RFID Privacy Models Revisited

Ching Yu Ng[1], Willy Susilo[1], Yi Mu[1], and Rei Safavi-Naini[2]

[1] Centre for Computer and Information Security Research (CCISR)
University of Wollongong, Australia
{cyn27,wsusilo,ymu}@uow.edu.au
[2] Department of Computer Science, University of Calgary, Canada
rei@ucalgary.ca

Abstract. In Asiacrypt 2007, Vaudenay proposed a formal model addressing privacy in RFID, which separated privacy into eight classes. One important conclusion in the paper is the impossibility of achieving strong privacy in RFID. He also left an open question whether forward privacy without PKC is possible. In our paper, first we revisit the eight RFID privacy classes and simplify them into three classes that will address the same goal. Second, we show that strong privacy in RFID is achievable. Third, we answer the open question by pointing out the possibility to achieve forward privacy without PKC both within Vaudenay's model and in practice.

1 Privacy in RFID

Radio frequency identification (RFID) systems are designed to uniquely identify any RFID tagged objects from a distance by using authorized RFID readers to pick up the responses from RFID tags. Reasonable concerns on privacy have been raised. In particular, when individual entities are bound with these RFID tagged objects (e.g. human implantation [1]), even the location of the tag wearer can be compromised [2]. For these reasons, adequate privacy protections have been devoted as the main research area in RFID (eg. [3,4,5,6,7,8,9,1,10,11,12,13,14,15]).

To measure the privacy level of various RFID protocols, we need a formal model that defines privacy, available resources in the system and abilities of different classes of adversaries. In an effort to design a widely accepted privacy model for the RFID environment, different innovative designs have been proposed. The first work due to Avoine's adversarial model [4] proposed flexible definitions for different levels of privacy. Later, Juels proposed minimalist cryptography [9] in which a very restrictive adversary is defined specially for RFID. Juels and Weis commented on Avoine's model in [11] where they presented a powerful desynchronizing attack.The well known OSK protocol [12] was shown to be secure under Avoine's model [4], but later considered insecure in [11]. This has shown the need of more researches on this topic.

Very recently, Vaudenay proposed a new model [16] with eight classes of privacy levels. He concluded his paper by showing that strong privacy in RFID is impossible. Furthermore, an open questions whether forward privacy without requiring public key cryptography (PKC) is possible to be achieved was presented.

S. Jajodia, and J. Lopez (Eds.): ESORICS 2008, LNCS 5283, pp. 251–266, 2008.

Our Contributions
The contributions of this paper are threefold. First, having observed the classification of privacy presented in [16], we show that the eight privacy classes can be reduced to three privacy classes under appropriate assumptions. Second, based on our simplified classification, we show that the strongest privacy level is indeed achievable, in contrast to the result presented in [16]. This is a positive result that supports the use of RFID in practice. Third, we answer the open question in [16] by pointing out the possibility to achieve forward privacy without PKC both within Vaudenay's formal model and in practice.

2 Preliminaries

The following basic assumptions will be used throughout this paper. We note that these assumptions have been used in the existing works as well. We consider an RFID system with one reader and many tags. The reader is not corruptible and all the data stored in reader side are secure. Only the wireless link established between the reader and the involving tag during a protocol instance is insecure. Tags are not tamper-proofed. All the internal secrets stored, the memory contents written and the algorithms defined are assumed to be readily available to the adversary when a tag is corrupted. The reader will always initiate the protocol by sending out the first query message (may contain a challenge) as the tags are passive.

We briefly summarize Vaudenay's privacy model, in particular the terms that will be used frequently in the following sections. We refer the readers to [16] for the complete definition and a more complex account.

System Model. An RFID scheme is defined by two setup algorithms and the actual protocol.

- $\texttt{SetupReader}(1^s)$ is used to generate the required system parameters K_P and K_S by supplying a security parameter s. K_P denotes all the public parameters available to the environment and K_S denotes the private parameters stored inside the reader and will never be revealed to the adversary.
- $\texttt{SetupTag}^b_{K_P}(ID)$ [1] is used to generate necessary tag secrets K_{ID} and S_{ID} by inputting K_P and a custom unique ID. K_{ID} denotes the key stored inside the tag, rewritable when needed according to the protocol. S_{ID} denotes the memory states pre-set to the tag, updatable during the protocol. A bit b is also specified to indicate this newly setup tag is legitimate or not. An entry of the pair (ID, K_{ID}) will be added into the database of the reader to register this new tag when $b = 1$. Otherwise, if $b = 0$, the reader will not recognize this tag as a legitimate tag and no entry is added. Notice that K_{ID} and S_{ID} are not public and are not available to the adversary unless the tag is corrupted.
- the actual protocol used to identify/authenticate tags with the reader.

[1] This b notation was not explicitly specified originally in [16] for this algorithm, we see the need to add it to make the description more precise.

Adversarial Model. The following eight oracles are defined to represent the abilities of the adversary. We may remove and omit some details in some of the defined oracles but their main functionalities are still maintained.

- $\texttt{CreateTag}^b(ID)$ allows the creation of a free tag. The tag is further prepared by $\texttt{SetupTag}^b_{K_P}(ID)$ with b and ID passed along as inputs.
- $\texttt{DrawTag}()$ returns an ad-hoc handle $vtag$ (unique and never repeats) for one of the free tags (picked randomly). The handle can be used to refer to this same tag in any further oracles accesses until it is erased. A bit b is also returned to indicate whether the referencing tag is legitimate or not.
- $\texttt{Free}(vtag)$ simply marks the handle $vtag$ unavailable such that no further references to it are valid.
- $\texttt{Launch}()$ starts a protocol instance at the reader side and a handle π (unique and never repeats) of this instance is returned.
- $\texttt{SendReader}(m, \pi)$ sends a message m to the reader for a specific instance determined by the handle π. A reply message m' from the reader may be returned depending on the protocol.
- $\texttt{SendTag}(m, vtag)$ sends a message m to the tag determined by the handle $vtag$. A reply message m' from this tag may be returned depending on the protocol.
- $\texttt{Result}(\pi)$ returns either 1 if the protocol instance π being queried completed with success (i.e. the protocol identifies a legitimate tag) or 0 otherwise.
- $\texttt{Corrupt}(vtag)$ returns all the internal secrets K_{vtag} [2] and S_{vtag} of the tag determined by the handle $vtag$.

The interface (the environment) that provides the access to these oracles for the adversary also maintains a hidden table $\mathcal{T}$, which is not available to the adversary until the last step of the privacy experiment (to be reviewed below). When $\texttt{DrawTag}()$ is called, a new entry of the pair $(vtag, ID)$ is added into $\mathcal{T}$. When $\texttt{Free}(vtag)$ is called, the entry with the same $vtag$ handle will be marked unavailable. The true ID of the tag with handle $vtag$ is represented by $\mathcal{T}(vtag)$.

Privacy Experiment. The privacy experiment that runs on an RFID protocol is defined as a game to see whether the adversary outputs *True* or *False* after seeing the hidden table $\mathcal{T}$. At the beginning, the adversary is free to access any oracles within his allowed oracles collection (which defines different classes of adversary) according to his own attack strategy. Once the adversary finishes querying, the hidden table $\mathcal{T}$ will be released to him. The adversary will then analyze the table using the information obtained from the queries. If the adversary outputs *True*, then he wins the privacy experiment.

To measure the privacy level of an RFID protocol, a *blinder* is constructed to simulate $\texttt{Launch}()$, $\texttt{SendReader}(m, \pi)$, $\texttt{SendTag}(m, vtag)$ and $\texttt{Result}(\pi)$. If the adversary can still win with a similar probability in the above experiment

[2] Originally in [16], K_{vtag} was not included in the description. They assume that K_{vtag} is always extractable from S_{vtag}. We add K_{vtag} here to make the description clearer.

even in the present of a blinder (hence the simulations do not affect the winning probability too much), then his attack strategy is considered to be trivial. i.e. either the simulations are perfect or the attack strategy does not exploit the simulated oracles. If for all the possible attack strategies from this adversary, we can construct a blinder (possibly different) for each of them such that they are all trivial attacks, then the RFID protocol being experimented is called P-private where P is the privacy class. Let $\mathcal{A}$ be the adversary and $\mathcal{A}^{\mathcal{B}}$ be the same adversary blinded by the blinder $\mathcal{B}$, then $|\Pr[\mathcal{A} \text{ wins}] - \Pr[\mathcal{A}^{\mathcal{B}} \text{ wins}]| = \epsilon$ can be used to express the above measurement where ϵ is a negligible value.

Privacy Classes. The eight privacy classes are distinguished by different oracles collections and different natures on accessing $\texttt{Corrupt}(vtag)$ according to the strategies of the adversary.

- *Weak* : A basic privacy class where access to all the oracles are allowed except $\texttt{Corrupt}(vtag)$.
- *Forward* : It is less restrictive than *Weak* where access to $\texttt{Corrupt}(vtag)$ is allowed under the condition that when it is accessed the first time, no other types of oracle can be accessed subsequently except more $\texttt{Corrupt}(vtag)$ (can be on different handles).
- *Destructive* : It further relaxes the limitation on the adversary's strategies compares to *Forward* where there is no restriction on accessing other types of oracle after $\texttt{Corrupt}(vtag)$ under the condition that whenever $\texttt{Corrupt}(vtag)$ is accessed, such handle $vtag$ can not be used again (i.e. virtually destroyed the tag).
- *Strong* : It is even more unrestrictive than *Destructive* where the condition for accessing $\texttt{Corrupt}(vtag)$ is removed. It is the strongest defined privacy class in Vaudenay's privacy model.

Each of these privacy classes also has their *Narrow* counterparts. Namely, *Narrow-Strong*, *Narrow-Destructive*, *Narrow-Forward* and *Narrow-Weak*. These classes share the same definitions of their counterparts only there is no access to $\texttt{Result}(\pi)$.

By relaxing the limitation on the adversary's attack strategies from *Weak* to *Strong*, the adversary becomes more powerful. One can see that the privacy level is increasing from *Weak* to *Strong* if the protocol is secure against the respective class of adversary. Hence, for an RFID protocol to be *Strong*-private, it must also be *Destructive*-private. Likewise, to be *Destructive*-private, it must also be *Forward*-private, and so on. And then for a P-private protocol, it must also be *Narrow-P*-private since the *Narrow* counterparts are more restrictive. From these implications, the relations between the eight privacy classes are as follow:

$$\begin{array}{ccccccc} \textit{Strong} & \Rightarrow & \textit{Destructive} & \Rightarrow & \textit{Forward} & \Rightarrow & \textit{Weak} \\ \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow \\ \textit{Narrow-Strong} & \Rightarrow & \textit{Narrow-Destructive} & \Rightarrow & \textit{Narrow-Forward} & \Rightarrow & \textit{Narrow-Weak} \end{array}$$

3 New Privacy Classification

In this section, we firstly comment on the privacy model defined in [16]. In particular, we comment that the separation of eight privacy classes is rather excessive and unnecessary for most of the RFID protocols under proper assumptions. Then, we provide our simplified privacy model that will merge some of the privacy classes defined in [16] into a single class. The main aim of this section is to prove the following proposition:

Proposition 1. *For protocols without correlated keys and do not produce false-negative results, the eight privacy classes can be reduced to three major privacy classes if the adversary only makes "wise" oracle access.*

The "no false-negative" assumption that we will incorporate also appear in Lemma 8 of [16] where narrow-forward and narrow-weak privacy classes are reduced to forward and weak privacy classes respectively (i.e. from eight classes to six classes). The lemma assumes that any legitimate tag will *always* be identifiable, which means *no* false-negative is possible. Hence, accessing the $\texttt{Result}(\pi)$ oracle becomes redundant. As a result, the separation between *Forward* (*Weak*) and *Narrow-Forward* (*Narrow-Weak*) becomes unnecessary. We further extend this to the strong and destructive classes and consider also the false-positive case in the following proposition.

Proposition 2. *If the privacy model considers only RFID protocols that are correct and no false-negative is possible and we assume that the adversary $\mathcal{A}$ only makes "wise" oracle access whenever $\mathcal{A}$ has a non-trivial attack strategy, then the separation between narrow and non-narrow classes is unnecessary.*

The idea of proposition 2 is that if we can be sure and verify that the RFID protocol being examined will never give out false-negative, then we can examine the protocol only according to the definition of the privacy classes *Strong*, *Destructive*, *Forward* and *Weak* by assuming a "wise" adversary. This means that whether the $\texttt{Result}(\pi)$ oracle is accessed or not, it does not affect the privacy experiment results. We can remove the necessity of this oracle and reduce the eight privacy classes into four privacy classes.

Before proofing the proposition, we have to define what is "wise" oracle access and redefine what are trivial and non-trivial attacks. We also introduce perfect blinders and partial blinders.

Wise Adversary. An adversary $\mathcal{A}$ who is "wise" on oracle access will not make any oracle access that is redundant, or in other words, brings no advantage to him in attacking privacy of the protocol. Simply speaking, $\mathcal{A}$ will not waste any oracle access. More formally, let S and S' denote two different attack strategies of $\mathcal{A}$ in the privacy experiment for the same privacy class. Let q and q' be the total number of oracle accesses after executing S and S' respectively. S defines a "wiser" oracle access strategy compares to S' if and only if $\Pr[\mathcal{A}_S \text{ wins}] = \Pr[\mathcal{A}_{S'} \text{ wins}]$ and $q < q'$. Overall, a "wise" adversary can be generally defined

such that for all his attack strategies, the total numbers of oracle accesses are always minimal. Of course, such general definition of "wise" is not specific enough because q is not known before the end of attack. Specific rules are needed to keep q minimal. Consider the following as the special properties of our "wise" adversary:

- No access to the same oracle (if not probabilistic) with the same input twice.
- No access to oracles where the results can be precisely predicted.

Property 2 may be too general and should receive more justification. However, to serve our purpose in reducing the privacy classes, it is enough to focus on the $\texttt{Result}(\pi)$ oracle only, i.e. if a certain result is expected, the "wise" adversary will not access the $\texttt{Result}(\pi)$ oracle. Indeed, if the RFID protocol is *Correct*, then any legitimate or non-legitimate tag should be identified correctly, i.e. if the protocol instance π was completed for a legitimate tag, then $\texttt{Result}(\pi)$ should return 1; otherwise, 0 should be returned if it was a non-legitimate tag. This should be true as long as there are no adversarial attacks or the attacks are *insignificant*. We say that an attack is *significant* if and only if it causes the $\texttt{Result}(\pi)$ oracle to return an opposite result. This means that if there is a significant attack on a legitimate tag, then $\texttt{Result}(\pi)$ would return 0 instead of 1, and we have a *false-negative*; if there is a significant attack on a non-legitimate tag, then $\texttt{Result}(\pi)$ would return 1 instead of 0, and we have a *false-positive*. Notice that we do not need to consider incorrect identification here where a legitimate tag with ID a is identified as ID b because the $\texttt{Result}(\pi)$ oracle will only return 1 either way, making it indistinguishable by looking at the returned value only. After all, impersonation is not the goal of the privacy adversary.

Redefining Trivial and Non-Trivial Attacks. By definition, if there is a blinder $\mathcal{B}$ such that $|\Pr[\mathcal{A} \text{ wins}] - \Pr[\mathcal{A}^{\mathcal{B}} \text{ wins}]| = \epsilon$ where ϵ is a negligible value, then we say that the attack by $\mathcal{A}$ is trivial, otherwise if the value is non-negligible then the attack is non-trivial. It naturally follows that we can express this difference in the success probability of $\mathcal{A}$ under normal oracle access and simulated oracle access as the potential advantage loss of $\mathcal{A}$ because $\mathcal{A}$ has a different failure probability during the interactions with simulated oracles due to abortion of the blinder. We define this disadvantage as $\mathcal{D}^{\mathcal{B}}{}_{abort} = |\Pr[\mathcal{A} \text{ wins}] - \Pr[\mathcal{A}^{\mathcal{B}} \text{ wins}]| = |(1-\Pr[\mathcal{A} \text{ fails}]) - (1-\Pr[\mathcal{A}^{\mathcal{B}} \text{ fails}])| = |\Pr[\mathcal{A} \text{ fails}] - \Pr[\mathcal{A}^{\mathcal{B}} \text{ fails}]|$. Hence, $\mathcal{D}^{\mathcal{B}}{}_{abort}$ is the difference in the probability that $\mathcal{A}$ will fail after the introduction of $\mathcal{B}$ and if $\mathcal{D}^{\mathcal{B}}{}_{abort} = \epsilon$, then the attack by $\mathcal{A}$ is trivial; otherwise if $\mathcal{D}^{\mathcal{B}}{}_{abort} = \theta$ where θ is some non-negligible value, then the attack by $\mathcal{A}$ is non-trivial.

Perfect Blinder. A perfect blinder $\bar{\mathcal{B}}$ is a blinder that can simulate all the four blinded oracles ($\texttt{Launch}()$, $\texttt{SendReader}(m, \pi)$, $\texttt{SendTag}(m, vtag)$ and $\texttt{Result}(\pi)$) perfectly such that $\mathcal{D}^{\bar{\mathcal{B}}}{}_{abort} = \epsilon$.

Partial Blinder. Similarly, a partial blinder $\dot{\mathcal{B}}$ is a blinder that has at least one of the four blinded oracles where the simulation is not perfect. i.e. $\dot{\mathcal{B}}$ will have a chance to abort if an imperfect simulated oracle is being accessed. Notice that

we may or may not end up with $\mathcal{D}^{\dot{\mathcal{B}}}{}_{abort} = \theta$ because $\mathcal{A}$ may or may not have effectively exploited the imperfect simulated oracle(s), it depends on the attack strategy of $\mathcal{A}$.

We have the following lemma that changes a partial blinder to a perfect blinder.

Lemma 1. *A partial blinder can be viewed as a perfect blinder if and only if the adversary does not effectively exploit the imperfect simulated oracle(s).*

Proof. Let $\dot{\mathcal{B}}$ be the partial blinder where at least one of the four simulated oracles is imperfect. Let $\mathcal{O}$ denote the set of simulated oracles, then we have $\mathcal{O}_p$ be the set of perfect simulated oracles and $\mathcal{O}_p^c$ be the set of imperfect simulated oracles. $\mathcal{O}_p \cup \mathcal{O}_p^c = \mathcal{O}$ and $\mathcal{O}_p \cap \mathcal{O}_p^c = \emptyset$. Let $\mathcal{O}'$ be the set of non-simulated oracles and let E^* be the event that an abortion happens in oracle $*$. (if part) It is easy to justify that $\Pr[\mathcal{A}^{\dot{\mathcal{B}}} \text{ fails}] = \Pr[E^{\mathcal{O}}] + \Pr[E^{\mathcal{O}'}] = \Pr[E^{\mathcal{O}_p \cup \mathcal{O}_p^c}] + \Pr[E^{\mathcal{O}'}] = \Pr[E^{\mathcal{O}_p}] + \Pr[E^{\mathcal{O}_p^c}] + \Pr[E^{\mathcal{O}'}]$. Since the adversary does not effectively exploit the imperfect simulated oracle, which means $\Pr[E^{\mathcal{O}_p^c}]$ is negligible. Note we also have $\Pr[\mathcal{A}^{\bar{\mathcal{B}}} \text{ fails}] = \Pr[E^{\mathcal{O}_p}] + \Pr[E^{\mathcal{O}'}]$, which is basically $\Pr[\mathcal{A}^{\dot{\mathcal{B}}} \text{ fails}]$ - $\Pr[E^{\mathcal{O}_p^c}]$. i.e. $|\Pr[\mathcal{A}^{\bar{\mathcal{B}}} \text{ fails}] - \Pr[\mathcal{A}^{\dot{\mathcal{B}}} \text{ fails}]| = \epsilon$. (only if part) Suppose the adversary did effectively exploit the imperfect simulated oracle, then we have $\mathcal{D}^{\dot{\mathcal{B}}}{}_{abort} = \theta$, which can not be a perfect blinder for $\mathcal{D}^{\bar{\mathcal{B}}}{}_{abort} = \epsilon$ □

Corollary, we can divide the following similar lemma that changes a partial blinder of one privacy class to a perfect blinder of another privacy class using a similar proof.

Lemma 2. *A partial blinder of a stronger privacy class can be viewed as a perfect blinder of a weaker privacy class if and only if the imperfect simulated oracle is not available in the weaker privacy class.*

We do not repeat the proof here as it is very similar to the pervious proof. Clearly, not using effectively is an analogue to not available. These lemmas are general, which applies to any oracles and privacy classes. But since our goal is to show the relation between *Narrow* and *Non-narrow* classes where the $\texttt{Result}(\pi)$ oracle is available only to non-narrow classes, without loss of generality we will specifically use the $\texttt{Result}(\pi)$ oracle as an example in the following proof. We are now ready to prove the proposition.

Proof. The significance of calling the $\texttt{Result}(\pi)$ oracle is when there will be an opposite output, i.e. getting 1 when it supposes to be 0 or vice versa. This means that at least some of the attack sequences in the attack strategy have significant effect to the protocol, which makes the reader misidentify a legitimate tag as a non-legitimate one (false-negative) or a non-legitimate one as a legitimate one (false-positive). Otherwise, it would not be "wise" for the adversary to access $\texttt{Result}(\pi)$ if he did not execute any significant attacks since either 1 or 0 will be the guaranteed output for legitimate or non-legitimate tag. Indeed, the adversary always knows this fact (whether a tag is legitimate or not) when he

calls $\texttt{DrawTag}()$ to obtain a handle to a tag where a bit b is also provided to indicate the legitimacy of that tag. According to the behavior of the blinder $\mathcal{B}$ in simulating the $\texttt{Result}(\pi)$ oracle, there can be different situations:

	True oracle	Perfect simulation	Imperfect simulation
Legitimate	1	$(vtag, 1) \leftarrow \texttt{DrawTag}()$	$(vtag, 1) \leftarrow \texttt{DrawTag}()$
Non-legitimate	0	$(vtag, 0) \leftarrow \texttt{DrawTag}()$	$(vtag, 0) \leftarrow \texttt{DrawTag}()$
False-negative	N/A	N/A	N/A
False-positive	1	$1 \leftarrow \texttt{Result}^{\mathcal{B}}(\pi)$	unknown

Since we have the hypothesis that there is no false-negative, we do not need to consider it in the proof. We now have four cases to consider: i) when the attack is trivial, ii) when the attack is non-trivial and there is/are imperfect simulated oracle(s) other than $\texttt{Result}^{\mathcal{B}}(\pi)$, iii) when $\texttt{Result}^{\mathcal{B}}(\pi)$ is the only imperfect simulated oracle but $\mathcal{A}$ does not make effective use of it, and iv) when $\texttt{Result}^{\mathcal{B}}(\pi)$ is the only imperfect simulated oracle and $\mathcal{A}$ exploited it effectively.

(Case i) Consider when the attack strategies of $\mathcal{A}$ are all trivial. Then, by definition, the four oracles $\texttt{Launch}()$, $\texttt{SendReader}(m, \pi)$, $\texttt{SendTag}(m, vtag)$ and $\texttt{Result}(\pi)$ must be simulated successfully without non-negligibly affecting the success probability of the blinded $\mathcal{A}$. Since the simulation is perfect, $\mathcal{A}$ should expect no advantage gained by accessing any one of these blinded oracles in compare to when they are not blinded, i.e. we always have a perfect blinder $\bar{\mathcal{B}}$ such that $\mathcal{D}^{\bar{\mathcal{B}}}{}_{abort} = \epsilon$ where ϵ is some negligible value. Hence the RFID protocol is secure in the an non-narrow class. As the narrow counterpart is a subset of the non-narrow class, the protocol is also secure in the corresponding narrow class. As a result, protocols are both secure in narrow and non-narrow classes if the adversary's attacks are all trivial, which makes the separation unnecessary.

(Case ii) We consider when $\mathcal{A}$ has a non-trivial attack strategy. This means that there is at least one of the four blinded oracles that failed to simulate the real oracle perfectly. Suppose that it is not the $\texttt{Result}^{\mathcal{B}}(\pi)$ oracle which is/are imperfect or if $\texttt{Result}^{\mathcal{B}}(\pi)$ is imperfect, there is/are other imperfect blinded oracle(s). Since the imperfect blinded oracle(s) other than $\texttt{Result}^{\mathcal{B}}(\pi)$ is/are available to both the narrow and non-narrow classes, which means $\mathcal{A}$ can always launch non-trivial attacks through them, i.e. the RFID protocol is not secure in both classes anyway, hence the separation is unnecessary.

(Case iii) Suppose that it is now only the $\texttt{Result}^{\mathcal{B}}(\pi)$ oracle which is imperfect. Then, we have a partial blinder $\dot{\mathcal{B}}$. Assume that $\mathcal{A}$ did not make effective use of $\texttt{Result}^{\mathcal{B}}(\pi)$ during his attack; then by lemma 1, the partial blinder $\dot{\mathcal{B}}$ of the non-narrow classes can be viewed as a perfect blinder $\bar{\mathcal{B}}$ for the same privacy classes. Also by lemma 2, $\dot{\mathcal{B}}$ of the non-narrow classes is also a perfect blinder of the narrow classes since $\texttt{Result}^{\mathcal{B}}(\pi)$ is not available in the narrow classes. Since the blinder is perfect in both classes, $\mathcal{A}$'s attacks can only be trivial and the RFID protocol is secure in both non-narrow and narrow classes. Hence, even if $\texttt{Result}^{\mathcal{B}}(\pi)$ can not be simulated perfectly, there is no difference in the privacy experiments for both classes if the imperfect $\texttt{Result}^{\mathcal{B}}(\pi)$ is not exploited effectively.

(Case iv) Now for $\mathcal{A}$ to exploit the imperfect $\texttt{Result}^{\mathcal{B}}(\pi)$ effectively, $\mathcal{A}$ must cause an opposite output to happen when accessing $\texttt{Result}^{\mathcal{B}}(\pi)$. Since false-

negative is not possible as it is the hypothesis, we only need to look at false-positive, i.e. getting 1 instead of 0. False-positive happens when a non-legitimate tag is wrongly identified by the reader as a legitimate tag. Let us denote this event as E. Assume that $\mathcal{A}$ is "wise" enough not to waste any oracle accesses. When E occurs, $\mathcal{A}$ must have done some significant attacks to a non-legitimate tag or else the protocol is simply incorrect. In order to attack the tag, $\mathcal{A}$ must have obtained a handle $vtag$ to this tag, which means $\mathcal{A}$ must have called the `DrawTag`() oracle. Recall that `DrawTag`() returns $vtag$ and a bit b indicating whether $vtag$ is legitimate or not. Since $vtag$ is non-legitimate, we have $b = 0$. Recall that `DrawTag`() is not simulated by the blinder $\mathcal{B}$, $\mathcal{B}$ can also observe the returned pair $(vtag, 0)$ when `DrawTag`() is accessed by $\mathcal{A}$, hence $\mathcal{B}$ must also know $vtag$ is a non-legitimate tag. Since $\mathcal{B}$ does not know K_S, $\mathcal{B}$ has no way to tell if the reader will accept $vtag$ or not for $\mathcal{A}$ may have attacked $vtag$ at any moment, hence $\mathcal{B}$ may not be able to output the same value as the real $\texttt{Result}(\pi)$ oracle. $\mathcal{B}$ can only hope that whenever $\mathcal{A}$ accesses the $\texttt{Result}^{\mathcal{B}}(\pi)$ oracle, $\mathcal{A}$ must have already attacked $vtag$ successfully, hence $\mathcal{B}$ can be constructed to simulate $\texttt{Result}^{\mathcal{B}}(\pi)$ by returning 1 if π is the protocol instance with $vtag$ where $(vtag, 0)$ is observed when `DrawTag`() is accessed. The simulation is perfect as long as $\mathcal{A}$ performs significant attacks to $vtag$, which causes the results change from 0 to 1. The simulation will fail when $\mathcal{A}$ makes the $\texttt{Result}^{\mathcal{B}}(\pi)$ query for the protocol instance where $vtag$ is not being attacked. In that case, $\mathcal{B}$ should return 0 instead of 1. However, this should not happen because this contradicts the second property of the "wise" $\mathcal{A}$ who will not waste any oracle accesses as he knows that the reader must be able to identify a non-legitimate tag (i.e. returning 0) if it has not been attacked. Hence $\mathcal{A}$ would not have called $\texttt{Result}^{\mathcal{B}}(\pi)$ for the protocol instance with $vtag$ when $\mathcal{A}$ did not perform any significant attacks to $vtag$. At the end, $\mathcal{B}$ can simulate the oracles perfectly in front of the "wise" $\mathcal{A}$ and hence $\mathcal{D}^{\bar{\mathcal{B}}}{}_{abort} = \epsilon$, making $\mathcal{A}$'s strategy trivial, which contradicts that $\mathcal{A}$ has a non-trivial attack strategy. Hence $\mathcal{A}$ would not have let E occur, which becomes case iii. □

This proof shows that the $\texttt{Result}(\pi)$ oracle will never help the adversary if the RFID protocol being examined renders no false-negative. Furthermore, the adversary should not waste time on causing a false-positive since the attack should be on privacy and not on impersonation nor unauthorized access. In other words, from all the possible attack strategies of $\mathcal{A}$, there will be no $\texttt{Result}(\pi)$ queries if the RFID protocol being attacked does not give out false-negative. One can also extend proposition 2 to include RFID protocols where false-negative occurs with negligible if not zero probability with the same proof. Now, we have obtained the result that a P-private adversary's strategy performs as best as a *Narrow-P*-private adversary's strategy under proposition 2. Hence, we have reduced eight classes to four classes, as follows.

$$\textit{Strong} \Rightarrow \textit{Destructive} \Rightarrow \textit{Forward} \Rightarrow \textit{Weak}$$

Next, we analyze the usefulness of the destructive class. In fact, it is also mentioned in [16] that the purpose of separating the strong and destructive classes

is unclear. The destructive class is a rarely happen privacy level. Perhaps, this is the reason why there is no example provided, which is secure for this class in [16][3]. Therefore, we come up with the following proposition.

Proposition 3. *If the privacy model considers only RFID protocols that use no correlated keys among tags, then it is unnecessary to consider the destructive classes (both narrow and non-narrow).*

In other words, the destructive class is only useful to examine RFID protocols where the tags share some correlated secrets. Such type of protocols is not common in RFID. To date, we only see two constructions in [17] and [18]. The motivation behind these protocols by providing correlated key protocols is to reduce the workload and time required to lookup a matching key to verify the tag in the reader side. In most of the proposed RFID protocols under symmetric key settings [5,8,12,13], it is unavoidably to engage in an exhaustive key search process in the reader side in order to compute and match the response of any tag from all the possible keys stored inside the database. Attempts to solve this problem by providing some means to keep tags and the reader synchronized on the next expected key to be used [19,20] are found to have security loopholes [4,11]. Furthermore, Jules exploited this to attack various protocols by constructing side-channel attacks thank to the obvious different key lookup time detectable from each protocol session in [11]. A recent attempt to provide a constant lookup time [21] turns out to use a one-way trapdoor function, which is considered as one of the public key settings. Hence, there is still no efficient protocol known to solve this issue under symmetric key settings.

Additionally, correlated keys protocol under symmetric key settings can reduce the number of keys search to a logarithmic scale but with a sacrifice on strong privacy [17]: any corruption of the tag will degrade the privacy level because a tag stores not only its secret keys but also keys that share with other tags. One typical example of a correlated key protocol can be constructed using $\log_2 n$ keys for n tags. Suppose there are 8 tags in the system. One can generate only 6 keys, namely: $K_0^a, K_0^b, K_0^c, K_1^a, K_1^b, K_1^c$. Each tag is equipped with a unique set of keys, i.e. $Tag_1 \leftarrow \{K_0^a, K_0^b, K_0^c\}$, $Tag_2 \leftarrow \{K_0^a, K_0^b, K_1^c\}$, ... $Tag_n \leftarrow \{K_1^a, K_1^b, K_1^c\}$. It is easy to verify that these tags can be uniquely identified by checking at most 6 instead of 8 keys as in the independent key protocols by the reader. However, corrupting any one of these tags provides the adversary with a full potential to distinguish each of these tags responses. Damgård [18] provided a result on the tradeoff between the number of correlated keys and the number of corrupted tags as:

$$\frac{ctu}{v} + \frac{ctu}{v-u}$$

where c is the number of keys stored in each tag, v is the number of different keys per column, t is the number of tags queried by the adversary and u is the

[3] Notice that the example provided in [16] for the narrow-destructive class that use independent keys is not different from a protocol for the narrow-forward class, while the example given that uses dependent keys is insecure in the narrow-destructive class.

number of corrupted tag. As long as the result of the formula is negligible, the protocol is secure. In our example given above, $c = 3, v = 2$ and $u = 1$, hence we have $\frac{3t}{2} + 3t \leq \epsilon$, which means $t < 1$, i.e. the protocol is secure only if the adversary does not query any tag at all, or simply the protocol is never secure against tag corruptions.

From the above discussion, it is clear that correlated key protocols are extremely weak against tag corruption. One can only expect the protocol to be secure if $t, u << v << n << v^c$. In the model of [16], since there is no limitation on either t or u, there can be no correlated key protocols that is secure in both strong and destructive privacy classes. The proof of proposition 3 follows.

Proof. Recall the destructive class definition, after calling `Corrupt`($vtag$), the same tag handle $vtag$ is not allowed to be used anymore. It is clear to see from the definition that this destructive corruption cannot provide the adversary any additional advantage in winning the privacy game if each of the tags is independent to each other. In order for the `Corrupt`($vtag$) oracle to become significant under the destructive class definition, the corrupted internal secrets K_{vtag} and S_{vtag} have to be useful in some following oracle accesses (if it is useful to the results obtained from some pervious oracles, then we have gone backward to the forward class). Since the corrupted tag of handle $vtag$ cannot be accessed again, the secrets must only be used on some other tags. If the tags are independent to each other, K_{vtag} and S_{vtag} would have revealed no information about any other tags. As there is no effect on other tags, the simulation of the blinder can be easily constructed, making the adversary's strategy trivial and hence the attack is insignificant. □

Combining the above results, the destructive class is rather not very meaningful. It is only useful to examine protocols that use correlated keys while these protocols can never achieve strong and destructive privacy classes under the model in [16]. Together with proposition 2, we have successfully reduced the eight privacy classes into three major classes, as follows.

$$\mathit{Strong} \Rightarrow \mathit{Forward} \Rightarrow \mathit{Weak}$$

Our result simplifies the previous privacy classification due to Vaudenay [16]. Furthermore, in contrast to Vaudenay's result, we shall show that strong privacy is indeed possible, and hence this result will indeed make RFID protocols more useful in its real applications.

4 New Results

In this section, we will present our new results in privacy model in RFID. In particular, we shall show that strong privacy is indeed possible (cf. [16]) and we shall present our affirmative answer to the open problem posed in [16] in regards

to the construction of RFID scheme with forward privacy without requiring the public key cryptography (PKC).

4.1 Strong Privacy Is Possible

One of the results in [16] is that strong privacy is impossible. This is supported by a theorem that *a Destructive-private RFID protocol is not Narrow-Strong-private*. Since *Strong*-private (**S**) implies both *Destructive*-private (**D**) and *Narrow-Strong*-private (**NS**) by definition[4], we have $\mathbf{S} \subseteq \mathbf{D}$ and $\mathbf{S} \subseteq \mathbf{NS}$. i.e. $\exists p \in \mathbf{S}$ s.t. $p \in \mathbf{D}$ and $p \in \mathbf{NS}$ where p is an RFID protocol. However, we would like to use our results in Section 3 to show that strong privacy is actually possible. We consider the same example of PKC-based RFID protocol provided in Section 4.3 of [16], which is *Narrow-Strong*-private. By applying proposition 2, we show that it is also *Strong*-private.

We look at the following example PKC protocol where $\texttt{Enc}()$ is IND-CPA secure and (K_P, K_S) is the public and private keys pair. For completeness, we present the protocol below.

Tag$\{K_P, ID, K_{ID}\}$		Reader$\{K_S, K_M\}$
	$\xleftarrow{\quad a \quad}$	pick $a \in \{0,1\}^s$ randomly
$c = \texttt{Enc}_{K_P}(K_{ID}\|\|ID\|\|a)$	$\xrightarrow{\quad c \quad}$	$\texttt{Dec}_{K_S}(c) = K_{ID}\|\|ID\|\|a'$
		if $a' = a$, verifies $K_{ID} = F_{K_M}(ID)$

To apply proposition 2, we have to observe whether false-negative could be generated. Since c is the only message received by the reader, a false-negative can only happen if c is malicious (i.e. ID and K_{ID} are replaced), or c happens to be the same encrypted value c' where $c' = \texttt{Enc}_{K_P}(ID'||K_{ID'}||a)$. The former is safe guarded by the IND-CPA secure property of the PKC algorithm, which states that it is infeasible for any computationally bounded adversary to retrieve the private key by looking at the ciphertexts of arbitrarily chosen plaintexts only. That means, the only possible option is to guess the private key, which happens with negligible probability. The latter will not happen as decryption is unique, otherwise both c and c' will be decrypted to a same value. Therefore, we can apply proposition 2 and the PKC protocol is also *Strong*-private if it is *Narrow-Strong*-private. This gives us the result $\mathbf{S} = \mathbf{NS}$. Since $\mathbf{S} \subseteq \mathbf{D}$, we also have $\mathbf{NS} \subseteq \mathbf{D}$. Together with the theorem in [16], we conclude with $\mathbf{NS} \subset \mathbf{D}$.

4.2 Truly Random Source Is Required

Let us observe the PKC protocol above again. The protocol assumes that the underlying encryption algorithm is IND-CPA. Due to the randomness of the IND-CPA property, which is needed to provide indistinguishability, c is different every

[4] This is easy to verify. As *Narrow-Strong* is *Strong* without the $\texttt{Result}(\pi)$ oracle access. *Destructive* is *Strong* with additional limitation on accessing the $\texttt{Corrupt}(vtag)$ oracle. Both are more restrictive (i.e. the adversary is less powerful) than *Strong*. A protocol secure in *Strong* must also be secure in *Destructive* and *Narrow-Strong*.

time even if the same a is received by the same tag, i.e. $c = \mathtt{Enc}_{K_P}(K_{ID}||ID||a)$ in protocol instance π is not equal to $\tilde{c} = \mathtt{Enc}_{K_P}(K_{ID}||ID||a)$ in another protocol instance $\tilde{\pi}$. This randomness is implicitly included in the IND-CPA assumption. We can change the notation a little bit to reveal this hidden randomness. We rewrite the PKC protocol as follows.

Tag$\{K_P, ID, K_{ID}\}$		Reader$\{K_S, K_M\}$
pick $r \in \{0,1\}^s$ randomly	$\xleftarrow{\quad a \quad}$	pick $a \in \{0,1\}^s$ randomly
$c = \mathtt{Enc}_{K_P}(K_{ID}\|\|ID\|\|a\|\|r)$	$\xrightarrow{\quad c \quad}$	$\mathtt{Dec}_{K_S}(c) = K_{ID}\|\|ID\|\|a'\|\|r$
		if $a' = a$, verifies $K_{ID} = F_{K_M}(ID)$

In fact, even under the IND-CPA assumption, the tag still needs to pick a random value r for every encryption (e.g. using the ElGamal scheme). This is just abstracted in [16]. With the new notation, we can now consider the following question: *If a tag is corrupted, will the future random values be revealed as well?* If PRNG is implemented in the tag to generate random values, the answer to this question should be 'yes'. It is easy to see that if the PRNG algorithm is revealed after corrupting the tag, the adversary can easily trace the tag by computing $r = \mathtt{PRNG}(S)$ (S is the memory state of the tag) and then verifies that if $c = \mathtt{Enc}_{K_P}(K_{ID}||ID||a||r)$ where K_{ID}, ID, S_{ID}, and $\mathtt{PRNG}()$ are all revealed after tag corruption. Since c is unique, the adversary must be able to trace the tag. However, if the tag has a truly random source (e.g. another module attached to the tag), this can be modeled as a random oracle and the answer should be 'no'. We conclude that a truly random source (under the random oracle model) is required for the PKC protocol to be *Strong*-private, which was missing in the definition provided in [16].

4.3 Forward Privacy without PKC

Consider a variant of the OSK protocol [12] that appeared in [16] as follows.

Tag$\{K_{ID}\}$		Reader$\{(ID_1, K_1), (ID_2, K_2), ..., (ID_n, K_n)\}$
	$\xleftarrow{\quad a \quad}$	pick $a \in \{0,1\}^s$ randomly
$c = F(K_{ID}, a)$	$\xrightarrow{\quad c \quad}$	for $j \in \{1, n\}$ and $i \in \{0, t-1\}$
set $K_{ID} = G(K_{ID})$		find (ID_j, K_j) s.t. $c = F(G^i(K_j), a)$
		set $K_j = G^{i+1}(K_j)$

This protocol is proven to be *Narrow-Destructive*-private in [16]. Recall that *Narrow-Destructive* $\Rightarrow$ *Narrow-Forward*, this protocol is also *Narrow-Forward*-private. We note that our proposition 2 *cannot* be applied to this protocol because false-negative can happen when a legitimate tag has been queried for t times by an adversary before it is queried by the reader again. Since K_{ID} would become $G^t(K_{ID})$ by then and the reader will only try $0 \leq i \leq t-1$ different $G^i(K_j)$ values per (ID_j, K_j) pair to find a matching $F(G^i(K_j), a)$ for c, that legitimate tag will not be identified successfully by the reader, hence a false-negative occurs. In

other words, calling $\texttt{Result}(\pi)$ in this case helps the adversary to gain advantage in winning the privacy experiment, which causes this protocol to be *Narrow-Forward*-private only but not *Forward*-private. Hence, leaving the question *"whether Forward-private without PKC is possible"* open.

Here, we would like to apply proposition 2, so that *Forward* is not different from *Narrow-Forward*, and the OSK variant protocol will become also *Forward*-private, and hence, it will answer the open problem. First of all, we notice that the reason why there can be false-negative is due to $i \leq t-1$[5]. Next, we consider the number of queries to a tag the adversary can make be q and we assume that $q \leq t$. In other words, the adversary can never query any particular tag for more than t times and the reader is now always able to identify any legitimate tag, which also means there will not be any false-negative. This implies that proposition 2 can be applied and we have the OSK variant protocol become *Forward*-private.

The only thing that is arguable is whether the assumption ($q \leq t$) makes any sense or not. Clearly, one can also argue that when $q > t$, then the privacy will not be satisfied any longer. Hence, the problem has turned to a a scalability issue: *"Can we always have a more resourceful reader compared to an adversary?"* In fact, the ability of an adversary can be limited by different means in reality. Limited tag queries due to the mobility of tags and throttling [9] are some realistic examples to support the assumption. In particular, for the low-cost RFID tag environment, it is more appropriate to consider a less almighty adversary model. Furthermore, seeking strong privacy in front of a powerful adversary for RFID that is known by its limited resources characteristic seems to be impractical.

5 Conclusion

In this paper, we examined the RFID privacy model provided in [16] in a great detail and presented some new results. Firstly, we examined the eight different classes presented in [16] and applied some reasonable assumptions to simplify the classification. Then, we presented a counter argument to [16] by stating that strong privacy in RFID is indeed achievable. In summary, to achieve strong privacy, tags are required to perform not only public key cryptography, but also require an additional reliable random source, which was missing from the description provided in [16]. Nonetheless, this results in a high manufacturing cost for RFID tags. However, in contrast to Vaudenay's result, we have shown that strong privacy is indeed achievable. Furthermore, we believe that in the future development of RFID, privacy will have to be sacrificed to keep the cost low. Hence, it is worthwhile to reconsider whether RFID should face such a strong adversary model. Due to the short communication range and infrequent access properties of RFID tags, we believe it is not necessary to assume the presence of

[5] In the original OSK paper [12], this limitation does not exist in the protocol description, which is why Avoine showed that this protocol is secure in his paper [4], but later on Juels and Weis disagreed in [11] when this limitation was considered.

powerful adversaries. Henceforth, an adequate and appropriate privacy model, which takes into account the constraints of RFID is still missing.

References

1. Juels, A.: RFID Security and Privacy: A Research Survey. IEEE Journal on Selected Areas in Communications 24(2), 381–394 (2006)
2. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, 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. 201–212. Springer, Heidelberg (2004)
3. Avoine, G.: Privacy Issues in RFID Banknote Protection Schemes. In: CARDIS, pp. 34–38. Kluwer, Dordrecht (2004)
4. Avoine, G.: Adversarial Model for Radio Frequency Identification (2005), `http://citeseer.ist.psu.edu/729798.html`
5. Avoine, G., Oechslin, P.: A Scalable and Provably Secure Hash-Based RFID Protocol. In: PerSec, pp. 110–114. IEEE Computer Society Press, Los Alamitos (2005)
6. Avoine, G., Oechslin, P.: RFID Traceability: A Multilayer Problem. In: S. Patrick, A., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 125–140. Springer, Heidelberg (2005)
7. Burmester, M., de Medeiros, B.: RFID Security: Attacks, Countermeasures and Challenges. In: The 5th RFID Academic Convocation, The RFID Journal Conference (2007)
8. Burmester, M., van Le, T., de Medeiros, B.: Provably Secure Ubiquitous Systems: Universally Composable RFID Authentication Protocols. In: SecureComm. (2006)
9. Juels, A.: Minimalist Cryptography for Low-Cost RFID Tags. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 149–164. Springer, Heidelberg (2005)
10. Juels, A., Molnar, D., Wagner, D.: Security and Privacy Issues in E-passports. In: SecureComm. (2005)
11. Juels, A., Weis, S.A.: Defining Strong Privacy for RFID (2006), `http://citeseer.ist.psu.edu/741336.html`
12. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Approach to "Privacy-Friendly" Tags. In: RFID Privacy Workshop (2003)
13. Ohkubo, M., Suzuki, K., Kinoshita, S.: Efficient hash-chain based RFID privacy protection scheme. In: UbiComp Workshop, Ubicomp Privacy: Current Status and Future Directions (2004)
14. Ohkubo, M., Suzuki, K., Kinoshita, S.: Hash-Chain Based Forward-Secure Privacy Protection Scheme for Low-Cost RFID. In: SCIS (2004)
15. Ohkubo, M., Suzuki, K., Kinoshita, S.: RFID Privacy Issues and Technical Challenges. Communications of the ACM 48(9), 66–71 (2005)
16. Vaudenay, S.: On Privacy Models for RFID. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)
17. Molnar, D., Soppera, A., Wagner, D.: A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 276–290. Springer, Heidelberg (2006)
18. Damgård, I., Pedersen, M.Ø.: RFID Security: Tradeoffs between Security and Efficiency. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 318–332. Springer, Heidelberg (2008)

19. 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)
20. Henrici, D., Muller, P.: Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers. In: PerSec, pp. 149–153. IEEE Computer Society Press, Los Alamitos (2004)
21. Burmester, M., de Medeiros, B., Motta, R.: Robust, Anonymous RFID Authentication with Constant Key-Lookup. In: ASIACCS, pp. 283–291. ACM, New York (2008)