indis}}$ and $\mathcal{AG}^{\mathsf{FS}}$. Fig 2 shows how the adversary $\mathcal{A}$ runs the attack game $\mathcal{AG}^{\mathsf{Indis}}$ between the adversary and oracles for indistinguishability, while Fig 3 represents the attack game $\mathcal{AG}^{\mathsf{FS}}$ for forward secrecy. The difference between two games resides in the challenge phase: (1) Reveal

```
Attack Game $\mathcal{AG}^{\mathsf{FS}}$ for RFID Location Privacy: Forward Secrecy

Attack Game $\mathcal{AG}^{\mathsf{FS}}_{\mathcal{A},S}[e, r, f, k]$

  Phase 1: Initialization
    (1) Run algorithm $\mathcal{G}(1^k) \rightarrow (k_1, \ldots, k_n)$.
    (2) Set each $\mathcal{T}_l$'s secret key as $k_l$, where $\mathcal{T}_l \in \mathcal{TS} = \{\mathcal{T}_1, \ldots \mathcal{T}_n\}$.
    (3) Save each $\mathcal{T}_l$'s $k_l$ generated in step (1) in $\mathcal{DB}$'s field.
  Phase 2: Learning
    (1) $\mathcal{A}$ executes oracles for all $n-1$ tags, except only one $\mathcal{T}_c \in \mathcal{TS}$ used in
        challenge phase.
        i. $\mathcal{A}_{\mathcal{P}}$ calls $\mathsf{Execute}(\mathcal{T}_l^i, \mathcal{R}^j, \mathcal{DB}^k)$ and $\mathsf{Execute*}(\mathcal{T}_l^i, \mathcal{R}^j, \mathcal{DB}^k)$.
        ii. $\mathcal{A}_{\mathcal{A}}$ calls $\mathsf{Query}(\mathcal{R}^j, m_1)$, $\mathsf{Reply}(\mathcal{T}_l^i, m_1, m_2)$, $\mathsf{Reply*}(\mathcal{T}_l^i, m_1, m_2)$ and
            $\mathsf{Forward}_2(\mathcal{R}^j, m_4, m_5)$ additionally.
  Phase 3: Challenge
    (1) $\mathcal{A}$ selects a challenge tag $\mathcal{T}_c$ from $\mathcal{TS}$.
    (2) $\mathcal{A}$ executes oracles including $\mathsf{Reveal}(\mathcal{T}_c, i)$ for $\mathcal{T}_c$'s $i$th instance.
        i. $\mathcal{A}_{\mathcal{P}}$ calls $\mathsf{Execute}(\mathcal{T}_c^i, \mathcal{R}^j, \mathcal{DB}^k)$, $\mathsf{Execute*}(\mathcal{T}_c^i, \mathcal{R}^j, \mathcal{DB}^k)$ and $\mathsf{Reveal}(\mathcal{T}_c, i)$.
        ii. $\mathcal{A}_{\mathcal{A}}$ calls $\mathsf{Query}(\mathcal{R}^j, m_1)$, $\mathsf{Reply}(\mathcal{T}_c^i, m_1, m_2)$, $\mathsf{Reply*}(\mathcal{T}_c^i, m_1, m_2)$ and
            $\mathsf{Forward}_2(\mathcal{R}^j, m_4, m_5)$ additionally.
    (3) $\mathcal{A}$ calls the oracle $\mathsf{Test}(T_c, i-1)$.
    (4) For the $\mathcal{A}$'s $\mathsf{Test}$, Oracle $\mathcal{O}$ tosses a fair coin $b \in \{0, 1\}$; let $b \overset{R}{\longleftarrow} \{0, 1\}$.
        i. If $b = 1$, $\mathcal{A}$ is given the message corresponding to $\mathcal{T}_c$'s $i-1$th instance.
        ii. If $b = 0$, $\mathcal{A}$ is given a random value.
    (5) $\mathcal{A}$ executes oracles for $n-1$ tags of $\mathcal{TS}$ except $\mathcal{T}_c$ like learning phase.
    (6) $\mathcal{A}$ outputs a guess bit $b'$.

$\mathcal{A}$ wins $\mathcal{AG}^{\mathsf{FS}}_{\mathcal{A},S}$ if $b = b'$.
```

Fig. 3. Attack game between an adversary and oracles for forward security

query's possibility: Reveal is allowed in $\mathcal{AG}^{\mathsf{FS}}$, however, $\mathcal{AG}^{\mathsf{Indis}}$ prohibits it; (2) the applied instance's range: oracles related to the instances from 1 to $q-1$ are executed in $\mathcal{AG}^{\mathsf{Indis}}$, while $\mathcal{AG}^{\mathsf{FS}}$ executes oracles only for the ith instance; (3) additional learning phase: $\mathcal{AG}^{\mathsf{FS}}$ allows an additional learning phase, while $\mathcal{AG}^{\mathsf{Indis}}$ does not need it because the oracles' executions from 1 to $q-1$ have already been performed.

According to indistinguishability and forward secrecy, we formally define the notion of weak location privacy and strong location privacy for RFID systems.

Definition 1. (Weak Location Privacy: Indistinguishability). *An RFID protocol is secure for distinguishability if $\mathcal{A}$'s advantage for correctly guessing b' in $\mathcal{AG}^{\mathsf{Indis}}$, $\mathsf{Adv}^{\mathsf{Indis}}_{\mathcal{A},\mathcal{S}}(k) \overset{\mathrm{def}}{=} |2 \cdot \Pr[b = b'] - 1|$, is negligible for all PPT (Probabilistic Polynomial-Time) adversaries $\mathcal{A}$ ($\mathcal{A}_{\mathcal{P}}$ or $\mathcal{A}_{\mathcal{A}}$) with computational boundary e, r, f and k, where e, r, f and k is the number of $\mathsf{Execute}$ or $\mathsf{Execute*}$, Reply*

or Reply, Forward$_2$ and security parameter, thereby guaranteeing weak location privacy.*

Definition 2. (Forward Secrecy). *An RFID protocol guarantees forward secrecy if $\mathcal{A}$'s advantage in successfully guessing b' in $\mathcal{AG}^{\mathsf{FS}}$, for all PPT adversaries $\mathcal{A}$ ($\mathcal{A}_{\mathcal{P}}$ or $\mathcal{A}_{\mathcal{A}}$) with computational boundary e, r, f and k, $\mathsf{Adv}^{\mathsf{FS}}_{\mathcal{A},\mathcal{S}}(k) \overset{def}{=} |2 \cdot Pr[b = b'] - 1|$ is negligible, where e, r, f and k is the number of Execute or Execute*, Reply or Reply*, Forward$_2$ and security parameter.*

Definition 3 (Strong Location Privacy: Indistinguishability and Forward Secrecy) *An RFID protocol satisfies strong location privacy when both indistinguishability and forward secrecy are guaranteed for all PPT adversaries $\mathcal{A}$ ($\mathcal{A}_{\mathcal{P}}$ or $\mathcal{A}_{\mathcal{A}}$) with computational boundary e, r, f and k, where e, r, f and k is the number of Execute or Execute*, Reply or Reply*, Forward$_2$ and security parameter.*

5 Analysis of an RFID Protocol

Ha *et al.* [4] proposed a lightweight and resynchronous mutual authentication protocol (LRMAP) for RFID systems. Their scheme has been analyzed informally regarding the tag user's location privacy. Here we analyze the scheme again with the attack games under our formal model defined in Section 4.

5.1 LRMAP

The following notations are used for the entities and computational operations to simplify the description.

ID	:	identity of a tag, k bits
HID	:	hashed value of *ID*, k bits
PID	:	previous identity of a tag used in previous session, k bits
r_R	:	random number generated by reader $\mathcal{R}$
r_T	:	random number generated by tag $\mathcal{T}$
Query	:	request generated by $\mathcal{R}$
SYNC	:	parameter used to check whether both $\mathcal{T}$ and $\mathcal{DB}$ succeeded in *ID* updating simultaneously or not, 1 bit
$H()$	:	one-way hash function, $H : \{0,1\}^* \rightarrow \{0,1\}^k$
$L(m)$	:	left half of input message m
$R(m)$	:	right half of input message m
$\|\|$	:	concatenation of two inputs
$\overset{?}{=}$	:	comparison of two inputs

In LRMAP (see Fig 4), $\mathcal{DB}$ manages *ID*, *HID* and *PID* for each $\mathcal{T}$ in $\mathcal{DB}$'s field. According to the state of $\mathcal{T}$'s previous session, $\mathcal{DB}$ finds *ID* for the current session or *PID* used for the previous session by comparing the received P with *HID* and *PID*. After authenticating $\mathcal{T}$, it updates $\mathcal{T}$'s *ID* and transmits a message for authentication of $\mathcal{DB}$.

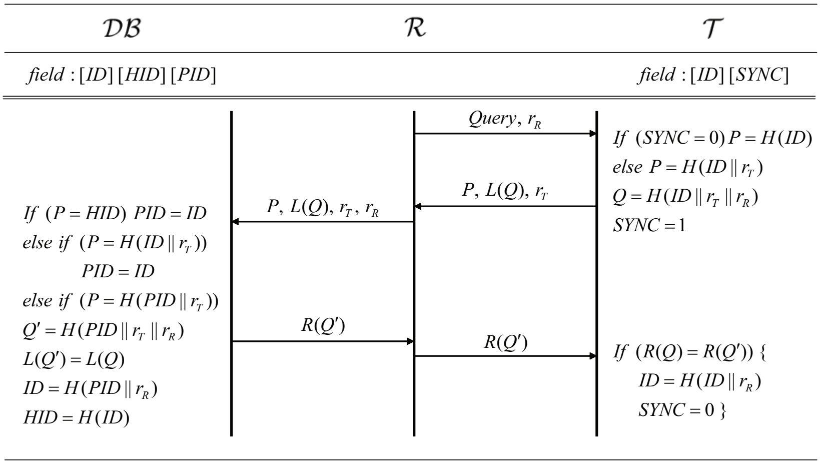

Fig. 4. LRMAP: Lightweight and resynchronous mutual authentication protocol

$\mathcal{T}$ emits $P = H(ID)$ or $P = H(ID\|r_T)$ according to the state of $SYNC$ in response to a query from $\mathcal{R}$. If $\mathcal{T}$ does not receive the last message from $\mathcal{R}$ due to a communication malfunction or the verification procedure failure, the $SYNC$ state is set as 1 and $\mathcal{T}$ responds with $P = H(ID\|r_T)$ to $\mathcal{R}$ in the next session. In the case the protocol finished normally, the $SYNC$ state becomes 0 and $\mathcal{T}$ transmits $P = H(ID)$ in the next session.

$\mathcal{R}$ broadcasts a query to $\mathcal{T}$ with a random number r_R and receives the information related to the authentication from $\mathcal{T}$, such as hashed values and random number r_T. It then forwards the messages received from $\mathcal{T}$ to $\mathcal{DB}$. After $\mathcal{DB}$ authenticates $\mathcal{T}$, $\mathcal{R}$ transmits the received message from $\mathcal{DB}$ to $\mathcal{T}$.

A step by step description of LRMAP is given below.

1. $\mathcal{R}$ chooses a random number r_R and broadcasts it to $\mathcal{T}$ with a $Query$.
2. $\mathcal{T}$ selects a random number r_T and computes P differently according to the state of *SYNC*. If $SYNC = 0$, then $P = H(ID)$, otherwise $P = H(ID\|r_T)$ using r_T generated by itself. It then computes $Q = H(ID\|r_T\|r_R)$ and sets the $SYNC$ field as 1. $\mathcal{T}$ transmits $P, L(Q)$ and r_T to $\mathcal{R}$ in response to the $Query$, $\mathcal{R}$ forwards the messages received from $\mathcal{T}$ to $\mathcal{DB}$ together with r_R generated by itself in step 1.
3. $\mathcal{DB}$ first compares the received $P = H(ID)$ with the *HID* values saved in the database. If the values match, $\mathcal{DB}$ regards the *ID* as the identity of $\mathcal{T}$ requesting authentication. This is a general case when the previous session is closed normally. If $\mathcal{DB}$ cannot find the *HID* in the first searching case, it then computes $H(ID\|r_T)$ with the received r_T and compares it with P. If the tag's response messages were blocked in the previous session, that is, $SYNC = 1$ and two *ID*s in the $\mathcal{DB}$ and tag are not updated, then $\mathcal{DB}$ finds a match with the *ID* of $\mathcal{T}$ in the second searching case. However, if $\mathcal{DB}$ cannot

find the *ID* of tag in the above two cases, it then computes $H(PID\|r_T)$ and compares it with P. $\mathcal{DB}$ finds a match with the *PID* of $\mathcal{T}$ when the reader's last messages were blocked in the previous session, that is, $SYNC = 1$ and $\mathcal{DB}$ updated the *ID*, yet the tag's *ID* was not updated. If $\mathcal{DB}$ cannot find the identity of $\mathcal{T}$ in the above three cases, it halts the searching of *ID* and can order $\mathcal{R}$ to query again in order to restart the process from the first step. If $\mathcal{DB}$ finds the *ID* or *PID* in the three searching cases, then it computes $Q' = H(PID\|r_T\|r_R)$ [4] and verifies that the following equation is satisfied:

$$L(Q') \stackrel{?}{=} L(Q). \tag{1}$$

If equation (1) is satisfied, $\mathcal{DB}$ computes $R(Q')$, transmits it to $\mathcal{R}$, and updates the *HID* for the next session. That is, it computes $ID = H(PID\|r_R)$ and updates $HID = H(ID)$.

4. $\mathcal{R}$ delivers the message $R(Q')$ received from $\mathcal{DB}$ to $\mathcal{T}$.
5. To verify the correctness of $R(Q')$, $\mathcal{T}$ tests the following equation:

$$R(Q) \stackrel{?}{=} R(Q'), \tag{2}$$

If equation (2) is correct, $\mathcal{T}$ updates the identity as $ID = H(ID\|r_R)$, then sets the *SYNC* value to 0.

5.2 Analysis of LRMAP

We now perform a formal analysis of LRMAP. For the detailed basic analysis, refer to [4]. With our formal proof model, LRMAP guarantees strong location privacy, as shown in the following Theorem 1. To induce Theorem 1, we first prove the following lemmas.

Lemma 1. *LRMAP guarantees indistinguishability for any polynomial bounded adversary $\mathcal{A}$ ($\mathcal{A}_{\mathcal{P}}$ or $\mathcal{A}_{\mathcal{A}}$), i.e., any security parameter k and $\mathcal{A}$'s computational boundary e_1, e_2, r_1 and r_2, where e_1, e_2, r_1 and r_2 is the number of* Execute, Execute*, Reply, *and* Reply*, *respectively.*

Proof: We use a similar proof method described in [8] [5].

First, we show LRMAP guarantees weak location privacy for $\mathcal{A}_{\mathcal{P}}$. For this, we specify the simulators $\mathsf{Sim}^{\mathsf{Exe}}$ and $\mathsf{Sim}^{\mathsf{Exe*}}$ for $\mathcal{T}_c$ in $\mathcal{AG}^{\mathsf{Indis}}$. $\mathsf{Sim}^{\mathsf{Exe}}$ and $\mathsf{Sim}^{\mathsf{Exe*}}$ do not know the value of b or any secret key k_c for $\mathcal{T}_c$. $\mathcal{A}_{\mathcal{P}}$'s interaction with $\mathsf{Sim}^{\mathsf{Exe}}$ and $\mathsf{Sim}^{\mathsf{Exe*}}$ will be computationally indistinguishable from an interaction with $\mathcal{T}_c$. Therefore, we suppose that $\mathcal{A}_{\mathcal{P}}$ gains no knowledge from its interaction with $\mathcal{T}_c$ in a real RFID system $\mathcal{S}$.

Note that $\mathcal{A}_{\mathcal{P}}$ chooses the challenge tag $\mathcal{T}_c$ from the un-revealed tags. Let L be the full list of the real quintuplets $(rn_1, hv_1, hv_2, rn_2, hv_3)$ outputted by $\mathcal{T}_c$

[4] Since *ID* is updated into *PID* after finding *ID* from *HID*, $Q' = H(PID\|r_T\|r_R)$ is computed regardless of *PID* or *ID*.

[5] Even though the defined attack games between our model and [8] are different, a similar proof can be used because both are based on the impossibility of distinguishing any two values.

during the challenge phase of the game, where hv_i means a hashed value for $i = 1, 2, 3$ and rn_j is a random number for $j = 1, 2$. During the challenge phase, $\mathsf{Sim}^{\mathsf{Exe}}$ simulates the result of a $\mathsf{Execute}$ call to $\mathcal{T}_c$ by generating $(r_{R,i}, P'_i = H(ID_i), L'_i(Q), r_{T,i}, R'_i(Q))$ for $i \leq \#\mathsf{Execute} = e_1$ and appending it to a list L'_1.

Similarly, $\mathsf{Sim}^{\mathsf{Exe*}}$ simulates the result of a $\mathsf{Execute*}$ call to $\mathcal{T}_c$ by generating $(r_{R,j}, P''_j = H(ID_j \| r_{T,j}), L''_j(Q), r_{T,j}, R''_j(Q))$ for $j \leq \#\mathsf{Execute*} = e_2$ and appending it to a list L''_1.

Note that L'_1 and L''_1 are empty at the beginning of the challenge phase and $q - 1 = e_1 + e_2$, where $q - 1$ means the maximum number of the queries executed by $\mathcal{A}_{\mathcal{P}}$ for $\mathcal{T}_c$'s instance. In addition to any valid tag quintuplets outputted by $\mathsf{Sim}^{\mathsf{Exe}}$ and $\mathsf{Sim}^{\mathsf{Exe*}}$, $\mathcal{DB}$ includes any quintuplet in L'_1 and L''_1.

In order for $\mathcal{A}_{\mathcal{P}}$ to distinguish between the simulated challenge phase and a real challenge phase, $\mathcal{A}_{\mathcal{P}}$ must be able to determine that some quintuplet $(r_R, P', L'(Q), r_T, R'(Q)) \in L'_1$ is invalid for $\mathcal{T}_c$. As a necessary condition for this determination, $\mathcal{A}_{\mathcal{P}}$ must identify a quintuplet $(r_R, P = H(ID), L(Q), r_T, R(Q))$ that is valid for $\mathcal{T}_c$, but such that $P \neq P', L(Q) \neq L'(Q)$ and $R(Q) \neq R'(Q)$. That is, $\mathcal{A}_{\mathcal{P}}$ has to remove an invalid $(r_R, P', L'(Q), r_T, R'(Q))$ from L'_1 to show that the correct $\mathsf{Sim}^{\mathsf{Exe}}$ is present.

Consequently, one of the following two conditions must occur at some point in the course of the challenge phase of the game.

1. *There is a random number pair (r_R, r_T) such that $(r_R, P', L'(Q), r_T, R'(Q)) \in L'_1$ and $(r_R, P, L(Q), r_T, R(Q)) \in L$ for some pair (X, Y), where $X = (P', L'(Q), R'(Q)) \in L'_1$, $Y = (P, L(Q), R(Q)) \in L$ and $P \neq P', L(Q) \neq L'(Q)$ and $R(Q) \neq R'(Q)$*: Since $\mathcal{A}_{\mathcal{P}}$ may make at most e_1 $\mathsf{Execute}$ calls to $\mathcal{T}_c$, we have $Min(\#\mathsf{Execute}, |L|) = e_1$, where $\#\mathsf{Execute} = e_1$ and $|L| = q - 1$. As r_R and r_T are random k-bit values, and thus the space of random numbers is 2^k, it follows that this condition occurs with probability at most $e_1^2/2^k$.
2. *For a pair $(r_R, r_T) \in L'_1, L$, $\mathcal{A}_{\mathcal{P}}$ directly computes $P, L(Q)$ and $R(Q)$ that are equal to X or Y*: Since $L(Q) \| R(Q) = H(ID \| r_T \| r_R)$ and $P = H(ID)$, $\mathcal{A}_{\mathcal{P}}$ first must be able to find out ID. At this time, the probability of recovering ID from $H(ID)$ is $1 - (1 - 1/2^k)^{e_1}$, given that e_1 $\mathsf{Execute}$ queries are called, in which it is approximately $e_1/2^k$ provided that e_1 is small compared to 2^k. Similarly, the probability of knowing ID from $L(Q)$ and $R(Q)$ is $e_1/2^{(k/2)}$ and $e_1/2^{(k/2)}$, respectively.

Therefore, $\mathcal{A}_{\mathcal{P}}$ can distinguish $\mathsf{Sim}^{\mathsf{Exe}}$ from $\mathcal{T}_c$ with probability at most $e_1^2/2^k + e_1/2^k + e_1/2^{(k/2)} + e_1/2^{(k/2)}$, which is negligible for polynomial bounded $\mathcal{A}_{\mathcal{P}}$.

With the similar method, $\mathcal{A}_{\mathcal{P}}$ must be able to determine that some quintuplet $(r_R, P'', L''(Q), r_T, R''(Q)) \in L''_1$ is invalid for $\mathcal{T}_c$. In other words, $\mathcal{A}_{\mathcal{P}}$ must identify a quintuplet $(r_R, P = H(ID \| r_T), L(Q), r_T, R(Q))$ that is valid for $\mathcal{T}_c$, but such that $P \neq P'', L(Q) \neq L''(Q)$ and $R(Q) \neq R''(Q)$. Finally, $\mathcal{A}_{\mathcal{P}}$ rules out invalid quintuplets from L''_1 to show that $\mathsf{Sim}^{\mathsf{Exe*}}$ is present.

Consequently, $\mathcal{A}_{\mathcal{P}}$ can distinguish $\mathsf{Sim}^{\mathsf{Exe*}}$ from $\mathcal{T}_c$ with probability at most $e_2^2/2^k + e_2/2^k + e_2/2^{(k/2)} + e_2/2^{(k/2)}$, which is negligible for polynomial bounded $\mathcal{A}$, where e_2 is the maximum number of $\mathsf{Execute*}$ calls.

Next, we show that indistinguishability is also guaranteed for $\mathcal{A}_{\mathcal{A}}$ in LRMAP. Note, $\mathcal{A}_{\mathcal{A}}$ can insert or modify messages in the real RFID communication in addition to eavesdropping.

For this reason, we additionally define some simulators $\mathsf{Sim}^{\mathsf{Que}}$, $\mathsf{Sim}^{\mathsf{Rep}}$, $\mathsf{Sim}^{\mathsf{Rep*}}$ and $\mathsf{Sim}^{\mathsf{For}}$ for $\mathcal{T}_c$ in $\mathcal{AG}^{\mathsf{Indis}}$. The simulators do not know the value of a fair coin b or any secret key k_c for $\mathcal{T}_c$. $\mathcal{A}_{\mathcal{A}}$'s interaction with $\mathsf{Sim}^{\mathsf{Que}}$, $\mathsf{Sim}^{\mathsf{Rep}}$, $\mathsf{Sim}^{\mathsf{Rep*}}$ and $\mathsf{Sim}^{\mathsf{For}}$ will be computationally indistinguishable from an interaction with $\mathcal{T}_c$. Therefore, we suppose that $\mathcal{A}_{\mathcal{A}}$ does not gain knowledge from its interaction with $\mathcal{T}_c$ in a real RFID system $\mathcal{S}$.

During the challenge phase, $\mathsf{Sim}^{\mathsf{Que}}$ simulates the result of a Query call to $\mathcal{T}_c$ by generating a random number $r'_{R,i}$ for $i \leq \#\mathsf{Query} = q - 1$ and appending it to a list M_0.

$\mathsf{Sim}^{\mathsf{Rep}}$ and $\mathsf{Sim}^{\mathsf{Rep*}}$ simulate the result of a Reply and $\mathsf{Reply*}$ call to $\mathcal{T}_c$, respectively. While $\mathsf{Sim}^{\mathsf{Rep}}$ generates $(P'_j = H(ID_j), L'_j(Q), r'_{T,j})$ for $j \leq \#\mathsf{Reply} = r_1$ and appends it to a list M'_1, $\mathsf{Sim}^{\mathsf{Rep*}}$ makes $(P''_k = H(ID_k \| r''_{T,k}), L''_k(Q), r''_{T,k})$ for $k \leq \#\mathsf{Reply*} = r_2$ and appends it to a list M''_1, in which $r_1 + r_2 = q - 1$.

Meanwhile, $\mathsf{Sim}^{\mathsf{For}}$ simulates the result of a $\mathsf{Forward}_2$ call to $\mathcal{T}_c$ by generating $R'_i(Q)$ for $i \leq \#\mathsf{Forward}_2 = q - 1$ and appending it in a list M_2.

To simplify the analysis, here we assume that the result of $\mathsf{Sim}^{\mathsf{Que}}$ influences the simulated results of $\mathsf{Sim}^{\mathsf{Rep}}$, $\mathsf{Sim}^{\mathsf{Rep*}}$ and $\mathsf{Sim}^{\mathsf{For}}$. This is because $r'_{R,i}$ outputted by $\mathsf{Sim}^{\mathsf{Que}}$ is included in the computation of $Q = H(ID \| r_T \| r_R)$ of $\mathsf{Sim}^{\mathsf{Rep}}$, $\mathsf{Sim}^{\mathsf{Rep*}}$ and $\mathsf{Sim}^{\mathsf{For}}$, where $r'_{R,i} = r_R$. Of course, we can consider the random number $r'_{R,i}$ is independent of r_R in Q, i.e., $r'_{R,i} \neq r_R$, which causes the complicated analysis.

Recall that $\mathcal{A}_{\mathcal{A}}$ selects the challenge tag $\mathcal{T}_c$ from the un-revealed tags, and L is the full list of quintuplets $(rn_1, hv_1, hv_2, rn_2, hv_3)$ outputted by $\mathcal{T}_c$ during the challenge phase of the game. Note that M_0, M'_1, M''_1 and M_2 are empty at the beginning of the challenge phase.

In order for $\mathcal{A}_{\mathcal{A}}$ to distinguish between the simulated challenge phase and a real phase, $\mathcal{A}_{\mathcal{A}}$ must determine that some triplet $(P', L'(Q), r_T) \in M'_1$ is invalid for $\mathcal{T}_c$. For this, $\mathcal{A}_{\mathcal{A}}$ must identify a triplet $(P = H(ID), L(Q), r_T)$ that is valid for $\mathcal{T}_c$, but such that $P' \neq P$ and $L'(Q) \neq L(Q)$. In other words, $\mathcal{A}_{\mathcal{A}}$ has to remove an invalid $(P, L'(Q), r_T)$ to show that $\mathsf{Sim}^{\mathsf{Rep}}$ is present.

Consequently, one of the following two cases must occur at some point in the course of the challenge phase of the game.

1. *There is a random number r_T such that $(P', L'(Q), r_T) \in M'_1$ and $(P, L(Q), r_T) \in L$ for some pair (X, Y), where $X = (P', L'(Q)) \in M'_1$ and $Y = (P, L(Q)) \in L$*: Since $\mathcal{A}_{\mathcal{A}}$ may execute at most r_1 Reply calls to $\mathcal{T}_c$, we have $Min(\#\mathsf{Reply}, |L|) = e_1$, where $\#\mathsf{Reply} = r_1$ and $|L| = q - 1$. As r_T is a random k-bit value, and thus the space of random number is 2^k, it follows that this case occurs with probability at most $r_1^2/2^k$.
2. *For a random number $r_T \in M'_1, L$, $\mathcal{A}_{\mathcal{A}}$ computes the values corresponding to X or Y*: Since $L(Q)$ is the bit string from MSB to half of $H(ID \| r_T \| r_R)$ and $P = H(ID)$, $\mathcal{A}_{\mathcal{A}}$ must know *ID* and r_R to compute $L(Q)$ or P corresponding to X or Y. Given that at most r_1 Reply queries are called, the probability of recovering *ID* is $r_1/2^{(k/2)}$.

Therefore, $\mathcal{A}_{\mathcal{A}}$ can distinguish $\mathsf{Sim}^{\mathsf{Rep}}$ from $\mathcal{T}_c$ with probability at most $r_1^2/2^k + r_1/2^{(k/2)}$, which is negligible for polynomial bounded $\mathcal{A}_{\mathcal{A}}$.

With the similar method, $\mathcal{A}_{\mathcal{A}}$ can distinguish $\mathsf{Sim}^{\mathsf{Rep*}}$ from $\mathcal{T}_c$ with probability at most $r_2^2/2^k + r_2/2^{(k/2)}$, which is also negligible for polynomial bounded $\mathcal{A}_{\mathcal{A}}$. Meanwhile, $\mathcal{A}_{\mathcal{A}}$ can distinguish $\mathsf{Sim}^{\mathsf{For}}$ from $\mathcal{T}_c$ with probability at most $(q-1)/2^{(k/2)}$, which is also negligible for polynomial bounded $\mathcal{A}_{\mathcal{A}}$.

We omit the analysis of $\mathcal{A}_{\mathcal{A}}$'s $\mathsf{Sim}^{\mathsf{Exe}}$ and $\mathsf{Sim}^{\mathsf{Exe*}}$ because $\mathcal{A}_{\mathcal{A}}$'s execution for Execute and Exectue* oracles is the same with $\mathcal{A}_{\mathcal{P}}$'s one. □

Next, forward secrecy in LRMAP is guaranteed by the following Lemma 2.

Lemma 2. *LRMAP guarantees forward secrecy for any polynomial bounded adversary $\mathcal{A}$ ($\mathcal{A}_{\mathcal{P}}$ or $\mathcal{A}_{\mathcal{A}}$), any security parameter k and $\mathcal{A}$'s computational boundary e_1, e_2, r_1, r_2 and f, where e_1, e_2, r_1, r_2 and f is the number of Execute, Execute*, Reply, Reply*, and Forward$_2$, respectively.*

Proof: we show LRMAP guarantees forward secrecy for $\mathcal{A}_{\mathcal{P}}$ [6].

In the challenge phase, $\mathcal{A}_{\mathcal{P}}$ makes e_1 Execute and e_2 Execute* calls for $(n-1)$'s each tag except $\mathcal{T}_c$ as in the learning phase. At this time, let $\mathcal{A}_{\mathcal{P}}$'s advantage for recovering $\mathcal{T}_i$'s ID_i be $\mathsf{Adv}^{\mathsf{Rec}}_{\mathcal{A}_{\mathcal{P}},\mathcal{T}_i}(k)$ from the collected transaction of Execute and Execute* queries. In other words, the probability of finding out ID_i from a quintuplet $(r_{R,i}, P_i, L_i(Q), r_{T,i}, R_i(Q))$ is $2 - (1-1/2^k)^{e_1} - (1-1/2^k)^{e_2}$, given e_1 Execute queries and e_2 Execute* queries for $\mathcal{T}_i$, where $P_i = H(ID_i)$ or $P_i = H(ID_i \| r_{R,i})$ and $i = 1, \ldots, n-1$.

Meanwhile, when $\mathcal{A}_{\mathcal{P}}$ is given a random value or $\mathcal{T}_c$'s real message in response of Test query, it must be able to compute $P_c, L_c(Q), R_c(Q)$ corresponding to $\mathcal{T}_c$'s $(i-1)$th instance for the correct guessing, i.e., $b = b'$, where $Q = H(ID_c \| r_T \| r_R)$. As the necessary condition, $\mathcal{A}_{\mathcal{P}}$ has to recover ID_c from ith instance $H(ID)$, where $ID = H(ID_c \| r_T)$. Note that $\mathcal{A}_{\mathcal{P}}$ already knows ID related to ith instance with $\mathsf{Reveal}(\mathcal{T}_c, i)$. We now define $\mathcal{A}_{\mathcal{P}}$'s advantage for guessing the correct fair coin b as $\mathsf{Adv}^{\mathsf{FS}}_{\mathcal{A}_{\mathcal{P}},\mathcal{S}}(k)$, thus the following equation is induced:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{FS}}_{\mathcal{A}_{\mathcal{P}},\mathcal{S}}(k) &\leq \mathsf{Adv}^{\mathsf{Rec}}_{\mathcal{A}_{\mathcal{P}},\mathcal{T}_1}(k) + \mathsf{Adv}^{\mathsf{Rec}}_{\mathcal{A}_{\mathcal{P}},\mathcal{T}_2}(k) + \cdots + \mathsf{Adv}^{\mathsf{Rec}}_{\mathcal{A}_{\mathcal{P}},\mathcal{T}_{n-1}}(k) \\
&\leq (n-1) \cdot \mathsf{Adv}^{\mathsf{Rec}}_{\mathcal{A}_{\mathcal{P}},\mathcal{T}_1}(k) \\
&\leq (n-1) \cdot \{2 - (1 - \frac{1}{2^k})^{e_1} - (1 - \frac{1}{2^k})^{e_2}\} \\
&\simeq (n-1) \cdot \frac{e_1 + e_2}{2^k}
\end{aligned}$$

From the above equation, $\mathcal{A}_{\mathcal{P}}$ can distinguish a random value from the real message with probability at most $(n-1) \cdot (e_1 + e_2)/2^k$, which is negligible for polynomial bounded A_P.

[6] A passive adversary $\mathcal{A}_{\mathcal{P}}$ cannot execute direct and stronger attacks such as break-in, compromising of a tag, reveal of tag's memory, etc. except eavesdropping. That is, $\mathcal{A}_{\mathcal{P}}$ is generally not allowed to execute Reveal. However, when considering a disclosure of tag's internal state due to the tag holder's carelessness, it is modeled with Reveal. Therefore, we assume that $\mathcal{A}_{\mathcal{P}}$ calls Reveal throughout the paper.

With the similar method, we can show that forward secrecy for $\mathcal{A}_{\mathcal{A}}$ is satisfied in LRMAP. When the maximum number of Reply, $\mathsf{Reply^*}$ and $\mathsf{Forward}_2$ for $(n-1)$'s $\mathcal{T}_i$ is r_1, r_2 and f, respectively, the adversary $\mathcal{A}_{\mathcal{A}}$ can correctly guess b' with probability at most $(n-1) \cdot (r_1 + r_2 + f)/2^{(k/2)} + (n-1) \cdot (r_1 + r_2)/2^k$, which is negligible for polynomial bounded $\mathcal{A}_{\mathcal{A}}$.

We omit the analysis of $\mathcal{A}_{\mathcal{A}}$'s $\mathsf{Sim}^{\mathsf{Exe}}$ and $\mathsf{Sim}^{\mathsf{Exe^*}}$ because $\mathcal{A}_{\mathcal{A}}$'s execution for $\mathsf{Execute}$ and $\mathsf{Exectue^*}$ oracles is the same as $\mathcal{A}_{\mathcal{P}}$'s one. □

From Lemma 1, Lemma 2 and Definition 3, we can induce the following security theorem.

Theorem 1. (LRMAP: Strong Location Privacy). *LRMAP guarantees strong location privacy for any polynomial bounded adversary $\mathcal{A}$ ($\mathcal{A}_{\mathcal{P}}$ or $\mathcal{A}_{\mathcal{A}}$), any security parameter k and $\mathcal{A}$'s computational boundary e_1, e_2, r_1, r_2 and f, where e_1, e_2, r_1, r_2 and f is the number of Execute, Execute*, Reply, Reply*, and $\mathsf{Forward}_2$, respectively.*

6 Conclusion and Future Work

We proposed a new formal proof model for provable location privacy in RFID systems, in which two attack games are defined for indistinguishability and forward secrecy. That is, we considered not only passive/active attacks to the message flows, but also physical attacks for disclosing tag's internal state. Thus, the proposed model is practical for real-world RFID systems. We further applied our model to analyze location privacy of an existing RFID protocol LRMAP. With the similar method, our model can be applied to the RFID protocols based on hash function [5,10,12,16] for provable location privacy.

As the future work, we will consider an authentication model suitable for the RFID environment using the previously defined oracles. In this case, we have to consider both the secure channel between a database and a reader and the insecure channel between the reader and tags. It will be possible by inducing the notion of matching conversation proposed by Bellare and Rogaway [2].

Acknowledgement

We would like to thank anonymous reviewers for their helpful comments to improve our manuscript. This research was supported by the MKE (Ministry of Knowledge Economy) of Korea, under the ITRC support program supervised by the IITA (IITA-2008-C1090-0801-0026).

References

1. Avoine, G.: Adversarial Model for Radio Frequency Identificatin. Cryptology ePrint Archieve, Report 2005/049 (2005), `http://eprint.iacr.org`
2. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994)

3. Golle, P., Jakobsson, M., Jules, A., Syverson, P.: Universal Re-encryption for Mixnets. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 163–178. Springer, Heidelberg (2004)
4. Ha, J., Ha, J., Moon, S., Boyd, C.: LRMAP: Lightweight and Resynchronous Mutual Authentication Protocol for RFID System. In: Stajano, F., Kim, H.-J., Chae, J.-S., Kim, S.-D. (eds.) ICUCT 2006. LNCS, vol. 4412, pp. 80–89. Springer, Heidelberg (2007)
5. Henrici, D., Müller, P.: Hash-based Enhancement of Loaction Privacy for Radio Frequency Identification Devices using Varing Identifiers. In: PERCOMW 2004, pp. 149–162. IEEE, Los Alamitos (2004)
6. Juels, A., Pappu, R.: Squealing Euros: Privacy Protection in RFID-enabled Banknotes. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 103–121. Springer, Heidelberg (2003)
7. Juels, A., Rivest, R.L., Szydlo, M.: The Blocker Tag: Selective Blocking of RFID Tags for consumer Privacy. In: ACM CCS 2003, pp. 103–111. ACM, New York (2003)
8. Jules, A., Weis, S.A.: Defining Strong Privacy for RFID, Cryptology ePrint Archieve, Report 2006/137 (2006), http://eprint.iacr.org
9. Juels, A.: RFID Security and Privacy: A Research Survey, RSA Laboratories (2005)
10. Li, Y., Cho, Y., Um, N., Lee, S.: Security and Privacy on Authentication for Low-cost RFID. In: Wang, Y., Cheung, Y.-m., Liu, H. (eds.) CIS 2006. LNCS (LNAI), vol. 4456, pp. 788–794. Springer, Heidelberg (2007)
11. Li, Y., Jeong, Y., Sun, N., Lee, S.: Low-cost Authenticatoin Protocol of the RFID System Using Partial ID. In: Wang, Y., Cheung, Y.-m., Liu, H. (eds.) CIS 2006. LNCS (LNAI), vol. 4456, pp. 598–604. Springer, Heidelberg (2007)
12. Lee, S., Asano, T., Kim, K.: RFID Mutual Authentication Scheme based on Synchronized Secret Information. In: SCIS 2006 (2006)
13. Mao, W.: Modern Cryptography, Theory and Practice. Prentice Hall, Englewood Cliffs (2003)
14. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Apprach to Privacy-Friendly Tags. In: RFID Privacy Workshop (2003)
15. Ohkubo, M., Suzuki, K., Kinoshita, S.: Hash-Chain Based Forward-Secure Privacy Protection Scheme for Low-Cost RFID. In: SCIS 2004, pp. 719–724 (2004)
16. Rhee, K., Kwak, J., Kim, S., Won, D.: Challenge-Response Based RFID Authentication Protocol for Distributed Database Envirionment. In: Hutter, D., Ullmann, M. (eds.) SPC 2005. LNCS, vol. 3450, pp. 70–84. Springer, Heidelberg (2005)
17. Saito, J., Ryou, J., Sakurai, K.: Enhancing Privacy of Universal Re-encryption Scheme for RFID Tags. In: Yang, L.T., Guo, M., Gao, G.R., Jha, N.K. (eds.) EUC 2004. LNCS, vol. 3207, pp. 879–890. Springer, Heidelberg (2004)
18. Sarma, S.E., Weis, S.A., Engels, D.W.: Radio-Frequency Identification: Security Risks and Challenges, RSA Laboratories, vol. 6(1) (2003)
19. Vaudenay, S.: On Privacy Models for RFID. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)
20. Weis, S.A.: Security and Privacy in Radio-Frequency Identification Devices, MS Thesis, MIT (2003)
21. Weis, S.A., Sarma, S.E., Rivest, R.L., Engles, D.W.: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 285–289. Springer, Heidelberg (2004)