Privacy of Stateful RFID Systems with Constant Tag Identifiers

Cristian Hristea and Ferucio Laurenţiu Ţiplea

Abstract

There is a major interest in designing RFID schemes based on symmetric-key cryptography and ensuring efficient tag identification. These requirements taken together often lead to a decrease in the degree of privacy provided by the scheme. This issue, as we know, has been treated in an ad-hoc manner so far.

In this paper, we introduce the class of *stateful RFID schemes with constant tag identifiers*, that ensure tag identification in no more than logarithmic time. In order to study their privacy, we propose an appropriate general model obtained by constraining Vaudenay's model. We then propose two symmetric-key cryptography based RFID schemes in this class that achieve weak and destructive privacy, respectively, in addition to mutual authentication. We also discuss on the degree of privacy provided by other schemes proposed in the literature, that fall in this class.

1 Introduction

The core of a *Radio Frequency Identification* (RFID) *system* [1], [2] is the *tag*, which is usually a resource constrained device without an internal power source. Its aim is to provide identification information about the object to which it is attached. So, when the tag is powered by a *reader*, it sends information to the reader through a radio frequency field. The reader, viewed as an unconstrained device, has secure access to a *database* where tag related information is stored.

In recent years, the RFID technology has become increasingly popular, being introduced in a variety of fields, such as: process automation, tracking and identification, toll collection, public transportation, national IDs and passports, healthcare systems, and so on. The size of RFID systems has also increased greatly, thus raising the issue of *efficient tag identification*. In the worst case, tag identification can be done in linear time with respect to the size of the database. In large-scale systems, this complexity may not be the one you want. At a first sight, it seems that the problem of tag identification is a search problem in a database. This is true, but its efficiency greatly depends on the information received from the tag. For example, if using public-key cryptography, the tag can send information to the reader by encrypting it with the reader's public-key (see, for instance, the public-key based RFID scheme in [3], [4]). By decrypting the tag's message, the reader can extract pointers to the relevant information in the database. Thus, the search in the database can be done in constant time.

Unfortunately, public-key cryptography is costly in comparison to symmetric-key cryptography [5]. On the other side, using symmetric-key cryptography in building RFID protocols is no longer that easy, and faces the privacy issue more than in the case of public-key cryptography. Thus, the idea of partial sacrifice of privacy in favor of efficient identification is emerging, through symmetric-key based protocols. Obviously, the partial sacrifice of privacy must be intimate with

The authors 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.

the practical application in the sense that, if a certain privacy property is not needed then the protocol should be properly relaxed.

This idea appeared sporadically to various researchers and was treated in ad-hoc manner [6]–[11]. The common direction that can be extracted from these is that the tag responds to the reader with a message that explicitly contains a temporary identifier that ensures efficient search in the database. This identifier needs to change only after the tag is identified in the database and not before. This is because uncompleted queries of the tag would only modify the temporary identifier and desynchronize it from the related information in the database. So, the database search process would become more difficult. The need for the temporary tag identifier to remain constant until tag identification, certainly leads to a loss of privacy. The analysis of this aspect is entirely lacking in the above-mentioned papers. In fact, the study of privacy for the schemes in [6]–[11] was essentially done through ad-hoc models.

Contribution: In this paper we introduce the class of *stateful RFID schemes with constant tag identifiers*, where each tag has a temporary identifier that is kept constant until the tag is identified. Due to this property, the identification can be done in no more than logarithmic time. Moreover, depending on the tag identifier and protocol, the identification time may even be lowered to constant time. The schemes in [6]–[11] fall, more or less, in this class.

We then show that stateful RFID schemes with constant tag identifiers do not provide any kind privacy in Vaudenay's model. However, the loss of privacy in this model does not mean that these schemes are useless, but that the model by which they are analyzed is not the most appropriate. So, we next emphasize two categories of practical applications where these types of schemes are appropriate, arguing thus the necessity to study their privacy in a weaker privacy model.

The privacy of stateful RFID schemes with constant tag identifiers was studied in literature either in an informal way or by ad hoc privacy models. We review most of these approaches and we show that the Refresh privacy model [9], the most significant step toward analysis of stateful RFID schemes with constant tag identifiers, is limited to just unilateral authentication protocols. Moreover, we show that the RFID scheme LAST [9] is insecure and does not achieve any privacy in the Refresh model, contrary to what its authors claimed.

The paper proposes next a privacy model to discuss in a unitary way the privacy of stateful RFID schemes with constant tag identifiers. The model is called the *randomized Vaudeny's model*, and it is obtained by modifying the $Free$ oracle in Vaudenay's model. Then, examples of schemes that achieve mutual authentication together with weak and destructive privacy in this model, are provided. The proofs for these are rigorous and complete.

We emphasize that the randomized Vaudenay's model is just a privacy model used to discuss the privacy of a class of RFID schemes that cannot achieve any privacy in the original Vaudenay's model, but that are useful in some practical applications.

Paper structure: The paper is divided into seven sections, the first being the introductory section. The basic concepts and notations used in this paper are presented in Sections 2 and 3 (the latter being especially dedicated to RFID systems). Section 4 introduces the class of stateful RFID schemes with constant tag identifiers and shows that this class can not provide privacy in Vaudenay's model. As a result, Section 5 proposes an appropriate model to study privacy of this new class of schemes. The sixth section proposes two stateful RFID schemes with constant tag identifiers that achieve mutual authentication and weak/destructive privacy in the randomized Vaudenay's model. The final section concludes the paper.

2 BASIC DEFINITIONS AND NOTATION

We fix in this section the basic terminology and notation used throughout this paper. For details the reader is referred to [12] for probabilistic algorithms, to [13] for cryptographic concepts, and to [14] for physically unclonable functions.

Given a positive integer ℓ, $\{0,1\}^\ell$ stands for the set of all binary strings of length ℓ. The length of a binary string K is denoted $|K|$.

If $\mathcal{A}$ is a *probabilistic polynomial time* (PPT) algorithm and $\mathcal{O}$ 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.

For a set A, $a \leftarrow A$ 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.

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 $U_\lambda = \{f \mid f : \{0,1\}^{\ell_1(\lambda)} \rightarrow \{0,1\}^{\ell_2(\lambda)}\}$ and $U = (U_\lambda)_{\lambda \in \mathbb{N}}$.

We say that F is *computationally indistinguishable* from U if, for any PPT algorithm $\mathcal{A}$, 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 ($P(E)$ is the probability that the event E occurs).

$F = (F_K)_{K \in \mathcal{K}}$ 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 $F_K(x)$;
2) *Pseudo-random* : F is computationally indistinguishable from U.

To prove that F is a PRF, we usually use a *bit guessing game* between a *challenger* $\mathcal{C}$ and an adversary $\mathcal{A}$ (the game is parameterized by a security parameter λ):

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 probability $\mathcal{A}$ wins the game is denoted $P(b' = b)$. Now, one can see that F is a PRF if it is efficiently computable and $Adv^{prf}_{\mathcal{A},F}(\lambda) = |P(b = b') - 1/2|$ is negligible.

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, unpredictable, and tamper-evident. Unfortunately, PUFs are subject to noise induced by the operating conditions and, therefore, they 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 Based on these, we adopt here the concept of an *ideal PUF* slightly different than in [15]. Namely, an *ideal PUF* is a physical object with a challenge/response behavior that implements a function $P : \{0,1\}^p \rightarrow \{0,1\}^k$, 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).

3 (PUF BASED) RFID SCHEMES AND SYSTEMS

We recall in this section basic notions regarding RFID systems and Vaudenay's security and privacy model. For details, the reader is referred to [3], [4]. The references [16], [17] may also be consulted for an approach closely related to the one used in this paper.

The memory of an RFID tag is typically split into *permanent* (or *internal*) and *temporary* (or *volatile*). The permanent memory stores the state values of the tag, while the temporary memory can be viewed as a set of *volatile/temporary variables* used to carry out the calculations required by the communication protocol.

Given a *reader identifier* $\mathcal{R}$ and a set $\mathcal{T}$ of *tag identifiers* whose cardinal is polynomial in some security parameter λ, define an *RFID scheme over* $(\mathcal{R}, \mathcal{T})$ [3], [4] 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 tag specific secret K. The pair (ID, K) is stored in the reader's database DB;
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*).

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.

RFID systems are instantiations of RFID schemes.

One of the most influential security and privacy model for RFID schemes, which we follow in this paper, is *Vaudenay's model* [3], [4]. We recall below this model as in [16], [17]. Thus, we assume first that the oracles the adversary may query share and manage a common list of tags $ListTags$, 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 use of $ListTags$ does not change anything in Vaudenay's model; however, it will be very useful from a technical point of view (the challenger of the security or privacy games maintains such a list, vólens nólens).

The adversary is given access to the following oracles:

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, 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.
   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 does 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 does 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);
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).

It is customary to assume that the RFID tags can be corrupted to reveal not only their permanent memory but also the temporary variables. When the $Corrupt$ oracle is considered in such a way, we will refer to Vaudenay's model as being *Vaudenay's model with temporary state disclosure*.

The adversaries in Vaudenay's model are classified into the following classes:

- *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*: after the adversary has queried $Corrupt(vtag)$ and obtained the corresponding information, the tag identified by $vtag$ is destroyed (marked as destroyed in $ListTags$) and the temporary identifier $vtag$ wil 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 (in this approach, the database cannot store multiple records for the same tag identifier);

- *Strong adversaries*: there are no restrictions on the use of oracles.

Orthogonal to these classes, we have the class of *narrow* adversaries that do not have access to the $Result$ oracle. The narrow property can be combined with any of the previous properties in order to get another four classes of adversaries, *narrow weak*, *narrow forward*, *narrow destructive*, and *narrow strong*.

Now we are ready to introduce the *tag* and *reader authentication* properties as proposed in [3], [4], 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 $\mathcal{A}$ and an RFID scheme $\mathcal{S}$ (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 π s.t.:
   – π authenticates an uncorrupted legitimate tag $\mathcal{T}_{ID}$;
   – π had no tag authentication matching conversation with $\mathcal{T}_{ID}$.

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 π with a tag $\mathcal{T}_{ID}$ s.t.:
   – $\mathcal{T}_{ID}$ is a uncorrupted legitimate tag that authenticates the reader;
   – π 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 $RFID^{t\_auth}_{\mathcal{A},\mathcal{S}}(\lambda)$).

Privacy of an RFID scheme $\mathcal{S}$ [4] is captured in Vaudenay's model by means if a *blinder* for a class V of adversaries, which 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 (privacy games):

Experiment $RFID^{prv-0}_{\mathcal{A},\mathcal{S}}(\lambda)$

Set up the reader;
$\mathcal{A}$ gets the public key pk;
$\mathcal{A}$ queries the oracles;
$\mathcal{A}$ gets the secret table of the $DrawTag$ oracle;
$\mathcal{A}$ outputs a bit b';
Return b'.

Experiment $RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}(\lambda)$

Set up the reader;
$\mathcal{A}^{\mathcal{B}}$ gets the public key pk;
$\mathcal{A}^{\mathcal{B}}$ queries the oracles;
$\mathcal{A}^{\mathcal{B}}$ gets the secret table of the $DrawTag$ oracle;
$\mathcal{A}^{\mathcal{B}}$ outputs a bit b';
Return b'.

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

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

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.

The newest technologies allow *PUF tags* that are tags with PUFs inside them. An RFID scheme with PUF tags will sometimes be called *PUF based RFID scheme*. The tags that do not include PUFs will sometimes be referred to as *ordinary tags*, and the corresponding schemes, *ordinary RFID schemes*.

In order to adapt Vaudenay's model (with or without temporary state disclosure) to PUF based RFID schemes, we have to clarify what corruption means in this case. Taking into account that PUFs are tamper-evident, the approach we follow is that corruption on a PUF tag reveals the permanent (and temporary, if the model is with temporary state disclosure) memory of the tag, but the tag is considered destroyed. By corruption, the values computed by PUFs cannot be obtained (except when they were saved in the permanent memory or in some temporary variables that can be disclosed).

4 STATEFUL RFID SCHEMES WITH CONSTANT TAG IDENTIFIERS

We introduce in this section the class of *stateful RFID schemes with constant tag identifiers* (Section 4.1), we discuss their loss of privacy in Vaudenay's model (Section 4.2), we point out their applications (Section 4.3), and we review some papers that have "touched" them in various approaches (Section 4.4).

4.1 Efficient identification of RFID tags

Increasing the applicability of RFID technology in large-scale systems requires efficient identification and authentication of RFID tags. This led to the proposal of various RFID schemes with identification time varying from constant to linear. Indubitable, getting a better identification time is done at a certain price, and most of the times with a sacrifice of privacy. Finding a good balance between the tag identification time and the privacy level, is very important.

The goal of this section is to introduce a class of RFID schemes that allow an efficient identification time with a reasonable loss of privacy. This is the class of *stateful RFID schemes with constant tag identifiers*. To understand this class, we begin with the remark that RFID schemes can be classified into:

1) *stateless*: these are schemes where the tag's permanent state remains unchanged during the protocol execution;
2) *stateful*: these are schemes where the tag's permanent state is changed during the protocol execution. These schemes can further be classified into:
   a) *schemes with constant tag identifiers*: in these schemes, the first message sent by tag has a constant part (tag identifier) that allows the reader to efficiently identity the tag in its database. A tag identifier should not necessarily be thought as the tag's identity or some fixed message. It may change after the identification process (but not before);
   b) *schemes with random tag identifiers*: opposite to the previous class.

RFID schemes with constant tag identifiers are particularly relevant in case of mutual authentication, where the tag changes its tag identifier after it authenticates the reader.

Assumption 4.1. In this paper, we assume that the first message of the tag in a stateful RFID scheme with constant tag identifiers has one of the following forms:

1) $f(S)$ or
2) $(f(S), g(S, L))$,

where f and g are efficient computable functions, S is a tag state, and L is a list, possible empty, of random values. $f(S)$ is thought as being the constant tag identifier. Any of $f(S)$ and $g(S, L)$ may be a vector of values.

The value of a tag identifier, as it was defined, depends on the current tag's state. In order to achieve its goal, a tag identifier should uniquely identify its tag. Therefore, we adopt the following assumption.

Assumption 4.2. A tag identifier in a stateful RFID scheme with constant tag identifiers uniquely identifies its tag with overwhelming probability, no matter of the tag's current state.

The identification time of a tag in the reader's database DB depends on how the tag identifiers $f(S)$ are viewed as indices [18]. There are two main approaches: ordered indices and hash indices.

If we use ordered indices, then each index entry is a pair of a search key value ($f(S)$ in our case) and a pointer to the corresponding record or to a disk block containing it in DB. The sequential organization of all index entries requires that they are ordered by the search key value.

Therefore, the identification time of a tag is proportional to $\log n$ (n being the database size). Once the tag is identified and the tag's state is updated, the identifier $f(S)$ changes. Therefore, the index structure has to be updated as well. This can simply be done just by deleting the old index entry and inserting the new one in the right position, which takes $\log n$ time. Therefore, the entire process is proportional to $\log n$. Remark also that the new index entry is obtained from the old one by replacing the search key value (the pointer remains unaltered).

The sequential organization of indices has the main disadvantage that performance degrades as the index file grows. In such a case, one may think to organize indices on multiple levels or even as a B^+-tree. Lookup on B^+-trees is efficient; deletion and insertion are somewhat more complicated but still efficient. Thus, if the number of pointers in a non-leaf node is k, the height of the B^+-tree is proportional to $\log_k(n)$, and the identification and updating time is proportional to $\log_{k/2}(n)$. The value of k is often around 50 or 100.

The second method to organize indices is with the help of hashing. Two approaches can be distinguished here. In the first approach, a hash function maps a search key to the address of the desired record or to a *bucket* containing it. In such a case, lookup time is usually a constant, independent of the database size. In our case, the search key is the tag identifier that changes from session to session (when it is completed). Therefore, we cannot take advantage directly of this method because two hash values should identify the same record (this means hash collision). However, we may use hashing in a different approach. Namely, we compute hash indices for all possible search keys of each tag, we associate the corresponding pointers to the database records, and we view the hash index (the hash file structure) such obtained as secondary (hash) indices. For this hash index we may use the first hashing approach to search within it. A similar idea was used in [19], where constant time identification was claimed.

4.2 Privacy of stateful RFID schemes with constant tag identifiers

In stateful RFID schemes with constant tag identifiers the tag updates its permanent state after it is identified or authenticated by reader. This delay in updating the tag's state can be exploited by an adversary especially to trace tags: it opens a protocol session with the tag, gets the tag identifier, closes the session without allowing the tag to update its state, opens later a new protocol session with the tag and so on. Therefore, this class of RFID schemes loses privacy in Vaudenay's model. Formally, we have the following result.

Lemma 4.1. No stateful (PUF based) RFID scheme with constant tag identifiers achieves any form of privacy in Vaudenay's model (with or without temporary state disclosure).

Proof: Let $\mathcal{S}$ be a stateful (PUF based) RFID scheme with constant tag identifiers. Without loss of generality we assume that the tag takes the first step in the protocol (the other case is similar to this). We also denote by $TI(vtag)$ the tag identifier, assuming that there is just one, of the tag $vtag$ ($TI(vtag)$ also depends on the protocol session; however, this will be clear from the context so we avoid to overload the notation).

Consider now an adversary $\mathcal{A}$ that plays the following privacy game with the RFID scheme:

$RFID^{prv-0}_{\mathcal{A},\mathcal{S}}(\lambda)$

1) $\mathcal{A}$ creates two distinct tags with identities ID_1 and ID_2 and states S_1 and S_2, respectively;
2) $(vtag_1, 1) \leftarrow Draw(P(ID_1) = P(ID_2) = 1/2)$;
3) $\pi_1 \leftarrow Launch()$;
4) $m_1 \leftarrow SendTag(\emptyset, vtag_1)$;
5) $Free(vtag_1)$;
6) $\pi_2 \leftarrow Launch()$;

7) $(vtag_2, 1) \leftarrow Draw(P(ID_1) = P(ID_2) = 1/2)$;
8) $m_2 \leftarrow SendTag(\emptyset, vtag_2)$;
9) $Free(vtag_2)$;
10) $\mathcal{A}$ gets the association table Γ of $Draw$;
11) If $(\Gamma(vtag_1) = \Gamma(vtag_2)$ and $TI(vtag_1) \neq TI(vtag_2))$ or
$(\Gamma(vtag_1) \neq \Gamma(vtag_2)$ and $TI(vtag_1) = TI(vtag_2))$ then output 1 else output 0.

Remark first that each of the two tags will have the same state in π_2 as in π_1 (because π_1 has not completed the identification phase of $vtag_1$). As a conclusion, Assumption 4.2 shows that $P(TI(vtag_1) \neq TI(vtag_2))$ must be zero with overwhelming probability when $\Gamma(vtag_1) = \Gamma(vtag_2)$. Similarly, $P(TI(vtag_1) = TI(vtag_2))$ must be zero with overwhelming probability when $\Gamma(vtag_1) \neq \Gamma(vtag_2)$. Therefore, the output of $RFID_{\mathcal{A},\mathcal{S}}^{prv-0}(\lambda)$ is 1 with negligible probability.

Consider now an arbitrary blinder $\mathcal{B}$ and the blinded privacy game $RFID_{\mathcal{A},\mathcal{S},\mathcal{B}}^{prv-1}(\lambda)$ defined as $RFID_{\mathcal{A},\mathcal{S}}^{prv-0}(\lambda)$ is defined but with the difference that the $Launch$ and $SendTag$ oracles are simulated by $\mathcal{B}$. Without loss of generality we may assume that the blinder's answer to $SendTag(\emptyset, vtag)$ must include a part playing the role of the tag identifier (otherwise, the adversary can easily distinguish between the real privacy game and the blinded one).

To compute the probability $RFID_{\mathcal{A},\mathcal{S},\mathcal{B}}^{prv-1}(\lambda)$ returns 1, remark first that the events $\Gamma(vtag_1) = \Gamma(vtag_2)$ and $\Gamma(vtag_1) \neq \Gamma(vtag_2)$ happen with probability $1/2$. Therefore, the probability the final test in this game is true can be computed as

$$\frac{1}{2}(P(TI(vtag_1) \neq TI(vtag_2) + P(TI(vtag_1) = TI(vtag_2)),$$

which is $1/2$ with overwhelming probability.

As a conclusion,

$$\mid P(RFID_{\mathcal{A},\mathcal{S}}^{prv-0}(\lambda) = 1) - P(RFID_{\mathcal{A},\mathcal{S},\mathcal{B}}^{prv-1}(\lambda) = 1) \mid = \frac{1}{2}$$

with overwhelming probability.

As $\mathcal{A}$ is a narrow weak adversary, we conclude that $\mathcal{S}$ does not achieve any form of privacy in Vaudenay's model (with or without temporary state disclosure). □

4.3 Applications of stateful RFID schemes with constant tag identifiers

As we have seen in the previous section, stateful RFID schemes with constant tag identifiers lose privacy in Vaudenay's model. This is because the tag can be traced between two full protocol executions: the adversary launches consecutive protocol sessions with the tag without closing them. In this way, the tag identifier does not change, which makes the adversary know what tag it communicates with.

As this class of RFID schemes allow efficient identification time of tags in the database, it make it be very suitable to all applications that tolerate tracings of tags between two consecutive and complete protocol sessions. We will further discuss two such applications.

Tags with no mobility: Assume that an RFID scheme is designed to inventory items in a store or warehouse. The items, that have associated RFID tags, have very little or even no mobility. The fact that an adversary can trace tags has probably very little importance for the owner of the store or warehouse.

Tags with high mobility: Another useful application for this category of RFID schemes is in public transportation systems [1]. The tags are attached to tickets or passes that have to be validated when entering and leaving the public transportation (this also includes when changing it). A validation means that a protocol session is played with the tag (by a real reader) and the

tag's state is updated (randomized). Therefore, an adversary may trace the tag in between two consecutive validations. However, this might not leak relevant information to it.

4.4 Related work

The idea of using tag identifiers to make the search procedure in the reader's database more efficient was touched in papers like [6]–[11], although their authors have not used the terminology of constant tag identifier. We will briefly discuss a few of them.

Dimitrou's RFID scheme: The RFID scheme in [6] is one of the earliest stateful scheme with constant tag identifiers. The scheme uses a hash function and a MAC to perform authentication. Moreover, each tag shares a common secret K with the reader.

The protocol is as follows. The reader sends first a random integer r_1 to the tag. After receiving it, the tag computes $M_1 = h(K)$, which acts as a constant tag identifier, and sends it together with $M_2 = h_K(r_1, r_2)$ and r_2 to the reader. The reader uses M_1 to search the database for the tag. If it is found, the reader authenticates the tag according to the equality $M_2 = h_K(r_1, r_2)$. In case of successful authentication, the secret K is updated to a new one K'. Next, the scheme proceeds to the reader authentication phase. The reader sends $M_3 = h_{K'}(r_1, r_2)$ to the tag. Using the same key update method, the tag computes K' and checks if $M_3 = h_{K'}(r_1, r_2)$. If the equality holds then the reader is authenticated and K is replaced by K'.

The key update function is not specified in [6], but it is assumed to be "irreversible" (that is, no one can easily obtain K from K').

This protocol falls clearly in the class of stateful RFID schemes with constant tag identifiers. However, the security and privacy analysis in [6] was done in an ad hoc manner, without any formal model. This is mainly due to the fact that no formal analysis model had crystallized until the date the scheme was proposed. What we may clearly say now is that the scheme does not achieve any form of privacy in Vaudenay's model (this follows from Lemma 4.1).

The Refresh model and the RFID scheme LAST: The RFID scheme LAST proposed in [9] also falls into the class of stateful RFID schemes with constant tag identifiers. To study its privacy, the authors of [9] have introduced the *Refresh* model. We recall first this model and show that it is limited to RFID schemes that only provide unilateral authentication, and then recall the RFID scheme LAST and show that it is insecure and does not provide privacy in the Refresh model, contrary to what its authors claimed.

The Refresh model is a modification of the privacy model in [20] to allow the analysis of protocols with constant identifiers. Within this model, the adversary may consult the oracles $Launch$, $TagQuery$ (similar to $SendTag$), $ReaderSend$ (similar to $SendReader$), $Corrupt$ (reveals the permanent state of the tag and then destroys it), and $Result$. In the privacy game, the adversary interacts with a challenger that runs the system for it. First, the adversary may query freely the oracles. At some time it chooses two tags $\mathcal{T}_0$ and $\mathcal{T}_1$ and sends them to the challenger. The challenger "refreshes" the tags' states by querying them (with $TagQuery$) exactly once, chooses uniformly at random one of them say $\mathcal{T}_b$, and makes $\mathcal{T}_b$ available to the adversary (without revealing b). Then, the adversary is allowed to query again the oracles on all tags, including $\mathcal{T}_b$, except that it cannot corrupt $\mathcal{T}_b$. Finally, the adversary will make a guess $b' \in \{0, 1\}$. It wins the game if $b' = b$.

Using this model, the authors of [9] analyzed the protocols OSK/AO [21], [22] and YA-TRAP [7].

However, despite its intended purpose, the Refresh model turns out to be quite limited as the internal state of the tag is meant to be refreshed by querying it just once (please see step 7 from Figure 2 in [9]). Thus, the Refresh model cannot properly work for most stateful protocols

with constant tag identifiers that provide mutual authentication. This is because the tag's state in such protocols is refreshed after the tag authenticates the reader and this happens when the tag is usually queried the second time or later. The Refresh model works for OSK/AO and YA-TRAP just because these two protocols provide unilateral authentication.

Now, let us recall the RFID scheme LAST proposed in [9]. The scheme is stateful and uses constant tag identifiers. Within this scheme, the tag accommodates a hash function h and a random number generator. Each tag ID has an internal state $(Index, K)$ and a corresponding entry $(ID, Index, K)$ in the reader database, where $Index$ represents the current tag identifier and K is the tag's key. The reader starts the protocol with a random integer r_1. Upon receiving r_1, the tag generates a random number r_2, computes $V = h(r1, r2, K)$, and sends $(r_2, Index, V)$ to the reader. Using $Index$ as a search key, the reader retrieves the tag's database entry, computes $V' = h(r_1, r_2, K)$, and authenticates the tag (outputs ID) if the equality $V = V'$ holds. After authenticating the tag the reader updates $Index$ to $Index' = h(r_1, r_2, Index, K)$, K to $K' = h(r_1, r_2, K)$, and sends $\delta = h(r_1, r_2, K')$ to the tag. Upon receiving δ, the tag computes $K' = h(r_1, r_2, K)$ and checks if $\delta = h(r_1, r_2, K')$. If the equality holds the reader is authenticated and the tag proceeds to update its state by replacing $Index$ by $h(r_1, r_2, Index, K)$ and K by K'.

The protocol LAST is severely flawed with respect to security. This is because the new key K' is identical to the message V that is sent on the insecure channel and, therefore, even an eavesdropping adversary (with no other capabilities) can compute $\delta = h(r_1, r_2, V)$. Therefore, a tag can be cloned after observing two consecutive successful sessions.

As with respect to privacy, LAST is not even private in the Refresh model (therefore, Theorem 1 in [9] does not hold). To see this consider the following privacy game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ (in the Refresh model):

1) $\mathcal{A}$ selects two tags $\mathcal{T}_0$ and $\mathcal{T}_1$;
2) $Launch \rightarrow \pi_0$ ($\mathcal{A}$ creates a session);
3) $\mathcal{A}$ chooses a random number r_1;
4) $TagQuery(r_1, \pi_0, \mathcal{T}_0) \rightarrow (r_2, Index_0, V)$;
5) $\mathcal{A}$ submits $\mathcal{T}_0$ and $\mathcal{T}_1$ as its challenge tags;
6) The challenger $\mathcal{C}$ refreshes the two tags (it uses just one $TagQuery$ query for each tag), chooses a bit b, and makes $\mathcal{T}_b$ available to $\mathcal{A}$ (without revealing b);
7) $Launch \rightarrow \pi_1$ ($\mathcal{A}$ creates a new session);
8) $TagQuery(r_1, \pi_1, T_b) \rightarrow (r_2', Index_b, V')$;
9) If $Index_b = Index_0$ then output 0 else output 1.

Clearly, $\mathcal{A}$ wins the game with overwhelming probability, showing that LAST is not private in the Refresh model, contrary to what its authors claimed in [9] (Theorem 1).

Looking for destructive privacy: More recently [10], a PUF based RFID scheme was proposed to achieve destructive privacy and mutual authentication. The scheme is stateful and uses constant tag identifiers of the form $F_K(S \oplus hello)$, where F_K is a PRF with the key K and S is a tag's temporary identity that is updated when the protocol session is completed. The privacy of this scheme was analyzed in a variant of Vaudenay's model with a modified $Free$ oracle.

5 A PRIVACY MODEL FOR STATEFUL RFID SCHEMES WITH CONSTANT TAG IDENTIFIERS

The analysis of the stateful RFID schemes with constant tag identifiers proposed so far has been made either informally, as in the case of Dimitriou's scheme, or in limited ad hoc models, as in the case of the Refresh model. Thus, it is impossible to compare such scheme with each other in terms of privacy. It becomes then imperative to have a unitary privacy model for stateful schemes with

constant tag identifiers. As Vaudenay's model is, arguably, the most general and widely used privacy model for RFID systems, it is natural to try to adapt it to such schemes. In fact, what we have to do is to check privacy of stateful schemes with constant tag identifiers against a limited class of adversaries, namely against adversaries that can draw a tag at most once in between two complete protocol sessions. This is because we already know that stateful schemes with constant tag identifiers are not even weak private in Vaudenay's model if the adversary draws a tag more than once in between two complete executions of the protocol (Lemma 4.1).

Perhaps the simplest way to use Vaudenay's model with a restricted class of adversaries as mentioned above is to modify one of the oracles $DrawTag$ or $Free$: when a tag is drawn or freed, respectively, its state is randomized by at least one complete protocol session. The Refresh model implements somehow a randomization of the first type: the challenge tags are randomized first and then are made available to the adversary. The HPVP model [23] suggests a randomization of the second type in order to deal with stateful RFID schemes (please see Section IV(C) in [23]).

Whether we modify the oracle $DrawTag$ or $Free$ as discussed above, such a change captures the idea that Vaudenay's model is restricted to consider privacy only against adversaries that can draw a tag at most once in between two complete protocol sessions. However, to be in line with the approaches in [10], [23], we prefer to change the oracle $Free$. Therefore, let us proceed now to the detailed description of the new oracle $Free$. First, we draw the attention to illegitimate tags: these tags cannot be randomized using standard protocol sessions because they do not have entries in the reader's database. If this cannot be done, a similar attack as in the proof of Lemma 4.1 can be mounted with illegitimate tags instead of legitimate tags. Therefore, stateful RFID schemes with constant tag identifiers would not achieve any privacy in the new model.

The most natural solution to this problem is to maintain a database with illegitimate tags, similar to the database with legitimate tags, and to run the $Ident$ protocol with this database whenever an illegitimate tag has to be re-randomized. As we have already defined a list of all tags created in the system, $ListTags$, we may expand the entries in this list and change the oracles as follows (please see Section 3 for notations):

- $CreateTag^b(ID)$: calls the PPT algorithm $SetupT(pk, ID)$ that generates a pair (K, S). If $b = 1$, (ID, S', K) is added to DB (the tag is legitimate) and a vector $(ID, \_, \_, \_, 1)$ is added to $ListTags$; if $b = 0$ (the tag is illegitimate), a vector $(ID, S', K, \_, 0)$ is added to $ListTags$. The fourth component of these vectors is initially empty; it will hold the temporary identity of the tag when it is drawn by adversary;
- $rFree(vtag)$: this replaces the oracle $Free(vtag)$. Assume that $vtag$ refers to some tag with identity ID. The following steps are taken:
  1) If this tag is legitimate, the oracle calls $Ident(pk; \mathcal{R}(sk, DB); ID(S))$. Following the protocol run, the tag entry in DB is updated correspondingly;
  2) If this tag is illegitimate, the oracle calls $Ident(pk; ListTags; ID(S))$. This means that the scheme challenger runs the protocol with the tag $\mathcal{T}_{ID}$ as if it were the reader with the database $ListTags$ and the tag were legitimate. Following the protocol run, the tag entry in $ListTags$ is updated correspondingly;
  3) Remove 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.
- All the other oracles in Section 3 remain unchanged.

The model such obtained will be called the *randomized Vaudenay's model*. Its plain version is without temporary state disclosure, as Vaudenay's model is.

Remark 5.1. We would like to emphasize once again that the adoption of $rFree$ is based on the fact that, from some practical applications' point of view, it does not matter how many times the adversary queries a tag in between two consecutive and legitimate protocol sessions.

Remark 5.2. The result established by Lemma 4.1 might not hold in the randomized Vaudenay's model simply because the tag $vtag_1$ is randomized by at least one complete session before making it free (please see the proof of Lemma 4.1).

All the concepts introduced in Section 3 are naturally translated to the randomized Vaudenay's model. As a result, we get the same eight privacy levels. It is clear that privacy in Vaudenay's model (with or without temporary state disclosure, resp.) implies the same level of privacy in the randomized Vaudenay's model (with or without temporary state disclosure, resp.).

6 WEAK AND DESTRUCTIVE PRIVACY IN THE RANDOMIZED VAUDENAY'S MODEL

Our main goal now is to give examples of stateful (PUF based) RFID schemes with constant tag identifiers that achieve various privacy levels in the randomized Vaudenay's model. Moreover, we will be interested in schemes for which the tags are as light as possible, especially with no random generators.

6.1 Weak privacy

Dimitriou's scheme [6], already discussed in Section 4.4, seems to achieve weak privacy in the randomized Vaudenay' model. Of course, a formal proof is needed for this, but it is beyond the scope of this paper. We prefer to propose a stateful RFID scheme with constant tag identifiers that is lighter than Dimitriou's scheme and achieves weak privacy in the randomized Vaudenay's model.

So, the first RFID scheme we propose is the one whose communication protocol is represented Figure 1. The scheme uses two polynomials ℓ_1 and ℓ_2 in some security parameter λ and a PRF $F = (F_K)_{K\in\mathcal{K}}$, where $F_K : \{0,1\}^{2\ell_1+2} \to \{0,1\}^{\ell_2}$ for all $K \in \mathcal{K}_\lambda$ (for the sake of simplicity we use ℓ instead of $\ell(\lambda)$). The internal state of a tag consists of a pair (K, x), where K is a randomly chosen key and $x \in \{0,1\}^{\ell_1}$ acts as a "dynamic" identifier of the tag. The reader maintains a database DB with entries for all legitimate tags. Each entry is a vector (ID, K, x), where ID is the tag's identity and (K, x) is its state.

The mutual authentication protocol is as follows. The tag computes $z = F_K(0, 0, x)$ and sends it to the reader. The reader checks its database for a triple (ID, K, x) such that $z = F_K(0, 0, x)$ or $z = F_K(0, 0, x + 1)$. The reason is that at most one step desynchronization may occur between reader and tag; that is, when x is on tag, either x or $x - 1$ is on reader. When the reader finds the right value, resynchronizes with the tag and prepares the answer w. The tag evaluates the PUF, checks the value w received from reader, and takes a decision. It also updates x correspondingly and prepares the answer for reader. On receiving the tag's answer, the reader checks it, takes a decision, and updates x correspondingly.

Remark that if the reader does not update x (because it rejects the tag), then it will do so in the second step of the next protocol session (with the same tag). Therefore, the desynchronization between reader and tag is at most one step.

It is straightforward to check the correctness of this scheme. We list below a few properties of it.

Remark 6.1. Let $\mathcal{S}_1$ be the RFID scheme in Figure 1.

1) $\mathcal{S}_1$ provides reader-first authentication.
2) $\mathcal{S}_1$ does not use temporary variables.

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

Fig. 1. Stateful RFID scheme with constant tag identifiers that achieves weak privacy and reader-first authentication in the randomized Vaudenay's model with temporary state disclosure

3) The tag only needs to compute F.
4) There is no dedicated random generator on tag.
5) The desynchronization between reader and tag is at most one step. This allows for a quite efficient search procedure in the reader's database. Two ordered sets of indices are used: the first one with indices of the form $F_K(0, 0, x)$, and the second one with indices of the form $F_K(0, 0, x + 1)$. Both $F_K(0, 0, x)$ and $F_K(0, 0, x + 1)$ will point to the same database record (ID, K, x).
   When a z is received by reader, the first set of indices is searched for it; if not found, then the second set is searched for it. Therefore, it takes $\mathcal{O}(\log n)$ time to search for a tag, where n is the size of the databases.
   When the reader authenticates the tag, x is updated both in the database and in the indices sets. Moreover, the indices sets must be resorted. This can be simply done in $\mathcal{O}(\log n)$ time because the old indices have to be removed and the new ones have to be reinserted in the right position.
   A more efficient search can be performed by using hash indices as in [19], where constant time is claimed for search. Moreover, the technique in [19] applied to our scheme works much better than for the scheme in [19]. This is because the reader-tag desynchronization in our scheme is at most one step, while in [19] it is bounded by some polynomial $c(\lambda)$ in the security parameter λ. This fact leads to $c(\lambda)$ sets of indices in [19], while in our case we have only two such sets.

The RFID scheme $\mathcal{S}_1$ is not weak private in Vaudenay's model because Lemma 4.1 applies to it. However, we have the following results.

Theorem 6.1. The RFID scheme in Figure 1 achieves tag authentication in Vaudenay's model with temporary state disclosure, provided that F is a PRF.

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 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;
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. 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 $x \in \{0,1\}^{\ell_1}$ and records a corresponding entry into $\mathcal{A}'_{ListTags}$. 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 $x \leftarrow \{0,1\}^{\ell_1}$. 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 $\mathcal{C}$;
   b) $DrawTag$: $\mathcal{A}'$ knows the list of all tags created by $\mathcal{A}$, and updates it correspondingly whenever $\mathcal{A}$ draws some tag;
   c) $rFree(vtag)$: when $\mathcal{A}$ wants to free $vtag$, the adversary $\mathcal{A}'$, which knows the tag identity, updates the parameter x of the tag to $x+1$. This value may exceed $\{0,1\}^{\ell_1}$ with negligible probability. In this way, $\mathcal{A}'$ simulates that the tag is randomized by exactly one complete protocol session;
   d) $Launch()$: $\mathcal{A}'$ launches a new protocol instance whenever $\mathcal{A}$ asks for it;
   e) $SendTag(\emptyset, vtag)$: This is the first message $vtag$ sends in a protocol instance. If the tag referred by $vtag$ is ID^*, then $\mathcal{A}'$ will query $\mathcal{C}$ for $(0,0,x)$. If z is $\mathcal{C}$'s response, than $\mathcal{A}'$ answers with 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;
   f) $SendReader(z, \pi)$: Assume the reader (run by $\mathcal{A}'$) has received z in the protocol instance π from a tag identified by $vtag$ (in other words, $z \leftarrow SendTag(\emptyset, vtag)$).
      If $vtag$ refers to some tag ID such that $(ID, K, x) \in DB$ for some (K, x), 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 queering $\mathcal{C}$ (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 w is randomly chosen;

g) $SendTag(w, vtag)$: If the tag referred by $vtag$ is ID^*, then $\mathcal{A}'$ queries $\mathcal{C}$ for $(0, 1, x)$ and then compares the answer with w. If they match, the tag outputs OK; otherwise, it outputs $\perp$. In the first case $\mathcal{A}'$ increments x and queries $\mathcal{C}$ for $(1, 1, x)$ to get w'; in the second case, it queries $\mathcal{C}$ for $(1, 0, x)$ to get w'. 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';

h) $Result(\pi)$: $\mathcal{A}'$ can infer the decision of the reader in the last step of π because it can obtain the value $F_K(1, 1, x+i+1)$ 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;

5) If $\mathcal{A}$ is able to make the reader to authenticate the tag ID^*, then this means that $\mathcal{A}$ can compute $w' = F_{K^*}(1, 1, x + 1)$ without knowing K^*, provided that K^* is the key chosen by $\mathcal{C}$ and x is the current third component of the entry $(ID^*, ?, x)$. Then, $\mathcal{A}'$ can prepare the challenge phase for $\mathcal{C}$ as follows:
   a) $\mathcal{A}'$ sends $(1, 1, x + 1)$ to $\mathcal{C}$;
   b) $\mathcal{C}$ randomly chooses $b \in \{0, 1\}$; if $b = 1$, then $\mathcal{C}$ returns $w'' = F_{K^*}(1, 1, x + 1)$, 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 $\mathcal{C}$ 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 the polynomial 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 $\mathcal{C}$'s challenge with non-negligible probability; 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 6.2. The RFID scheme in Figure 1 achieves reader authentication in Vaudenay's model with temporary state disclosure, provided that F is a PRF.

Proof: The main idea is similar to the one in the Theorem 6.1. Instead of using the fourth verification protocol step to derive a contradiction, we use the second one, where the tag authenticates the reader. The proof details are omitted. □

In order to prove that our scheme in Figure 1 achieves privacy in the randomized Vaudenay's model with temporary state disclosure, we use the *sequence-of-games* approach [24]. With this approach, a sequence of games 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 6.3. The RFID scheme in Figure 1 achieves weak privacy in the randomized Vaudenay's model with temporary state disclosure, provided that F is a PRF.

Proof: Let $\mathcal{A}$ be a weak adversary against our RFID scheme denoted $\mathcal{S}_1$. We will show that there is a blinder $\mathcal{B}$ such that $Adv^{prv}_{\mathcal{A},\mathcal{S}_1,\mathcal{B}}(\lambda)$ is negligible. The blinder $\mathcal{B}$ that we construct

has to answer to the oracles *Launch*, *SendReader*, *SendTag*, and *Result* without knowing any secret information. Before defining the blinder we recall that each tag can be involved in at most one protocol session (the reader may run several sessions in parallel). Without loss of generality we assume the following (see also Section 3):

- In any protocol session, the reader has at most one matching conversations with exactly one tag. Each conversation follows the order of the steps in protocol;
- When a new session π is launched and a matching conversation with $vtag$ is initiated by $SendTag(\emptyset, vtag)$ (that is, the reader powers the tag), it is implicitly assumed that any matching conversation between reader and $vtag$ in any previous session (if any) is automatically closed (no matter if it is complete or not).

The blinder we define keeps track of all sessions and matching conversations between reader and tags, according to the above assumptions. It simulates the $SendTag$ and $SendReader$ oracles as in Figure 2.

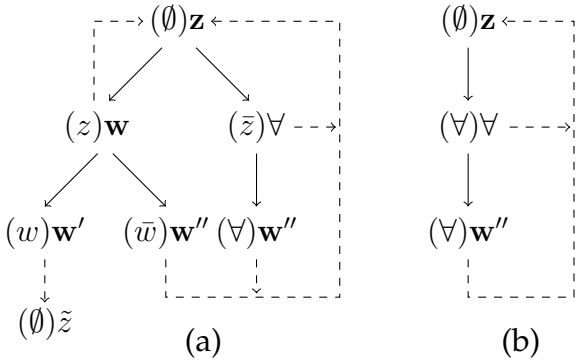

Fig. 2. Simulation of $SendTag$ and $SendReader$ oracles by blinder: (a) legitimate tags; (b) illegitimate tags

The meaning of the diagram in Figure 2(a) is as follows. Assume that a protocol session π is initiated and it is the first one that involves a legitimate $vtag$. The blinder generates at random four integers z, w, w', and w'' that are thought as corresponding to $F_K(0,0,x)$, $F_K(0,1,x+i)$, $F_K(1,1,x)$, and $F_K(1,0,x)$, respectively (using exactly the notation in protocol). These integers are kept fixed as long as $vtag$ does not change its state ($vtag$ changes its state when it reaches the node $(w)w'$ in Figure 2(a). $\bar{z}$ and $\bar{w}$ are integers different from z and w, respectively. Their meaning is that the adversary uses them instead of z and w. Therefore, $\bar{z}$ and $\bar{w}$ may be fresh or old.

A node $(\alpha)\beta$ means that the tag/reader answers by β when it is queried by α. $\beta = \forall$ means that the answer is a random integer, and $\alpha = \forall$ means that the answer is β no matter of the query value.

An arc from $(\alpha)\beta$ to $(\beta')\gamma$ says that the tag/reader was queried for α, it answered with β, then the reader/tag was queried for β' (that might be different from β), and the tag/reader's answer was γ. Remark that the blinder sees what the adversary sees and, therefore, it may work in this way (although it does not know the secrets). The other arcs have a similar meaning.

A dashed arc from a node $(\alpha)\beta$ to the tree root means that π was closed in that node and a new protocol session π' with $vtag$ was initiated. In this new protocol session, the blinder has to use the same integers z, w, w', and w''. For instance, the tag has to answer by z to the initial query. Then, if the adversary queries the reader by some value different from z, the reader answers by some random value. No matter of the value used then to query the tag, the answer must be w''. The dashed arrow to $(\emptyset)\tilde{z}$ means that the value x of the tag was updated and, therefore, any new protocol session that involves $vtag$ must use new randoms $\tilde{z}$, $\tilde{w}'$, $\tilde{w}'$, and $\tilde{w}''$.

In a similar way is interpreted the diagram in Figure 2(b). Remark that, under the blinder simulation, an illegitimate tag never updates its state. In the real privacy game, an illegitimate tag may update its state only if the tag and reader authentication property are broken (that is, the adversary can determine the reader to authenticate an illegitimate tag and then the tag authenticates the reader). However, in the case of our RFID scheme this may happen with negligible probability.

The two oracles that remains to be discussed are:

- $Launch()$: the blinder returns a unique identifier π for a new protocol instance;
- $Result(\pi)$: 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 $tr_\pi = (z, w, w')$ has been generated by
  - $z \leftarrow SendTag(\emptyset, vtag)$,
  - $w \leftarrow SendReader(z, \pi)$,
  - $w' \leftarrow SendTag(w, vtag)$, 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 $G_0, \ldots, G_6$, 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 $0 \leq i < 6$. By $P(G_i)$ we denote the probability the adversary $\mathcal{A}$ wins the game G_i.

Game G_1: We replace in G_0 the oracle $Result$ by the oracle $Result_{\mathcal{B}}$ which is the simulation of $Result$ by the blinder $\mathcal{B}$ (please see above the definition $\mathcal{B}$). Denote by G_1 the game such obtained. We prove that $P(G_0) = P(G_1)$.

If $\mathcal{A}$ queries the oracle $Result$ 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 $tr_\pi = (z, w, w')$ defined by a sequence of oracle queries $z \leftarrow SendTag(\emptyset, vtag)$, $w \leftarrow SendReader(z, \pi)$, $w' \leftarrow SendTag(w, vtag)$, and $SendReader(w', \pi)$ such that $vtag$ refers to some tag $\mathcal{T}_{ID}$ whose state is (K, x), $z = F_K(0, 0, x)$, (ID, K, x) is in the reader's database (that is, $\mathcal{T}_{ID}$ is legitimate), $w = F_K(0, 1, x + i)$ for some $i \in \{0, 1\}$, and $w' = F_K(1, 1, x + i + 1)$. 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 $tr_\pi = (z, w, w')$ 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 state is (K, x) and in DB there is a record (ID, K, x'). According to the description of the protocol, x' is either $x - 1$ or x. Because the oracles $SendReader$ and $SendTag$ are the real ones (and not simulated by blinder), the reader finds $i \in \{0, 1\}$ such that $z = F_K(0, 0, x - i)$. Therefore, w must be of the form $F_K(0, 1, x - i)$, and this value will match $F_K(0, 1, x)$ computed by tag. Therefore, the tag authenticates the reader and replies by $w' = F_K(1, 1, x + 1)$. But then, the reader will successfully check the equality between w and $F_K(1, 1, x - i + 1)$ (computed by itself) and, therefore, authenticates the tag. As a conclusion, $Result(\pi) = 1$.

This shows that $P(G_0) = P(G_1)$.

Game G_2: This game is identical to G_1 except that the $Launch()$ oracle is simulated according to the blinder description. No difference is encountered between the two games and,

therefore, $P(G_1) = P(G_2)$.

Game G_3: This game is identical to G_2 except that the $SendTag(\emptyset, vtag)$ oracle is simulated according to the blinder description. By doing this, $z = F_K(0, 0, x)$, where x is random, is replaced by a random $z \leftarrow \{0,1\}^{\ell_2}$. As F is a PRF, $\mathcal{A}$ will not see the difference, except with a negligible probability. Therefore, $|P(G_2) - P(G_3)|$ is negligible. The formal proof of this is quite straightforward. The main idea is as follows. Assume that an adversary $\bar{\mathcal{A}}$ can distinguish with non-negligible probability between G_2 and G_3. Define an adversary $\bar{\mathcal{A}}'$ for PRF that uses $\bar{\mathcal{A}}$ as a subroutine and sends $(0, 0, x)$ as a challenge. When the PRF challenger returns, with equal probability, either $z = F_K(0, 0, x)$ or $z \leftarrow \{0,1\}^{\ell_2}$, $\mathcal{A}'$ sends this value to $\bar{\mathcal{A}}$. The probability $\bar{\mathcal{A}}'$ guesses between the two possibilities for z is exactly the probability $\bar{\mathcal{A}}$ distinguishes between the two games.

Game G_4: This game is identical to G_3 except that the $SendReader(z, \pi)$ oracle is simulated according to the blinder description. That is, for each tag whose current state is (K, x), the reader answer $w = F_K(0, 1, x + i)$ or $w \leftarrow \{0,1\}^{\ell}_2$ is replaced by $w \leftarrow \{0,1\}^{\ell}_2$, where $i \in \{0, 1\}$. As F is a PRF, $\mathcal{A}$ will not notice this difference and, therefore, $P(G_3) - P(G_4)|$ is negligible. The formal proof is by contradiction and it is quite similar to the proof that establishes the transition from G_2 to G_3.

Game G_5: This game is identical to G_4 except that the $SendTag(w, vtag)$ oracle is simulated by blinder. That is, the tag answer $w' = F_K(1, 1, x)$ or $w' = F_K(1, 0, x)$ (using the notation in Figure 1) is replaced by $w' \leftarrow \{0,1\}^{\ell_2}$. As F is a PRF, 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 in Game G_4. Therefore, it is omitted.

Game G_6: This game is identical to G_5 except that the $SendReader(w', \pi)$ oracle is simulated by blinder. However, this does not change the probability distribution from G_5. Therefore, $P(G_5) = P(G_6)$.

It is straightforward to see that G_6 is in fact $RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}$. Now, to derive the final conclusion of the proof we remark that $P_{\mathcal{A}}(G_0) = P(RFID^{prv-0}_{\mathcal{A},\mathcal{S}_1}(\lambda) = 1)$ and $P_{\mathcal{A}}(G_6) = P(RFID^{prv-1}_{\mathcal{A},\mathcal{S}_1,\mathcal{B}}(\lambda) = 1)$. 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 weak privacy. □

6.2 Destructive privacy

We will discuss now on destructive privacy in the randomized Vaudenay's model. First, by an inspection of the proof of Theorem 1 in [25] we remark that the $Free$ oracle is not used at all. Therefore, this theorem works in this case as well. More precisely, we have the following result.

Theorem 6.4. There is no RFID scheme that achieve reader authentication and narrow forward privacy in the randomized Vaudenay's model with temporary state disclosure.

According to Theorem 6.4, any destructive private and mutual authentication RFID scheme in randomized Vaudenay's model with temporary state disclosure cannot be based only on cryptographic primitives.

Remark 6.2. According to Theorem 6.4, the scheme in [6] does not achieve more than weak privacy in the randomized Vaudenay's model.

As we did in [16], [17], PUF based constructions are the most natural solution to obtain schemes with a higher level of privacy than the weak one. It turns out that a very simple modification to the scheme in Figure 1 gives rise to a destructive private and reader-first authentication RFID scheme. More precisely, what we have to do it to generate the tag key by a PUF. We thus

obtain the scheme in Figure 3. As one can see, we endowed each tag with a unique PUF and a random seed s. The database will store a record (ID, K, x) for each tag with the identity ID and PUF P for which $K = P(s)$, s being the PUF seed. Everything else is similar to the scheme in Figure 1. Remark 6.1 also applies to this new scheme with the difference that, in this case, each tag evaluates the PRF F and its proper PUF. As with respect to security and privacy, we have the following result.

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

Fig. 3. Stateful RFID scheme with constant tag identifiers that achieves destructive privacy and reader-first authentication in the randomized Vaudenay's model with temporary state disclosure

Theorem 6.5. The RFID scheme in Figure 3 achieves reader-first authentication in Vaudenay's model and destructive privacy in the randomized Vaudenay's model with temporary state disclosure, provided that F is a PRF and the tags are endowed with ideal PUFs.

Proof: For tag and reader authentication, the proofs are similar to those in Theorem 6.1 and 6.2 (the existence of PUFs on tags do not change anything, as PUFs are ideal), with the following three amendments:

- (item (2) in proof of Theorem 6.1) The adversary $\mathcal{A}'$ 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";
- (item (4a) in proof of Theorem 6.1) The key K generated by $\mathcal{A}'$ plays the role of the tag's internal PUF value $P(s)$. 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 $\mathcal{T}_{ID^*}$);
- $\mathcal{A}'$ has to simmulate the $Corrupt(vtag)$ oracle as well. This is done as follows: if the tag referred by $vtag$ is different from ID^*, then $\mathcal{A}'$ returns its current state; otherwise, it aborts.

As with respect to destructive privacy, we add one more game in between G_0 and G_1 in the proof of Theorem 6.3. This game, denoted G'_0, aims to replace the PUF values (keys) by random values (keys).

Game G'_0: 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 replaced by the uniform probability distribution (in G'_0). 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 $|P(G_0) - P(G'_0)|$ 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'_0 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, 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, x) for this tag and sends s 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.

After some time, $\mathcal{A}$ will output a guess $b' \in \{0,1\}$ about the game it thinks it is playing ($b = 0$ for G_0 and $b = 1$ for G'_0). 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'_0. If this is non-negligible, then $\mathcal{A}'$ has non-negligible probability to break the PUF.

After inserting G'_0 in between G_0 and G_1, we continue with the sequence of games as in the proof of Theorem 6.3. At the end, we have to prove one more thing, namely that G_6 is in fact $RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}$. This was straightforward in the proof of Theorem 6.3; it is not difficult here too, but some arguments should be provided.

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_6 is indeed $RFID^{prv-1}_{\mathcal{A},\mathcal{S},\mathcal{B}}$.

The proof can now be completed as for Theorem 6.3. □

One may compare our scheme in Figure 3 against the scheme in Figure 4 in [17]. In that scheme, x is updated in the first step of the protocol. This makes loosing synchronization between tag and reader (and the schemes becomes narrow). However, the update of x in the first step randomizes the tag state so that Lemma 4.1 cannot be applied. As the key is PUF protected, the scheme reaches narrow destructive privacy in Vaudenay's model with temporary state disclosure.

In the scheme in Figure 3, the variable x is updated in last tag step, which makes the scheme avoid desynchronization (well, it is just one step desynchronization) and falling into the narrow privacy class. However, not updating x in the first step makes Lemma 4.1 apply to this scheme. As the key is PUF protected, the scheme reaches destructive privacy in the randomized Vaudenay's model with temporary state disclosure.

Remark 6.3. The scheme in [10] achieves destructive privacy in the randomized Vaudenay's model (the proof in [10] is only sketched), but it is more costly than the scheme in Figure 3

7 Conclusions

In order to efficiently search for tags in the reader's database, various RFID schemes have been proposed [6]–[11], where the tag's first message to the reader includes a "constant" piece of information. This piece of information, called *tag identifier* in our paper, needs to remain constant until the tag is identified, and then it may be updated. The privacy of such RFID schemes have been studied either in an informal way or by ad-hoc models, making it difficult to compare each other from a privacy point of view.

In this paper we have introduced the class of *stateful RFID schemes with constant tag identifiers* as a unifying approach for schemes as those mentioned above. We have shown that these kind of schemes do not achieve any level of privacy in Vaudenay's model. However, they are quite suitable in some practical applications, and the "quantity of privacy they lose" (measured in Vaudenay's model) might not affect at all their applicability. This fact shows that Vaudenay's model is probably not the most appropriate to analyze privacy of such schemes. Therefore, we have proposed a modified model, called the *randomized Vaudenay's model*, which seems very appropriate to analyze this class of schemes. Thus, we were able to redefine the same eight privacy levels as in Vaudenay's model. In addition to this, we have proposed two RFID schemes that achieve weak and respectively destructive privacy in the randomized Vaudenay's model.

Based on this privacy model, we can now compare stateful RFID schemes with constant tag identifiers with respect to the privacy level offered by each one. The table in Figure 4 provides such a comparison.

RFID scheme	Primitives and number of evaluations	Security	Privacy in randomized Vaudenay's model
[6] (2005)	3 Hash + RNG	Yes	At most Weak Private
[9] (2010)	4 Hash + RNG	No	No privacy
This paper (Section 6.1)	3 PRF	Yes	Weak Private
[10] (2018)	4 PRF + 2 PUF	Yes	Destructive Private
This paper (Section 6.2)	3 PRF + 2 PUF	Yes	Destructive Private

Fig. 4. Comparisons between stateful RFID schemes with constant tag identifiers in the randomized Vaudenay's model

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] 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.

[4] 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.

[5] B. Preneel. (2018, Feb) Cryptography best practices. [Online]. Available: https://secappdev.org/handouts-2018.html

[6] T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," in *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, ser. SECURECOMM '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 59–66. [Online]. Available: http://dx.doi.org/10.1109/SECURECOMM.2005.4

[7] G. Tsudik, "YA-TRAP: Yet another trivial RFID authentication protocol," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, ser. PERCOMW '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 640–. [Online]. Available: http://dx.doi.org/10.1109/PERCOMW.2006.152

[8] B. Alomair, L. Lazos, and R. Poovendran, *Securing low-cost RFID systems: An unconditionally secure approach*, ser. Cryptology and Information Security Series, 2010, vol. 4, pp. 1–17.

[9] L. Lu, Y. Liu, and X.-Y. Li, "Refresh: Weak privacy model for RFID systems," in *Proceedings of the 29th Conference on Information Communications*, ser. INFOCOM'10. Piscataway, NJ, USA: IEEE Press, 2010, pp. 704–712. [Online]. Available: http://dl.acm.org/citation.cfm?id=1833515.1833640

[10] C. Hristea and F. L. Ţiplea, "A PUF-based destructive private mutual authentication RFID protocol," in *Innovative Security Solutions for Information Technology and Communications*, J.-L. Lanet and C. Toma, Eds. Cham: Springer International Publishing, 2019, pp. 331–343.

[11] P. Gope, J. Lee, and T. Q. S. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, Nov 2018.

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

[13] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 2nd ed. Chapman & Hall/CRC, 2014.

[14] R. Maes, *Physically Unclonable Functions: Constructions, Properties and Applications*. Springer Verlag, 2013.

[15] 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.

[16] 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.

[17] F. L. Ţiplea and C. Hristea, "Privacy and reader-first authentication in Vaudenay's RFID model with temporary state disclosure," Cryptology ePrint Archive, Report 2019/113, 2019, https://eprint.iacr.org/2019/113.

[18] A. Silberschatz, H. Korth, and S. Sudarshan, *Database Systems Concepts*, 6th ed. New York, NY, USA: McGraw-Hill Education, Inc., 2010.

[19] B. Alomair, A. Clark, J. Cuéllar, and R. Poovendran, "Scalable RFID systems: A privacy-preserving protocol with constant-time identification," *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 8, pp. 1536–1550, 2012.

[20] 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.

[21] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Efficient hash-chain based RFID privacy protection scheme," in *International Conference on Ubiquitous Computing (Ubicomp 2004), Workshop "Ubicomp Privacy: Current Status and Future Directions"*, Sept 2004.

[22] G. Avoine and P. Oechslin, "A scalable and provably secure hash-based RFID protocol," in *Third IEEE International Conference on Pervasive Computing and Communications Workshops*, March 2005, pp. 110–114.

[23] 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.

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

[25] 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.