6-2010

Revisiting unpredictability-based RFID privacy models

Junzuo LAI
Shanghai Jiaotong University

Robert Huijie DENG
Singapore Management University, robertdeng@smu.edu.sg

Yingjiu LI
Singapore Management University, yjli@smu.edu.sg

DOI: https://doi.org/10.1007/978-3-642-13708-2_28

Applied Cryptography and Network Security: 8th International Conference, ACNS 2010,
Beijing, China, June 22-25: Proceedings. pp. 475-492.
http://dx.doi.org/10.1007/978-3-642-13708-2_28

Revisiting Unpredictability-Based RFID Privacy Models

Junzuo Lai[1,2], Robert H. Deng[2], and Yingjiu Li[2]

[1] Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200030, China
laijunzuo@sjtu.edu.cn
[2] School of Information Systems,
Singapore Management University, Singapore 178902
{robertdeng,yjli}@smu.edu.sg

Abstract. Recently, there have been several attempts in establishing formal RFID privacy models in the literature. These models mainly fall into two categories: one based on the notion of indistinguishability of two RFID tags, denoted as *ind*-privacy, and the other based on the unpredictability of the output of an RFID protocol, denoted as *unp*-privacy. Very recently, at CCS'09, Ma et al. proposed a modified *unp*-privacy model, referred to as *unp*′-privacy. In this paper, we first revisit the existing RFID privacy models and point out their limitations. We then propose a new RFID privacy model, denoted as unp^*-privacy, based on the indistinguishability of a real tag and a virtual tag. We provide justification for the new model and formally clarify its relationship with *ind*-privacy model. Finally, we modify Ma et al.'s 2-round RFID protocol to a 3-round mutual authentication RFID protocol and prove that it is of unp^*-privacy.

Keywords: RFID, privacy, security.

1 Introduction

Radio Frequency Identification (RFID) has been widely envisioned as an inevitable replacement of barcodes and other consumer labeling techniques for automatic object identification. An RFID system consists of small devices called RFID tags, one or more RFID readers and a back-end database. Unlike barcodes, each RFID tag records a sufficiently long bitstring to uniquely identify the tag or its bearer. RFID readers communicate with RFID tags using RF signals at a distance from a few inches to several feet. Since RF signals are invisible and penetrating, RFID systems provide a perfect environment for attackers. The prevalence of RFID technologies introduces various serious risks and poses unique security concerns [9,15].

Security problems in RFID systems can be classified into two types. The first is concerned with attacks which aim to wipe out the functioning of the system. The second type, the one which interests us here, is related to privacy. In particular, unauthorized tracking of RFID system users and RFID tag bearers has been

recognized as one of the most imperative privacy concerns in the deployments of RFID systems. A privacy-preserving RFID system should therefore provide anonymity (i. e., confidentiality of a tag's identity) as well as unlinkability of the protocol transcripts of a tag [17]. Much attention has been devoted to RFID security, and various schemes have been proposed. The research for secure RFID systems can be mainly categorized into physical technologies [11,5] and protocol-based techniques [21,7,14,19,10,18,6,1]. Juels provides a survey of much of the related literature in [9] and Avoine maintains a current online bibliography at [2]. Nevertheless, most of the existing RFID security research efforts lack formal analysis and mainly offer ad hoc notions of security. In this paper, we are concerned with formal provable privacy models for RFID systems, with a focus on protocol-based techniques.

1.1 Related Work

Avoine [3] first formalizes the adversary model in RFID systems and proposes very general and flexible definitions of RFID privacy. Based on the formal adversary model, Juels and Weis [12] define the notion of strong privacy. The aim of Avoine [3] is to capture a range of adversarial abilities, while Juels and Weis [12] seek to characterize a very strong adversary with a relatively simple definition. In other words, Juels and Weis [12] aim for specificity and simplicity over flexibility. The privacy notion in [12] is based on the indistinguishability of two RFID tags, denoted as *ind*-privacy. However, to our knowledge, there is no RFID protocol that has been *directly* proven to be of *ind*-privacy; On the other hand, if an RFID protocol is not of *ind*-privacy, it can be checked against the *ind*-privacy model easily.

Vaudenay [20] considers side-channel attacks in his RFID privacy model and proposes eight privacy classes which are later consolidated to three by Ng et al. [22]. Paise and Vaudenay [16] extend the definitions in [20] to address mutual authentication. However, the privacy definitions in [20,22,16] contradict reader authentication for any privacy notion that allows tag corruption.

In [8], Ha et al. propose a different privacy model based on the unpredictability of tag outputs, denoted as *unp*-privacy. Unfortunately, this model was later shown to have some deficiencies in its definition [4]. Recently, Ma et al. [13] propose a refined *unp*-privacy model for RFID systems, denoted as unp'-privacy, and investigate the relationship between *ind*-privacy and unp'-privacy.

1.2 Our Contributions

In this paper, we address formal RFID privacy models with the following main contributions:

1. We revisit the unp'-privacy model in [13] and point out its limitation. Specifically, though the unp'-privacy model is robust for 2-round RFID protocols but falls short in dealing with 3-round (i. e., mutual authentication) protocols. We demonstrate this by presenting a 3-round RFID protocol which has a flaw with respect to privacy but can be proven to be of unp'-privacy.

2. We propose a new privacy model, denoted as unp^*-privacy, based on the indistinguishability of a real tag and a virtual tag. We clarify the relationship between the *ind*-privacy and the unp^*-privacy by formally proving that the former is weaker than the latter. To understand which level of privacy an RFID system provides, it is critical to clarify the relationship between the privacy notions.
3. We modify and extend the RFID protocol in [13] to 3-round mutual authentication protocol and show that it is of unp^*-privacy.

1.3 Organization

The rest of the paper is organized as follows. In Section 2, we briefly discuss the formal definitions for the *ind*-privacy and the *unp*-privacy models. We revisit and re-examine the unp'-privacy model in Section 3. In Section 4, we introduce our privacy model, establish its relation with the *ind*-privacy model and show that an improved version of the protocol in [13] is of unp^*-privacy. We conclude in Section 5.

2 Preliminaries

If S is a set, then $s \in_R S$ indicates that s is chosen uniformly at random from S. If $x_1, x_2, \ldots$ are strings, then $x_1 \| x_2 \| \ldots$ denotes their concatenation. Let $y \leftarrow \mathcal{A}^{\mathcal{O}_1,\ldots,\mathcal{O}_n}(x_1, x_2, \ldots)$ denote that y be assigned with the output of the algorithm $\mathcal{A}$ which takes $x_1, x_2, \ldots$ as inputs and has accesses to oracles $\mathcal{O}_1, \ldots, \mathcal{O}_n$.

2.1 Pseudorandom Functions

A pseudorandom function is a family of functions with the property that the input-output behavior of a random instance of the family is "computationally indistinguishable" from that of a random function. Let $F : \mathsf{Keys}(F) \times D \rightarrow R$ be a family of functions and let $\mathsf{Rand}^{D \rightarrow R}$ be the family of all functions with domain D and range R, where $\mathsf{Keys}(F)$ is the set of keys (or indexes) of F. Consider the following game between an attack algorithm $\mathcal{A}$ and a challenger.

$\mathbf{Game}_{\mathcal{A}}^{prf}$
$\triangleright$ $\beta \in_R \{0, 1\}$;
$\triangleright$ If $\beta = 0$ then $g \in_R \mathsf{Rand}^{D \rightarrow R}$, else $g \in_R F$;
$\triangleright$ $\beta' \leftarrow \mathcal{A}^g$.

Throughout the game, we assume that $\mathcal{A}$ makes at most q oracle queries. We define $\mathcal{A}$'s advantage in the above game as

$$Adv_{\mathcal{A}}(q) = \left| \mathsf{Pr}[\beta = \beta'] - \frac{1}{2} \right|.$$

Definition 1. *An adversary $\mathcal{A}$ (t, q, ϵ)-breaks the pseudorandomness of the function family F if the advantage $Adv_{\mathcal{A}}(q)$ of $\mathcal{A}$ in the above game is at least ϵ and the running time of $\mathcal{A}$ is at most t.*

Definition 2. *A function family F is said to be (t, q, ϵ)-pseudorandom if there exists no adversary who can (t, q, ϵ)-break the pseudorandomness of F.*

2.2 An RFID System Model

Without loss of generality, we assume a fixed, polynomial-size tag set $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$, a reader $\mathcal{R}$ and a back-end database $\mathcal{DB}$ as the elements of our RFID system, denoted as $\mathcal{S} = \{\mathcal{T}, \mathcal{R}, \mathcal{DB}\}$. Typically, each tag is a passive transponder identified by a unique ID and has only limited memory which can be used to store several keys and/or state information. The reader $\mathcal{R}$ is composed of one or more transceivers and a processing subsystem. The database $\mathcal{DB}$ maintains $\mathcal{T}$'s authentication data such as tag IDs, secret keys, states and session identifiers. Communications between $\mathcal{R}$ and $\mathcal{T}$ take place over an insecure air interface, while communications between $\mathcal{R}$ and $\mathcal{DB}$ are assumed to be over a secure channel.

In addition, the RFID system $\mathcal{S}$ includes a tuple of algorithms described bellow.

$\mathsf{Initialize}(\kappa)$: It takes as input a security parameter κ, generates key k_i for each tag $\mathcal{T}_i \in \mathcal{T}$ and sets the tag's initial state; it also associates $\mathcal{T}_i$ with its unique ID_i and setups the back-end database $\mathcal{DB}$ for $\mathcal{R}$ to store necessary information for tag identification.

$\mathsf{ReaderStart}()$: It invokes $\mathcal{R}$ to output a new session identifier sid and the first protocol message m_1 of the session.

$\mathsf{TagCompute}(sid, m_1, \mathcal{T}_i)$: It takes as input a session identifier sid, a protocol message m_1 and $\mathcal{T}_i$, outputs a message m_2. This algorithm is run by $\mathcal{T}_i$.

$\mathsf{ReaderCompute}(sid, m_2)$: It takes as input a session identifier sid and a protocol message m_2, outputs a protocol message m_3. This algorithm is run by $\mathcal{R}$.

$\mathsf{Execute}(\mathcal{R}, \mathcal{T}_i)$: It takes as input $\mathcal{R}$ and $\mathcal{T}_i$, runs the interactive authentication protocol between $\mathcal{R}$ and $\mathcal{T}_i$ and outputs the entire protocol transcript. For the three-round canonical RFID protocol shown in Fig 1, we have

$$(m_1, m_2, m_3) \leftarrow \mathsf{Execute}(\mathcal{R}, \mathcal{T}_i),$$

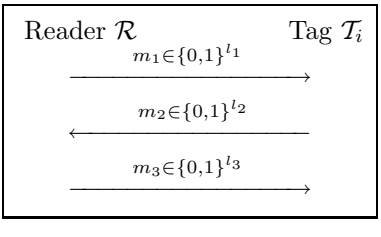

Fig. 1. The canonical RFID Protocol

where $(sid, m_1) \leftarrow \mathsf{ReaderStart}(), m_2 \leftarrow \mathsf{TagCompute}(sid, m_1, \mathcal{T}_i)$ and $m_3 \leftarrow \mathsf{ReaderCompute}(sid, m_2)$.

2.3 Adversaries

An adversary $\mathcal{A}$ is a probabilistic polynomial time (PPT) algorithm and is assumed to have complete control over all communications between $\mathcal{R}$ and $\mathcal{T}$. The interaction between $\mathcal{A}$ and the protocol participants occurs only via oracle queries, which model the adversary's capabilities in a real attack. In the following, we specify oracles $\mathcal{A}$ is permitted to query.

Launch($\mathcal{R}$): It invokes $\mathcal{R}$ to start a session of the protocol and responds with a session id sid and the first protocol message m_1.
SendTag($sid, m'_1, \mathcal{T}_i$): It invokes $\mathcal{T}_i$ and responds with a protocol message m_2.
SendReader(sid, m'_2): It invokes $\mathcal{R}$ and responds with a protocol message m_3.
Reveal($\mathcal{T}_i$): It invokes $\mathcal{T}_i$ and returns the tag's current secret key and internal state.

Queries to **SendTag** and **SendReader** model active attacks, in which the adversary may tamper with the message being sent over the insecure RF channel. Queries to **Reveal** model the leakage of tags' secret information.

Let $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$ and $\mathcal{O}_4$ denote **Launch**, **SendTag**, **SendReader** and **Reveal** oracles, respectively. All privacy models in this paper are defined using a game between an adversary $\mathcal{A}$ and a challenger. Throughout a game, we assume that $\mathcal{A}$ is allowed to launch $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$ and $\mathcal{O}_4$ oracle queries without exceeding q_{ini}, q_{st}, q_{sr} and q_{rv} overall calls, respectively.

2.4 The *ind*-Privacy and *unp*-Privacy Models

2.4.1 The *ind*-Privacy Model

Juels and Weis [12] present an indistinguishability-based RFID privacy model which is reminiscent of the classic indistinguishability under chosen-plaintext attack (IND-CPA) and under chosen-ciphertext attack (IND-CCA) cryptosystem security.

Figure 2 illustrates the *ind*-privacy game $\mathbf{Game}^{ind}_{\mathcal{A}}[\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}]$, in which $\mathcal{A}$ is comprised of a pair of algorithms $(\mathcal{A}_1, \mathcal{A}_2)$. The game proceeds as follows. At first, the challenger initializes the RFID system $\mathcal{S}$ by producing a reader $\mathcal{R}$ and a set of tags $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_n\}$ according to the security parameter κ. Then, $\mathcal{A}_1$ issues $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$ and $\mathcal{O}_4$ oracle queries, and outputs two uncorrupted tags $\{\mathcal{T}_i, \mathcal{T}_j\}$ (i.e., tags to which no **Reveal** queries have been issued) as challenge candidates. It also outputs a state information st which will be transmitted to algorithm $\mathcal{A}_2$. One of the two candidates $\mathcal{T}_c$ is then selected based on the value of a random bit and presented to $\mathcal{A}$ (effectively as a tag oracle). $\mathcal{A}_2$ is allowed to query $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$ and $\mathcal{O}_4$ oracles on $\mathcal{R}$, $\mathcal{T}_c$ and the tag set $\mathcal{T}' = \mathcal{T} - \{\mathcal{T}_i, \mathcal{T}_j\}$ with the restriction that it cannot query **Reveal**($\mathcal{T}_c$). Finally, $\mathcal{A}_2$ is asked to guess the random bit.

$\mathbf{Game}_{\mathcal{A}}^{ind}[\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}]$
$\triangleright$ Setup the reader $\mathcal{R}$ and a set of tags $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$;
$\triangleright$ $\{\mathcal{T}_i, \mathcal{T}_j, st\} \leftarrow \mathcal{A}_1^{\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3,\mathcal{O}_4}(\mathcal{R}, \mathcal{T})$;
$\triangleright$ Set $\mathcal{T}' = \mathcal{T} - \{\mathcal{T}_i, \mathcal{T}_j\}$;
$\triangleright$ $\beta \in_R \{0, 1\}$;
$\triangleright$ If $\beta = 0$ then $\mathcal{T}_c = \mathcal{T}_i$, else $\mathcal{T}_c = \mathcal{T}_j$;
$\triangleright$ $\beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3,\mathcal{O}_4}(\mathcal{R}, \mathcal{T}', \mathcal{T}_c, st)$.

Fig. 2. The *ind*-Privacy Game

Definition 3. *The advantage of an adversary $\mathcal{A}$ in the above game is defined as*

$$Adv_{\mathcal{A}}^{ind}(\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}) = |Pr[\beta' = \beta] - \frac{1}{2}|,$$

where the probability is taken over the choice of tag set $\mathcal{T}$ and the coin tosses of $\mathcal{A}$.

Definition 4. *An adversary $\mathcal{A}$ $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-breaks the ind-privacy of an RFID system $\mathcal{S}$ if the advantage $Adv_{\mathcal{A}}^{ind}(\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$ of $\mathcal{A}$ in the above game is at least ϵ and the running time of $\mathcal{A}$ is at most t.*

Definition 5. *An RFID system $\mathcal{S}$ is said to be $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-ind-privacy if there exists no adversary who can $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-break the ind-privacy of $\mathcal{S}$.*

2.4.2 The *unp*-Privacy Model

The goal of the adversary in the above *ind*-privacy game is to distinguish two different tags within its computational power and parameters. The idea is intuitively appealing; however, the *ind*-privacy model is difficult to apply *directly* in proving given a protocol is of *ind*-privacy. Juels and Weis [12] only prove the *ind*-privacy of a simple randomized hash-lock RFID protocol. To our knowledge, no mutual authentication RFID protocol has been proven *directly* to be of *ind*-privacy. Ha et al. [8] propose a different privacy model based on the unpredictability of tag outputs, denoted as *unp*-privacy. In fact, Juels and Weis [12] prove the *ind*-privacy of the randomized hash-lock RFID protocol by showing that no adversary can distinguish the real output of a tag from a random value. In other words, Juels and Weis [12] in fact prove the *unp*-privacy of the randomized hash-lock RFID protocol.

Figure 3 depicts the *unp*-privacy game $\mathbf{Game}_{\mathcal{A}}^{unp}[\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}]$, in which an adversary $\mathcal{A}$ is comprised of a pair of algorithms $(\mathcal{A}_1, \mathcal{A}_2)$. At first, a challenger initializes the RFID system by producing a reader $\mathcal{R}$ and a set of tags $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_n\}$ according to the security parameter κ. Then, $\mathcal{A}_1$ issues $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$ and $\mathcal{O}_4$ oracle queries, and outputs an uncorrupted tag $\mathcal{T}_c$ as the challenge tag. It also outputs a state information st which will be transmitted to algorithm $\mathcal{A}_2$. Next, the challenger selects a random bit β and sends m_2^* to

$\mathcal{A}_2$, where m_2^* is taken from $(m_1^*, m_2^*, m_3^*) \leftarrow \mathbf{Execute}(\mathcal{R}, \mathcal{T}_c)$ if $\beta = 1$, and $m_2^* \in_R \{0,1\}^{l_2}$ otherwise. Finally, $\mathcal{A}_2$ is asked to guess the random bit. Note that $\mathcal{A}_2$ is not allowed to query any oracle.

$\mathbf{Game}_{\mathcal{A}}^{unp}[\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}]$
- $\triangleright$ Setup the reader $\mathcal{R}$ and a set of tags $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$;
- $\triangleright$ $\{\mathcal{T}_c, st\} \leftarrow \mathcal{A}_1^{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4}(\mathcal{R}, \mathcal{T})$;
- $\triangleright$ $\beta \in_R \{0,1\}$;
- $\triangleright$ If β=1 then m_2^* is taken from $(m_1^*, m_2^*, m_3^*) \leftarrow \mathbf{Execute}(\mathcal{R}, \mathcal{T}_c)$, else $m_2^* \in_R \{0,1\}^{l_2}$;
- $\triangleright$ $\beta' \leftarrow \mathcal{A}_2(m_2^*, st)$.

Fig. 3. The *unp*-Privacy Game

Definition 6. *The advantage of an adversary $\mathcal{A}$ in the above game is defined as*

$$Adv_{\mathcal{A}}^{unp}(\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}) = |Pr[\beta' = \beta] - \frac{1}{2}|,$$

where the probability is taken over the choice of tag set $\mathcal{T}$ and the coin tosses of $\mathcal{A}$.

Definition 7. *An adversary $\mathcal{A}$ $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-breaks the unp-privacy of RFID system $\mathcal{S}$ if the advantage $Adv_{\mathcal{A}}^{unp}(\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$ of $\mathcal{A}$ in the above game is at least ϵ and the running time of $\mathcal{A}$ is at most t.*

Definition 8. *An RFID system $\mathcal{S}$ is said to be $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-unp-privacy if there exists no adversary who can $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-break the unp-privacy of $\mathcal{S}$.*

3 The *unp′*-Privacy Model, Revisited

Note that in the *unp*-privacy game, the adversary $\mathcal{A}_2$ does not get the full transcript of the protocol execution between the reader and the challenge tag, but only m_2^* which is either a random message or the message sent by the tag. As a result, an RFID protocol may have known weakness in privacy but can be shown to be of *unp*-privacy, as confirmed by Deursen and Radomirović [4]. At CCS'09, Ma et al. [13] propose an improved *unp*-privacy model, denoted as *unp′*-privacy. In the *unp′*-privacy model, the adversary is given not only m_2^*, but also the last message m_3^* of the protocol. The *unp′*-privacy model is robust for 2-round RFID protocols, as demonstrated in [13]; however, we will show in this section that the model has a deficiency when applied to 3-round protocols.

3.1 The Model

Figure 4 presents the unp'-privacy game $\mathbf{Game}_{\mathcal{A}}^{unp'}[\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}]$, in which an adversary $\mathcal{A}$ is comprised of a pair of algorithms $(\mathcal{A}_1, \mathcal{A}_2)$. At the start of the game, a challenger initializes the RFID system by producing $\mathcal{R}$ and $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_n\}$ according to the security parameter κ. Then, $\mathcal{A}_1$ issues $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$ and $\mathcal{O}_4$ oracle queries, and outputs an uncorrupted challenge tag $\mathcal{T}_c$ and a message m_1^*. It also outputs a state information st which will be transmitted to algorithm $\mathcal{A}_2$. Next, the challenger selects a random bit β and sends (m_2^*, m_3^*) to $\mathcal{A}_2$, where $(m_1^*, m_2^*, m_3^*) \leftarrow \mathbf{Execute}(\mathcal{R}, \mathcal{T}_c)$ if $\beta = 1$, and $(m_2^*, m_3^*) \in_R \{0,1\}^{l_2} \times \{0,1\}^{l_3}$ otherwise. Finally, $\mathcal{A}_2$ has oracle accesses to tags except $\mathcal{T}_c$ and is required to infer the value of β.

$\mathbf{Game}_{\mathcal{A}}^{unp'}[\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}]$
- $\triangleright$ Setup the reader $\mathcal{R}$ and a set of tags $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$;
- $\triangleright$ $\{\mathcal{T}_c, m_1^*, st\} \leftarrow \mathcal{A}_1^{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4}(\mathcal{R}, \mathcal{T})$;
- $\triangleright$ Set $\mathcal{T}' = \mathcal{T} - \{\mathcal{T}_c\}$;
- $\triangleright$ $\beta \in_R \{0,1\}$;
- $\triangleright$ If $\beta = 1$ then $(m_1^*, m_2^*, m_3^*) \leftarrow \mathbf{Execute}(\mathcal{R}, \mathcal{T}_c)$, else $(m_2^*, m_3^*) \in_R \{0,1\}^{l_2} \times \{0,1\}^{l_3}$;
- $\triangleright$ $\beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4}(\mathcal{R}, \mathcal{T}', m_2^*, m_3^*, st)$.

Fig. 4. The unp'-Privacy Game

Definition 9. *The advantage of an adversary $\mathcal{A}$ in the above game is defined as*

$$Adv_{\mathcal{A}}^{unp'}(\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}) = |Pr[\beta' = \beta] - \frac{1}{2}|,$$

where the probability is taken over the choice of tag set $\mathcal{T}$ and the coin tosses of $\mathcal{A}$.

Definition 10. *An adversary $\mathcal{A}$ $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-breaks the unp'-privacy of RFID system $\mathcal{S}$ if the advantage $Adv_{\mathcal{A}}^{unp'}(\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$ of $\mathcal{A}$ in the above game is at least ϵ and the running time of $\mathcal{A}$ is at most t.*

Definition 11. *An RFID system $\mathcal{S}$ is said to be $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-$unp'$-privacy if there exists no adversary who can $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-break the unp'-privacy of $\mathcal{S}$.*

3.2 A Counterexample

Ma et al. [13] introduce an efficient 2-round protocol and prove that it is of unp'-privacy. We now modify the protocol to a 3-round mutual authentication protocol and show that the new protocol has clear weakness with respect to

privacy but can be proven to be of unp'-privacy. This example therefore exposes a deficiency of the unp'-privacy model when it is applied to 3-round mutual authentication protocols.

Let $F : \{0,1\}^{l_k} \times \{0,1\}^{l_d} \rightarrow \{0,1\}^{l_r}$ be a PRF family. Let $ctr \in \{0,1\}^{l_r}$ be a counter, $pad_1 \in \{0,1\}^{l_{p1}}$ and $pad_2 \in \{0,1\}^{l_{p2}}$ be two paddings such that $l_r + l_{p1} = l_d$. The RFID system is constructed as follows.

$\mathsf{Initialize}(\kappa)$: It randomly chooses a key $k_i \in \{0,1\}^{l_k}$ for each tag $\mathcal{T}_i \in \mathcal{T}$. $\mathcal{T}_i$ stores k_i, a counter $ctr_i \in \{0,1\}^{l_r}$, and a 1-bit flag s_i in its memory. Initially, $ctr_i = 1$ and $s_i = 0$. It also associates $\mathcal{T}_i$ with a unique ID_i, and stores the tuple $(I_i, k_i, ctr_i, \mathsf{ID}_i)$ in the back-end database $\mathcal{DB}$, where $I_i = F_{k_i}(ctr_i \| pad_1)$.

$\mathsf{Execute}(\mathcal{R}, \mathcal{T}_i)$: $\mathcal{R}$ first sends a challenge $c \in_R \{0,1\}^{l_c}$ to $\mathcal{T}_i$, where $l_c + l_r + l_{p2} = l_d$. Upon receiving c, $\mathcal{T}_i$ executes the following steps:

1. Randomly choose $r_2 \in \{0,1\}^{l_{p2}}$ and compute $I_i = F_{k_i}(ctr_i \| pad_1)$;
2. Set $r_1 = F_{k_i}(c \| I_i \| pad_2) \oplus ctr_i$ if $s_i = 0$, else set $r_1 = F_{k_i}(c \| I_i \| r_2) \oplus ctr_i$;
3. Respond with $(r_1 \| I_i, r_2)$, increment ctr_i by 1 and set $s_i = 1$.

Upon receiving the response $(r_1 \| I_i, r_2)$, $\mathcal{R}$ identifies the tag from its database as follows:

1. Search for the tuple $(I_i, k_i, ctr'_i, \mathsf{ID}_i)$ using I_i as an index. If such a tuple exists, compute $F_{k_i}(c \| I_i \| pad_2)$ and then
   (a) If $ctr'_i = F_{k_i}(c \| I_i \| pad_2) \oplus r_1$, update $ctr'_i = ctr'_i + 1$ and $I_i = F_{k_i}(ctr'_i \| pad_1)$, respond with $f = F_{k_i}(c \| ctr'_i \| r_2)$ and accept the tag;
   (b) Else abort the protocol.
2. Else look up the database for a tuple $(I'_i, k_i, ctr'_i, \mathsf{ID}_i)$ in an exhaustive search such that $ctr_i = F_{k_i}(c \| I_i \| r_2) \oplus r_1$ and $F_{k_i}(ctr_i \| pad_1) = I_i$. Then
   (a) If such a tuple exists, update $ctr'_i = ctr_i + 1$ and $I'_i = F_{k_i}(ctr'_i || pad_1)$, respond with $f = F_{k_i}(c \| ctr'_i \| r_2)$ and accept the tag;
   (b) Else abort the protocol.

Upon receiving f, $\mathcal{T}_i$ checks whether $f = F_{k_i}(c \| ctr_i \| r_2)$. If not, $\mathcal{T}_i$ rejects the reader. Else, $\mathcal{T}_i$ sets $s_i = 0$ and accepts the reader.

Note that the $\mathsf{ReaderStart}$, $\mathsf{TagCompute}$ and $\mathsf{ReaderCompute}$ algorithms are not shown explicitly in the above description since they are embedded in the $\mathsf{Execute}$ algorithm. The protocol is depicted in 5.

A flaw of the protocol is that an active attacker can find out whether a tag's state is $s = 0$ or $s = 1$. If a tag is in state $s = 0$, the reader does not verify the integrity of r_2; while if the tag is in state $s = 1$, this verification occurs implicitly. Note that under normal circumstances tags will be in state $s = 0$. Hence, an active attacker can flag a tag by setting its state to $s = 1$ and trace the tag in subsequent protocol sessions. However, the following theorem states that the protocol is of unp'-privacy.

Theorem 1. *The above mutual authentication RFID protocol is of* $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$*-*$unp'$*-privacy, assuming the function family* $F : \{0,1\}^{l_k} \times \{0,1\}^{l_d} \rightarrow \{0,1\}^{l_r}$ *is* (t', q, ϵ')*-pseudorandom, where*

$$t' \approx t,\ q \approx q_{st} + q_{sr},\ \epsilon' = \epsilon / n.$$

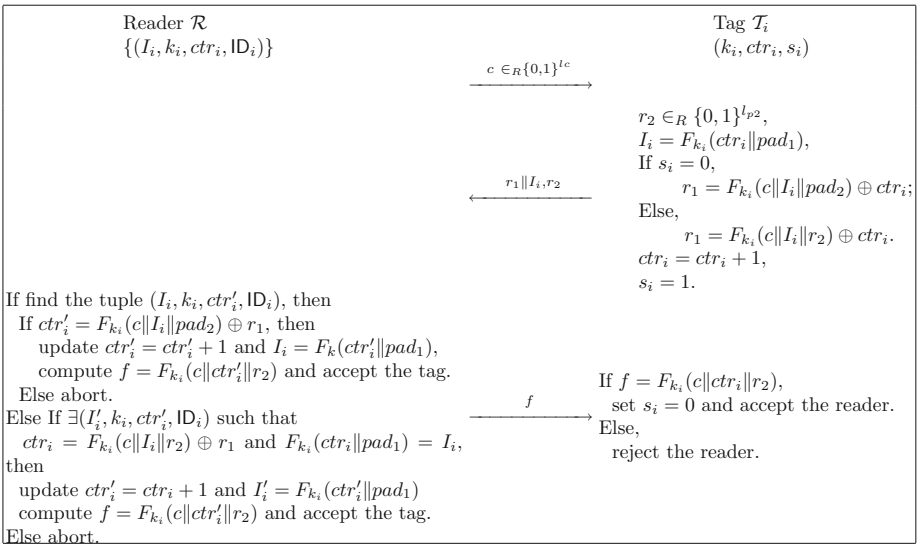

Fig. 5. The Counterexample RFID Protocol

Proof. Suppose there exists an adversary $\mathcal{A}$ $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-breaks the unp'-privacy of the RFID protocol in Figure 5. We are going to construct another PPT $\mathcal{B}$ that makes use of $\mathcal{A}$ to (t', q, ϵ')-break the pseudorandomness of the function family F.

$\mathcal{B}$ is provided oracle access to a function g and tries to decide if g is drawn at random from F, namely $g \in_R F$ (which means that a key is chosen via $k \in_R \{0,1\}^{l_k}$ and then g is set to F_k.), or is drawn at random from $\mathsf{Rand}^{\{0,1\}^{l_d} \to \{0,1\}^{l_r}}$, namely $g \in_R \mathsf{Rand}^{\{0,1\}^{l_d} \to \{0,1\}^{l_r}}$. $\mathcal{B}$'s goal is to output 0 if $g \in_R \mathsf{Rand}^{\{0,1\}^{l_d} \to \{0,1\}^{l_r}}$ and 1 otherwise. $\mathcal{B}$ runs $\mathcal{A}$ as a subroutine and proceeds as follows.

Setup $\mathcal{B}$ randomly chooses an index $j \in \{1, \ldots, n\}$. Without loss of generality, we assume $j = n$. $\mathcal{B}$ then randomly chooses a key $k_i \in \{0,1\}^{l_k}$ for each tag $\mathcal{T}_i \in \{\mathcal{T}_1, \ldots, \mathcal{T}_{n-1}\}$. $\mathcal{T}_i$ stores k_i, a counter $ctr_i \in \{0,1\}^{l_r}$, and a 1-bit flag s_i. Initially, $ctr_i = 1$ and $s_i = 0$. $\mathcal{B}$ associates $\mathcal{T}_i$ with a unique ID_i, and stores the tuple $(I_i, k_i, ctr_i, \mathsf{ID}_i)$ in the database $\mathcal{DB}$, where $I_i = F_{k_i}(ctr_i \| pad_1)$.
$\mathcal{B}$ associates tag $\mathcal{T}_n$ with a unique ID_n. $\mathcal{T}_n$ keeps a counter ctr_n and a 1-bit flag s_n with initially values 1 and 0, respectively. *Note that, the key k_n of $\mathcal{T}_n$ is unknown to $\mathcal{B}$.* $\mathcal{B}$ queries its oracle on $ctr_n \| pad_1$ and gets the response of the oracle I_n. $\mathcal{B}$ stores the tuple $(I_n, *, ctr_n, \mathsf{ID}_n)$ in the database $\mathcal{DB}$.
In the following, we let $x = ctr_n \| pad_1$, $m'_1 \| I_n \| pad_2$ or $m'_1 \| I_n \| r_2$ depending on the context. The basic idea is that $\mathcal{B}$ queries its oracle g on x and gets either $F_{k_n}(x)$ or a random message as response .

Query phase 1 $\mathcal{A}$ issues **Launch**, **SendTag**, **SendReader** and **Reveal** queries to which $\mathcal{B}$ answers as follows:

- **Launch** query on $\mathcal{R}$: $\mathcal{B}$ responds according to the protocol.
- **SendTag** query on $(sid, m'_1, \mathcal{T}_i)$: Respond according to the protocol. Note that $\mathcal{B}$ can do it for $i \in \{1, \ldots, n-1\}$ because $\mathcal{B}$ knows the keys and the internal information of $\mathcal{T}_i$. For $\mathcal{T}_n$, $\mathcal{B}$ also can do it by querying its oracle on x whenever it needs to compute $F_{k_n}(x)$.
- **SendReader** query on (sid, m'_2): Respond according to the protocol. Whenever $\mathcal{B}$ needs to compute $F_{k_n}(x)$, $\mathcal{B}$ queries its oracle on x.
- **Reveal** query on $\mathcal{T}_i$: If $\mathcal{T}_i = \mathcal{T}_n$, abort and randomly output a bit; else, forward the key k_i and internal state (ctr_i, s_i) of $\mathcal{T}_i$ to $\mathcal{A}$.

Challenge $\mathcal{A}$ submits a message $m_1 \in \{0,1\}^{l_c}$ and an uncorrupted challenge tag $\mathcal{T}_c$ to $\mathcal{B}$ which proceeds as follows:

- If $T_c \neq T_n$, abort and randomly output a bit.
- Else, randomly choose $r_2 \in \{0,1\}^{l_{p2}}$.
- Set $x = ctr_n \| pad_1$, query its oracle on x and get the response I_n.
- If $s_n = 0$, query its oracle on $x = m_1 \| I_n \| pad_2$, get the response y and set $r_1 = y \oplus ctr_n$; else query its oracle on $x = m_1 \| I_n \| r_2$, get the response y and set $r_1 = y \oplus ctr_n$.
- Set $m_2 = (r_1 \| I_n, r_2)$, update $ctr_n = ctr_n + 1$ and $s_n = 0$.
- Query its oracle on $m_1 \| ctr_n \| r_2$, get the response m_3, and send (m_2, m_3) to $\mathcal{A}$.

Query phase 2 Let $\mathcal{T}'$ denote the tag set $\mathcal{T} - \mathcal{T}_c = \{\mathcal{T}_1, \ldots, \mathcal{T}_{n-1}\}$. $\mathcal{A}$ continues to issue **Launch**, **SendTag**, **SendReader** and **Reveal** queries. $\mathcal{B}$ answers them in as follows:

- **Launch** query on $\mathcal{R}$: Respond according to the protocol.
- **SendTag** query on $(sid, m'_1, \mathcal{T}_i \in \mathcal{T}')$: Respond according to the protocol. $\mathcal{B}$ can do it because it knows the keys and the internal information of the tags in $\mathcal{T}'$.
- **SendReader** query on (sid, m'_2): Respond according to the protocol. $\mathcal{B}$ can do it because it knows $\mathcal{DB}$.
- **Reveal** query on $\mathcal{T}_i \in \mathcal{T}'$: Forward $\mathcal{T}_i$'s key k_i and internal state (ctr_i, s_i) to $\mathcal{A}$.

Guess $\mathcal{A}$ outputs a bit β' which $\mathcal{B}$ also takes as its output.

If $\mathcal{B}$ does not abort during the simulation, $\mathcal{B}$'s simulation is perfect and are identically distributed as the real one from the construction. It is obvious that the probability that $\mathcal{B}$ does not abort during the simulation is $1/n$. In the simulation, $\mathcal{B}$ needs to query its oracle in response to $\mathcal{A}$'s **SendTag** and **SendReader** queries. So, $q \approx q_{st} + q_{sr}$. The running time of $\mathcal{B}$ is approximately that of $\mathcal{A}$. This completes the proof.

4 Our Model and Results

The limitation in the definition of the unp'-privacy model, as shown in the above example, is due to the constraint imposed on the adversary $\mathcal{A}_2$, i. e., $\mathcal{A}_2$ only has access to m_2^* and m_3^* as supplied by the challenger and is not allowed to query

oracles on the challenge tag $\mathcal{T}_c$. In this section, we propose a new RFID privacy model, denoted as unp^*-privacy, as a remedy to this problem.

The intuition of the unp^*-privacy model is that no adversary should be able to distinguish the output of a real tag from that of a virtual tag, which is defined as a tag without any secret information. This implies that no adversary can link a real tag and its behavior without learning its secret key. We emphasis that our unp^*-privacy model does not impose any restrictions on the number of rounds in an RFID protocol. In what follows, we introduce the unp^*-privacy model, investigate the relationship between this new model and the *ind*-privacy model. We also extend the 2-round RFID protocol in [13] to a 3-round mutual authentication protocol and show that it is of unp^*-privacy.

4.1 The unp^*-Privacy Model

Figure 6 illustrates the unp^*-privacy game $\mathbf{Game}_{\mathcal{A}}^{unp^*}[\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}]$ between an adversary $\mathcal{A}$ and a challenger, in which $\mathcal{A}$ consists of a pair of algorithms $(\mathcal{A}_1, \mathcal{A}_2)$. The challenger initializes the RFID system $\mathcal{S}$ by producing a reader $\mathcal{R}$ and a set of tags $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_n\}$ according to the security parameter κ. Then, $\mathcal{A}_1$ issues $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$ and $\mathcal{O}_4$ oracle queries, and outputs an uncorrupted challenge tag $\mathcal{T}_c$. It also outputs a state information st which will be transmitted to algorithm $\mathcal{A}_2$. Next, the challenger selects a random bit β. Finally, $\mathcal{A}_2$ is asked to guess the value of the random bit. $\mathcal{A}_2$ is allowed to query $\mathcal{O}_1, \mathcal{O}_2$ and $\mathcal{O}_3$ oracles on $\mathcal{R}$ and $\mathcal{T}_c$. The challenger responds to $\mathcal{A}_2$ queries as follows:

- **Launch** query on $\mathcal{R}$: If $\beta = 0$, generate a new session identifier sid, randomly choose $m_1 \in \{0,1\}^{l_1}$ and forward (sid, m_1) to $\mathcal{A}_2$; else, run the algorithm $\mathsf{ReaderStart}$, and forward the result to $\mathcal{A}_2$.
- **SendTag** query on $(sid, m_1', \mathcal{T}_c)$: If $\beta = 0$, randomly choose $m_2 \in \{0,1\}^{l_2}$ and forward m_2 to $\mathcal{A}_2$; else, run the algorithm $\mathsf{TagCompute}(sid, m_1', \mathcal{T}_c)$ and forward the result to $\mathcal{A}_2$.
- **SendReader** query on input (sid, m_2'): If $\beta = 0$, randomly choose $m_3 \in \{0,1\}^{l_3}$ and forward m_3 to $\mathcal{A}_2$; else, run the algorithm $\mathsf{ReaderCompute}(sid, m_3)$ and forward the result to $\mathcal{A}_2$.

Definition 12. *The advantage of an adversary $\mathcal{A}$ in the above game is defined as*

$$Adv_{\mathcal{A}}^{unp^*}(\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}) = |Pr[\beta' = \beta] - \frac{1}{2}|,$$

where the probability is taken over the choice of tag set $\mathcal{T}$ and the coin tosses of $\mathcal{A}$.

Definition 13. *An adversary $\mathcal{A}$ $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-breaks the unp^*-privacy of an RFID system $\mathcal{S}$ if the advantage $Adv_{\mathcal{A}}^{unp^*}(\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$ of $\mathcal{A}$ in the above game is at least ϵ and the running time of $\mathcal{A}$ is at most t.*

Definition 14. *An RFID system $\mathcal{S}$ is said to be $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-$unp^*$-privacy if there exists no adversary who can $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-break the unp^*-privacy of $\mathcal{S}$.*

$\mathbf{Game}_{\mathcal{A}}^{unp^*}[\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}]$
$\triangleright$ Setup the reader $\mathcal{R}$ and a set of tags $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$;
$\triangleright$ $\{\mathcal{T}_c, st\} \leftarrow \mathcal{A}_1^{\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3,\mathcal{O}_4}(\mathcal{R}, \mathcal{T})$;
$\triangleright$ $\beta \in_R \{0, 1\}$;
$\triangleright$ $\beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3}(\mathcal{R}, \mathcal{T}_c, st)$.

Fig. 6. The unp^*-Privacy Game

4.2 Relationship with *ind*-Privacy Model

In order to clarify the relationship between the *ind*-privacy and unp^*-privacy, we introduce another model, called ind^*-privacy model, as a "bridge" between the two models. We first show that ind^*-privacy is equivalent to *ind*-privacy and then prove that unp^*-privacy implies ind^*-privacy and hence *ind*-privacy.

Figure 7 shows the ind^*-privacy game $\mathbf{Game}_{\mathcal{A}}^{ind^*}[\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}]$. The ind^*-privacy game is identical to the *ind*-privacy game given in Figure 2 except that $\mathcal{A}_2$ in the former is only allowed to query $\mathcal{O}_1, \mathcal{O}_2$, and $\mathcal{O}_3$ oracles on $\mathcal{R}$ and $\mathcal{T}_c$.

$\mathbf{Game}_{\mathcal{A}}^{ind^*}[\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}]$
$\triangleright$ Setup reader $\mathcal{R}$ and a set of tags $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$;
$\triangleright$ $\{\mathcal{T}_i, \mathcal{T}_j, st\} \leftarrow \mathcal{A}_1^{\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3,\mathcal{O}_4}(\mathcal{R}, \mathcal{T})$;
$\triangleright$ $\beta \in_R \{0, 1\}$;
$\triangleright$ If $\beta = 0$ then $\mathcal{T}_c = \mathcal{T}_i$, else $\mathcal{T}_c = \mathcal{T}_j$;
$\triangleright$ $\beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3}(\mathcal{R}, \mathcal{T}_c, st)$.

Fig. 7. The ind^*-Privacy Game

Definition 15. *The advantage of an adversary $\mathcal{A}$ in the above game is defined as*

$$Adv_{\mathcal{A}}^{ind^*}(\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv}) = |Pr[\beta' = \beta] - \frac{1}{2}|,$$

where the probability is taken over the choice of tag set $\mathcal{T}$ and the coin tosses of the adversary $\mathcal{A}$.

Definition 16. *An adversary $\mathcal{A}$ $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-breaks the ind^*-privacy of RFID system $\mathcal{S}$ if the advantage $Adv_{\mathcal{A}}^{ind^*}(\kappa, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$ of $\mathcal{A}$ in the above game is at least ϵ and the running time of $\mathcal{A}$ is at most t.*

Definition 17. *An RFID system $\mathcal{S}$ is said to be $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-$ind^*$-privacy if there exists no adversary who can $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-break the ind^*-privacy of $\mathcal{S}$.*

Theorem 2. *For an RFID system $\mathcal{S}$, the ind-privacy and the ind*-privacy are equivalent.*

Proof. It is obvious that *ind*-privacy $\Longrightarrow$ *ind**-privacy holds. Now we prove that *ind*-privacy $\Longleftarrow$ *ind**-privacy also holds.

Suppose there exists an adversary $\mathcal{A}$ $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-breaks the *ind*-privacy of the RFID system $\mathcal{S}$. We are going to construct another PPT $\mathcal{B}$ that makes use of $\mathcal{A}$ $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv} + n - 1)$-breaks the *ind**-privacy of the RFID system $\mathcal{S}$. Let $\mathcal{C}$ denote an *ind**-privacy challenger against $\mathcal{B}$. $\mathcal{B}$ runs $\mathcal{A}$ executing the following steps.

Setup $\mathcal{B}$ maintains a list KS-List. Initially the list is empty.

Query phase 1 $\mathcal{A}$ issues **Launch**, **SendTag**, **SendReader** and **Reveal** queries. $\mathcal{B}$ answers them in the following way:

- **Launch** query on $\mathcal{R}$: Issue a **Launch** query on $\mathcal{R}$ to $\mathcal{C}$ and forward the result to $\mathcal{A}$.
- **SendTag** query on $(sid, m'_1, \mathcal{T}_i \in \mathcal{T})$: Issue a **SendTag** query on $(sid, m'_1, \mathcal{T}_i)$ to $\mathcal{C}$ and forward the result to $\mathcal{A}$.
- **SendReader** query on (sid, m'_2): Issue a **SendReader** query on (sid, m'_2) to $\mathcal{C}$ and forward the result to $\mathcal{A}$.
- **Reveal** query on $\mathcal{T}_i \in \mathcal{T}$: Issue a **Reveal** query on $\mathcal{T}_i$ to $\mathcal{C}$ and forward the result to $\mathcal{A}$.

Challenge Adversary $\mathcal{A}$ submits two uncorrupted tags $\mathcal{T}_{c0}, \mathcal{T}_{c1} \in \mathcal{T}$. $\mathcal{B}$ submits the same two tags $\mathcal{T}_{c0}$ and $\mathcal{T}_{c1}$ to $\mathcal{C}$ which responds with a challenge tag $\mathcal{T}_c \in \{\mathcal{T}_{c0}, \mathcal{T}_{c1}\}$. Then $\mathcal{B}$ *issues Reveal queries on the tag set* $\mathcal{T} - \{\mathcal{T}_{c0}, \mathcal{T}_{c1}\}$ *and stores the results in the list KS-List.* $\mathcal{B}$ forwards $\mathcal{T}_c$ to $\mathcal{A}$. Let $\mathcal{T}'$ denote the tag set $\mathcal{T} - \{\mathcal{T}_{c0}, \mathcal{T}_{c1}\} + \mathcal{T}_c$.

Query phase 2 $\mathcal{A}$ continues to issue **Launch**, **SendTag**, **SendReader** and **Reveal** queries. $\mathcal{B}$ answers them in the following way:

- **Launch** query on $\mathcal{R}$: Issue a **Launch** query on $\mathcal{R}$ to $\mathcal{C}$ and forward the result to $\mathcal{A}$.
- **SendTag** query on $(sid, m'_1, \mathcal{T}_i \in \mathcal{T}')$: If $\mathcal{T}_i = \mathcal{T}_c$, issue a **SendTag** query on $(sid, m'_1, \mathcal{T}_i)$ and forward the result to $\mathcal{A}$; else, use the list KS-List to respond.
- **SendReader** query on (sid, m'_2): Use **SendReader** oracle and the list KS-List to respond.
- **Reveal** query on $\mathcal{T}_i \in \{\mathcal{T} - \{\mathcal{T}_{c0}, \mathcal{T}_{c1}\}\}$: Use the list KS-List to respond.

Guess $\mathcal{A}$ outputs a bit β' which $\mathcal{B}$ also takes as its output.

It is obvious that the simulation is perfect. Thus we have shown an adversary $\mathcal{A}$ against the *ind*-privacy of the RFID system $\mathcal{S}$ with advantage ϵ can be used to construct another adversary $\mathcal{B}$ against the *ind**-privacy of the same RFID system with an identical advantage. Note that, the number of times that $\mathcal{B}$ queries the **Reveal** oracle is $q_{rv} + n - 1$. The running time of $\mathcal{B}$ is approximate to that of $\mathcal{A}$. This completes the proof.

Theorem 3. *Assume that an RFID system $\mathcal{S}$ is of $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-unp*-privacy, then it is also of $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-ind*-privacy.*

Proof. Suppose there exists an adversary $\mathcal{A}$ $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-breaks the ind^*-privacy of the RFID system $\mathcal{S}$. We are going to construct another PPT $\mathcal{B}$ that makes use of $\mathcal{A}$ $(\epsilon, t, n, q_{ini}, q_{st}, q_{sr}, q_{rv})$-breaks the unp^*-privacy of the same RFID system $\mathcal{S}$. Let $\mathcal{C}$ denote an unp^*-privacy challenger against $\mathcal{B}$. $\mathcal{B}$ runs $\mathcal{A}$ executing the following steps.

Setup $\mathcal{B}$ does nothing.

Query phase 1 $\mathcal{A}$ issues **Launch**, **SendTag**, **SendReader** and **Reveal** queries. $\mathcal{B}$ answers them in the following way:

- **Launch** query on $\mathcal{R}$: Issue a **Launch** query on $\mathcal{R}$ to $\mathcal{C}$ and forward the result to $\mathcal{A}$.
- **SendTag** query on $(sid, m'_1, \mathcal{T}_i \in \mathcal{T})$: Issue a **SendTag** query on $(sid, m'_1, \mathcal{T}_i)$ to $\mathcal{C}$ and forward the result to $\mathcal{A}$.
- **SendReader** query on (sid, m'_2): Issue a **SendReader** query on (sid, m'_2) to $\mathcal{C}$ and forward the result to $\mathcal{A}$.
- **Reveal** query on $\mathcal{T}_i \in \mathcal{T}$: Issue a **Reveal** query on $\mathcal{T}_i$ to $\mathcal{C}$ and forward the result to $\mathcal{A}$.

Challenge $\mathcal{A}$ submits two uncorrupted tags $\mathcal{T}_{c0}, \mathcal{T}_{c1} \in \mathcal{T}$. $\mathcal{B}$ selects a random bit $\beta \in \{0, 1\}$ and sets the challenge tag $\mathcal{T}_c = \mathcal{T}_{c0}$ if $\beta = 0$ and $\mathcal{T}_c = \mathcal{T}_{c1}$ otherwise. $\mathcal{B}$ submits $\mathcal{T}_c$ to $\mathcal{C}$.

Query phase 2 The adversary continues to issue **Launch**, **SendTag** and **Send Reader** queries. $\mathcal{B}$ answers them as follows:

- **Launch** query on $\mathcal{R}$: Issue a **Launch** query on $\mathcal{R}$ to $\mathcal{C}$ and forward the result to $\mathcal{A}$.
- **SendTag** query on $(sid, m'_1, \mathcal{T}_c)$: Issue a **SendTag** query on $(sid, m'_1, \mathcal{T}_c)$ to $\mathcal{C}$ and forward the result to $\mathcal{A}$.
- **SendReader** query on (sid, m'_2): Issue a **SendReader** query on (sid, m'_2) to $\mathcal{C}$ and forward the result to $\mathcal{A}$.

Guess $\mathcal{A}$ outputs a bit β'. If $\beta = \beta'$, $\mathcal{B}$ outputs 1; else, $\mathcal{B}$ outputs 0.

The simulation of $\mathcal{B}$ is perfect. When the binary coin flipped by the unp^*-privacy challenger $\mathcal{C}$ is equal to 1, the probability of $\beta = \beta'$ is equal to $1/2 \pm \epsilon$; otherwise, the probability of $\beta = \beta'$ is equal to $1/2$, because in this case the challenge tag $\mathcal{T}_c$ is in fact a virtual tag in adversary $\mathcal{A}$'s view. Hence, the advantage of $\mathcal{B}$ is equal to that of $\mathcal{A}$ (i. e., ϵ). The running time of $\mathcal{B}$ is exactly the same as that of $\mathcal{A}$. This completes the proof.

Theorem 4. *There exists an RFID system that is of ind^*-privacy but is not of unp^*-privacy.*

Proof. Suppose an RFID system $\mathcal{S} = (\mathcal{R}, \mathcal{T}, \mathcal{DB}, \mathsf{Initialize}, \mathsf{Execute})$ is of ind^*-privacy, and the output of the algorithm $\mathsf{Execute}$ is (c, r, f). We construct a new RFID system $\mathcal{S}' = (\mathcal{R}, \mathcal{T}, \mathcal{DB}, \mathsf{Initialize}, \mathsf{Execute}')$ such that $(c, r \| r, f) \leftarrow \mathsf{Execute}'$. It is easy to see that $\mathcal{S}'$ is also of ind^*-privacy. Since every protocol transcript of $\mathcal{S}'$ is of the form $(c, r \| r, f)$, an adversary can easily distinguish it from a random tuple $(c', r_1 \| r_2, f')$ by checking whether $r_1 = r_2$. Therefore, $\mathcal{S}'$ is not of unp^*-privacy.

4.3 A Protocol with *unp**-Privacy

We now present a 3-round mutual authentication protocol with *unp**-privacy by modifying the 2-round protocol in [13]. This RFID protocol is shown in Figure 8. It is important to note that when the reader $\mathcal{R}$ fails to identify a tag, it does not simply abort, but responds with a random message. A detailed description of the protocol and its proof of *unp**-privacy are given in the full version.

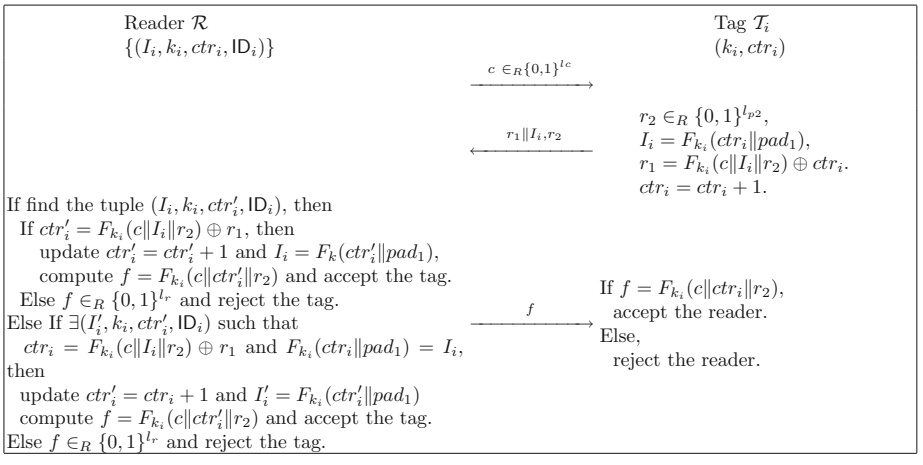

Fig. 8. The Mutual Authentication Protocol with *unp**-Privacy

5 Conclusions

In this paper we first revisited the formal privacy models for RFID systems existing in the literature, including the *ind*-privacy model [12], the *unp*-privacy model [8] and the newly proposed *unp*′-privacy model [13]. In doing so, we have highlighted their potential limitations or flaws. In particular, for the first time, we pointed out that though the *unp*′-privacy model is robust when applied to 2-round RFID protocols but has a deficiency in dealing with 3-round mutual authentication RFID protocols. This deficiency arises from the constraint that the adversary in the guessing stage of the *unp*′-privacy game is not given any oracle access to the challenge tag. We demonstrated this through a counterexample protocol which has problem with respect to privacy but can be proven to be of *unp*′-privacy.

We proposed a new privacy model, denoted as *unp**-privacy, based on the indistinguishability of the output of a real tag from that of a virtual tag (e. g., a tag without any secret key). The adversary in the *unp**-privacy game is given multiple oracle accesses to the challenge tag in the guessing stage. The new model does not suffer from the limitations of the *unp*-privacy and the *unp*′-privacy models. Furthermore, we formally established the relationship between the *ind*-privacy and the *unp**-privacy notions by proving that the former is

weaker than the latter. Finally, we extended the 2-round RFID protocol in [13] to a 3-round mutual authentication RFID protocol and showed that it is of unp^*-privacy.

Acknowledgement

We are grateful to the anonymous reviewers for their helpful comments. This work is partly supported by the Office of Research, Singapore Management University, and also supported in part by A*Star SERC Grant No. 082 101 0022 in Singapore.

References

1. Ateniese, G., Camenisch, J., de Medeiros, B.: Untraceable RFID Tags via Insubvertible Encryption. In: Conference on Computer and Communications Security, CCS '05, pp. 92–101 (2005)
2. Avoine, G.: Security and privacy in RFID systems (2006), `http://lasecwww.epfl.ch/~gavoine/rfid/`
3. Avoine, G.: Adversary Model for Radio Frequency Identification. Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC) (2005)
4. van Deursen, T., Radomirović, S.: On a New Formal Proof Model for RFID Location Privacy. Cryptology ePrint Archive, Report 2008/477.
5. Fishkin, K., Roy, S., Jiang, B.: Some methods for privacy in RFID communication. In: Castelluccia, C., Hartenstein, H., Paar, C., Westhoff, D. (eds.) ESAS 2004. LNCS, vol. 3313, pp. 42–53. Springer, Heidelberg (2005)
6. Golle, P., Jakobsson, M., Juels, A., Syverson, P.: Universal re-encryption for mixnets. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 163–178. Springer, Heidelberg (2004)
7. Henrici, D., Müller, P.: Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In: Workshop on Pervasive Computing and Communications Security-PerSec 2004, pp. 149–153. IEEE Computer Society, Los Alamitos (2004)
8. Ha, J., Moon, S., Zhou, J., Ha, J.: A new formal proof model for RFID location privacy. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 267–281. Springer, Heidelberg (2008)
9. Juels, A.: RFID Security and Privacy: A Research Survey. IEEE Journal on Selected Areas in Communications 24(2), 381–394 (2006)
10. 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)
11. Juels, A., Rivest, R., Szydlo, M.: The blocker tag: Selective blocking of RFID tags for consumer privacy. In: CCS '03, pp. 103–111. ACM Press, New York (2003)
12. Juels, A., Weis, S.A.: Defining strong privacy for RFID. ePrint, Report 2006/137, 2006, PerCom 2007
13. Ma, C., Li, Y., Deng, R.H., Li, T.: RFID Privacy: Relation Between Two Notions, Minimal Condition, and Efficient Construction. In: ACM CCS 2009 (2009)

14. Ohkubo, M., Suzuki, K., Kinoshita, S.: Efficient Hash-Chain Based RFID Privacy Protection Scheme. In: International Conference on Ubiquitous Computing-Ubicomp, Workshop Privacy: Current Status and Future Directions (2004)
15. Pedro, P.L., Cesar, H.C.J., Juan, M.E.T., Arturo, R.: RFID Systems: A Survey on Security Threats and Proposed Solutions. In: Cuenca, P., Orozco-Barbosa, L. (eds.) PWC 2006. LNCS, vol. 4217, pp. 159–170. Springer, Heidelberg (2006)
16. Paise, R.-I., Vaudenay, S.: Mutual authentication in RFID: Security and privacy. In: Proc. of ASIACCS, pp. 292–299. ACM Press, New York (2008)
17. Sadeghi, A.-R., Visconti, I., Wachsmann, C.: Anonymizer-enabled security and privacy for RFID. In: Miyaji, A., Echizen, I., Okamoto, T. (eds.) CANS 2009. LNCS, vol. 5888, pp. 134–153. Springer, Heidelberg (2009)
18. Saito, J., Ryou, J.-C., 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)
19. Tsudik, G.: YA-TRAP: Yet Another Trivial RFID Authentication Protocol. In: International Conference on Pervasive Computing and Communications, PerCom 2006, pp. 640–643 (2006)
20. Vaudenay, S.: On privacy models for RFID. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)
21. Weis, S., Sarma, S., Rivest, R., Engels, D.: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In: International Conference on Security in Pervasive Computing-SPC 2003 (2003)
22. Yu Ng, C., Susilo, W., Mu, Y., Safavi-Naini, R.: RFID privacy models revisited. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 251–266. Springer, Heidelberg (2008)