Computer Science Journal of Moldova, vol.**30**, no.3(90), 2022

Privacy and Reader-first Authentication in Vaudenay's RFID Model with Temporary State Disclosure

Ferucio Laurenţiu Ţiplea, Cristian Hristea, Rodica Bulai

Abstract

Privacy and mutual authentication under corruption with temporary state disclosure are two significant requirements for real-life applications of RFID schemes. This paper proposes two practical RFID schemes that meet these requirements. They differ from other similar schemes in that they provide reader-first authentication. Regarding privacy, our first scheme achieves destructive privacy, while the second one – narrow destructive privacy in Vaudenay's model with temporary state disclosure. To achieve these privacy levels, we use Physically Unclonable Functions (PUFs) to assure that the internal secret of the tag remains hidden from an adversary with invasive capabilities. Both of our schemes avoid the use of random generators on tags. Detailed security and privacy proofs are provided.

Keywords: Computer security, authentication, privacy, cryptography, PUF, RFID system.

MSC 2010: 94A60, 94A62.

1 Introduction

Radio Frequency Identification (RFID) refers to a technology whereby digital data encoded in *RFID tags* is transmitted to a *reader* via radio waves. A *back-end system*, which has an online database, is securely connected to the reader to collect, filter, process, and manage RFID data. It also stores complete information associated with the RFID tags in order to be able to authenticate them.

doi:10.56415/csjm.v30.18

The RFID technology has been implemented in many significant areas such as toll collection systems, identification and tracking of various kinds of objects, consumer products, or access control. With the increasing usages to healthcare, electronic passports, and personal ID cards, the potential security threats and compliance risks have become enormous. In such a context, the need for secure and private communication protocols between reader and tags becomes crucial. Moreover, when developing such protocols, account must be taken of the adversary model to which they should resist. A widely accepted adversary model was proposed in [1], [2], now called *Vaudenay's RFID model.* According to it, an adversary can create legitimate or illegitimate tags, draw one or more tags according to some chosen probability distribution, release drawn tags, launch protocol instances with drawn tags, send messages to reader or drawn tags, corrupt drawn tags to retrieve their internal states, or get the result of a completed protocol instance.

Vaudenay's model classifies adversaries into eight classes and provides, consequently, eight levels of RFID privacy. Among these, destructive privacy (corruption destroys the tag) together with reader-first authentication under corruption with temporary state disclosure plays an important role in practice. For instance, tag destruction under corruption is an important requirement when the tag is used for access control. Likewise, the disclosure of temporary state under tag corruption is a serious threat in practice. Reader-first authentication [3] assures that the tag will give its private data only when it authenticates the reader. Therefore, tag tracking and data theft are prevented when the reader is fake. All these together mean that we need RFID schemes that provide destructive privacy and reader-first authentication under corruption with temporary state disclosure.

Contribution. When Vaudenay's model was proposed, it was not very clear whether the tag corruption reveals the permanent state or the full (permanent and temporary) state of the tag. Later, this aspect was clarified and it was shown that the mutual authentication protocols proposed in [2] do not achieve the claimed privacy level under corruption with temporary state disclosure. Additionally, this does not even happen [4] with newer protocols like those in [5], [6].

In this paper, we provide two mutual authentication RFID schemes that achieve destructive and narrow destructive privacy in Vaudenay's model with temporary state disclosure. Moreover, in our schemes, the tag authenticates first the reader (this is the reader-first approach [3]), which guarantees the information goes from tag to a trusted reader.

It is known that no privacy level can be achieved with ordinary tags (tags that only run cryptographic primitives) under mutual authentication and corruption with temporary state disclosure [7]. Therefore, the schemes we propose are based on *PUF tags*, that is tags endowed with *physically unclonable functions* (PUFs), a novel class of hardware security primitives that are in use for a while. The security proofs we provide to our schemes are very detailed. We also elaborate on the tag-reader desynchronization problem.

Related work. The pseudo-random function (PRF) based RFID scheme in [2] achieves weak privacy and mutual authentication in Vaudenay's model. It is straightforward to see that the proof in [2] works even in the case of corruption with temporary state disclosure. The first PUF-based RFID scheme that achieves destructive privacy and mutual authentication in Vaudenay's model (where corruption does not disclose the temporary state of tags) was proposed in [4], as an extension of the scheme in [8], [9] (that only achieves unilateral authentication).

In [5], [6], two PUF-based RFID schemes have been proposed and claimed that they achieve (narrow) destructive privacy and mutual authentication in Vaudenay's model with temporary state disclosure. Unfortunately, neither of them reaches even narrow forward privacy [4].

Paper structure. The paper consists of six sections, the first one being the introduction. The basic terminology and notation used throughout this paper is introduced in Sections 2 and 3. Our first RFID scheme, that achieves destructive privacy and mutual authentication in Vaudenay's model with temporary state disclosure, is presented in Section 4. In the fifth section, we introduce our second RFID scheme that achieves narrow destructive privacy and mutual authentication in the same model. The last section concludes the paper.

2 Basic definitions and notation

We fix here the basic terminology and notation for our paper.

Probabilistic polynomial time algorithms and negligible functions. We use *probabilistic polynomial time* (PPT) algorithms $\mathcal{A}$ as defined in [10]. If $\mathcal{O}$ is an oracle, then $\mathcal{A}^{\mathcal{O}}$ denotes that $\mathcal{A}$ has oracle access to $\mathcal{O}$. When the oracle $\mathcal{O}$ implements some function f, we simply write $\mathcal{A}^f$ to denote that $\mathcal{A}$ has oracle access to f. This means that whenever $\mathcal{A}$ sends a value x to the oracle, it gets back $f(x)$.

If A is a set, then $a \leftarrow A$ means that a is uniformly at random chosen from A. If $\mathcal{A}$ is a probabilistic algorithm, then $a \leftarrow \mathcal{A}$ means that a is an output of $\mathcal{A}$ for some given input.

The asymptotic approach to security makes use of security parameters, denoted by λ in our paper. A positive function $f(\lambda)$ is called *negligible* if, for any positive polynomial $poly(\lambda)$, there exists n_0 such that $f(\lambda) < 1/poly(\lambda)$, for any $\lambda \geq n_0$.

Pseudo-random functions. Let ℓ_1 and ℓ_2 be two polynomials with positive values. Given a set $\mathcal{K}$ of *keys* and $\lambda \in \mathbb{N}$, define $\mathcal{K}_\lambda = \{K \in \mathcal{K} \mid |K| = \lambda\}$. A *family of functions* indexed by $\mathcal{K}$ is a construction $F = (F_K)_{K \in \mathcal{K}}$, where F_K is a function from $\{0,1\}^{\ell_1(|K|)}$ to $\{0,1\}^{\ell_2(|K|)}$. We also define $U_\lambda = \{f \mid f : \{0,1\}^{\ell_1(\lambda)} \to \{0,1\}^{\ell_2(\lambda)}\}$ and $U = (U_\lambda)_\lambda$.

We say that F is *computationally indistinguishable* from U if, for any PPT algorithm $\mathcal{A}$ with oracle access to functions, its *advantage* $Adv^{prf}_{\mathcal{A},F}(\lambda) = |P(1 \leftarrow \mathcal{A}^{F_K}(1^\lambda) : K \leftarrow \mathcal{K}_\lambda) - P(1 \leftarrow \mathcal{A}^{g}(1^\lambda) : g \leftarrow U_\lambda)|$ is negligible (as a function of λ).

$F = (F_K)_{K \in \mathcal{K}}$ is called a *pseudo-random function* (PRF) if it is efficiently computable and computationally indistinguishable from U.

Physically unclonable functions. A *physically unclonable function* (PUF) can be seen as a physical object that, when queried with a challenge x, generates a response y that depends on both x and the specific physical properties of the object. PUFs are typically assumed to be *physically unclonable* (it is infeasible to produce two PUFs that cannot be distinguished based on their challenge/response behavior), *unpredictable* (it is infeasible to predict the response to an unknown challenge), and *tamper-evident* (any attempt to physically access the PUF irreversible changes the challenge/response behavior).

From a theoretical point of view, a PUF (sometimes called *ideal PUF*) is a physical object with a challenge/response behavior that implements a function $P : \{0,1\}^p \to \{0,1\}^k$, where p and k are of polynomial size in λ, such that P is computationally indistinguishable from U, and any attempt to physically tamper with the object implementing P results in the destruction of P (P cannot be evaluated any more).

3 RFID schemes

From an informal point of view, an RFID system [11], [12] consists of a *reader*, a set of *tags*, and a *communication protocol* between reader and tags. The reader is a transceiver that has associated a database that stores information about tags. Its task is to identify *legitimate tags* (that is, tags with information stored in its database) and to reject all the other incoming communication. The reader and its database are trusted entities, and the communication between them is secure. A tag is a transponder device with much more limited computation capabilities than the reader. Depending on tag, it can perform simple logic operations, symmetric key, or even public key cryptography. Each tag has a *permanent* (or *internal*) *memory* that stores the state values, and a *temporary* (or *volatile*) *memory* that can be viewed as a set of *volatile variables* used to carry out the necessary computations.

RFID schemes. Let $\mathcal{R}$ be a *reader identifier* and $\mathcal{T}$ be a set of *tag identifiers* whose cardinal is polynomial in some security parameter λ. An *RFID scheme over* $(\mathcal{R}, \mathcal{T})$ [1], [2] is a triple $\mathcal{S} = (SetupR, SetupT, Ident)$ of PPT algorithms, where:

1. $SetupR(\lambda)$ inputs a security parameter λ and outputs a triple (pk, sk, DB) consisting of a key pair (pk, sk) and an empty database DB. pk is public, while sk is kept secret by reader;
2. $SetupT(pk, ID)$ initializes the tag identified by ID. It outputs an initial tag state S and a secret key K. A triple $(ID, f(S), K)$ is stored in the reader's database DB, where f is a public function that extracts some information from tag's initial state S;
3. $Ident(pk; \mathcal{R}(sk, DB); ID(S))$ is an interactive protocol between the reader identified by $\mathcal{R}$ (with its private key sk and database DB) and a tag identified by ID (with its state S) in which the reader ends with an output consisting of ID or $\perp$. The tag may end with no output (*unilateral authentication*), or it may end with an output consisting of OK or $\perp$ (*mutual authentication*).

SetupR(λ) "creates" a reader $\mathcal{R}$ and initializes it, *SetupT*(pk, ID) "creates" a tag $\mathcal{T}_{ID}$, initializes it with an initial tag state, and also registers this tag with the reader by storing some information about it in the reader's database.

The *correctness* of an RFID scheme means that, regardless of how the system is set up, after each complete execution of the interactive protocol between the reader and a legitimate tag, the reader outputs tag's identity with overwhelming probability. For mutual authentication of RFID schemes, *correctness* means that the reader outputs the tag's identity, and the tag outputs OK with overwhelming probability.

An *RFID system* is an instantiation of an RFID scheme.

Adversaries. The two most basic security requirements for RFID schemes are *authentication* and *untraceability.* To formalize them, the concept of an *adversary model* is needed. There have been several proposals for this, such as [1], [2], [13]–[18]. One of the most influential, which we follow in this paper, is *Vaudenay's model* [1], [2]. We recall below this model as in [4]. Thus, we assume first that some oracles the adversary may query share and manage a common list of tags $ListTags$, which is initially empty. This list includes exactly one entry for each tag created and active in the system. A tag entry consists of several fields with information about the tag, such as: the (permanent) identity of the tag (which is an element from $\mathcal{T}$), the temporary identity of the tag (this field may be empty saying that the tag is *free*), a bit value saying whether the tag is legitimate (the bit is one) or illegitimate (the bit is zero). When the temporary identity field is non-empty, its value uniquely identifies the tag, which is called *drawn* in this case. The adversary may only interact with drawn tags by means of their temporary identities.

The oracles an adversary may query are:

1. $CreateTag^b(ID)$: Creates a free tag $\mathcal{T}_{ID}$ with the identifier ID by calling the algorithm *SetupT*(pk, ID) to generate a pair (K, S). If $b = 1$, $(ID, f(S), K)$ is added to DB, and the tag is considered *legitimate*; otherwise ($b = 0$), the tag is considered *illegitimate.* Moreover, a corresponding entry is added to $ListTags$;
2. $DrawTag(\delta)$: This oracle chooses a number of free tags according to the distribution δ, let us say n, and draws them. That is, n

temporary identities $vtag_1, \ldots, vtag_n$ are generated, and the corresponding tag entries in $ListTags$ are filled with them. The oracle outputs $(vtag_1, b_1, \ldots, vtag_n, b_n)$, where b_i specifies whether the tag $vtag_i$ is legitimate or not;

3. $Free(vtag)$: Removes the temporary identity $vtag$ in the corresponding entry in $ListTags$, and the tag becomes free. The identifier $vtag$ will no longer be used. We assume that when a tag is freed, its temporary state is erased;
4. $Launch()$: Launches a new protocol instance and assigns a unique identifier to it. The oracle outputs the identifier;
5. $SendReader(m, \pi)$: Outputs the reader's answer when the message m is sent to it as part of the protocol instance π. When m is the empty message, abusively but suggestively denoted by $\emptyset$, this oracle outputs the first message of the protocol instance π, assuming that the reader does the first step in the protocol;
6. $SendTag(m, vtag)$: outputs the tag's answer when the message m is sent to the tag referred to by $vtag$. When m is the empty message, this oracle outputs the first message of the protocol instance π, assuming that the tag does the first step in the protocol;
7. $Result(\pi)$: Outputs $\perp$ if in session π the reader has not yet made a decision on tag authentication (this also includes the case when the session π does not exist), 1 if in session π the reader authenticated the tag, and 0 otherwise (this oracle is both for unilateral and mutual authentication);
8. $Corrupt(vtag)$: Outputs the current permanent (internal) state of the tag referred to by $vtag$, when the tag is not involved in any computation of any protocol step (that is, the permanent state before or after a protocol step).

We emphasize that $Corrupt$ does not return snapshots of the tag's memory during its computations. When the $Corrupt$ oracle returns the full state, we will refer to this model as being *Vaudenay's model with temporary state disclosure.*

Now, the adversaries are classified into the following classes, according to the access they get to these oracles:

- *Weak adversaries*: they do not have access to the $Corrupt$ oracle;

- *Forward adversaries*: once they access the *Corrupt* oracle, they can only access the *Corrupt* oracle;
- *Destructive adversaries*: after querying $Corrupt(vtag)$ and obtaining the corresponding information, the tag identified by $vtag$ is destroyed (marked as destroyed in $ListTags$), and the temporary identifier $vtag$ will no longer be available. The database DB will still keep the record associated to this tag (the reader does not know the tag was destroyed). As a consequence, a new tag with the same identifier cannot be created;
- *Strong adversaries*: there are no restrictions on the use of oracles.

Orthogonal to these classes, there is the class of *narrow* adversaries that do not have access to the *Result* oracle. We may now combine the narrow constraint with any of the previous constraints in order to get another four classes of adversaries, *narrow weak*, *narrow forward*, *narrow destructive*, and *narrow strong*.

Security. Now we are ready to introduce the *tag* and *reader authentication* properties as proposed in [1], [2], simply called the *security* of RFID schemes. First of all, we say that a tag $\mathcal{T}_{ID}$ and a protocol session π *had a matching conversation* if they exchanged well interleaved and faithfully (but maybe with some time delay) messages according to the protocol, starting with the first protocol message but not necessarily completing the protocol session. If the matching conversation leads to tag authentication, then it will be called a *tag authentication matching conversation*; if it leads to reader authentication, it will be called a *reader authentication matching conversation*.

Now, the tag authentication property is defined by means of an experiment that a challenger sets up for a *strong* adversary $\mathcal{A}$ (after the security parameter λ is fixed). In the experiment, the adversary is given the public parameters of the scheme and is allowed to query the oracles. If there has been a session in which the reader has authenticated an uncorrupted tag without a tag authentication matching conversation, then the experiment returns 1 (or 0 otherwise).

The advantage of $\mathcal{A}$ in the experiment $RFID^{t\text{-}auth}_{\mathcal{A},\mathcal{S}}(\lambda)$ is defined as

$$Adv^{t\text{-}auth}_{\mathcal{A},\mathcal{S}}(\lambda) = Pr(RFID^{t\text{-}auth}_{\mathcal{A},\mathcal{S}}(\lambda) = 1).$$

An RFID scheme $\mathcal{S}$ achieves *tag authentication* if $Adv^{t\text{-}auth}_{\mathcal{A},\mathcal{S}}$ is negligible, for any strong adversary $\mathcal{A}$.

The experiment for reader authentication, denoted $RFID^{r\text{-}auth}_{\mathcal{A},\mathcal{S}}(\lambda)$, is quite similar to that above. The main difference compared to the previous experiment is that the adversary $\mathcal{A}$ tries to make some legitimate tag to authenticate the reader. As π and $\mathcal{T}_{ID}$ have no matching conversation, $\mathcal{A}$ computes at least one message that makes the tag to authenticate the reader.

An RFID scheme $\mathcal{S}$ achieves *reader authentication* if the advantage of $\mathcal{A}$, $Adv^{r\text{-}auth}_{\mathcal{A},\mathcal{S}}$, is negligible, for any strong adversary $\mathcal{A}$ ($Adv^{r\text{-}auth}_{\mathcal{A},\mathcal{S}}$ is defined as above, by using $RFID^{r\text{-}auth}_{\mathcal{A},\mathcal{S}}(\lambda)$ instead of $RFID^{t\text{-}auth}_{\mathcal{A},\mathcal{S}}(\lambda)$).

Privacy. *Privacy* for RFID systems [2] captures anonymity and untraceability. It basically means that an adversary cannot learn anything new from intercepting the communication between a tag and the reader. To model this, the concept of a *blinder* was introduced in [2].

A *blinder for an adversary* $\mathcal{A}$ that belongs to some class V of adversaries is a PPT algorithm $\mathcal{B}$ that simulates the *Launch*, *SendReader*, *SendTag*, and *Result* oracles for $\mathcal{A}$, without having access to the corresponding secrets. Moreover, it looks passively at the communication between $\mathcal{A}$ and the other oracles allowed to it by the class V (that is, $\mathcal{B}$ gets exactly the same information as $\mathcal{A}$ when querying these oracles).

When the adversary $\mathcal{A}$ interacts with the RFID scheme by means of a blinder $\mathcal{B}$, we say that $\mathcal{A}$ is *blinded by* $\mathcal{B}$ and denote this by $\mathcal{A}^{\mathcal{B}}$.

Given an adversary $\mathcal{A}$, define the experiment (privacy game):

```
Experiment $RFID^{prv-0}_{\mathcal{A},\mathcal{S}}(\lambda)$
1: Set up the reader;
2: $\mathcal{A}$ gets the public key $pk$;
3: $\mathcal{A}$ queries the oracles;
4: $\mathcal{A}$ gets the secret table of the $DrawTag$ oracle;
5: $\mathcal{A}$ outputs a bit $b'$;
6: Return $b'$.
```

In the same way, by replacing "$\mathcal{A}$" with "$\mathcal{A}^{\mathcal{B}}$", we define the experiment $RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda)$. Now, the *advantage* of $\mathcal{A}$ blinded by $\mathcal{B}$ is

$$Adv^{prv}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda) = \mid P(RFID^{prv-0}_{\mathcal{A},\mathcal{S}}(\lambda) = 1) - P(RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda) = 1) \mid .$$

An RFID scheme is private for a class V of adversaries if, for any $\mathcal{A} \in V$, there exists a blinder $\mathcal{B}$ such that $Adv^{prv}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda)$ is negligible.

We thus obtain eight concepts of privacy: *strong privacy*, *narrow strong privacy*, *destructive privacy*, and so on.

4 Destructive privacy and reader-first authentication

An interesting question that arises when designing mutual authentication RFID schemes is whether the tag or the reader should be authenticated first. We have thus two approaches: *tag-first* and *reader-first authentication*, respectively [3]. The *tag-first authentication* has some advantage with respect to desynchronization: the tag computes its new state and sends information about it to the reader. However, the tag state is updated only when the reader authenticates the tag and confirms the new state to the tag. The disadvantage of this approach is that the tag should provide some information to the reader before it is confident of the reader's identity.

The *reader-first authentication* might enhance the tag privacy because the tag gives private information to the reader when it is confident of its identity. This also might help preventing adversaries from tracking tags. Another advantage is when the tag is designed only for a limited number of authentications. In such a case, the reader-first approach prevents a form of the denial of service attack that would "consume" all the tag's authentication answers.

In this section, we address the problem to construct a destructive private and mutual authentication RFID scheme in Vaudenay's model with temporary state disclosure. For mutual authentication, we follow the reader-first approach and endow all tags with PUFs.

To describe our scheme, let us assume that λ is a security parameter, $\ell_1(\lambda)$ and $\ell_2(\lambda)$ are two polynomials, and $F = (F_K)_{K \in \mathcal{K}}$ is a pseudorandom function, where $F_K : \{0,1\}^{2\ell_1(\lambda)+1} \to \{0,1\}^{\ell_1(\lambda)}$ for all $K \in \mathcal{K}_\lambda$. Each tag is equipped with a (unique) PUF $P : \{0,1\}^{p(\lambda)} \to \mathcal{K}_\lambda$ and has the capacity to compute F, where $p(\lambda)$ is a polynomial. The internal state of the tag consists of a pair (s, x), where $s \in \{0,1\}^{p(\lambda)}$ is randomly chosen as a seed to evaluate P, and $x \in \{0,1\}^{\ell_1(\lambda)}$ is a random string that gets incremented after each protocol instance. The reader maintains a database DB with entries for all legitimate tags. Each entry is a vector (ID, K), where ID is the tag's identity and $K = P(s)$, where P is the tag's PUF, and (s, x) is its state.

	Reader (DB, F)		**Tag** (P, s, F, x)
1		$\xleftarrow{u, z}$	$x = x + 1$, $K = P(s)$ $u = F_K(0, 0, x)$, $z = F_K(0, u, 0)$ erase K, u, z
2	If $\exists(ID, K) \in DB$ s.t. $z = F_K(0, u, 0)$ then $v \leftarrow \{0, 1\}^{\ell_1(\lambda)}$, $w = F_K(1, v, u)$ else $v \leftarrow \{0, 1\}^{\ell_1(\lambda)}$ $w \leftarrow \{0, 1\}^{\ell_1(\lambda)}$	$\xrightarrow{v, w}$	
3		$\xleftarrow{w'}$	$K = P(s)$, $u = F_K(0, 0, x)$ If $w = F_K(1, v, u)$ then $w' = F_K(1, v, u + 1)$ else $w' = F_K(1, v + 1, u)$ erase K, v, w, w'
4	If $w' = F_K(1, v, u + 1)$ then output ID else output $\perp$		

Figure 1. Destructive private and reader-first authentication

The mutual authentication protocol is given in Figure 1. As we can see, the tag initially increments x and computes $K = P(s)$, $u = F_K(0, 0, x)$, $z = F_K(0, u, 0)$. The tuple (u, z) is then sent to the reader. The reader checks its database for a tuple (ID, K) such that $z = F_K(0, u, 0)$. When the reader finds out the right value, it prepares the answer for the tag by generating a random $v \leftarrow \{0, 1\}^{\ell_1(\lambda)}$ and computes $w = F_K(1, v, u)$. If no such entry is found, then the reader chooses both v and w as random values. The tag evaluates the PUF, checks the value w received from the reader, takes a decision, and prepares the answer for the reader. On receiving the tag's answer, the reader checks it and takes a decision.

Theorem 4.1. *The RFID scheme in Figure 1 is correct.*

Proof. Assuming that a tag $\mathcal{T}_{ID}$ is legitimate, the reader's database contains an entry (ID, K), where $K = P(s)$, (s, x) is the tag's state, and P is its PUF.

When the reader receives (u, z) from the tag $\mathcal{T}_{ID}$, exactly the equality $z = F_K(0, u, 0)$ holds with overwhelming probability (we use the same notation as in Figure 1).

If the reader has found the tag in its database (i.e., identified it), the equality $w = F_K(1, v, u)'$ on tag's side holds with overwhelming probability. This means that the tag authenticates the reader. In such a case, the equality $w' = F_K(1, v, u + 1)$ holds with overwhelming probability, meaning that the reader authenticates the tag.

As a final remark, if the tag does not authenticate the reader, then the reader will authenticate the tag with negligible probability. □

We will focus now on the security of our RFID scheme.

Theorem 4.2. *The RFID scheme in Figure 1 achieves tag authentication in Vaudenay's model with temporary state disclosure, provided that F is a PRF and the tags are endowed with ideal PUFs.*

Proof. Assume that the scheme does not achieve tag authentication, and let $\mathcal{A}$ be an adversary that has non-negligible advantage over the scheme, with respect to the tag authentication property. We will show that there exists a PPT algorithm $\mathcal{A}'$ that can break the pseudorandomness property of the function F.

The main idea is the next one. Let $\mathcal{C}$ be a challenger for the pseudorandomness security game of the function F. The adversary $\mathcal{A}'$ will play the role of challenger for $\mathcal{A}$. Thus, $\mathcal{A}'$ guesses the identity ID^* of an uncorrupted legitimate tag that has no matching conversation with the reader, but $\mathcal{A}$ can make the reader authenticate it with a non-negligible probability (recall that there is a polynomial number $t(\lambda)$ of tags). Then, it creates the tag $\mathcal{T}_{ID^*}$ with the help of $\mathcal{C}$. Specifically, $\mathcal{A}'$

will not associate any key with the identity ID^*. However, the function chosen by $\mathcal{C}$ will be the one used to do all the calculations of this tag. This function is either $F_{K^*} \leftarrow F$ for some K^*, or a random function. $\mathcal{T}_{ID^*}$ will be regarded by $\mathcal{A}$ as a legitimate tag. The adversary $\mathcal{A}'$ does not know this function but, with the help of $\mathcal{A}$, he will try to distinguish between the two cases with non-negligible probability.

The details on $\mathcal{A}'$ are as follows (λ is a security parameter):

1. The challenger $\mathcal{C}$ chooses uniformly at random $F_{K^*} \leftarrow F$ for some K^*, or a random function. Let us denote it by f;
2. $\mathcal{A}'$ plays the role of challenger for $\mathcal{A}$. It will run the reader and all tags created by $\mathcal{A}$, answering all $\mathcal{A}$'s oracle queries. Therefore, using $SetupR(\lambda)$, it generates a triple (pk, sk, DB), gives the public key pk to $\mathcal{A}$, and keeps the private key sk.

   $\mathcal{A}'$ will maintain a list of tag entries $\mathcal{A}'_{ListTags}$ similar to $ListTags$ (see Section 3) but with the difference that each entry in this list also includes the current state of the tag as well as a special field designated to store the "key generated by the tag's internal PUF". The legitimate entries in this list define the reader's database DB. Initially, $\mathcal{A}'_{ListTags}$ is empty;
3. $\mathcal{A}'$ guesses the tag identity ID^* that $\mathcal{A}$ will authenticate to reader (please see the discussion above);
4. $\mathcal{A}'$ will simulate for $\mathcal{A}$ all the corresponding oracles in a straightforward manner, but with the following modifications:

   (a) $CreateTag^b(ID)$: If $\mathcal{T}_{ID}$ was already created, then $\mathcal{A}'$ does nothing. If $\mathcal{T}_{ID}$ was not created and $ID \neq ID^*$, then $\mathcal{A}'$ randomly chooses $K \in \{0,1\}^\lambda$ and $x \in \{0,1\}^{\ell_1(\lambda)}$ and records a corresponding entry into $\mathcal{A}'_{ListTags}$ (K plays the role of the key generated by the tag's internal PUF). Thus, $\mathcal{T}_{ID}$ has just been created. If $\mathcal{T}_{ID}$ was not created and $ID = ID^*$, then $\mathcal{A}'$ records $(ID^*, ?, x)$ into $\mathcal{A}'_{ListTags}$, where $x \leftarrow \{0,1\}^{\ell_1(\lambda)}$. The meaning of "?" is that this field should have contained a key for F. However, $\mathcal{A}'$ does not even know if $\mathcal{C}$ chose a function from F (so a key) or a random function. However, $\mathcal{A}'$ does not need to know this because it can answer all $\mathcal{A}$'s queries regarding ID^* with the help of $\mathcal{C}$.

As the tags are endowed with ideal PUFs and the keys are uniformly at random chosen by $\mathcal{A}'$, including the function chosen by $\mathcal{C}$, $\mathcal{A}'$ implements correctly the functionality of all tags (including $\mathcal{T}_{ID^*}$);

(b) $DrawTag$ and $Free$: $\mathcal{A}'$ knows the list of all tags created by $\mathcal{A}$, and updates it correspondingly whenever $\mathcal{A}$ draws or frees some tag;

(c) $Launch()$: $\mathcal{A}'$ launches a new protocol instance whenever $\mathcal{A}$ asks for it;

(d) $SendTag(\emptyset, vtag)$: This is the first message $vtag$ sends in a protocol instance. If the tag referred by $vtag$ is ID^*, then $\mathcal{A}'$ will increment x and then query $\mathcal{C}$ for $(0, 0, x)$, which will become u, and for $(0, u, 0)$, which will become z. If u, z are $\mathcal{C}$'s responses, then $\mathcal{A}'$ answers with (u, z).

If $vtag$ refers to some $ID \neq ID^*$, then $\mathcal{A}'$ can prepare the answer because it knows the corresponding key for ID;

(e) $SendReader((u, z), \pi)$: Assume the reader (run by $\mathcal{A}'$) has received (u, z) in the protocol instance π from a tag identified by $vtag$ (in other words, $(u, z) \leftarrow SendTag(\emptyset, vtag)$).

If $vtag$ refers to some tag ID such that $(ID, K) \in DB$ for some K, then the reader (run by $\mathcal{A}'$) can compute the answer according to the protocol.

If $vtag$ refers to ID^*, then the reader (run by $\mathcal{A}'$) can compute the answer according to the protocol by querying $\mathcal{C}$ (recall that $\mathcal{T}_{ID^*}$ is regarded by $\mathcal{A}$ as a legitimate tag).

If $vtag$ refers to some ID for which no entry can be found in DB, then the answer (v, w) is randomly chosen;

(f) $SendTag((v, w), vtag)$: If the tag referred by $vtag$ is ID^*, then $\mathcal{A}'$ queries $\mathcal{C}$ for $(1, v, u)$ and then compares the answer with w. If they match, the tag outputs OK; otherwise, it outputs $\perp$. In the first case, $\mathcal{A}'$ queries $\mathcal{C}$ for $(1, v, u + 1)$ to get w'; in the second case, it queries $\mathcal{C}$ for $(1, v + 1, u)$. If $vtag$ refers to some $ID \neq ID^*$ that has associated a pair (K, x), then $\mathcal{A}'$ can compute by itself w' (according to the protocol). In all cases, the oracle returns w';

(g) *Result*(π) : $\mathcal{A}'$ can infer the decision of the reader in the last step of π because it can obtain the value $F_K(1, v, u)$ for all tags (either it can compute it or query $\mathcal{C}$ for it). Therefore, $\mathcal{A}'$ can simulate *Result*(π) according to its definition;

(h) *Corupt*$(vtag)$: If the tag referred by $vtag$ is different from ID^*, then $\mathcal{A}'$ returns its current state; otherwise, it aborts.

If $\mathcal{A}'$ sees that $\mathcal{A}$ could make the reader authenticate $\mathcal{T}_{ID^*}$ without corrupting it and without any matching conversation between tag and reader, it answers to $\mathcal{C}$ that f is PRF; otherwise, f is random. In the first case,

$$P(1 \leftarrow (\mathcal{A}')^{F_K}(1^\lambda) \;:\; K \leftarrow \mathcal{K}_\lambda) = P(RFID^{t\_auth}_{\mathcal{A},\mathcal{S}}(\lambda) = 1).$$

In the second case,

$$P(1 \leftarrow \mathcal{A}^f(1^\lambda) \;:\; f \leftarrow U_\lambda) = \eta(\lambda)$$

for some negligible function $\eta(\lambda)$. That is because $\mathcal{A}$ does not play the real tag authentication game, the function implemented by $\mathcal{T}_{ID^*}$ is random, and the tag does not have any matching conversation with the reader. So, the reader (simulated by $\mathcal{A}'$) should authenticate the tag on behalf of a random message (u, z) sent by $\mathcal{A}$ to reader, message that is verified for correctness by $\mathcal{C}$.

Therefore,

$$Adv^{prf}_{\mathcal{A},F}(\lambda) = |P(RFID^{t\_auth}_{\mathcal{A},\mathcal{S}}(\lambda) = 1) - \eta(\lambda)|.$$

If we assume now that $\mathcal{A}$ has a non-negligible probability to make the reader authenticate the tag $\mathcal{T}_{ID^*}$, then $\mathcal{A}'$ will have a non-negligible advantage against F; this contradicts the fact that F is a pseudo-random function. □

As with respect to the reader authentication property, we have the following result.

Theorem 4.3. *The RFID scheme in Figure 1 achieves reader authentication in Vaudenay's model with temporary state disclosure, provided that F is a PRF and the tags are endowed with ideal PUFs.*

Proof. Assume that our scheme does not achieve reader authentication, and let $\mathcal{A}$ be an adversary that has a non-negligible advantage over the scheme, with respect to the reader authentication property. We will show that there exists a PPT algorithm $\mathcal{A}'$ that can break the pseudo-randomness property of the function F.

The main idea is somewhat similar to the one in the Theorem 4.2. Let $\mathcal{C}$ be a challenger for the pseudo-randomness property of the function F. The adversary $\mathcal{A}'$ will play the role of a challenger for $\mathcal{A}$. First, $\mathcal{A}'$ guesses the identity ID^* of an uncorrupted legitimate tag that has no matching conversation with the reader, but $\mathcal{A}$ can make the tag authenticate the reader with a non-negligible probability (recall that there is a polynomial number $t(\lambda)$ of tags). Then, it creates the tag $\mathcal{T}_{ID^*}$ with the help of $\mathcal{C}$, exactly as in the proof of Theorem 4.2. This tag will be regarded by $\mathcal{A}$ as a legitimate one.

The description of $\mathcal{A}'$ is very similar to the one in the proof of Theorem 4.2, so we will focus on the differences between them (λ denotes a security parameter):

1. The challenger $\mathcal{C}$ chooses uniformly at random $F_{K^*} \leftarrow F$ for some K^*, or a random function. Let us denote it by f;

2. $\mathcal{A}'$ plays the role of a challenger for $\mathcal{A}$. It will run the reader and all tags created by $\mathcal{A}$, answering all $\mathcal{A}$'s oracle queries. Therefore, using $SetupR(\lambda)$ it generates a triple (pk, sk, DB), gives the public key pk to $\mathcal{A}$, and keeps the private key sk.

   $\mathcal{A}'$ will maintain a list of tag entries $\mathcal{A}'_{ListTags}$ exactly as in the proof of Theorem 4.2;

3. $\mathcal{A}'$ guesses the tag identity ID^* that authenticates $\mathcal{A}$ as a valid reader;

4. $\mathcal{A}'$ will simulate for $\mathcal{A}$ all the corresponding oracles exactly as in the proof of Theorem 4.2.

The advantage of $\mathcal{A}'$ against the PRF F is computed as in the proof of Theorem 4.2. Therefore, the assumption that $\mathcal{A}$ has a non-negligible probability to make $\mathcal{T}_{ID^*}$ authenticate it as a valid reader contradicts the pseudo-randomness of the function F. □

By using the *sequence-of-games* approach [19], we will prove that our protocol reaches destructive privacy. With this approach, a sequence of games (probabilistic experiments) is defined. The initial game is the original privacy game with respect to a given adversary. The transition from one game G_i to another one G_{i+1} is done by indistinguishability in our case. This means that a probability distribution in G_i is replaced by another one that is indistinguishable from the previous one. In this way, the difference between the probabilities the adversary wins G_i and G_{i+1}, is negligible.

Theorem 4.4. *The RFID scheme in Figure 1 achieves destructive privacy in Vaudenay's model with temporary state disclosure, provided that F is a PRF and the tags are endowed with ideal PUFs.*

Proof. Let $\mathcal{A}$ be a destructive adversary against our RFID scheme denoted $\mathcal{S}$. We will show that there is a blinder $\mathcal{B}$ such that $Adv^{prv}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda)$ is negligible. The blinder $\mathcal{B}$ that we construct, which has to answer to the oracles *Launch*, *SendReader*, *SendTag*, and *Result* without knowing any secret information, works as follows:

- $Launch()$: returns a unique identifier π for a new protocol instance;
- $SendTag(\emptyset, vtag)$: returns (u, z), where $u, z \leftarrow \{0,1\}^{\ell_1(\lambda)}$;
- $SendReader((u, z), \pi)$: returns (v, w), where $v, w \leftarrow \{0,1\}^{\ell_1(\lambda)}$;
- $SendTag((v, w), vtag)$: returns $w' \leftarrow \{0,1\}^{\ell_1(\lambda)}$;
- $SendReader(w', \pi)$: the blinder does not do anything because, in this case, the reader does not answer;
- $Result(\pi)$: if the session π does not exist or exists but is not completed, the blinder outputs $\perp$. If π has been issued by the $Launch()$ oracle and a protocol transcript $tr_\pi = ((u, z), (v, w), w')$ has been generated by
  - $(u, z) \leftarrow SendTag(\emptyset, vtag)$,
  - $(v, w) \leftarrow SendReader((u, z), \pi)$,
  - $w' \leftarrow SendTag((v, w), vtag)$, and
  - $SendReader(w', \pi)$,

where *vtag* refers to some legitimate tag, the blinder outputs 1; otherwise, outputs 0 (remark that the blinder sees what $\mathcal{A}$ sees and, therefore, it knows whether *vtag* refers to some legitimate tag or not).

We further prove that $Adv^{prv}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda)$ is negligible. To this we define a sequence of games $G_0, \ldots, G_7$, where G_0 is the experiment $RFID^{prv-0}_{\mathcal{A},\mathcal{S}}$ and G_{i+1} is obtained from G_i as described below, for all $0 \leq i < 7$. By $P(G_i)$ we denote the probability the adversary $\mathcal{A}$ wins the game G_i.

Game G_1: This is identical to G_0 except that the game challenger will not use the PRF keys generated by PUFs to answer the adversary's oracle queries, but randomly generated keys, one for each tag created by adversary. Of course, the game challenger must maintain a secret table with the association between each tag and this new secret key. From the adversary's point of view, this means that the probability distribution given by each tag's PUF (in G_0) is replaced by the uniform probability distribution (in G_1). As the PUFs are ideal, the two distributions are indistinguishable. Taking into account that there are a polynomial number of tags, it must be the case that $|P(G_0) - P(G_1)|$ is negligible.

Game G_2: We replace in G_1 the oracle *Result* by $Result_{\mathcal{B}}$, which is the simulation of *Result* by the blinder $\mathcal{B}$ (please, see the definition $\mathcal{B}$). Denote by G_2 the game such obtained. We prove that $P(G_1) = P(G_2)$.

Recall first that in game G_1 the tags are still endowed with PUFs, but their secret PRF keys are not computed by PUFs. They are randomly generated by the game challenger that maintains a secret table with the key associated to each tag. In this way, the *Corrupt* oracle will never reveal the secret key, but it destroys the tag when queried.

If $\mathcal{A}$ queries *Result* or $Result_{\mathcal{B}}$ for a protocol session that does not exist or is incomplete, both oracles return $\perp$. Therefore, let us assume that these oracles are queried on a complete protocol session π. In this case, we will show that $Result(\pi) = 1$ if and only if $Result_{\mathcal{B}}(\pi) = 1$.

Assume $Result(\pi) = 1$. Then, there is a transcript $tr_\pi = ((u, z), (v, w), w')$ defined by a sequence of oracle queries

- $(u, z) \leftarrow SendTag(\emptyset, vtag)$
- $(v, w) \leftarrow SendReader((u, z), \pi)$

- $w' \leftarrow SendTag((v, w), vtag)$
- and $SendReader(w', \pi)$

such that $vtag$ refers to some tag $\mathcal{T}_{ID}$ whose state is (s, x) and secret key is K, $u = F_K(0, 0, x)$, $z = F_K(0, 0, u)$, and (ID, K) is in the reader's database (that is, $\mathcal{T}_{ID}$ is legitimate). All these facts show that $Result_{\mathcal{B}}(\pi) = 1$ (recall that the blinder $\mathcal{B}$ sees what $\mathcal{A}$ sees and, therefore, it knows whether $vtag$ refers to some legitimate tag or not).

The inverse implication is a bit more elaborate. Assume that $Result_{\mathcal{B}}(\pi) = 1$. This means that there is a transcript $tr_\pi = ((u, z), (v, w), w')$ defined by a sequence of oracle queries as those above and the tag $\mathcal{T}_{ID}$ referred by $vtag$ is legitimate. Assume that the tag's key is K and its state is (s, x), and in DB there is a record (ID, K). Because the oracles $SendReader$ and $SendTag$ are the real ones (and not simulated by blinder), the reader finds a record such that $z = F_K(0, 0, u)$. Therefore, w must be of the form $F_K(0, v, u)$, and this value will match $F_K(0, v, u)$ computed by tag. Therefore, the tag authenticates the reader and replies by $w' = F_K(1, v, u)$. But then, the reader will successfully check the equality between w and $F_K(1, v, u)$ (computed by itself) and, therefore, authenticates the tag. As a conclusion, $Result(\pi) = 1$.

This shows that $P(G_1) = P(G_2)$.

Game G_3**:** This game is identical to G_2, except that the $Launch()$ oracle is simulated according to the blinder description. No difference is encountered between the two games and, therefore, $P(G_2) = P(G_3)$.

Game G_4**:** This is identical to G_3 except that the $SendTag(\emptyset, vtag)$ oracle is simulated according to the blinder description. By doing this, the probability distribution $\{(u, z) \mid u = F_K(0, 0, x),\ z = F_K(0, u, 0)\}$ is replaced by $\{(u, z) \mid u, z \leftarrow \{0, 1\}^{\ell_1(\lambda)}\}$.

As F is a PRF, $|P(G_3) - P(G_4)|$ is negligible. The proof is quite straightforward. The main idea is as follows. Assume that an adversary $\mathcal{A}$ can distinguish with a non-negligible probability between G_3 and G_4. Define an adversary $\mathcal{A}'$ for PRF that uses $\mathcal{A}$ as a subroutine and send $(0, u, 0)$ as a challenge. When the PRF challenger returns, with equal probability, either $z = F_K(0, u, 0)$ or $z \leftarrow \{0, 1\}^{\ell_1(\lambda)}$, $\mathcal{A}'$ sends this value to $\mathcal{A}$. The probability $\mathcal{A}'$ guesses between the two possibilities for z is exactly the probability $\mathcal{A}$ distinguishes between the two games.

Game G_5: This game is identical to G_4, except that the oracle $SendReader((u,z),\pi)$ is simulated according to the blinder description. That is, for each tag $\mathcal{T}_{ID}$ whose secret key is K and current state is (s,x), one of the two probability distributions

$$\begin{cases} \{(u,z,v,w) \mid u,v,z \leftarrow \{0,1\}^{\ell_1(\lambda)}, w = F_K(1,v,u)\}, \\ \{(u,z,v,w) \mid u,v,z,w \leftarrow \{0,1\}^{\ell_1(\lambda)}\}, \end{cases}$$

is replaced by $\{(u,z,v,w) \mid u,v,z,w \leftarrow \{0,1\}^{\ell_1(\lambda)}\}$.

As F is a PRF, and the key K was chosen at random, it must be the case that $|P(G_4) - P(G_5)|$ is negligible. The proof is by contradiction, and it is quite similar to the proof that establishes the transition from G_3 to G_4.

Game G_6: This game is identical to G_5, except that the oracle $SendTag((v,w),vtag)$ is simulated by blinder. That is, for each tag $\mathcal{T}_{ID}$, one of the two probability distributions

$$\begin{cases} \{(u,z,v,w,w') \mid u,v,z,w \leftarrow \{0,1\}^{\ell_1(\lambda)},\ w' = F_K(1,v,u+1)\}, \\ \{(u,z,v,w,w') \mid u,v,z,w \leftarrow \{0,1\}^{\ell_1(\lambda)},\ w' = F_K(1,v+1,u)\}, \end{cases}$$

is replaced by $\{(u,z,v,w,w') \mid u,v,z,w,w' \leftarrow \{0,1\}^{\ell_1(\lambda)}\}$.

As F is a PRF, and the key K was chosen at random, it must be the case that $|P(G_5) - P(G_6)|$ is negligible. The proof is by contradiction, and it is quite similar to the proof in Game G_5. Therefore, it is omitted.

Game G_7: This is identical to G_6, except that $SendReader(w',\pi)$ is simulated by blinder. However, this does not change the probability distribution from G_6. Therefore, $P(G_6) = P(G_7)$.

Now, we show that G_7 is in fact $RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}$. The blinded adversary $\mathcal{A}^{\mathcal{B}}$ sees each tag as a standard PUF tag, although random secret keys are used instead of the keys generated by PUFs. The oracles $CreateTag$, $Draw$, $Free$, and $Corrupt$ that can be queried directly by $\mathcal{A}$ do not use the keys generated by PUFs in order to answer the adversary's queries (in fact, they do not use any secret key). The answer to the other oracles is simulated by a blinder that does not use the secret keys either. Therefore, G_7 is indeed $RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}$.

Now, remark that $P_{\mathcal{A}}(G_0) = P(RFID^{prv-0}_{\mathcal{A},\mathcal{S}}(\lambda) = 1)$ and $P_{\mathcal{A}}(G_7) = P(RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda) = 1)$. Combining all the probabilities $P(G_i)$ together, we obtain that $Adv^{prv}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda)$ is negligible and, therefore, our protocol achieves destructive privacy. □

5 Narrow destructive privacy and reader-first authentication

With little effort, we can design a similar scheme that achieves narrow destructive privacy and reader-first authentication RFID in Vaudenay's model with temporary state disclosure. The mutual authentication protocol of this new RFID scheme is presented in Figure 2; all the other elements are as in Section 4, except that F_K is a function from $\{0,1\}^{\ell_1(\lambda)+2}$ to $\{0,1\}^{\ell_2(\lambda)}$ and t is polynomial in the security parameter.

As one can see, there is no random generator on tag. Because of this, the synchronization between tag and reader can be lost. The only thing we can do is to check (on the reader side) for a polynomial bounded desynchronization. Due to this, the scheme can be at most narrow destructive private: if an adversary desynchronizes the tag and reader sufficiently enough (for more than t steps), then it will be able to distinguish the real privacy game from the blinded one by means of the *Result* oracle. Roughly speaking, this is because, in the real privacy game, the *Result* oracle returns 0 (when the tag and reader are desynchronized for more than t steps), while in the blinded privacy game, it returns 1. We, therefore, have the following result.

Theorem 5.1. *The RFID scheme in Figure 2 achieves mutual authentication and narrow destructive privacy in Vaudenay's model with temporary state disclosure, provided that F is a PRF and the tags are endowed with ideal PUFs.*

Proof. It is straightforward to see that the proof follows a similar line to the proofs of Theorems 4.2 and 4.3 for mutual authentication, and Theorem 4.4 for narrow destructive privacy. Remark that for privacy, the *Result* oracle is not used. □

	Reader (DB, F)		**Tag** (P, s, F, x)
1		$\xleftarrow{z}$	$K = P(s)$, $z = F_K(0,0,x)$ erase K, z $x = x + 1$
2	If $\exists(ID, K, x) \in DB$ and $0 \leq i < t$ s.t. $z = F_K(0,0,x+i)$ then $x = x + i$ $w = F_K(0,1,x+1)$ else $w \leftarrow \{0,1\}^{\ell_2(\lambda)}$	$\xrightarrow{w}$	
3		$\xleftarrow{w'}$	$K = P(s)$ If $w \neq F_K(0,1,x)$ then $w' = F_K(1,1,x)$ else $w' = F_K(1,0,x)$ erase K, w, w'
	If $w' = F_K(1,1,x+1)$ then output ID, $x = x + 1$ else output $\perp$		

Figure 2. Narrow destructive private and reader-first authentication

It is good to remark that our RFID scheme in Figure 2 also provides an appropriate practical solution to the narrow destructive privacy in the plain Vaudenay's model, where the existing solution is based on random oracles [1], [2] .

A few more words on desynchronization are in order. If we look to the protocol in Figure 2, we remark that the desynchronization is a result of the fact that the tag and reader share a common variable x that is updated by tag before authenticating the reader. This allows an adversary to query a tag for more than t times and, therefore, to desynchronize the tag and the reader.

To prevent desynchronization between reader and tag in reader-first authentication RFID schemes, the tag should update the shared permanent variables after authenticating the reader, and not before.

6 Conclusions

Modern applications of RFID systems ask for advanced security and privacy properties. For instance, tag destruction under corruption is an important requirement when the tag is used for access control. Likewise, the disclosure of temporary state under tag corruption is a serious threat in practice. Reader-first authentication [3] assures that the tag will give its private data only when it authenticates the reader. Therefore, tag tracking and data theft is prevented when the reader is fake. All these together mean that we need RFID schemes that provide destructive privacy and reader-first authentication under corruption with temporary state disclosure.

The aim of this paper is to propose two RFID schemes that fill this gap. The first one is destructive private and the second one is narrow destructive private. Both of them assure reader-first authentication, are practical, and efficient. Also, both schemes avoid random number generators on tags. As (narrow) destructive privacy cannot be achieved with ordinary tags, we have used PUFs as secure hardware containers for the secret key of tags. Detailed security and privacy proofs are provided for our schemes.

References

[1] S. Vaudenay, “On privacy models for RFID,” in *Proceedings of the Advances in Crypotology 13th International Conference on Theory and Application of Cryptology and Information Security*, ser. ASIACRYPT’07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 68–87.

[2] R.-I. Paise and S. Vaudenay, “Mutual authentication in RFID: Security and privacy,” in *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS ’08. New York, USA: ACM, 2008, pp. 292–299.

[3] R. Peeters, J. Hermans, and J. Fan, “IBIHOP: Proper privacy preserving mutual RFID authentication,” in *The 2013 Workshop on Radio Frequency Identification/Internet of Things Security (RFIDsec’13 Asia)*, ser. Cryptology and Information Security Series, C. Ma and J. Weng, Eds., vol. 11. IOS Press, 2013.

[4] C. Hristea and F. L. Ţiplea, "Destructive privacy and mutual authentication in Vaudenay's RFID model," Cryptology ePrint Archive, Report 2019/073, 2019, https://eprint.iacr.org/2019/073.

[5] S. Kardaş, S. Çelik, M. Yildiz, and A. Levi, "PUF-enhanced offline RFID security and privacy," *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 2059–2067, Nov. 2012.

[6] M. Akgün and M. U. Çaglayan, "Providing destructive privacy and scalability in RFID systems using PUFs," *Ad Hoc Netw.*, vol. 32, no. C, pp. 32–42, Sep. 2015.

[7] F. Armknecht, A.-R. Sadeghi, A. Scafuro, I. Visconti, and C. Wachsmann, "Impossibility results for RFID privacy notions," in *Transactions on Computational Science XI*, M. L. Gavrilova, C. J. K. Tan, and E. D. Moreno, Eds. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 39–63.

[8] A.-R. Sadeghi, I. Visconti, and C. Wachsmann, "PUF-enhanced RFID security and privacy," in *Workshop on secure component and system identification (SECSI)*, vol. 110, 2010.

[9] ——, *Enhancing RFID Security and Privacy by Physically Unclonable Functions.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 281–305.

[10] M. Sipser, *Introduction to the Theory of Computation.* Cengage Learning, 2012.

[11] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 3rd ed. Wiley Publishing, 2010.

[12] Y. Li, H. R. Deng, and E. Bertino, *RFID Security and Privacy*, ser. Synthesis Lectures on Information Security, Privacy, and Trust. Morgan & Claypool Publishers, 2013.

[13] A. Juels and S. A. Weis, "Defining strong privacy for RFID," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 1, pp. 7:1–7:23, Nov. 2009.

[14] S. Canard, I. Coisel, J. Etrog, and M. Girault, "Privacy-preserving RFID systems: Model and constructions," https://eprint.iacr.org/2010/405.pdf, 2010.

[15] R. H. Deng, Y. Li, M. Yung, and Y. Zhao, "A new framework for RFID privacy," in *Proceedings of the 15th European Conference*

on Research in Computer Security, ser. ESORICS'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 1–18.

[16] J.-M. Bohli and A. Pashalidis, "Relations among privacy notions," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 4:1–4:24, Jun. 2011.

[17] J. Hermans, F. Pashalidis, Andreasand Vercauteren, and B. Preneel, "A new RFID privacy model," in *Computer Security – ESORICS 2011*, V. Atluri and C. Diaz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 568–587.

[18] J. Hermans, R. Peeters, and B. Preneel, "Proper RFID privacy: Model and protocols," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2888–2902, Dec 2014.

[19] V. Shoup, "Sequences of games: A tool for taming complexity in security proofs," 2004.

Ferucio Laurenţiu Ţiplea, Cristian Hristea,
Rodica Bulai

Received April 11, 2022
Revised April 28, 2022
Accepted April 28, 2022

Ferucio Laurenţiu Ţiplea
Department of Computer Science
"Alexandru Ioan Cuza" University of Iaşi
Iaşi, Romania
E–mail: `ferucio.tiplea@uaic.ro`

Cristian Hristea
Simion Stoilow Institute of Mathematics of the Romanian Academy
Bucharest, Romania
E–mail: `cristi.hristea@gmail.com`

Rodica Bulai
Faculty of Computers, Informatics and Microelectronics
Technical University of Moldova
Chisinau, Republic of Moldova
E–mail: `rodica.bulai@ati.utm.md`