Untangling RFID Privacy Models

Iwen Coisel* and Tania Martin

Université catholique de Louvain
ICTEAM/Crypto Group and ICTEAM/GSI
B-1348 Louvain-La-Neuve, Belgium
{iwen.coisel, tania.martin}@uclouvain.be

Abstract. This article investigates privacy in Radio Frequency IDentification (RFID) systems. We survey the eight most well-known RFID privacy models. We examine their advantages and drawbacks, and provide a comprehensive comparison of these models. The first conclusion is that none of these models is complete, and the association of all their positive features does not either provide the best privacy model. Then, we give a privacy analysis of five distinct RFID authentication protocols according to these models, and exhibit a main worry: no model is able to distinguish perfectly all the five RFID protocols regarding privacy.

1 Introduction

Radio Frequency IDentification (RFID) is the latest popular technology used to identify and/or authenticate remote objects or persons, through a radio frequency channel using devices called RFID readers and RFID tags. In a simple manner, a tag (i.e. a transponder composed of a micro-circuit and an antenna) is embedded into an object, and is questioned by a reader when it comes into the reader's electromagnetic field. Even if RFID had been introduced as a recent technology, the beginning of its use goes back to the earlier 1940's, during World War II, when the Royal Air Force deployed the IFF (Identify Friend or Foe) system. It enabled to identify a friend aircraft from an enemy. Today, RFID is more and more exploited in many domains such as library management, pet identification, anti-theft cars, anti-counterfeiting, ticketing in public transportation, access control, or even biometric passports.

As predictable, some problems come up with this large-scaled deployment. One of them is related to the contactless feature of RFID. The messages exchanged between tags and readers can easily be intercepted by an adversary, by only eavesdropping. Another problem is related to the computation capabilities of RFID tags. Indeed, RFID aims to be as cheap as possible. The problem, nowadays, is that low-cost tags do not use strong cryptographic building blocks (such as hash functions or public-key cryptography), and therefore may be more vulnerable to attacks. An example is the case of the NXP Mifare Classic tag. NXP decides to use a proprietary cryptographic building block (for obscure reasons). The result is that, in 2008, the Mifare Classic technology has been totally broken by several research groups [15, 21, 29, 35]. Such fragilities can damage the RFID system, as users' tracking or, even worst, impersonation of the customers can consequently be carried out by such adversaries. Thus, secure authentication protocols are necessary to protect the use of RFID.

Many articles are published every day about the dangers of RFID with respect to privacy. The problem of information disclosure is real when the data sent by a tag reveal some information about its holder (called "information leakage"), but also when the eavesdropping of communications allows to track a tag at different places or times (called "malicious traceability"), and consequently its holder.

This problem is a major concern for companies using RFID. An example is the public transportation of Paris (i.e., "Régie Autonome des Transports Parisiens" (RATP)). The RATP has built its ticketing system with the RFID Calypso solution [26]. Since few months, the RATP offers to its users to enhance the security of their personal data included in their season ticket for 5 additional euros during the ticket purchase. However, the user member to this service does not receive any information about the methods used to protect his

* This work was partially funded by the Walloon Region Marshall plan through the 816922 Project SEE

data. He furthermore does not have any guarantee that these methods are actually implemented. This raises a feeling of discontent from the public transportation customers. A second example is the case of biometric passports, put into circulation in 2004: privacy (in particular malicious traceability) was a concept that was not so much a worry at the beginning of their conception while it is nowadays considered as a major concern.

Various researches have emerged these last years to fight against information leakage and malicious traceability in RFID, but the search for a generic solution that is efficient, secure, and that can be implemented in reasonably-costly tags remains open [3, 7, 33]. The proposed solutions are designed empirically and have only been analyzed with ad hoc methods that do not detect all their weaknesses. The flaws are usually only discovered after that the products are put into circulation. This was for example the case of passport as demonstrated in [4].

In parallel, many investigations have been done to formalize the privacy in RFID. In 2005, Avoine was the first to present a privacy model [1]. Since then, many attempts [8, 11, 16, 18, 24, 25, 28, 30, 31, 37, 39] have been done to propose a convenient and appropriate privacy model for RFID. But each one suffers from distinct shortcomings. For instance, these models generally do not take into account some important adversary features, such that the behavior of a "corrupted" tag (can it still be used in the system?). Given all these proposals for privacy analysis, Vaudenay's model [39], presented at ASIACRYPT 2007, is known to be nowadays one of the most evolved.

In this article, our contribution is threefold. First, we present and discuss the advantages and weaknesses of eight existing privacy models (Sections 3 to 10). All these models are design for analyzing identification/authentication protocols preserving anonymity. Some of them are very popular like Avoine's model [1], Juels and Weis' model [28] or Vaudenay's model [39], whereas other ones have interesting features like Deng, Li, Yung and Zhao [16], or van Deursen, Mauw and Radomirovic [18]. Then, we compare in Section 11 all these models based on different expected properties (e.g., the levels of adversary's power). We thus expose the powerful and poor attributes of each model, and deduce that none of the presented models can be considered as the best one. Finally, we analyze five different authentication protocols, regarding each model (Section 12). We show that none of these models is able to correctly differentiate the privacy levels of the given protocols.

2 Common Definitions

In this section, we give all the common definitions that are used in the presented privacy models.

2.1 The RFID System

For all the privacy models, an RFID system $\mathcal{S}$ is composed of three kind of entities: tags, readers and a centralized database. It is generally considered that the database and the readers are connected online all together through a secure channel, and therefore they form one unique entity, the reader.

We will denote $\mathcal{T}$ as a tag, $\mathcal{R}$ as the reader, and DB as the reader's database. A tag $\mathcal{T}$ is able to communicate with $\mathcal{R}$, when it enters into $\mathcal{R}$'s electromagnetic field. Then both reader and tag can participate together to an RFID protocol instance π. This protocol can be an identification or an authentication protocol. We define a i-pass RFID protocol as being a protocol where i messages are exchanged between $\mathcal{R}$ and a tag.

The reader $\mathcal{R}$ is a powerful transreceiver device. Its computation capabilities approach the ones of a small computer. A tag $\mathcal{T}$ is a transponder with identifier $\mathsf{ID}_{\mathcal{T}}$. Its memory is around a few Kbytes. Its computation capabilities are generally much lower than a reader, but, depending on the tag, it can perform for instance logic operations, symmetric-key cryptography, hash functions, or even public-key cryptography. The database DB stores, at least, the identifier $\mathsf{ID}_{\mathcal{T}}$ and potentially a secret $\mathsf{k}_{\mathcal{T}}$ of each legitimate tag $\mathcal{T}$ involved in the system.

2.2 Basic Definitions

First, we define λ as the security parameter of the system $\mathcal{S}$, and $poly(.)$ as a polynomial function. Thus, we define $\epsilon(\lambda) : \mathbb{N} \rightarrow \mathbb{R}$ as being a negligible function in λ if, for every positive function $poly(.)$, there exists an integer N such that, for all $\lambda > N$, $|\epsilon(\lambda)| < \frac{1}{poly(\lambda)}$.

Then, we define all the different entities that may play a role in the presented privacy models. An *adversary* $\mathcal{A}$ of the system is a malicious entity whose aim is to perform some attack, either through the wireless communications between readers and tags (e.g., eavesdropping), or on the RFID devices themselves (e.g., corruption a device and getting all the information stored on it). The adversary advantage is the success measure of an attack performed by $\mathcal{A}$. In some models, $\mathcal{A}$ is requested to answer to a kind of riddle, which is determined by an honest entity, called *challenger* $\mathcal{C}$. A *challenge tag* is a tag which is suffering from an attack performed by $\mathcal{A}$. It can be chosen either by $\mathcal{A}$ or by $\mathcal{C}$.

A modelization with oracles is used to represent the possible interactions between $\mathcal{A}$ and the system. Thus, $\mathcal{A}$ carries out her attack on the system, performing some queries to the oracles that simulate the system. The generic oracles used in all the presented privacy models are detailed in Section 2.4.

We consider that $\mathcal{A}$ is able to play/interact with a tag when the latter is in $\mathcal{A}$'s neighborhood. Generally, this happens when the tag is within her visual contact. At that moment, the tag is called by its temporary name $\mathcal{T}$ (not by its identifier $\mathsf{ID}_{\mathcal{T}}$). During an attack, if a tag goes out and comes back to $\mathcal{A}$'s view, then it is considered that its temporary name has changed. This notion is explained more in detail in Vaudenay's model [39] (see Section 5). The same case happens when a set of tags is given to the challenger $\mathcal{C}$: when $\mathcal{C}$ gives back the tags to $\mathcal{A}$, their temporary names are changed.

2.3 Procedures

Most of the models studied in this survey focus on an RFID system, denoted $\mathcal{S}$, based on an anonymous identification protocol implying a single reader and several tags. The system is generally composed of several procedures, either defining how to set up the system, the reader and the tags, or defining the studied protocol. One way to formalize these procedures is done below. Note that this is just a generalization but it can be different in some models.

- $\mathsf{SetupReader}(1^{\lambda})$ is a setup scheme which defines $\mathcal{R}$'s parameters (e.g., generating a private/public key pair $(\mathsf{K_S}, \mathsf{K_P})$) depending on the security parameter λ. It also creates an empty database DB which will later contain, at least, the identifiers and secrets of all tags.
- $\mathsf{SetupTag}_{\mathsf{K_P}}(\mathsf{ID}_{\mathcal{T}})$ is a polynomial-time algorithm which returns $\mathsf{k}_{\mathcal{T}}$, i.e. the secret $\mathsf{k}_{\mathcal{T}}$ of the tag $\mathcal{T}$ with identifier $\mathsf{ID}_{\mathcal{T}}$. $(\mathsf{ID}_{\mathcal{T}}, \mathsf{k}_{\mathcal{T}})$ is stored in the reader's database DB when the tag is legitimate.
- Ident is a polynomial-time interactive protocol between the reader $\mathcal{R}$ and a tag $\mathcal{T}$, where $\mathcal{R}$ ends with a private tape Output. At the end of the protocol, the reader either accepts the tag (if legitimate) and $\mathsf{Output} = \mathsf{ID}_{\mathcal{T}}$, or rejects it (if not) and $\mathsf{Output} = \perp$.

2.4 The Generic Oracles

An adversary $\mathcal{A}$ is able to interact/play with the system with the following oracles. First, she can setup a new tag of identifier $\mathsf{ID}_{\mathcal{T}}$.

- CreateTag($\mathsf{ID}_{\mathcal{T}}$): creates a tag $\mathcal{T}$ with a unique identifier $\mathsf{ID}_{\mathcal{T}}$. This oracle uses $\mathsf{SetupTag}_{\mathsf{K_P}}$ to set up the tag. It updates DB, adding this new tag.

  $\mathcal{A}$ can ask for a full execution of the protocol on a tag $\mathcal{T}$.

- Execute($\mathcal{T}$) $\rightarrow (\pi, \mathsf{transcript})$: executes an Ident protocol between $\mathcal{R}$ and $\mathcal{T}$. This oracle outputs the $\mathsf{transcript}$ of the protocol instance π, that is the whole list of the successive messages of the instance π.

  Also, she can decompose a protocol execution, combining the following oracles.

- LAUNCH() $\rightarrow \pi$: makes $\mathcal{R}$ start a new Ident protocol instance π.
- SENDREADER$(m, \pi) \rightarrow r$: sends a message m to $\mathcal{R}$ in the protocol instance π. It outputs the response r of the reader.
- SENDTAG$(m, \mathcal{T}) \rightarrow r$: sends a message m to $\mathcal{T}$. It outputs the response r of the tag.

Then, $\mathcal{A}$ can ask for the reader's result of a protocol instance π.

- RESULT$(\pi) \rightarrow x$: when π is completed, it outputs $x = 1$ if Output $\neq \perp$, and $x = 0$ otherwise.

And finally, she can ask the corruption of a tag $\mathcal{T}$ in order to recover its secret.

- CORRUPT$(\mathcal{T}) \rightarrow \mathsf{k}_{\mathcal{T}}$: returns the current secret $\mathsf{k}_{\mathcal{T}}$ of $\mathcal{T}$.

If the conditions of the oracles' uses are not respected, then the oracles return $\perp$. Note that these definitions are generic ones. Some models do not use exactly the same generic oracles: in those cases, they will be redefined.

3 Avoine [1], 2005

Avoine is the first one who proposes a privacy model for RFID. It is based on the untraceability notion.

Since $\mathcal{T}$ and $\mathcal{R}$ can run several protocol instances, Avoine defines them as follows: $\pi_{\mathcal{T}}^i$ denotes the i^{th} instance of $\mathcal{T}$, and $\pi_{\mathcal{R}}^j$ denotes the j^{th} instance of $\mathcal{R}$. These notations favor the precise description of $\mathcal{R}$'s and $\mathcal{T}$'s lifetime.

Avoine considers that each tag has a unique and independent secret, and that, at the initialization of the system, DB already stores all the tags' secrets, i.e., a SetupTag has already been performed on every tag.

In his model, Avoine only considers 3-pass RFID protocols as depicted in Fig. 1.

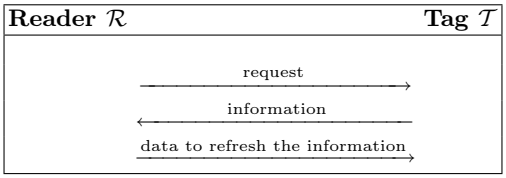

Fig. 1. Avoine's 3-pass RFID protocol.

3.1 The Oracles

In this model, $\mathcal{A}$ only has access to the following modified generic oracles. The first modification is that the oracles are adapted to only be used on 3-pass protocols. The second modification is that, instead of using the entities' names, Avoine uses the protocol instances names.

- SENDTAG$(m_1, m_3, \pi_{\mathcal{T}}^i) \rightarrow r$: sends a request m_1 to $\mathcal{T}$, and then $\mathcal{A}$ sends the message m_3 after receiving $\mathcal{T}$'s answer r. This is done during the instance $\pi_{\mathcal{T}}^i$ of $\mathcal{T}$.
- SENDREADER$(m_2, \pi_{\mathcal{R}}^j) \rightarrow r$: sends the message m_2 to $\mathcal{R}$ in the protocol instance $\pi_{\mathcal{R}}^j$. It outputs $\mathcal{R}$'s answer r.
- EXECUTE$(\pi_{\mathcal{T}}^i, \pi_{\mathcal{R}}^j) \rightarrow$ (transcript): executes a whole instance of the protocol between $\mathcal{T}$ and $\mathcal{R}$. This is done during the instance $\pi_{\mathcal{T}}^i$ of $\mathcal{T}$ and the instance $\pi_{\mathcal{R}}^j$ of $\mathcal{R}$. $\mathcal{A}$ obtains the whole transcript.
- EXECUTE$^*(\pi_{\mathcal{T}}^i, \pi_{\mathcal{R}}^j) \rightarrow (\mathcal{R}$-transcript): this is the same as the normal EXECUTE. But it only returns the $\mathcal{R}$-transcript, i.e. the messages sent by $\mathcal{R}$.
- CORRUPT$(\pi_{\mathcal{T}}^i) \rightarrow \mathsf{k}_{\mathcal{T}}$: returns the the current secret $\mathsf{k}_{\mathcal{T}}$ of $\mathcal{T}$ when the tag is in its i^{th} instance.

Two remarks are of interest for the CORRUPT oracle. First, CORRUPT can be used only once by $\mathcal{A}$. After this oracle query, $\mathcal{A}$ can no more use the other oracles. Second, CORRUPT is called on the tag instance number, and not the tag itself. This allows $\mathcal{A}$ to specify exactly the targeted moment of the tag's life.

During her attack, $\mathcal{A}$ has access to the oracles $\mathcal{O} \subset \{\text{T, R, E, E}^*\text{, C}\} = \{\text{SENDTAG}, \text{SENDREADER}, \text{EXECUTE}, \text{EXECUTE}^*, \text{CORRUPT}\}$.

Avoine denotes $\omega_i(\mathcal{T})$ as being the result of an oracle call on $\mathcal{T}$: therefore $\omega_i(\mathcal{T}) \in \{\text{SENDTAG}(*, *, \pi^i_{\mathcal{T}}), \text{EXECUTE}(\pi^i_{\mathcal{T}}, *), \text{EXECUTE}^*(\pi^i_{\mathcal{T}}, *), \text{CORRUPT}(\pi^i_{\mathcal{T}})\}$. Avoine defines an *interaction* $\Omega_I(\mathcal{T})$ as being a set of executions on the same tag $\mathcal{T}$ during an interval I when $\mathcal{A}$ can play with $\mathcal{T}$. Formally, $\Omega_I(\mathcal{T}) = \{\omega_i(\mathcal{T}) |\ i \in I\} \cup \{\text{SENDREADER}(*, \pi^j_*) |\ j \in J\}$, where $I, J \subset \mathbb{N}$. By this definition, the length of $\Omega_I(\mathcal{T})$ is $|I|$.

Avoine also defines a function *Oracle* which takes as parameters a tag $\mathcal{T}$, an interval I and the oracles $\mathcal{O}$, and which outputs the interaction $\hat{\Omega}_I(\mathcal{T})$ that maximizes $\mathcal{A}$'s advantage.

3.2 Untraceability Experiments

λ_{ref} and λ_{chal} represent respectively a reference length and a challenge length, and are function of the security parameter λ.

The first experiment works as follows. First, $\mathcal{A}$ receives the interactions of a tag $\mathcal{T}$ during an interval I that she chooses. Then, she receives the interactions of the challenge tags $\mathcal{T}_0$ and $\mathcal{T}_1$, also during the intervals I_0 and I_1 that she chooses, such that $\mathcal{T} = \mathcal{T}_0$ or $\mathcal{T}_1$. This last information is unknown to $\mathcal{A}$. Additionally here, none of these two intervals I_0 and I_1 crosses the interval I of $\mathcal{T}$. At the end, $\mathcal{A}$ has to decide which one of the challenge tags is the tag $\mathcal{T}$.

Experiment $Exp^{\mathsf{Existential-UNT}}_{\mathcal{S},\mathcal{A}}[\lambda_{\text{ref}}, \lambda_{\text{chal}}, \mathcal{O}]$

1. $\mathcal{C}$ initializes the system $\mathcal{S}$.
2. $\mathcal{A}$ requests $\mathcal{C}$ to receive a tag $\mathcal{T}$.
3. $\mathcal{A}$ chooses I, calls $Oracle(\mathcal{T}, I, \mathcal{O})$ where $|I| \leq \lambda_{\text{ref}}$, and then receives $\hat{\Omega}_I(\mathcal{T})$.
4. $\mathcal{A}$ requests $\mathcal{C}$ to receive two challenge tags $\mathcal{T}_0$ and $\mathcal{T}_1$, such that $\mathcal{T} = \mathcal{T}_0$ or $\mathcal{T}_1$.
5. $\mathcal{A}$ chooses I_0 and I_1 such that $|I_0| \leq \lambda_{\text{chal}}$, $|I_1| \leq \lambda_{\text{chal}}$, and $(I_0 \cup I_1) \cap I = \emptyset$
6. $\mathcal{A}$ calls $Oracle(\mathcal{T}_0, I_0, \mathcal{O})$ and $Oracle(\mathcal{T}_1, I_1, \mathcal{O})$, and then receives $\hat{\Omega}_{I_0}(\mathcal{T}_0)$ and $\hat{\Omega}_{I_1}(\mathcal{T}_1)$.
7. $\mathcal{A}$ decides which of $\mathcal{T}_0$ or $\mathcal{T}_1$ is $\mathcal{T}$, and outputs a guess bit b.

$Exp^{\mathsf{Existential-UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{T} = \mathcal{T}_b$.

The second experiment has the same mechanism. The only difference is that, now, $\mathcal{C}$ is the one who chooses the intervals I_0 and I_1 of the challenge tags, and not $\mathcal{A}$ anymore.

Experiment $Exp^{\mathsf{Universal-UNT}}_{\mathcal{S},\mathcal{A}}[\lambda_{\text{ref}}, \lambda_{\text{chal}}, \mathcal{O}]$

1. $\mathcal{C}$ initializes the system $\mathcal{S}$.
2. $\mathcal{A}$ requests $\mathcal{C}$ to receive a tag $\mathcal{T}$.
3. $\mathcal{A}$ chooses I, calls $Oracle(\mathcal{T}, I, \mathcal{O})$ where $|I| \leq \lambda_{\text{ref}}$, and then receives $\hat{\Omega}_I(\mathcal{T})$. Here I is known by $\mathcal{C}$.
4. $\mathcal{A}$ requests $\mathcal{C}$ to receive two challenges $\mathcal{T}_0$, $\mathcal{T}_1$, I_0 and I_1, such that $\mathcal{T} = \mathcal{T}_0$ or $\mathcal{T}_1$.
5. $\mathcal{A}$ calls $Oracle(\mathcal{T}_0, I_0, \mathcal{O})$ and $Oracle(\mathcal{T}_1, I_1, \mathcal{O})$, and then receives $\hat{\Omega}_{I_0}(\mathcal{T}_0)$ and $\hat{\Omega}_{I_1}(\mathcal{T}_1)$.
6. $\mathcal{A}$ decides which of $\mathcal{T}_0$ or $\mathcal{T}_1$ is $\mathcal{T}$, and outputs a guess bit b.

$Exp^{\mathsf{Universal-UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{T} = \mathcal{T}_b$.

3.3 Untraceability Notions

From the experiments defined above, the notions of $\mathsf{Existential-UNT}$ and $\mathsf{Universal-UNT}$ are extended in this model, depending on restrictions about the choices of I_0 and I_1. $\mathsf{Existential-UNT}$ is when $\mathcal{A}$ chooses I_0 and I_1, whereas $\mathsf{Universal-UNT}$ is when $\mathcal{C}$ chooses them. Then, if $I < I_0, I_1$ (resp. $I > I_0, I_1$), that means I_0 and I_1 take place after (resp. before) I, with respect to the lifetime of the system.

– If $\mathcal{A}$ (resp. $\mathcal{C}$) chooses I_0 and I_1 such that $I < I_0, I_1$, then it is denoted $\mathsf{Existential}^+$ (resp. $\mathsf{Universal}^+$).
– If $\mathcal{A}$ (resp. $\mathcal{C}$) chooses I_0 and I_1 such that $I > I_0, I_1$, then it is denoted $\mathsf{Existential}^-$ (resp. $\mathsf{Universal}^-$).

The notion of $\mathsf{Universal}^-$ when the CORRUPT oracle is used is called $\mathsf{Forward-UNT}$.

Definition 1 (Untraceability). *An RFID system $\mathcal{S}$ is said $P-\mathsf{UNT}-\mathcal{O}$ (for $P \in \{\mathsf{Existential}, \mathsf{Forward}, \mathsf{Universal}\}$) if, for every adversary $\mathcal{A}$:*

$$|\Pr(Exp^{P-\mathsf{UNT}}_{\mathcal{S},\mathcal{A}}[\lambda_{\text{ref}}, \lambda_{\text{chal}}, \mathcal{O}] \text{ succeeds}) - 1/2| \leq \epsilon(\lambda_{\text{ref}}, \lambda_{\text{chal}}).$$

Direct implications are made from these notions:

$$\boxed{\mathsf{Existential-UNT}-\mathcal{O} \Rightarrow \mathsf{Forward-UNT}-\mathcal{O} \Rightarrow \mathsf{Universal-UNT}-\mathcal{O}}$$

3.4 Discussion

As remarks, this model is the first one designed for RFID systems. It follows the idea of communication intervals: $\mathcal{A}$ asks some oracles' queries on specific intervals for the tags involved in the experiment. Furthermore, these tags can only be legitimate.

One of the good features of this model is that it presents the notion of forward untraceability, thanks to the CORRUPT oracle which allows $\mathcal{A}$ to obtain a tag's secret. This is linked to the restriction done on CORRUPT: it is not possible to use any other oracle after a CORRUPT query. Also, the adversary is able to perform several untraceability goals/experiments (i.e. she can be either Existential, or Forward, or Universal). The model is thus flexible.

Yet, several drawbacks appear. Following the definition and the use of CORRUPT in this model, $\mathcal{A}$ is not always able to corrupt the challenge tags, since no other oracle can be used after. Another weakness is that the model only allows $\mathcal{A}$ to interact with two tags, and furthermore, these tags are not chosen by $\mathcal{A}$. Thus, Juels and Weis showed in [28] that Avoine's model cannot catch an important attack on systems where tags have correlated secrets. Furthermore, the oracles to interact with the system can only be used for 3-pass protocols. Therefore, as it is defined, it is not possible to analyze every kind of identification/authentication protocols with the model, but it can be adapted (e.g., to analyze a 2-pass classical challenge-response). Additionally, Avoine is the first author who propose an RFID untraceability model and therefore did not have hindsight regarding all the possible attacks that can be performed on a protocol. This is why this model does not capture all the relevant information that can be extracted from a protocol execution. For instance, this model does not consider that $\mathcal{A}$ has access to the oracle RESULT.

4 Juels and Weis [28], 2007

Juels and Weis propose a privacy model based on indistinguishability of tags. During the sequel, we refer this model as JW's model.

At the initialization of the system, DB already stores the contents of all the tags, i.e., a SetupTag has already been performed on every tag.

The RFID protocol considered by the authors is a classical challenge/response protocol, with possible additional messages in order to update the tags keys.

4.1 Oracles

$\mathcal{A}$ has access to the generic oracles LAUNCH, SENDTAG. She has also access to SENDREADER, with the difference that the output of SENDREADER also includes the output of RESULT. She has furthermore access to the following oracle.

– TAGINIT($\mathcal{T}$) $\rightarrow \pi$: when $\mathcal{T}$ receives this query, it begins a new protocol instance π and deletes the information related to an existing instance.

– SetKey($\mathcal{T}, \mathsf{k}_{\mathcal{T}}^{\text{new}}$) $\rightarrow \mathsf{k}_{\mathcal{T}}$: when $\mathcal{T}$ receives this query, it outputs its current key $\mathsf{k}_{\mathcal{T}}$, and replaces it by a new one, $\mathsf{k}_{\mathcal{T}}^{\text{new}}$.

The SetKey oracle can be clearly associated to the generic Corrupt oracle. The main difference is that SetKey replace the tag's current key by another one.

4.2 Privacy Experiment

Let r, s and t be respectively the numbers of Launch, computation steps (represented by the SendReader and SendTag queries) and TagInit that are allowed to $\mathcal{A}$. Let n be the total number of tags involved in the system $\mathcal{S}$.

Experiment $Exp_{\mathcal{S},\mathcal{A}}^{\mathsf{JW\text{-}priv}}[\lambda, n, r, s, t]$

Setup:

1. $\mathcal{C}$ initializes the system $\mathcal{S}$.

Phase 1 (Learning):

2. $\mathcal{A}$ may do the following in any interleaved order:
   a. Make Launch calls, without exceeding r overall calls.
   b. Make TagInit calls, without exceeding t overall calls.
   c. Make arbitrary SetKey calls to any $(n-2)$ tags.
   d. Make SendReader and SendTag, without exceeding s overall steps.

Phase 2 (Challenge):

3. $\mathcal{A}$ selects two challenge tags $\mathcal{T}_i$ and $\mathcal{T}_j$ to which she did not send SetKey queries.
4. Let $\mathcal{T}_0^* = \mathcal{T}_i$ and $\mathcal{T}_1^* = \mathcal{T}_j$, and remove both from the current tag set.
5. $\mathcal{C}$ chooses a bit b at random, and provides $\mathcal{A}$ access to $\mathcal{T}_b^*$.
6. $\mathcal{A}$ may do the following in any interleaved order:
   a. Make Launch calls, without exceeding r overall calls.
   b. Make TagInit calls, without exceeding t overall calls.
   c. Make arbitrary SetKey calls to any tag in the current tag set, *except* $\mathcal{T}_b^*$.
   d. Make SendReader and SendTag, without exceeding s overall steps.
7. $\mathcal{A}$ outputs a guess bit b'.

$Exp_{\mathcal{S},\mathcal{A}}^{\mathsf{JW\text{-}priv}}$ succeeds if $b = b'$.

4.3 Privacy Notions

Definition 2 ((r, s, t)-Privacy). *A protocol initiated by $\mathcal{R}$ in an RFID system $\mathcal{S}$ with security parameter λ is* (r, s, t)-private *if, for every adversary $\mathcal{A}$:*

$$\left|\Pr(Exp_{\mathcal{S},\mathcal{A}}^{\mathsf{JW\text{-}priv}}[\lambda, n, r, s, t] \text{ succeeds}) - \frac{1}{2}\right| \leq \epsilon(\lambda).$$

Considering a variant of experiment $Exp_{\mathcal{S},\mathcal{A}}^{\mathsf{JW\text{-}priv}}$, where the "*except* $\mathcal{T}_b^*$" is removed from step (6.c), then forward-(r, s, t)-privacy can be defined in the same way as the previous definition.

4.4 Discussion

The authors only aim to analyze protocols based on symmetric-key cryptography. But this model can nevertheless be used to analyze protocols with public-key cryptography. Note that, as for Avoine's model, the tags created during the experiment can only be legitimate.

In this model, several good points can be extracted. First, the model is intuitive and easy to use. The privacy experiment is also precisely detailed with two distinct phases. Then, $\mathcal{A}$ can play/interact with all the tags during the experiment, except $\mathcal{T}^*_{b\oplus 1}$ during the challenge phase. Thus, this model is able to analyze protocols with correlated secrets, whereas it is not the case for Avoine's model. The authors make the experiment $Exp^{\mathsf{JW\text{-}priv}}_{\mathcal{S},\mathcal{A}}$ flexible, in order to catch the notion of forward privacy. Also, $\mathcal{A}$ chooses the challenge tags. She is a stronger adversary than Avoine's one, which makes the privacy notion also stronger.

This is the first model which introduces the notion of "side information bit" for the adversary. Here, it is related to the fact that $\mathcal{A}$ is able to know whether the reader authenticates successfully the tag or not. JW implicitly introduces this notion with the output of SendReader which includes this side information. This simple information allows to model a special kind of attacks on desynchronizable protocols like OSK, as explained in Section 12.2, and in [10].

Even if the model is generally clear, the result of a SetKey query on a tag is ambiguous. It is obvious that SetKey is equivalent to the Corrupt oracle given in Section 2.4 in the sense that it reveals to $\mathcal{A}$ the tag's current key. But there is no information on what happens if $\mathcal{A}$ tries to put an illegitimate key on a tag: is this new key registered in the reader's database? Or is the tag illegitimate after this query?

Also, as for Avoine's model, the challenge tags must be uncorrupted. This limits the adversary power. Therefore, it is not possible to analyze a system where any corruption is allowed. Finally, in Avoine's model, $\mathcal{A}$ does not have access to the Result oracle. Here, she is forced to have this information: this is not adaptable.

5 Vaudenay [39], 2007

Vaudenay proposes formal definitions for RFID systems and adversaries, and considers that a system $\mathcal{S}$ can be characterized by two notions: security and privacy. In this survey we only present the privacy notion.

Vaudenay defines tags with respect to the adversary possibility to interact with them, as explained in Section 2.2. A tag is said drawn when it is within $\mathcal{A}$'s neighborhood, and has a temporary identity. A tag is said free when it is in the contrary situation, i.e. not drawn. Therefore, all the tags are possibly not accessible to $\mathcal{A}$ during all the attack. Therefore in this model, a tag can be free or drawn at every time. For example, the same tag with identifier $\mathsf{ID}_{\mathcal{T}}$ which is drawn, freed and drawn again has two temporary identities: $\mathcal{A}$ sees two different tags. In this model, when $\mathcal{A}$ wants to interact with a tag, $\mathcal{A}$ can only do it when it is drawn.

During the initialization of the system, DB is empty.

5.1 Oracles

$\mathcal{A}$ has access to all the generic oracles. The only modification done on these ones is that $\mathcal{A}$ can create a fake tag with CreateTag. In that case, no information related to this tag is stored in DB.

Then she additionally can query the following ones.

- DrawTag(distr) $\rightarrow (\mathcal{T}_1, b_1, \ldots, \mathcal{T}_k, b_k)$: following distribution probability distr, it randomly selects k tags between all the existing (not already drawn) ones. For each chosen tag, the oracle assigns to it a new pseudonym, denoted $\mathcal{T}_i$ and changes its status from free to drawn. Finally, the oracle outputs all the generated temporary tags $(\mathcal{T}_1, \cdots, \mathcal{T}_k)$ in any order. If there is not enough free tags (i.e. less than k), then the oracle outputs $\perp$. We further assume that this oracle returns bits $(b_1, \ldots, b_k)$ telling if each of the drawn tags are legitimate or not. All relations $(\mathcal{T}_i, \mathsf{ID}_{\mathcal{T}_i})$ are kept in an *a priori* secret table denoted Tab.
- Free($\mathcal{T}$): moves the tag $\mathcal{T}$ from the status drawn to the status free. This makes $\mathcal{T}$ unavailable from now on (in particular, $\mathcal{A}$ cannot interact with $\mathcal{T}$ anymore).

5.2 Privacy Experiment

From the oracles given above, Vaudenay defines five classes of polynomial-time adversary, characterized by $\mathcal{A}$'s possible access to some specific oracles.

Definition 3 (Adversary Class). *An adversary $\mathcal{A}$ against the RFID system $\mathcal{S}$ is an algorithm which takes a public key $\mathsf{K_P}$ as input and runs the oracles defined above. $\mathcal{A}$ is said to be:*

- STRONG *if $\mathcal{A}$ has no limit on all the oracles;*
- DESTRUCTIVE *if $\mathcal{A}$ cannot use anymore a "corrupted" tag (i.e. the tag has been destroyed);*
- FORWARD *if $\mathcal{A}$ is committed to only use the* Corrupt *oracle after her first call to the* Corrupt *oracle;*
- WEAK *if $\mathcal{A}$ has no access to the* Corrupt *oracle;*
- NARROW *if $\mathcal{A}$ has no access to the* Result *oracle.*

Remark 1. The following relation is clear: WEAK $\subseteq$ FORWARD $\subseteq$ DESTRUCTIVE $\subseteq$ STRONG.

Vaudenay privacy experiment is as follows. P is the adversary class, $P \in \{\emptyset, \mathsf{NARROW}\} \cup \{\mathsf{WEAK}, \mathsf{FORWARD}, \mathsf{DESTRUCTIVE}, \mathsf{STRONG}\}$:

Experiment $Exp^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda]$

1. $\mathcal{C}$ initializes the system and sends 1^λ, and $\mathsf{K_P}$ to $\mathcal{A}$.
2. $\mathcal{A}$ interacts with the whole system, limited by her class P.
3. $\mathcal{A}$ analyzes the system without oracle queries.
4. $\mathcal{A}$ receives the hidden table Tab of the DrawTag oracle.
5. $\mathcal{A}$ returns *true* or *false*.

$Exp^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{A}$ returns *true*.

5.3 Privacy Notions

Definition 4 (Blinder, trivial adversary). *A blinder $\mathcal{B}$ for an adversary $\mathcal{A}$ is a polynomial-time algorithm which sees the same messages as $\mathcal{A}$ and simulates the* Launch, SendReader, SendTag *and* Result *oracles to $\mathcal{A}$. $\mathcal{B}$ does not have access to the reader tapes, so does not know the secret key nor the database.*

A blinded adversary $\mathcal{A}^{\mathcal{B}}$ is itself an adversary who does not use the Launch, SendReader, SendTag *and* Result *oracles.*

An adversary $\mathcal{A}$ is trivial if there exists a blinder $\mathcal{B}$ such that:

$$|\Pr(Exp^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}) - \Pr(Exp^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}^{\mathcal{B}}}[\lambda] \text{ succeeds})| \leq \epsilon(\lambda).$$

Definition 5 (Privacy). *The RFID system $\mathcal{S}$ is said $P-$*private *if all the adversaries which belong to class P are trivial.*

The implications between Vaudenay's privacy notions are the following:

STRONG	$\Rightarrow$	DESTRUCTIVE	$\Rightarrow$	FORWARD	$\Rightarrow$	WEAK
$\Downarrow$		$\Downarrow$		$\Downarrow$		$\Downarrow$
NARROW-STRONG	$\Rightarrow$	NARROW-DESTRUCTIVE	$\Rightarrow$	NARROW-FORWARD	$\Rightarrow$	NARROW-WEAK

The main result of Vaudenay is that STRONG-privacy is impossible, by proving that a DESTRUCTIVE-private protocol is not NARROW-STRONG-private.

Note that the WIDE notion is the contrary to the NARROW notion. If an adversary $\mathcal{A}$ is not said NARROW, then nothing is said, but the term WIDE is implicitly meant.

5.4 Paise and Vaudenay [37], 2008

Paise and Vaudenay extend Vaudenay's model to analyze mutual authentication protocols. Actually, they enrich the definition of the RFID system $\mathcal{S}$ by introducing an output on the tag side: either the tag accepts the reader (if legitimate) and outputs OK, or rejects it (if not) and outputs $\perp$. This defines the concept of *reader authentication*. Nevertheless, their extension of Vaudenay's model does not imply any changes regarding the privacy model.

They also show an important impossibility result: if the corruption of a tag reveals its entire state (and not only its secret $\mathsf{k}_\mathcal{T}$), then no RFID scheme providing reader authentication is NARROW-FORWARD private. To counter this issue, they claim that the temporary memory of a tag should be automatically erased from it as soon as the tag is put back as free. However, this idea is not formalized in the paper.

5.5 Discussion

As remarks, this model is the only one which allows $\mathcal{A}$ to create fake tags, that is tags that are not registered in the reader's database during their creation. On the other hand, other models do not prevent $\mathcal{A}$ to emulate fake tags, as the protocol is assumed public. $\mathcal{A}$ can thus select a key, which will not be in the reader's database with high probability, and then she can interact with a legitimate reader. Consequently, this notion of fake tags is really unclear. This concept is only used for Vaudenay's impossibility result. Unfortunately, it is ambiguous if it can be useful for other reasons. Note also that Vaudenay does not precise if a $\textsc{Free}(\mathcal{T})$ query erases the relation $(\mathcal{T}, \mathsf{ID}_\mathcal{T})$ from the table Tab, even if it seems that this is not the case.

Nevertheless, this model has many good features. First, all the untraceability/privacy notions of Avoine's and JW's model are included in Vaudenay's model. Therefore, this model is the most complete one so far. It is also very powerful since nothing is really specified in the experiment: $\mathcal{A}$ can do whatever she wants, she is not restricted (except by her class). Vaudenay also clarifies the different kinds of corruption that can be performed on a tag. Note that if the adversary's class authorizes corruption, then all tags can be corrupted. This leads to several privacy levels which were not present in the previous models. Additionally, Vaudenay formalizes the access to the "side information bit" introduced by JW with the creation of the oracle $\textsc{Result}$ and with the notion of NARROW adversary. With the latter notion, Vaudenay gives a level of flexibility to his model that is not present in Avoine's and JW's models.

There is no specific challenge tags, but $\mathcal{A}$ can play with any tag of the system during the experiment, which is not the case for Avoine and JW. $\mathcal{A}$ is able to choose the tags for her attack, as for JW (i.e. the challenge tags), which is not the case for Avoine.

Regarding the drawbacks, this model is too vague, and the experiment is not clear at all. Moreover, there is no precise definition about the privacy notion: does privacy represent untraceability, information leakage, both notions, more? Also, since STRONG-privacy is impossible, Vaudenay does not define which privacy level should be targeted by a protocol: is NARROW-STRONG-privacy better than DESTRUCTIVE-privacy? Lastly, it is not explicit how the blinded adversary $\mathcal{A}^\mathcal{B}$ works. Should $\mathcal{A}^\mathcal{B}$ aim the same probability or the same behavior of $\mathcal{A}$? It is obvious that the first solution allows to prove the privacy of some protocols which are actually not private, but this should be correctly formalized.

6 Le, Burmester and de Medeiros [38, 9], 2007

Le, Burmester and de Medeiros introduce a privacy model in [38] (and an extended version in [9]) based on the universal composability framework [13, 14]. This model is denoted as LBM's model in the following. In a nutshell, to prove security properties in the UC model, an ideal functionality which is trivially secure has to be designed. Then an Ident protocol Π is considered secure if no one can distinguish if it is interacting with the real protocol and an adversary or with the ideal functionality and a kind of simulator. This should be true for any adversary even when the distinguisher define the adversary behavior. Before introducing LBM's model, we first recall some generalities about the UC-model.

6.1 General Statements About the UC Model

The main particularity in BLM's model is its UC-fashion that requires the introduction of a new entity $\mathcal{Z}$ called *environment*. Its purpose is to manage the evolution of the system. In other words, this entity is responsible of the activation of all the parties, even the adversary $\mathcal{A}$. $\mathcal{Z}$ is the only entity able to request a party to initiate a new subsession of the studied protocol Π. It is also able to read the output tapes of the parties and $\mathcal{A}$'s ones. On the other hand, $\mathcal{Z}$ is not assumed to read the incoming and outgoing messages of the parties during a protocol subsession. Indeed, in the UC-framework, $\mathcal{A}$ is "responsible" of the communication channels. Consequently, it is assumed that she can eavesdrop, modify and schedule all the communication channels between the parties in an arbitrary way. Furthermore, $\mathcal{A}$ may be able to corrupt parties and thus obtains the full knowledge of their state. Corrupted parties are assumed to be totally controlled by $\mathcal{A}$. A remarkable fact of the UC-model is that $\mathcal{Z}$ and $\mathcal{A}$ can discuss in an arbitrary way. Consequently, if $\mathcal{A}$ wants so, she can forward all the communications to $\mathcal{Z}$. She can even ask $\mathcal{Z}$ to launch new instances of Π. At the end of the experiment, $\mathcal{A}$ may send her final output to $\mathcal{Z}$ which is the last activated entity of the system. Then $\mathcal{Z}$ outputs an arbitrary string which, as proven by Canetti in [13, 14] can be reduced to one bit. This output is denoted $\text{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}}$.

While this new entity is quite unusual compared to previous privacy models in RFID, it permits to model systems where there is an underlying communication structure which may be unknown of the adversary. In the other models where $\mathcal{A}$ is the only one who can activate parties, if this structure is unknown to her, she cannot respect it and may loose information.

In order to prove that a protocol Π ensures some security properties in the UC framework, an ideal functionality, denoted $\mathcal{F}$, has to be defined. It may ensure trivially the desired security objectives and does not depend on any cryptographic mechanisms. In the ideal world, instead of communicating through the adversary (remind that she is responsible of the communication channels), parties invoke this ideal functionality in order to obtain the correct output. $\mathcal{F}$ can thus be seen as a trusted party which perfectly ensures the objectives of the application. In this ideal world, an ideal adversary, denoted $\mathcal{S}$, is defined. As the adversary, $\mathcal{S}$ can arbitrarily discuss with $\mathcal{Z}$ which is defined as in the real world. On the other hand, $\mathcal{S}$ can no longer directly interact with parties. For this, $\mathcal{S}$ has to communicate through the ideal functionality. The main goal of $\mathcal{S}$ is to reproduce as faithfully as possible $\mathcal{A}$'s behavior in the real world. As $\mathcal{A}$ may transfer messages of the protocol Π to $\mathcal{Z}$, it means that $\mathcal{S}$ should try to simulate these messages, as it does not have access to Π, and $\mathcal{F}$ does not produce such messages. In the ideal world, the final output of $\mathcal{Z}$ is denoted $\text{EXEC}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$.

To conclude that a protocol Π is as secure as the ideal functionality, it must be proven that no environment $\mathcal{Z}$ can distinguish if it is interacting with $\mathcal{A}$ and Π or with $\mathcal{S}$ and $\mathcal{F}$. Canetti has formally define this concept in [13] as follows, where PPT denotes probabilistic polynomial time Turing machine.

Definition 6 (UC-Emulation). *A protocol Π UC-emulates a protocol $\mathcal{F}$ if for all PPT adversary $\mathcal{A}$, there exists a PPT ideal adversary $\mathcal{S}$ such that for all PPT environment $\mathcal{Z}$ the distribution* $\text{EXEC}_{\Pi,\mathcal{A},\mathcal{Z}}$ *and* $\text{EXEC}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ *are indistinguishable.*

Informally, UC-emulation proves that the studied protocol is as secure as the ideal functionality. Consequently, the latter must be well-defined such that all the targeted properties are trivially ensured. In the next section, we present one of the ideal functionalities introduced in LBM's model [38, 9].

6.2 Description of LBM's Model

In [38], Le, Burmester and de Medeiros define an RFID system as composed of one trusted server, several readers and several tags. Nevertheless, as they consider that channels between the trusted server and readers are synchronous and secure, they only formally model one entity called *server* which can be seen as the reader $\mathcal{R}$ in the other models. Without loss of generality, we thus called this entity "reader" in the following. LBM considers that only tags can be corrupted by an adversary $\mathcal{A}$. Upon corruption of a tag, $\mathcal{A}$ obtains its keys and any persistent memory value.

The ideal functionnality $\mathcal{F}_{aauth}$ of an ideal anonymous authentication is formally described below. In this description, parties p are either tags or the reader. Two parties are said *feasible partners* if and only if they are respectively the reader and a tag. In the ideal world, communication channels between tags and the reader are assumed to be anonymous, meaning that they only reveal the type (tag or reader) of a party. Furthermore, a sent message is necessarily delivered to the recipient. In the ideal functionality definition, $type(p)$ refers to the type of a party, either tag or reader, $state(p)$ is the list of all subsession records and $active(p)$ is the list of all preceding incomplete subsession.

Functionality $\mathcal{F}_{aauth}$

- **Upon receiving** INITIATE **from** p : if p is corrupted then ignore this message. Else generate a unique subsession identification s, record $\texttt{init}(s, p)$ and send $\texttt{init}(s, type(p), active(p))$ to the adversary.
- **Upon receiving** ACCEPT(s, s') **from** $\mathcal{A}$: if there are two records $\texttt{init}(s, p)$ and $\texttt{init}(s', p')$ where p and p' are feasible partners, then remove them, record $\texttt{partner}(s', p', s, p)$ and write output **ACCEPT**(p') to p. Else if there is a record $\texttt{partner}(s, p, s', p')$ then remove it and write output **ACCEPT**(p') to p.
- **Upon receiving** IMPERSONATE(s, p') **from** $\mathcal{A}$: if there is a record $\texttt{init}(s, p)$ and party p' is corrupted then remove this record and write output **ACCEPT**(p') to p.
- **Upon receiving** CORRUPT(s) **from** $\mathcal{A}$: if there is a record $\texttt{init}(s, p)$ or $\texttt{partner}(s, p, s', p')$ such that p is corruptible then mark p as corrupted and remove $state(p)$.

As described here, the ideal functionality ensures that an adversary only learns the type of a party through the message $\texttt{init}(s, type(p), active(p))$. Upon corruption, the ideal adversary gets the identifier p of the corresponding party and is then able to impersonate this party, using the command IMPERSONATE. She is furthermore able to link all incomplete subsessions of this party, up to the last successfully complete one, based on the knowledge of $active(p)$. Note that as usually done in the UC-framework, a corrupted party is considered as totally controlled by the adversary. Consequently, the ideal functionality will reject all INITIATE command from a corrupted party. As $state(p)$ is removed after a corruption, an adversary is not able to link the related party to its previous authentication. Nevertheless, using the knowledge $active(p)$, an adversary is able to link all incomplete subsessions up to the last successfully one. Thus, the ideal functionality obviously provides forward-security for all previous completed subsessions.

As presented here, the ideal functionality ensures mutual authentication as both parties can output **ACCEPT**. Nevertheless, the authors present in [9] an anonymous client authentication where only the reader authenticates other parties. The authors also introduce in both papers [38, 9] an ideal functionality for anonymous authenticated key exchange which differs from the one presented here by the extension of the output **ACCEPT** which additionally contains a random key k which stands for the exchanged key. As these models are similar, we only focus on the one for anonymous authentication.

6.3 Discussion

One of the greatest contributions of these papers is that they transpose RFID privacy in a UC-fashion which is considered as one of the most powerful tool for security especially when composition among several functionalities is required. Even if the model does not seem intuitive at first sight, it is not so different than the other ones. As an example, reconsidering Vaudenay's model, the blinded adversary $\mathcal{A}^{\mathcal{B}}$ can be considered as the ideal adversary $\mathcal{S}$ which runs $\mathcal{A}$ as a subroutine, and which aims to simulate the behavior of the system to $\mathcal{A}$. Then, a proof in Vaudenay's model has to exhibit that no one can distinguish the real adversary from the blinded/simulated one, as should do an environment $\mathcal{Z}$ for BLM. The interested thing in the UC-framework is that the distinguisher can interact with the system and thus can try to help $\mathcal{A}$ to perform her attack, while Vaudenay's adversary is on its own. This fact has been frequently used in the UC literature to prove that some "considered secure" constructions are indeed not. Another good point of using UC-framework is that all adversaries and environments are taken into consideration, even if they do not precisely model how an adversary should interact with the system.

Unfortunately, the authors do not present some known results about UC-framework which may provide better understanding of its usability. As an example, the notion of *dummy* adversary (see [13, 14]) defines

an adversary which transfer all its knowledge to $\mathcal{Z}$ and acts exactly as $\mathcal{Z}$ tells her. Canetti has proved that a protocol UC-emulates an ideal functionality if and only if it is true for a dummy adversary. Note that this does not represent at all a drawback of their model as this adversary can be used in practice without any modification of the ideal functionality.

On the other hand, their ideal functionality only ensures one level of privacy namely the *forward-security*, which unfortunately is not as strong as some properties introduced in previous models. First, as corruption implies that the corresponding tag can no longer be used normally, i.e., the adversary can only impersonate it, the adversary is equivalent to a DESTRUCTIVE adversary of Vaudenay's model. Consequently, there is no notion of STRONG adversary which can corrupt a tag without destroying it. Secondly, the output tape of all parties is always available for $\mathcal{Z}$. As $\mathcal{Z}$ can communicate arbitrarily with $\mathcal{A}$, the latter may also learn it. Yet, it is impossible to model a kind of NARROW experiment as the distinguisher always knows the result of a protocol execution. Therefore, BLM defines a WIDE-DESTRUCTIVE adversary in Vaudenay's sense.

Regarding the privacy property, the authors have designed their ideal functionality such that all successfully completed subsessions of a party are protected against corruption. In other words, $\mathcal{A}$ cannot learn any information about these previous subsessions and thus their privacy is ensured. But, she may link previous incomplete subsessions of a corrupted party without compromising the security: this is considered as an attack in JW's and Vaudenay's model. The obtained notion is thus more related to the $\mathsf{Universal} - \mathsf{UNT}$ of Avoine where privacy is ensured only for some intervals.

7 Canard, Coisel, Etrog and Girault [11, 12], 2010

In the same vein as Vaudenay's model, Canard, Coisel, Etrog and Girault propose a security model that comprises the properties of (strong) correctness, soundness and untraceability. We only present their untraceability notion. Contrary to Vaudenay, they only talk about untraceability (and not privacy in general) and their main goal is to use the strongest adversary of Vaudenay's model. During the following, this model will be denoted CCEG's model.

As in Vaudenay's model, DB is empty after the setup of the system, and a tag can be either free or drawn.

7.1 Oracles

$\mathcal{A}$ has access to all the generic oracles. She may also use the following ones.

- DrawTag(k): works similarly as the one of Vaudenay. It first randomly and uniformly selects k tags between all existing (not already drawn) ones. For each chosen tag, the oracle gives it a new pseudonym denoted $\mathcal{T}_i$ and changes its status from free to drawn. Finally, since $\mathcal{A}$ cannot create here fake tags, then the oracle only outputs all the generated pseudonyms $\mathcal{T}_1, \cdots, \mathcal{T}_k$ in any order. If there is not enough free tags (i.e. less than k), then the oracle outputs $\perp$. All relations $(\mathcal{T}_i, \mathsf{ID}_{\mathcal{T}_i})$ are kept in a *a priori* secret table denoted Tab.
- Free($\mathcal{T}$): works exactly as the one of Vaudenay.

7.2 Untraceability Experiment

From the oracles given above, the authors defines three classes of polynomial-time adversaries for the untraceability experiment.

Definition 7 (Adversary Class). *An adversary $\mathcal{A}$ against the RFID system $\mathcal{S}$ is an algorithm which takes a public key K_P as input and runs the oracles defined above. $\mathcal{A}$ is said to be:*

- STRONG *if $\mathcal{A}$ has no limit on all the oracles;*
- DESTRUCTIVE *if $\mathcal{A}$ cannot use anymore a "corrupted" tag (i.e. the tag has been destroyed);*
- WEAK *if $\mathcal{A}$ has no access to the* Corrupt *oracle;*

A *link* is a couple of pseudonyms $(\mathcal{T}_i, \mathcal{T}_j)$ associated to the same identifier in Tab. Some links are considered obvious (e.g. both $\mathcal{T}_i$ and $\mathcal{T}_j$ have been corrupted). Therefore, the authors define the notion of *non-obvious link*. As remark, links are chronologically ordered, i.e. $(\mathcal{T}_i, \mathcal{T}_j)$ means that $\mathcal{T}_i$ has been freed before that $\mathcal{T}_j$ has been drawn.

Definition 8 (Non-Obvious Link (NOL)). *$(\mathcal{T}_i, \mathcal{T}_j)$ is a* non-obvious link *if $\mathcal{T}_i$ and $\mathcal{T}_j$ refer to the same $\mathsf{ID}_{\mathcal{T}}$ in Tab and if a "dummy" adversary $\mathcal{A}_d$, who only has access to* CreateTag, Draw, Free, *and* Corrupt*, is not able to output this link with a probability better than $\frac{1}{2}$. Moreover, a non-obvious link is said:*

- standard *if $\mathcal{A}$ has not corrupted $\mathcal{T}_i$ or $\mathcal{T}_j$;*
- past *if $\mathcal{A}$ has corrupted $\mathcal{T}_j$;*
- future *if $\mathcal{A}$ has corrupted $\mathcal{T}_i$.*

Notice that this model uses a "dummy" adversary $\mathcal{A}_d$, instead of a blinded adversary $\mathcal{A}^{\mathcal{B}}$ as in Vaudenay's model. Both adversaries are equivalent but not identical. Indeed, the main difference is that the blinder $\mathcal{B}$ used in Vaudenay's model is an entity clearly separated from $\mathcal{A}^{\mathcal{B}}$. Therefore $\mathcal{B}$ does not know the random choices done by the $\mathcal{A}^{\mathcal{B}}$ during the experiment. On the opposite in CCEG, $\mathcal{A}_d$ is a single entity, and consequently she is aware of her random choices.

A WEAK adversary is only able to output a *standard* NOL as she cannot call the Corrupt oracle. A DESTRUCTIVE adversary is not able to output a *future* NOL as a tag corruption destroys the tag (and thus prevents the tag to be drawn again). However, this adversary can output a *standard* or *past* NOL. Then, a STRONG adversary is able to output every NOL.

CCEG's untraceability experiment is as follows. P is the adversary class, $P \in \{\mathsf{STRONG}, \mathsf{DESTRUCTIVE}, \mathsf{WEAK}\}$

Experiment $Exp^{\mathsf{CCEG\text{-}UNT}}_{\mathcal{S},\mathcal{A}}[\lambda]$

1. $\mathcal{C}$ initializes the system and sends 1^λ, param and $\mathsf{K_P}$ to $\mathcal{A}$.
2. $\mathcal{A}$ interacts with the whole system, limited by her class P.
3. $\mathcal{A}$ returns one link $(\mathcal{T}_i, \mathcal{T}_j)$.

$Exp^{\mathsf{CCEG\text{-}UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds if $(\mathcal{T}_i, \mathcal{T}_j)$ is a NOL.

7.3 Untraceability Notions

Definition 9 (Untraceability). *An RFID system $\mathcal{S}$ is said* untraceable *(resp.* past-untraceable */* future-untraceable*) if, for every* WEAK *(resp.* DESTRUCTIVE */* STRONG*) adversary $\mathcal{A}$ running in polynomial-time, it is possible to define a "dummy" adversary $\mathcal{A}_d$ who only has access to oracles* CreateTag, Draw, Free *and* Corrupt *such that:*

$$|\Pr(Exp^{\mathsf{CCEG\text{-}UNT}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}) - \Pr(Exp^{\mathsf{CCEG\text{-}UNT}}_{\mathcal{S},\mathcal{A}_d}[\lambda] \text{ succeeds})| \leq \epsilon(\lambda).$$

Direct implications are made from these notions:

Future-Untraceability $\Rightarrow$ Past-Untraceability $\Rightarrow$ Untraceability

The main result of this paper is that Future-Untraceability (the strongest privacy property) is achievable.

7.4 Discussion

Note that, here as for all the other models (except Vaudenay), created tags can only be legitimate. CCEG explains that an adversary $\mathcal{A}$ is completely able to simulate a fake tag: the authors thus consider that fake tags seem a meaningless notion. Also, their purpose is to analyze the untraceability of an RFID system: a

fake tag is (by definition) not part of the system. CCEG also considers that a FORWARD adversary (defined as for Vaudenay) has the same goal as a DESTRUCTIVE one: she is able to output a *standard* or *past* NOL, but not a *future* NOL. Therefore, a FORWARD adversary is useless in CCEG's model.

This model has some good features inherited from Vaudenay's model, because CCEG has many common characteristic with Vaudenay. First, there is no specific challenge tags, but $\mathcal{A}$ can play/interact with any tag of the system during the experiment, as Vaudenay's model, which is not the case for Avoine's and JW's models. Furthermore, $\mathcal{A}$ is able to choose the tags for her attack, as in JW's (i.e. the challenge tags) and Vaudenay's models, which is not the case for Avoine's model. Also, all the tags can be corrupted, as for Vaudenay's model, and contrary to Avoine's and JW's model: $\mathcal{A}$ is more powerful.

However, CCEG has two main drawbacks. The first one is that, contrary to Vaudenay's model, no NARROW adversary is used for the untraceability experiment. Their justification is because CCEG's model aims to be as powerful as possible: the NARROW notion weakens the adversary. Unfortunately, the same problem as for JW appears: the voluntary omission of NARROW implies (as explained in Section 12.2) that some protocols with decent security features are not considered untraceable. Secondly, the model has only three untraceability levels: not so much granularity on the model, in comparison to Vaudenay's one.

8 Deng, Li, Yung and Zhao [16], 2010

Deng, Li, Yung and Zhao propose a new framework to define the security and privacy of RFID systems, based on zero-knowledge formulation. Here, we only present the *zero-knowledge* privacy (denoted ZK-privacy). This model will be denoted DLYZ's model.

In a nutshell, DLYZ aims to analyze protocols where entities' secrets may potentially be updated at every protocol execution. Therefore, the model automatically enumerates the internal information of each entity.

At the initialization of the system, the database is in an initial state, called DB^0, and already stores the secrets of all the tags, i.e. a SetupTag has already been performed on every tag. The only differences in the initialization are the following:

- SetupReader additionally generates $\mathcal{R}$'s initial internal state $\mathsf{s}_{\mathcal{R}}^0$;
- SetupTag associates to every tag $\mathcal{T}$ a triplet $(\xi_{\mathcal{T}}, \mathsf{k}_{\mathcal{T}}^0, \mathsf{s}_{\mathcal{T}}^0)$, which is respectively $\mathcal{T}$'s public parameter, initial secret key, and initial internal state.

All these informations are stored in DB^0. Finally, let $\mathsf{param} = (\mathsf{K_P}, \{\xi_{\mathcal{T}}\}_{\forall \mathcal{T}})$ denote the public parameters of the system $\mathcal{S}$. At the end of the system's initialization, all the tags are accessible to an adversary.

8.1 Considered Protocol

Deng, Li, Yung and Zhao consider that an RFID protocol instance $\pi(\mathcal{R}, \mathcal{T})$ is, w.l.o.g., always initialized by $\mathcal{R}$, and π consists of $2\gamma + 1$ rounds for some $\gamma \geq 1$. Each protocol instance π is associated to a unique identifier *sid*. At each instance, a tag may update its internal state and secret key, and $\mathcal{R}$ may update its internal state and database. The update process (of the secret key or the internal state) on a tag always erases the old values.

The authors also assume that a tag may participate to at most s instances in its life with $\mathcal{R}$, thus $\mathcal{R}$ is involved in at most sn instances, where s is polynomial in λ, and n is the total number of tags involved in the system.

We give here an example of protocol instance with identifier *sid*, for the j^{th} instance of $\mathcal{R}$, which corresponds to the v^{th} instance of $\mathcal{T}$, where $0 \leq v < s$. At the beginning, $\mathcal{R}$ takes as input $(\mathsf{param}, \mathsf{K_S}, \mathsf{s}_{\mathcal{R}}^j, \mathsf{DB}^j)$. At the end, $\mathcal{R}$ updates its internal state to $\mathsf{s}_{\mathcal{R}}^{j+1}$ and its database DB^{j+1}. $\mathcal{R}$ also outputs a bit $o_{\mathcal{R}}^{sid}$ that says if $\mathcal{R}$ accepts the instance ($o_{\mathcal{R}}^{sid} = 1$ if so, $o_{\mathcal{R}}^{sid} = 0$ otherwise). At the beginning, $\mathcal{T}$ takes as input $(\mathsf{param}, \mathsf{k}_{\mathcal{T}}^v, \mathsf{s}_{\mathcal{T}}^v)$. At the end, $\mathcal{T}$ updates its internal state $\mathsf{s}_{\mathcal{T}}^{v+1}$ and secret key $\mathsf{k}_{\mathcal{T}}^{v+1}$. $\mathcal{T}$ also outputs a bit $o_{\mathcal{T}}^{sid}$ that says if $\mathcal{T}$ accepts the instance ($o_{\mathcal{T}}^{sid} = 1$ if so, $o_{\mathcal{T}}^{sid} = 0$ otherwise). Notice that the updates can be done also during the protocol instance, and not necessarily at the end. Notice also that, in case of unilateral authentication, $o_{\mathcal{T}}^{sid} = 0$. The transcript of one instance is $(sid, c_1, \alpha_1, \ldots, c_\gamma, \alpha_\gamma, c_{\gamma+1}, o_{\mathcal{R}}^{sid}, o_{\mathcal{T}}^{sid})$, where c_i and α_i are respectively the i^{th} messages of $\mathcal{R}$ and $\mathcal{T}$, $1 \leq i < \gamma$.

8.2 Oracles

In this model, $\mathcal{A}$ has access to the following modified generic oracles.

- $\textsc{Launch}() \rightarrow (\pi, m)$: makes $\mathcal{R}$ launch a new protocol instance π, and generates the 1^{st}-round message m which is also used as the instance identifier sid. If this new instance is the j^{th} instance run by $\mathcal{R}$, then $\mathcal{R}$ stores $c_1 = m$ into its internal state $\mathsf{s}_{\mathcal{R}}^j$.
- $\textsc{SendTag}(m, \mathcal{T}) \rightarrow r$: sends m to $\mathcal{T}$. The output response r of $\mathcal{T}$ is as follows.
  1. If $\mathcal{T}$ currently does not run any instance, then $\mathcal{T}$:
     - initiates a new instance with identifier $sid = m$,
     - treats m as the 1^{st}-round message of the new instance,
     - and returns the 2^{nd}-round message $(sid, r = \alpha_1)$.
  2. If $\mathcal{T}$ is currently running an incomplete instance with identifier sid, and is waiting for the u^{th} message from $\mathcal{R}$ ($u \geq 2$), then $\mathcal{T}$ works as follows:
     - if $2 \leq u \leq \gamma$, $\mathcal{T}$ treats m as the u^{th} message from $\mathcal{R}$, and returns the next round message $(sid, r = \alpha_u)$;
     - if $u = \gamma + 1$ (i.e. the last-round message of the instance), $\mathcal{T}$ returns its output $o_{\mathcal{T}}^{sid}$, and updates its internal state to $\mathsf{s}_{\mathcal{T}}^{v+1}$ (assuming that sid corresponds to the v^{th} instance run by $\mathcal{T}$, where $1 \leq v \leq s$).
- $\textsc{SendReader}(m, sid) \rightarrow r$: sends m to $\mathcal{R}$ for the protocol instance with identifier sid. After receiving m, $\mathcal{R}$ checks from its internal state whether it is running an instance sid, and $\mathcal{R}$'s response r is as follows.
  1. If $\mathcal{R}$ is currently running an incomplete instance with identifier sid, and is waiting for the u^{th} message from a tag ($1 \leq u \leq \gamma$), then $\mathcal{R}$ works as follows:
     - if $u \leq \gamma$, $\mathcal{R}$ treats m as the u^{th} message from the tag, and returns the next round message $r = c_{u+1}$;
     - if $u = \gamma$, $\mathcal{R}$ returns the last-round message $r = c_{\gamma+1}$ and its output $o_{\mathcal{R}}^{sid}$, and updates its internal state to $\mathsf{s}_{\mathcal{R}}^{j+1}$ and the database DB^{j+1} (assuming that sid corresponds to the j^{th} instance run by $\mathcal{R}$).
  2. In all other cases, $\mathcal{R}$ returns $\perp$ (for invalid queries).
- $\textsc{Corrupt}(\mathcal{T}) \rightarrow (\mathsf{k}_{\mathcal{T}}^v, \mathsf{s}_{\mathcal{T}}^v)$: returns the secret key $\mathsf{k}_{\mathcal{T}}^v$ and the internal state $\mathsf{s}_{\mathcal{T}}^v$ currently held by $\mathcal{T}$. Once $\mathcal{T}$ is corrupted, all its actions are controlled and performed by $\mathcal{A}$.

Let $\mathcal{O}_1$, $\mathcal{O}_2$, $\mathcal{O}_3$ and $\mathcal{O}_4$ denote respectively the above oracles, and $\mathcal{O}$ denote the set of these four oracles.

Let denote $\mathcal{A}^{\mathcal{O}}(\mathcal{R}, T, \mathsf{param})$ a PPT algorithm $\mathcal{A}$ that takes on input the system public parameters param, the reader $\mathcal{R}$ and the tags set T of the already initialized system. Then the algorithm interacts with $\mathcal{R}$ and the tags of T via the four oracles. As remark, this model considers that the output bits of $\mathcal{R}$ and tags are publicly known.

Let also denote $\mathcal{A}^{\mathcal{O}}(\mathcal{R}, \widehat{T}, \mathcal{I}(\mathcal{T}_g), aux)$ a PPT algorithm $\mathcal{A}$ that takes on input $aux \in \{0,1\}^*$ (in general, aux includes param or some historical state information of $\mathcal{A}$). Then the algorithm interacts with $\mathcal{R}$ and the tags of $\widehat{T}$ via the four oracles. $\mathcal{A}$ is said to have a *blinded access* to a *challenge* tag $\mathcal{T}_g \notin \widehat{T}$ if $\mathcal{A}$ interacts with $\mathcal{T}_g$ via a special interface $\mathcal{I}$ (i.e. a PPT algorithm which runs $\mathcal{T}_g$ internally, and interacts with $\mathcal{A}$ externally). To send a message m to $\mathcal{T}_g$, $\mathcal{A}$ sends to $\mathcal{I}$ a $\textsc{SendTag}(m, challenge)$; then $\mathcal{I}$ invokes $\mathcal{T}_g$ with $\textsc{SendTag}(m, \mathcal{T}_g)$, and answers $\mathcal{T}_g$'s output to $\mathcal{A}$. $\mathcal{A}$ does not know which tag is interacting with her. $\mathcal{A}$ interacts with $\mathcal{T}_g$ via $\textsc{SendTag}$ queries only.

Definition 10 (Clean Tag). *A tag $\mathcal{T}$ is said* clean *if it is not corrupted (i.e. no call to* $\textsc{Corrupt}$ *on $\mathcal{T}$) and is not currently running an incomplete instance with $\mathcal{R}$ (i.e. $\mathcal{T}$'s last instance is either finished or aborted).*

The main goal of this definition is to make clear that the adversary needs to have some uncorrupted or running tags to proceed the ZK-privacy experiment (see next section). This notion of tags not running a protocol instance is very similar to the $\textsc{TagInit}$ oracle of JW.

8.3 Privacy Experiments

In the experiments, a PPT CMIM (concurrent man-in-the-middle) adversary $\mathcal{A}$ (resp. PPT simulator Sim) is composed of a pair of algorithms $(\mathcal{A}_1, \mathcal{A}_2)$ (resp. (Sim_1, Sim_2)), and runs in two stages. Notice that, if $\delta = 0$, then no challenge tag is selected, and $\mathcal{A}$ is reduced to $\mathcal{A}_1$ in the experiment.

The first experiment is the one performed by the real adversary $\mathcal{A}$. After the system initialization, $\mathcal{A}_1$ plays with all the entities and returns a set of clean tags C. Then from this set C, a challenge tag $\mathcal{T}_g$ is chosen at random. Then $\mathcal{A}_2$ plays with all the entities, including the challenge tag via the interface $\mathcal{I}$, except the set of clean tags. At the end, $\mathcal{A}$ outputs a view of the system.

Experiment $Exp^{\mathsf{ZK\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda, n]$ (real world)

1. $\mathcal{C}$ initializes the system and sends 1^λ, param to $\mathcal{A}$.
2. $\{C, st\} \leftarrow \mathcal{A}_1^{\mathcal{O}}(\mathcal{R}, T, \mathsf{param})$, where $C = \{\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, \ldots, \mathcal{T}_{i_\delta}\} \subseteq T$ is a set of *clean* tags $(0 \leq \delta \leq n)$, and st is a state information.
3. $g \in_R \{1, \ldots, \delta\}$, set $\mathcal{T}_g = \mathcal{T}_{i_g}$ and $\widehat{T} = T - C$.
4. $view_{\mathcal{A}} \leftarrow \mathcal{A}_2^{\mathcal{O}}(\mathcal{R}, \widehat{T}, \mathcal{I}(\mathcal{T}_g), st)$.
5. Output $(g, view_{\mathcal{A}}(\lambda, n))$.

Then, the second experiment is the one performed by the simulator Sim. As in the previous experiment, Sim_1 plays with all the entities and returns a set of clean tags C. Then from this set C, a challenge tag $\mathcal{T}_g$ is chosen at random, but Sim is not informed about its identity and cannot play anymore with this tag. Then Sim_2 plays with all the entities, except the set of clean tags. At the end, Sim outputs a simulated view of the system.

Experiment $Exp^{\mathsf{ZK\text{-}priv}}_{\mathcal{S},Sim}[\lambda, n]$ (simulated world)

1. $\mathcal{C}$ initializes the system and sends 1^λ, param to $\mathcal{A}$.
2. $\{C, st\} \leftarrow Sim_1^{\mathcal{O}}(\mathcal{R}, T, \mathsf{param})$, where $C = \{\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, \ldots, \mathcal{T}_{i_\delta}\} \subseteq T$ is a set of *clean* tags $(0 \leq \delta \leq n)$, and st is a state information.
3. $g \in_R \{1, \ldots, \delta\}$ unknown to Sim, and set $\widehat{T} = T - C$.
4. $sview \leftarrow Sim_2^{\mathcal{O}}(\mathcal{R}, \widehat{T}, st)$, where $sview$ particularly includes all oracle answers to queries made by Sim.
5. Output $(g, sview(\lambda, n))$.

8.4 Privacy Notions

From the previous experiments, the ZK-privacy of a system $\mathcal{S}$ is proved when no one is able to distinguish if he is interacting with the real world or with the simulated one.

Definition 11 (ZK-Privacy). *An RFID system $\mathcal{S}$ satisfies computational (resp. statistical)* ZK-privacy *if, for any PPT CMIM adversary $\mathcal{A}$, there exists a polynomial-time simulator Sim such that, for all sufficiently large λ and any n which is polynomial in λ, the following ensembles are computationally (resp. statistically) indistinguishable:*

- $\{g, view_{\mathcal{A}}(\lambda, n)\}_{\lambda \in \mathbb{N}, n \in poly(\lambda)}$
- $\{g, sview(\lambda, n)\}_{\lambda \in \mathbb{N}, n \in poly(\lambda)}$

That is, for any polynomial-time (resp. any computationally power unlimited) algorithm D, it holds that:

$$|\Pr[D(\lambda, n, g, view_{\mathcal{A}}(\lambda, n)) = 1] - \Pr[D(\lambda, n, g, sview(\lambda, n)) = 1]| = \epsilon(\lambda).$$

Definition 12 (Backward/Forward-ZK-Privacy). *Let denote* $(k^f_{\mathcal{T}_g}, s^f_{\mathcal{T}_g})$ *(resp.* $(k^0_{\mathcal{T}_g}, s^0_{\mathcal{T}_g})$*) the final (resp. initial) secret key and internal state of the challenge tag* $\mathcal{T}_g$ *at the end (resp. beginning) of* $Exp^{\mathsf{ZK\text{-}priv}}_{\mathcal{S},\mathcal{A}}$*. An RFID system* $\mathcal{S}$ *is* forward *(resp.* backward*)* ZK-private *if, for any PPT CMIM adversary* $\mathcal{A}$*, there exists a polynomial-time simulator Sim such that, for all sufficiently large* λ *and any n which is polynomial in* λ*, the following distributions are indistinguishable:*

- $\{k^f_{\mathcal{T}_g}, s^f_{\mathcal{T}_g}(resp., k^0_{\mathcal{T}_g}, s^0_{\mathcal{T}_g}), g, view_{\mathcal{A}}(\lambda, n)\}$
- $\{k^f_{\mathcal{T}_g}, s^f_{\mathcal{T}_g}(resp., k^0_{\mathcal{T}_g}, s^0_{\mathcal{T}_g}), g, sview(\lambda, n)\}$

It is required that $\mathcal{T}_g$ *should remain clean at the end of* $Exp^{\mathsf{ZK\text{-}priv}}_{\mathcal{S},\mathcal{A}}$*. Note that* $\mathcal{A}$ *is allowed to corrupt it after the end of* $Exp^{\mathsf{ZK\text{-}priv}}_{\mathcal{S},\mathcal{A}}$*.*

8.5 Models based on the notion of *unpredictability*

Several models are based on a very closed notion of ZK-privacy, called *unpredictability*. All these models, as DLYZ, rely on the unpredictability of the output returned by a tag or a reader in a protocol execution.

First, Ha, Moon, Zhou, and Ha publish the first model in [24]. Then, Ma, Li, Deng, and Li intent to repair the former model in [31]. Finally, Lai, Deng, and Li propose an extension on the repaired model in [30]. Note that all these models have been proposed before DLYZ's model. In this survey, we decide to only present DLYZ, since it is the most achieved model in this category.

8.6 Discussion

This model is the only one using the zero-knowledge notion: this is a new way of thinking in privacy for RFID. As remark, the tags created during the experiment can only be legitimate, as for all the other models, except Vaudenay.

Unfortunately, this model has many disadvantages. First, this model can only analyze $(2\gamma + 1)$-pass protocols where $\gamma \geq 1$. This is quite restrictive. As for JW's and CCEG's models, $\mathcal{A}$ is forced to know the result information regarding the protocol success: this is not adaptable.

Also, $\mathcal{A}$ cannot play/interact with all the tags during the experiment. She can play with all the tags during the "learning" phase, but then she cannot play with the whole set of clean tags in the "challenge" phase, but just with the challenge tag $\mathcal{T}_g$. This is more restrictive than Vaudenay and JW, but less than Avoine. $\mathcal{A}$ chooses the set of clean tags, but she does not choose the challenge tag $\mathcal{T}_g$: it is picked at random, which also limits the adversary's power. Besides $\mathcal{T}_g$ cannot be corrupted in the experiment: its secrets are revealed at the end in case of backward-ZK and forward-ZK-privacy. One justification of the authors is that it is sufficient to give the secrets to $\mathcal{A}$ at the end. Another authors' reason is that backward-ZK or forward-ZK-privacy cannot be achieved if $\mathcal{A}$ corrupts the challenge tag before the end of the experiment. This significant restriction implies that the attack is not adaptable regarding corruption.

Then, the authors claim that each tag $\mathcal{T}$ has its output bit $o^{sid}_{\mathcal{T}} = 0$ if the protocol is not mutual. This seems too restrictive: even if $\mathcal{T}$ may not authenticate the reader, it can nevertheless have an output unknown by $\mathcal{A}$. For instance, its output can be "I arrived correctly at the end of the protocol on my side".

The authors declare that ZK-privacy is stronger than JW's privacy. Indeed, their proof about that fact is very questionable and doubtful. They argue that JW's privacy does not imply ZK-privacy with two examples of protocols that satisfy this claim. The first example is a system composed of only one tag: clearly this protocol cannot be analyzed in JW since the model requires two tags in the experiment. Thus this example is not a proof that JW's privacy does not imply ZK-privacy. Anyway, it seems reasonable to say that such a system does not have privacy at all. Furthermore in this special case, the authors say that ZK-privacy is reduced to the basic zero-knowledge definition which, according to them, provides privacy. This is not true, since zero-knowledge does not necessarily imply privacy. The second example is a system with IND-CPA security where all tags share a pair of public/private keys. The reader sends to a tag the cipher of a nonce with the public key. The tag deciphers this message with the secret key, and sends back the initial nonce

concatenated with another nonce. This example is also controversial because JW claims to only analyze the privacy of protocols based on symmetric-key cryptography. Therefore this example is not fair regarding JW's model. Then, as explained in Section 12.5, such a system is difficult to be proven private. Indeed, the argument given by DLYZ (saying that this protocol cannot be ZK-private) can be simply applied to JW: with IND-CPA security, it does not seem possible for the simulator in $Exp^{\mathsf{JW\text{-}priv}}_{\mathcal{S},\mathcal{A}}$ (simulating $\mathcal{T}^*_b$) to output a correct answer for a SendTag query. Furthermore, [34] shows that ZK-privacy is equivalent to JW's privacy.

Finally, the authors claim that the adversary in ZK-privacy corresponds to Vaudenay's STRONG-privacy with a slightly difference: their adversary cannot corrupt any challenge tag before the end of the experiment. This difference is actually the basis of the STRONG adversary definition in Vaudenay's model. Therefore this unproved claim is not convincing.

9 Hermans, Pashalidis, Vercauteren and Preneel [25], 2011

Following the path opened by Vaudenay with his privacy model, Hermans, Pashalidis, Vercauteren and Preneel present a new model based on indistinguishability between two "worlds": it is most commonly called the "left or right" paradigm. This model will now be denoted HPVP's model.

As for Vaudenay and CCEG, DB is empty after the initialization of the system, and a tag can be either free or drawn.

9.1 Oracles

$\mathcal{A}$ has access to the generic oracles CreateTag (here it additionally returns a reference $\mathcal{T}$ to the new created tag), SendReader, Result. Then, $\mathcal{A}$ has also access to these other following oracles.

- DrawTag$(\mathcal{T}_i, \mathcal{T}_j) \rightarrow (\mathcal{T}_{\mathsf{drawn}})$: generates a drawn tag $\mathcal{T}_{\mathsf{drawn}}$ and stores $(\mathcal{T}_{\mathsf{drawn}}, \mathcal{T}_i, \mathcal{T}_j)$ in an *a priori* secret table Tab. Depending on the bit b chosen at the start of the privacy experiment (see next section), $\mathcal{T}_{\mathsf{drawn}}$ will either reference $\mathcal{T}_i$ or $\mathcal{T}_j$. If one of the two tags $(\mathcal{T}_i, \mathcal{T}_j)$ is already referenced in Tab, then it outputs $\perp$.
- Free$_b(\mathcal{T}_{\mathsf{drawn}})$: recovers the tuple $(\mathcal{T}_{\mathsf{drawn}}, \mathcal{T}_i, \mathcal{T}_j)$ in Tab. If $b = 0$ then it resets $\mathcal{T}_i$, otherwise it resets $\mathcal{T}_j$. Then it removes the tuple from Tab. When a tag is reset, its volatile memory is erased, not its non-volatile memory (which contains its secret $\mathsf{k}_{\mathcal{T}}$).

Finally $\mathcal{A}$ has access to the following modified generic oracles.

- Launch$() \rightarrow (\pi, m)$: makes $\mathcal{R}$ launch a new Ident protocol instance π, together with $\mathcal{R}$'s first message m.
- SendTag$(m, \mathcal{T}) \rightarrow r$: retrieves the tuple $(\mathcal{T}, \mathcal{T}_i, \mathcal{T}_j)$ in Tab. It sends a message m to the corresponding tag ($\mathcal{T}_i$ if $b = 0$, $\mathcal{T}_j$ otherwise). It outputs the response r of the tag. If $\mathcal{T}$ is not found in Tab, it returns $\perp$.
- Corrupt$(\mathcal{T}) \rightarrow \mathsf{k}_{\mathcal{T}}$: returns the current secret $\mathsf{k}_{\mathcal{T}}$ of $\mathcal{T}$. If $\mathcal{T}$ is drawn, it returns $\perp$.

All these oracles are very similar to the ones of Vaudenay, but with important differences. First, DrawTag is only applied on two tags chosen by the adversary when she does this query. Then, Free specifies clearly that it erases the volatile memory of the chosen tag. Lastly, Corrupt is not authorized on a drawn tag.

9.2 Privacy Experiment

The authors keep the same adversary classes as the ones given by Vaudenay: STRONG, DESTRUCTIVE, FORWARD, WEAK, and NARROW.

Their privacy experiment is as follows, where P represents the adversary class: $P \in \{\emptyset, \mathsf{NARROW}\} \cup \{\mathsf{WEAK}, \mathsf{FORWARD}, \mathsf{DESTRUCTIVE}, \mathsf{STRONG}\}$.

Experiment $Exp^{\mathsf{HPVP\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda, b]$

1. $\mathcal{C}$ initializes the system, chooses a bit b at random, and sends 1^λ and param to $\mathcal{A}$.
2. $\mathcal{A}$ interacts with the whole system, limited by her class P.
3. $\mathcal{A}$ outputs a guess bit b'.

$Exp^{\mathsf{HPVP\text{-}priv}}_{\mathcal{S},\mathcal{A}}$ succeeds if $b = b'$.

9.3 Privacy Notions

Definition 13 (Privacy). *The RFID system $\mathcal{S}$ is said to unconditionally (resp. computationally) provide $P-$*privacy *if and only if, for all the adversaries (resp. polynomial time adversaries) which belong to class P, it holds that:*

$$|\Pr(Exp^{\mathsf{HPVP\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda,0] \text{ succeeds}) + \Pr(Exp^{\mathsf{HPVP\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda,1] \text{ succeeds}) - 1| = 0 \text{ (resp. } \leq \epsilon(\lambda)).$$

9.4 Discussion

In this model, there are no challenge tags: $\mathcal{A}$ must differentiate the two worlds. Also, as for all the other models (except Vaudenay), the tags created can only be legitimate. Another remark is that the FREE oracle reset the drawn tag, by erasing its volatile memory: this comes from one important comment done in Paise-Vaudenay's model (see Section 5.4).

Since HPVP is very similar to Vaudenay (i.e., they have the same unclear privacy notion), this model also inherits from its good features. First, $\mathcal{A}$ is able to choose the tags for her attack, as in JW, Vaudenay, and CCEG. Then, all the tags can be corrupted, as for Vaudenay's and CCEG's models. The best advantage in this model is that it is possible to reach the highest privacy level, i.e., STRONG-privacy (see Section 12.5 for more details).

On the other side, this model has several drawbacks. First, HPVP clearly specifies that its aim is not to analyze systems where the tag may start a protocol instance: this seems a useless restriction. Then, at the same moment in the experiment, $\mathcal{A}$ cannot play with all the tags of the system. Also, the CORRUPT oracle can only be call on a free tag $\mathcal{T}$. But the intrinsic definition of a free tag is that it is not accessible to $\mathcal{A}$ (i.e., not in her neighborhood). Thus, how can $\mathcal{A}$ make a CORRUPT on a tag that she cannot manipulate (i.e., not drawn)? Finally, during all the paper, the authors claim that the already existing models do not take care about some privacy leakage information such as the cardinality of the tags' set. But they never prove nor explain how their model can handle this issue, nor why this is a privacy issue.

10 Another Different Model: van Deursen, Mauw and Radomirovic [18], 2008

The model of van Deursen, Mauw and Radomirovic defines *untraceability* on the standard Dolev-Yao intruder model [20]. The untraceability notion is inspired by the anonymity theory given in [22, 32]. This model will now be called DMR's model.

10.1 Definition of the System

Here, we give the basic definitions given in DMR, in order to understand their notion of untraceability.

First, the system is composed by a number of *agents* (e.g., Alice or Bob) that execute a *security protocol*, the latter being described by a set of *traces*.

A security protocol represents the behavior of a set of *roles* (i.e., initiator, responder, server), each one specifying a set of actions. These actions depict the role specifications with a sequence of of *events* (e.g., sending or reception of a message). A *role term* is a message contained in an event, and it is built from *basic role terms* (e.g., nonces, role names or keys). A *complex term* is built with functions (e.g., tupling, encryption, hashing, XOR).

Each trace is composed of interleaved runs and run prefixes, denoted *subtraces*. A *run* of a role R is an execution of a protocol from R's point of view, denoted $R\#\theta$, where θ is a (possibly unique) run identifier. Thus, a run is an instantiation of a role. A *run event* is an instantiation of a role event, that is an instantiation of an event's role terms. A *run term* denotes an instantiated role term. A *run prefix* is an unfinished run.

An adversary $\mathcal{A}$ of the system is in the Dolev-Yao model, and is characterized by her *knowledge*. This knowledge is composed of a set of run terms known at the beginning, and the set of run terms that she will observe during her attack. The adversary is allow to manipulate the information of her knowledge to

understand terms or build new ones. However, perfect cryptography is assumed. The inference of a term m from a term set M is denoted $M \vdash m$.

Corrupted agents are modeled: $\mathcal{A}$ is given all the secrets of a corrupted agent in her initial knowledge. When an agent is corrupted, it is said "destroyed", i.e., he cannot be used during $\mathcal{A}$'s attack. Yet, the security evaluation of a system is done on non-corrupted agents, that is $\mathcal{A}$ cannot have access to the secret of an agent after the start of her attack.

10.2 Untraceability Notion

First, the authors define several notions of *linkability*, *reinterpretation*, and *indistinguishability*, before giving the *untraceability* one.

Definition 14 (Linkability of subtraces). *Two subtraces t_i^R and t_j^R are linked, denoted $L(t_i^R, t_j^R)$, if they are instantiated by the same agent:*

$$L(t_i^R, t_j^R) \equiv (agent(t_i^R) = agent(t_j^R))$$

The notion of *reinterpretation* has been introduced in [22] in order to show that subterms of a message can be replaced by other subterms if the adversary $\mathcal{A}$ is not able to understand these subterms. Note that, when $\mathcal{A}$ is able to understand a subterm, it remains unchanged.

Definition 15 (Reinterpretation). *A map π from run terms to run terms is called a reinterpretation under knowledge set M if it and its inverse π^{-1} satisfy the following conditions:*

- $\pi(m) = m$ *if m is a basic run term,*
- $\pi(m) = (\pi(m_1), \ldots, \pi(m_n))$ *if $m = (m_1, \ldots, m_n)$ is n-tuple,*
- $\pi(\{m\}_k) = \{\pi(m)\}_k$ *if $M \vdash k^{-1}$ or $(M \vdash m \wedge M \vdash k)$,*
- $\pi(f(m)) = f(\pi(m))$ *if $M \vdash m$ or f is not a hash function.*

Reinterpretations are used to define *indistinguishability* of traces.

Definition 16 (Indistinguishability of traces). *Let M be the adversary's knowledge at the end of trace t. The trace t is* indistinguishable *from a trace t', denoted $t \sim t'$, if there is a reinterpretation π under M, such that $\pi(t_i^R) = {t'}_i^R$ for all roles R and subtraces t_i^R.*

From all the above notions, the authors define the untraceability notion of a role as follows.

Definition 17 (Untraceability). *A protocol P is said* untraceable *with respect to role R if:*

$$\big(\forall t \in Traces(P)\big)\big(\forall i \neq j\big)\Big(L(t_i^R, t_j^R) \Rightarrow \big(\exists t' \in Traces(P)\big)\big((t \sim t') \wedge \neg L({t'}_i^R, {t'}_j^R)\big)\Big).$$

In this survey, if no role is specified, we consider that "untraceability" means "untraceability for role $\mathcal{T}$".

10.3 Discussion

The first observation is that this model is not based on oracles. Here as for all the other models (except Vaudenay), the agents created for the adversary's attack can only be legitimate, but there is no specific challenge agents for the attack.

One interesting feature is that the *insider adversary* (as explained in details in [17]) is modeled here when $\mathcal{A}$ obtains the secrets of some corrupted agents before the beginning of her attack. This is a little bit stronger than Vaudenay's NARROW-WEAK-privacy. This adversary is also very closed to the one of JW, but JW seems stronger as $\mathcal{A}$ can corrupt any non challenge tag at any moment of the learning phase.

Despite this attractive attribute, this model does not seem to be intuitive at all: it is based on formal logic. As for BLM, no experiment is given. Therefore it is difficult to know how to use this model to prove the untraceability of a given system. Consequently, it is not clear how $\mathcal{A}$ acts to perform her attack. Furthermore,

there is no way to model the equivalence of a CORRUPT oracle during the attack: $\mathcal{A}$ can only have secrets of some corrupted agents before the beginning of her attack. Additionally, $\mathcal{A}$ does not know when a protocol between two agents succeeded: there is no equivalent of the RESULT oracle in DMR.

Lastly, the model allows to analyze untraceability of a given role, for instance tag's role. When $\mathcal{A}$ want to do so, she can only corrupt the agents of that specific role. She cannot, for instance, corrupt a reader to be able to check tag's untraceability. This is a restriction that does not exist in the other models. Reader's corruption is actually not include in the first version of the other models. But it has been shown in [5, 23] that this notion can be added very easily to them (e.g., via a dedicated oracle). This is not the case in DMR.

11 General Comparison

Table 1 illustrate the comparison between the privacy models presented in this article. From all the information given here, some results are worth to be highlighted.

Vaudenay is the only model where it is possible to create fake tags, that is tags which do not belong to the system. The author is not precise regarding the relevance of this notion. He only uses fake tags to prove that STRONG-privacy is impossible (see [39] for more details on this proof).

Vaudenay, LBM and CCEG are the only models where an adversary $\mathcal{A}$ can play with all the tags at the same time. This is a substantial asset over the other models: $\mathcal{A}$ has more power than in the other models. Therefore this property strengthens the privacy notions of these models.

The fact that $\mathcal{A}$ chooses the challenge tags and that she can corrupt them is also a force in a model. Thus, even if Vaudenay, LBM, CCEG and HPVP do not have defined challenge tags, their adversary can play with a tag with the same restrictions: she can choose the tags to interact with during the experiment, and corrupt them as well. Therefore, Vaudenay, LBM, CCEG and HPVP have an advantage on the other models.

Several models are restricted to be NARROW or WIDE: this does not provide modularity of the adversary strength. Consequently, this can prevent some models to analyze fairly a protocol (see the analysis given in Section 12.2). Here again, Vaudenay and HPVP are the best choice.

Some models are designed "by default" to analyze specific identification/authentication protocols (e.g., 3-pass protocols for Avoine, and SK-based protocols for JW). On the contrary, Vaudenay's, LBM, CCEG's and DMR's models can (a priori) analyze any identification/authentication protocol. Some of the restrictive models can nevertheless be adapted to analyze most existing identification/authentication protocols.

To conclude on this comparison, none of the presented privacy models complies with all the properties given in Table 1. The models are not easily extensible and modifiable to analyze the realistic RFID environments. Even if we put together all the positive features of each model, it does not seem possible to have the ultimate and best privacy model. Clearly, it seems necessary nowadays to have a model that compels with all these expected properties. First, this model would propose a wide-ranging adversary. This would imply more granularity in the possible privacy levels that a protocol would reach. Finally, this would allow to get a more precise measure of privacy for a given protocol.

Note that LBM's model is said WIDE even if the adversary does not necessarily have access to the reader's output. Nevertheless, the environment always has access to the output tape of the reader and may send these results to the adversary. Thus, it seems more relevant to consider that this model defines a WIDE adversary.

12 Privacy Analysis of Different Existing Protocols

To investigate more deeply the differences between the presented models, we study the privacy level of five different protocols in all these models. These protocols differ according to their building blocks and their underlying key infrastructure. Thus they may reach different security properties. Consequently, this fact should be reflected in the different privacy models. As detailed in this section, we will show that some models give the same level of privacy to some protocols while other models clearly differentiate them, e.g., by taking into account an attack which cannot be modeled in other models.

Table 1. Comparison of the presented privacy models. "✓" (resp. "✗") means the property is (resp. is not) ensured by the model. "?" means that no information is given on that property.

Property	Avoine	JW	Vaudenay	LBM	CCEG	DLYZ	HPVP	DMR
DB complete at start	✓	✓	✗	? a priori ✓	✗	✓	✗	N/A
$\mathcal{A}$ creates fake tags	✗	✗	✓	✗	✗	✗	✗	✗
$\mathcal{A}$ interacts with all the tags	✗ only 2	✗ not $\mathcal{T}^*_{b\oplus 1}$	✓	✓	✓	✗ not all clean tags	✗ at most half of the tags	✗ not corrupted agents
$\mathcal{A}$ Corrupt any tag	✗	✗	✓	✓	✓	✗	✓	✗
$\mathcal{A}$ chooses challenge tags	✗	✓	N/A	N/A	N/A	✗ (at random)	N/A	N/A
$\mathcal{A}$ Corrupt challenge tags	it depends	✗	N/A	N/A	N/A	✗	N/A	N/A
NARROW/WIDE	NARROW	WIDE	both	WIDE	WIDE	WIDE	both	NARROW
Simulator/blinder defined	✗	✗	✓	✓	✓	✓	✗	✗
Predefined analyzable protocols	3-pass	SK prot	? a priori all	? a priori all	? a priori all	$(2\gamma+1)$-pass	not protocols with tag init	? a priori all

In the following, a tag $\mathcal{T}$ has a unique identifier $\mathsf{ID}_\mathcal{T}$, and wants to be authenticated successfully by a legitimate reader $\mathcal{R}$. F and G refer to pseudo random functions, while f and g refer to one-way functions. (Enc/Dec) refers to an encryption scheme. Finally, λ denotes the security parameter of the system.

12.1 SK-Based Challenge/Response Authentication Protocol

This protocol is the ISO/IEC 9798-2 Mechanism 2 [27] based on a PRF with an additional nonce chosen by the tag. A tag $\mathcal{T}$ has a unique secret key $\mathsf{k}_\mathcal{T}$ known by $\mathcal{R}$, used for the authentication.

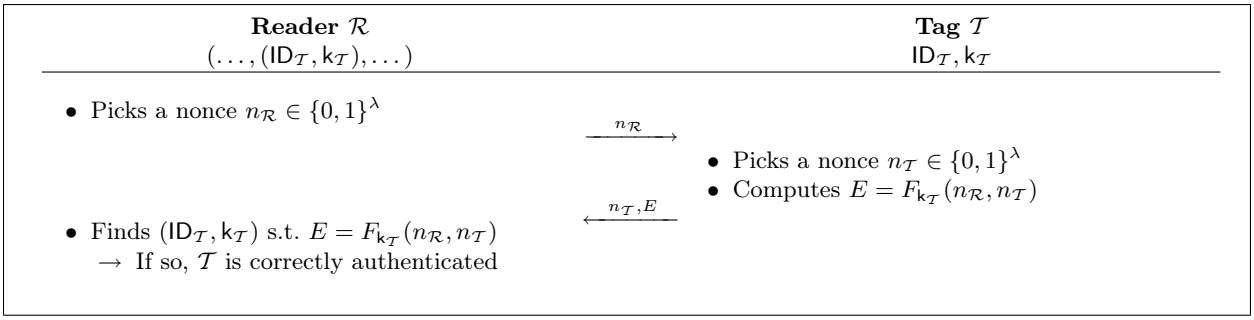

Fig. 2. SK-based authentication protocol.

In this protocol, it is obvious that one single corruption of a tag $\mathcal{T}$ allows to trace it at any time. This is feasible as the secret key of a tag is a fixed value and the nonces used in the pseudo-random function are sent in the clear. Thus, an adversary is able to recompute the value E for the corrupted tag and compare it with the previously sent one. If these values are equal, then the adversary is convinced that the corrupted tag performed this authentication[1]. Nevertheless, the corruption of another tag $\mathcal{T}'$ does not help to trace the tag $\mathcal{T}$, since all secret keys are independent. Consequently, this protocol can only reach privacy properties when the adversary is not allowed to corrupt the challenge tags.

Therefore this protocol is $\mathsf{Existential} - \mathsf{UNT}-$RTE in Avoine's model (proved for this kind of protocols in [1]), and (r,s,t)-private in JW's model (proved in [28]).

It is WEAK-private for Vaudenay (proved in [39]) and for HPVP. It is untraceable for CCEG. The proofs for HPVP and CCEG are very similar to the one of Vaudenay.

This protocol is ZK-private in DLYZ's model (the proof of a similar protocol in [16] can be trivially adapted) and untraceable for DMR (proved in [18]). Finally, this protocol cannot UC-emulate the ideal

[1] Note that this equality can be due to a collision, but this happens with a negligible probability.

functionality in LBM's model as the attack presented here permit to an adversary to link several subsessions while this is not possible for the simulator (as $state(p)$ is removed after a corruption).

12.2 OSK-Based Authentication Protocol

The original OSK protocol [36] is an identification protocol, where there is no proof of the tag identity. At the setup, $\mathcal{T}$ is initialized with a unique key $\mathsf{k}_\mathcal{T}$ shared with $\mathcal{R}$. All the tags' keys are independent. $\mathcal{T}$ just sends the result of a pseudorandom function done on its key. The main feature of OSK is that $\mathcal{T}$ and $\mathcal{R}$ update the shared key after each complete protocol execution.

The OSK protocol has been introduced to ensure the *forward security* property, i.e., data sent by a given tag $\mathcal{T}$ today will still be secure even if $\mathcal{T}$'s secret is disclosed by tampering this tag in the future, contrary to the SK-protocol introduced in the previous subsection. The protocol presented here is slightly different from OSK as $\mathcal{R}$ additionally sends a nonce to $\mathcal{T}$ in order to prevent replay attacks, as described in [3]. The resulting protocol ensures tag authentication rather than tag identification.

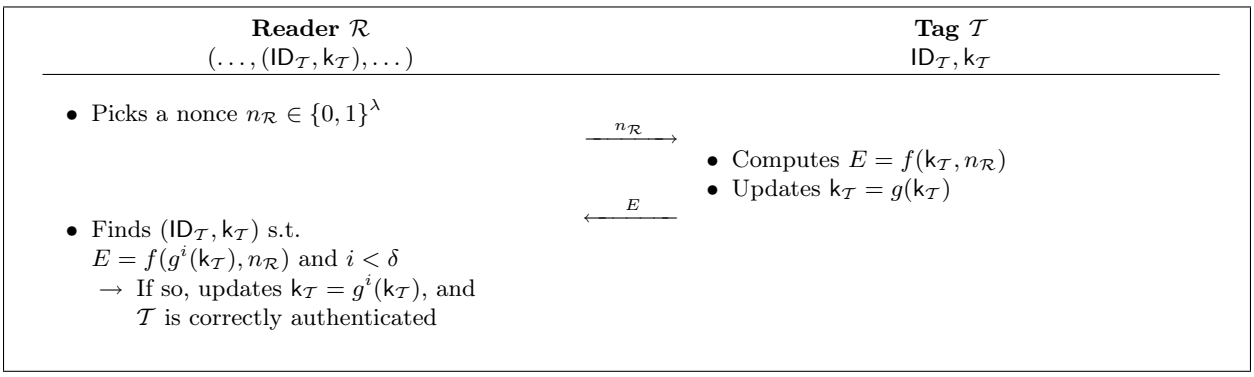

Fig. 3. OSK-based authentication protocol.

A significant attack on this kind of protocols has been defined by Juels and Weis in their privacy model [28], based on the fact that a tag's key can be updated while the equivalent one stored by the reader is not. Note that upon reception of a message E, $\mathcal{R}$ try to find a match with all tags' keys and their δ first updates. Thus, if the adversary $\mathcal{A}$ sends more than δ consecutive authentication requests to a tag without transferring the answers to $\mathcal{R}$, the "shared" secrets stored in $\mathcal{T}$ and $\mathcal{R}$ are consequently desynchronized. Therefore, if $\mathcal{A}$ has access to the authentication result on the reader's side, she is able to recognize the desynchronized tag $\mathcal{T}$ from another random tag as $\mathcal{T}$ will be rejected. This attack is generally called a *desynchronization attack*.

Recall that a NARROW $\mathcal{A}$ does not have access to the authentication result on the reader's side. On the other hand, a WIDE does have this access (e.g., through a RESULT request).

Considering a NARROW adversary, under the one-wayness assumption of g it is obviously infeasible to link a secret key to a previous authentication transcript as this is equivalent to invert g. Furthermore, since all tags' secrets are independent, then corrupting one tag does not allow to trace the others. Since $\mathcal{A}$ is restricted to be NARROW in Avoine's and DMR's models, the desynchronization attack does not work and thus the security level is equivalent to the one of the SK-based protocol, namely the protocol is respectively $\mathsf{Existential}-\mathsf{UNT}-\mathrm{RTE}$ (proved in [1]) and untraceable (proof similar to the one in [18]). Considering tags corruption, it is furthermore $\mathsf{Forward}-\mathsf{UNT}-\mathrm{RTEC}$ in Avoine's model (proved in [1]). Regarding Vaudenay's and HPVP models, the protocol is NARROW-DESTRUCTIVE-private (proved in [39, 25]).

On the other hand, when $\mathcal{A}$ is WIDE, the protocol is vulnerable to the desynchronization attack as explained above. Therefore, the protocol is not (r, s, t)-private for JW when $(r \geq 1, s > \delta, t > \delta)$ (proved in [28]), and not untraceable for CCEG. In LBM's model, a legitimate tag cannot be rejected in the ideal

world as the ideal functionality will always accept it while the desynchronization attack works in the real world.

For DLYZ, the choice of the set C of clean tags clearly influences the experiments' outputs. To show it, we use the same method provided in [16]. We consider that Sim runs as subroutine the underlying adversary $\mathcal{A}$. Sim_1 just runs basically $\mathcal{A}_1$.

If $|C| = 1$, then, at the end of this first phase, Sim knows if the challenge tag in C is desynchronized or not. Thus, Sim_2 is clearly able to simulate the interactions with the challenge tag (as described in the proof in [16]): the views produced by $\mathcal{A}$ and Sim will be indistinguishable. This is furthermore true, even if the final secrets of the challenge tag are revealed at the end of the experiment.

If $|C| = 2$ and one of the two tag has been desynchronized by $\mathcal{A}_1$, then $\mathcal{A}_2$ can distinguish these tags depending on the result of an execution in the second phase. But Sim_2 does not know which challenge tag has been chosen. Thus Sim_2 has to choose at random a tag (victim or not of the desynchronization attack) to simulate. At the end of the experiment, $\mathcal{A}$ is always able to retrieve the correct challenge tag, which is not the case of Sim. That implies that the views of $\mathcal{A}$ and Sim will be distinguishable. Therefore, the protocol is not private, because at least one adversary can produce a distinguishable view.

12.3 Undesynchronizable Authentication Protocol

Many undesynchronizable authentication protocols [10, 19, 38] have been proposed to counter the main drawback of OSK, that is the desynchronization attack. Here, we analyze O-FRAP, introduced by Le, Burmester and de Medeiros in [38].

At the setup, $\mathcal{T}$ is initialized with a couple containing a key and a nonce $(\mathsf{k}_\mathcal{T}, n_\mathcal{T})$, such that all tags' couples are independent. $(\mathsf{k}_\mathcal{T}, n_\mathcal{T})$ is stored by $\mathcal{R}$ as the current secrets $cur_\mathcal{T}$ of $\mathcal{T}$. Then a mutual authentication between $\mathcal{R}$ and $\mathcal{T}$ is performed, where $\mathcal{T}$'s key and/or nonce are updated at the end of the protocol execution by both entities. The main difference with OSK is that the tag always updates at least one value, even when the protocol is incomplete (in this case the random $n_\mathcal{T}$).

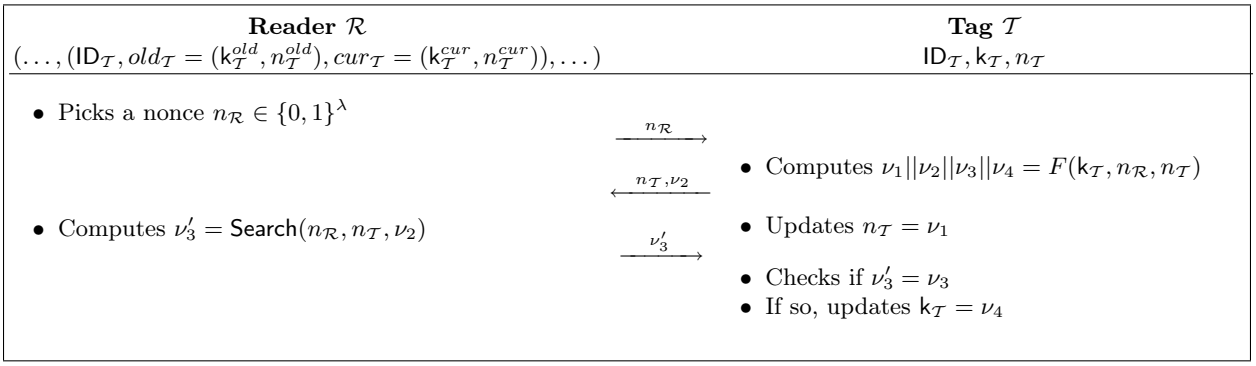

Fig. 4. O-FRAP authentication protocol.

The Search procedure is detailed in Algorithm 1 where $\mathsf{Update}(\mathcal{T})$ works as follows. First, if $\mathcal{R}$ uses $cur_\mathcal{T}$ to identify $\mathcal{T}$, then $\mathcal{R}$ replaces the content of $old_\mathcal{T}$ with the one of $cur_\mathcal{T}$. Secondly, $\mathcal{R}$ refreshes $cur_\mathcal{T} = (\mathsf{k}_\mathcal{T}^{cur}, n_\mathcal{T}^{cur})$ by (ν_4', ν_1').

Avoine, Coisel and Martin describe in [2] an attack which works when the adversary $\mathcal{A}$ is able to corrupt the challenge tag. This attack can be applied to the undesynchronizable protocols presented in [10, 19, 38]. First, $\mathcal{A}$ makes $\mathcal{T}$ and $\mathcal{R}$ start a new protocol execution, but $\mathcal{A}$ blocks the last message sent from $\mathcal{R}$ to $\mathcal{T}$. Then, if $\mathcal{A}$ corrupts $\mathcal{T}$ directly after this incomplete instance, then she is able to recognize $\mathcal{T}$ by recomputing ν_2 as $\mathsf{k}_\mathcal{T}$ has not been updated and the nonces $(n_\mathcal{R}, n_\mathcal{T})$ have been sent in the clear.

Algorithm 1 : The Search procedure

Input: $n_{\mathcal{R}}, n_{\mathcal{T}}, \nu_2$
Output: ν'_3

$Out \leftarrow \perp$
if $\exists(\mathsf{ID}_{\mathcal{T}}, old_{\mathcal{T}}, cur_{\mathcal{T}})$ s.t. $n_{\mathcal{T}} = n_{\mathcal{T}}^{old}$ (resp. $n_{\mathcal{T}} = n_{\mathcal{T}}^{cur}$) **then**
    $\nu'_1||\nu'_2||\nu'_3||\nu'_4 \leftarrow F(\mathsf{k}_{\mathcal{T}}^{cur}, n_{\mathcal{R}}, n_{\mathcal{T}}^{cur})$ (resp. $\nu'_1||\nu'_2||\nu'_3||\nu'_4 \leftarrow F(\mathsf{k}_{\mathcal{T}}^{old}, n_{\mathcal{R}}, n_{\mathcal{T}}^{old})$)
    if $\nu'_2 = \nu_2$ **then**
        $Out \leftarrow \nu'_3$
        Update($\mathcal{T}$)
    end if
end if
for all $(\mathsf{ID}_{\mathcal{T}}, old_{\mathcal{T}}, cur_{\mathcal{T}})$ **and** $c \in \{old_{\mathcal{T}}, cur_{\mathcal{T}}\}$ **do**
    $\nu'_1||\nu'_2||\nu'_3||\nu'_4 \leftarrow F(\mathsf{k}_{\mathcal{T}}^{c}, n_{\mathcal{R}}, n_{\mathcal{T}})$
    if $\nu'_2 = \nu_2$ **then**
        $Out \leftarrow \nu'_3$
        Update($\mathcal{T}$)
    end if
end for
return Out

Therefore, no CORRUPT query is allowed to an adversary $\mathcal{A}$ of this protocol. In that case, the desynchronization attack of OSK does not work here. As a consequence, for JW, Vaudenay, CCEG and HPVP, the privacy level of O-FRAP is the same as the one of the SK-based protocol (proofs are equivalent): it is respectively (r, s, t)-private, WEAK-private, untraceable, and WEAK-private.

In Avoine's and DMR's models, the protocol is Existential − UNT-RTE and untraceable: the attack presented above without corruption does not work since tags' keys are needed. The proofs are thus similar to the ones of the SK-based protocol. The protocol is furthermore Forward − UNT-RTEC for Avoine, because in that case, $\mathcal{C}$ can give to $\mathcal{A}$ non-consecutive intervals (contrary to the ones needed for the above attack): thus corrupting a tag does not help $\mathcal{A}$ to trace a tag.

Since the analysis in BLM's model only concerns complete execution of a protocol, this attack can be perfectly simulated in the ideal world using the knowledge of *active*(p) as proved in [38]. The protocol is thus forward-secure.

For DLYZ, remind that the challenge tag $\mathcal{T}_g$ is a clean tag, that is it does not have any unfinished protocol execution. Therefore, the previous attack is not possible. The protocol is ZK-private (the proof is similar to the one of the SK-based protocol).

Regarding the forward-ZK-privacy, it is possible to define an adversary $\mathcal{A}$ who has a distinguishable view than the simulator's one. Let consider that $|C| \geq 2$. Sim_1 just runs $\mathcal{A}_1$ as subroutine. Then $\mathcal{A}_2$ forces an interaction between $\mathcal{R}$ and $\mathcal{T}_g$, and blocks the last message. Sim_2 has to provide a simulated incomplete interaction of $\mathcal{R}$ with $\mathcal{T}_g$: since Sim_2 does not have any information about $\mathcal{T}_g$, this interaction can only be composed of random messages. At the end, $\mathcal{T}_g$'s secrets are revealed to a distinguisher D. Thus D is able to recognize if $\mathcal{A}_2$'s interaction corresponds to a real incomplete interaction with $\mathcal{T}_g$, or a simulated one. The protocol is therefore not forward-ZK-private.

12.4 Tree-Based Authentication Protocol

This protocol is based on the key-tree infrastructure given by Molnar and Wagner in [33].

In a system of n tags, a key-tree is generated with $\beta^d \geq n$ leaves, where d is its depth and β is its branching factor. Each leaf is randomly associated to a tag $\mathcal{T}$ of the system, and each node is associated to a partial key $\mathsf{k}_{i,j}$ where i is the depth of the node and j the branch.

We define w.l.o.g. $(p_0, p_1, p_2, \ldots, p_d)$ the path in the tree from the root (denoted p_0) to the leaf (denoted p_d) that is associated to the tag $\mathcal{T}$. At the setup of the system, $\mathcal{T}$ is initialized with a set of partial keys

$\{\mathsf{k}_{p_1}, \mathsf{k}_{p_2}, \ldots, \mathsf{k}_{p_d}\}$, where each k_{p_i} is the key attached to its path node p_i (except the root). $\mathcal{R}$ knows the entire tree arrangement, and thus all the keys associated to all the nodes.

The protocol is carried out in d rounds. For each round, $\mathcal{R}$ and $\mathcal{T}$ perform a challenge/response authentication as described in Fig. 5. If $\mathcal{T}$ answers correctly at each round, then $\mathcal{R}$ successfully authenticates $\mathcal{T}$ at the end of the last round.

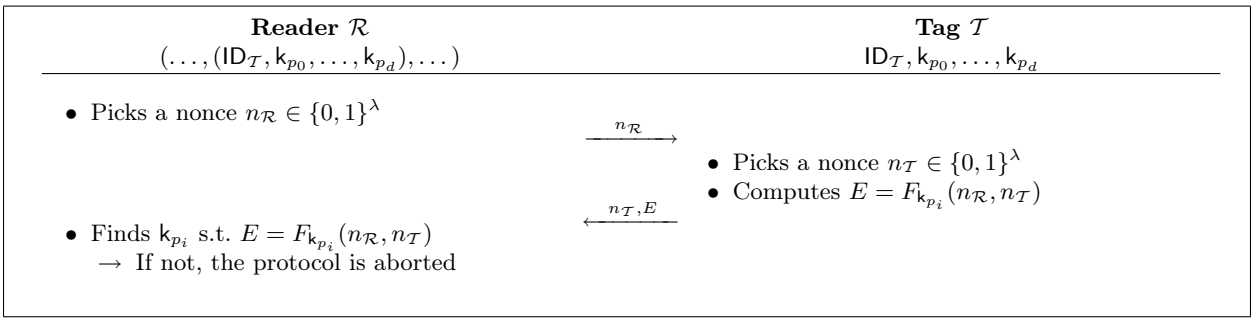

Fig. 5. One round of the tree-based authentication protocol.

In this protocol, the main drawback is that some partial keys are shared by several tags. For instance, let first say that a random tag $\mathcal{T}$ is chosen and corrupted: its secret keys $(\mathsf{k}_0, \mathsf{k}_{1,0}, \mathsf{k}_{2,0}, \ldots)$ are revealed. Then, let define the tags $\mathcal{T}_0$ and $\mathcal{T}_1$ as follows: $\mathcal{T}_0$'s keys are $(\mathsf{k}_0, \mathsf{k}_{1,0}, \mathsf{k}_{2,0}, \ldots)$, and $\mathcal{T}_1$'s keys are $(\mathsf{k}_0, \mathsf{k}_{1,0}, \mathsf{k}_{2,1}, \ldots)$. Clearly, $\mathcal{T}_0$ and $\mathcal{T}_1$ share the same path for the two first nodes, since they have the same keys for p_0 and p_1. But they have different keys for p_2. From the keys revealed during $\mathcal{T}$'s corruption, it is therefore possible to differentiate $\mathcal{T}_0$ and $\mathcal{T}_1$: $\mathcal{T}_0$'s answers will always be verifiable with $(\mathsf{k}_0, \mathsf{k}_{1,0}, \mathsf{k}_{2,0})$, but this is not the case for $\mathcal{T}_1$ since it does not use the revealed key $\mathsf{k}_{2,0}$. Note that, in the example, the challenge tags are not corrupted: only one other tag is corrupted.

Also, this protocol faces the same problem as the SK-based protocol: tag's corruption allows to trace it unconditionally. Thus for all the models, we consider that the adversary $\mathcal{A}$ is not allow to corrupt (at least) the challenge tags. Note that this option is not available in LBM's model, and this protocol is consequently not forward-private in this model.

In Avoine's model, since $\mathcal{A}$ only plays with the two challenge tags, the protocol does not suffer from the previous attack. Therefore, the protocol is $\mathsf{Existential}-\mathsf{UNT}-$RTE (same proof as for the SK-based protocol). For Vaudenay and HPVP, the protocol is WEAK-private, and untraceable for CCEG: clearly, since no secret is revealed, the proof is similar to the one for an SK-based protocol.

Then $\mathcal{A}$ is able to corrupt the non challenge tags in JW, and the tags that will not be part of her attack in DMR. Thus, the attack presented above appears in these two models. As a consequence, the protocol is not private for JW (explained in [28] and proved in [3, 6]), and not past-untraceable for CCEG.

For DLYZ, the same problem as for an OSK-based protocol appears. Let consider that $|C| \geq 2$. Remember that $\mathcal{A}_1$ and Sim_1 obtain several keys from the corruption of non clean tags in the first phase. Let also consider that $\mathcal{A}_1$ and Sim_1 returns a set C of clean tags where each tag in C can be easily recognizable, thanks to the revealed keys. Then $\mathcal{A}_2$ will be able the recognize the challenge tag. But, Sim_2 does not know which challenge tag has been chosen. Thus Sim_2 has to choose at random a tag to simulate. At the end of the experiment, $\mathcal{A}$ will always retrieve the correct challenge tag, contrary to Sim: the views of $\mathcal{A}$ and Sim will be distinguishable. Therefore, the protocol is not private.

12.5 PK-Based Challenge/Response Authentication Protocol

This protocol is the one presented by Vaudenay in [39]. $\mathcal{R}$ has a pair of public/private keys $(\mathsf{K_P}, \mathsf{K_S})$, and a tag $\mathcal{T}$ has a unique secret key $\mathsf{k}_\mathcal{T}$ known by $\mathcal{R}$. All the tags' keys are independent. The encryption scheme (Enc/Dec) is considered to be either IND-CPA or IND-CCA secure.

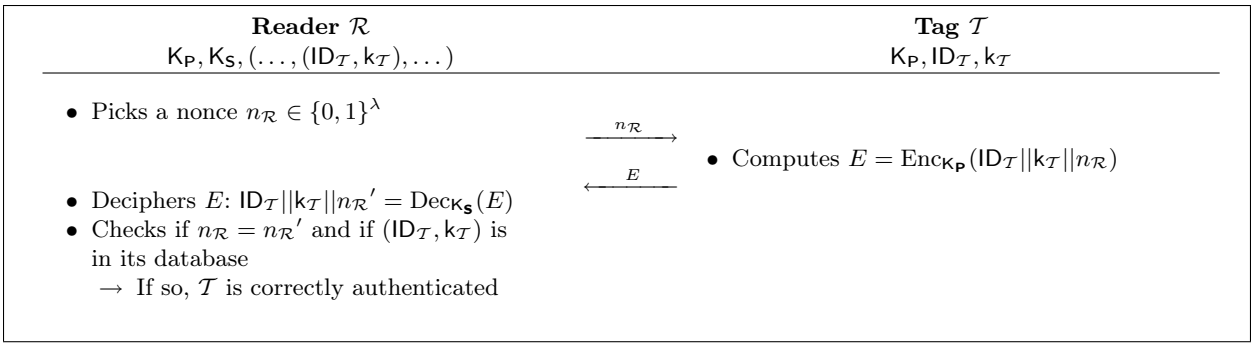

Fig. 6. PK-based authentication protocol.

First, it is important to notice that, under IND-CPA security, this protocol may not be easily proven private for WIDE adversary in any model. The main reason is that the simulator/blinder in the proof does not have access to a decryption oracle in the IND-CPA experiment. Therefore, this simulator/blinder is unable to simulate correctly the RESULT oracle, and thus has to answer at random 0 or 1 in some cases. Here, an adversary $\mathcal{A}$ may be able to detect if it is interacting with the real world or with a simulated one. CCEG prove in their model that untraceability can nevertheless be reached by PK-based protocols using IND-CPA cryptosystem, but adding other security mechanisms to the protocol (namely a MAC scheme).

For Avoine and DMR, since $\mathcal{A}$ is NARROW, this problem does not appear (i.e. no call to RESULT). The protocol is thus $\mathsf{Existential}-\mathsf{UNT}-\mathrm{RTE}$ and $\mathsf{Forward}-\mathsf{UNT}-\mathrm{RTEC}$ for Avoine when the cryptosystem is IND-CPA secure.

The proof is as follows in Avoine's model, but can be easily adapted for the DMR model. We show that, if there exists an adversary $\mathcal{A}$ who wins $Exp_{\mathcal{S},\mathcal{A}}^{P-\mathsf{UNT}}$ (with $P \in \{\mathsf{Existential}, \mathsf{Forward}\}$), then it is possible to construct an adversary $\mathcal{A}'$ who wins the IND-CPA game. To do so, $\mathcal{A}'$ runs $\mathcal{A}$ as subroutine, simulating the system $\mathcal{S}$ to $\mathcal{A}$ by answering to all oracles queries made by $\mathcal{A}$. At the end of the IND-CPA game, $\mathcal{A}'$ answers what $\mathcal{A}$ answers for $Exp_{\mathcal{S},\mathcal{A}}^{P-\mathsf{UNT}}$. Here, $\mathcal{A}'$ knows the secrets of $\mathcal{T}_0$ and $\mathcal{T}_1$ at the beginning of the IND-CPA game, in order to perform it. When $\mathcal{A}$ asks the interactions for $\mathcal{T}_0$ and $\mathcal{T}_1$, $\mathcal{A}'$ answers the corresponding ciphertexts for these interactions using the correct plaintext. When $\mathcal{A}$ asks the interactions for $\mathcal{T}$, then $\mathcal{A}'$ submits the plaintexts for both $\mathcal{T}_0$ and $\mathcal{T}_1$ for these interactions to the IND-CPA challenge $\mathcal{C}'$. $\mathcal{A}'$ receives the ciphertexts answered by $\mathcal{C}'$ for $\mathcal{T}_b$, where b is the unknown bit of the IND-CPA experiment, and transfers them to $\mathcal{A}$. So far, the simulation done by $\mathcal{A}'$ to $\mathcal{A}$ is perfect. Then, two cases can occur.

1. $\mathcal{A}$ does not need $\mathcal{T}$'s secrets (i.e. $\mathcal{A}$ is playing the Existential experiment). $\mathcal{A}$ wins $Exp_{\mathcal{S},\mathcal{A}}^{\mathsf{Existential}-\mathsf{UNT}}$, thus her advantage is non negligible, so is the advantage of $\mathcal{A}'$.
2. $\mathcal{A}$ asks $\mathcal{T}$'s secrets (i.e. $\mathcal{A}$ is playing the Forward experiment). $\mathcal{A}'$ does not know b, thus she sends at random $\mathcal{T}_0$'s or $\mathcal{T}_1$'s secrets. If $\mathcal{A}'$ sends the expected ones, then $\mathcal{A}$ wins $Exp_{\mathcal{S},\mathcal{A}}^{\mathsf{Forward}-\mathsf{UNT}}$, thus her advantage is non negligible, so is the advantage of $\mathcal{A}'$. If not, at worst $\mathcal{A}$ answers at random 0 or 1. Therefore, the whole advantage of $\mathcal{A}$ is non negligible, so is the advantage of $\mathcal{A}'$.

Consequently, $\mathcal{A}'$ is an adversary who wins the IND-CPA game with non negligible advantage, which concludes the proof.

Vaudenay proves in [39] that the protocol is NARROW-STRONG-private with IND-CPA security, and that it is furthermore FORWARD-private with IND-CCA. Since the privacy notions of JW are included in Vaudenay, the protocol is thus forward-(r, s, t)-private for JW. HPVP proves in [25] that the protocol is also NARROW-STRONG-private with IND-CPA security, but that it is STRONG-private with IND-CCA.

In LBM's model, if an environment is able to distinguish the real world from the ideal one, it can easily be transform into a distinguisher of the IND-CCA property of the underlying encryption scheme. Thus it is obvious that this protocol is forward-secure.

In CCEG's model, the protocol is future-untraceable with IND-CCA security (proved in [11]). In DLYZ's model, the protocol is also backward-ZK-private with IND-CCA security: the proof follows the same reasoning as the one of CCEG.

13 Analysis Summary and Conclusion

Table 2 sums up the security analysis of the studied protocols regarding each privacy model. From this, several results can be highlighted. In some models, several protocols are proved to ensure the same privacy level, while some attacks do not work for all these protocols. For example in Avoine's model, OSK-based, O-FRAP and PK-based protocols reach the same privacy (namely Existential−UNT−RTE and Forward−UNT−RTEC). However, as detailed in the previous section, the OSK-based protocol can be desynchronized contrary to the two others, and O-FRAP is vulnerable against an attack based on a tag corruption, while the PK-based protocol seems perfectly private. This fact happens in almost all models (e.g. {SK-based, O-FRAP, tree-based} for Vaudenay and CCEG, or {SK-based, OSK-based, O-RAP} for DMR). The main drawback of this fact is that a designer unfamiliar with privacy will probably choose the cheapest protocol (regarding the computing complexity), thinking that these protocols are equivalent regarding their privacy level. As presented in the previous section this is not necessarily the case.

Table 2. Analysis summary of the protocols. "✗" means no privacy.

Protocol	SK-based	OSK-based	O-FRAP	tree-based	PK-based
Avoine	Existential − UNT−RTE	Existential − UNT−RTE Forward − UNT−RTEC	Existential − UNT−RTE Forward − UNT−RTEC	Existential − UNT−RTE	Existential − UNT−RTE (if IND-CPA) Forward − UNT−RTEC (if IND-CPA)
JW	(r, s, t)-privacy	✗	(r, s, t)-privacy	✗	Forward-(r, s, t)-privacy (if IND-CCA)
Vaudenay	WEAK-privacy	NARROW-DESTRUCTIVE-privacy	WEAK-privacy	WEAK-privacy	NARROW-STRONG-privacy (if IND-CPA) + FORWARD-privacy (if IND-CCA)
LBM	✗	✗	Forward-security	✗	Forward-security
CCEG	Untraceability	✗	Untraceability	Untraceability	Future-Untraceability (if IND-CCA)
DLYZ	ZK-privacy	✗	ZK-privacy	✗	Backward-ZK-privacy (if IND-CCA)
HPVP	WEAK-privacy	NARROW-DESTRUCTIVE-privacy	WEAK-privacy	WEAK-privacy	NARROW-STRONG-privacy (if IND-CPA) STRONG-privacy (if IND-CCA)
DMR	Untraceability	Untraceability	Untraceability	✗	Untraceability (if IND-CPA)

Nevertheless, some models have features that permit to attribute different privacy levels to quite similar protocols. As an example, JW, DMR and DLYZ point out an important characteristic of protocols based on correlated secrets: they prove that the tree-based protocol is not secure, while the SK-based one is. This comes from the fact that an adversary may know some secrets without being authorized to corrupt the challenge tags. For instance, this adversary could be a tag's owner, who knows her tag's secrets, but who is not able to corrupt other tags. It seems consequently normal that the SK-based protocol is considered more private than the tree-based one. However, Avoine, Vaudenay, CCEG and HPVP classify the SK-based protocol and the tree-based protocol in the same privacy level.

Defining a model where an adversary is either WIDE or NARROW also seems a good feature for a privacy model. Among all the presented models, only Vaudenay and HPVP's model ensure this option. Nevertheless, this permits to grant some protocols with a reasonable privacy level (e.g. NARROW-DESTRUCTIVE for the OSK-based protocol and NARROW-STRONG for the IND-CPA-PK-based protocol), while some other models claim that the OSK-based ensures no privacy at all or cannot prove the privacy of the IND-CPA-PK-based protocol. But these protocols are clearly more private than the dummy identification protocol where tags send their identifier in the clear.

All the models (except Avoine and LBM) give the same privacy level for the SK-based protocol and for O-FRAP. This is a clear example about one of the issues found in these models. Indeed, these two protocols do not manage the tags' secrets in the same way: O-FRAP uses a key update mechanism, while nothing is done on the SK-based protocol regarding key update. For O-FRAP, even if the previously presented attack works, it only permits to link a freshly corrupted tag to its last previous incomplete protocol execution. But

all the previous completed ones are unlinkable. It is thus obvious that O-FRAP is more private than the SK-based protocol. This is unfortunately not highlighted by most of the models.

A remarkable fact is that the highest level of privacy defined by Vaudenay cannot be achieved (as proved in [39]). On the other hand, the highest privacy level of all the other models can be reached, at least with the IND-CCA-PK-based protocol. Unfortunately, to the best of our knowledge, nobody has been able to explain what is the privacy breach present in Vaudenay's model which is not considered in the other models. Does it means that the best privacy property can never be achieved or that Vaudenay's definition englobes more than the privacy notion?

To conclude this survey, it is clear that a protocol can be considered less private than another one in a given model, while the same protocols can be classified in the other way in another model. Thus, it may be really hard to choose one protocol rather than another one, just by analyzing their privacy level in several models. On the other hand, regarding only one model, two protocols can be sometimes classified with the same privacy, while some attacks may only affect one of them in practice. Consequently, the feeling is that no model permits to distinguish correctly the different privacy levels of all the existing protocols. This may represent a burden for application designers as they will not be able to choose the correct protocol corresponding to their needs.

References

1. Gildas Avoine. Adversary Model for Radio Frequency Identification. Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), Lausanne, Switzerland, 2005.
2. Gildas Avoine, Iwen Coisel, and Tania Martin. Time Measurement Threatens Privacy-Friendly RFID Authentication Protocols. In S.B. Ors Yalcin, editor, *Workshop on RFID Security – RFIDSec'10*, volume 6370 of *Lecture Notes in Computer Science*, pages 138–157, Istanbul, Turkey, June 2010. Springer.
3. Gildas Avoine, Etienne Dysli, and Philippe Oechslin. Reducing Time Complexity in RFID Systems. In Bart Preneel and Stafford Tavares, editors, *Selected Areas in Cryptography – SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 291–306, Kingston, Canada, 2005. Springer.
4. Gildas Avoine, Kassem Kalach, and Jean-Jacques Quisquater. ePassport: Securing International Contacts with Contactless Chips. In Gene Tsudik, editor, *Financial Cryptography and Data Security – FC'08*, volume 5143 of *Lecture Notes in Computer Science*, pages 141–155, Cozumel, Mexico, January 2008. IFCA, Springer-Verlag.
5. Gildas Avoine, Cédric Lauradoux, and Tania Martin. When Compromised Readers Meet RFID. In *Workshop on Information Security Applications – WISA'09*, volume 5932 of *Lecture Notes in Computer Science*, pages 36–50. Springer, 2009.
6. Gildas Avoine, Benjamin Martin, and Tania Martin. Tree-Based RFID Authentication Protocols Are Definitively Not Privacy-Friendly. In S.B. Ors Yalcin, editor, *Workshop on RFID Security – RFIDSec'10*, volume 6370 of *Lecture Notes in Computer Science*, pages 103–122, Istanbul, Turkey, June 2010. Springer.
7. Gildas Avoine and Philippe Oechslin. A Scalable and Provably Secure Hash Based RFID Protocol. In *International Workshop on Pervasive Computing and Communication Security – PerSec 2005*, pages 110–114, Kauai Island, HI, USA, 2005. IEEE.
8. Mike Burmester, Tri van Le, and Breno de Medeiros. Provably Secure Ubiquitous Systems: Universally Composable RFID Authentication Protocols. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2006*, pages 1–10, Baltimore, Maryland, USA, August–September 2006. IEEE, IEEE Computer Society.
9. Mike Burmester, Tri Van Le, Breno de Medeiros, and Gene Tsudik. Universally Composable RFID Identification and Authentication Protocols. *ACM Trans. Inf. Syst. Secur.*, 12(4), 2009.
10. Sébastien Canard and Iwen Coisel. Data Synchronization in Privacy-Preserving RFID Authentication Schemes. In *Workshop on RFID Security – RFIDSec'08*, Budapest, Hungary, July 2008.
11. Sébastien Canard, Iwen Coisel, Jonathan Etrog, and Marc Girault. Privacy-Preserving RFID Systems: Model and Constructions. Cryptology ePrint Archive, Report 2010/405, 2010.
12. Sébastien Canard, Iwen Coisel, and Marc Girauld. Security of Privacy-Preserving RFID Systems. In *IEEE International Conference on RFID-Technology and Applications – RFID-TA'10*, pages 269–274, Guangzhou, China, June 2010. Sun Yat-sen University, IEEE.
13. Ran Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. Cryptology ePrint Archive, Report 2000/067, 2000.

14. Ran Canetti. Security and Composition of Cryptographic Protocols: A Tutorial. Cryptology ePrint Archive, Report 2006/465, 2006.
15. Nicolas T. Courtois, Karsten Nohl, and Sean O'Neil. Algebraic Attacks on the Crypto-1 Stream Cipher in MiFare Classic and Oyster Cards. Cryptology ePrint Archive, Report 2008/166, 2008.
16. Robert H. Deng, Yingjiu Li, Moti Yung, and Yunlei Zhao. A New Framework for RFID Privacy. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *15th European Symposium on Research in Computer Security – ESORICS 2010*, volume 6345 of *Lecture Notes in Computer Science*, pages 1–18, Athens, Greece, September 2010. Springer.
17. Ton van Deursen. *Security of RFID Protocols*. PhD thesis, University of Luxembourg, Luxembourg, September 2011.
18. Ton van Deursen, Sjouke Mauw, and Saša Radomirović. Untraceability of RFID Protocols. In *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks*, volume 5019 of *Lecture Notes in Computer Science*, pages 1–15, Sevilla, Spain, 2008. Springer.
19. Tassos Dimitriou. A Lightweight RFID Protocol to protect against Traceability and Cloning attacks. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm*, Athens, Greece, 2005. IEEE.
20. Danni Dolev and Andrew C. Yao. On the Security of Public Key Protocols. *IEEE Transaction on Information Theory*, 29(2):198–208, March 1983.
21. Flavio D. Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter van Rossum, Roel Verdult, and Ronny Wichers Schreur. Dismantling MIFARE Classic. In Sushil Jajodia and Javier López, editors, *13th European Symposium on Research in Computer Security – ESORICS 2008*, volume 5283 of *Lecture Notes in Computer Science*, pages 97–114, Malaga, Spain, October 2008. Springer.
22. Flavio D. Garcia, Ichiro Hasuo, Wolter Pieters, and Peter van Rossum. Provable Anonymity. In *ACM Workshop on Formal Methods in Security Engineering – FMSE'05*, pages 63–72, Alexandria, VA, USA, November 2005. ACM, ACM Press.
23. Flavio D. Garcia and Peter van Rossum. Modeling Privacy for Off-line RFID Systems. In *9th Smart Card Research and Advanced Applications – CARDIS 2010*, volume 6035 of *Lecture Notes in Computer Science*, pages 194–208. Springer, 2010.
24. JungHoon Ha, SangJae Moon, Jianying Zhou, and JaeCheol Ha. A New Formal Proof Model for RFID Location Privacy. In Sushil Jajodia and Javier López, editors, *13th European Symposium on Research in Computer Security – ESORICS 2008*, volume 5283 of *Lecture Notes in Computer Science*, pages 267–281, Malaga, Spain, October 2008. Springer.
25. Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, and Bart Preneel. A New RFID Privacy Model. In *16th European Symposium on Research in Computer Security – ESORICS 2011*, Lecture Notes in Computer Science, Leuven, Belgium, September 2011. Springer.
26. Innovatron. Calypso Electronic Ticketing Standard, 1993.
27. International Organization for Standardization. ISO/IEC 9798: Information technology – Security techniques – Entity authentication, 1991–2010.
28. Ari Juels and Stephen Weis. Defining Strong Privacy for RFID. In *International Conference on Pervasive Computing and Communications – PerCom 2007*, pages 342–347, New York City, NY, USA, 2007. IEEE.
29. Gerhard de Koning Gans, Jaap-Henk Hoepman, and Flavio D. Garcia. A Practical Attack on the MIFARE Classic. In Gilles Grimaud and François-Xavier Standaert, editors, *Proceedings of the 8th Smart Card Research and Advanced Applications – CARDIS 2008*, volume 5189 of *Lecture Notes in Computer Science*, pages 267–282, Royal Holloway University of London, United Kingdom, September 2008. Springer.
30. Junzuo Lai, Robert H. Deng, and Yingjiu Li. Revisiting Unpredictability-Based RFID Privacy Models. In Jianying Zhou and Moti Yung, editors, *Proceedings of the 8th International Conference on Applied Cryptography and Network Security – ACNS 2010*, volume 6123 of *Lecture Notes in Computer Science*, pages 475–492, Beijing, China, June 2010. Springer.
31. Changshe Ma, Yingjiu Li, Robert H. Deng, and Tieyan Li. RFID Privacy: Relation Between Two Notions, Minimal Condition, and Efficient Construction. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *Conference on Computer and Communications Security – ACM CCS'09*, pages 54–65, Chicago, Illinois, USA, November 2009. ACM, ACM Press.
32. Sjouke Mauw, Jan Verschuren, and Erik P. de Vink. A Formalization of Anonymity and Onion Routing. In Pierangela Samarati, Peter Y. A. Ryan, Dieter Gollmann, and Refik Molva, editors, *9th European Symposium on Research in Computer Security – ESORICS 2004*, volume 3193 of *Lecture Notes in Computer Science*, pages 109–124, Sophia Antipolis, France, September 2004. Springer.

33. David Molnar and David Wagner. Privacy and Security in Library RFID: Issues, Practices, and Architectures. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *Conference on Computer and Communications Security – ACM CCS'04*, pages 210–219, Washington, DC, USA, October 2004. ACM, ACM Press.
34. Daisuke Moriyama, Shin'Ichiro Matsuo, and Miyako Ohkubo. Relation among the Security Models for RFID Authentication Protocol. In *ECRYPT Workshop on Lightweight Cryptography*, Louvain-la-Neuve, Belgium, November 2011.
35. Karsten Nohl, David Evans, Starbug, and Henryk Plotz. Reverse-Engineering a Cryptographic RFID Tag. In Paul C. van Oorschot, editor, *17th USENIX Security Symposium – USENIX'08*, pages 185–194, San Jose, California, USA, July 2008. USENIX.
36. Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic Approach to "Privacy-Friendly" Tags. In *RFID Privacy Workshop*, MIT, MA, USA, 2003.
37. Radu-Ioan Paise and Serge Vaudenay. Mutual Authentication in RFID: Security and Privacy. In Masayuki Abe and Virgil D. Gligor, editors, *Proceedings of the 3rd ACM Symposium on Information, Computer and Communications Security – ASIACCS'08*, pages 292–299, Tokyo, Japan, March 2008. ACM, ACM Press.
38. Tri Van Le, Mike Burmester, and Breno de Medeiros. Universally Composable and Forward-secure RFID Authentication and Authenticated Key Exchange. In Feng Bao and Steven Miller, editors, *ACM Symposium on Information, Computer and Communications Security – ASIACCS 2007*, pages 242–252, Singapore, Republic of Singapore, March 2007. ACM, ACM Press.
39. Serge Vaudenay. On Privacy Models for RFID. In *Advances in Cryptology - ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 68–87, Kuching, Malaysia, 2007. Springer.