New Privacy Results on Synchronized RFID Authentication Protocols against Tag Tracing

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

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

Abstract. Many RFID authentication protocols with randomized tag response have been proposed to avoid simple tag tracing. These protocols are symmetric in common due to the lack of computational power to perform expensive asymmetric cryptography calculations in low-cost tags. Protocols with constantly changing tag key have also been proposed to avoid more advanced tag tracing attacks. With both the symmetric and constant-changing properties, tag and reader re-synchronization is unavoidable as the key of a tag can be made desynchronized with the reader due to offline attacks or incomplete protocol runs. In this paper, our contribution is to classify these synchronized RFID authentication protocols into different types and then examine their highest achievable levels of privacy protections using the privacy model proposed by Vaudenay in Asiacrypt 2007 and later extended by Ng et al. in ESORICS 2008. Our new privacy results show the separation between *weak privacy* and *narrow-forward privacy* in these protocols, which effectively fills the missing relationship of these two privacy levels in Vaudenay's paper and answer the question raised by Paise and Vaudenay in ASIACCS 2008 on why they cannot find a candidate protocol that can achieve both privacy levels at the same time. We also show that *forward privacy* is impossible with these synchronized protocols.

1 Tag Tracing Problem

Since the design of RFID authentication protocols, tag tracing has been one of the major privacy concerns. Passive RFID tags, without their own power sources, are designed to respond to every reader query in nature when the query signal powers them up for authentication purpose. Each tag response is unique in order to avoid misidentification. A reader that picks up these responses can identify each tag and authenticate legitimate ones by matching the known information about these tags from a back-end database. Adversaries with compatible readers can take advantage of this response-to-all property to attack tag privacy. It is not hard to imagine how these unique-per-tag responses can aid adversaries in tracing or locating any specific tag. This tag tracing behavior violates the location privacy of RFID tag bearers. A pessimistic way to deal with tag tracing is to "kill" the tag with some

M. Backes and P. Ning (Eds.): ESORICS 2009, LNCS 5789, pp. 321–336, 2009.

deactivation commands [1,30]. However, this will only sacrifice the benefits and convenience of using RFID to provide potential services in the future [27]. Other methods like the use of signal blocking devices [10] does more harm than good. Consider the use of RFID to collect auto-toll payments or shoplifting preventions, misbehaving users can easily sabotage the underlying RFID system. To keep RFID tags "alive" and to protect them from being traced at the same time, it is essential to guarantee *untraceability* in RFID protocols[1].

Researchers have devoted a lot of efforts to design secure RFID authentication protocols that are untraceable, although a promising candidate is still yet to be seen. There are some RFID protocols that guarantee untraceability in a strong privacy sense [35,23,29], but these protocols require Public Key Cryptography (PKC). These asymmetric cryptography calculations are commonly agreed to be too expensive to implement and not suitable for RFID tags due to the low cost and low computational power natures of RFID. To the best of our knowledge, there *does not* exist a single RFID protocol in the symmetric key setting that provides untraceability to a satisfactory level. This leads us to believe there exists limitations in this type of RFID protocols on providing untraceability in any stronger privacy senses.

Related works

We do not create any new RFID authentication protocol in this paper. Instead, we are the first to provide classification for synchronized RFID authentication protocols based on their construction methods and prove their limitations against tag tracing. We cited more than thirty recently proposed protocols into our classifications. We use the privacy model created by Vaudenay in [35] where eight levels of privacy: *Weak privacy*, *Forward privacy*, *Destructive privacy*, *Strong privacy* and their *Narrow* counterparts are defined (we will review these privacies in section 3). Examples of symmetric key RFID authentication protocols that can achieve *Weak privacy*, *Narrow-weak privacy* and *Narrow-forward privacy* are provided in [35] while a question on achieving *Forward privacy* without PKC is left open. Paise and Vaudenay used the same privacy model of [35] and extended the results to mutual RFID authentication protocols in [29]. They also left an open question asking whether it is feasible to achieve both *Weak privacy* and *Narrow-forward privacy* at the same time using symmetric key protocols only. Later on, Ng et al. reduced the eight levels of privacy in the Vaudenay model into three main levels by introducing two useful lemmas in [23]. We use their results to reduce the complexity of this paper in analyzing the achievable privacy levels of synchronized RFID authentication protocols.

Our Contributions

In this paper, we have the following contributions. First, we look into the general constructions of symmetric key RFID authentication protocols. Both

[1] We only focus on the protocol level in this paper. Avoine studied the tag tracing problem even in the physical level [4], where RFID tags may emit distinguishable unique radio signals that allows simple tracing by anyone due to hardware manufacturing diversities. This will render all the protocol level protections useless.

tag-to-reader and mutual (i.e. tag and reader) authentication protocols are examined. Second, we deduce that all of these protocols unavoidably require *tag key update* in the tag side and *tag key synchronization* between tag and reader at some point of the protocol in order to provide better untraceability against stronger attacks. Third, we classify these protocols into four main construction types based on when the *tag key update* and *tag key synchronization* operations are carried out. Fourth, we adopt the privacy model proposed by Vaudenay in [35] and a modified one in [23] to prove the highest privacy levels that can be attained in these protocols for each construction type. We do this by combining the results of [35] and [29] and constructing an universal generic attack for each construction type targeting a higher privacy level. Notice that our attacks are purely taking advantages of the adversary model defined in [35] but not exploiting various flaws in protocol designs. Fifth, according to our results, we can show the separation between *Weak privacy* and *Narrow-forward privacy* in these protocols, which was not shown in [35]. Lastly, we answer the open questions left by Vaudenay in [35] and by Paise and Vaudenay in [29] on the feasibility to achieve *Forward privacy* without PKC and on the possibility to achieve both *Weak* and *Narrow-forward privacies* at the same time using only symmetric key protocols.

2 RFID System Model

Throughout this paper, we will use the following definitions and assumptions for our RFID system. We note that these assumptions are commonly used in existing works and hence, they reflect a common RFID environment in privacy evaluation.

2.1 Basic Assumptions

We consider an RFID system with a back-end database, a reader and more than one tag. Only the legitimate reader can access the database. Tags that have registered in the database are legitimate and only then they can be identified and authenticated by the legitimate reader. A correct authentication protocol should allow only the legitimate reader (with access right to the database) to be able to identify these tags. During the protocol's execution, an appropriate and secure singulation mechanism is always assumed to be available such that only a single tag will be involved in the communication with the reader in each communication instance. The reader can always retrieve necessary data from the database whenever it is required. The link that connects the reader and the database is assumed to be secure and always reliable and available. Hence it is common to consider the reader and the database as a single entity. The reader is not corruptible either, which means all the data stored in the reader side (i.e. inside the database) are secure. Only the wireless messages exchange between the reader and tag during a protocol instance are free to be intercepted, tampered and replayed, etc. Tags can be corrupted easily and are not tamper-proofed.

Once corrupted, all the stored internal secrets, memory contents and algorithms defined are assumed to be readily available to the adversary. Reader will always initiate a protocol instance by sending out the first query message (which may or may not contain a challenge) because tags are passive entities.

2.2 RFID Protocol

An RFID protocol is defined by two setup algorithms and a message exchange sequence.

- $\texttt{SetupReader}(1^s)$ is used to generate the required system parameters P by supplying a security parameter s. P denotes all the public parameters available to the environment (tags, reader and adversary).
- $\texttt{SetupTag}^b_P(ID)$ is used to generate necessary tag secret K_{ID} by inputting P and a custom unique ID. K_{ID} denotes the key stored inside the tag, rewritable when needed according to the protocol. A bit b is also specified to indicate this newly setup tag is legitimate or not. If $b = 1$, an entry (ID, K_{ID}) will be added into the database to register the tag and the tag becomes legitimate. Otherwise, no entry is added and the tag will not be authenticated by the reader in later protocol instances. Notice that K_{ID} will become available to the adversary when the tag is corrupted.
- a message exchange sequence is implemented in tags and reader governing the authentication process.

3 RFID Privacy Model

Our privacy model is based on the Vaudenay privacy model defined in [35]. We briefly summarize the privacy model below, in particular the terms that will be used frequently in the coming sections.

3.1 Adversary Oracles

The following eight oracles are defined to represent the abilities of adversaries.

- $\texttt{CreateTag}^b(ID)$ allows the creation of a free tag. The tag is further prepared by $\texttt{SetupTag}^b_P(ID)$ 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 together with the initial messages m broadcasted by the reader.

- `SendReader`(π, m) sends a message m to the reader for a specific instance determined by the handle π. A message m' from the reader may be returned depending on the protocol.
- `SendTag`$(vtag, m)$ sends a message m to a tag determined by the handle $vtag$. A message m' from this tag may be returned depending on the protocol.
- `Result`(π) returns either 1 if the protocol instance π completed with success (i.e. the protocol identifies a legitimate tag) or 0 otherwise.
- `Corrupt`$(vtag)$ returns the internal secret K_{vtag} of the tag $vtag$.

3.2 Privacy Levels

The eight privacy levels are distinguished by their different natures on accessing `Corrupt`$(vtag)$ in the strategies of the adversary and whether `Result`(π) is accessed or not.

- *Weak* : The most basic privacy level where access to all the oracles are allowed except `Corrupt`$(vtag)$.
- *Forward* : It is less restrictive than *Weak* where access to `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 `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 `Corrupt`$(vtag)$ under the condition that whenever `Corrupt`$(vtag)$ is accessed, such handle $vtag$ cannot be used again (i.e. virtually destroyed the tag).
- *Strong* : It is even more unrestrictive than *Destructive* where the condition for accessing `Corrupt`$(vtag)$ is removed. It is the strongest defined privacy level in the Vaudenay privacy model.

Each of these privacy levels also has its *Narrow* counterpart. Namely, *Narrow-Strong*, *Narrow-Destructive*, *Narrow-Forward* and *Narrow-Weak*. These levels share the same definitions of their counterparts, only there is no access to `Result`(π).

By relaxing the limitation on the adversary's attack strategies from *Weak* to *Strong*, the adversary becomes more powerful, hence the privacy level is increasing from *Weak* to *Strong*. This implies that 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. Similarly, for an L-private protocol, it must also be *Narrow-L*-private since the *Narrow* counterparts are more restrictive. From these implications, the relations between the eight privacy levels are as follow:

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

3.3 Privacy Experiment

The setup of privacy experiment requires a hidden table $\mathcal{T}$ to be maintained whenever the oracles `DrawTag`() and `Free`($vtag$) are called. This hidden table is not available to the adversary until the last step of the privacy experiment (to be reviewed below). When `DrawTag`() is called, a new entry of the pair $(vtag, ID)$ is to be added into $\mathcal{T}$. When `Free`($vtag$) is called, the entry with the same $vtag$ handle is to be marked unavailable. The true ID of the tag with handle $vtag$ is represented by $\mathcal{T}(vtag)$.

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 oracle collection according to his own attack strategy (which defines the maximum targeting privacy level to attack). Once the adversary finishes querying, the hidden table $\mathcal{T}$ will be released to him. The adversary will then analyze the $(vtag, ID)$ entries in the table using the information obtained before from the queries. If the adversary finally outputs *True* for the question whether $\mathcal{T}(vtag) = ID$ in a non-trivial sense (i.e. not blindly outputs *True* because $\mathcal{T}(vtag) = ID$ as listed in the table), then he has successfully traced a victim tag of identity ID and won the privacy experiment. We say that the RFID protocol being experimented is not L-private where L is the highest privacy level achievable from the oracle collection of the adversary.

4 New Privacy Results of Symmetric Key RFID Protocols

We look at different constructions of RFID authentication protocols (both tag-to-reader and mutual) under the symmetric key setting with or without *tag key update* and *tag key synchronization*. We show the limitation of each of the constructions on achieving a certain privacy level in tag tracing.

4.1 Protocol Constructions

Before we define our protocol construction classifications, we have these notations:

- $\mathcal{O}^{Tag}(), \mathcal{O}^{Reader}()$: A collection of operations denoted as an oracle following the protocol specification carried out on the tag and reader sides respectively.
- K_{ID}^{i} : The tag key at instance i where the initial key is K_{ID}^{0}.
- S_{ID}^{i} : The tag state at instance i denoted as an encapsulation of the tag key K_{ID}^{i} and other per instance generated and received values. If S_{ID}^{i} is updated to S_{ID}^{i+1}, K_{ID}^{i} is updated to K_{ID}^{i+1} as well.
- $\mathcal{O}^{Update}(S_{ID}^{i})$: A tag key update oracle performed on the tag side which takes S_{ID}^{i} as input and outputs an updated K_{ID}^{i+1}.

– $\mathcal{O}^{Sync}(S^i_{ID})$: A tag key synchronization oracle performed on the reader side which takes S^i_{ID} as input and outputs a synchronized K^d_{ID}. It is a recursive function which has an upper bound n where $n + i \geq d > i$ or $d = i - 1$. The upper bound is added to reflect the side-channel attack effect described in [11].

It is important for us to state that we are not concerned about how RFID authentication protocols are implemented. Some may use simple bitwise operations like XOR, some may use hashing functions, some may even use symmetric encryption/decryption. We only classify them based on how and when $\mathcal{O}^{Update}(S^i_{ID})$ is executed. For an RFID authentication protocol to fall into one of the following construction types, the bottom line is that the protocol has to be at least correct (i.e. when the protocol is started with $\pi \leftarrow \texttt{Launch}()$, then by calling $\texttt{Result}(\pi)$, it should output 1, with overwhelming probability, for legitimate tags and 0 otherwise). Protocols that fail this basic requirement should not be defined as authentication protocol at all. We classify RFID authentication protocols into the following four construction types:

– **Type 0** : Protocols that are correct and lack tag key update mechanisms or equivalently even with $\mathcal{O}^{Update}(S^i_{ID})$ implemented it can not be executed properly as if it is not there, which causes K^i_{ID} remains static at the end of the protocol [2].
– **Type 1** : Protocols that are correct and $\mathcal{O}^{Update}(S^i_{ID})$ can be executed properly, which causes K^i_{ID} to change every time the protocol is executed.
– **Type 2a** : Mutual authentication protocols that are correct and $\mathcal{O}^{Update}(S^i_{ID})$ is executed properly *after* the final reader authentication message is received, which causes K^i_{ID} to change after the reader is authenticated.
– **Type 2b** : Mutual authentication protocols that are correct and $\mathcal{O}^{Update}(S^i_{ID})$ is executed properly *before* the final reader authentication message is received, which causes K^i_{ID} to change before the reader is authenticated.

4.2 Achievable Privacy Levels

As pointed out in [35] and [23], (narrow-)strong privacy for tag authentication protocols is only achievable with PKC under the asymmetric key setting. The same result is supported by [29] for mutual authentication protocols. From the results we obtained, which will be presented below, we also agree to this impossibility result for RFID protocols under symmetric key setting. Hence, this will leave us with these six privacy levels:

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

[2] Some protocols, for example the YA-TRAP [33], although they have some tag key update mechanisms, they are known to have design flaws that effectively render their key update mechanisms useless (i.e. as if the tag key is never updated), we do not classify these protocols to have tag key update. Readers can refer to [2,11,34] for more specific attacks on existing protocols based on their design flaws.

It has also been proved in [23] that the destructive levels are only distinguishable from the forward levels as long as the RFID protocols share correlated secrets (e.g. global key, partial group key, etc.) among tags. Corrupting one tag in these protocols will also reveal (partial) secrets of related tags. The majority of RFID protocols do not belong to this special protocol category. Hence we will only focus on RFID protocols where each tag is independent from each other and does not store any correlated secrets. This leaves us with four main privacy levels to be examined in the rest of the paper:

$$\begin{array}{ccc} \textit{Forward} & \Rightarrow & \textit{Weak} \\ \Downarrow & & \Downarrow \\ \textit{Narrow-Forward} & \Rightarrow & \textit{Narrow-Weak} \end{array}$$

We can now formally analyze the four symmetric RFID protocol construction types. For each of them, we will prove the impossibility for it to achieve a certain privacy level with an universal attack. It is important to note that these attacks are *generic* and *universal* as they are only constructed using the oracles defined in section 3. We do not need to exploit any design flaw in the protocols in order to make the attacks success. Hence the attacks are valid as long as the same adversary model is applied.

Also, as our results are about the highest achievable privacy levels, not the lowest, there can be some protocols of the same construction type that only achieve a weaker privacy level. For protocols that do not provide privacy protection at all, we represent them with a special class *Nil*. Since we are not claiming the lowest achievable privacy level for the protocols, we do not consider the separation between any weaker privacy levels weaker than *Weak privacy* as defined in [35] and just group them all into the special class *Nil*.

For each of the construction types, we abstract the common form of that type of protocols in a figure for illustration purpose. There can be variations on how the reader verifies legitimate tags responses and how the messages flow. But what in common is whether there is tag key update or not and if there is, when is it executed? Again, our universal attacks do not concern the implementation details of these protocols, hence they are universal.

4.3 Type 0 Protocols Can Never Achieve Forward Privacy Levels

Construction. Type 0 represents the most basic form of an RFID authentication protocol that uses symmetric key without tag key update. Protocols in [5,31,13,14,19,21,20,22,36,33] are some examples. It should be trivial for most readers that forward privacy is impossible in this type of construction, since tag corruption will reveal the static tag key. It still serves as a base in our classifications because we will reduce some other construction types to this type in the following sections. Here we look at the common construction of this type of protocols.

Tag$\{K_{ID}\}$		Reader$\{ID, K_{ID}\}$
v: random value	$\xleftarrow{\quad Query, c \quad}$	*c: random challenge*
$S_{ID} : \{K_{ID}, c, v\}$		
$Response \leftarrow \mathcal{O}^{Tag}(S_{ID})$	$\xrightarrow{\quad Response \quad}$	*r: Response*
		$\forall i \in \{ID\}, S_i : \{K_i, r, c\}$
		Verify if $r = \tilde{r} \leftarrow \mathcal{O}^{Reader}(S_i)$
		if $FOUND$, *set* `Result`$(.) = 1$
		else set `Result`$(.) = 0$

Since there is no $\mathcal{O}^{Update}(S^i_{ID})$, both tag and reader keep the same K_{ID} value through out the life time of the tag. Without tag key update, protocols with this construction can never achieve forward privacy and narrow-forward privacy. Because forward privacy is harder than narrow-forward privacy, we only need to show that narrow-forward privacy is not achievable. Consider the following attack:

1. `CreateTag`$^1(ID_0)$, `CreateTag`$^1(ID_1)$
2. $vtag \leftarrow$ `DrawTag`()
3. $\pi \leftarrow$ `Launch`()
4. $c \leftarrow$ `SendReader`$(\pi, Init)$
5. $r : Response \leftarrow$ `SendTag`$(vtag, c)$
6. (Forward r to reader to close π) $null \leftarrow$ `SendReader`(π, r)
7. `Free`$(vtag)$
8. $vtag' \leftarrow$ `DrawTag`()
9. $K_{ID_x} \leftarrow$ `Corrupt`$(vtag')$
10. Queries ended, receive $\mathcal{T}(vtag) = ID_b$
11. Let $S_{ID_x} : \{K_{ID_x}, r, c\}$, if $r = \tilde{r} \leftarrow \mathcal{O}^{Reader}(S_{ID_x})$ then $x = b$. Otherwise $x = |1 - b|$
12. Output whether $\mathcal{T}(vtag') = ID_x$

The idea of the attack is to record a protocol instance between a legitimate tag and a reader. A random tag is then corrupted and its tag key is exposed. By simulating a protocol run using the exposed tag key, if the result is the same as the recorded one, then the same tag is found with high confident. An adversary running the attack above will only fail (i.e. $\mathcal{T}(vtag') \neq ID_x$) if $\mathcal{O}^{Reader}(S_{ID_0}) = \mathcal{O}^{Reader}(S_{ID_1})$. This should only happen with a negligible probability, otherwise the protocol is simply incorrect, which produces wrong identification. Hence the adversary will succeed with overwhelming probability. Since there is no further oracle access after `Corrupt`$(vtag')$ and no `Result`(π) in the attack, this is a significant narrow-forward privacy level attack. We have shown that RFID protocols without tag key update is not narrow-forward private and hence not forward private.

Remark 1. A **Type 0** construction RFID protocol presented in [35] using pseudorandom function (PRF) has been proved to provide weak privacy. Hence it is the highest privacy level that can be attained by RFID protocols with **Type 0** construction. Our conclusion is summarized as follows.

Type 0	*Forward levels*	*Weak levels*	*Nil*
Non-narrow levels	-	✓	✓
Narrow levels	-	✓	

4.4 Type 1 Protocols Can Never Achieve Non-narrow Privacy Levels

Since the static tag key has limited the highest achievable privacy level of **Type 0** protocols to weak privacy only, tag key update is incorporated in the construction of protocols to help rising the privacy level. Protocols in [7,24,25,26,3] are some examples. **Type 1** protocols are **Type 0** protocols with tag key update and tag key synchronization.

Tag$\{K_{ID}^i\}$		Reader$\{ID, K_{ID}^i\}$
v: random value	$\xleftarrow{Query, c}$	*c: random challenge*
$S_{ID}^i : \{K_{ID}^i, c, v\}$		
$Response \leftarrow \mathcal{O}^{Tag}(S_{ID}^i)$		
$K_{ID}^{i+1} \leftarrow \mathcal{O}^{Update}(S_{ID}^i)$	$\xrightarrow{Response}$	*r: Response*, $\forall j \in \{ID\}$
$i = i + 1$		$K_j^d \leftarrow \mathcal{O}^{Sync}(S_j^i)$, $S_j^d : \{K_j^d, r, c\}$
		Verify if $r = \tilde{r} \leftarrow \mathcal{O}^{Reader}(S_j^d)$
		if $FOUND$, *set* `Result`(.) $= 1$,
		$K_j^i = K_j^d$; *else set* `Result`(.) $= 0$

Since $\mathcal{O}^{Update}(S_{ID}^i)$ is executed every time on the tag side, the stored K_{ID} inside the tag is always changing [3]. Although now there is tag key update, an adversary can cause desynchronization between tag and reader so that protocols with this construction can never achieve forward privacy and weak privacy. Because forward privacy is harder than weak privacy, we only need to show that weak privacy is not achievable. Consider the following attack:

1. `CreateTag`$^1(ID_0)$, `CreateTag`$^1(ID_1)$
2. $vtag \leftarrow$ `DrawTag`()
3. $\pi \leftarrow$ `Launch`()
4. $c \leftarrow$ `SendReader`$(\pi, Init)$
5. $r : Response \leftarrow$ `SendTag`$(vtag, c)$
6. (Forward r to reader to close π) $null \leftarrow$ `SendReader`(π, r)
7. (Use the same c to query $vtag$) Repeat n times:
8. $\quad r : Response \leftarrow$ `SendTag`$(vtag, c)$
9. `Free`$(vtag)$
10. $vtag' \leftarrow$ `DrawTag`()

[3] Notice that $\mathcal{O}^{Update}(S_{ID}^i)$ is executed before the tag response is sent out. Although updating the key after response does not change the protocol result, this is a good practice to avoid tag corruption by an adversary at the moment right after the response is captured but before $\mathcal{O}^{Update}(S_{ID}^i)$ is executed (i.e. keeping the old tag key in the memory).

11. $\pi' \leftarrow \texttt{Launch}()$
12. $c' \leftarrow \texttt{SendReader}(\pi', Init)$
13. $r' : Response \leftarrow \texttt{SendTag}(vtag', c')$
14. $null \leftarrow \texttt{SendReader}(\pi', r')$
15. $z \leftarrow \texttt{Result}(\pi')$
16. Queries ended, receive $\mathcal{T}(vtag) = ID_b$
17. If $z = 0$ then $x = b$. Otherwise $x = |1 - b|$
18. Output whether $\mathcal{T}(vtag') = ID_x$

An adversary running the attack above makes use of the maximum desynchronized key states n such that K_{ID}^i becomes K_{ID}^{n+1+i}. The desynchronized tag will not be recognized by the reader anymore because $\mathcal{O}^{Sync}(S_{ID}^i)$ will not run recursively beyond n (or even if n is infinity, desynchronized tag can be distinguished with a side-channel attack on the time taken for the reader to recognize that tag as described in [11]). The adversary will only fail if $\texttt{Result}(\pi')$ still outputs 1 for the desynchronized-beyond-n-tag (i.e. the tag is still authenticated). This means $K_{ID}^{n+1+i} = K_j^m$ for some $j \in \{ID\}$ and $0 \leq m \leq n$ (i.e. a duplicate tag key), which should only happen with negligible probability. Hence the adversary will succeed with overwhelming probability. Since there is no $\texttt{Corrupt}(vtag')$ in the attack, this is a significant weak privacy level attack. We have shown that RFID protocols with tag key update is not forward private and not weak private.

Remark 2. A **Type 1** protocol presented in [35] using random oracle model has been proved to provide narrow-destructive privacy, which is equivalent to narrow-forward privacy since the protocol does not have correlated secrets among tags. Hence the highest privacy level that can be attained by **Type 1** protocols is narrow-forward. We conclude with the following figure.

Type 1	*Forward levels*	*Weak levels*	*Nil*
Non-narrow levels	-	-	✓
Narrow levels	✓	✓	

Remark 3. Another interesting remark is the separation result of the weak privacy level and the narrow-forward privacy level, which was not obtained in [35] and it was asked in [29] if achieving both privacy levels with symmetric key only is feasible or not. Clearly, there are only protocols that either do not update the tag key (**Type 0**) or protocols that update it (**Type 1**). They span the whole protocol set and we do not have overlapping between weak privacy level and narrow-forward privacy level according to our results in 4.3 and 4.4. Hence we have shown the separation here and answered the question.

Remark 4. As pointed out in [23], let q be the number of queries in the above attack and assume that $q \leq n$, then there can be protocols, using symmetric key only, that achieve forward privacy level. This is the highest privacy level for symmetric key protocols. However, we do not consider that assumption in this paper.

4.5 Type 2a Protocols Can Be Reduced to Type 0 Protocols

Without reader authentication, any adversary can keep querying a tag with any compatible reader until it is desynchronized with legitimate reader. Mutual authentication protocols add an additional authentication message for the reader in the protocol construction to safeguard the query is in fact coming from a legitimate reader. **Type 2a** protocols update the tag key after such reader authentication message is received. Protocols in [9,8,12,16,18,28,32,37,6,17] are some examples. Their construction can be represented by the following figure.

Tag$\{K_{ID}^i\}$		Reader$\{ID, K_{ID}^i\}$
v: random value	$\xleftarrow{Query, c}$	*c: random challenge*
$S_{ID}^i : \{K_{ID}^i, c, v\}$		
$Response \leftarrow \mathcal{O}_1^{Tag}(S_{ID}^i)$	$\xrightarrow{Response}$	*r: Response*, $\forall j \in \{ID\}$
		$K_j^d \leftarrow \mathcal{O}^{Sync}(S_j^i)$, $S_j^d : \{K_j^d, r, c\}$
		Verify if $r = \tilde{r} \leftarrow \mathcal{O}_1^{Reader}(S_j^d)$
$a : Auth$, *Verify if*	$\xleftarrow{Auth}$	*if* $FOUND$, *set* `Result`$(.) = 1$,
$a = \tilde{a} \leftarrow \mathcal{O}_2^{Tag}(S_{ID}^i)$		$K_j^i = K_j^d$, $Auth \leftarrow \mathcal{O}_2^{Reader}(S_j^d)$;
if $MATCHED$,		*else set* `Result`$(.) = 0$
$K_{ID}^{i+1} \leftarrow \mathcal{O}^{Update}(S_{ID}^i)$,		
$i = i + 1$		

With tag key update after reader authentication, it protects the protocol from the desynchronized-beyond-n attack discussed before because each update must now come with a valid reader authentication message, which can be hard to forge. As a result, the tag key can only be desynchronized within one update. If the reader stores both the updated tag key value and the previous tag key value, in case the tag fails to update its tag key (most likely because of adversarial attacks), the reader can still authenticate the victim tag using the previous tag key in the next protocol instance. This measure is enough to provide weak privacy to this type of protocol construction.

However, imagine an offline attack to tag where invalid reader authentication message is sent. This has the same effect as if the valid reader authentication message is blocked or intercepted in an online attack but of course the former one is easier to launch. These kinds of attacks cause the tag fail to execute $\mathcal{O}^{Update}(S_{ID}^i)$ because the reader is never authenticated. It is not hard to see that the protocol is now reduced to **Type 0** protocol as if there is never an $\mathcal{O}^{Update}(S_{ID}^i)$ oracle being implemented in the protocol construction. As inherited from **Type 0** protocol, forward privacy levels cannot be achieved. A formal description of the attack is presented below:

1. $\texttt{CreateTag}^1(ID_0)$, $\texttt{CreateTag}^1(ID_1)$
2. $vtag \leftarrow \texttt{DrawTag}()$
3. $\pi \leftarrow \texttt{Launch}()$
4. $c \leftarrow \texttt{SendReader}(\pi, Init)$
5. $r : Response \leftarrow \texttt{SendTag}(vtag, c)$

6. (Forward r to reader to close π) $Auth \leftarrow \texttt{SendReader}(\pi, r)$
7. (Replace $Auth$ with a random value $a \neq Auth$)
8. $null \leftarrow \texttt{SendTag}(vtag, a)$
9. (No $\mathcal{O}^{Update}(.)$ is executed) $\texttt{Free}(vtag)$
10. $vtag' \leftarrow \texttt{DrawTag}()$
11. $K_{ID_x} \leftarrow \texttt{Corrupt}(vtag')$
12. Queries ended, receive $\mathcal{T}(vtag) = ID_b$
13. Let $S_{ID_x} : \{K_{ID_x}, r, c\}$, if $r = \tilde{r} \leftarrow \mathcal{O}^{Reader}(S_{ID_x})$ then $x = b$. Otherwise $x = |1 - b|$
14. Output whether $\mathcal{T}(vtag') = ID_x$

Other than the negligible case where $\mathcal{O}^{Reader}(S_{ID_0}) = \mathcal{O}^{Reader}(S_{ID_1})$, the above attack will only fail if the random value a is accepted by the tag such that $\mathcal{O}^{Update}(.)$ is executed to update the tag key. This should also happen with negligible probability, otherwise the reader authentication message can be easily forged. Hence the adversary will succeed with overwhelming probability. Since there is no further oracle access after $\texttt{Corrupt}(vtag')$ and no $\texttt{Result}(\pi)$ in the attack, this is a significant narrow-forward privacy level attack. We have shown that RFID protocols with tag key update after the reader is authenticated work as best as the **Type 0** protocols. We conclude with the following table.

Type 2a	*Forward levels*	*Weak levels*	*Nil*
Non-narrow levels	-	✓	✓
Narrow levels	-	✓	

4.6 Type 2b Protocols Can Be Reduced to Type 0 or Type 1 Protocols

Type 2b protocols update the tag key before the reader authentication message is received. Examples are in [29,15]. We acknowledge that the reduction from this construction type to **Type 1** is simple: an adversary just needs to block the last reader authentication message and the protocol is identical to a **Type 1** protocol. In fact, it is very uncommon to see protocols with such construction. It is only included in here for completeness. The construction can be represented by the following figure.

Tag$\{K_{ID}^i\}$		Reader$\{ID, K_{ID}^i\}$
v: random value	$\xleftarrow{Query, c}$	*c: random challenge*
$S_{ID}^i : \{K_{ID}^i, c, v\}$		
$Response \leftarrow \mathcal{O}_1^{Tag}(S_{ID}^i)$	$\xrightarrow{Response}$	*r: Response,* $\forall j \in \{ID\}$
$K_{ID}^{i+1} \leftarrow \mathcal{O}^{Update}(S_{ID}^i)$		$K_j^d \leftarrow \mathcal{O}^{Sync}(S_j^i)$, $S_j^d : \{K_j^d, r, c\}$
$i = i + 1$		*Verify if* $r = \tilde{r} \leftarrow \mathcal{O}^{Reader}(S_j^d)$
		if FOUND, set $\texttt{Result}(.) = 1$,
a : *Auth, Verify if*	$\xleftarrow{Auth}$	$K_j^i = K_j^d$, $Auth \leftarrow \mathcal{O}^{Reader}(S_j^d)$
$a = \tilde{a} \leftarrow \mathcal{O}_2^{Tag}(S_{ID}^i)$		*else set* $\texttt{Result}(.) = 0$

With tag key update before reader authentication, it makes sure that the tag key is changed even if the reader authentication message is blocked or incorrect, such that when facing a (narrow) forward privacy adversary, the corrupted tag key cannot be used to relate to any previous protocol instance. However, this is true only if tags update their keys regardless of the correctness of the reader authentication result. This means that the tag key is updated as if there is no reader authentication or a failed reader authentication does not affect the next protocol instance (e.g. a stateless RFID tag). An adversary can launch a desynchronization attack to these protocols because they do not take advantage of reader authentication. Clearly, this performs as best as **Type 1** protocols (an example in [29]). The only exception we can think of is when the tag takes the reader authentication result into account (e.g. rewinds back to the previous tag key if the reader authentication is failed) or the result will affect the next protocol instance (e.g. a stateful RFID tag). However, an adversary can still use the same attack described in section 4.5 to freeze the tag key or tag state and the protocol is reduced into a **Type 2a** protocol. We do not repeat the same attack here but conclude with the following table.

Type 2b	*Forward levels*	*Weak levels*	*Nil*
Non-narrow levels	-	✓ (stateful tag)	✓
Narrow levels	✓ (stateless tag)	✓	

5 Conclusion

We defined four RFID authentication protocol constructions and investigated on their highest achievable privacy levels. From the results we obtained, forward privacy cannot be achieved by any type of synchronized symmetric protocol constructions. Furthermore, there is no privacy improvements at all with an extra reader authentication message. After all, under the symmetric key setting, RFID authentication protocols have limited privacy protections against tag tracing and a candidate that provides both weak privacy and narrow-forward privacy protections does not exist. This provides us a potential answer to the open question in [35], which is, forward privacy without PKC is impossible. This claim remains valid until some special symmetric protocols that do not fall into one of our four constructions types can be found, then we need another examination. However, it is important for us to make ourselves clear that we do not claim our results on all the symmetric RFID protocols, instead, all our findings are bounded by the current adversary model defined in [35], [23] and [29]. This leaves the possibility that there may exist some symmetric RFID protocols not included in or well described by the Vaudenay's model where our results do not apply on them. Hence, one may be able to find alternative ways to overcome the limitations of RFID protocols by choosing more expensive cryptographic primitives in the design of RFID protocols or tweaking the privacy model where different assumptions are used in order to reflect some other RFID applications

or scenarios. With this in mind, our results are still valid as long as the RFID protocol being examined has the same settings and assumptions as stated in this paper.

References

1. Avoine, G.: Privacy Issues in RFID Banknote Protection Schemes. In: CARDIS, pp. 34–38. Kluwer Academic Publishers, Dordrecht (2004)
2. Avoine, G.: Adversarial Model for Radio Frequency Identification (2005), http://citeseer.ist.psu.edu/729798.html
3. 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)
4. 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)
5. Chien, H.-Y., Huang, C.-W.: A Lightweight RFID Protocol Using Substring. In: EUC, pp. 422–431 (2007)
6. Dimitriou, T.: A Lightweight RFID Protocol to Protect Against Traceability and Cloning Attacks. In: SecureComm (2005)
7. 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)
8. Ha, J., Moon, S.-J., Nieto, J.M.G., Boyd, C.: Low-cost and Strong-security RFID Authentication Protocol. In: EUC Workshops, pp. 795–807 (2007)
9. 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)
10. Juels, A.: RFID Security and Privacy: A Research Survey. IEEE Journal on Selected Areas in Communications 24(2), 381–394 (2006)
11. Juels, A., Weis, S.A.: Defining Strong Privacy for RFID (2006), http://citeseer.ist.psu.edu/741336.html
12. Kang, J., Nyang, D.: RFID Authentication Protocol with Strong Resistance Against Traceability and Denial of Service Attacks. In: Molva, R., Tsudik, G., Westhoff, D. (eds.) ESAS 2005. LNCS, vol. 3813, pp. 164–175. Springer, Heidelberg (2005)
13. Kim, I.J., Choi, E.Y., Lee, D.H.: Secure Mobile RFID System Against Privacy and Security Problems. In: SecPerU (2007)
14. Kim, K.H., Choi, E.Y., Lee, S.-M., Lee, D.H.: Secure EPCglobal Class-1 Gen-2 RFID System Against Security and Privacy Problems. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4277, pp. 362–371. Springer, Heidelberg (2006)
15. Lee, J., Yeom, Y.: Efficient RFID Authentication Protocols Based on Pseudorandom Sequence Generators (2008), http://eprint.iacr.org/2008/343.pdf
16. Lee, S., Asano, T., Kim, K.: RFID Mutual Authentication Scheme Based on Synchronized Secret Information. In: Symposium on Cryptography and Information Security (2006)
17. Lee, S.M., Hwang, Y.J., Lee, D.-H., Lim, J.-I.: Efficient authentication for low-cost RFID systems. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3480, pp. 619–627. Springer, Heidelberg (2005)
18. Li, Y., Ding, X.: Protecting RFID Communications in Supply Chains. In: ASIACCS, pp. 234–241. ACM Press, New York (2007)

19. Lo, N.W., Yeh, K.-H.: An Efficient Mutual Authentication Scheme for EPCglobal Class-1 Generation-2 RFID System. In: TRUST - EUC Workshops, pp. 43–56 (2007)
20. Lo, N.W., Yeh, K.-H.: Hash-based Mutual Authentication Protocol for Mobile RFID Systems with Robust Reader-side Privacy Protection. In: SenseID - ACM SenSys Workshops (2007)
21. Lo, N.W., Yeh, K.-H.: Novel RFID Authentication Schemes for Security Enhancement and System Efficiency. In: VLDB - Secure Data Management Workshops, pp. 203–212 (2007)
22. Molnar, D., Wagner, D.: Privacy and Security in Library RFID: Issues, Practices, and Architectures. In: ACM CCS, pp. 210–219 (2004)
23. Ng, C.Y., 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)
24. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Approach to "Privacy-Friendly" Tags. In: RFID Privacy Workshop (2003)
25. 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)
26. Ohkubo, M., Suzuki, K., Kinoshita, S.: Hash-Chain Based Forward-Secure Privacy Protection Scheme for Low-Cost RFID. In: SCIS (2004)
27. Ohkubo, M., Suzuki, K., Kinoshita, S.: RFID Privacy Issues and Technical Challenges. Communications of the ACM 48(9), 66–71 (2005)
28. Osaka, K., Takagi, T., Yamazaki, K., Takahashi, O.: An efficient and secure RFID security method with ownership transfer. In: Wang, Y., Cheung, Y.-m., Liu, H. (eds.) CIS 2006. LNCS (LNAI), vol. 4456, pp. 778–787. Springer, Heidelberg (2007)
29. Paise, R.-l., Vaudenay, S.: Mutual Authentication in RFID. In: ASIACCS, pp. 292–299. ACM Press, New York (2008)
30. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: 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)
31. Di Pietro, R., Molva, R.: Information Confinement, Privacy, and Security in RFID Systems. In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 187–202. Springer, Heidelberg (2007)
32. Seo, Y., Lee, H., Kim, K.: A Scalable and Untraceable Authentication Protocol for RFID. In: EUC Workshops, pp. 252–261 (2006)
33. Tsudik, G.: A Family of Dunces: Trivial RFID Identification and Authentication Protocols. In: Borisov, N., Golle, P. (eds.) PET 2007. LNCS, vol. 4776, pp. 45–61. Springer, Heidelberg (2007)
34. van Deursen, T., Radomirović, S.: Attacks on RFID Protocols (2008), http://eprint.iacr.org/2008/310.pdf
35. Vaudenay, S.: On Privacy Models for RFID. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)
36. 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)
37. Yang, J., Park, J., Lee, H., Ren, K., Kim, K.: Mutual Authentication Protocol for Low-cost RFID. In: Handout of the Ecrypt Workshop on RFID and Lightweight Crypto (2005)