

# Mutual Authentication in RFID

## Security and Privacy

Radu-Ioan Paise

EPFL

CH-1015 Lausanne, Switzerland

radu-ioan.paise@epfl.ch

Serge Vaudenay

EPFL

CH-1015 Lausanne, Switzerland

serge.vaudenay@epfl.ch

### ABSTRACT

In RFID protocols, tags identify and authenticate themselves to readers. At Asiacrypt 2007, Vaudenay studied security and privacy models for these protocols. We extend this model to protocols which offer reader authentication to tags. Whenever corruption is allowed, we prove that secure protocols cannot protect privacy unless we assume tags have a temporary memory which vanishes by itself. Under this assumption, we study several protocols. We enrich a few basic protocols to get secure mutual authentication RFID protocols which achieve weak privacy based on pseudorandom functions only, narrow-destructive privacy based on random oracles, and narrow-strong and forward privacy based on public-key cryptography.

### Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;

D.2.8 [Software Engineering]: Metrics—*complexity measures, performance measures*

### Keywords

identification protocol, privacy, RFID

## 1. RFID SECURITY AND PRIVACY

RFID refers to wireless protocols which make it possible to identify mobile tags to readers in a given system. Typically, the reader together with the system is considered as a single powerful and secure participant, but tags are tiny inexpensive devices which are not secure, prone to corruption, and with little computational capabilities. RFID tags become pervasive, thus lead us to privacy threats. For this, the study for security and privacy of RFID protocols (while preserving efficiency) has become a hot research topic.

RFID schemes are characterized by a three-fold quality. Correctness ensures that legitimate tags interacting with the reader are correctly identified or authenticated. Security says that an adversary cannot impersonate a legitimate tag

to the reader. Privacy makes sure that an adversary cannot link relationships (such as being the same) between a tag which was observed at some point at a given moment and another one. Clearly, the purpose of RFID is to identify to the reader but to nobody else.

We typically assume adversaries who can tamper with any wireless communication, pick random tags, force protocols to run with tags or the reader. Ohkubo, Suzuki, and Kinoshita [10, 11, 9] first introduced RFID which could still provide privacy even though an adversary would eventually corrupt tags to open their memory. This fits the notion of *forward privacy*. Their protocol was formally proven in an ad-hoc privacy model well fitted to their protocol. A formal definition for privacy was considered by Avoine [1, 2] with the notion of corruption except on target tags. It was extended by Juels and Weis [6] who introduced side channels: an adversary could learn whether or not a reader succeeded to identify a tag. Those models were suffering from eliminating many attacks such as those tampering with target tags, mainly to eliminate trivial attacks. Those categories of models have been generalized and classified by Vaudenay [14]. To eliminate trivial attacks, these models compare interaction of the adversary with the tags to interaction of the adversary with blinded communication to the tags.

An alternate approach relates to the universal composability model. For this, Burmester, van Le, and de Medeiros [3, 13] defined several versions of an ideal functionality.

Following [14], adversaries are called *weak* if corruption is not permitted, *forward* if corruption is performed at the end of the attack only, *destructive* if corruption destroys the tag, or *strong* otherwise. Orthogonally, adversaries are called *narrow* if they cannot learn whether the reader completed the protocol by identifying any tag or not. It was proven that narrow-strong privacy requires at least techniques which are enough to build a secure key agreement protocol and could be achieved with a secure public-key cryptosystem. Conversely, narrow-destructive privacy is possible using random oracles, and weak privacy is possible using pseudorandom functions only.

One of the concerns users may have is that a malicious reader can obtain unauthorized information from a tag, raising security or privacy issues. In order to fix this problem, beside tag's authentication, a protocol must ensure reader's authentication: it means that a tag must be confident of the reader's identity before sending any information or its ID. In this case we obtain a mutual authentication protocol. Several such protocols have been proposed: Burmester, van Le and de Medeiros proposed the O-FRAP protocol [13].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ASIACCS '08, March 18–20, Tokyo, Japan

Copyright 2008 ACM 978-1-59593-979-1/08/0003 ...\$5.00.

Tsudik [12] proposed the YA-TRIP and YA-TRAP schemes, both based on timestamps. Lim and Kwon proposed another protocol in [7]. These schemes do no use of public-key cryptography, but they are either not weak private or not narrow-forward private under our model. Although this was already mentioned by their authors, we simply stress that they are constant in not achieving both properties at the same time. So the question of achieving both without public-key cryptography is open.

### Structure of the paper.

We first recall definitions from [14] in Section 2 and extend them to address mutual authentication. Section 3 provides useful results to prove security and privacy. Then, we show the impossibility of narrow-forward privacy in Section 4. This leads us to modifying the tag model so that some temporary memory is automatically erased whenever the tag no longer received any power. Section 5 relates to already proposed protocols. Finally, Section 6 enriches the three protocols from [14] and proves their security and privacy. We obtain 3-pass protocols which are secure RFID schemes with mutual authentication. Weak privacy is achieved based on pseudorandom functions only. Narrow-destructive privacy is achieved based on random oracles. Narrow-strong and forward privacy is achieved based on an IND-CCA secure public-key cryptosystem.

## 2. DEFINITIONS

Following [14], an RFID scheme is defined by

- an algorithm  $\text{SetupReader}(1^s)$  to generate common input (typically: domain parameters, a public key  $K_P$ ), a *secret key*  $K_S$ , and initialize a *database*
- an algorithm  $\text{SetupTag}(\text{ID})$  using the common input to generate a tag-specific secret  $K$  and its initial state  $S$ . When the tag is meant to be a *legitimate* one, the entry  $(\text{ID}, K)$  is inserted in the database
- a 2-party protocol between the reader and a tag in which the reader protocol uses the common input, the database, and the secret, produces an output equal to  $\perp$  if identification failed or some ID if it succeeded, and may update the database.

The protocol is correct if executing it honestly leads to the reader to infer the correct ID of a tag when it is legitimate, or  $\perp$  when it is not, except with negligible probability.

To address mutual authentication, we enrich this definition by introducing an output on the tag side which should be OK or  $\perp$ . The protocol is correct if executing it honestly with a legitimate tag, it outputs OK, except with negligible probability.

We further define *simple protocols* in which the reader protocol follows a special form in which the communication protocol algorithm, the tag identification algorithm, and the database update algorithm satisfy the following properties.

- The reader protocol computes the protocol messages without any access to the database.
- There is a predicate  $R_{K_S}$  based on the secret key  $K_S$  on  $(\text{ID}, K, \tau)$  triplets used to define a set of all tag IDs having a database entry  $(\text{ID}, K)$  which are called *compatible* with a protocol final transcript  $\tau$ .

- The reader protocol runs an algorithm  $S_{K_S}$  on a set of compatible IDs as input to produce the output ID. This algorithm always picks an element in the input set (or fails if empty).
- After ID is output, an extra algorithm with input  $K_S$ ,  $\tau$ , and the selected  $(\text{ID}, K)$  database entry may update this entry in the database before the reader protocol terminates.

$R_{K_S}$  and  $S_{K_S}$  may be invoked as oracles in the “simple security” definition. For simplicity we omit  $K_S$  from notations.

### Adversaries.

Adversaries use the common input and may use the following oracles.

- $\text{CREATETAG}^b(\text{ID})$  to create a tag with a given ID which is legitimate (if  $b = 1$ ) or not (if  $b = 0$ ). If legitimate, this oracle updates the database of the reader. The oracle returns nothing.
- $\text{DRAWTAG}(\text{distr})$  to run a sampling algorithm  $\text{distr}$  to generate a tuple  $(\text{ID}_1, \dots, \text{ID}_n)$  of pairwise different values. If tag  $\text{ID}_i$  is legitimate,  $b_i$  is set to 1. Otherwise it is set to 0. If tag  $\text{ID}_i$  is currently *free*, it is moved to a set of *drawn* tags and assigned with a new temporary identity  $\text{vtag}_i$ ,  $p_i$  is set to  $(\text{vtag}_i, b_i)$ , and a new entry  $\text{vtag}_i \mapsto \text{ID}_i$  is inserted to the table  $\mathcal{T}$ . Otherwise,  $p_i$  is set to  $\perp$ . The oracle returns  $(p_1, \dots, p_n)$ . All ID's and table  $\mathcal{T}$  remain unknown to the adversary.
- $\text{FREE}(\text{vtag})$  to move the drawn tag with temporary identity  $\text{vtag}$  back to the set of free tags. The tag can no longer be accessed with its temporary identity  $\text{vtag}$ . The oracle returns nothing.
- $\text{LAUNCH}$  to start a new protocol session on the reader side. This oracle returns a new session identification number  $\pi$ . We assume that sessions are associated to an internal state and that sessions can run concurrently on the reader. In contrast, tags have a single state so only one session can be run.
- $\text{SENDREADER}$  and  $\text{SENDTAG}$  to send a message to a given protocol session on the reader (identified by some  $\pi$  value) or on a drawn tag (identified by its  $\text{vtag}$  value). These oracles return a message to be sent back to the counterpart.
- $\text{RESULT}(\pi)$  to get 0 if the output on session  $\pi$  is  $\perp$  and 1 otherwise.
- $\text{CORRUPT}(\text{vtag})$  to get the internal state of tag with temporary identity  $\text{vtag}$ . When  $\text{vtag}$  is no longer used, we say that the tag is *destroyed*.

The capabilities of adversaries will be kept unchanged to study mutual authentication.

We say that  $\text{vtag}$  and  $\pi$  had a *matching conversation* if there is a protocol session on  $\text{vtag}$  in which the adversary faithfully forwarded messages from one to the other (among other interaction with other tags or other reader sessions) in the right interleaved sequence.

### Security.

A scheme provides security if it provides secure *tag authentication* and secure *reader authentication*. The notion of secure tag authentication is unchanged from [14]. Basically, tag authentication is insecure if there exists a polynomial-time adversary such that one reader protocol session identified some tag ID before it was corrupted but had no matching conversation with tag ID on any drawn form, with non-negligible probability of success. Similarly, reader authentication is insecure if there exists a polynomial-time adversary such that one tag session on a legitimate tag output OK but had no matching conversation with any reader session, with non-negligible probability.

In [14], a weaker notion of security called *simple security* restricts to adversaries making no use of the RESULT oracle, creating a single tag and ending on a final SENDREADER on a reader session  $\pi$  but using the two additional oracles  $R$  and  $S$  of the simple protocol definition. The adversary succeeds if the session  $\pi$  identified the tag but that it did not have any matching conversation. It was shown that a scheme based on a simple protocol form achieving simple security also achieves secure tag authentication.

Since [14] only considered tag authentication, this definition relates to *simple tag authentication*. We enrich it with the notion of *simple reader authentication* by saying that an adversary wins if there is a tag session which ended by accepting the reader but had no matching conversation with any reader session. We have *simple security* if we have both simple tag authentication and simple reader authentication.

### Privacy.

The definition of privacy is unchanged from [14]. Basically, a scheme offers *privacy* against adversaries in class  $\mathcal{P}$  if for any adversary  $\mathcal{A}$  in  $\mathcal{P}$  which ends by getting the final table  $\mathcal{T}$  from the DRAWTAG oracle and output a Boolean, there exists a blinder  $B$  such that executing  $\mathcal{A}$  or  $\mathcal{A}^B$  leads to undistinguishable output.

A *blinder* is an algorithm who sees the common input and all interaction between the adversary and the oracles and who simulates the answers from SENDREADER, SENDTAG, and RESULT. Namely, a blinder simulates the protocol messages. An adversary breaks privacy if its result could not have been obtained without the protocol messages.

All polynomial-time adversaries are in the class STRONG. Adversaries who always destroy tags after corruption are in the subclass DESTRUCTIVE. Adversaries who never query any oracle except CORRUPT after corruption are in the subclass FORWARD. Adversaries who do no corruption are in the subclass WEAK. In addition to this, adversaries who do not query RESULT are in the subclass NARROW.

## 3. TOOLS FOR PROVABLE SECURITY

To prove security, we use the same technique as in [14]. Essentially, for a scheme accommodating a simple protocol following our definition, if we can prove simple security, i.e. security when the adversary is using a single tag, then we obtain full security. The proof is basically the same as for [14, Lemma 5].

LEMMA 1. *Let us consider an RFID scheme based on a simple protocol which is simply secure. We assume there exists a computable predicate  $R'$  such that for any matching conversation of transcript  $\tau$  between a tag ID and the reader*

*having  $(ID, K)$  in database, we have  $R(ID, K, \tau) \iff R'(n)$  where  $n$  is the number of consecutive completed protocol executions on the tag ID before since the last one with a matching conversation that led to the reader identifying ID. The scheme is secure.*

Typically,  $R'(n)$  is always true with the exception of OSK-like protocols (see Section 6.2). In this case,  $R'(n)$  is true only when  $n$  is less than a given threshold  $t$ .

Another useful tool concerns RFID protocols which have been “enriched” with an extra round from the reader to the tag which does not modify the tag state.

LEMMA 2. *Let  $S$  and  $S'$  be two RFID protocols in which  $S'$  is enriched from  $S$  by simply adding an extra message from the reader to the tag for mutual authentication. We assume that the final message does not modify the tag state. If  $S$  is correct, then  $S'$  is correct as for the reader output.*

We further recall the following lemma form [14].

LEMMA 3. *We consider an RFID scheme with the property that whenever a legitimate tag and the reader have some matching conversation, the reader does not output  $\perp$ . If the scheme offers tag authentication, then narrow-forward (resp. narrow-weak) privacy implies forward (resp. weak) privacy.*

## 4. IMPOSSIBILITY RESULT

We first show that our basic model for tags does not leave any room for privacy whenever corruption is allowed.

THEOREM 1. *In the basic model where corruption reveals the entire tag state, no RFID scheme providing secure reader authentication is narrow-forward private.*

PROOF. We consider the following narrow-forward adversary.

- 1: create two legitimate tags  $ID_0$  and  $ID_1$
- 2: draw one at random and get  $vtag$
- 3: execute a protocol between  $vtag$  and the reader but stops before the last SENDTAG( $vtag, m$ ) query (if the protocol makes it unclear which message is the last one, just guess it) and stores  $m$
- 4: free  $vtag$
- 5: draw tags  $ID_0$  and  $ID_1$
- 6: corrupt them and get their states  $S_0$  and  $S_1$
- 7: set a bit  $b$  such that simulating a tag of state  $S_b$  with the incoming message  $m$  leads to output OK (if no or both  $S_b$  work, set  $b$  to a random bit)
- 8: get  $\mathcal{T}$  and output whether  $\mathcal{T}(vtag) = ID_b$

Assuming that the SENDTAG query is really the last one, due to reader authentication we know that the tag outputs OK with negligible probability when fed with message different from  $m$  or with a non-final message  $m$  and  $\perp$  with negligible probability when fed with the final message  $m$ . So, if  $p$  is the probability for guessing the last query right, the adversary wins with probability close to  $p + \frac{1}{2}(1-p) = \frac{1+p}{2}$ . For any blinded adversary, tags run no protocol so there is a negligible probability for getting an  $m$  leading to OK, the probability for winning is close to  $\frac{1}{2}$ . Hence, the advantage is  $p/2$  which is non-negligible for any blinder. So, the adversary is significant.  $\square$

To fix this impossibility problem, we must change the tag model. Indeed, from now on we assume that some temporary memory is automatically erased from the tag as soon

as the tag is put back in the set of free tags. This is quite a reasonable assumption since temporary memory requires power to be maintained. It further thwarts the previous attack since step 4 will flush out the information about the internal state which is needed in step 7.

## 5. CASE STUDIES

In this section, we study several protocols based on symmetric cryptography only. We notice that they all fail to provide either weak privacy or narrow-forward privacy. Our results do not contradict the security and privacy results from their authors. As a matter of fact, this was already made clear from their papers. Our point is that all protocols so far are constant in not achieving both properties. It thus seems to be hard to address weak privacy and narrow-forward privacy at the same time by using only symmetric cryptography. So far, we do not know whether this is feasible or not.

### 5.1 Weak Privacy Failures

In [12], Tsudik proposed YA-TRAP and YA-TRAP\*, two authentication protocols. These protocols are both based on timestamps, which makes the system vulnerable to denial of service attacks. In YA-TRAP, the reader sends a random challenge  $R_r$ , together with a reader timestamp  $T_r$ . The tag checks  $T_r$ , and if it is valid (reader's timestamp should be higher than tag's timestamp, but should be lower than the maximum value  $T_{max}$ ), it updates its timestamp  $T_t$ , and computes the response to the reader's challenge, using a MAC and its specific secret  $K_i$ . For the authentication, it picks a random challenge  $R_t$ , and computes a MAC using the two random challenges ( $R_r, R_t$ ) and  $K_i$ . The reader verifies the two MACs in the database.

For YA-TRAP\*, Tsudik introduced a denial of service resistance, which in fact only limits the period in which the tag is out of service. As the significant adversaries for weak privacy for the two protocols are similar, we present the one for YA-TRAP for more readability.

We can perform the same kind of attack as the one of Juels-Weis against the Modified Ohkubo-Suzuki-Kinoshita protocol [6]. The difference appears in the manners in which the tag is desynchronized with the reader: in the initial case (Juels-Weis attack) it is desynchronized by several fake authentication request, while in this case it can be easily desynchronized by sending as the timestamp, the maximum possible value. In this case, the tag will update its timestamp value to the maximum possible value, so future authentication requests will clearly fail. The formal attack is:

- 1: create two legitimate tags  $ID_0$  and  $ID_1$
- 2:  $(vtag_{0..}) \leftarrow \text{GETTAG}(ID_0)$
- 3:  $\text{SENDTAG}(vtag_0, T_{max}, R_r)$
- 4:  $\text{FREE}(vtag_0)$
- 5: draw one tag at random and get  $vtag$
- 6:  $\pi \leftarrow \text{LAUNCH}$
- 7:  $\text{EXECUTE}(vtag)$
- 8:  $x \leftarrow \text{RESULT}(\pi)$
- 9: output whether  $\mathcal{T}(vtag) = ID_x$

This is clearly a significant adversary for weak privacy. Thus, YA-TRAP is not weak private, and furthermore neither is YA-TRAP\*. It can be observed that the tag specific secret  $K_i$  is not updated, so narrow-forward privacy is not achieved either.

Due to the Juels-Weis attack [6] on OSK [10, 11], the protocol based on OSK from Section 6 does not achieve weak privacy. (See [14].)

### 5.2 Narrow-Forward Privacy Failures

Our OSK-based protocol from Section 6 achieves narrow-forward privacy but not weak privacy because the tag state is updated before the tag authenticated the reader. If we now consider the same protocol in which the tag state is updated *after* the reader authentication, we show that narrow-forward privacy is no longer achieved. Let us consider the following adversary.

- 1: create two legitimate tags  $ID_0$  and  $ID_1$
- 2:  $(vtag_{0..}) \leftarrow \text{GETTAG}(ID_0)$
- 3:  $\pi \leftarrow \text{LAUNCH}$
- 4:  $a \leftarrow \text{SENDREADER}(\pi)$
- 5:  $c \leftarrow \text{SENDTAG}(vtag_0, a)$
- 6:  $\text{FREE}(vtag_0)$
- 7:  $d \leftarrow \text{SENDREADER}(\pi, c)$
- 8: draw one tag at random and get  $vtag$
- 9:  $S \leftarrow \text{CORRUPT}(vtag)$
- 10: **if**  $d = F'(S, a)$  **then**
- 11:      $x \leftarrow 0$
- 12: **else**
- 13:      $x \leftarrow 1$
- 14: **end if**
- 15: output whether  $\mathcal{T}(vtag) = ID_x$

We have  $\Pr[A \text{ wins}] \approx 1$ . For any blinder  $B$ ,  $\Pr[A^B \text{ wins}] = \frac{1}{2}$ . Therefore the adversary is a significant narrow-forward adversary (no active action after corruption and no RESULT query), so this protocol would not be narrow-forward private.

We now present the attack against the O-FRAP protocol [3]. This protocol is initiated by the reader which sends a random value  $r_{sys}$ . The tag uses a pseudo-random function  $F$  to compute four values  $\nu_1, \nu_2, \nu_3, \nu_4$  from the tag's key  $k_{tag}^a, r_{sys}$  and a tag's random value  $r_{tag}$ . The first value,  $\nu_1$  is used to update  $r_{tag}$ , the second one,  $\nu_2$  is sent by the tag to the reader in order to authenticate itself;  $\nu_3$  is sent by the reader, to authenticate itself for the tag and  $\nu_4$  is used to update the tag's internal key  $k_{tag}^a$ . The reader keeps a copy of the previous value of the tag key in order to be able to authenticate the tag, even if for any reason, the tag did not update its key. The problem appears when the last message of the protocol (that authenticate the reader) is blocked. We present a significant narrow-forward adversary against this protocol:

- 1: create two legitimate tags  $ID_0$  and  $ID_1$
- 2: draw one at random and get  $vtag$
- 3:  $(r_{tag}, \nu_2) \leftarrow \text{SENDTAG}(r_{sys})$
- 4:  $\text{FREE}(vtag)$
- 5:  $(vtag_{0..}) \leftarrow \text{GETTAG}(ID_0)$
- 6:  $K \leftarrow \text{CORRUPT}(vtag_0)$
- 7: **if**  $(\cdot, \nu_2, \cdot, \cdot) = F(K, r_{sys}, r_{tag})$  **then**
- 8:      $x \leftarrow 0$
- 9: **else**
- 10:      $x \leftarrow 1$
- 11: **end if**
- 12: output whether  $\mathcal{T}(vtag) = ID_x$

We have  $\Pr[A \text{ wins}] \approx 1$ . For any blinder  $B$ ,  $\Pr[A^B \text{ wins}] = \frac{1}{2}$ . Therefore the adversary is a significant narrow-forward

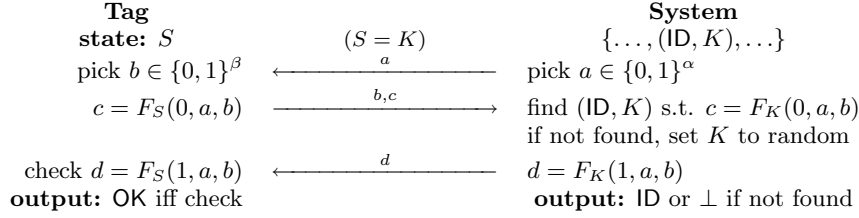


Figure 1: A Weak-Private RFID Scheme based on PRF.

adversary (no active action after corruption and no RESULT query), so the scheme is not narrow-forward private.

In [7], Lim and Kwon proposed an authentication protocol that we will call LK. It is based on the OSK key chain protocol. In order to avoid key desynchronization, they introduce a second key chain, which is updated by the reader and verified by the tag. The problem appears when an adversary queries a tag more than  $m$  times: the tag state becomes static until the first correct authentication with the reader. During this time, there exists a significant narrow-forward adversary  $\mathcal{A}$ , which we present below:

- 1: create two legitimate tags  $\text{ID}_0$  and  $\text{ID}_1$
- 2:  $(\text{vtag}_{0,\cdot}) \leftarrow \text{GETTAG}(\text{ID}_0)$
- 3: **for**  $i = 1$  **to**  $m + 1$  **do**
- 4:   pick a random  $r_1$
- 5:    $\text{SENDTAG}(\text{vtag}_{0,\cdot}, r_1)$
- 6:   wait for the tag's response
- 7:   pick a random  $\sigma_1$
- 8:    $\text{SENDTAG}(\text{vtag}_{0,\cdot}, \sigma_1)$
- 9: **end for**
- 10:  $\pi \leftarrow \text{LAUNCH}$
- 11:  $r_1 \leftarrow \text{SENDREADER}(\pi)$
- 12:  $(t_i, r_2, \sigma_1) \leftarrow \text{SENDTAG}(\text{vtag}_{0,\cdot}, a)$
- 13:  $\text{FREE}(\text{vtag}_{0,\cdot})$
- 14: draw one tag at random and get  $\text{vtag}$
- 15:  $s_i \leftarrow \text{CORRUPT}(\text{vtag})$
- 16: **if**  $\sigma_1 = \text{ext}(f(s_i, r_1 \parallel r_2), l_1)$  **then**
- 17:    $x \leftarrow 0$
- 18: **else**
- 19:    $x \leftarrow 1$
- 20: **end if**
- 21: output whether  $\mathcal{T}(\text{vtag}) = \text{ID}_x$

We have  $\Pr[\mathcal{A} \text{ wins}] \approx 1$ . For any blinder  $B$ , we have  $\Pr[\mathcal{A}^B \text{ wins}] = \frac{1}{2}$ . Therefore the adversary is a significant narrow-forward adversary (no active action after corruption and no RESULT query), so LK protocol is not narrow-forward private.

## 6. ENRICHED PROTOCOLS

### 6.1 Weak Privacy based on PRF

A pseudorandom function family (PRF) is a family of functions  $(F_{s,K})_{K \in \{0,1\}^{k(s)}}$  from  $\{0,1\}^{\delta(s)}$  to  $\{0,1\}^{\gamma(s)}$  such that  $k, \delta, \gamma$  are polynomially bounded,  $2^{-\delta(s)}$ , and  $2^{-\gamma(s)}$  are negligible,  $F_{s,K}(x)$  is computable in polynomial time, and any distinguisher with polynomial complexity has a negligible advantage for distinguishing an oracle simulating  $F_{s,K}$  initialized with a random  $K$  from an oracle initialized with

a truly random function. (For more readability we omit the parameter  $s$ .)

We enrich the protocol from [14] based on a pseudorandom function to achieve security and weak privacy with an extra round. We follow Fig. 1 with  $\alpha = \beta = \frac{\delta-1}{2}$ . The tag setup  $\text{SetupTag}(\text{ID})$  picks a random  $k$ -bit key  $K$  and sets  $S = K$ .

1. The reader picks a random  $\alpha$ -bit string  $a$  and sends it to the tag.
2. The tag with state  $S$  sends a random  $\beta$ -bit string  $b$  and  $c = F_S(0, a, b)$  to the reader.
3. The reader looks for  $(\text{ID}, K)$  in the database such that  $c = F_K(0, a, b)$ , gets  $\text{ID}$ , and sends back  $d = F_K(1, a, b)$ . (If no entry is found,  $d$  is computed with a random  $K$ .)
4. The reader checks  $d$  to authenticate the reader.

The protocol is equivalent to the ISO/IEC 9798-2 3-pass mutual authentication protocol that is used in [5] and to the CR building block of [8]. It originally comes from the variant from Weis et al. [15].

**THEOREM 2.** *If  $F$  is a PRF, the above RFID scheme is secure and weak private.*

The original scheme from [14] is not narrow-forward. The enriched one is not either. This is the same as in Section 5.2.

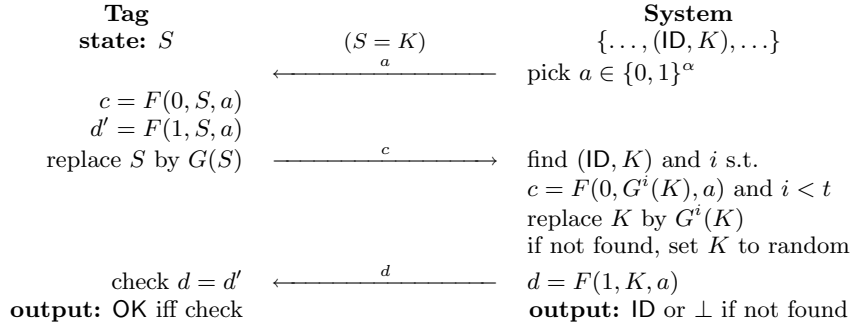
**PROOF.** We apply Lemma 2 to get reader correctness. Correctness of the tag output works as in [14].

In the simple security model where there is a single tag, no corruption, and no RESULT queries, we replace the  $F$  computations by using the lazy sampling technique as follows. First, we assume that the reader never picks the same  $a$  twice and that the tag never picks the same  $b$  twice so all computations hold on different inputs. Then, we simulate all  $c$  and  $d$  releases by random strings and show this does not affect the success probability of the adversary, thanks to the PRF property. We can then easily show that the success probability is  $2^{-\gamma}$  times the total number of session instances on both sides. Since this is negligible, we deduce simple security thus security from Lemma 1.

To prove weak privacy, we only prove narrow-weak privacy and apply Lemma 3. The above lazy sampling technique is also well fitted to narrow-weak model, so a trivial blinder which just picks random  $a, b, c$ , and  $d$  will work.  $\square$

### 6.2 Narrow-Destructive Privacy in the Random Oracle Model

We now enrich the protocol from [14] based on random oracles. We use two oracles  $F$  and  $G$  implementing two random functions from  $\{0,1\}^{\alpha+k+1}$  and  $\{0,1\}^k$  to  $\{0,1\}^k$ ,



**Figure 2: A Narrow-Destructive-Private RFID Scheme based on a Random Oracle.**

respectively. The tag generation  $\text{SetupTag}(\text{ID})$  picks a random  $k$ -bit key  $K$  and sets the initial state to  $S = K$ . The protocol is depicted on Fig. 2.

1. The reader picks a random  $\alpha$ -bit string  $a$  and sends it to the tag.
2. The tag with state  $S$  sends the value  $c = F(0, S, a)$ , stores  $d' = F(1, S, a)$  in temporary memory, then refreshes its state  $S$  with  $G(S)$ .
3. The reader looks for  $(\text{ID}, K)$  in the database such that  $c = F(0, G^i(K), a)$  with  $i < t$ , gets ID, sends  $d = F(1, G^i(K), a)$  to the tag, and replaces entry  $(\text{ID}, K)$  by  $(\text{ID}, G^i(K))$  in the database.
4. The tag checks  $d = d'$ .

After  $t$  malicious queries to the tag, it is “grilled” because its state and the database are de-synchronized. Thus, the hypothesis of Lemma 3 is not fulfilled.

There exist many variants of this protocol [2, 4, 10, 11, 9].

**THEOREM 3.** *Assuming that the parameters  $k$  and  $t$  are polynomially bounded and that  $2^{-k}$  is negligible, the above scheme is a secure and narrow-destructive private RFID scheme in the random oracle model.*

As already seen, this protocol fails to be weak private. This is the same as in Section 5.1.

**PROOF.** We apply Lemma 2 to get reader correctness. In cases where the tag was correctly identified by the reader, tag correctness is rather trivial.

To prove security, we only have to prove simple reader authentication and to apply Lemma 1. To do so, we assume w.l.o.g. that the reader never picks the same  $a$  twice and that iterating  $S \leftarrow G(S)$  on the tag does not cycle during the attack. We consider a protocol transcript  $(a, c, d)$  on the tag side with no matching conversation with the reader and we stop the adversary before sending  $d$  to the tag.

If  $a$  was not released by any protocol session, then the query  $(1, S, a)$  was only made once to compute the expected  $d'$ . Hence, the value of  $d'$  is random and matches  $d$  with negligible probability.

If  $a$  was released by a protocol session, from our assumption it was selected only once in a session  $\pi$ . Let  $\hat{c}$  the value sent back to the reader and  $\hat{d}$  its response. If  $c \neq \hat{c}$ , the reader cannot identify the tag thanks to simple tag authentication. Hence,  $\hat{d}$  comes from a random query. So, the value of  $d'$  is random and matches  $d$  with negligible probability.

Finally, if  $c = \hat{c}$ , since conversations are not matching, we have  $d \neq \hat{d}$ . If the reader does not identify the tag, we are back to the previous case. Otherwise, we have  $d' = \hat{d}$  thus  $d \neq d'$ : the tag does not authenticate the reader.

Hence, we have a secure simple reader authentication. To prove a secure simple tag authentication, we proceed similarly.

The proof for narrow-destructive privacy from [14] also works for the enriched protocol if we simulate  $d$  on the reader side in the same way that we simulate  $c$  on the tag side: following the lazy sampling technique.  $\square$

### 6.3 Narrow-Strong+Forward Privacy

Finally, we enrich the protocol from [14] based on an IND-CCA Public-Key Cryptosystem (PKC). A PKC includes a key generator, an encryption algorithm, and a decryption algorithm. Correctness ensures that the decryption of the encryption of any  $x$  is always  $x$ . The scheme is IND-CCA-secure if all polynomial-time adversaries win the IND-CCA with negligible advantage. In the IND-CCA game, the adversary receives a public key, does decryption queries, submits two plaintexts, receives the encryption of one of the two, further do decryption queries except on the challenged ciphertext, and tries to guess which plaintext was encrypted.

The reader setup algorithm first generates a private/public key pair  $(K_S, K_P)$ . The tag setup algorithm  $\text{SetupTag}(\text{ID})$  picks a  $k$ -bit key  $K$  and sets the initial state to

$$S = (K_P, \text{ID}, K).$$

The parameter  $k$  and  $\alpha$  must be polynomially bounded. The protocol is depicted on Fig. 3.

1. The reader sends an identification request with an  $\alpha$ -bit random  $a$ .
2. The tag picks a random  $\beta$ -bit  $b$ , stores it in temporary memory, and sends  $c = \text{Enc}_{K_P}(\text{ID}||K||a||b)$  to the reader.
3. The reader gets  $\text{ID}||K||a||d = \text{Dec}_{K_S}(c)$  and checks that  $a$  is correct and that  $(\text{ID}, K)$  is in database.<sup>1</sup> If not,  $d$  is sent to a random value. The reader then sends  $d$  to the tag.

<sup>1</sup>As in [14] we can later use  $K = F_{K_M}(\text{ID})$  with a PRF  $F$  and a master secret  $K_M$  as depicted on Fig. 3 to get rid of the database. Thanks to the PRF property, this change does not modify the privacy result.

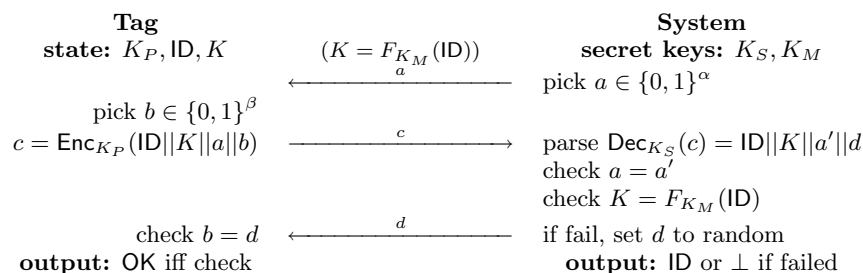


Figure 3: A Narrow-Strong and Forward -Private RFID Scheme based on a PKC.

4. The tag checks that  $b = d$ .

**THEOREM 4.** *If the public-key cryptosystem is IND-CPA-secure then the above RFID scheme is narrow-strong private. If the cryptosystem is IND-CCA-secure and  $2^{-k}$  is negligible, the RFID scheme is further secure and forward private.*

Namely, with an IND-CCA PKC, this RFID scheme achieves privacy with respect to the class

FORWARD  $\cup$  (NARROW  $\cap$  STRONG).

**PROOF.** Correctness of the protocol is trivial from the correctness of the cryptosystem.

To prove security, we only have to prove simple security and to apply Lemma 1.

We assume w.l.o.g. that the reader never picks the same  $a$  twice. We consider a protocol transcript  $(a, c, d)$  on the tag side with no matching conversation with the reader and we stop the adversary before sending  $d$  to the tag. If  $(a, c)$  has a matching conversation, then the reader released some  $\hat{d}$  which must be different from  $d$  so reader authentication fails. Otherwise, sending  $c$  in any other reader protocol session would not match the corresponding  $\hat{a}$  since it must differ from  $a$ , so the simulation of the reader for sending  $c$  is easy: one just picks a random  $\hat{a}$ . So, we can simulate the reader by using a decryption oracle that is never queried with  $c$ . If we now simulate the tag by asking the encryption of  $\text{ID}||K||a$  concatenated with a random  $b$  chosen by a challenger, we obtain an IND-CCA adversary who guesses this  $b$ . Thanks to IND-CCA security, it succeeds with negligible probability. This proves simple reader authentication.

Simple tag authentication works as in the proof from [14].

The scheme is narrow-strong private, thus narrow-forward private. Thanks to Lemma 3, we deduce that it is forward private. So, we only have to prove narrow-strong privacy. But the proof of [14] works the same in the enriched protocol.  $\square$

## 7. CONCLUSION

We have shown how to formalize the notion of mutual authentication in RFID schemes, as well as security and privacy. To address corruption, we must assume that tags have temporary memory which erases itself when the tag does not receive any power. We have identified protocols which fail to provide privacy. We further enriched previously proposed protocols to achieve mutual authentication. Finally, we let open the problem of achieving weak and narrow-forward privacy based on no public-key cryptography.

## 8. REFERENCES

- [1] G. Avoine. *Cryptography in Radio Frequency Identification and Fair Exchange Protocols*. PhD thesis, EPFL, Lausanne, Switzerland, December 2005. <http://library.epfl.ch/theses/?nr=3407>.
- [2] G. Avoine, E. Dysli, and P. Oechslin. Reducing time complexity in RFID systems. In B. Preneel and S. Tavares, editors, *Selected Areas in Cryptography – SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 291–306, Kingston, Canada, August 2005. Springer-Verlag.
- [3] M. Burmester, T. van Le, and B. de Medeiros. Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2006*, Baltimore, Maryland, USA, August-September 2006. IEEE.
- [4] T. Dimitriou. A lightweight RFID protocol to protect against traceability and cloning attacks. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*, Athens, Greece, September 2005. IEEE.
- [5] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. In M. Joye and J.-J. Quisquater, editors, *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370, Boston, Massachusetts, USA, August 2004. IACR, Springer-Verlag.
- [6] A. Juels and S. Weis. Defining strong privacy for RFID. Cryptology ePrint Archive, Report 2006/137, 2006.
- [7] C. H. Lim and T. Kwon. Strong and robust RFID authentication enabling perfect ownership transfer. In P. Ning, S. Qing, and N. Li, editors, *Conference on Information and Communications Security – ICICS 2006*, volume 4307 of *Lecture Notes in Computer Science*, pages 1–20, Raleigh, North Carolina, USA, December 2006. Springer-Verlag.
- [8] D. Molnar and D. Wagner. Privacy and security in library RFID: Issues, practices, and architectures. In B. Pfitzmann and P. Liu, editors, *Conference on Computer and Communications Security – ACM CCS 2004*, pages 210–219, Washington, DC, USA, October 2004. ACM, ACM Press.
- [9] M. Ohkubo and K. Suzuki. RFID privacy issues and

- technical challenges. *Communications of the ACM*, 48(9):66–71, 2005.
- [10] M. Ohkubo, K. Suzuki, and S. Kinoshita. Cryptographic approach to “privacy-friendly” tags. In *RFID Privacy Workshop 2003*, MIT, MA, USA, November 2003.
- [11] M. Ohkubo, K. Suzuki, and S. Kinoshita. Efficient hash-chain based RFID privacy protection scheme. In *International Conference on Ubiquitous Computing – Ubicomp 2004, Workshop Privacy: Current Status and Future Directions*, Nottingham, England, September 2004.
- [12] G. Tsudik. A family of dunces: Trivial RFID identification and authentication protocols. In N. Borisov and P. Golle, editors, *Privacy Enhancing Technologies – PET 2007*, volume 4776 of *Lecture Notes in Computer Science*, pages 45–61, Ottawa, Canada, 2007. Springer-Verlag.
- [13] T. van Le, M. Burmester, and B. de Medeiros. Universally composable and forward secure RFID authentication and authenticated key exchange. In F. Bao and S. Miller, editors, *ACM Symposium on Information, Computer and Communications Security – ASIACCS 2007*, pages 242–252, Singapore, 2007. ACM.
- [14] S. Vaudenay. On privacy models for RFID. In T. Okamoto, editor, *Advances in Cryptology – Asiacrypt 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 68–87, Kuching, Malaysia, December 2007. Springer-Verlag.
- [15] S. Weis, S. Sarma, R. Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In D. Hutter, G. Müller, W. Stephan, and M. Ullmann, editors, *International Conference on Security in Pervasive Computing – SPC 2003*, volume 2802 of *Lecture Notes in Computer Science*, pages 454–469, Boppard, Germany, March 2003. Springer-Verlag.