Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2013, Article ID 710275, 26 pages
http://dx.doi.org/10.1155/2013/710275

*Research Article*

# Untangling RFID Privacy Models

**Iwen Coisel and Tania Martin**

*ICTEAM/Crypto Group and ICTEAM/GSI, Université Catholique de Louvain, 1348 Louvain-la-Neuve, Belgium*

Correspondence should be addressed to Tania Martin; tania.martin@uclouvain.be

Received 25 May 2012; Accepted 24 July 2012

Academic Editor: Agusti Solanas



The rise of wireless applications based on RFID has brought up major concerns on privacy. Indeed nowadays, when such an application is deployed, informed customers yearn for guarantees that their privacy will not be threatened. One formal way to perform this task is to assess the privacy level of the RFID application with a model. However, if the chosen model does not reflect the assumptions and requirements of the analyzed application, it may misevaluate its privacy level. Therefore, selecting the most appropriate model among all the existing ones is not an easy task. This paper investigates the eight most well-known RFID privacy models and thoroughly examines their advantages and drawbacks in three steps. Firstly, five RFID authentication protocols are analyzed with these models. This discloses a main worry: although these protocols intuitively ensure different privacy levels, no model is able to accurately distinguish them. Secondly, these models are grouped according to their features (e.g., tag corruption ability). This classification reveals the most appropriate candidate model(s) to be used for a privacy analysis when one of these features is especially required. Furthermore, it points out that none of the models are comprehensive. Hence, some combinations of features may not match any model. Finally, the privacy properties of the eight models are compared in order to provide an overall view of their relations. This part highlights that no model globally outclasses the other ones. Considering the required properties of an application, the thorough study provided in this paper aims to assist system designers to choose the best suited model.

## 1. Introduction

Radio Frequency IDentification (RFID) is a technology that permits identifying and authenticating remote objects or persons without line of sight. In a simple manner, a tag (i.e., a transponder composed of a microcircuit and an antenna) is embedded into an object and interacts with a reader when it enters within its electromagnetic field. The first use of RFID goes back to the early 1940s, during World War II, when the Royal Air Force deployed the IFF (Identify Friend or Foe) system to identify the Allies airplanes. Today, RFID is more and more exploited in many domains such as library management, pet identification, antitheft cars, anticounterfeiting, ticketing in public transportation, access control, or even biometric passports. It thus covers a wide ranging of wireless technologies, from systems based on low-cost tags (such as EPCs [1]) to more evolved ones operating with contactless smartcards [2, 3].

As predictable, some problems come up with this large-scale deployment. One general assumption of RFID systems is that the messages exchanged between the tags and the readers can easily be eavesdropped by an adversary. This raises the problem of information disclosure when the data emitted by a tag reveal details about its holder (called "information leakage"), but also when the eavesdropping of communications allows tracking a tag at different places or times (called "malicious traceability") and consequently its holder. Many articles pointed out the dangers of RFID with respect to privacy, and the authorities are now aware of this problem. For instance, Ontario Information and Privacy Commissioner Cavoukian aims to advocate the concept of "privacy-by-design" [4] which states that privacy should be put in place in every IT system before its widespread use. In 2009, the European Commissioner for Justice, Fundamental Rights and Citizenship issued a recommendation [5] which strongly supports the implementation of privacy in RFID-based applications.

Various researches have emerged these last years to fight against information leakage and malicious traceability in RFID. However, the search for a generic, efficient, and

secure solution that can be implemented in reasonably costly tags remains open [6–8]. Solutions are usually designed empirically and analyzed with ad hoc methods that do not detect all their weaknesses. In parallel, many investigations have been conducted to formalize the privacy notion in RFID. In 2005, Avoine was the earliest researcher to present a privacy model [9]. Since then, many attempts [10–22] have been carried out to propose a convenient and appropriate privacy model for RFID. But each one suffers from distinct shortcomings. In particular, most of these models generally do not take into account all the alternatives that a power may offer to an adversary. For instance, when an adversary is allowed to corrupt a tag, then several possibilities may arise: a corrupted tag could be either destroyed or not, and, in the last case, this tag could still be requested to interact within the system. At Asiacrypt 2007, Vaudenay introduced the most evolved RFID privacy model [22] known so far. However, this model is not as convenient as some protocol designers may expect, and they sometimes prefer to use a less comprehensive model to analyze a system. Consequently, providing an analysis and a comparison of the major RFID privacy models is meaningful to help designers in their choice. Such a work aims to highlight the strengths and weaknesses of each model. Su et al. already achieved a similar work in [23]. Unfortunately, they only focused on privacy notions and did not consider all the subtleties that are brought by different models. As a consequence, their study considers some models as weak, even though they offer interesting properties.

Our contribution is threefold. Firstly, in Sections 3 to 10, we chronologically present eight well-known models designed to analyze identification/authentication protocols preserving privacy. Some of them are very popular like [9, 16, 22]. Other ones have interesting frameworks like [12, 13, 18] (e.g., [18] is derived from the well-known universal composability framework). Other alternative models are attractive successors of [22], such as [11, 15]. Secondly, in Section 11, we analyze five different authentication protocols with each of these models in order to exhibit the lack of granularity of the state of the art. Finally, in Sections 12 and 13, we thoroughly compare the eight models regarding their different features and their privacy notions. We show that none of these models can fairly analyze and compare protocols. This fact is especially undeniable when the system's assumptions (that can differ from one system to another) are taken into account for an analysis.

## 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 kinds 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 denote $\mathcal{T}$ as a tag, $\mathcal{R}$ as the reader, and $\mathsf{DB}$ 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 execution $\pi$. This protocol can be an identification or an authentication protocol. We define an $i$-pass RFID protocol as being a protocol where $i$ messages are exchanged between $\mathcal{R}$ and $\mathcal{T}$.

The reader $\mathcal{R}$ is a powerful transceiver device whose computation capabilities approach the ones of a small computer. A tag $\mathcal{T}$ is a transponder with identifier $\mathsf{ID}_{\mathcal{T}}$. Its memory can vary from a hundred of bits (as for EPC tags [1]) to a few Kbytes (such as contactless smartcards [2, 3]). Its computation capabilities are generally much lower than a reader, but, depending on the tag, it can perform simple logic operations, symmetric-key cryptography, or even public-key cryptography. A tag is considered as *legitimate* when it is registered in the database $\mathsf{DB}$ as being an authorized entity of the system. The database $\mathsf{DB}$ 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 $\lambda$ as the security parameter of the system $\mathcal{S}$ and poly(·) as a polynomial function. Thus, we define $\varepsilon(\lambda) : \mathbb{N} \to \mathbb{R}$ as being a negligible function in $\lambda$ if, for every positive function poly(·), there exists an integer $N$ such that, for all $\lambda > N$, $|\varepsilon(\lambda)| < 1/\text{poly}(\lambda)$.

Then, we define all the different entities that may play a role in the presented privacy models. An *adversary* $\mathcal{A}$ is a malicious entity whose aim is to perform some attacks, either through the wireless communications between readers and tags (e.g., eavesdropping), or on the RFID devices themselves (e.g., corruption of a device and obtaining 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 $\mathcal{C}$.

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

We consider that $\mathcal{A}$ is able to play/interact with a tag when this last one is in $\mathcal{A}$'s neighborhood. At that moment, the tag is called by its pseudonym $\mathcal{T}$ (not by its identifier $\mathsf{ID}_{\mathcal{T}}$). During an attack, if a tag goes out and comes back to $\mathcal{A}$'s neighborhood, then it is considered that its pseudonym has changed. This notion is detailed in the Vaudenay model [22] (see Section 5). The same case happens when a set of tags is given to the challenger $\mathcal{C}$: when $\mathcal{C}$ gives the tags back to $\mathcal{A}$, their pseudonyms are changed.

*2.3. Procedures.* Most of the models studied in this paper focus on an RFID system $\mathcal{S}$ 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 define these procedures is detailed in the following. Note that this is just a generalization but it may be different in some models.

(i) $\mathsf{SetupReader}(1^\lambda)$ defines $\mathcal{R}$'s parameters (e.g., generating a private/public key pair $(\mathsf{K_S}, \mathsf{K_P})$) depending on the security parameter $\lambda$. It also creates an empty database DB which will later contain, at least, the identifiers and secrets of all tags.

(ii) $\mathsf{SetupTag}_{\mathsf{K_P}}(\mathsf{ID}_\mathcal{T})$ returns $\mathsf{k}_\mathcal{T}$, that is, 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 database DB of the reader.

(iii) 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, it can setup a new tag of identifier $\mathsf{ID}_\mathcal{T}$.

(i) CreateTag($\mathsf{ID}_\mathcal{T}$) creates a tag $\mathcal{T}$ with a unique identifier $\mathsf{ID}_\mathcal{T}$. It 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 $\mathcal{T}$.

(i) Execute($\mathcal{T}$) $\rightarrow$ ($\pi$, transcript) executes an Ident protocol between $\mathcal{R}$ and $\mathcal{T}$. It outputs the transcript of the protocol execution $\pi$, that is the whole list of the successive messages of the execution $\pi$.

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

(i) Launch() $\rightarrow$ $\pi$ makes $\mathcal{R}$ start a new Ident protocol execution $\pi$.

(ii) SendReader($m, \pi$) $\rightarrow$ $r$ sends a message $m$ to $\mathcal{R}$ in the protocol execution $\pi$. It outputs the response $r$ of the reader.

(iii) 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 obtain for the reader's result of a protocol execution $\pi$.

(i) Result($\pi$) $\rightarrow$ $x$: when $\pi$ is completed, it outputs $x = 1$ if $\mathsf{Output} \neq \perp$, and $x = 0$ otherwise.

And finally, it can corrupt a tag $\mathcal{T}$ in order to recover its secret.

(i) 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, some refinements will be provided on their definitions.

## 3. Avoine [9], 2005

In 2005, Avoine proposed the first privacy model for RFID systems. The goal was to analyze the untraceability notion of 3-pass protocols following the idea of communication intervals: the adversary $\mathcal{A}$ asks some oracles' queries on specific intervals of the targeted tags lives. The privacy notion behind this model represents the unfeasibility to distinguish one tag among two.

*3.1. The Oracles.* This model 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, that is, a SetupTag has already been performed on every tag.

Then $\mathcal{A}$ has only access to the following modified generic oracles adapted for 3-pass protocols. Instead of using the entities' names, Avoine uses the protocol executions names. Since $\mathcal{T}$ and $\mathcal{R}$ can run several protocol executions, $\pi^i_\mathcal{T}$ (resp., $\pi^j_\mathcal{R}$) denotes the $i$th (resp., $j$th) execution of $\mathcal{T}$ (resp., $\mathcal{R}$). These notations favor the precise description of $\mathcal{R}$'s and $\mathcal{T}$'s lifetimes.

(i) SendTag($m_1, m_3, \pi^i_\mathcal{T}$) $\rightarrow$ $r$ 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 execution $\pi^i_\mathcal{T}$ of $\mathcal{T}$.

(ii) SendReader($m_2, \pi^j_\mathcal{R}$) $\rightarrow$ $r$ sends the message $m_2$ to $\mathcal{R}$ in the protocol execution $\pi^j_\mathcal{R}$. It outputs $\mathcal{R}$'s answer $r$.

(iii) Execute($\pi^i_\mathcal{T}, \pi^j_\mathcal{R}$) $\rightarrow$ transcript executes a whole execution of the protocol between $\mathcal{T}$ and $\mathcal{R}$. This is done during the execution $\pi^i_\mathcal{T}$ of $\mathcal{T}$ and the execution $\pi^j_\mathcal{R}$ of $\mathcal{R}$. $\mathcal{A}$ obtains the whole transcript.

(iv) Execute$^*$($\pi^i_\mathcal{T}, \pi^j_\mathcal{R}$) $\rightarrow$ $\mathcal{R}$-transcript this is the same as the normal Execute. But it only returns the $\mathcal{R}$-transcript, that is, the messages sent by $\mathcal{R}$.

(v) Corrupt($\pi^i_\mathcal{T}$) $\rightarrow$ $\mathsf{k}_\mathcal{T}$: returns the current secret $\mathsf{k}_\mathcal{T}$ of $\mathcal{T}$ when the tag is in its $i$th execution.

The goal of the Execute$^*$ oracle is to simulate the fact that the forward channel (from reader to tag) has a longer communication range than the backward channel (from tag to reader) and therefore can be easily eavesdropped. It formalizes the asymmetry regarding the channels.

Two remarks are of interest for the Corrupt oracle. First, Corrupt can be used only once by $\mathcal{A}$. After this oracle query, $\mathcal{A}$ cannot use the other oracles anymore. Second, Corrupt is called on the tag execution number, and not the tag itself. This allows $\mathcal{A}$ to specify exactly the targeted moment of the tag's life.

During its attack, $\mathcal{A}$ has access to the oracles $\mathcal{O} \subset \{T, R, E, E^*, C\}$ = {SendTag, SendReader, Execute, Execute$^*$, Corrupt}.

Avoine denotes $\omega_i(\mathcal{T})$ as being the result of an oracle query 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, $\Omega_I(\mathcal{T}) = \{\omega_i(\mathcal{T}) \mid i \in I\} \cup \{\text{SendReader}(*, \pi_*^j) \mid j \in J\}$, 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 $\widehat{\Omega_I}(\mathcal{T})$ that maximizes $\mathcal{A}$'s advantage.

*3.2. Untraceability Experiments.* Avoine defines two experiments to represent two untraceability notions. They depend on $\lambda_{\text{ref}}$ and $\lambda_{\text{chal}}$, which represent, respectively, a reference length and a challenge length and which are function of the security parameter $\lambda$.

The first experiment given in Box 1 works as follows. First, $\mathcal{A}$ receives the interactions of a tag $\mathcal{T}$ during an interval $I$ that it chooses. Then, it receives the interactions of the challenge tags $\mathcal{T}_0$ and $\mathcal{T}_1$, also during the intervals $I_0$ and $I_1$ that it 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 $I_1$ cross the interval $I$ of $\mathcal{T}$. At the end, $\mathcal{A}$ has to decide which one of the challenge tags is the tag $\mathcal{T}$.

The second experiment given in Box 2 has the same mechanism. The only difference is that, now, $\mathcal{C}$ is the one that chooses the intervals $I_0$ and $I_1$ of the challenge tags, and not $\mathcal{A}$ anymore.

*3.3. Untraceability Notions.* From the experiments defined above, the notions of Existential-UNT and Universal-UNT are extended in this model, depending on restrictions about the choices of $I_0$ and $I_1$. Existential-UNT is when $\mathcal{A}$ chooses $I_0$ and $I_1$, whereas 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.

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

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

The notion of Universal$^-$ when the Corrupt oracle is used is called Forward-UNT.

*Definition 1* (untraceability [9]). An RFID system $\mathcal{S}$ is said to be $P$-UNT-$\mathcal{O}$ (for $P \in \{\text{Existential}, \text{Forward}, \text{Universal}\}$) if, for every adversary $\mathcal{A}$,

$$\left|\Pr\left(\text{Exp}_{\mathcal{S},\mathcal{A}}^{P\text{-UNT}}[\lambda_{\text{ref}}, \lambda_{\text{chal}}, \mathcal{O}] \text{ succeeds}\right) - \frac{1}{2}\right| \leq \varepsilon(\lambda_{\text{ref}}, \lambda_{\text{chal}}). \tag{1}$$

Direct implications are made from these notions:

$$\boxed{\text{Existential-UNT-}\mathcal{O} \Longrightarrow \text{Forward-UNT-}\mathcal{O} \Longrightarrow \text{Universal-UNT-}\mathcal{O}} \tag{2}$$

## 4. Juels and Weis [16], 2007

Two years after Avoine's publication, Juels and Weis proposed a new privacy model, referred in the sequel as JW, based on indistinguishability of tags. It intended to analyze classical challenge/response protocols based on symmetric-key cryptography (with possible additional messages in order to update the tags keys).

In their article, the authors highlighted that the Avoine model lacks two important features. Firstly, they proved that it is unable to catch an important attack on systems where tags have correlated secrets, because Avoine's adversary can only play with two tags. Secondly, they showed that Avoine did not have hindsight regarding all the possible attacks that can be performed on a protocol. The Avoine model does not capture all the relevant information that can be extracted from a protocol execution. For instance, it does not consider that $\mathcal{A}$ has access to any execution result. However, this simple "side information bit" allows formalizing a special kind of attacks on desynchronizable protocols like OSK, as explained in Appendix B.3. and in [24]. Therefore, the JW model aimed to fill that gap.

*4.1. Oracles.* At the initialization of the system, DB already stores all the tags' content, that is, a SetupTag has already been performed on every tag. Then $\mathcal{A}$ has access to the generic oracles Launch SendTag and SendReader, with the difference that the Output of SendReader includes the output of Result. It has furthermore access to the following oracles.

(i) $\text{TagInit}(\mathcal{T}) \rightarrow \pi$: when $\mathcal{T}$ receives this query, it begins a new protocol execution $\pi$ and deletes the information related to any existing execution.

(ii) $\text{SetKey}(\mathcal{T}, \mathsf{k}_{\mathcal{T}}^{\text{new}}) \rightarrow \mathsf{k}_{\mathcal{T}}$: 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 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. Note that its use and its result have an interesting feature: $\mathcal{A}$ is able to put any new key in the targeted tag: either the revealed one or a random one (that can be illegitimate).

*4.2. Privacy Experiment.* Let $\rho$, $\sigma$, and $\tau$ 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 $\mathcal{S}$. The privacy experiment is given in Box 3.

*4.3. Privacy Notions.* From the previous experiment, the JW model defines the following privacy property, where $\rho$, $\sigma$, and $\tau$ can be function of the system security parameter $\lambda$.

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

(1) $\mathcal{C}$ initializes the system $\mathcal{S}$.
(2) $\mathcal{A}$ requests $\mathcal{C}$ to receive a tag $\mathcal{T}$.
(3) $\mathcal{A}$ chooses $I$, queries *Oracle* $(\mathcal{T}, I, \mathcal{O})$ where $|I| \leq \lambda_{\text{ref}}$, and then receives $\widehat{\Omega_I}(\mathcal{T})$.
(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 $(I_0 \cup I_1) \cap I = \varnothing$.
(6) $\mathcal{A}$ queries *Oracle* $(\mathcal{T}_0, I_0, \mathcal{O})$ and *Oracle* $(\mathcal{T}_1, I_1, \mathcal{O})$, and then receives $\widehat{\Omega_{I_0}}(\mathcal{T}_0)$ and $\widehat{\Omega_{I_1}}(\mathcal{T}_1)$.
(7) $\mathcal{A}$ decides which of $\mathcal{T}_0$ or $\mathcal{T}_1$ is $\mathcal{T}$, and outputs a guess bit $b$.
$\mathrm{Exp}^{\text{Existential-UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{T} = \mathcal{T}_b$.

Box 1

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

(1) $\mathcal{C}$ initializes the system $\mathcal{S}$.
(2) $\mathcal{A}$ requests $\mathcal{C}$ to receive a tag $\mathcal{T}$.
(3) $\mathcal{A}$ chooses $I$, queries *Oracle* $(\mathcal{T}, I, \mathcal{O})$ where $|I| \leq \lambda_{\text{ref}}$, and then receives $\widehat{\Omega_I}(\mathcal{T})$. 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) $\mathcal{A}$ queries *Oracle* $(\mathcal{T}_0, I_0, \mathcal{O})$ and *Oracle* $(\mathcal{T}_1, I_1, \mathcal{O})$, and then receives $\widehat{\Omega_{I_0}}(\mathcal{T}_0)$ and $\widehat{\Omega_{I_1}}(\mathcal{T}_1)$.
(6) $\mathcal{A}$ decides which of $\mathcal{T}_0$ or $\mathcal{T}_1$ is $\mathcal{T}$, and outputs a guess bit $b$.
$\mathrm{Exp}^{\text{Universal-UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{T} = \mathcal{T}_b$.

Box 2

*Definition 2* ($(\rho, \sigma, \tau)$-privacy [16]). A protocol initiated by $\mathcal{R}$ in an RFID system $\mathcal{S}$ with security parameter $\lambda$ is $(\rho, \sigma, \tau)$-*private* if, for every adversary $\mathcal{A}$,

$$\left|\Pr\left(\mathrm{Exp}^{\text{JW-priv}}_{\mathcal{S},\mathcal{A}}[\lambda, n, \rho, \sigma, \tau] \text{ succeeds}\right) - \frac{1}{2}\right| \leq \varepsilon(\lambda). \quad (3)$$

Considering a variant of experiment $\mathrm{Exp}^{\text{JW-priv}}_{\mathcal{S},\mathcal{A}}$ where the "*except* $\mathcal{T}_b^*$" is removed from step (6.b), then forward-$(\rho, \sigma, \tau)$-privacy can be defined in the same way as the previous definition.

Note that, if $\mathcal{A}$ uses SetKey to put an illegitimate key in a tag, then this last one will possibly no longer be authenticated successfully by the reader. Nevertheless, whether this is performed on the nonchallenge tags or on $\mathcal{T}_b^*$ (only for the forward-$(\rho, \sigma, \tau)$-privacy experiment), this does not help $\mathcal{A}$ to find more easily the bit $b$ and thus does not influence its success to win the experiment.

## 5. Vaudenay [22], 2007

Later the same year, Vaudenay proposed formal definitions for RFID systems and adversaries and considered that a system $\mathcal{S}$ can be characterized by two notions: security and privacy. In this paper, we only present the privacy notion. Vaudenay's article followed some joint work done with Bocchetti [25], and its goal was to propose a comprehensive model that can formalize a wide range of adversaries. This characteristic is missing in the previous models and turns to be an asset of the Vaudenay model.

This model defines tags with respect to the adversary possibility to interact with them, as explained in Section 2.2. Clearly, when a tag is within $\mathcal{A}$'s neighborhood, it is said to be drawn and has a pseudonym so that $\mathcal{A}$ is able to communicate with the tag. In the opposite situation, a tag is said to be free (i.e., not drawn), and $\mathcal{A}$ cannot communicate with it. Consequently, the model considers that, at any given time, a tag can be either free or drawn. For example, the same tag with identifier $\mathrm{ID}_{\mathcal{T}}$ which is drawn, freed, and drawn again has two pseudonyms: $\mathcal{A}$ sees two different tags. Additionally, all the tags may not be accessible to $\mathcal{A}$ during all the attack: for instance, $\mathcal{A}$ may only play with two (drawn) tags during its attack.

*5.1. Oracles.* Contrary to the other previous models, DB is empty at the initialization of the system. Then $\mathcal{A}$ has access to all the generic oracles defined in Section 2.4. 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. It can also query the following ones.

(i) DrawTag(distr) $\rightarrow$ $(\mathcal{T}_1, b_1, \ldots, \mathcal{T}_k, b_k)$: following the distribution probability distr (which is specified by a polynomially bounded sampling algorithm), 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 $(\mathcal{T}_1, \ldots, \mathcal{T}_k)$ in any random order. If there is not

Experiment $\mathrm{Exp}^{\mathsf{JW\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda, n, \rho, \sigma, \tau]$

**Setup:**
(1) $\mathscr{C}$ initializes the system $\mathcal{S}$.
**Phase 1 (Learning):**
(2) $\mathcal{A}$ may do the following in any interleaved order:
  (a) Make Launch and TagInit queries, without exceeding $\rho$ and $\tau$ overall queries respectively.
  (b) Make arbitrary SetKey queries to any $(n-2)$ tags.
  (c) Make SendReader and SendTag queries, without exceeding $\sigma$ overall queries.
**Phase 2 (Challenge):**
(3) $\mathcal{A}$ selects two challenge tags $\mathcal{T}_i$ and $\mathcal{T}_j$ to which it 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) $\mathscr{C}$ 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 and TagInit queries, without exceeding $\rho$ and $\tau$ overall queries respectively.
  (b) Make arbitrary SetKey queries to any tag in the current tag set, *except* $\mathcal{T}_b^*$.
  (c) Make SendReader and SendTag queries, without exceeding $\sigma$ overall queries.
(7) $\mathcal{A}$ outputs a guess bit $b'$.
$\mathrm{Exp}^{\mathsf{JW\text{-}priv}}_{\mathcal{S},\mathcal{A}}$ succeeds if $b = b'$.

Box 3

enough free tags (i.e., less than $k$), or tags already drawn, then the oracle outputs $\perp$. It is further assumed that this oracle returns bits $(b_1, \ldots, b_k)$ telling if each of the drawn tags is legitimate or not. All relations $(\mathcal{T}_i, \mathsf{ID}_{\mathcal{T}_i})$ are kept in an *a priori* secret table denoted Tab.

(ii) Free($\mathcal{T}$) moves the tag $\mathcal{T}$ from the status drawn to the status free. $\mathcal{T}$ is unavailable from now on.

*5.2. Privacy Experiment.* From the oracles given above, Vaudenay defines five classes of polynomial-time adversary, characterized by $\mathcal{A}$'s ability to use the oracles.

*Definition 3* (adversary class [22]). An adversary class is said to be

(i) STRONG if $\mathcal{A}$ has access to all the oracles;

(ii) DESTRUCTIVE if $\mathcal{A}$ cannot use anymore a "corrupted" tag (i.e., the tag has been destroyed);

(iii) FORWARD if $\mathcal{A}$ can only use the Corrupt oracle after its first query to the Corrupt oracle;

(iv) WEAK if $\mathcal{A}$ has no access to the Corrupt oracle;

(v) NARROW if $\mathcal{A}$ has no access to the Result oracle.

*Remark 4.* The following relation is clear: WEAK $\subseteq$ FORWARD $\subseteq$ DESTRUCTIVE $\subseteq$ STRONG.

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

Vaudenay's privacy experiment is given in Box 4. $P$ is the adversary class, $P \in \{\varnothing, \mathsf{NARROW}\} \times \{\mathsf{WEAK}, \mathsf{FORWARD}, \mathsf{DESTRUCTIVE}, \mathsf{STRONG}\}$.

*5.3. Privacy Notions.* To define the privacy property of Vaudenay, it is first needed to define the notions of *blinder* (i.e., an algorithm able to simulate the answers of some specific oracles) and *trivial adversary* (i.e., an adversary that learns nothing about the system).

*Definition 5* (blinder, trivial adversary [22]). 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 it does not know the secret key nor the database.

A blinded adversary $\mathcal{A}^{\mathcal{B}}$ is itself an adversary that 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

$$\left|\Pr\left(\mathrm{Exp}^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}\right) - \Pr\left(\mathrm{Exp}^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}^{\mathcal{B}}}[\lambda] \text{ succeeds}\right)\right| \leq \varepsilon(\lambda). \tag{4}$$

*Definition 6* (privacy [22]). The RFID system $\mathcal{S}$ is said to be *P-private* if all the adversaries which belong to class $P$ are trivial following Definition 5.

The implications between Vaudenay's privacy notions are as follows:

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

(1) $\mathcal{C}$ initializes the system and sends $1^{\lambda}$, and $\mathsf{K_p}$ to $\mathcal{A}$.
(2) $\mathcal{A}$ interacts with the whole system, limited by its 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*.
$\mathrm{Exp}^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{A}$ returns *true*.

Box 4

$$\begin{array}{ccccccc} \mathsf{STRONG} & \Longrightarrow & \mathsf{DESTRUCTIVE} & \Longrightarrow & \mathsf{FORWARD} & \Longrightarrow & \mathsf{WEAK} \\ \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow \\ \mathsf{NARROW\text{-}STRONG} & \Longrightarrow & \mathsf{NARROW\text{-}DESTRUCTIVE} & \Longrightarrow & \mathsf{NARROW\text{-}FORWARD} & \Longrightarrow & \mathsf{NARROW\text{-}WEAK} \end{array} \quad (5)$$

The main result of Vaudenay is that STRONG-privacy is impossible, by proving that a DESTRUCTIVE-private protocol is not NARROW-STRONG-private. However, Vaudenay does not define which privacy level should be targeted by a protocol: it is never specified if NARROW-STRONG-privacy is better or not than DESTRUCTIVE-privacy.

Also, it is not explicit how the blinded adversary $\mathcal{A}^{\mathcal{B}}$ operates. Basically, there are two options: (i) $\mathcal{A}^{\mathcal{B}}$ aims the same probability than $\mathcal{A}$, or (ii) $\mathcal{A}^{\mathcal{B}}$ aims the same behavior than $\mathcal{A}$. It is obvious that the first option allows proving the privacy of some protocols which are actually not private, but this should be correctly formalized.

### 5.4. Extensions of the Model

*5.4.1. Model [21], 2008.* Paise and Vaudenay extended the Vaudenay model to analyze mutual authentication protocols. Actually, they enriched 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 formalizes the concept of *reader authentication*. Nevertheless, their extension does not modify the core of the Vaudenay model.

They also showed 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 claimed that the temporary memory of a tag should be automatically erased as soon as the tag is put back as free. However, this idea is not formalized in the paper.

This division between the persistent and the temporary memory of a tag has also been investigated by Armknecht et al. [26]. Based on the work of Paise and Vaudenay, they showed several impossibility results in attack scenarios with special uses of tag corruption.

*5.4.2. Model [20], 2011.* Ouafi presented in his thesis an adaptation of the Vaudenay model in order to counterVaudenay's impossibility result of STRONG-privacy. Concretely, the author proposed to incorporate the blinder with the adversary, so that the blinder has the knowledge of all the random choices and incoming messages made by the adversary. With this new definition of the blinder, Ouafi proved that STRONG-privacy can be ensured. This result is demonstrated with a public-key-based authentication protocol where the encryption scheme is IND-CCA2 secure and PA1+ plaintext-aware. (More details about these security notions can be found in [27].)

*5.4.3. Other Extensions.* The Vaudenay model has also been broadened in different works. In a nutshell, this is generally performed via the addition of a new oracle to the adversary capabilities (e.g., Timer in [28], MakeInactive in [29], or DestroyReader in [30]) and the corresponding new adversary class (e.g., the TIMEFUL class when $\mathcal{A}$ is allowed to use Timer).

## 6. Van Le et al. [10, 18], 2007

Also in 2007, van Le et al. introduced a privacy model in [18] (and an extended version in [10]) that is derived from the universal composability (UC) framework [31, 32] (and not on the oracle-based framework). Their aim was to provide security proofs of protocols under concurrent and modular composition, such that protocols can be easily incorporated in more complex systems without reanalyses. Basically, the model, denoted LBM in the following, is based on the indistinguishability between two worlds: the real world and the ideal one.

The transposition of RFID privacy into such a framework is a great contribution since universal composability is considered as one of the most powerful tools for security, especially when composition among several functionalities is required.

*6.1. UC Security.* General statements about the UC framework are briefly detailed in Appendix A for the reader

nonfamiliar with the field. Here, we present the security notion provided in such a framework.

To prove that an Ident protocol is as secure as the corresponding ideal functionality $\mathcal{F}$, no environment $\mathcal{Z}$ should distinguish if it is interacting with the real adversary $\mathcal{A}$ and Ident (i.e., the real world), or with the simulated adversary *Sim* and $\mathcal{F}$ (i.e., the ideal world). Consequently, $\mathcal{F}$ must be well defined such that all the targeted security properties are trivially ensured. Canetti formally defines this concept in [31] as follows, where PPT denotes probabilistic polynomial time turing machine.

*Definition 7* (UC-emulation [31]). A protocol Ident *UC-emulates* a protocol $\Phi$ if, for all PPT adversary $\mathcal{A}$, there exists a PPT simulated adversary *Sim* such that, for all PPT environment $\mathcal{Z}$, the distributions $\text{EXEC}_{\text{Ident},\mathcal{A},\mathcal{Z}}$ and $\text{EXEC}_{\Phi,Sim,\mathcal{Z}}$ are indistinguishable.

Based on this security framework, van Le et al. designed in [10, 18] several ideal functionalities to formalize anonymous authentication as well as anonymous authenticated key exchange.

*6.2. Description of the LBM Model.* The advantage of using this UC-based model is that all the possible adversaries and environments are considered during the security proof that can be carried out with LBM. In this paper, we only focus on the forward-security objective led by anonymous authentication.

*6.2.1. Assumptions of an RFID System $\mathcal{S}$.* First, the LBM model establishes that the reader $\mathcal{R}$ is the only entity that can start a protocol execution. Then, it considers that only tags can be corrupted by an adversary $\mathcal{A}$. Upon corruption of a tag, $\mathcal{A}$ obtains its keys and all its persistent memory values.

*6.2.2. The LBM Ideal Functionality $\mathcal{F}_{aauth}$.* This ideal functionality represents the *anonymous authentication* security objective of a given protocol. To do so, several parties (at least $\mathcal{R}$ and one tag) may be involved in a protocol execution. Two parties $\mathcal{P}$ and $\mathcal{P}'$ are said to be *feasible partners* if and only if they are, respectively, $\mathcal{R}$ and a tag. In the ideal world, communication channels between tags and $\mathcal{R}$ are assumed to be anonymous (meaning that they only reveal the type $type(\mathcal{P})$ of a party, either tag or reader), and a sent message is necessarily delivered to the recipient. Finally, $state(\mathcal{P})$ is the list of all the execution records, and $active(\mathcal{P})$ is the list of all the preceding incomplete executions (Box 5).

*6.2.3.* Forward-*Security.* When the adversary corrupts a tag $\mathcal{T}$, it gets its identifier $\text{ID}_{\mathcal{T}}$ and is then able to impersonate this tag using the IMPERSONATE command. A corrupted tag is thereafter considered as totally controlled by the adversary. Consequently, $\mathcal{F}_{aauth}$ will no longer manage the behavior of this corrupted tag and thus will reject every INITIATE command from this tag. As $state(\mathcal{T})$ is removed after a corruption, the adversary is not able to link the related tag to its previous authentication.

However, the adversary is able to link all the incomplete protocol executions of a corrupted tag $\mathcal{T}$ up to the last successfully completed one, based on the knowledge of $active(\mathcal{T})$. Thus, the ideal functionality obviously provides forward-security for all previous completed protocol executions.

## 7. Van Deursen et al. [13], 2008

The model of van Deursen et al., published in 2008, defines *untraceability* in the standard Dolev-Yao intruder model [33]. The untraceability notion is inspired by the anonymity theory given in [34, 35] and is used as a formal verification of RFID protocols. Such a technique is based on *symbolic protocol analysis* approach (and not on the oracle-based framework). This model will be called DMR in what follows.

*7.1. Definition of the System.* We remind below the basic definitions given in DMR.

First, the system is composed of 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, and server), each one specifying a set of actions. These actions depict the role specifications with a sequence 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, and XOR).

Each trace $t$ is composed of interleaved runs and run prefixes, denoted *subtraces*. A *run* of a role $R$ is a protocol execution from $R$'s point of view, denoted $R\#sid$, where *sid* 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 $\mathcal{A}$ is in the Dolev-Yao model and is characterized by its *knowledge*. This knowledge is composed of a set of run terms known at the beginning, and the set of run terms that it will observe during its attack. The adversary is allowed to manipulate the information of its knowledge to understand terms or build new ones. However, perfect cryptography is assumed (i.e., cryptographic primitives are assumed unbreakable and considered as black boxes). The inference of term $a$ from term set $K$ is denoted by $K \vdash a$.

Corrupted agents are modeled. (Note that, regarding corruption, there is no restriction about the role of such an agent: it can be either a tag or a reader.) $\mathcal{A}$ is given all the secrets of a corrupted agent in its initial knowledge. When an agent is corrupted, it is said to be "destroyed," that is, it cannot be used during $\mathcal{A}$'s attack. Yet, the security evaluation of a system is done on noncorrupted agents, that is, $\mathcal{A}$ cannot have access to the secret of an agent after the beginning of its attack.

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

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

Box 5

*7.2. Untraceability Notion.* First, the model defines several notions of *linkability*, *reinterpretation*, and *indistinguishability*, before giving the *untraceability* one.

*Definition 8* (linkability of subtraces [13]). Two subtraces $t_i^R$ and $t_j^R$ are linked, denoted by $L(t_i^R, t_j^R)$, if they are instantiated by the same agent:

$$L\left(t_i^R, t_j^R\right) \equiv \left(\text{agent}\left(t_i^R\right) = \text{agent}\left(t_j^R\right)\right). \tag{6}$$

The notion of *reinterpretation* has been introduced in [34] 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 9* (reinterpretation [13]). A map $\mu$ from run terms to run terms is called a reinterpretation under knowledge set $K$ if it and its inverse $\mu^{-1}$ satisfy the following conditions:

(i) $\mu(a) = a$ if $a$ is a basic run term,

(ii) $\mu(a) = (\mu(a_1), \ldots, \mu(a_n))$ if $a = (a_1, \ldots, a_n)$ is $n$-tuple,

(iii) $\mu(\{a\}_{\mathsf{k}}) = \{\mu(a)\}_{\mathsf{k}}$ if $K \vdash \mathsf{k}^{-1}$ or $(K \vdash a \wedge K \vdash \mathsf{k})$, and $\{\cdot\}_{\mathsf{k}}$ is an encryption under key $\mathsf{k}$,

(iv) $\mu(f(a)) = f(\mu(a))$ if $K \vdash a$ or $f$ is not a hash function.

Reinterpretations are used to define *indistinguishability* of traces.

*Definition 10* (indistinguishability of traces [13]). Let $K$ 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 $\mu$ under $K$, such that $\mu(t_i^R) = t'^R_i$ for all roles $R$ and subtraces $t_i^R$.

From all the above notions, the untraceability notion of a role is defined as follows.

*Definition 11* (untraceability [13]). An Ident protocol is said to be *untraceable* with respect to role $R$ if:

$$(\forall t \in \text{Traces}(\mathsf{Ident}))\ (\forall i \neq j)$$
$$\left(L\left(t_i^R, t_j^R\right) \Longrightarrow \left(\exists t' \in \text{Traces}(\mathsf{Ident})\right)\left(\left(t \sim t'\right) \wedge :L\left(t'^R_i, t'^R_j\right)\right)\right). \tag{7}$$

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

## 8. Canard et al. [11, 36], 2010

In the same vein as the Vaudenay model, Canard et al. proposed in 2010 a security model that comprises the properties of (strong) correctness, soundness, and untraceability. We only present the last notion. Contrary to Vaudenay, the authors only defined untraceability (and not privacy in general) and their main goal was to use the strongest adversary of the Vaudenay model. During the following, this model will be denoted CCEG.

*8.1. Oracles.* As for Vaudenay, DB is empty after the setup of the system, and a tag can be either free or drawn. Then $\mathcal{A}$ has access to all the generic oracles. It may also use the following ones.

(i) DRAWTAG$(k) \rightarrow (\mathcal{T}_1, \ldots, \mathcal{T}_k)$ 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 by $\mathcal{T}_i$ 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 $(\mathcal{T}_1, \ldots, \mathcal{T}_k)$ 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 an *a priori* secret table denoted by Tab.

(ii) FREE$(\mathcal{T})$ works exactly as the one of Vaudenay.

*8.2. Untraceability Experiment.* From the oracles given above, CCEG defines three classes of polynomial-time adversaries for the untraceability experiment.

*Definition 12* (adversary class [11]). An adversary class is said to be

(i) STRONG if $\mathcal{A}$ has access to all the oracles;

(ii) DESTRUCTIVE if $\mathcal{A}$ cannot use anymore a "corrupted" tag (i.e., the tag has been destroyed);

(iii) WEAK if $\mathcal{A}$ has no access to the CORRUPT oracle;

The authors do not define the NARROW adversary class introduced in the Vaudenay model (see Section 5 for more details). They consider that the model aims to be as powerful as possible: the NARROW notion weakens the adversary.

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 *nonobvious link*. As remark, links are chronologically ordered, that is, $(\mathcal{T}_i, \mathcal{T}_j)$ means that $\mathcal{T}_i$ has been freed before $\mathcal{T}_j$ has been drawn.

*Definition 13* (nonobvious link (NOL) [11]). $(\mathcal{T}_i, \mathcal{T}_j)$ is a *nonobvious link* if $\mathcal{T}_i$ and $\mathcal{T}_j$ refer to the same $\mathsf{ID}_{\mathcal{T}}$ in Tab and if a "dummy" adversary $\mathcal{A}_d$, that only has access to CREATETAG, DRAWTAG, FREE, and CORRUPT, is not able to output this link with a probability better than 1/2. Moreover, a nonobvious link is said to be

(i) *standard* if $\mathcal{A}$ has not corrupted $\mathcal{T}_i$ or $\mathcal{T}_j$;

(ii) *past* if $\mathcal{A}$ has corrupted $\mathcal{T}_j$;

(iii) *future* if $\mathcal{A}$ has corrupted $\mathcal{T}_i$.

Note that this model uses a "dummy" adversary $\mathcal{A}_d$, instead of a blinded adversary $\mathcal{A}^{\mathcal{B}}$ as in the Vaudenay model. Both adversaries are equivalent but not identical. Indeed, the main difference is that Vaudenay's blinder $\mathcal{B}$ 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 it is aware of its random choices.

A WEAK adversary is only able to output a *standard* NOL as it cannot query 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 from being 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 given in Box 6. $P$ is the adversary class, $P \in \{\text{STRONG}, \text{DESTRUCTIVE}, \text{WEAK}\}$.

*8.3. Untraceability Notions.* With the previous experiment, the CCEG untraceability of a system $\mathcal{S}$ is proved if no adversary is able to output a NOL with a probability better than the one of the dummy adversary $\mathcal{A}_d$.

*Definition 14* (untraceability [11]). An RFID system $\mathcal{S}$ is said to be standard-*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 $\mathcal{A}_d$ that only has access to oracles CREATETAG, DRAWTAG, FREE, and CORRUPT such that

$$\left|\Pr\left(\text{Exp}^{\text{CCEG-UNT}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}\right) - \Pr\left(\text{Exp}^{\text{CCEG-UNT}}_{\mathcal{S},\mathcal{A}_d}[\lambda] \text{ succeeds}\right)\right| \leq \varepsilon(\lambda). \tag{8}$$

Direct implications are made from these notions:

$$\boxed{\text{Future-untraceability} \Longrightarrow \text{Past-untraceability} \Longrightarrow \text{Standard-untraceability}} \tag{9}$$

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

## 9. Deng et al. [12], 2010

Also in 2010, Deng et al. proposed a new framework based on zero-knowledge formulation to define the security and privacy of RFID systems. Here, we only present the *zero-knowledge* privacy (denoted ZK-privacy), which is a new way of thinking in privacy for RFID. This model, denoted DLYZ in the sequel, is part of the *unpredictability models* family [12, 14, 17, 19]. They all rely on the unpredictability of the output returned by a tag or a reader in a protocol execution. In this paper, we decide to only present DLYZ since it is the most achieved model of this family.

*9.1. Considered Protocol.* This model considers that an RFID protocol execution $\pi$ is, w.l.o.g., always initialized by $\mathcal{R}$, and $\pi$ consists of $2\gamma + 1$ rounds for some $\gamma \geq 1$. Each protocol execution $\pi$ is associated to a unique identifier *sid*. At each execution, 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 outputs bits $o^{sid}_{\mathcal{R}}$ and $o^{sid}_{\mathcal{T}}$ (equal to 1 if $\mathcal{R}$ and $\mathcal{T}$ accept the protocol execution with identifier *sid*, or 0 otherwise) are publicly known. Note that the authors claim that each tag $\mathcal{T}$ has its output bit $o^{sid}_{\mathcal{T}} = 0$ if the authentication protocol is not mutual. However, we consider this fact too limiting since $\mathcal{T}$ can have an output (possibly known by $\mathcal{A}$), even if it may not authenticate the reader.

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

(1) $\mathcal{C}$ initializes the system and sends $1^\lambda$, $\mathcal{S}$'s public parameters param (including $\mathsf{K}_P$) to $\mathcal{A}$.
(2) $\mathcal{A}$ interacts with the whole system, limited by its class $P$.
(3) $\mathcal{A}$ returns one link $(\mathcal{T}_i, \mathcal{T}_j)$.
$\mathrm{Exp}^{\text{CCEG-UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds if $(\mathcal{T}_i, \mathcal{T}_j)$ is a NOL.

Box 6

For instance, its output can be "I arrived correctly at the end of the protocol on my side."

DLYZ assumes that a tag may participate to at most $s$ executions in its life with $\mathcal{R}$; thus $\mathcal{R}$ is involved in at most $sn$ executions, where $s$ is polynomial in $\lambda$ and $n$ is the total number of tags involved in the system.

*9.2. Oracles.* 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 $\mathsf{DB}^0$, and already stores the secrets of all the tags, that is, a SetupTag has already been performed on every tag. The only differences in the initialization are the following:

(i) SetupReader additionally generates $\mathcal{R}$'s initial internal state $\mathsf{s}^0_{\mathcal{R}}$;

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

This information is stored in $\mathsf{DB}^0$. 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 the adversary.

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

(i) $\textsc{Launch}() \to (\pi, m)$ makes $\mathcal{R}$ launch a new protocol execution $\pi$ and generates the 1st-round message $m$ which is also used as the execution identifier *sid*. If this is the $j$th new execution run by $\mathcal{R}$, then $\mathcal{R}$ stores $\eta_1 = m$ into its internal state $\mathsf{s}^j_{\mathcal{R}}$.

(ii) $\textsc{SendTag}(m, \mathcal{T}) \to r$ sends $m$ to $\mathcal{T}$. The output response $r$ of $\mathcal{T}$ is as follows.

(1) If $\mathcal{T}$ currently does not run any execution, then $\mathcal{T}$
  (a) initiates a new execution with identifier $sid = m$,
  (b) treats $m$ as the 1st-round message of the new execution,
  (c) and returns the 2nd-round message $(sid, r = \alpha_1)$.

(2) If $\mathcal{T}$ is currently running an incomplete execution with identifier *sid* and is waiting for the $u$th message from $\mathcal{R}$ $(u \geq 2)$, then $\mathcal{T}$ works as follows:
  (a) if $2 \leq u \leq \gamma$, $\mathcal{T}$ treats $m$ as the $u$th message from $\mathcal{R}$ and returns the next round message $(sid, r = \alpha_u)$;
  (b) if $u = \gamma + 1$ (i.e., the last-round message of the execution), $\mathcal{T}$ returns its output $o^{sid}_{\mathcal{T}}$ and updates its internal state to $\mathsf{s}^{v+1}_{\mathcal{T}}$ (where *sid* corresponds to the $v$th execution run by $\mathcal{T}$, where $1 \leq v \leq s$).

(iii) $\textsc{SendReader}(m, sid) \to r$ sends $m$ to $\mathcal{R}$ for the execution with identifier *sid*. After receiving $m$, $\mathcal{R}$ checks from its internal state whether it is running such an execution, and $\mathcal{R}$'s response $r$ is as follows.

(1) If $\mathcal{R}$ is currently running an incomplete execution 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:
  (a) if $u \leq \gamma$, $\mathcal{R}$ treats $m$ as the $u$th message from the tag and returns the next round message $r = \eta_{u+1}$;
  (b) if $u = \gamma$, $\mathcal{R}$ returns the last-round message $r = \eta_{\gamma+1}$ and its output $o^{sid}_{\mathcal{R}}$ and updates its internal state to $\mathsf{s}^{j+1}_{\mathcal{R}}$ and the database $\mathsf{DB}^{j+1}$ (where *sid* corresponds to the $j$th execution run by $\mathcal{R}$).

(2) In all the other cases, $\mathcal{R}$ returns $\perp$ (for invalid queries).

(iv) $\textsc{Corrupt}(\mathcal{T}) \to (\mathsf{k}^v_{\mathcal{T}}, \mathsf{s}^v_{\mathcal{T}})$ returns the secret key $\mathsf{k}^v_{\mathcal{T}}$ and the internal state $\mathsf{s}^v_{\mathcal{T}}$ currently held by $\mathcal{T}$. Once $\mathcal{T}$ is corrupted, all its actions are controlled and performed by $\mathcal{A}$.

For a completed protocol execution with identifier *sid*, the transcript of the exchanged messages is $(sid, \eta^{sid}_1, \alpha^{sid}_1, \ldots, \alpha^{sid}_\gamma, \eta^{sid}_{\gamma+1})$, excluding the entities' outputs.

Let $\mathcal{O}$ denote the set of these four oracles. $\mathcal{A}^{\mathcal{O}}(\mathcal{R}, T, \mathsf{param})$ denotes a PPT adversary $\mathcal{A}$ that takes on input the system public parameters param, the reader $\mathcal{R}$, and the tags set $T$ of the already initialized system. Then $\mathcal{A}$ interacts with $\mathcal{R}$ and the tags of $T$ via the four oracles. $\mathcal{A}'^{\mathcal{O}}(\mathcal{R}, \widehat{T}, \mathcal{J}(\mathcal{T}_c), \mathsf{aux})$ denotes a PPT adversary $\mathcal{A}'$ equivalent to $\mathcal{A}$, where $\mathsf{aux} \in \{0,1\}^*$ generally includes param

or some historical state information of $\mathcal{A}'$. Then $\mathcal{A}'$ interacts with $\mathcal{R}$ and the tags set $\widehat{T}$ via the four oracles. $\mathcal{A}'$ is said to have a *blinded access* to a *challenge* tag $\mathcal{T}_c \notin \widehat{T}$ if it interacts with $\mathcal{T}_c$ via a special interface $\mathcal{I}$ (i.e., a PPT algorithm which runs $\mathcal{T}_c$ internally and interacts with $\mathcal{A}'$ externally). To send a message $m$ to $\mathcal{T}_c$, $\mathcal{A}'$ sends a SendTag($m$, *challenge*) to $\mathcal{I}$; then $\mathcal{I}$ invokes $\mathcal{T}_c$ with SendTag($m, \mathcal{T}_c$) and answers $\mathcal{T}_c$'s output to $\mathcal{A}'$. $\mathcal{A}'$ does not know which tag is interacting with it. $\mathcal{A}'$ interacts with $\mathcal{T}_c$ via SendTag queries only.

*Definition 15* (clean tag [12]). A tag $\mathcal{T}$ is said to be *clean* if it is not corrupted (i.e., no query to Corrupt on $\mathcal{T}$) and is not currently running an incomplete execution with $\mathcal{R}$ (i.e., $\mathcal{T}$'s last execution is either finished or aborted).

The main goal of this definition is to force the adversary to use some uncorrupted and nonrunning tags to proceed the ZK-privacy experiment (see next section). This notion of nonrunning tags is very similar to the TagInit oracle of JW.

*9.3. 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 adversaries $(\mathcal{A}_1, \mathcal{A}_2)$ (resp., $(Sim_1, Sim_2)$) and runs in two stages. Note 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 given in Box 7 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 $\mathcal{T}_c$ 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.

Then, the second experiment given in Box 8 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 $\mathcal{T}_c$ 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.

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

*Definition 16* (ZK-privacy [12]). 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 $\lambda$ and any $n$ which is polynomial in $\lambda$, the following ensembles are computationally (resp., statistically) indistinguishable:

(i) $\{c, view_{\mathcal{A}}(\lambda, n)\}_{\lambda \in \mathbb{N}, n \in \text{poly}(\lambda)}$,

(ii) $\{c, sview(\lambda, n)\}_{\lambda \in \mathbb{N}, n \in \text{poly}(\lambda)}$.

That is, for any polynomial-time (resp., any computationally power unlimited) algorithm $\mathcal{D}$, it holds that

$$\left|\Pr[\mathcal{D}(\lambda, n, c, view_{\mathcal{A}}(\lambda, n)) = 1] - \Pr[\mathcal{D}(\lambda, n, c, sview(\lambda, n)) = 1]\right| = \varepsilon(\lambda). \quad (10)$$

The probability is taken over the random coins used during the system initialization, the random coins used by $\mathcal{A}$, *Sim*, $\mathcal{R}$, and all (uncorrupted) tags, the choice of $c$, and the coins used by the distinguisher algorithm $\mathcal{D}$.

*Definition 17* (Forward/Backward-ZK-privacy [12]). Let us denote $(\mathsf{k}^{\text{final}}_{\mathcal{T}_c}, \mathsf{s}^{\text{final}}_{\mathcal{T}_c})$ (resp., $(\mathsf{k}^{0}_{\mathcal{T}_c}, \mathsf{s}^{0}_{\mathcal{T}_c})$) the final (resp., initial) secret key and internal state of the challenge tag $\mathcal{T}_c$ at the end (resp., beginning) of $\text{Exp}^{\text{ZK-priv}}_{\mathcal{S},\mathcal{A}}$. 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 $\lambda$ and any $n$ which is polynomial in $\lambda$, the following distributions are indistinguishable:

(i) $\{\mathsf{k}^{\text{final}}_{\mathcal{T}_c}, \mathsf{s}^{\text{final}}_{\mathcal{T}_c}(\text{resp.}, \mathsf{k}^{0}_{\mathcal{T}_c}, \mathsf{s}^{0}_{\mathcal{T}_c}), c, view_{\mathcal{A}}(\lambda, n)\}$,

(ii) $\{\mathsf{k}^{\text{final}}_{\mathcal{T}_c}, \mathsf{s}^{\text{final}}_{\mathcal{T}_c}(\text{resp.}, \mathsf{k}^{0}_{\mathcal{T}_c}, \mathsf{s}^{0}_{\mathcal{T}_c}), c, sview(\lambda, n)\}$.

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

One justification of the authors on the way of corrupting $\mathcal{T}_c$ is that it is enough to give its secrets to $\mathcal{A}$ at the end. Another reason pointed out by the authors is that forward-ZK or backward-ZK-privacy cannot be achieved if $\mathcal{A}$ corrupts $\mathcal{T}_c$ before the end of the experiment.

## 10. Hermans et al. [15], 2011

Following the path opened by Vaudenay with his privacy model, Hermans et al. presented in 2011 a new model, denoted here HPVP, based on indistinguishability between two "worlds": it is most commonly called the "left-or-right" paradigm.

The main goal of the authors was to propose a model with a clear defined purpose, that is straightforward to use for proving privacy. Also as CCEG, HPVP aimed to use Vaudenay's strongest adversary.

*10.1. Oracles.* As for Vaudenay and CCEG, DB is empty after the initialization of the system, and a tag can be either free or drawn. Then $\mathcal{A}$ has access to the generic oracles CreateTag (here it additionally returns a reference $\mathcal{T}$ to the new created tag), SendReader, and Result. Then, $\mathcal{A}$ has also access to these other following oracles.

(i) DrawTag($\mathcal{T}_i, \mathcal{T}_j$) $\rightarrow$ $\mathcal{T}_{\text{drawn}}$ generates a drawn tag $\mathcal{T}_{\text{drawn}}$ and stores $(\mathcal{T}_{\text{drawn}}, \mathcal{T}_i, \mathcal{T}_j)$ in a table Tab. Depending on the bit $b$ chosen at the start of the privacy experiment (see next section), $\mathcal{T}_{\text{drawn}}$ 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 $\perp$.

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

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

Box 7

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

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

Box 8

(ii) $\textsc{Free}_b(\mathcal{T}_{\mathsf{drawn}})$ 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 nonvolatile memory (which contains its secret $\mathsf{k}_{\mathcal{T}}$).

This specific definition of the Free oracle comes from one important statement highlighted by Paise and Vaudenay in their model (see Section 5.4 for more details).

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

(i) $\textsc{Launch}() \rightarrow (\pi, m)$ makes $\mathcal{R}$ launch a new Ident protocol execution $\pi$, together with $\mathcal{R}$'s first message $m$.

(ii) $\textsc{SendTag}(m, \mathcal{T}) \rightarrow r$ 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 $\perp$.

(iii) $\textsc{Corrupt}(\mathcal{T}) \rightarrow \mathsf{k}_{\mathcal{T}}$ returns the whole memory (including 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 it queries this oracle. Then, Free specifies clearly that it erases the volatile memory of the chosen tag. Lastly, Corrupt is only authorized on a free tag. However, the intrinsic definition of a free tag (given in the Vaudenay model [22]) is that it is not accessible to $\mathscr{A}$, since it is not in its neighborhood. Thus, it seems impossible for $\mathscr{A}$ to query a Corrupt on a tag that it cannot manipulate (i.e., not drawn).

*10.2. 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 given in Box 9, where $P$ represents the adversary class: $P \in \{\varnothing, \mathsf{NARROW}\} \times \{\mathsf{WEAK}, \mathsf{FORWARD}, \mathsf{DESTRUCTIVE}, \mathsf{STRONG}\}$.

*10.3. Privacy Notions.* From the previous experiment, the HPVP privacy property is based on the adversary advantage to distinguish the two worlds.

*Definition 18* (privacy [15]). 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

$$\begin{aligned}&\left|\Pr\left(\mathrm{Exp}^{\mathsf{HPVP\text{-}Priv}}_{\mathcal{S},\mathcal{A}}[\lambda, 0]\ \text{ succeeds}\right)\right.\\&\qquad\left.+\Pr\left(\mathrm{Exp}^{\mathsf{HPVP\text{-}Priv}}_{\mathcal{S},\mathcal{A}}[\lambda, 1]\ \text{ succeeds}\right) - 1\right| \qquad (11)\\&= 0 \quad (\text{resp.} \le \varepsilon(\lambda))\,.\end{aligned}$$

Note that, all along 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. Yet, they never prove nor explain how their model can handle this issue, nor why this is indeed a privacy issue.

## 11. 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

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

(1) $\mathscr{C}$ initializes the system, chooses a random bit $b$, and sends $1^\lambda$ and $\mathcal{S}$'s public parameters $\mathsf{param}$ to $\mathcal{A}$.
(2) $\mathcal{A}$ interacts with the whole system, limited by its class $\mathscr{P}$.
(3) $\mathcal{A}$ outputs a guess bit $b'$.
$\mathrm{Exp}^{\mathsf{HPVP\text{-}priv}}_{\mathcal{S},\mathcal{A}}$ succeeds if $b = b'$.

Box 9

according to their building blocks and their underlying key infrastructure. The first protocol [37] is based on unique long-term secret key for each tag. On the contrary in the tree-based protocol [8], tags share between them some long-term partial secret keys so as to speed up the authentication. Two protocols [18, 38] use key-update mechanisms to increase the privacy level in case of tag corruption. In particular, the second one [18] provides mutual authentication in order to be undesynchronizable. The last analyzed protocol [22] is based on public-key cryptography. Due to their differences, these protocols may thus ensure different privacy levels. However, we will show in this section that some models assign the same privacy level to some protocols while other models clearly differentiate them, for example, by taking into account an attack which cannot be modeled in other models.

In the following, a tag $\mathcal{T}$ has a unique identifier $\mathsf{ID}_\mathcal{T}$ and should be authenticated by a legitimate reader $\mathcal{R}$.

*11.1. Analyzed Protocols.* The five RFID protocols chosen for this study are sketched in the following. Their complete descriptions and whole privacy analyses are detailed in Appendix B.

*11.1.1. SK-Based Challenge/Response Authentication Protocol.* The first studied protocol is the ISO/IEC 9798-2 Mechanism 2 [37] 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. All the tags' keys are independent.

*11.1.2. Tree-Based Authentication Protocol.* It is based on the key-tree infrastructure given by Molnar and Wagner in [8]. Basically in a system of $n$ tags, a key-tree is generated with $\beta^d \geq n$ leaves, where $d$ is its depth and $\beta$ 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 unique secret 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 $\mathsf{k}_{p_i}$ is the secret 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 Figure 2 of Appendix B.2. If $\mathcal{T}$ answers correctly at each round, then $\mathcal{R}$ successfully authenticates $\mathcal{T}$ at the end of the last round.

*11.1.3. OSK-Based Authentication Protocol.* The original OSK protocol [38] is an identification protocol, where there is no proof of the tag identity. At the setup, $\mathcal{T}$ is initialized with a unique secret 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, that is, 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-based protocol. The protocol presented here (proposed in [22]) is slightly different from OSK as $\mathcal{R}$ additionally sends a nonce to $\mathcal{T}$ in order to prevent replay attacks, as described in [6]. The resulting protocol ensures tag authentication rather than tag identification.

*11.1.4. O-FRAP Authentication Protocol.* Many undesynchronizable authentication protocols [18, 24, 39] have been proposed to counter the main drawback of OSK, that is the desynchronization attack. Here, we analyze O-FRAP, introduced by van Le et al. in [18].

At the setup, $\mathcal{T}$ is initialized with a couple containing a secret key and a nonce $(\mathsf{k}_\mathcal{T}, n_\mathcal{T})$, such that all the couples of tags 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 $n_\mathcal{T}$).

*11.1.5. PK-Based Challenge/Response Authentication Protocol.* It is one of the protocols given by Vaudenay in [22]. $\mathcal{R}$ 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 (indistinguishable under chosen-plaintext attack) or IND-CCA (indistinguishable under chosen-ciphertext attack) secure.

*11.2. Analysis Comparison.* Table 1 sums up the security analysis of the studied protocols regarding each privacy model.

*11.2.1. The Lack of Comprehensiveness.* In some models, several protocols are proved to ensure the same privacy level, because some attacks on these protocols cannot be formalized. For example in the Avoine model, OSK-based, O-FRAP, and PK-based protocols reach the same privacy (i.e., Existential-UNT-RTE and Forward-UNT-RTEC). However as detailed in Appendix B.3, the OSK-based protocol can be desynchronized contrary to the other two, and O-FRAP is subject to a specific attack based on tag corruption (see Appendix B.4), while the PK-based protocol is not vulnerable to such attacks. This misevaluation of privacy happens in almost all models (e.g., {SK-based, tree-based, O-FRAP} for Vaudenay, CCEG, and HPVP, or {SK-based, OSK-based, O-FRAP} for DMR). The main drawback of this fact is that system designers unfamiliar with privacy will probably choose the cheapest protocol (regarding the computing complexity), thinking that these protocols are equivalent regarding their privacy level.

*11.2.2. The Case of Correlated Secrets.* Nevertheless, some models have features that permit attributing 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 (as explained in Appendix B.2). For instance, this adversary could be a tag owner that only knows its tags' secrets and that is not able to corrupt other tags that it wants to trace. It is consequently normal that the SK-based protocol is more private than the tree-based one. Note that this differentiation cannot be established in the Avoine, Vaudenay, CCEG, and HPVP models because their adversary does not have the modularity to only corrupt certain tags. As a consequence, these models classify the SK-based and the tree-based protocols with the same privacy level.

*11.2.3. The Key-Update Mechanism Dilemma.* All the models (except Avoine and LBM) give the same privacy level for the SK-based protocol and for O-FRAP. This is another obvious example about the issue related to the privacy definitions of these models. Indeed, the two protocols do not manage the tags' secrets in the same way: a tag updates one of its secrets each time it starts an execution of O-FRAP, while a tag always keeps the same secret when it runs the SK-based protocol. For O-FRAP, the attack presented in Appendix B.4 only permits linking a freshly corrupted tag to its last previous incomplete protocol execution. But all the previous completed ones are unlinkable. This is not the case with the SK-based protocol, where a tag corruption allows tracing the tag at any time (past or future). This obvious distinction of the two protocols is however not highlighted by most of the models.

*11.2.4. Accuracy Refinement of the* NARROW *Adversary.* The NARROW nuance provided in some models permits granting some protocols with a reasonable privacy level. For instance, Vaudenay and HPVP confer NARROW-DESTRUCTIVE-privacy on the OSK-based protocol and NARROW-STRONG-privacy on the IND-CPA-PK-based protocol, while some other models argue that the OSK-based protocol ensures no privacy at all or that the IND-CPA-PK-based protocol cannot be proved private. These last claims are highly restrictive since these two protocols are clearly more private than the dummy identification protocol where tags send their identifier in the clear.

*11.2.5. The Vaudenay Problem.* Finally, Vaudenay proved in [22] that the highest privacy level of his model cannot be achieved. Yet, the highest privacy level of all the other seven presented models can be reached, at least with the IND-CCA-PK-based protocol. To the best of our knowledge, Ouafi is the only author who tries to explain in [20] that the Vaudenay model (i) does not reflect the exact notion of privacy that was targeted at first sight and (ii) may englobe more than only privacy. As explained in Section 5.4, Ouafi reformulates the Vaudenay model in order to achieve STRONG-privacy.

## 12. Classification of the Models

In this section, we compare the different features of all the privacy models presented in this paper. We point out which model(s) is(are) the most appropriate to use according to whether one of these features is wished or not. Table 2 sums up the features that are achieved by each model.

Note that "protocols" (resp., "tag-init protocols") refer to authentication/identification protocols where the reader (resp., tag) is the only entity that can start a protocol execution.

*12.1. Adversary Experiment.* Privacy models can be compared according to the similarities and differences of their experiment. To do so, we first need to define the notion of *challenge tags* in some models. Indeed, Vaudenay, LBM, DMR, CCEG, and HPVP do not stipulate this specific notion in their experiment. However, since their adversary must use some tags for its attack, we consider that all the tags are challenge ones. Note that the agents that can be corrupted before $\mathcal{A}$'s attack in the DMR model are considered as *nonchallenge tags*.

*12.1.1. Number of Tags Allowed in the Experiment.* Vaudenay, LBM, and CCEG are the only models where the adversary $\mathcal{A}$ is free to play with all the tags of the system at the same time during its attack.

At one moment of their experiment, JW and HPVP can only play with at most $(n - 1)$ tags (where $n$ is the total number of tags of the studied system). For the DLYZ model, the adversary cannot play with the set of clean tags it chose, except with the challenge tag $\mathcal{T}_c$ picked at random in this set. If this set contains only two tags, it can however play with at most $(n - 1)$ tags. Then, DMR's adversary cannot play with the agents that were corrupted before the beginning of its attack. Finally, the Avoine model is the most limiting one, since $\mathcal{A}$ can only play with two tags. This fact prevents

Table 1: Analysis summary of the protocols. "✗" means no privacy. For the PK-based protocol, a property followed by "*" means that it is at least achieved with IND-CPA-security. For the sake of clarity, we denote "N" and "DESTR" as being, respectively, "NARROW" and "DESTRUCTIVE."

| Model | Protocol | | | | |
|---|---|---|---|---|---|
| | SK based [37] | Tree based [8] | OSK based [22] | O FRAP [18] | PK based [22] |
| Avoine | Existential-UNT-RTE | Existential-UNT-RTE | Existential-UNT-RTE<br>Forward-UNT-RTEC | Existential-UNT-RTE<br>Forward-UNT-RTEC | Existential-UNT-RTE*<br>Forward-UNT-RTEC* |
| JW | $(\rho, \sigma, \tau)$-privacy | ✗ | ✗ | $(\rho, \sigma, \tau)$-privacy | Forward-$(\rho, \sigma, \tau)$-privacy |
| Vaudenay | WEAK-privacy | WEAK-privacy | N-DESTR-privacy | WEAK-privacy | N-STRONG-privacy*<br>FORWAD-privacy |
| LBM | ✗ | ✗ | ✗ | Forward-security | Forward-security |
| DMR | Untraceability | ✗ | Untraceability | Untraceability | Untraceability* |
| CCEG | Standard-untraceability | Standard-untraceability | ✗ | Standard-untraceability | Future-untraceability |
| DLYZ | ZK-privacy | ✗ | ✗ | ZK-privacy | Backward-ZK-privacy |
| HPVP | WEAK-privacy | WEAK-privacy | N-DESTR-privacy | WEAK-privacy | N-STRONG-privacy*<br>STRONG-privacy |

Table 2: Comparison of the presented privacy models. "✓" (resp., "✗") means that the feature is (resp., is not) given to the adversary $\mathcal{A}$. "N/A" means that the feature is not applicable in the model.

| Feature | Model | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Avoine | JW | Vaudenay | LBM | DMR | CCEG | DLYZ | HPVP |
| Interaction with all the tags | Only 2 tags | not $\mathcal{T}^*_{b\oplus 1}$ | ✓ | ✓ | not corrupted agents | ✓ | not all clean tags | all-but-one |
| Choice of the challenge tags | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Attack on incomplete executions | Both | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Corrupt challenge tags | Only $\mathcal{T}$ | Only $\mathcal{T}^*_b$ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ |
| Corrupt nonchallenge tags | N/A | ✓ | N/A | N/A | ✓ | N/A | ✓ | N/A |
| Corrupt any tag | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ |
| NARROW/WIDE | NARROW | WIDE | both | WIDE | NARROW | WIDE | WIDE | Both |
| Channels asymmetry | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Protocols analyzable | 3-pass with independent secrets | SK based | all | all | all | all | $(2\gamma + 1)$-pass | All |
| Tag-init protocols analyzable | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ |

the Avoine model from analyzing protocols with correlated secrets, which is not the case for all the other models.

Therefore, if $\mathcal{A}$ is allowed to play with all the tags of the system, then it is preferable to use the Vaudenay, LBM, and CCEG models for the privacy analysis.

*12.1.2. Choice of the Challenge Tags.* All the models (except the Avoine one) allow $\mathcal{A}$ to choose the challenge tags of its attack. In the Avoine model, the challenger $\mathcal{C}$ is the entity that performs this task, choosing $\mathcal{T}$, $\mathcal{T}_0$, and $\mathcal{T}_1$ (such that $\mathcal{T} = \mathcal{T}_0$ or $\mathcal{T}_1$). $\mathcal{A}$ has no option on the tags used for its attack: it is weaker than the adversaries of the other models. Thus, if it is considered that $\mathcal{A}$ has the possibility to choose the challenge tags, protocol should be analyzed with all the models except the Avoine one.

*12.1.3. Attack on Incomplete Protocol Executions.* In the JW, Vaudenay, DMR, CCEG, DLYZ, and HPVP models, $\mathcal{A}$ is allowed to perform its attack on incomplete protocol executions. As illustrated in Appendix B.4, it can start an execution with a tag and not finish it. Afterward, it can use this tag during its game to break its privacy. If $\mathcal{A}$ succeeds to do so, then the protocol is not considered as private.

For LBM, such an attack is not taken into account. $\mathcal{F}_{aauth}$ is designed such that all the successfully completed protocol executions of a tag are protected against corruption. In other words, $\mathcal{A}$ cannot learn any information about these previous executions, and thus the privacy of a tag is ensured. However, it is authorized to link the previous incomplete executions of a corrupted tag up to the last completed one without compromising the security.

For the Avoine model, both scenarios are allowed. During the Existential game, $\mathcal{A}$ chooses the intervals $I_0$ and $I_1$ of the challenge tags that help it the most to perform its attack. It can choose $I_0$ and $I_1$ such that these intervals are directly consecutive to $I$ (the interval of the targeted tag $\mathcal{T}$). In that case, nothing prevents $\mathcal{A}$ from using incomplete protocol executions during the experiment. For the Universal game, the challenger $\mathcal{C}$ is the one that chooses $I_0$ and $I_1$ that

help $\mathscr{A}$ the less, contrary to the Existential game. If $\mathscr{A}$ uses incomplete protocol executions, then $\mathscr{C}$ can choose nonconsecutive intervals such that the incomplete executions remain meaningless to $\mathscr{A}$ (as for LBM). For instance, some completed executions may separate the executions (completed or not) performed within the intervals.

Therefore, if a protocol must be protected against this attack, then Avoine, JW, Vaudenay, DMR, CCEG, DLYZ, and HPVP are the most appropriate models to study its privacy. If such a feature is not wished, then it can be analyzed with the Avoine and LBM models. Note that the Avoine model is the most flexible one since it can handle both scenarios.

*12.2. Tag Corruption.* The tamper resistance of RFID tags is a highly questionable assumption. Fortunately, all the models are flexible regarding the capacity of an adversary to corrupt tags. The two extreme cases are the impossibility to corrupt tags or the possibility to perform this action without restrictions. Yet, as detailed in the previous sections, intermediate levels of corruption have been introduced. To have an overall view of these levels, the models are gathered below based on their similarities from the weakest corruption level to the strongest one.

*12.2.1. Weak Adversary.* Obviously the weakest corruption level is when $\mathscr{A}$ is not allowed to corrupt tags. This feature is present in the Avoine, Vaudenay, LBM, CCEG and HPVP models. It permits formalizing the assumption of tags tamper resistance.

Although the JW, DMR and DLYZ models consider that it is always possible to corrupt non-challenge tags, they also define a weak level of corruption where $\mathscr{A}$ is not able to corrupt the challenge tags. This adversary, called *insider adversary* in [40], may be a tag owner that only knows its tags' secrets and that wants to break the privacy of other tags. As explained in Section 11.2 and in Appendix B.2, this subtle adversary can be used to perform a dedicated attack on a system with correlated secrets. However, even if this attack can be caught in other models by an overpowerful adversary (e.g., Vaudenay's FORWARD adversary), the Vaudenay, LBM, CCEG, and HPVP models are unable to precisely formalize such an intermediate adversary, since these models allow $\mathscr{A}$ to corrupt either every tag or any tag at all.

Therefore on the one hand, if it is assumed that $\mathscr{A}$ can never corrupt a tag, then the Avoine, Vaudenay, LBM, CCEG, and HPVP models should be chosen for a protocol analysis. On the other hand, if it is assumed that only the nonchallenge tags can be corrupted, then the most appropriate and fair models to use are JW, DMR, and DLYZ.

*12.2.2. Nonadaptive Adversary.* A higher level of corruption consists in authorizing $\mathscr{A}$ to only corrupt tags at the end of the experiment. It corresponds to the FORWARD adversary of Vaudenay and HPVP and to the Forward-UNT notion of Avoine. It can be viewed as a nonadaptive corruption ability as, except other corruptions, $\mathscr{A}$ cannot adapt its attack according to the corruption result.

The forward-ZK-privacy of DLYZ is close to this property since the last key of the challenge tag is given to the distinguisher at the end of the experiment. Yet in this case, $\mathscr{A}$ is still allowed to adaptively corrupt the nonchallenge tags during the experiment without stopping it. This fact slightly increases the strength of DLYZ's adversary.

*12.2.3. Destructive Adversary.* To increase the adversary power, some models give $\mathscr{A}$ the ability to pursue its attack after a corruption, leading to adaptive attacks regarding corruption. However, some constraints are still put into place in some models. In fact, the JW model considers that the challenge tags may be corrupted in the forward-$(\rho, \sigma, \tau)$-privacy, but only during the challenge phase. In other words, a tag corruption can only be used to trace its previous interactions. It is thus possible to establish a parallel between this constraint and the destructive corruption ability defined in other models (i.e., the DESTRUCTIVE adversary of Vaudenay, CCEG and HPVP, and the forward-security of LBM). Indeed, the key material obtained through a tag corruption may allow tracing its previous interactions but not the future ones as the tag is destroyed.

*12.2.4. Strong Adversary.* The strongest level that can be defined is obviously when $\mathscr{A}$ has no restriction regarding tag corruption. This corresponds to the STRONG adversary defined in the Vaudenay, CCEG, and HPVP models. A relatively similar notion is also defined by DLYZ, namely, the backward-ZK-privacy. However, as for the forward-ZK-privacy, while every nonchallenge tag may be corrupted during the experiment, the challenge tag cannot, and its initial key is only revealed at the end of the experiment. It may still help to distinguish the following interactions of this tag, but $\mathscr{A}$ cannot adapt its attack to this result. This consequently leads to a nonadaptive adversary that may be useful in some cases. Nevertheless, one may prefer the Vaudenay, CCEG, and HPVP models to catch the strongest adversary definition regarding corruption ability.

As a conclusion, the Vaudenay, CCEG, and HPVP models offer a wider adversary granularity regarding tag corruption. (Note that the CCEG's authors consider that FORWARD and DESTRUCTIVE adversaries (in Vaudenay's sense) are equivalent in their experiment: both are able to output a *standard* or *past* NOL, but not a *future* NOL. Therefore, a FORWARD adversary is useless in their model.) Only these three models take into account the strongest adversary which can corrupt with no restriction. Nevertheless, they do not consider the insider adversary that represents a relevant assumption and affords, to our mind, an interesting granularity for some analyses. In this case, protocols may thus be studied with a more appropriate model, namely, either JW, or DMR, or DLYZ.

*12.3. Other Features.* The remaining features of Table 2 are discussed in the following.

*12.3.1.* NARROW/WIDE *Adversaries.* As previously said, an adversary $\mathscr{A}$ is said to be NARROW (resp., WIDE) when

it does not (resp., does) receive the result of a protocol execution. Several models restrict their adversary with one of these features.

Avoine does not define a Result oracle, and there is no equivalence of such an oracle in DMR (since $\mathscr{A}$ does not know if a protocol between two agents succeeds). Both models only consider NARROW adversaries.

On the contrary, the adversaries of JW, LBM, CCEG, and DLYZ are only WIDE ones. For JW, there is no Result oracle defined in the model, but the adversary is forced to obtain the result of a protocol execution via the output of each SendReader. The DLYZ's adversary has the same behavior: it is forced to know this result information since $o_{\mathscr{R}}^{sid}$ and $o_{\mathscr{T}}^{sid}$ are public. In the LBM model, the output tape of each party is always available to $\mathscr{Z}$. Additionally, the adversary may also learn it as $\mathscr{Z}$ can communicate arbitrarily with it. Thus, it is impossible to model a NARROW adversary since the distinguisher may always know the result of a protocol execution. For CCEG, no NARROW adversary can be used for the untraceability experiment. Yet, as stressed in OSK's analysis given in Appendix B.3, this voluntary restriction implies that this kind of protocols with decent security features are not considered private.

The Vaudenay and HPVP models are the most flexible ones since it is possible to choose either a NARROW or a WIDE adversary. Note that the other models can however be (more or less easily) adapted to provide both adversary classes.

*12.3.2. Channels Asymmetry.* As already explained in Section 3, the forward channel (reader to tag) has a longer communication range than the backward channel (tag to reader). This characteristic is of interest as it has been shown in [41] that the former can be more easily eavesdropped than the latter in practice. Yet, the Avoine model is the only one that formalizes this feature through the Execute* oracle: $\mathscr{A}$ may only obtain the messages sent by $\mathscr{R}$ on the forward channel.

All the other models (as a matter of fact, created after the Avoine one) lost this feature and cannot represent this kind of weaker but realistic adversary. Thus, assuming that $\mathscr{A}$ is only able to get the messages sent from $\mathscr{R}$, the analysis must be performed with the Avoine model.

*12.3.3. Analyzable Protocols.* Some models are designed "by default" to analyze specific identification/authentication protocols. In the Avoine model, the oracles to interact with the system can only be used for 3-pass protocols. Then, JW's authors only aim to analyze protocols based on symmetric-key cryptography. Finally, DLYZ can only analyze $(2\gamma + 1)$-pass protocols with $\gamma \geq 1$.

On the contrary, Vaudenay, LBM, DMR, CCEG, and HPVP can analyze any identification/authentication protocol. Some of the restrictive models can nevertheless be adapted to analyze most existing protocols. For instance, the Avoine model can be slightly modified to analyze 2-pass classical challenge-response protocols, and the JW model does not forbid the analysis of protocols with public-key cryptography.

Finally, considering protocols where the tag starts an execution, JW and DMR are the only models that are not restricted by default to analyze such protocols.

## 13. Privacy Properties

In the previous section, we discussed the features that are present (or not) in each of the studied models. To conclude the investigation, we go a step further and compare the privacy properties between them.

This task is not an easy one as the different features of each model make it tough to compare them in some cases. Indeed in the following section, we highlight the fact that, when a privacy property of a given model is said to be "stronger" than the one of another model, the "weaker" model may present some features that are not present in the "stronger" one. We assume that system designers are aware of this fact and that, in this special case, they may thus prefer to use the weaker model for their privacy analysis. Except when this fact must be highlighted, we will not detail it in each comparison.

*13.1. Indistinguishability of Tags.* Regarding only the privacy notions, the Avoine and JW models are really close. Indeed, they both define privacy as the unfeasibility for an adversary to recognize one tag among two. The JW model has been designed after the Avoine one, as an improved model since it takes into account several flaws of the Avoine model. It can be easily proved that JW's $(\rho, \sigma, \tau)$-privacy (resp., forward-$(\rho, \sigma, \tau)$-privacy) implies Avoine's Existential-UNT (resp., Forward-UNT): the goal is the same and any request of an Avoine's adversary can be performed by a JW's adversary.

In the DMR model, the privacy property corresponds to the unfeasibility to link two *traces* that are produced by the same agent (in our case, a tag). This notion is also really close to the one defined in the JW model. Clearly for JW, the adversary capacity to retrieve the tag associated to the bit $b$ permits linking two traces and reciprocally. However, as the DMR model only defines a nonadaptive adversary regarding corruption, JW's $(\rho, \sigma, \tau)$-privacy is obviously stronger than DMR's untraceability.

Largely inspired by the design of the Vaudenay model (on which we will come back later), the CCEG and HPVP models offer a comprehensive list of oracles that permit any JW's adversary to be represented in their models. Regarding the privacy definition, it is obvious that the output of a JW's adversary is exactly a CCEG's nonobvious link (*standard* or *past*) and can thus be directly exploited by a CCEG's adversary. As a consequence, CCEG's standard-untraceability (resp., past-untraceability) property obviously implies JW's $(\rho, \sigma, \tau)$-privacy (resp., forward-$(\rho, \sigma, \tau)$-privacy). The reciprocal does not lead to a tight reduction. Indeed, a CCEG's adversary may shuffle the tags' pseudonyms several times (by performing successive DrawTag and Free queries), which are hard to simulate in the JW model.

The HPVP model defines privacy using the well-known "left-or-right" paradigm. As detailed in Section 10, it splits the tags space into two worlds. Nevertheless, a JW's adversary can be simulated in this model. First the HPVP's adversary

draws each tag of the system. (A single tag can be given as the two inputs of the DrawTag oracle.) Then, the two selected challenge tags of JW are freed and given as input of the DrawTag oracle. If the JW's adversary is able to recognize the outputted tag, then it may be used by an HPVP's adversary to output the guessed bit. Here again, the reciprocal is not true for the same reasons as for the CCEG model.

As a conclusion, assuming that privacy is defined as indistinguishability of tags, the most comprehensive models are HPVP and CCEG. Intuitively, these two models have equivalent privacy notions. Indeed, an adversary that succeeds in the HPVP experiment can easily output a nonobvious link. On the opposite, a nonobvious link permits distinguishing one tag from the others and can thus be used in the "left-or-right" paradigm. However, it is not obvious to formally prove this equivalence result due to the following facts. Firstly, at one moment of the HPVP experiment, the adversary must use (at least once) the DrawTag oracle on two different tags in order to obtain information about the challenge bit. At that moment, this adversary can no longer interact with all the tags whereas a CCEG's adversary can always interact with all the tags if it wants to. Secondly, a CCEG's adversary may draw more than one tag in a DrawTag request (e.g., three tags out of four). If an HPVP's adversary wants to use such an adversary as a subroutine to succeed in the HPVP experiment, the simulation of this fact entails that some choices are mandatory and thus leads to a nontight reduction.

*13.2. Real World versus Simulated World.* The last three models (i.e., Vaudenay, LBM, and DLYZ) define privacy as, in a nutshell, the unfeasibility to distinguish the interactions of an adversary against the real system from the interactions of a simulated adversary against a simulated world. In this second world, the simulator does not know the keys of the system. Nevertheless, when a tag corruption is asked, the tag's real secret key is returned. The idea behind this privacy notion is that, if there exists a distinction between these two worlds, then some information must leak from the messages of the real world (which contains the real keys of the system).

The most adaptive and comprehensive model using this principle is clearly the Vaudenay model. First, this model offers the widest range of adversaries. Then, these adversaries can be adaptive, contrary to the ones of DLYZ. Finally, as explained in Section 12.1, the LBM model only ensures the privacy of authentications prior to the last complete one, while the Vaudenay model considers privacy of all the possible authentications. As a consequence, for equivalent adversary classes, the Vaudenay model is stronger than LBM and DLYZ.

From another point of view, the UC framework is generally used to analyze protocols that are not run alone, but in parallel/concurrency with other protocols. Here, the interesting feature is that the environment $\mathcal{Z}$ can interact with the system and thus may help $\mathcal{A}$ to perform its 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. As a consequence, if the protocol to analyze is designed to belong to a complex system, its privacy may be studied in the LBM model. Nevertheless, if a strong privacy property is wished, the protocol should also be analyzed in the Vaudenay model.

*13.3. Between the Two Families.* The oracles description of the CCEG model is really close to the one of Vaudenay. The authors of the former describe their model as a restriction of the Vaudenay one, mainly on the experiment. Indeed, CCEG's adversary is required to output a nonobvious link, while any adversary assumption can be output in the Vaudenay model. Consequently, CCEG's privacy notion is intuitively weaker than Vaudenay's one (for equivalent adversary). Nevertheless, as proved in [11], CCEG's future-untraceability is a reachable property while Vaudenay's STRONG-privacy is impossible. Furthermore, to increase their result, CCEG's authors also prove with a "toy scheme" that their future-untraceability considers attacks that are not taken into account in the two "highest" reachable privacy levels of Vaudenay (i.e., the NARROW-STRONG and DESTRUCTIVE-privacy). As a consequence, the CCEG model defines a potentially weaker privacy notion, but, under this framework, protocol privacy can be studied against a stronger adversary than in the Vaudenay model.

Similar results may be proved for the HPVP model. First, its authors exhibit in their paper a protocol that ensures STRONG-privacy in their model. Then, using the "toy scheme" defined in [11], it can be proved that the same attacks (highlighted by CCEG) are also taken into account in HPVP's STRONG-privacy, which are again not considered in the reachable privacy levels of Vaudenay. However, as for the CCEG model, it can be proved that Vaudenay's privacy implies HPVP's one for equivalent adversary class. As this final result is not intuitive, we prove it in Appendix C.

To conclude this discussion, we highlight some existing results about the DLYZ model. The authors of the original paper argue that JW's $(\rho, \sigma, \tau)$-privacy does not imply ZK-privacy and used several schemes to illustrate their claim. One example is a system composed of only one tag. Clearly, such a scheme cannot be analyzed in the JW model since it requires at least two tags in the experiment. Thus, their claim that the proposed scheme is $(\rho, \sigma, \tau)$-private is doubtful. Additionally, the argument claiming that this scheme is not ZK-private is also not considered as acceptable, according to the authors of [42]. Furthermore, in such a special case of single-tag systems, DLYZ's authors say that ZK-privacy is reduced to the basic zero-knowledge definition which, according to them, provides a reasonable privacy. However in practice, each time this lonely tag is accepted by a reader, a WIDE adversary is obviously able to link this authentication to the previous ones. To our mind this is obviously a breach of privacy. Finally, the authors of [42] go one step beyond and formally prove that JW's $(\rho, \sigma, \tau)$-privacy is equivalent to ZK-privacy (Theorem 1 of [42]).

## 14. Conclusion

In this paper, we first presented eight of the most well-known existing privacy models for RFID in details. We exhibited and

discussed the differences between these models regarding their features and their privacy notions. As a preliminary conclusion, none of the existing models encompass all the others. The first reason is that no model offers enough granularity to provide all the features detailed previously. Even if it is sometime possible to extend an existing model to take into account a new property or a new assumption, it is not always a trivial task to add all of them.

Throughout our study, it appears that the Vaudenay model is the one that integrates the greatest number of features and which defines the strongest privacy notion. As a default choice, the Vaudenay model is probably the best one. Nevertheless, some drawbacks have been highlighted. Firstly, the strongest privacy property of this model cannot be ensured by any protocol. To study the security of a protocol against the strongest (known) adversary, one may thus prefer the CCEG of the HPVP model. Secondly, the Vaudenay model (as other ones) considers that tracing a tag after an incomplete protocol execution compromises the privacy. On the one hand, this is a relevant consideration that ensures a strong privacy level. On the other hand, relaxing this constraint helps to design more efficient protocols with a still reasonable privacy level using the Avoine and LBM models. Finally, the lack of granularity of all the models involves difficulties to fairly distinguish, in a given model, protocols with different security levels.

If system designers have precisely defined the requested properties of their application and the assumptions regarding potential adversaries, then they might use our results to select the most appropriate model. Thereby, they can design or select the most adapted and efficient protocol for their needs. Nevertheless, we are convinced that unifying and simplifying the models would help the community to design and compare protocols meaningfully.

## Appendices

## A. General Statements about the UC Framework

*A.1. The Environment $\mathcal{Z}$.* In the UC framework, $\mathcal{Z}$'s purpose is to manage the evolution of the system $\mathcal{S}$. In other words, this entity is in charge of the activation of all the parties, including the adversary $\mathcal{A}$. $\mathcal{Z}$ is the only entity able to request a party $\mathcal{P}$ to initiate a new execution of the studied Ident protocol. It is also able to read the output tapes of the system and $\mathcal{A}$'s parties. On the other hand, $\mathcal{Z}$ is not assumed to read the incoming and outgoing messages of the parties during a protocol execution.

While this new entity is quite unusual compared to the other privacy models in RFID, it permits formalizing systems where there is an underlying communication structure which may be unknown to the adversary. In the other models, $\mathcal{A}$ is in charge of the activation of the parties. As a consequence, if there exists an underlying activation sequence that is unknown to the adversary, it cannot respect it and thus may lose information that would help it to perform its attack. The potential activation scheduling performed by $\mathcal{Z}$ thus strengthens the power of the adversary.

*A.2. The Real World.* The system $\mathcal{S}$ is composed of several *honest parties* that interact together through an Ident protocol in order to achieve a well-defined objective.

An adversary $\mathcal{A}$ is in charge of the communication channels: it can eavesdrop, modify, and schedule all the communication channels between the honest parties in an arbitrary way. $\mathcal{A}$ may also be able to corrupt parties and obtain the full knowledge of their state. Corrupted parties are assumed to be totally controlled by $\mathcal{A}$ afterwards.

$\mathcal{Z}$ and $\mathcal{A}$ can be discussed in an arbitrary way. Consequently, if $\mathcal{A}$ wants to, it can forward all the communications to $\mathcal{Z}$. It can also ask $\mathcal{Z}$ to launch new executions of Ident. At the end of the experiment, $\mathcal{A}$ may send its final output to $\mathcal{Z}$ which is the last activated entity of the system. Then, $\mathcal{Z}$ outputs an arbitrary string, denoted by $\text{EXEC}_{\text{Ident},\mathcal{A},\mathcal{Z}}$, which can be reduced to one bit as proved by Canetti in [31, 32].

*A.3. The Ideal World.* Here, all the honest parties have access to the *ideal functionality* $\mathcal{F}$, that is a trusted and uncorrupted party. $\mathcal{F}$ must trivially ensure the desired security objectives of the Ident protocol, and does not depend on any cryptographic mechanism.

Equivalently to the adversary $\mathcal{A}$ in the real world, a simulated adversary *Sim* is defined such that *Sim* can arbitrarily discuss with $\mathcal{Z}$. However, *Sim* can no longer directly interact with parties: it can only communicate with the ideal functionality $\mathcal{F}$ which manages all the entities' communications. The main goal of *Sim* is to reproduce the behavior of $\mathcal{A}$ in the real world as faithfully as possible. Since (i) $\mathcal{A}$ may transfer messages of the Ident protocol to $\mathcal{Z}$, (ii) *Sim* does not have access to Ident, and (iii) $\mathcal{F}$ does not produce such messages, then *Sim* should simulate these messages to $\mathcal{Z}$. The final output of $\mathcal{Z}$ is denoted by $\text{EXEC}_{\Phi,Sim,\mathcal{Z}}$, where the protocol $\Phi$ UC-realizes the ideal functionality $\mathcal{F}$ (as defined in [31]).

## B. Detailed Privacy Analysis of Five Protocols

In the following, $F$ and $G$ refer to pseudorandom functions, while $f$ and $g$ refer to one-way functions. (Enc/Dec) refers to an encryption scheme. Finally, $\lambda$ denotes the security parameter of the system.

*B.1. SK-Based Challenge/Response Authentication Protocol [37].* In this protocol, it is obvious that one single corruption of a tag $\mathcal{T}$ allows it to be traced at any time. This is feasible as the secret key of a tag is a fixed value and the nonces used in the pseudorandom 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. (Note that this equality can be due to a collision, but this happens with a negligible probability.) Nevertheless, the corruption of another tag $\mathcal{T}'$ does not help to trace the tag $\mathcal{T}$, since all the 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 Existential-UNT-RTE in the Avoine model (proved for this kind of protocols in [9]), and $(\rho, \sigma, \tau)$-private in the JW model (proved in [16]). It is untraceable for DMR (proved in [13]) and ZK-private in the DLYZ model (the proof of a similar protocol in [12] can be trivially adapted).

This protocol is WEAK-private for Vaudenay (proved in [22]) and for HPVP. It is standard-untraceable for CCEG. The proofs for HPVP and CCEG are very similar to the ones of Vaudenay.

Finally, this protocol cannot UC-emulate the ideal functionality in the LBM model as the attack presented here permits an adversary to link several executions while this is not possible for the simulator (as $state(\mathcal{T})$ is removed after a corruption).

*B.2. Tree-Based Authentication Protocol [8].* In this protocol, the main drawback is that some partial keys are shared by several tags. For instance, let us first say that a random tag $\mathcal{T}$ is chosen and corrupted: its secret keys $(k_0, k_{1,0}, k_{2,0}, \ldots)$ are revealed. Then, let us define the tags $\mathcal{T}_0$ and $\mathcal{T}_1$ as follows: $\mathcal{T}_0$'s keys are $(k_0, k_{1,0}, k_{2,0}, \ldots)$, and $\mathcal{T}_1$'s keys are $(k_0, k_{1,0}, k_{2,1}, \ldots)$. Clearly, $\mathcal{T}_0$ and $\mathcal{T}_1$ share the same path for the first two 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 $(k_0, k_{1,0}, k_{2,0})$, but this is not the case for $\mathcal{T}_1$ since it does not use the revealed key $k_{2,0}$. 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: the corruption of a tag allows tracing it unconditionally. Thus for all the models, we consider that the adversary $\mathcal{A}$ is not allowed to corrupt (at least) the challenge tags. Note that this option is not available in LBM, and this protocol is consequently not forward-secure in this model.

It should not be possible to study this kind of protocols in the Avoine model because of the correlated secrets, but the analysis is given here to show the contrasts between the different models. Thus in the Avoine model, since $\mathcal{A}$ only plays with the two challenge tags, the protocol does not suffer from the previous attack. Therefore, the protocol is Existential-UNT-RTE (same proof as for the SK-based protocol). For Vaudenay and HPVP, the protocol is WEAK-private, and standard-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 nonchallenge tags in JW, and the tags that are not part of its attack in DMR. Thus, the attack presented above can be formalized in these two models. Consequently, the protocol is not $(\rho, \sigma, \tau)$-private for JW (explained in [16] and proved in [6, 43]) and not untraceable for DMR.

For DLYZ, we use the method provided in [12] to show that the protocol is not ZK-private. We consider that *Sim* runs as subroutine the underlying adversary $\mathcal{A}$. $Sim_1$ just runs basically $\mathcal{A}_1$, and both adversaries obtain several keys from the corruption of nonclean tags in the first phase. Let us also consider that $\mathcal{A}_1$ and $Sim_1$ return a set $C$ of clean tags where (i) $|C| \geq 2$ and (ii) each tag in $C$ can be easily recognizable, thanks to the revealed keys. Then $\mathcal{A}_2$ will be able to 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 ZK-private.

*B.3. OSK-Based Authentication Protocol [22].* A significant attack on this kind of protocols has been defined by Juels and Weis in their privacy model [16], 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 receipt of a message $E$, $\mathcal{R}$ tries to find a match with all tags' keys and their $\delta$ first updates. Thus, if the adversary $\mathcal{A}$ sends more than $\delta$ 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, it is able to recognize a 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 adversary does not have access to the authentication result on the reader's side, while a WIDE one does have this access (e.g., through a Result query).

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 tracing the other ones. Since $\mathcal{A}$ is restricted to be NARROW in the Avoine and DMR models, the desynchronization attack does not work and thus the security level is equivalent to the one of the SK-based protocol (Figure 1), namely, the protocol is, respectively, Existential-UNT-RTE (proved in [9]) and untraceable (proof similar to the one in [13]). Considering tag corruption, it is furthermore Forward-UNT-RTEC in the Avoine model (proved in [9]). Regarding the Vaudenay and HPVP models, the protocol is NARROW-DESTRUCTIVE-private (proved in [15, 22]).

When $\mathcal{A}$ is WIDE, the protocol is vulnerable to the desynchronization attack explained above. Therefore, the protocol is not $(\rho, \sigma, \tau)$-private for JW when $(\rho \geq 1, \sigma > \delta, \tau > \delta)$ (proved in [16]), and not standard-untraceable for CCEG. In the LBM 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 same problem as for the tree-based protocol appears. If $|C| = 2$ and one of the two tags 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

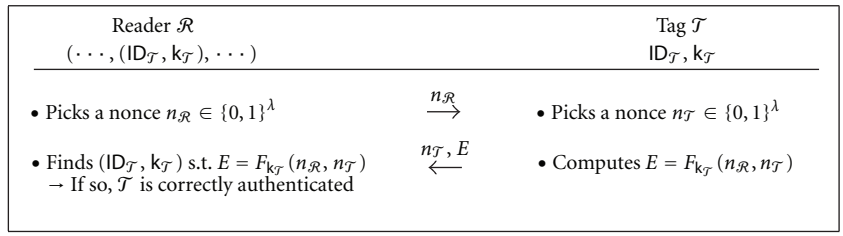


Figure 1: SK-based authentication protocol.

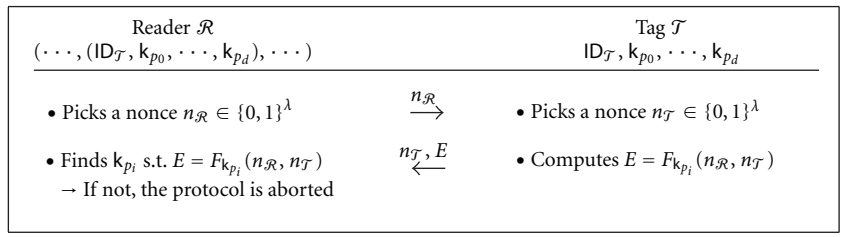


Figure 2: One round of the tree-based authentication protocol.

tag, which is not the case of *Sim*. This implies that the views of $\mathcal{A}$ and *Sim* will be distinguishable. Therefore, the protocol is not ZK-private, because at least one adversary can produce a distinguishable view (Figure 3).

*B.4. O-FRAP Authentication Protocol [18].* The Search procedure is detailed in Algorithm 1 where 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 $(\nu_4', \nu_1')$.

Avoine et al. describe in [28] 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 [18, 24, 39]. 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 execution, it is able to recognize $\mathcal{T}$ by recomputing $\nu_2$ as $\mathsf{k}_{\mathcal{T}}$ has not been updated and the nonces $(n_{\mathcal{R}}, n_{\mathcal{T}})$ have been sent in the clear. Note that the traceability attack of O-FRAP presented in [44] is specific to the way they define Algorithm 1 and does not apply here.

Therefore, no Corrupt query is allowed to an adversary 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, $(\rho, \sigma, \tau)$-private, WEAK-private, standard-untraceable, and WEAK-private.

In the Avoine and DMR models, the protocol is Existential-UNT-RTE and untraceable: the attack presented above without corruption does not work since the 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 $\mathcal{A}$ nonconsecutive 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 for LBM is only related to completed protocol executions, this attack can be perfectly simulated in the ideal world using the knowledge of $active(\mathcal{T})$ as proved in [18]. The protocol is thus forward-secure.

For DLYZ, the protocol is ZK-private: the proof is similar to the one of the SK-based protocol when no corruption is allowed. Regarding the forward-ZK-privacy, it is possible to define an adversary $\mathcal{A}$ that has a distinguishable view than the simulator's one. Let us 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 $\mathcal{T}_c$ and blocks the last message. $Sim_2$ has to provide a simulated incomplete interaction of $\mathcal{R}$ with $\mathcal{T}_c$: since $Sim_2$ does not have any information about $\mathcal{T}_c$, this interaction can only be composed of random messages. At the end, $\mathcal{T}_c$'s secrets are revealed to a distinguisher $\mathcal{D}$. Thus $\mathcal{D}$ is able to recognize if $\mathcal{A}_2$'s interaction corresponds to a real incomplete interaction with $\mathcal{T}_c$ or a simulated one. The protocol is therefore not forward-ZK-private (Figure 4).

*B.5. PK-Based Challenge/Response Authentication Protocol [22].* First, it is important to note that, under IND-CPA security, this protocol may not be easily proved private for WIDE adversaries 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 correctly simulate 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 proves that standard-untraceability can nevertheless be reached by PK-based protocols using IND-CPA cryptosystem but by adding other security mechanisms to the protocol (i.e., a MAC scheme).

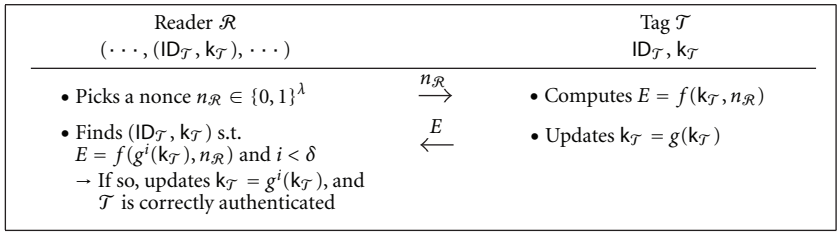


Figure 3: OSK-based authentication protocol.

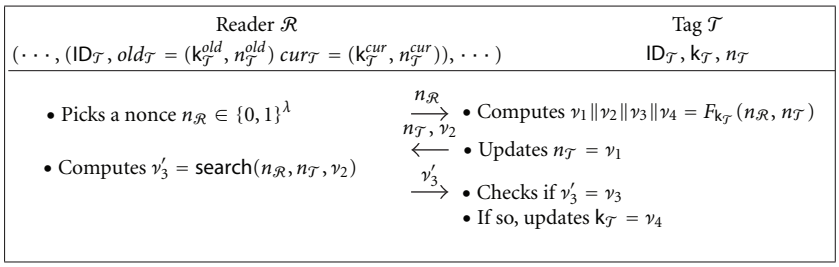


Figure 4: O-FRAP authentication protocol.

**Input**: $n_{\mathcal{R}}'; n_{\mathcal{T}}, \nu_2$
**Output**: $\nu_3$
(1) $Out \longleftarrow \perp$
(2) **if** $\exists\ (\mathrm{ID}_{\mathcal{T}}, old_{\mathcal{T}}', cur_{\mathcal{T}}')$ s.t. $n_{\mathcal{T}} = n_{\mathcal{T}}^{old}$ (resp. $n_{\mathcal{T}} = n_{\mathcal{T}}^{\mathrm{cur}}$) **then**
(3) $\quad \nu_1' \| \nu_2' \| \nu_3' \| \nu_4' \longleftarrow F_{\mathsf{k}_{\mathcal{T}}^{old}}(n_{\mathcal{R}}, n_{\mathcal{T}}^{old})$ (resp. $\nu_1' \| \nu_2' \| \nu_3' \| \nu_4' \longleftarrow F_{\mathsf{k}_{\mathcal{T}}^{cur}}(n_{\mathcal{R}}, n_{\mathcal{T}}^{cur})$)
(4) $\quad$ **if** $\nu_2' = \nu_2$ **then**
(5) $\quad\quad \mathcal{T}$ is correctly authenticated
(6) $\quad\quad Out \longleftarrow \nu_3'$
(7) $\quad\quad \mathsf{Update}\ (\mathcal{T})$
(8) $\quad$ **end if**
(9) **end if**
(10) **for all** $(\mathsf{ID}_{\mathcal{T}}, old_{\mathcal{T}}', cur_{\mathcal{T}}')$ **and** $c\varepsilon\{old_{\mathcal{T}}, cur_{\mathcal{T}}\}$**do**
(11) $\quad \nu_1' \| \nu_2' \| \nu_3' \| \nu_4' \longleftarrow F_{\mathsf{k}_{\mathcal{T}}^{c}}(n_{\mathcal{R}}, n_{\mathcal{T}})$
(12) $\quad$ **if** $\nu_2' = \nu_2$ **then**
(13) $\quad\quad \mathcal{T}$ is correctly authenticated
(14) $\quad\quad Out \longleftarrow \nu_3'$
(15) $\quad\quad \mathsf{Update}\ (\mathcal{T})$
(16) $\quad$ **end if**
(17) **end for**
(18) **return** $Out$

Algorithm 1: The search procedure.

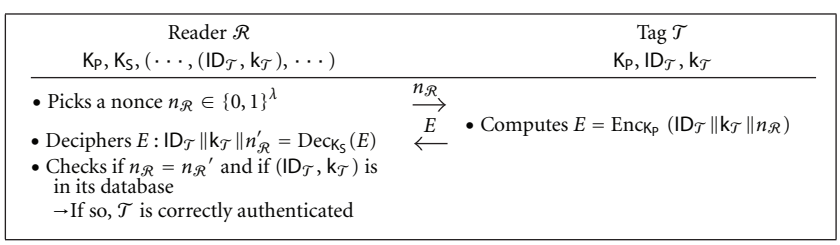


Figure 5: PK-based authentication protocol.

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

The proof is as follows in the Avoine model but can be easily adapted for the DMR model. We show that, if there exists an adversary $\mathcal{A}$ that wins $\mathrm{Exp}_{\mathcal{S},\mathcal{A}}^{P\text{-UNT}}$ (with $P \in \{\text{Existential}, \text{Forward}\}$), then it is possible to construct an adversary $\mathcal{A}'$ that 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 all oracles queries made by $\mathcal{A}$. At the end of the IND-CPA game, $\mathcal{A}'$ answers what $\mathcal{A}$ answers for $\mathrm{Exp}_{\mathcal{S},\mathcal{A}}^{P\text{-UNT}}$. 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 challenger $\mathcal{C}'$. $\mathcal{A}'$ 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 $\mathrm{Exp}_{\mathcal{S},\mathcal{A}}^{\text{Existential-UNT}}$, thus its advantage is nonnegligible, 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 it sends at random $\mathcal{T}_0$'s or $\mathcal{T}_1$'s secrets. If $\mathcal{A}'$ sends the expected ones, then $\mathcal{A}$ wins $\mathrm{Exp}_{\mathcal{S},\mathcal{A}}^{\text{Forward-UNT}}$, thus its advantage is nonnegligible, 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 nonnegligible, so is the advantage of $\mathcal{A}'$.

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

Vaudenay proves in [22] 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 (as explained in Section 13), the protocol is thus forward-$(\rho, \sigma, \tau)$-private for JW. HPVP proves in [15] that the protocol is also NARROW-STRONG-private with IND-CPA security but that it is STRONG-private with IND-CCA.

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

In the CCEG model, the protocol is future-untraceable with IND-CCA security (proved in [11]). In the DLYZ model, the protocol is also backward-ZK-private with IND-CCA security: the proof follows the same reasoning as the one of CCEG (Figure 5).

## C. The Vaudenay Model Implies the HPVP Model

The following theorem proves that, for a given adversary class, the privacy property of the Vaudenay model is at least stronger than the one of HPVP.

**Theorem 19**. *For any adversary class* $P \in \{\varnothing, \text{NARROW}\} \times \{\text{WEAK}, \text{FORWARD}, \text{DESTRUCTIVE}, \text{STRONG}\}$, *then the P-privacy property of the Vaudenay model implies the P-privacy property of the HPVP model.*

*Proof.* Both models define the same adversary classes but differ in their experiment. However, we show here that, for a given class $P$, Vaudenay's $P$-privacy implies HPVP's one. To do so, we exhibit an adversary in Vaudenay, denoted $\mathcal{A}_{\text{Vaud}}$, that emulates the system to an adversary playing the HPVP's $P$-experiment, denoted $\mathcal{A}_{\text{HPVP}}$, and uses the output of the latter to break Vaudenay's $P$-privacy.

First, $\mathcal{A}_{\text{Vaud}}$ can answer all the possible queries performed by $\mathcal{A}_{\text{HPVP}}$ during its experiment. The SENDTAG, SENDREADER, RESULT, CREATETAG, and LAUNCH queries can be easily emulated by $\mathcal{A}_{\text{Vaud}}$ due to their large similarity. For the DRAWTAG oracle, the Vaudenay model should be slightly modified in order to emulate the one of HPVP. Indeed in HPVP, this oracle formalizes the "left-or-right" paradigm. To handle this issue, we assume that, when $\mathcal{A}_{\text{Vaud}}$ gives as input of DRAWTAG a probability distribution with the form "$\Pr[\mathsf{ID}_i] = 1/2, \Pr[\mathsf{ID}_j] = 1/2$," then this also follows the "left-or-right" paradigm as well.

Also, $\mathcal{A}_{\text{HPVP}}$ can only corrupt free tags while only drawn tags can be corrupted in the Vaudenay model. Nevertheless, $\mathcal{A}_{\text{Vaud}}$ can correctly reply to these queries: upon a corruption query of the tag $\mathcal{T}$, $\mathcal{A}_{\text{Vaud}}$ draws $\mathcal{T}$ using a special distribution probability which attribute a probability of 1 to $\mathcal{T}$ and 0 for all the other tags. Then, it can corrupt it, transmits the data to $\mathcal{A}_{\text{HPVP}}$, and then frees $\mathcal{T}$. This method correctly works for DESTRUCTIVE and STRONG adversaries (and their NARROW variants). However, it must be adapted for a FORWARD adversary. Indeed, in both models, such an adversary can only perform corrupt queries after that the first one has been made, and $\mathcal{A}_{\text{Vaud}}$ must anticipate all these possible queries of $\mathcal{A}_{\text{HPVP}}$. Thus, upon the first corruption query, $\mathcal{A}_{\text{Vaud}}$ first frees all tags and then draws them one by one in order to know the correspondences between all the tags identifiers and their pseudonyms. Finally, $\mathcal{A}_{\text{Vaud}}$ is able to reply to all the corruption queries correctly.

This simulation is perfect and cannot be detected by $\mathcal{A}_{\text{HPVP}}$ that, as a consequence, will output its guessed bit $b'$ with its habitual probability. Then, using this bit, $\mathcal{A}_{\text{Vaud}}$ can decide which tag has been drawn by the DRAWTAG queries. Therefore, the success probability of $\mathcal{A}_{\text{Vaud}}$ is exactly the one

of $\mathcal{A}_{\mathsf{HPVP}}$. As Vaudenay's blinder cannot decide in advance which tag should be simulated after a DrawTag, the success probability of this blinded adversary is necessary one half (random guess of the bit).

Thus, if there exists an attack for a given system against the $P$-privacy in HPVP, then there exists an attack against the $P$-privacy that succeeds with the same probability in the Vaudenay model. Therefore, for any adversary class $P$, Vaudenay's $P$-privacy implies HPVP's one. □

The reciprocal is hard to prove for two main reasons. Firstly, Vaudenay's experiment output is not specified and may thus be unexploitable by $\mathcal{A}_{\mathsf{HPVP}}$. Secondly, the DrawTag oracle may receive as input an arbitrary distribution that can be hard to simulate using the "left-or-right" DrawTag of HPVP.

## Acknowledgment

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

## References

[1] EPCglobal. Class-1 Generation 2 UHF Air Interface Protocol Standard Version 1. 2. 0: Gen 2, 2008, http://www .epcglobal-inc.org/standards/.

[2] Infineon, Contactless SLE 66 Family, http://www.infineon .com/.

[3] NXP Semiconductors, DESFire Tags, http://www.nxp.com/.

[4] A. Cavoukian, Privacy-by-Design, http://privacybydesign.ca/.

[5] Viviane Reding. Commission recommendation of 12. 05. 2009—SEC(2009) 585/586, on the implementation of privacy and data protection principles in applications supported by radio-frequency identification, 2009.

[6] G. Avoine, E. Dysli, and P. Oechslin, "Reducing time complexity in RFID systems," in *Proceedings of the 12th International Conference on Selected Areas in Cryptography (SAC '05)*, vol. 3897 of *Lecture Notes in Computer Science*, pp. 291–306, Springer, Kingston, Canada, 2005.

[7] G. Avoine and P. Oechslin, "A scalable and provably secure hash-based RFID protocol," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW '05) Workshops*, pp. 110–114, IEEE, Kauai Island, Hawaii, USA, March 2005.

[8] D. Molnar and D. Wagner, "Privacy and security in library RFID issues, practices, and architectures," in *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS '04)*, pp. 210–219, ACM, Washington, DC, USA, October 2004.

[9] G. Avoine, "Adversary model for radio frequency identification," LASEC-REPORT 2005-001, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), Lausanne, Switzerland, 2005.

[10] M. Burmester, T. van Le, B. de Medeiros, and G. Tsudik, "Universally composable RFID identification and authentication protocols," *ACM Transactions on Information and System Security*, vol. 12, no. 4, article 21, 2009.

[11] S. Canard, I. Coisel, J. Etrog, and M. Girault, "Privacy-preserving RFID systems: model and constructions," Cryptology ePrint Archive, Report 2010/405, 2010.

[12] R. H. Deng, Y. Li, M. Yung, and Y. Zhao, "A new framework for RFID Privacy," in *Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS '10)*, vol. 6345 of *Lecture Notes in Computer Science*, pp. 1–18, Springer, Athens, Greece, 2010.

[13] T. van Deursen, S. Mauw, and S. Radomirović, "Untraceability of RFID protocols," in *Proceedings of the 2nd IFIP WG 11.2 International Conference on Information Security Theory and Practices: Smart Devices, Convergence and Next Generation Networks (WISTP '08)*, vol. 5019 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, Sevilla, Spain, May 2008.

[14] J.-H. Ha, S.-J. Moon, J. Zhou, and J.-C. Ha, "A new formal proof model for RFID location privacy," in *Proceedings of the 13th European Symposium on Research in Computer Security (ESORICS '08)*, vol. 5283 of *Lecture Notes in Computer Science*, pp. 267–281, Springer, Malaga, Spain, 2008.

[15] J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel, "A new RFID privacy model," in *Proceedings of the 16th European Symposium on Research in Computer Security (ESORICS '11)*, vol. 6879 of *Lecture Notes in Computer Science*, pp. 568–587, Springer, Leuven, Belgium, 2011.

[16] A. Juels and S. A. Weis, "Defining strong privacy for RFID," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom '07)*, pp. 342–347, IEEE, New York, NY, USA, March 2007.

[17] J. Lai, R. H. Deng, and Y. Li, "Revisiting unpredictability-based RFID privacy models," in *Proceedings of the 8th International Conference on Applied Cryptography and Network Security (ACNS '10)*, vol. 6123 of *Lecture Notes in Computer Science*, pp. 475–492, Springer, Beijing, China, 2010.

[18] T. van Le, M. Burmester, and B. de Medeiros, "Universally composable and forward-secure RFID authentication and authenticated key exchange," in *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS '07)*, pp. 242–252, ACM, Singapore, March 2007.

[19] C. Ma, Y. Li, R. H. Deng, and T. Li, "RFID privacy: relation between two notions, minimal condition, and efficient construction," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 54–65, ACM, Chicago, Ill, USA, November 2009.

[20] K. Ouafi, *Security and privacy in RFID systems [Ph.D. thesis]*, EPFL, Lausanne, Switzerland, 2011.

[21] R.-I. Paise and S. Vaudenay, "Mutual authentication in RFID: security and privacy," in *Proceedings of the 3rd ACM Symposium on Information, Computer and Communications Security (ASIACCS '08)*, pp. 292–299, ACM, Tokyo, Japan, March 2008.

[22] S. Vaudenay, "On privacy models for RFID," in *Proceedings of 13th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT '07)*, vol. 4833 of *Lecture Notes in Computer Science*, pp. 68–87, Springer, Kuching, Malaysia, December 2007.

[23] C. Su, Y. Li, Y. Zhao, R. H. Deng, Y. Zhao, and J. Zhou, "A survey on privacy frameworks for RFID authentication," *IEICE Transactions on Information and Systems*, vol. 95, no. 1, pp. 2–11, 2012.

[24] S. Canard and I. Coisel, "Data synchronization in privacy-preserving RFID authentication schemes," in *Proceedings of the 4th Workshop on RFID Security (RFIDSec '08)*, Budapest, Hungary, July 2008.

[25] S. Bocchetti, *Security and privacy in RFID protocols [M.S. thesis]*, Università degli Studi di Napoli Federico II, Naples, Italy, 2006.

[26] F. Armknecht, A. R. Sadeghi, A. Scafuro, I. Visconti, and C. Wachsmann, "Impossibility results for RFID privacy notions,"

*Transaction on Computational Science XI*, vol. 6480, pp. 39–63, 2010.

[27] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," in *Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '98)*, vol. 1462 of *Lecture Notes in Computer Science*, pp. 26–45, Springer, Santa Barbara, Calif, USA, 1998.

[28] G. Avoine, I. Coisel, and T. Martin, "Time measurement threatens privacy-friendly RFID authentication protocols," in *Proceedings of the 6th International Conference on Radio Frequency Identification: Security and Privacy Issues (RFIDSec '10)*, vol. 6370 of *Lecture Notes in Computer Science*, pp. 138–157, Springer, Istanbul, Turkey, 2010.

[29] P. D'Arco, A. Scafuro, and I. Visconti, "Revisiting DoS attacks and privacy in RFID-enabled networks," in *Proceedings of the 5th International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS '09)*, vol. 5804 of *Lecture Notes in Computer Science*, pp. 76–87, Springer, Rhodes, Greece, 2009.

[30] F. D. Garcia and P. van Rossum, "Modeling privacy for off-line RFID systems," in *Proceedings of the 9th Smart Card Research and Advanced Applications (CARDIS '10)*, vol. 6035 of *Lecture Notes in Computer Science*, pp. 194–208, Springer, Passau, Germany, 2010.

[31] R. Canetti, "Universally Composable Security: A New Paradigm for Cryptographic Protocols," Cryptology ePrint Archive, Report 2000/067, 2000.

[32] R. Canetti, "Security and Composition of Cryptographic Protocols: A Tutorial," Cryptology ePrint Archive, Report 2006/465, 2006.

[33] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[34] F. D. Garcia, I. Hasuo, W. Pieters, and P. van Rossum, "Provable anonymity," in *ACM Workshop on Formal Methods in Security Engineering (FMSE '05)*, pp. 63–72, ACM, Alexandria, VA, USA, November 2005.

[35] S. Mauw, J. H. S. Verschuren, and E. P. de Vink, "A formalization of anonymity and onion routing," in *Proceedings of the 9th European Symposium on Research in Computer Security (ESORICS '04)*, vol. 3193 of *Lecture Notes in Computer Science*, pp. 109–124, Springer, Sophia Antipolis, France, 2004.

[36] S. Canard, I. Coisel, and M. Girault, "Security of privacy-preserving RFID systems," in *Proceedings of IEEE International Conference on RFID-Technology and Applications (RFID-TA '10)*, pp. 269–274, IEEE, Guangzhou, China, June 2010.

[37] International Organization for Standardization, ISO/IEC, 9798: Information technology—Security techniques—Entity authentication, 1991–2010.

[38] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic approach to "privacy-friendly" tags," in *RFID Privacy Workshop*, MIT, Cambridge, Mass, USA, November 2003.

[39] T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," in *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm '05)*, pp. 59–66, IEEE, Athens, Greece, September 2005.

[40] T. van Deursen, *Security of RFID protocols [Ph.D. thesis]*, University of Luxembourg, Walferdange, Luxembourg, 2011.

[41] G. P. Hancke, "Practical eavesdropping and skimming attacks on high-frequency RFID tokens," *Journal of Computer Security*, vol. 19, no. 2, pp. 259–288, 2011.

[42] D. Moriyama, S. Matsuo, and M. Ohkubo, "Relation among the security models for RFID authentication protocol," in *ECRYPT Workshop on Lightweight Cryptography*, Louvain-la-Neuve, Belgium, 2011.

[43] G. Avoine, B. Martin, and T. Martin, "Tree-based RFID authentication protocols are definitively not privacy-friendly," in *Proceedings of the 6th International Conference on Radio Frequency Identification: Security and Privacy Issues (RFIDSec '10)*, vol. 6370 of *Lecture Notes in Computer Science*, pp. 103–122, Springer, Istanbul, Turkey, 2010.

[44] K. Ouafi and R. C. W. Phan, "Traceable privacy of recent provably-secure RFID protocols," in *Proceedings of the 6th International Conference on Applied Cryptography and Network Security (ACNS '08)*, vol. 5037 of *Lecture Notes in Computer Science*, pp. 479–489, Springer, New York City, NY, USA, June 2008.