Destructive Privacy and Mutual Authentication in Vaudenay's RFID Model

Cristian Hristea and Ferucio Laurenţiu Ţiplea

Abstract

With the large scale adoption of the Radio Frequency Identification (RFID) technology, a variety of security and privacy risks need to be addressed. Arguably, the most general and used RFID security and privacy model is the one proposed by Vaudenay. It considers concurrency, corruption (with or without destruction) of tags, and the possibility to get the result of a protocol session on the reader side. Security in Vaudenay's model embraces two forms, unilateral (tag) authentication and mutual (tag and reader) authentication, while privacy is very flexible and dependent on the adversary class. The construction of destructive private RFID schemes in Vaudenay's model was left open when the model was initially proposed. It was solved three years later in the context of unilateral authentication.

In this paper we propose a destructive private and mutual authentication RFID scheme in Vaudenay's model. The security and privacy of our scheme are rigorously proved. We also show that the only two RFID schemes proposed so far that claimed to achieve destructive privacy and mutual authentication are not even narrow forward private. Thus, our RIFD scheme is the first one to achieve this kind of privacy and security. The paper also points out some privacy proof flaws that have been met in previous constructions.

Index Terms

RFID scheme, authentication, privacy.

I. Introduction

RADIO Frequency Identification (RFID) technology [1], [2] enables wireless identification of objects or persons in a wide range of applications: access control, logistics and supply chain visibility and management, (item level inventory, tool, attendee, IT assets) tracking, kiosks, library systems, interactive marketing, real-time location systems, national IDs management, patient care management, and so on.

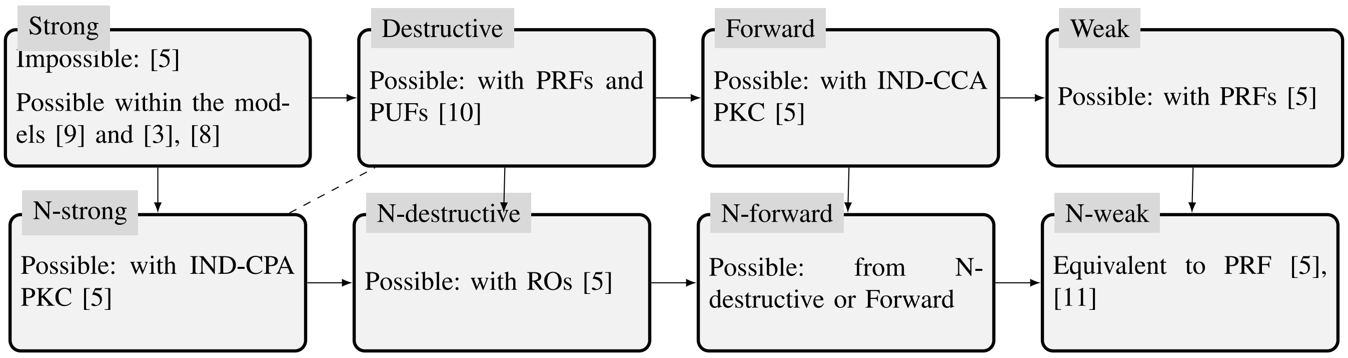
The key component of this technology is a device called *RFID tag*, or simply *tag*, which is a cheap resource constrained device capable of identifying the object to which it is attached. An *RFID tag reader*, or simply *reader*, emits a low-level radio frequency magnetic field that energizes the tag. Then, the tag responds to the reader's query, transmitting its unique identification data. This data is processed by the reader and passed then to the local application system. In the identification process, the reader has secure access to a *back-end database* that stores records associated with the tags.

With the adoption of RFID technology, a variety of security and privacy risks need to be addressed: unauthorized access and modification of tag data, eavesdropping, traffic analysis, spoofing, or denial of service. Thus, we arrive to the need for communication protocols between reader and tags, capable to assure the security and privacy of data and users, whilst remaining lightweight enough to fit on tags. Alongside the development of such protocols, there are several works [3]–[8] that have contributed to the development of a formal framework for the evaluation of RFID protocols.

One of the most influential RFID security and privacy model was proposed by Vaudenay [5], and was extended later in [6] to cover reader authentication. We recall this model here very briefly to create a clear overview of the current state-of-the-art and to clearly explain our contribution.

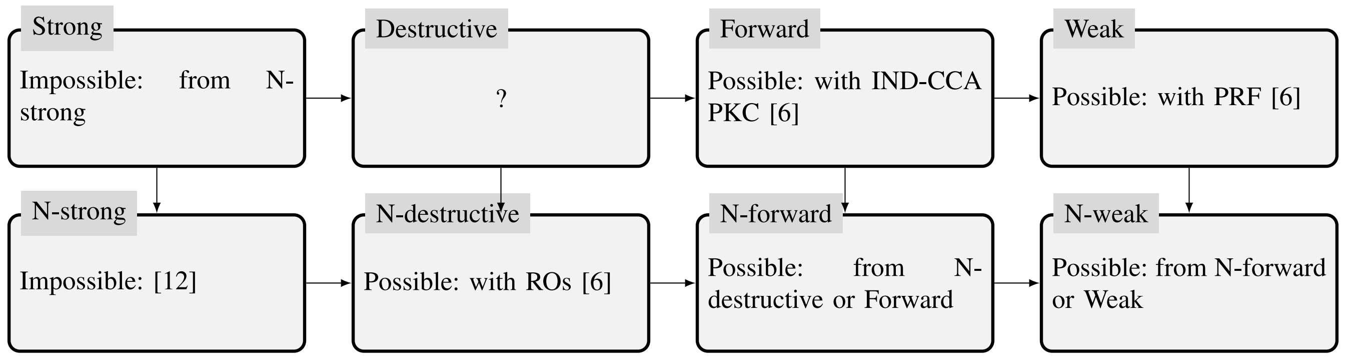
According to Vaudenay's model [5], [6], an adversary is a probabilistic polynomial-time algorithm that can

C. Hristea and F.L. Ţiplea are with the Department of Computer Science, Alexandru Ioan Cuza University of Iaşi, Iaşi, Romania, e-mail: cristi.hristea@gmail.com and ferucio.tiplea@uaic.ro



Strong
Impossible: [5]
Possible within the models [9] and [3], [8]
Destructive
Possible: with PRFs and PUFs [10]
Forward
Possible: with IND-CCA PKC [5]
Weak
Possible: with PRFs [5]
N-strong
Possible: with IND-CPA PKC [5]
N-destructive
Possible: with ROs [5]
N-forward
Possible: from N-destructive or Forward
N-weak
Equivalent to PRF [5], [11]

Fig. 1. Privacy and unilateral authentication in Vaudenay's model



Strong
Impossible: from N-strong
Destructive
?
Forward
Possible: with IND-CCA PKC [6]
Weak
Possible: with PRF [6]
N-strong
Impossible: [12]
N-destructive
Possible: with ROs [6]
N-forward
Possible: from N-destructive or Forward
N-weak
Possible: from N-forward or Weak

Fig. 2. Privacy and mutual authentication in Vaudenay's model

- request creation of legitimate or illegitimate tags;
- draw one or more anonymous tags with respect to some chosen probability;
- release drawn tags;
- launch protocol instances between reader and drawn tags;
- send messages to drawn tags;
- send messages to the reader;
- corrupt any drawn tag to retrieve its internal state;
- get the result of a completed protocol instance.

An adversary with all the above capabilities is a *strong adversary*; an adversary that destroys a tag after corrupting it is a *destructive adversary*; an adversary that can corrupt tags only at the end is a *forward adversary*; finally, an adversary that cannot corrupt tags is a *weak adversary*. Orthogonal to these classes of adversaries is the class of *narrow adversaries* that cannot see the result of completed protocol instances. Narrow adversaries with the capabilities discussed above give rise to *narrow strong*, *narrow destructive*, *narrow forward*, and *narrow weak* adversaries. We have thus obtained eight classes of adversaries.

It turns out that the analysis of RFID security and privacy not only depends on the adversary type, but also on the authentication type: *unilateral authentication* (tag is authenticated to the reader) or *mutual authentication* (tag is authenticated to the reader and the reader is authenticated to the tag, or vice versa).

The diagram in Figure 1 shows the relationship between the eight privacy concepts in Vaudenay's model in the context of unilateral authentication. In this diagram, "N-x" is a shortcut for "narrow x". An arrow from A to B means that A-privacy implies B-privacy. The dashed line means that the narrow strong and destructive privacy concepts are not possible at the same time [5]. The diagram also includes supplementary information about each class. We will further discuss on them.

Weak privacy can be achieved with *pseudo-random functions* (PRFs) [5]. It can be shown that any narrow weak private (and correct) RFID scheme can be transformed into a one-way function [11]. Therefore, narrow weak privacy is equivalent to the PRF property. Forward and narrow strong privacy are possible with *public-key cryptography* (PKC) [5]. The narrow-destructive privacy can be achieved with *random oracles* (ROs) [5], while destructive privacy can be achieved with PRFs and *physically unclonable functions* (PUFs) [10], [13]. Finally, strong privacy is impossible in Vaudenay's model [5].

The diagram in Figure 2 shows the relationship between the eight privacy concepts in Vaudenay's model in the context of mutual authentication. The results for narrow destructive, forward, and weak privacy are obtained by using extensions of the corresponding techniques in Figure 1 (please see [6]). The impossibility results have been obtained in [12], where some wrong statements from [6] have also been corrected.

The diagrams in Figure 1 and 2 list some of the most fundamental constructions of RFID schemes for the corresponding privacy classes; it is outside their scope to provide an exhaustive list of references.

a) Contribution: As one can see from our diagram in Figure 2, the existence of destructive private and mutual authentication RFID schemes is marked as an open problem. In fact, [14] proposes an RFID scheme and claims that it achieves destructive privacy and mutual authentication in Vaudenay's model with temporary state disclosure (this model is stronger than Vaudenay's model). However, our analysis in Section V shows that this scheme is not even narrow forward private in the proposed model. Another RFID scheme, proposed in [15], claims that it achieves narrow destructive privacy and unilateral authentication in Vaudenay's model with temporary state disclosure. Looking carefully at the scheme, one may say that it provides reader authentication as well (and, therefore, mutual authentication). However, the authors do not prove anything about this. Moreover, our analysis in Section V shows that this scheme has the same fate as the previous one: it is not even narrow forward private in the proposed model. For the time being, it remains to see if the two schemes, in their present form, achieve what they claim in Vaudenay's model (please see our Section V for more details).

Based on the discussion above, we address the problem of constructing an RFID scheme that achieves destructive privacy and mutual authentication in Vaudenay's model. The solution we propose is based on PRFs and PUFs and follows the line in [6]. We think this is important because the PRF based RFID scheme in [5] is a fundamental construction that provides weak privacy. Adding PUFs to this construction, we get another fundamental construction that provides destructive privacy and unilateral authentication. By extending this construction to achieve mutual authentication we may think that we have another fundamental construction, similar to those in [6] (for mutual authentication).

Summing up, the main contributions of the paper are as follows:

1) We propose a destructive private and mutual authentication RFID scheme in Vaudenay's model, as a natural extension of the PRF and PUF based scheme in [10], [13] or of the PRF based scheme in [6] (Section IV);
2) We provide very rigorous security and privacy proofs to our scheme (Section IV);
3) We show that the privacy proofs in [10], [13] are flawed, but they can be fixed (Section V);
4) We show that the RFID schemes in [14] and [15] do not achieve (narrow) destructive privacy in Vaudenay's model with temporary state disclosure, as they claim; in fact, they do not even achieve narrow forward privacy (Section V);
5) We make a comprehensive discussion on Vaudenay's model (Section III). This might be thought as a minor contribution; however, it is important for a deep understanding of the model and the security and privacy proofs.

b) 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 Section 2 and 3. Our RFID scheme, that achieves destructive privacy and mutual authentication in Vaudenay's model, is presented in Section 4, together with its security and privacy analysis. The fifth section points out some proof and design flaws in destructive private RFID schemes. The last section concludes the paper.

II. Basic definitions and notation

Throughout this paper we use *probabilistic polynomial time* (PPT) algorithms $\mathcal{A}$ as defined in [16]. 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 $\lambda \geq n_0$. $f(\lambda)$ is called *overwhelming* if $1 - f(\lambda)$ is negligible.

a) Pseudo-random functions: A *pseudo-random function* (PRF) is a family of functions with the property that if we randomly choose a function from this family then its input-output behavior is computationally indistinguishable from that of a random function. To be more precise, consider and fix two polynomials ℓ_1 and ℓ_2 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 functions of λ).

$F = (F_K)_{K \in \mathcal{K}}$ is called a *pseudo-random function* if it is:

1) *Efficiently computable* : there exists a deterministic polynomial-time algorithm that on input λ, $K \in \mathcal{K}_\lambda$, and $x \in \{0,1\}^{\ell_1(\lambda)}$, returns $f(x)$;
2) *Pseudo-random* : F is computationally indistinguishable from U.

To prove that F is a PRF, we usually assume the existence of a *challenger* $\mathcal{C}$ that plays the following security game, parameterized by a security parameter λ, with an adversary $\mathcal{A}$:

1) $\mathcal{C}$ randomly chooses $b \leftarrow \{0,1\}$;
2) if $b = 1$ then $\mathcal{C}$ randomly chooses $K \leftarrow \mathcal{K}_\lambda$ and sets $f = F_K$; otherwise, $\mathcal{C}$ randomly chooses $f \leftarrow U_\lambda$;
3) $\mathcal{C}$ provides oracle access to f for $\mathcal{A}$;
4) At some point, $\mathcal{A}$ outputs a bit b'.

The adversary $\mathcal{A}$ wins the game if $b' = b$.

Now, one can see that F is a PRF if it is efficiently computable and the probability to win the above security game is negligible, for all adversaries.

b) Physically unclonable function: 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).

Unfortunately, PUFs are subject to noise induced by the operating conditions, such as supply voltage or ambient temperature. Therefore, PUFs return slightly different responses when queried with the same challenge multiple times. However, from a theoretical point of view it is assumed that PUFs return a similar response when queried with the same challenge multiple times (this is usually called robustness).

Based on these, we adopt here the concept of an *ideal PUF* slightly different than in [10]. Namely, an *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:

1) P is computationally indistinguishable from U;
2) Any attempt to physically tamper with the object implementing P results in destruction of P (P cannot be evaluated any more).

III. RFID schemes and systems

An RFID system [1], [2] consists of a *reader*, a set of *tags*, and a *communication protocol* between reader and tags. A reader is a powerful transceiver[1] with an associated 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 other incoming communication. The reader and its database are trusted entities, and the communication between them is secure. A tag is a transponder[2] 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.

a) RFID schemes: From a formal point of view, the reader and the tags are PPT algorithms. To work with them we need to assign identifiers, initialize, and define a communication protocol between them. Therefore, given a security parameter λ, we consider a *reader identifier* $\mathcal{R}$ and a set $\mathcal{T}$ of *tag identifiers* whose cardinal is polynomial in λ.

Now, we define an *RFID scheme over* $(\mathcal{R},\mathcal{T})$ [5], [6] as a triple $\mathcal{S} = (SetupR,\ SetupT,\ Ident)$ of PPT algorithms, where:

1) $SetupR(\lambda)$ initializes the reader. It inputs a security parameter λ and outputs a triple (pk, sk, DB) consisting of 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 meaning of $SetupR(\lambda)$ is that it "creates" a reader identified by $\mathcal{R}$ and initializes it (and also establishes some public parameters of the system). We simply refer to the reader such created as being $\mathcal{R}$. The meaning of $SetupT(pk, ID)$ is that it "creates" a tag identified by ID, initializes it with an initial tag state, and also register this tag with the reader by storing some information about it in the reader's database. We denote this tag by $\mathcal{T}_{ID}$.

In the definition above, $Ident$ is an *authentication* protocol. It is assumed that the reader may be involved in concurrent runnings of $Ident$, but the tags cannot (that is, no tag can be involved in two or more runnings of $Ident$ at the same time). When the reader outputs ID, it means that it authenticated (identified) the tag $\mathcal{T}_{ID}$; $\perp$ means that the reader rejects the tag.

The $Ident$ protocol above allows the reader to authenticate or not the tag. That is, it allows *unilateral authentication*. There are also cases where *mutual authentication* is needed. A *mutual authentication RFID scheme* is defined as above except for the $Ident$ protocol that has to be replaced by:

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 $\perp$, and the tag ends with an output consisting of OK or $\perp$.

The meaning of the tag's output is that it outputs OK when authenticates the reader, and $\perp$, otherwise.

We emphasize that the protocol $Ident$ in case of mutual authentication does not specify whether the reader or tag authentication goes first.

[1]Contraction from transmitter and receiver.

[2]Contraction from transmitter and responder.

For the sake of simplicity we will refer to mutual authentication RFID schemes as being RFID schemes; the difference will follow from the context.

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 RFID schemes, *correctness* means that the reader outputs tag's identity and the tag outputs OK with overwhelming probability.

We draw the attention to the fact that the correctness concept for mutual authentication RFID schemes as defined in [6] is somewhat ambiguous or incomplete. The authors say "The protocol is correct if executing it honestly with a legitimate tag, it outputs OK, except with negligible probability." This might cover tag authentication too in a tag-first authentication approach (i.e., the reader authenticates the tag, then the tag authenticates the reader and, at the end, the tag outputs OK – the protocols proposed in [6] follow this line).

b) RFID system: An *RFID system* is an instantiation of an RFID scheme. This is done by a trusted *operator* $\mathcal{I}$ who establishes the reader identifier $\mathcal{R}$, the set $\mathcal{T}$ of tag identifiers, and runs an RFID scheme over $(\mathcal{R}, \mathcal{T})$. In a given setting, the reader is initialized exactly once, while each tag at most once. Thus, the reader's database does not store different entries for the same tag. However, different settings with the same RFID scheme may initialize the reader and the tags in different ways.

c) Adversaries: The interactive protocol $Ident$ of an RFID scheme defines the communication between reader and tags. This protocol also provides a way for attackers to interact with the RFID system components. Therefore, the RFID protocols must be able to thwart attacks that violate the security of an RFID system. The two most basic security requirements for RFID protocols are *authentication* and *untraceability*. Informally, authentication ensures that the reader and the tags can not be impersonated. Untraceability ensures that an attacker can not recognize the tags it has observed or communicated with in the past.

In order to formalize these security requirements, the concept of an *adversary model* is needed. Such a model defines the capabilities of an adversary by means of a set of oracles that simulate the interaction with the RFID system. There have been proposed several adversary models in the literature on RFID, such as [3]–[6], [8], [17]–[19]. One of the most influential, which we follow in this paper, is Vaudenay's model [5], [6]. We comprehensively recall this model, with some very small presentation changes too, to make the exposure as clear as possible. 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 the adversary may query are:

1) $CreateTag^b(ID)$: Creates a free tag $\mathcal{T}_{ID}$ with the identifier ID by calling $SetupT(pk, ID)$ 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 n of free tags according to the distribution δ, 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.
   As one can see, $DrawTag$ provides the adversary with access to some free tags by means of temporary identifiers, and gives information on whether the tags are legitimate or not (but no other information);

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. This is a natural assumption that corresponds to the fact that the tag is no longer powered by reader;
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 takes the first step in the protocol.
   We emphasize that the reader's answer is conceived as the message sent to the tag by the communication channel and not as the reader's decision output (tag identity or $\perp$). Therefore, if the reader does not send anything to the tag, the output of this oracle is empty;
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 takes the first step in the protocol.
   As in the case of the $SendReader$ oracle, we emphasize that the tag's answer is conceived as the message sent to the reader by the communication channel and not as the tag's decision output (OK or $\perp$). Therefore, if the tag does not send anything to the reader, the output of this oracle is empty;
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 – please see below for an extended discussion on this oracle);
8) $Corrupt(vtag)$: Outputs the current permanent (internal) state of the tag referred to by $vtag$ (please see below for an extended discussion on this oracle).

The $Result(\pi)$ oracle tries to capture the reader's decision output on tag authentication. However, there is somewhat ambiguity and non-uniformity in defining it. Some authors [12] ask for a complete session π when defining the output 1 of $Result(\pi)$. Our point of view is that the reader's job is completed when it authenticates the tag, no matter of the tag's decision (which might be unknown to the reader) or if the protocol session π is completed or not. For instance, if the reader of an access control system authenticates the RFID card (tag), then the access is allowed. The definition in [6] is somewhat ambiguous: "$Result(\pi)$ to get 0 if the output on session π is $\perp$ and 1 otherwise." This leaves the possibility for $Result(\pi)$ to return 1 when the session is incomplete or when it does not even exist. The approach in [8], which is for unilateral authentication, gives the possibility to return $\perp$ when the reader has not yet made a decision on tag authentication (this is what we have also adopted for mutual authentication).

The $Corrupt$ oracle has been the subject of intense discussions over time. To understand it well, let us note that each tag has a *permanent* (or *internal*) *state* that stores the tag's state values, and a *temporary* (or *volatile*) *state* that can be viewed as a set of *volatile variables* used to carry out the necessary computations. When Vaudenay's model was proposed [5], it was left unclear whether the $Corrupt$ oracle returns the *full state* of the tag (permanent and temporary state) or only the permanent state. This was exploited in [12] where it was shown that there is a huge difference, with respect to privacy, between the two variants. Later, Ouafi and Vaudenay [9] specified clearly that in Vaudenay's model the $Corrupt$ oracle returns only the current permanent state of the tag. 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*: if they access the $Corrupt$ oracle, then they can only access the $Corrupt$ oracle;

- *Destructive adversaries*: if the adversary queried $Corrupt(vtag)$, then no oracle for $vtag$ can further be queried (that is, the tag identified by $vtag$ is destroyed);
- *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 property with any of the previous properties in order to get another four classes of adversaries, *narrow weak*, *narrow forward*, *narrow destructive*, and *narrow strong*.

d) Security: Now we are ready to introduce the *tag* and *reader authentication* properties as proposed in [5], [6], 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 the following experiment that a challenger sets up for an adversary $\mathcal{A}$ (after the security parameter λ is fixed):

```
Experiment $RFID^{t\_auth}_{\mathcal{A},\mathcal{S}}(\lambda)$
1: Set up the reader;
2: $\mathcal{A}$ gets the public key $pk$;
3: $\mathcal{A}$ queries the oracles;
4: Return 1 if there is a protocol instance $\pi$ s.t.:
     – $\pi$ identifies an uncorrupted legitimate tag $ID$;
     – $\pi$ had no tag authentication matching conversation with $\mathcal{T}_{ID}$ or any drawn form of it.
   Otherwise, return 0.
```

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

$$Adv^{t\_auth}_{\mathcal{A},\mathcal{S}}(\lambda) = Pr(RFID^{t\_auth}_{\mathcal{A},\mathcal{S}}(\lambda) = 1)$$

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

The experiment for reader authentication, denoted $RFID^{r\_auth}_{\mathcal{A},\mathcal{S}}(\lambda)$, is quite similar to that above:

```
Experiment $RFID^{r\_auth}_{\mathcal{A},\mathcal{S}}(\lambda)$
1: Set up the reader;
2: $\mathcal{A}$ gets the public key $pk$;
3: $\mathcal{A}$ queries the oracles;
4: Return 1 if there is a protocol instance $(\pi, ID)$ s.t.:
     – $\mathcal{T}_{ID}$ is a uncorrupted legitimate tag that identifies the reader;
     – $\pi$ had no reader authentication matching conversation with $\mathcal{T}_{ID}$.
   Otherwise, return 0.
```

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 $\mathcal{A}$, $Adv^{r\_auth}_{\mathcal{A},\mathcal{S}}$, is negligible, for any strong adversary $\mathcal{A}$ (the advantage of $\mathcal{A}$ is defined as above, by using $RFID^{r\_auth}_{\mathcal{A},\mathcal{S}}(\lambda)$ instead of $RFID^{t\_auth}_{\mathcal{A},\mathcal{S}}(\lambda)$).

e) Privacy: Privacy for RFID systems [6] 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 [6].

A *blinder for an adversary* $\mathcal{A}$ that belongs to some class V of adversaries is a PPT algorithm $\mathcal{B}$ that:

1) simulates the $Launch$, $SendReader$, $SendTag$, and $Result$ oracles for $\mathcal{A}$, without having access to the corresponding secrets;
2) passively looks 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 $\mathcal{A}^{\mathcal{B}}$. We emphasize that $\mathcal{A}^{\mathcal{B}}$ is allowed to query the oracles $Launch$, $SendReader$, $SendTag$, and $Result$ only by means of $\mathcal{B}$; all the other oracles are queried as a standard adversary.

Given an adversary $\mathcal{A}$ and a blinder $\mathcal{B}$ for it, define the following two experiments:

```
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 1 if $b' = 0$, and 0, otherwise.
```

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

Now, the *advantage* of $\mathcal{A}$ blinded by $\mathcal{B}$ is defined by

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

The advantage of $\mathcal{A}$ acts like an indirect measure of the information leaked by the RFID scheme to the adversary $\mathcal{A}$: the more information is leaked, the more likely $\mathcal{A}$ will distinguish between the two worlds (cases).

An RFID scheme achieves privacy for a class V of adversaries if for any adversary $\mathcal{A} \in V$ 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.

IV. Destructive privacy and mutual authentication

As one can see from our Figure 2, no destructive private and mutual authentication protocol for RFID scheme has been proposed so far. Following a design line as in [6], we can transform the scheme in [10], [13] into a destructive private and mutual authentication RFID scheme. We emphasize that the scheme in [10], [13] is obtained from the weak private PRF-based authentication RFID scheme in [5], by endowing each tag with an internal PUF to generate the tag's PRF key.

Before describing our RFID scheme we have to say that Vaudenay's model can be easily adapted to tags that are endowed with PUFs. The only thing we have to do is to clarify how the $Corrupt$ oracle acts on such tags. As PUFs are tamper-evident, it is natural to assume that the $Corrupt$ oracle returns the permanent state of the tag but no value computed by its internal PUFs, except when such values are stored in the permanent memory of the tag.

	Reader (DB)		**Tag** (P, s)
1	$x \leftarrow \{0,1\}^{\ell_1(\lambda)}$	$\xrightarrow{x}$	
2		$\xleftarrow{y,\ z}$	$y \leftarrow \{0,1\}^{\ell_1(\lambda)}$, $K = P(s)$ $z = F_K(0, x, y)$, erase K
3	If $\exists(ID, K) \in DB$ s.t. $z = F_K(0, x, y)$ then output ID (tag auth.) else output $\perp$; $K \leftarrow \mathcal{K}_\lambda$; $w = F_K(1, x, y)$	$\xrightarrow{w}$	
			$K = P(s)$, $w' = F_K(1, x, y)$, erase K If $w = w'$ then output OK (reader auth.) else output $\perp$

Fig. 3. Destructive private and mutual authentication PUF-based RFID scheme

To describe the RFID scheme that we propose, 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 pseudo-random function, where $F_K : \{0,1\}^{2\ell_1(\lambda)+1} \to \{0,1\}^{\ell_2(\lambda)}$ 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 string $s \in \{0,1\}^{p(\lambda)}$ randomly chosen as a seed to evaluate P.

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.

The mutual authentication protocol is given in Figure 3. As we can see, the reader sends initially a random $x \leftarrow \{0,1\}^{\ell_1(\lambda)}$ to the tag. On receiving it, the tag generates a random $y \leftarrow \{0,1\}^{\ell_1(\lambda)}$, computes $K = P(s)$ and $z = F_K(0, x, y)$, erases K, and answers with (y, z). The reader checks its database for a pair (ID, K) such that $z = F_K(0, x, y)$. If such a pair is found, it outputs ID; otherwise, outputs $\perp$ and randomly chooses $K \in \mathcal{K}_\lambda$. No matter of the two cases (K is found in the database or is randomly generated), the reader computes $w = F_K(1, x, y)$ and sends it to the tag. On receiving it, the tag computes $P(s)$ and $w' = F_{P(s)}(1, x, y)$. Finally, it outputs OK or $\perp$ depending on the equality $w = w'$.

Theorem 4.1: The RFID scheme in Figure 3 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 is the tag's state, and P is its PUF.

When the reader receives (y, z) from the tag $\mathcal{T}_{ID}$, the equality $z = F_{K'}(0, x, y)$ holds with negligible probability if $K' \neq K$. Therefore, the reader authenticates the tag with overwhelming probability.

A similar reasoning shows that the tag authenticates the reader with overwhelming probability.

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

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

Theorem 4.2: The protocol in Figure 3 achieves tag authentication in Vaudenay's model, provided that F is a PRF and the tags are endowed with ideal PUFs.

Proof: Assume that the protocol does not achieve tag authentication, and let $\mathcal{A}$ be an adversary that has non-negligible advantage over the protocol, with respect to the tag 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 the next one. Let $\mathcal{C}$ be a challenger for the pseudo-randomness security game of the function F. The adversary $\mathcal{A}'$ will play the role of challenger for $\mathcal{A}$. Thus, $\mathcal{A}'$ guesses the tag identity ID^* that $\mathcal{A}$ can authenticate with the reader with 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 $\mathcal{C}$. Namely, the random key chosen by $\mathcal{C}$ will be thought as the key generated by the tag's PUF. The adversary $\mathcal{A}'$ does not know this key but, in fact, it does not need to. As $\mathcal{A}'$ impersonates the reader, it can provide $\mathcal{A}$ with correct answers by querying $\mathcal{C}$. Therefore, $\mathcal{T}_{ID^*}$ will be regarded by $\mathcal{A}$ as a legitimate tag.

When $\mathcal{A}$ succeeds to authenticate $\mathcal{T}_{ID^*}$ to the reader with non-negligible probability, $\mathcal{A}'$ will use the information obtained from $\mathcal{A}$ to answer correctly, with overwhelming probability, some challenge of $\mathcal{C}$.

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

1) The challenger $\mathcal{C}$ chooses uniformly at random a key for F and will answer all queries of $\mathcal{A}'$ with respect to this key. Recall that a query for F_K is of the form $\alpha \in \{0,1\}^{2\ell_1(\lambda)+1}$, and the answer provided by $\mathcal{C}$ is of the form $\beta = F_K(\alpha)$, provided that K is the key chosen by $\mathcal{C}$;
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 III) 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 (recall that the number of tag identities is polynomial in the security parameter);
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 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 $(ID^*, ?)$ into $\mathcal{A}'_{ListTags}$. The meaning of "?" is that this field should have contained the key chosen by $\mathcal{C}$, which is unknown to $\mathcal{A}'$. However, $\mathcal{A}'$ does not need to know this key 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 key 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) $SendTag(x, vtag)$: If the tag referred by $vtag$ is ID^*, then $\mathcal{A}'$ will randomly generate $y \in \{0,1\}^{\ell_1(\lambda)}$ and query $\mathcal{C}$ for $(0, x, y)$. If z is the $\mathcal{C}$'s response, than $\mathcal{A}'$ answers with (y, z).
      If $vtag$ refers to some $ID \neq ID^*$, then $\mathcal{A}'$ can prepare by itself the answer because it knows the corresponding key for ID;
   e) $SendReader((y,z), \pi)$: Assume the reader (run by $\mathcal{A}'$) has sent x in the protocol instance π to a tag identified by $vtag$.
      If $vtag$ refers to some tag ID such that $(ID, K) \in DB$ for some K, then compute $z' = F_K(0, x, y)$ and $w = F_K(1, x, y)$, and output ID or $\perp$ according to whether $z = z'$ or not.
      If $vtag$ refers to ID^*, then compute z' and w (as above) by querying $\mathcal{C}$, and output ID^* or $\perp$ according to whether $z = z'$ or not (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 randomly generate a key K and compute w as above.
In all cases, the oracle returns w;

f) $SendTag(w, vtag)$: If the tag referred by $vtag$ is ID^*, then $\mathcal{A}'$ queries $\mathcal{C}$ for $(1, x, y)$ and then compares the answer with w. If they match, the tag outputs OK; otherwise, it outputs $\perp$.
If $vtag$ refers to some $ID \neq ID^*$ that has associated a key K, then $\mathcal{A}'$ computes $w' = F_K(1, x, y)$ and compares w' and w. Accordingly, it outputs OK or $\perp$;

g) $Result(\pi)$: $\mathcal{A}'$ can infer the decision of the reader because it can obtain the value $F_K(0, x, y)$ for all tags (either it can compute it or query $\mathcal{C}$ for it). Therefore, $\mathcal{A}'$ can simulate $Result(\pi)$ 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;

5) If $\mathcal{A}$ is able to make the reader to authenticate the tag ID^*, then this means that $\mathcal{A}$ can compute $z = F_{K^*}(0, x, y)$ without knowing K^*, provided that K^* is the key chosen by $\mathcal{C}$ (x is from the reader and y is randomly chosen). Then, $\mathcal{A}'$ can prepare the challenge phase for $\mathcal{C}$ as follows:
   a) $\mathcal{A}'$ sends $(0, x, y)$ to $\mathcal{C}$;
   b) $\mathcal{C}$ randomly chooses $b \in \{0, 1\}$; if $b = 1$, then $\mathcal{C}$ returns $z' = F_{K^*}(0, x, y)$, else $\mathcal{C}$ returns a random z';
   c) $\mathcal{A}'$ prepares its guess b' as follows: if $z = z'$, then $b' = 1$, else $b' = 0$.

The probability that $\mathcal{A}'$ guesses the bit chosen by $\mathcal{C}$ can be computed as the product between the probability that $\mathcal{A}'$ guesses ID^* and the probability that $\mathcal{A}$ makes the reader to authenticate the tag ID^*.

The probability that $\mathcal{A}'$ guesses ID^* is $1/t(\lambda)$, where $t(\lambda)$ is a polynomial that gives the number of tag identities. If we assume now that $\mathcal{A}$ has non-negligible probability to make the reader authenticate the tag ID^*, then $\mathcal{A}'$ can successfully answer the $\mathcal{C}$'s challenge with non-negligible probability; this contradicts the fact that F is a pseudo-random function. ■

Remark 4.1: We would like to emphasize that the tag $\mathcal{T}_{ID^*}$ in the proof of Theorem 4.1 is not a legitimate one, but it appears like a legitimate one to the adversary $\mathcal{A}$. Therefore, $\mathcal{A}$ does not make any difference between a real challenger for a RFID system and the challenger role played by $\mathcal{A}'$. This makes $\mathcal{A}$ behave like in a standard security game $RFID^{t\_auth}_{\mathcal{A},\mathcal{S}}(\lambda)$, where it has a non-negligible probability to authenticate $\mathcal{T}_{ID^*}$ to the reader.

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

Theorem 4.3: The protocol in Figure 3 achieves reader authentication in Vaudenay's model, provided that F is a PRF and the tags are endowed with ideal PUFs.

Proof: Assume that our protocol does not achieve reader authentication, and let $\mathcal{A}$ be an adversary that has non-negligible advantage over the protocol, 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 somehow 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 tag identity ID^* that authenticates $\mathcal{A}$ as a valid reader, with 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. When $\mathcal{A}$ succeeds in making $\mathcal{T}_{ID^*}$ to authenticate it as a valid reader, $\mathcal{A}'$ will use the message sent by $\mathcal{A}$ in order to answer some challenge of $\mathcal{C}$.

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 a key for F and will answer all queries of $\mathcal{A}'$ with respect to this key (a query for F_K is of the form $\alpha \in \{0, 1\}^{2\ell_1(\lambda)+1}$, and the answer provided by $\mathcal{C}$ is of the form $\beta = F_K(\alpha)$, provided that K is the key chosen by $\mathcal{C}$);

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 $\mathcal{A}'_{ListTags}$ 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;
5) If $\mathcal{A}$ is able to make $\mathcal{T}_{ID^*}$ to authenticate it as a valid reader, then this means that $\mathcal{A}$ can compute $w = F_{K^*}(1, x, y)$ without knowing K^* (provided that K^* is the key chosen by $\mathcal{C}$). Then, $\mathcal{A}'$ can prepare the challenge phase for $\mathcal{C}$ as follows:
   a) $\mathcal{A}'$ sends $(1, x, y)$ to $\mathcal{C}$;
   b) $\mathcal{C}$ randomly chooses $b \in \{0, 1\}$; if $b = 1$, then $\mathcal{C}$ returns $w' = F_{K^*}(1, x, y)$, else $\mathcal{C}$ returns a random w';
   c) $\mathcal{A}'$ prepares its guess b' as follows: if $w = w'$, then $b' = 1$, else $b' = 0$.

The probability that $\mathcal{A}'$ guesses the bit chosen by $\mathcal{C}$ is non-negligible if $\mathcal{A}$ has a non-negligible probability to make $\mathcal{T}_{ID^*}$ to authenticate it as a valid reader (this is similar to the proof of Theorem 4.2). Therefore, the assumption that $\mathcal{A}$ has a non-negligible probability to make $\mathcal{T}_{ID^*}$ to authenticate it as a valid reader contradicts the pseudo-randomness of F. ■

The final result of this section establishes the destructive privacy property of the protocol. The proof follows the *sequence-of-games* approach [20] by which a sequence of games (probabilistic experiments) is defined, the initial game being the original attack game with respect to a given adversary. In our case, the transition from one game G_i to another one G_{i+1} is done by indistinguishability. This means that a probability distribution in G_i is changed by another one that is indistinguishable from the previous one. In this way, the difference between the probability that the adversary wins G_i and the probability that the adversary wins G_{i+1}, is negligible.

Theorem 4.4: The protocol in Figure 3 achieves destructive privacy in Vaudenay's model, provided that F is a PRF and the tags are endowed with ideal PUFs.

Proof: We will show that for any destructive adversary A there exists a blinder $\mathcal{B}$ such that $Adv^{prv}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda)$ is negligible, where $\mathcal{S}$ denotes our protocol in Figure 3.

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;
- $SendReader(\emptyset, \pi)$: returns $x \leftarrow \{0,1\}^{\ell_1(\lambda)}$;
- $SendTag(x, vtag)$: returns $(y, z) \leftarrow \{0,1\}^{\ell_1(\lambda)} \times \{0,1\}^{\ell_2(\lambda)}$;
- $SendReader((y, z), \pi)$: returns $w \leftarrow \{0,1\}^{\ell_2(\lambda)}$;
- $SendTag(w, vtag)$: the blinder does not do anything because, in this case, the tag neither answers nor changes its internal state;
- $Result(\pi)$: if the session π does not exist or if only its first step was taken, the blinder outputs $\perp$. If π has been issued by the $Launch()$ oracle and a partial protocol transcript (i.e., the sequence of messages corresponding to a matching conversation) $tr_\pi = (x, (y, z))$ has been generated by
  - $x \leftarrow SendReader(\emptyset, \pi)$ and
  - $(y, z) \leftarrow SendTag(x, vtag)$,

  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 will show now 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 the 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 changed 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. We will provide below a proof sketch of this.

Assume $\mathcal{A}$ is an adversary that can distinguish between G_0 and G_1 with non-negligible probability. Define then a new adversary $\mathcal{A}'$ that can break the PUF security with non-negligible probability. In order to interact with the RFID system, the adversary $\mathcal{A}$ must create some tags. As the tags' PUFs, as well as their seeds, are independently at random chosen, we may assume, without loss of generality, that $\mathcal{A}$ creates exactly one tag with some identity ID, interacts with it, and draws the final conclusion based on this interaction.

Now, the proof goes in a way somewhat similar to the proof of Theorem 4.2. Assume that $\mathcal{C}$ is a challenger for some PUF P. $\mathcal{A}'$ will play the role of challenger for $\mathcal{A}$. When $\mathcal{A}$ queries $CreateTag$ to create the tag $\mathcal{T}_{ID}$, legitimate or not, $\mathcal{A}'$ chooses at random a state s for this tag and sends it to the challenger $\mathcal{C}$. The challenger chooses at random a bit $b \leftarrow \{0,1\}$ and answers with $K = P(s)$, if $b = 0$, or $K \leftarrow \{0,1\}^\lambda$, if $b = 1$. The adversary $\mathcal{A}'$ will then use K to create the tag $\mathcal{T}_{ID}$. It will also answer $\mathcal{A}$'s all other oracle queries (similar to the proof of Theorem 4.2).

Remark that $\mathcal{A}$ will play the game G_b, without knowing b. After some time, $\mathcal{A}$ will output a guess $b' \in \{0,1\}$ about the game it thinks it is playing. Then, $\mathcal{A}'$ can make a decision about the key K: it was computed as $P(s)$, if $b' = 0$, or it is randomly chosen, if $b' = 1$. Clearly, the probability the adversary $\mathcal{A}'$ wins the PUF security game is the probability that $\mathcal{A}$ distinguishes between the two worlds, G_0 and G_1. If this is non-negligible, then $\mathcal{A}'$ has non-negligible probability to break the PUF.

Game G_2: This game is obtained from G_1 by replacing the oracle $Result$ by a new oracle $Result_{\mathcal{B}}$ that behaves exactly as the blinder simulates $Result(\pi)$ (please see above the definition of our blinder). We prove that there is no difference between G_1 and G_2 with respect to $\mathcal{A}$'s final decision. That is, $P(G_1) = P(G_2)$.

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.

We may assume, without loss of generality, that $\mathcal{A}$ queries the $Result$ ($Result_{\mathcal{B}}$) oracle after the second step of the protocol. If this is done before, both oracles return $\perp$. What we have to show next is that $Result(\pi) = 1$ if and only if $Result_{\mathcal{B}}(\pi) = 1$, for any protocol instance π.

Assume $Result(\pi) = 1$. Then, there is a partial transcript $tr_\pi = (x, (y, z))$ defined by a sequence of oracle queries $x \leftarrow SendReader(\emptyset, \pi)$ and $(y, z) \leftarrow SendTag(x, vtag)$ such that $vtag$ refers to some tag $\mathcal{T}_{ID}$ whose state is s and secret key is K, $z = F_K(0, x, y)$, 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 obtained in a similar way.

As a conclusion, $P(G_1) = P(G_2)$.

Game G_3: This game is identical to G_3 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 game is identical to G_3 except that the $SendReader(\emptyset, \pi)$ oracle is simulated according to the blinder description. By doing this, the probability distribution

$$\{x \mid x \leftarrow \{0,1\}^{\ell_1(\lambda)}\}$$

is not changed and, therefore, $P(G_3) = P(G_4)$.

Game G_5: This game is identical to G_4 except that the $SendTag(x, vtag)$ oracle is simulated according to the blinder description. That is, for each tag $\mathcal{T}_{ID}$ whose secret key is K, the probability distribution

$$\{(x, y, z) \mid x, y \leftarrow \{0,1\}^{\ell_1(\lambda)},\ z = F_K(0, x, y)\}$$

is replaced by

$$\{(x, y, z) \mid x, y \leftarrow \{0,1\}^{\ell_1(\lambda)},\ z \leftarrow \{0,1\}^{\ell_2(\lambda)}\}.$$

As the two distributions are indistinguishable (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 of Theorem 4.2. Therefore, we will only sketch the main idea.

Assume that an adversary $\mathcal{A}$ can distinguish with non-negligible probability between G_4 and G_5. Define an adversary $\mathcal{A}'$ for PRF that uses $\mathcal{A}$ as a subroutine and sends $(0, x, y)$ as a challenge. When the PRF challenger returns, with equal probability, either $z = F_K(0, x, y)$ or $z \leftarrow \{0,1\}^{\ell_2(\lambda)}$, $\mathcal{A}'$ sends this value to $\mathcal{A}$. The probability with which $\mathcal{A}'$ guesses between the two possibilities for z is exactly the probability with which $\mathcal{A}$ distinguishes between the two games.

Game G_6: This game is identical to G_5 except that the $SendReader((y, z), \pi)$ oracle is simulated as defined in the blinder description. That is, for each tag $\mathcal{T}_{ID}$, the probability distribution

$$\{(x, y, z, w) \mid x, y \leftarrow \{0,1\}^{\ell_1(\lambda)},\ z \leftarrow \{0,1\}^{\ell_2(\lambda)}, w = F_K(1, x, y)\}$$

or

$$\{(x, y, z, w) \mid x, y \leftarrow \{0,1\}^{\ell_1(\lambda)},\ z, w \leftarrow \{0,1\}^{\ell_2(\lambda)}\},$$

where K is as in the protocol (Figure 3), is replaced by

$$\{(x, y, z, w) \mid x, y \leftarrow \{0,1\}^{\ell_1(\lambda)},\ z, w \leftarrow \{0,1\}^{\ell_2(\lambda)}\}.$$

As these distributions are indistinguishable (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 game is identical to G_6 except that the $SendTag(w, vtag)$ oracle is simulated as defined in the blinder description. 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 blinder which does not use the secret keys either. Therefore, G_7 is indeed $RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}$.

Now, to derive the final conclusion of the proof we remark $P_{\mathcal{A}}(G_0) = P(RFID^{prv-0}_{\mathcal{A},\mathcal{S}}(\lambda) = 1)$ 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. ■

V. Destructive privacy flaws in previous RFID schemes

The aim of this section is to discuss the related work to our paper and to point out some proof and design flaws in destructive private RFID schemes.

a) Simulating the oracle $Result$ by blinder: The simulation of the $SendReader$ and $SendTag$ oracles by blinder is simply done by putting the blinder to reply with random messages. The simulation of the $Result$ oracle was done in [10], [13] similar to what we have in the proof of Theorem 4.4, except that the constraint "$vtag$ refers to some legitimate tag" was omitted. First of all we remark that this constraint can be checked by blinder because the blinder sees what the adversary sees (therefore, it sees the result of the $DrawTag$ oracle). Then, this constraint is crucial to have a correct proof because, otherwise, the adversary can distinguish between the real privacy game and the blinded privacy game. We will explain this on our protocol, but it can easily be translated to the protocols in [10], [13], [14]:

1) $CreateTag^0(ID)$ ($\mathcal{A}$ creates an illegitimate tag);
2) $(vtag, 0) \leftarrow DrawTag(P(ID) = 1)$ ($\mathcal{A}$ draws the tag);
3) $\pi \leftarrow Launch()$ ($\mathcal{A}$ launches a new protocol instance);
4) $x \leftarrow SendReader(\emptyset, \pi)$ ($\mathcal{A}$ queries the reader to send a message to the tag);
5) $(y, z) \leftarrow SendTag(x, vtag)$ ($\mathcal{A}$ queries the tag to answer to the reader);
6) $w \leftarrow SendReader((y, z), \pi)$ ($\mathcal{A}$ sends (y, z) to reader in order to enable the oracle $Result$);
7) $b \leftarrow Result(\pi)$ ($\mathcal{A}$ asks for the authentication result).

In the real privacy game, b must be 0 because the tag is illegitimate. However, in the blinded privacy game the result is 1. Therefore, the adversary $\mathcal{A}$ undoubtedly distinguishes what privacy game is playing.

b) When should the Result oracle be simulated?: In the sequence-of-games approach as we have used in the proof of Theorem 4.4, it is crucial to find the right position to simulate the $Result$ oracle. In our proof, we simulated it before simulating $SendReader$ and $SendTag$ (please see the transition from G_1 to G_2). If this oracle is simulated after $SendReader$ and $SendTag$, as it was done in [10], [13], [15], then it might be quite hard to correctly infer the right conclusion. For instance, in [10], [13] this oracle is simulated in the transition from G_2 to G_3, after $SendReader$ and $SendTag$. The main argument for the fact that $|P(G_2) - P(G_3)|$ is negligible was claimed as being the tag authentication property. However, the tag authentication property is achieved by the original privacy game. Therefore, one needs to show that this property is achieved by game G_2 as well, in order to have a sound proof.

Simulating the $Result$ oracle before $SendReader$ and $SendTag$, as we did in the proof of Theorem 4.4, avoids the trouble.

c) Corruption with temporary state disclosure: In a very nice article [12], a series of impossibility results regarding privacy of RFID schemes were presented. Among them, it was shown that there is a huge difference between corruption with and without temporary state disclosure (please see Section III for terminology). For instance, an interesting result that can mainly be found in [12] (Alg. 2) shows that a protocol cannot be destructive private if there are volatile variables that are assigned in one step of the protocol and their values are used later in other step of the protocol. This is simply because in the blinded privacy game these variables are inactive while in the original privacy game they are active. We will illustrate this on two RFID schemes that claimed they achieve destructive privacy in Vaudenay's model with temporary state disclosure.

The first RFID scheme is the one proposed in [14], pictorially represented in Figure 4. To understand this scheme, let us recall the *cold boot attack* for PUFs [21]. This attack says that it is possible to freeze the tag's state and recover the PUF value (if it was just computed). This can be regarded as a corruption with temporary state disclosure. To avoid the cold boot attack, the authors of [14] proposed a *two PUF evaluation technique*, which consists of evaluating the same PUF twice. If the attack is applied immediately after the first PUF evaluation, the value of the second PUF evaluation will be missed, and vice-versa.

Having this in mind, the authors of [14] claimed that their RFID scheme achieves destructive privacy in Vaudenay's model with temporary state disclosure. Unfortunately, this is not true because the volatile variable v_2 used in step 2 is used again in step 4. Then, no matter of the blinder, a narrow adversary $\mathcal{A}$ can do as follows in order to identify the blinder with high probability:

1) $CreateTag^1(ID)$ ($\mathcal{A}$ creates a legitimate tag);
2) $(vtag, 1) \leftarrow DrawTag(P(ID) = 1)$ ($\mathcal{A}$ draws the tag);
3) $\pi \leftarrow Launch()$ ($\mathcal{A}$ launches a protocol instance);
4) $(ID_R, r_1, c_R) \leftarrow SendReader(\emptyset, \pi)$ ($\mathcal{A}$ gets the reader's first message);
5) $(r_2, v_1) \leftarrow SendTag((ID_R, r_1, c_R), vtag)$ ($\mathcal{A}$ gets the tag's response to the reader's query);
6) $v_2' \leftarrow SendReader((r_2, v_1), \pi)$ ($\mathcal{A}$ gets the response for reader authentication);

```
7) $(\ldots, c_T, v_2) \leftarrow CorruptTag(vtag)$ ($\mathcal{A}$ corrupts the tag);
8) If $v_2 = v'_2$ and $c_R > c_T$
       then $b = 0$ ($\mathcal{A}$ plays in the original game)
       else $b = 1$ ($\mathcal{A}$ plays in the blinded game);
9) return $b$.
```

Remark that the test in step 8 of the algorithm can be carried out by $\mathcal{A}$ because it knows v_2, v'_2, c_R, and c_T.

As a conclusion, the two PUF evaluation technique is useless in the case of the RFID scheme in Figure 4 because, regardless if we use it or not, the scheme is not narrow destructive private in Vaudenay's model with temporary state disclosure. The question that remains is whether the scheme is destructive private in Vaudenay's model. Unfortunately, we failed to see in the destructive privacy proof (Theorem 5.5 in [14]) how a destructive adversary distinguishes between the original privacy game and the blinded privacy game.

	Reader $(ID_R, c_R DB = [(ID_i, K_i^1, K_i^2)])$		**Tag** (ID, G, c_T, P)
1	$r_1 \leftarrow \{0,1\}^\alpha$	$\xrightarrow{ID_R, r_1, c_R}$	
2		$\xleftarrow{r_2,\ v_1}$	$r_2 \leftarrow \{0,1\}^\alpha$ If $c_R \geqslant c_T$ then $S^1 = P(G)$, $K^1 = H(S^1, ID_R, c_R)$ $temp = H(K^1, r_1, r_2)$, delete (S^1, K^1) $S^2 = P(G \oplus ID)$ $K^2 = H(S^2, ID_R, c_R)$ $v_1, v_2 = H(K^2, temp)$, delete (S^2, K^2) else $v_1 \leftarrow \{0,1\}^\gamma$
3	If $\exists (ID_i, K_i^1, K_i^2)$ s.t. $v'_1, v'_2 = H(K_i^2, H(K_i^1, r_1, r_2))$ $v'_1 = v_1$ then output ID (tag auth.) else output $\perp$ $v'_2 \leftarrow \{0,1\}^\gamma$	$\xrightarrow{v'_2}$	
4			If $v_2 = v'_2$ && $c_R > c_T$ then output OK (reader auth.) $c_T = c_R$ else output $\perp$

Fig. 4. RFID scheme proposed in [14]

Our second example in this paragraph is the RFID scheme in [15] that claimed to achieve narrow destructive privacy in Vaudenay's model with temporary state disclosure (pictorially represented in Figure 5). The scheme uses the two PUF evaluation technique as well. Unfortunately, the protocol suffers from the same problem as the one above: it uses the volatile variable h to transmit values from step 2 to step 4. A similar attack as the one presented above can be mounted in this case too. Therefore, the protocol does not achieve narrow destructive privacy under state disclosure as it was claimed in [15]. As we did with the previous RFID scheme, we tried to see whether or not the destructive privacy proof holds in Vaudenay's model. Unfortunately, Theorem 6.4 in [15] is based on statements for which the proofs are missing.

	Reader $(S, DB = [(ID_i, a_i, b_i)])$		**Tag** (ID, a, b, c)
1	$r_1 \leftarrow \{0,1\}^l$	$\xrightarrow{r_1}$	
2		$\xleftarrow{M1,\ M2,\ k}$	$r_2 \leftarrow \{0,1\}^l$, $M1 = H(r_1, r_2, a)$ $M2 = H(r_2, r_1, 1) \oplus ID$ $h = H(r_2, 1, 2)$ $k = P(a) \oplus r_2$, delete $(P(a), r_2)$ $k = k \oplus P(b) \oplus c$, delete $P(b)$
3	$r_3 \leftarrow \{0,1\}^l$ $r_2' = S \oplus k$ $ID_i' = M2 \oplus H(r_2', r1, 1)$ Retrieve (ID_i, a_i, b_i) from DB s.t. $ID_i = ID_i'$ If $M1 = H(r_1, r_2', a_i)$ then output ID (tag auth.) $M3 = H(H(r_2', 1, 2), r_3, b_i)$ else output $\perp$ $M3 \leftarrow \{0,1\}^K$	$\xrightarrow{M3, r_3}$	

4			If $M3 = H(h, r_3, b)$
			  then output OK (reader auth.)
			  else output $\perp$

Fig. 5. RFID scheme proposed in [15]

VI. Conclusion

In this paper we have made a detailed presentation of Vaudenay's model and emphasized subtle aspects to be considered in achieving privacy in conjunction with mutual authentication. We have extended the protocol from [10], [13] with an extra round so as to provide mutual authentication. The resulting protocol can also be regarded as a PUF based extension of the weak private protocol from [6]. This falls in line with the extensions performed in [6], [10], [13] of the protocols from [5]. Our protocol achieves mutual authentication and destructive privacy, for which we have presented complete and rigorous proofs.

We have also pointed out some privacy proof flaws in [10], [13] and suggested a fix. Finally, we have shown that the only two RFID schemes proposed so far that claimed to achieve destructive privacy and mutual authentication are not even narrow forward private. Therefore, our protocol appears to be the first one that achieves this kind of security and privacy.

References

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

[2] 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.

[3] 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.

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

[5] 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.

[6] 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, NY, USA: ACM, 2008, pp. 292–299.

[7] G. Avoine, X. Carpent, and B. Martin, “Strong authentication and strong integrity (SASI) is not that strong,” in *Proceedings of the 6th International Conference on Radio Frequency Identification: Security and Privacy Issues*, ser. RFIDSec’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 50–64.

[8] 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.

[9] K. Ouafi and S. Vaudenay, “Strong privacy for rfid systems from plaintext-aware encryption,” in *Cryptology and Network Security*, J. Pieprzyk, A.-R. Sadeghi, and M. Manulis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 247–262.

[10] 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.

[11] K. Ouafi and S. Vaudenay, “Strong privacy for RFID systems from plaintext-aware encryption,” Slides presented at the International Conference on Cryptology and Network Security, 2012.

[12] F. Armknecht, A.-R. Sadeghi, A. Scafuro, I. Visconti, and C. Wachsmann, “Transactions on computational science xi,” M. L. Gavrilova, C. J. K. Tan, and E. D. Moreno, Eds. Berlin, Heidelberg: Springer-Verlag, 2010, ch. Impossibility Results for RFID Privacy Notions, pp. 39–63.

[13] A.-R. Sadeghi, I. Visconti, and C. Wachsmann, *Enhancing RFID Security and Privacy by Physically Unclonable Functions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 281–305.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] V. Shoup, “Sequences of games: A tool for taming complexity in security proofs,” 2004.

[21] S. Kardaş, M. S. Kiraz, M. A. Bingöl, and H. Demirci, “A novel RFID distance bounding protocol based on physically unclonable functions,” in *RFID. Security and Privacy*, A. Juels and C. Paar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 78–93.