Privacy-Preserving RFID Systems: Model and Constructions*

Sébastien Canard[1], Iwen Coisel[2], Jonathan Etrog[3], and Marc Girault[4]

[1] Orange Labs - 42 rue des Coutures - BP6234 - F-14066 Caen Cedex - France
[2] UCL - Place du Levant, 3 - B-1348 Louvain-la-Neuve - Belgium
[3] Orange Labs - 38-40 rue du Général Leclerc - F-92794 Issy les Moulineaux - France
[4] GREYC Université de Caen, Campus 2, Boulevard du Maréchal Juin, BP 5186, 14032 Caen Cedex, France

Abstract. In this paper, we study systems where a reader wants to authenticate and identify legitimate RFID tags. Such system needs thus to be correct (legitimate tags are accepted) and sound (fake tags are rejected). Moreover, an RFID tag in a privacy-preserving system should be anonymous and untraceable, except for the legitimate reader. We here present the first security model for RFID authentication/identification privacy-preserving systems which is at the same time complete and easy to use. Our correctness property permits to take into account active adversaries. Our soundness property incorporates the case of adversaries realizing relay attacks. Finally, our privacy model includes adversaries with no restrictions on their interactions with the system and moreover takes into account the case of "future correlations". We next propose several constructions, based on the work from Vaudenay, proving that (i) our strongest property is at least as strong as those of Vaudenay and (ii) this property is reachable by efficient schemes.

* This work has been partially financially supported by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II, by the French Agence Nationale de la Recherche under the RFID-AP project and by the Walloon Region Marshall plan through the 816922 Project SEE.

Table of Contents

1 Introduction

Due to the increasing number of interconnections between networks and communications between devices, it is commonly believed that preserving privacy is the main challenge that information society is faced to. Particularly concerning is the claimed irruption of RFID technology in our daily life. An RFID system is a vast bi-partite network made of readers, generally connected to back-end databases, and tags, small chips able to communicate through radio-frequency channels. Such tags are integrated into various objects, animals and even humans, in (premier) order to remotely identify them. Vast majority of presently used tags only provide an identity number (or Electronic Product Code if they comply with the EPC Global standard), and neither authentication nor any kind of privacy is achieved in such a case.

Nonetheless, many use cases for tags require authentication, identification and privacy properties. For instance, if the tag is embedded into a passport, it is highly desirable that the latter be authenticated and identified (*correctness* property) by immigration officials, and that cloned or counterfeited passports can be detected (*soundness* property). In addition, the holder may legitimately demand that authentications remain anonymous (and even untraceable) for other entities, so that indiscreet eavesdroppers cannot trace her. This leads to the notion of "privacy-preserving" authentication.

Several models for privacy preserving RFID authentication systems have already been proposed. Some of them focus on the correctness property [15, 11], some others on soundness [15, 32] but most of them on the privacy property [2, 21, 22, 32, 28, 11]. However, it seems that none of them are at the same time complete and easy to use. In fact, no soundness model takes into account "relay attacks" [3]. Moreover, none of existing privacy models permit the adversary against privacy to make future correlations (that is the target tags cannot have been previously corrupted by the adversary), except in the Vaudenay's model [32], which one is very elegant and complete. However, this latter model is very hard to handle and only few papers have use it, privileging the less complete Juels-Weis model [21], with the consequence that some schemes, considered as secure in the Juels-Weis model, may be in fact insecure. Moreover, the case of the strong adversary with no limitations is not completely handled in the case of future correlations, since it has been proven [32] to be impossible.

In this paper, based on our work in [13], we introduce the first security model for RFID authentication/identification privacy-preserving systems which is at the same time complete and easy to use. Our correctness property permits to take into account active adversaries, such as in [15, 11]. Our soundness property incorporates the case of adversaries using relay attacks. Finally, our privacy model includes adversaries with no restrictions on their interactions with the system and moreover takes into account the case of future correlations.

Next, we provide several constructions to prove that our strongest security definition is reachable and that our new untraceability property is at least as strong as the properties defined by Vaudenay in [32]. For this purpose, we focus on the generic construction from Vaudenay [32] based on the use of a public key cryptosystem. Based on our work in [12], we go further by giving the first concrete instantiation of the Vaudenay's result by using the IND-CCA secure cryptosystem DHAES. We next notice that the IND-CCA property is only reached by a few public key cryptosystems that can be embedded into an RFID tag and consequently, we argue that a weaker cryptosystem can also be used. More precisely, we introduce the "constant fixed non malleability". Next, we give a new generic construction based on the use of an IND-CPA secure public key cryptosystem (indistinguishability against chosen plaintext attack) together with a MAC scheme. We next give an example of a concrete implementation of this construction, using the so-called Hash El Gamal encryption scheme.

The paper is organized as follows. After the present introduction, we set up in Section 2 the model for authentication/identification RFID schemes. Then we define in Section 3 the correctness property. In Section 4, we introduce our new soundness property. Next, Section 5 studies related work on privacy models, while Section 6 describes our new characterization for the privacy property. Next, in Section 7, we recall the generic construction from Vaudenay, based on the use of an IND-CCA encryption scheme. We also prove in Section 8 that our future untraceability is at least as strong as the Vaudenay's narrow-strong privacy. In Section 9, we prove that our future untraceability can be reached by the Vaudenay's generic construction and we give the practical DHAES instantiation. Next, Section 10 introduced our new notion of constant fixed non malleability, give some examples, and prove that this property permits us to reach our untraeacbility property. In Section 11, we give our main scheme, based on the use of both the IND-CPA Hash El Gamal encryption scheme and a MAC scheme, and we finally make a global comparison, regarding security and efficiency, in Section 12.

2 Model for RFID Systems

In this section, we model an authentication/identification scheme for RFID systems. Then we describe all the possible interactions of an adversary with this system.

A *tag* $\mathcal{T}$ is a transponder, identified by a unique identifier ID, with limited memory and computational abilities, that can communicate with a *reader* $\mathcal{R}$ up to a limited distance. A reader is composed of (i) a transreceiver which communicates with possibly several tags and (ii) a back-end database containing all identifiers of valid tags and additional data such as secret keys. As for most of existing papers in the subject, we assume that communications between the transreceiver and the database are secure. In terms of security, the difference between a tag and a reader is that a tag cannot be considered as a tamper-resistant device (and thus an RFID tag can be corrupted by an adversary against the system) whereas the reader is more secure and more powerful.

2.1 Definition of the Procedures

A privacy-preserving RFID authentication scheme, denoted $\mathcal{S}$, is composed of the following procedures, where λ is a security parameter.

- $\textsc{Setup}(1^\lambda)$ is a probabilistic algorithm which on input λ outputs the parameters param of the system and generates a private/public key pair $(\mathsf{rsk}, \mathsf{rpk})$ for the reader. It also creates an empty database $\mathsf{DB}_\mathcal{R}$ which will later contain the identifiers and keys of all tags.
- $\textsc{TKeyGen}(1^\lambda, \mathsf{param}, \mathsf{ID}, \mathsf{rpk})$ is a probabilistic algorithm which returns a tag-dependent key set tk_ID. $(\mathsf{ID}, \mathsf{tk}_\mathsf{ID})$ is added in the reader's database $\mathsf{DB}_\mathcal{R}$.
- $\textsc{Ident}$ is an interactive protocol between the reader $\mathcal{R}$ taking on inputs 1^λ, param, rsk, rpk and $\mathsf{DB}_\mathcal{R}$, and a tag $\mathcal{T}$ with identifier ID taking on inputs 1^λ, param, tk_ID, rpk and eventually ID. At the end of the protocol, the reader either accepts the tag and outputs its identifier ID or rejects it and outputs $\perp$.

2.2 Definition of the Oracles

We have now to define the adversary $\mathcal{A}$ against such system. We consider that there is only one valid reader $\mathcal{R}$ in the system and we assume that this reader is sufficiently protected, such that the adversary is not able to corrupt it. However, as we will see below, the adversary may play the role of dishonest readers to interact with a tag and we assume that the latter does not know *a priori* if it is interacting with the valid reader $\mathcal{R}$ or the adversary $\mathcal{A}$.

At the beginning of each experiments (see Sections 3 to 6), we assume that the $\textsc{Setup}$ procedure has already been executed by the challenger denoted $\mathcal{C}$ and thus that the values 1^λ, param, rpk and rsk already exist. We next assume that $\mathcal{A}$ is always given 1^λ, param and rpk, while the secret rsk is never given to $\mathcal{A}$ (since the valid reader cannot be corrupted). At the beginning of one experiment, we consider that there is no tag in the system. We thus give to $\mathcal{A}$ the following oracle to introduce new ones.

- $\mathcal{O}^{\textsc{CreateTag}}()$: this oracle creates a legitimate tag with a unique identifier ID. This oracle uses $\textsc{TKeyGen}$ algorithm on input ID to set up the tag with tk_ID and updates $\mathsf{DB}_\mathcal{R}$ by adding this new one.

Next, as in the Vaudenay's model [32], we consider that the adversary can only interact with tags that are sufficiently close to her without having access to other existing ones. We thus use the concept of *free* and *drawn* tags introduced by Vaudenay. Drawn tags are the ones within "visual contact" to the adversary so that she can communicate while being able to link communications. Free tags are the other tags with which the adversary can not interact. At the creation of a new tag, that is after the call to $\mathcal{O}^{\text{CreateTag}}()$, the new tag has the status free and the adversary cannot interact with this tag. The adversary can modify these statuses by using the following oracles.

- $\mathcal{O}^{\text{Draw}}(k)$: this oracle 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 t_i and changes its status from free to drawn. Finally, the oracle outputs all generated pseudonyms $\mathsf{t}_1, \cdots, \mathsf{t}_k$ in any order. If there is not enough free tags (*i.e.* less than k), then the oracle outputs $\bot$. All relations $(\mathsf{t}_i, \mathsf{ID})$ are kept in a *a priori* secret table denoted Tab.
- $\mathcal{O}^{\text{Free}}(\mathsf{t})$: this oracle moves the tag with pseudonym t from the status drawn to the status free. This makes t unavailable from now on (in particular, $\mathcal{A}$ can not interact with tag t anymore).

Next, the adversary is only able to interact with tags by using the pseudonyms, and only if the tag has the status drawn. To simplify notation, we denote by tk_t the secret key of the tag with pseudonym t, which is equal to the secret key tk_ID of the underlying identifier ID of this tag. Using a pseudonym, the adversary has now several ways to interact with tags.

First, $\mathcal{A}$ is able to corrupt tags by using the following oracle.

- $\mathcal{O}^{\text{Corrupt}}(\mathsf{t})$: returns the tag-dependent key tk_ID of the related tag ID. The pseudonym t is now marked as "corrupted"[5].

Next, the adversary can *passively* witness in the whole protocol Ident between a legitimate tag and the valid reader $\mathcal{R}$ by using the following oracle.

- $\mathcal{O}^{\text{Execute}}(\mathsf{t}, \mathsf{step})$: executes an Ident protocol between the reader and the tag with pseudonym t. The value step permits the adversary to stop the execution at the step step. This oracle outputs the transcript of the protocol (partial or complete). Moreover, if $\mathsf{step} = final$ (the protocol is completely executed), it outputs 0 if the output of the reader during the Ident protocol is $\bot$ and 1 otherwise (but not the identifier of the legitimate tag, for privacy reasons).

$\mathcal{A}$ can also *actively* participate in the Ident protocol by playing the role of either a fake/corrupted tag, or an invalid reader. For this purpose, we introduce the following oracles which also permit $\mathcal{A}$ to stop at any step a "standard" identification protocol, delete or modify some messages.

- $\mathcal{O}^{\text{Launch}}()$: makes the legitimate reader $\mathcal{R}$ launch a new Ident protocol instance, that is the first request to an unknown tag so as to authenticate and identify it. It outputs the sent message r from the reader to the tag and the identifier π for this protocol instance.
- $\mathcal{O}^{\text{SendReader}}(m, \pi)$: sends a message m to the reader in the protocol π. It outputs the response r from the reader.
- $\mathcal{O}^{\text{SendTag}}(m, \mathsf{t})$: sends a message m to the tag with pseudonym t. It outputs the response r from the tag.
- $\mathcal{O}^{\text{Return}}(\pi)$: outputs 0 if the output of the reader during the Ident protocol is $\bot$ and 1 otherwise.

2.3 Definition of the Adversary

We now define the different classes of adversaries who will play security experiments (see Sections 5 and 6). We here adapt the classification introduced by Vaudenay in [32].

Definition 1 (Adversary Class). *An adversary $\mathcal{A}$ against the RFID system is said to be*

- passive *if $\mathcal{A}$ has access to no oracles;*

[5] Note that the underlying tag with identifier ID is also corrupted. However, a new pseudonym of this tag can also be corrupted. Thus, a pseudonym t can only be corrupted once while a tag ID may be corrupted several times.

- weak *if $\mathcal{A}$ has no access to the $\mathcal{O}^{\textsc{Corrupt}}$ oracle;*
- forward *if $\mathcal{A}$ is committed to only use the $\mathcal{O}^{\textsc{Corrupt}}$ oracle after her first call to the $\mathcal{O}^{\textsc{Corrupt}}$ oracle;*
- destructive *if $\mathcal{A}$ can not use anymore a corrupted tag;*
- strong *if $\mathcal{A}$ has no limit on the oracles.*

$\mathcal{A}$ is moreover said narrow *if she has no access to $\mathcal{O}^{\textsc{Return}}$.*

In the following, we denote by $\mathcal{A}_P$ the adversary with power $P \in \mathcal{P} \subset \big(\{\emptyset, \text{narrow}\} \cup \{\text{passive, weak, forward, destructive, strong}\}\big)$. We now give our security definitions, regarding correctness, soundness and privacy/untraceability.

3 The Correctness Property

As we consider (RFID) authentication schemes, we first need to ensure that a legitimate RFID tag is (almost) always accepted by the reader. In some cases (as those detailed in [22, 11]), it is necessary to introduce a strong correctness property, where the adversary is able to interact with uncorrupted tags so as to prevent them from being accepted. This is in fact useful for *e.g.* schemes [33, 25, 4, 17, 22, 11] which modify the internal shared secret key after each successful authentication.

In the following, we define both standard (called *passive* below) and strong correctness properties by using the following correctness experiment, where $P \in \{\text{passive}, \text{strong}\}$.

Experiment $Exp^{\mathsf{correct}}_{\mathcal{S},\mathcal{A}_P}$:

1. The challenger $\mathcal{C}$ initializes the system and sends 1^λ, param and rpk to $\mathcal{A}_P$.
2. $\mathcal{A}_P$ interacts with the system through the oracles.
3. $\mathcal{A}$ chooses an uncorrupted tag ID.
4. $\mathcal{A}$ launches a request $\mathcal{O}^{\mathsf{Execute}}(\mathsf{ID})$. The experiment returns the bit b outputted by this oracle.

For a scheme $\mathcal{S}$, we define the success of an adversary $\mathcal{A}_P$ for the correctness experiment as follows:

$$Succ^{\mathsf{correct}}_{\mathcal{S},\mathcal{A}_P}(1^\lambda) = \Pr\left[Exp^{\mathsf{correct}}_{\mathcal{S},\mathcal{A}_P} = 0\right].$$

Definition 2 (Correctness). *An RFID authentication scheme $\mathcal{S}$ is* P-correct *if for any adversary $\mathcal{A}_P$ running in polynomial time, $Succ^{\mathsf{correct}}_{\mathcal{S},\mathcal{A}_P}(1^\lambda)$ is negligible.*

Remark 1. The Strong-correctness is sometimes called the availability property [22].

4 The Soundness Property

The soundness property below formalizes the fact that a fake tag cannot be accepted by the system. We here define two different levels for this property, depending on the possibility for the adversary to interact (full soundness) or not (soundness) with the tag she tries to impersonate during the Ident protocol itself. To formalize the soundness property, we use the following experiment where the adversary $\mathcal{A}$ is *strong*.

Experiment $Exp^{\mathsf{sound}}_{\mathcal{S},\mathcal{A}}$:

1. The challenger $\mathcal{C}$ initializes the system and sends 1^λ, param and rpk to $\mathcal{A}$.
2. $\mathcal{A}$ interacts with the whole system through oracles.
3. At any time of the experiment, the adversary outputs a protocol instance identifier π outputted by the $\mathcal{O}^{\textsc{Launch}}$ oracle.
4. $\mathcal{A}$ can again interact with the whole system.
5. The experiment finally returns the bit b outputted by $\mathcal{O}^{\textsc{Return}}(\pi)$. We denote ID the identifier of the tag involved in the π protocol.

For a scheme $\mathcal{S}$, we define the success of an adversary $\mathcal{A}$ for the soundness experiment as follows:

$$Succ^{\mathsf{sound}}_{\mathcal{S},\mathcal{A}}(1^\lambda) = \Pr\left[Exp^{\mathsf{sound}}_{\mathcal{S},\mathcal{A}} = 1\right].$$

In the following, we say that the soundness experiment $Exp^{\mathsf{sound}}_{\mathcal{S},\mathcal{A}}$ is *not relevant* if at least one pseudonym t of the tag ID has been corrupted during $Exp^{\mathsf{sound}}_{\mathcal{S},\mathcal{A}}$. Using this characterization, we can define the soundness.

Definition 3 (Soundness). *An RFID authentication scheme $\mathcal{S}$ is* sound *if for any adversary $\mathcal{A}$ running in polynomial time, $Exp^{sound}_{\mathcal{S},\mathcal{A}}$ is relevant, $Succ^{sound}_{\mathcal{S},\mathcal{A}}(1^\lambda)$ is negligible and $\mathcal{A}$ did not interact with the tag ID after Step 3 of $Exp^{sound}_{\mathcal{S},\mathcal{A}}$.*

Definition 4 (Full Soundness). *n RFID authentication scheme $\mathcal{S}$ is* full-sound *if for any adversary $\mathcal{A}$ running in polynomial time, $Exp^{sound}_{\mathcal{S},\mathcal{A}}$ is relevant and $Succ^{sound}_{\mathcal{S},\mathcal{A}}(1^\lambda)$ is negligible.*

Remark 2. The *full-soundness* permits to model "relay attacks" where $\mathcal{A}$ just transmits messages from the reader to the tag and vice versa. For example, such attack permits an adversary $\mathcal{A}$ to open a car with the RFID car keys of the real owner of the car, without the permission of the latter. For this purpose, $\mathcal{A}$ is in the neighborhood of a RFID car keys and her accomplice is near the car. The relay attack permits to open the car even if the key is far away. Although these attacks seem very hard to stop, they can be avoided by using distance bounding techniques as described in many papers (*e.g.* [3, 24, 23, 20]).

5 Related Work on the Privacy Property

As stated in the introduction, privacy of RFID tags should be protected in authentication/identification systems. More precisely, a tag should be at least anonymous and untraceable for everyone except the valid reader. Moreover, the scheme has to preserve the anonymity and the untraceability of tags even if an adversary obtains its internal data: this is what is called forward privacy (a.k.a. forward untraceability).

5.1 Existing Privacy Models

Several attempts have been done to model the privacy of tags. Le *et al.* adopt in [22] a specific approach to the formalization of protocol security based on the Universal Composability (UC) framework. Some other proposals are based on a different concept, introduced by Avoine [2] in the RFID setting, where privacy is formalized by the ability for the adversary to distinguish two known tags. This model was refined by Juels and Weis [21] and later in [28, 11]. However, none of these models permit the adversary to make future correlations (that is the target tags cannot have been corrupted by the adversary). This case is taken into account in Vaudenay's model [32], which one is very elegant and complete, but quite hard to use. In the rest of this section, we give some words on this model in particular.

5.2 The Vaudenay's Model for Privacy

Vaudenay has introduced a new privacy model in [32]. Informally, a scheme ensures the privacy property, in the sense of Vaudenay, if for a given experiment (see below), the success probability of an adversary, which interacts with the system through oracles (as defined in section 2.2), is indistinguishable of a "blinded" adversary, which interacts with a simulated system, controlled by a simulator who does not know anything about secret values. More formally, Vaudenay defines the following experiment where P belongs to $\{\emptyset, \text{narrow}\} \cup \{\text{strong}, \text{destructive}, \text{forward}, \text{weak}\}$:

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

1. The challenger $\mathcal{C}$ initializes the system and sends 1^λ, param and rpk to $\mathcal{A}$.
2. $\mathcal{A}_P$ interacts with the whole system through the oracles, limited by her class P.
3. $\mathcal{A}_P$ submits a hypothesis about the system and receives the hidden table Tab of the $\mathcal{O}^{\text{Draw}}$ oracle.
4. $\mathcal{A}_P$ returns 1 if her hypothesis is correct and 0 otherwise.

The adversary wins if she returns 1.

Definition 5 (Trivial Adversary). *An adversary $\mathcal{A}$ is said* trivial *if it is possible to define a simulator Sim who perfectly simulates the system, without knowing any secrets, for a "blinded" adversary denoted $\mathcal{A}^{\mathsf{Sim}}$, such that $|Pr[\mathcal{A}\ wins] - Pr[\mathcal{A}^{\mathsf{Sim}}\ wins]|$ is negligible.*

If those success probabilities are indistinguishable, it means that there is no privacy loss through the communication channel. In other words, the adversary makes no effective use of the messages as their simulation (without using the secret values) leads to the same probability of success. Thus the scheme can be considered private.

Definition 6 (Privacy). *For $P \in \{\emptyset, narrow\} \cup \{strong, destructive, forward, weak\}$, a scheme is said* P-private *if all P-adversaries are trivial.*

Vaudenay proves moreover in [32] that a protocol cannot ensure at the same time *destructive-privacy* and *narrow-strong-privacy.* Thus, as a result, *strong-privacy* cannot be reached in the Vaudenay's model.

As a consequence, the best privacy property in this model implies that the adversaries' abilities have to be restrained: either the adversary is destructive or the adversary is narrow. Thus, this is not possible with such model to study its security against a strong adversary.

Relations between privacy properties [32]

The following implications between privacy properties are obvious.

$$\begin{tikzcd}
 & \text{Destructive} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{Forward} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \text{Weak} \arrow[d, Rightarrow] \\
\text{Narrow-Strong} \arrow[r, Rightarrow] & \text{Narrow-Destructive} \arrow[r, Rightarrow] & \text{Narrow-Forward} \arrow[r, Rightarrow] & \text{Narrow-Weak}
\end{tikzcd}$$

5.3 Some Recent Related Models

As it is impossible to reach the strongest property of Vaudenay's model, many adaptations of this model have been done in order to define a weakest, but reachable, strong property. For example, Ng et al. have introduced in [36] the notion of *wise* adversary. These adversaries are restrained (compared to those of Vaudenay) such that they are not able to access twice the same oracle with the same input, and they are also not able to access oracles where the results can be precisely predicted. Thus, they prove that it is indeed possible that a scheme ensures the strong privacy property against wise adversary. Furthermore, they prove equivalence between the eight privacy properties of Vaudenay and thus reduce them to three different properties as follows:

Note that the authors of [36] use the same denominations for their privacy properties as those defined by Vaudenay. This is a misuse of language as the modification of the adversary necessarily (unless if the equivalence is proven) modifies the relied property.

Deng et al. have introduced in [16] the notion of *zero-knowledge privacy.* There exists lots of similarity between their model and Vaudenay's model. Although they define a new experiment, the adversary's abilities are again restrained compared to those of Vaudenay's adversary. The related privacy property is again not shown against the strongest possible adversary.

In the following section, we define a new privacy model where, instead of restrain the adversaries' abilities, we reduce the success conditions of this adversary. As this will be detailed in Section 8, this permits to define a stronger privacy property than those (achievable) of Vaudenay.

6 Our New Privacy Property

The restriction we introduce is that an adversary will win the privacy experiment if and only if she is able to make the link between several authentications of a same tag. In other words, we consider that a scheme is private if tags are untraceable. Before defining more precisely our untraceability experiment, we first introduce the notion of *non-obvious link.*

6.1 Non-Obvious Link

A *link* is a couple of pseudonyms associated to the same identifier in Tab. As some links are obvious (*e.g.* if both pseudonyms have been corrupted), we define below the notion of *non-obvious link*, illustrated by Figure 1. Links are chronologically ordered, *i.e.* $(\mathsf{t}_i, \mathsf{t}_j)$ means that t_i has been freed before that t_j has been drawn. Informally, a *non-obvious link* (*n.o.l.*) is a link between two pseudonyms which cannot be defined without using some hidden (or not) information in the sent messages.

Definition 7 (Non-Obvious Link). *(t_i, t_j) is a* non-obvious link *if t_i and t_j refer to the same ID in Tab and if a "dummy" adversary, who only have access to $\mathcal{O}^{\text{CreateTag}}, \mathcal{O}^{\text{Draw}}, \mathcal{O}^{\text{Free}}, \mathcal{O}^{\text{Corrupt}}$, is not able to output this link with a probability better than $1/2$. Moreover, a non-obvious link is said:*

- standard *if $\mathcal{A}$ has not corrupted t_i or t_j (see link 1 in Figure 1);*
- past *if $\mathcal{A}$ has corrupted t_j (see link 2 in Figure 1);*
- future *if $\mathcal{A}$ has corrupted t_i (see link 3 in Figure 1).*

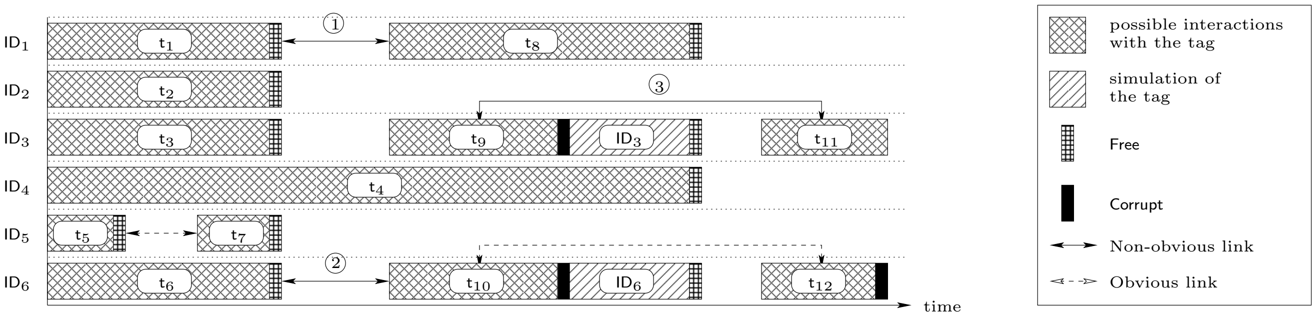

Fig. 1. Privacy Model

Note that this "dummy" adversary is equivalent to the blinded adversary of Vaudenay. Although this latter has access to the remaining oracles ($\mathcal{O}^{\text{Execute}}, \mathcal{O}^{\text{Launch}}, \mathcal{O}^{\text{SendReader}}, \mathcal{O}^{\text{SendTag}}, \mathcal{O}^{\text{Return}}$), all the answers of these oracles are simulated by a "Blinder" which don't know any secret values of the system. Consequently, the blinded adversary has no advantage compared to our dummy adversary.

It is straightforward that a weak adversary is only able to output standard non-obvious link as she cannot request the $\mathcal{O}^{\text{Corrupt}}$. A forward or a destructive adversary is not able to output a future link as the corruption of a tag either stops the system or destroys the tag (and thus prevent it to be drawn again). However, both of these adversaries can output standard and past link. Finally, a strong adversary is able to output each possible link. As a conclusion, we highlight the fact that the goal of the forward and the destructive adversaries are the same, thus in the following, we will never consider a forward adversary (as this one is weaker than the destructive one).

6.2 Description of Our Untraceability Experiment

It is obvious that the untraceability of a scheme is equivalent to the impossibility for an adversary to find some non-obvious links. Our new untraceability experiment is defined as follows, where the adversary class P belongs to $\{\text{strong}, \text{destructive}, \text{weak}\}$.

Experiment $Exp^{\mathsf{Unt}}_{\mathcal{S},\mathcal{A}}$

1. The challenger $\mathcal{C}$ initializes the system and sends 1^λ, param and rpk to $\mathcal{A}$.
2. $\mathcal{A}$ interacts with the whole system using the above oracles, limited by her class P.
3. The adversary $\mathcal{A}_P$ returns one link $(\mathsf{t}_i, \mathsf{t}_j)$.

For a scheme $\mathcal{S}$, we define the success of an adversary $\mathcal{A}$ for the untraceability experiment as follows:

$$Succ^{\mathsf{Unt}}_{\mathcal{S},\mathcal{A}}(1^\lambda) = \Pr\left[(\mathsf{t}_i, \mathsf{t}_j) \text{ is a non-obvious link}\right].$$

Thus, considering the three classes of adversary, we can define the following three levels of untraceability.

Definition 8 (Untraceability). *An RFID authentication scheme $\mathcal{S}$ is* untraceable *(resp.* past-untraceable */* future-untraceable*) if for any weak (resp. destructive / strong) adversary $\mathcal{A}$ running in polynomial time, it is possible to define a "dummy" adversary $\mathcal{A}_d$, who only have access to oracles $\mathcal{O}^{\text{CreateTag}}$, $\mathcal{O}^{\text{Draw}}$, $\mathcal{O}^{\text{Free}}$ and $\mathcal{O}^{\text{Corrupt}}$ such that:*

$$\left|Succ^{\mathsf{Unt}}_{\mathcal{S},\mathcal{A}}(1^\lambda) - Succ^{\mathsf{Unt}}_{\mathcal{S},\mathcal{A}_d}(1^\lambda)\right| \leq \epsilon(\lambda)$$

where $\epsilon(\lambda)$ is negligible.

In other words, a scheme is considered untraceable, if an adversary cannot output a non-obvious link with a probability better than 1/2 (the maximal success probability of the dummy adversary).
It is relatively straightforward that:

Relations between untraceability properties (this paper)

The following implications between untraceability properties are obvious.

Future-Untraceability ⇒ Past-Untraceability ⇒ Untraceability.

In the next sections, we study our new model and prove that our strongest untraceability property (*i.e.* future-untraceability) is achievable. We also describe an attack that is neither taken into account in the destructive-privacy nor the narrow-strong-privacy properties of Vaudenay's model. With those two arguments, it is obvious to understand why our model improves Vaudenay's one. Before that, we need to introduce a generic construction due to Vaudenay [32].

7 From Encryption to Privacy

We first recall the notion of public key cryptosystems and what does IND-CCA and IND-CPA say. We next give a generic RFID identification and authentication scheme, due to Vaudenay [32], based on the use of such public key cryptosystem.

7.1 Public Key Cryptosystem

Let a public-key encryption scheme $\mathcal{E}$ = (KeyGen, Enc, Dec) such that:

- KeyGen is a probabilistic key generation algorithm which on input the security parameter 1^λ outputs the encryption public key rpk and the corresponding decryption secret key rsk,
- Enc is a probabilistic encryption algorithm which on input a message m and the public key rpk outputs the corresponding ciphertext c,
- Dec is a deterministic decryption algorithm which on input a ciphertext c and the decryption secret key rsk outputs a plaintext m.

The correctness of the scheme is defined as $\text{Dec}(\text{Enc}(m, \mathsf{rpk}), \mathsf{rsk}) = m$.

Classes of Adversary. We then consider three different attacks for the adversary.

- Under *chosen-plaintext attack* (CPA), the adversary can obtain ciphertexts of plaintexts of her choice, using the public key.

- Under *non-adaptive chosen-cipher attack* (CCA1), the adversary gets, in addition to the public key, access to an oracle for the decryption function. The adversary may use this decryption function only for a period of time before receiving the challenge ciphertext c.
- Under *adaptive chosen-cipher attack* (CCA2) the adversary again gets, in addition to the public key, access to an oracle for the decryption function, but this time she may use this decryption function even on ciphertexts chosen after obtaining the challenge ciphertext c, the only restriction being that the adversary may not ask for the decryption of c itself.

One-wayness. Regarding security, the weakest property one encryption scheme should verify is the One-Wayness (OW), which says that this is infeasible to retrieve the plaintext from the ciphertext.

indistinguishability. Moreover, an encryption scheme should also be secure in the sense that it should not be possible for an adversary to learn any information about the plaintext m underlying a challenge ciphertext c. Such scheme is said to have the indistinguishability (IND) property.

Note that the notion of IND-CCA usually refers to the IND-CCA2 property while the IND-CCA1 is rarely used in practice. We utilize this notation in the following.

Non-malleability. The Non-Malleability (NM) property formalizes an adversary's inability, given a challenge ciphertext y, to output a different ciphertext y' such that the plaintexts x, x' underlying these two ciphertexts are "meaningfully related" (for example, $x' = x + 1$).

7.2 Some Examples

In this paper, we will use several different encryption schemes. The Rabin encryptions scheme [29] is OW-CPA. We will also use the El Gamal encryption scheme [19] (which is IND-CPA) and the hash variant [14], named Hash El Gamal in the following, which is also IND-CPA. Finally, the DHAES [1] has been proved to be IND-CCA.

7.3 Generic Construction from Vaudenay

Using a public key cryptosystem $\mathcal{E}$ such as defined above, Vaudenay introduces the following RFID identification scheme. In this scheme and in all the following ones in this paper, the reader key pair $(\mathsf{rsk}, \mathsf{rpk})$ corresponds to the public key cryptosystem key pair, that is, rsk is a secret decryption key and rpk is the corresponding encryption public key. Moreover, let $\mathsf{tk}_{\mathsf{ID}}$ be the λ-bit key of a tag, which is known by both the tag and the reader. The identification protocol is next described in Figure 2.

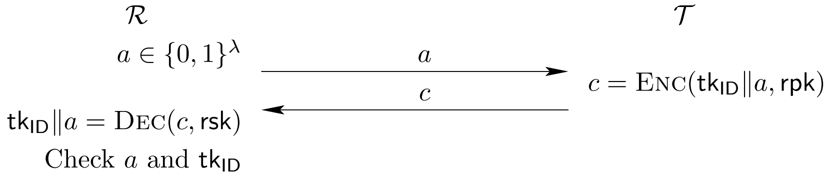
$\mathcal{R}$
$\mathcal{T}$
$a \in \{0,1\}^\lambda$
a
$c = \textsc{Enc}(\mathsf{tk}_{\mathsf{ID}} \| a, \mathsf{rpk})$
c
$\mathsf{tk}_{\mathsf{ID}} \| a = \textsc{Dec}(c, \mathsf{rsk})$
Check a and $\mathsf{tk}_{\mathsf{ID}}$

Fig. 2. Vaudenay's protocol

Remark 3. In [32], the scheme is a little bit different. More precisely, the reader has an additional secret key rk such that for all tag $(\mathsf{ID}, \mathsf{tk}_{\mathsf{ID}})$, the relation $\textsc{Prf}(\mathsf{ID}, \mathsf{rk}) = \mathsf{tk}_{\mathsf{ID}}$ holds, where $\textsc{Prf}$ is a pseudo-random function. The tag encrypts is identifier ID concatenated to $\mathsf{tk}_{\mathsf{ID}}||a$. Then, after the decryption of c, the reader can check the equality $\textsc{Prf}(\mathsf{ID}, \mathsf{rk}) = \mathsf{tk}_{\mathsf{ID}}$ and so identifies the tag without requesting the database.

In [32], Vaudenay proves that if the cryptosystem is IND-CPA, then the identification scheme is narrow-strong private and, if the cryptosystem is IND-CCA2, the scheme is further secure and forward private. We do not recall the security proof in this paper. We will however prove that this construction provides untraceability, regarding our model, in the next section.

Remark 4. Note that, as a consequence of the IND-CPA property, the encryption scheme needs to be probabilistic. This implies some randomness coming from the RFID tag, which is quite natural in order to achieve the forward privacy.

8 Future Untraceability $\geq$ Narrow-Strong Privacy

In this section, we define an attack which is only possible for a strong adversary. We first remember that a strong adversary can corrupt tags without destructing them. This means that after a corruption, a tag can be freed and affected again. Our strong adversary is also authorized to request the $\mathcal{O}^{\text{Result}}$ oracle, which is never possible in the Vaudenay's model, since strong-privacy is not achievable. Next, our "strong" attack uses these two characteristics of a strong adversary since the essence of this attack is to use the $\mathcal{O}^{\text{Result}}$ oracle in order to produce a future non-obvious link. Thus, a narrow-strong adversary is not able to make such attack, and thus cannot break the narrow-strong privacy, in the Vaudenay's sense. In fact, if the adversary makes no effect use of $\mathcal{O}^{\text{Result}}$, the narrow-strong adversary in Vaudenay's model is able to produce the same attack.

8.1 A Toy Scheme

In the following, we describe a protocol which does not resist to the above attack. This scheme is an instantiation of Vaudenay's generic protocol (see Figure 2) using the Hash ElGamal encryption scheme [6] (which is not IND-CCA secure). In this scheme, each tags knows the public key rpk of the encryption scheme and a couple of secrets $\mathsf{tk_{ID}} = (\mathsf{k_{ID}}, \mathsf{sk_{ID}})$, where $\mathsf{k_{ID}}$ is a secret key of a tag for the f function and $\mathsf{sk_{ID}}$ is a secret which is known by both the tag and the reader. The reader knows the secret decryption key $\mathsf{rsk} \in \mathbb{Z}_q^*$ associated to rpk such that $\mathsf{rpk} = g^{\mathsf{rsk}}$, where g is a generator of a cyclic group $\mathbb{G}$ of prime order q. Moreover, f is an unforgeable function which takes as input a secret value and a message, and $\mathcal{H}$ is a cryptographically secure hash function. The resulting identification scheme is depicted in Figure 3.

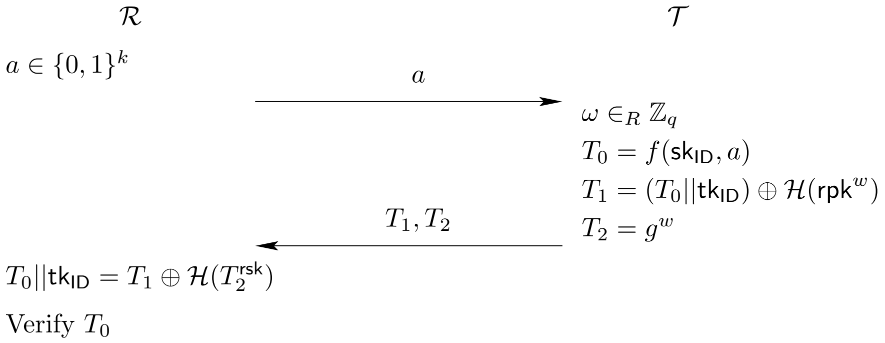
$\mathcal{R}$
$\mathcal{T}$
$a \in \{0,1\}^k$
a
$\omega \in_R \mathbb{Z}_q$
$T_0 = f(\mathsf{sk_{ID}}, a)$
$T_1 = (T_0 || \mathsf{tk_{ID}}) \oplus \mathcal{H}(\mathsf{rpk}^w)$
T_1, T_2
$T_2 = g^w$
$T_0 || \mathsf{tk_{ID}} = T_1 \oplus \mathcal{H}(T_2^{\mathsf{rsk}})$
Verify T_0

Fig. 3. Hash ElGamal based protocol

8.2 Security Considerations

We now give several lemmas to prove, using this toy scheme, that our future un traceability is at least as strong as the Vaudenay's narrow-strong privacy property.

Lemma 1. *The above toy scheme is not future untraceable.*

Proof. To break the privacy of this scheme, an adversary $\mathcal{A}$ has first to create two tags and corrupt one of them, denoted $\tilde{\mathsf{t}}$, in order to obtain the corresponding $\mathsf{tk_{ID}} = (\mathsf{k_{ID}}, \mathsf{sk_{ID}})$. Then, $\mathcal{A}$ frees this tag and then draws the two tags t_0 and t_1. To identify the tag $\tilde{\mathsf{t}}$, $\mathcal{A}$ initiates a protocol execution with the reader and received a challenge a. Then, $\mathcal{A}$ chooses one of the two affected tag, *e.g.* t_0, and sends him a nonce a'. $\mathcal{A}$ receives (T_1, T_2), computes $T_1' = T_1 \oplus f(a, \mathsf{k_{ID}})||0 \ldots 0 \oplus f(a', \mathsf{k_{ID}})||0 \ldots 0$ and sends (T_1', T_2) to the reader. If t_0 and $\tilde{\mathsf{t}}$ correspond to the same tag, the reader will accept $\mathcal{A}$ as $T_1' = f(a', \mathsf{k_{ID}})||\mathsf{sk_{ID}} \oplus \mathcal{H}(\mathsf{rpk}^w)$ which is a valid answer. As a conclusion, the adversary is able to recognize $\tilde{\mathsf{t}}$, and so to produce a future link, which concludes the proof. □

Lemma 2. *The above toy scheme is narrow strong.*

Proof (sketch). In fact, a narrow-strong adversary cannot perform this attack as the $\mathcal{O}^{\text{Result}}$ request is indispensable to distinguish the tag. □

Remark 5. As described here, the scheme is neither past-untraceable as this attack can be adapted to produce a past link. However, and contrary to the "future-link attack", it can be avoided by using a key update mechanism as *e.g.* in [25, 17, 22, 11]. We do not detail this here as we only want to sketch an attack which can not be realized in Vaudenay's model.

8.3 Vaudenay's Privacy vs. Our Untraceability

As a conclusion of this section, neither the destructive nor the narrow-strong privacy implies the future-untraceability.

Relations between privacy and untraceability properties ([32] and this paper)

We now have the following implications:

$$\begin{array}{ccc} \text{Future-Unt} & \not\Leftarrow & \text{Destructive} \\ \not\Uparrow & & \Downarrow \\ \text{N-Strong} & \Rightarrow & \text{N-Destructive} \end{array}$$

Consequently, our untraceability property is at least as strong as those of Vaudenay.

Remark 6. It seems possible to prove that Future-Untraceability implies Narrow-Strong privacy. However, this result is hard to obtain as it requires proving that all the narrow-adversaries taken in account in Vaudenay's model are useless, except the one who is able to produce a non-obvious link and it is impossible to model all of them.

9 IND-CCA ⇒ Future Untraceability

In this section, we prove that our strongest privacy property (future-untraceability) is achievable. For this purpose, we use the above generic RFID authentication protocol from Vaudenay, with an IND-CCA secure encryption scheme (*e.g.* the DHAES scheme [1]). It is quite obvious that the security of this scheme relies to the security of the underlying encryption scheme. However, the security goal of these two schemes is not the same. Thus, we have to prove that the security of the encryption scheme implies the security of the authentication scheme.

Vaudenay has proved in [32] that the scheme presented in Figure 2 ensures narrow-strong privacy and forward privacy if the encryption scheme had the IND-CCA2 property. In our model, we claim that it ensures the future untraceability. In order to prove this, we use the game technique proof, introduced by Shoup [31].

9.1 Instance Description

As our model is made up of a large number of oracles (which have to be formally described in the proof), we first introduce a table, denoted $\mathcal{I}nst$, which contains all the information concerning all the instances of the identification protocols. The different oracles have a read/write access to this table.

Each line of $\mathcal{I}nst$ are referenced by an instance identifier and contains all the messages exchanged by both entity (reader and tag), and the result of the protocol. As described here, this table can be seen as an historic of all instances of the protocol when it works normally. However, we have to consider in this model the intervention of an adversary. For example, after a call to the $\mathcal{O}^{\text{Launch}}$ oracle, an adversary initiates an instance protocol π but can transmit the obtained nonce N_R to several tags. This can be modeled by several sub-instances of the protocol. This instance is only "closed" when the adversary transmits a message such that the reader accept or reject a tag. As a consequence, all responses obtained from tags are not invalid until the final result for π. Then, when the responses of these tags are written in $\mathcal{I}nst$, the result cannot be instantiated and is, for the moment, defined as $\perp$. An example of $\mathcal{I}nst$ table is presented in Figure 4 for the generic protocol described in Figure 2.

Instance	$m_{\mathcal{R}}$	sub-instances *subinst*			
		ρ	*pseudo*	$m_{\mathcal{T}}$	*result*
π_1	a_1	$\perp$	$\perp$	$\perp$	$\perp$
		ρ	*pseudo*	$m_{\mathcal{T}}$	*result*
π_2	a_2	1	$\mathtt{t}$	c	1
		ρ	*pseudo*	$m_{\mathcal{T}}$	*result*
π_3	a_3	1	$\mathtt{t}_1$	c_1	$\perp$
		2	$\mathtt{t}_2$	c_2	$\perp$
		3	$\mathtt{t}_3$	c_3	$\perp$
		ρ	*pseudo*	$m_{\mathcal{T}}$	*result*
π_4	a_4	1	$\mathtt{t}'_1$	c'_1	0
		2	$\mathtt{t}'_2$	c'_2	1
		3	$\mathtt{t}'_3$	c'_3	0
⋮	⋮	⋮			

Fig. 4. Example de table $\mathcal{I}nst$

The four presented cases of this example can be described as follows.

- Instance π_1 is the result of a request to $\mathcal{O}^{\text{Launch}}$ oracle. The reader has generated a nonce N_{R1} which have not be transmitted to a tag (justifying why there is no sub-instance).
- Instance π_2 can be the result of a request to $\mathcal{O}^{\text{Execute}}$ oracle with input $\mathtt{t}$.
- Instance π_3 is the result of several requests to $\mathcal{O}^{\text{SendTag}}$ with the nonce N_{R_3}. In this example, the nonce has been obtained by a $\mathcal{O}^{\text{Launch}}$ request. If the adversary had choose this value, the result of all the sub-instances should be 0 as the reader has a negligible probability to start a new instance with this nonce, and will obviously rejects all these tags..
- Instance π_4 is also the result of several requests to $\mathcal{O}^{\text{SendTag}}$, but this time, the adversary transmits the response of the tag $\mathtt{t}'_2$ to the reader.

9.2 The Future Untraceability is Reachable

In [32], Vaudenay proves that the generic scheme in Figure 2 is narrow-strong private and forward private when the encryption scheme is IND-CCA secure. In our model, we prove the following result.

Theorem 1. *The encryption-based authentication protocol ensures the future-untraceability property if the encryption scheme is IND-CCA secure.*

Proof. In this proof, we use the game proof technique introduced by Shoup in [31] to base the success of an adversary against the untraceability of the scheme to the advantage of an adversary against the IND-CCA property of the encryption scheme. In this proof, our goal is to define a final game where all messages sent by a tag are completely dissociated and thus does not permit an adversary to rely some of them. In other words, we replace each message by the encryption of a nonce. To conclude, we show that the success of an adversary cannot be significantly different of the success of the adversary in the final game, which corresponds to the dummy adversary.

In the initial game, we model the normal behavior of the whole system (*i.e.* of all oracles).

Game 0:

- $\textsc{Setup}(1^\lambda)$:
  - $*$ $nbt := 0$; $req := 0$; $\mathcal{F}ree := \emptyset$;
  - $*$ generates $(\mathsf{rsk}, \mathsf{rpk})$; $\mathsf{param} := (\mathsf{Enc}, \mathsf{Dec}, \lambda, \mathsf{rpk})$.
  - $*$ return param.

nbt and *req* are counter which respectively give the total number of pseudonym which have been attributed, and the total number of started instances. *req* does not take in account the number of sub-instances.

In order to simplify notations, we use in this proof a value ρ_{new} which attribute a new unique value of sub-instance for a given instance. The set $\mathcal{F}ree$ contains all the free tags of the system.

- $\mathcal{O}^{\textsc{CreateTag}}$:
  - $*$ $\mathsf{ID} \in_R \mathcal{ID}$; $\mathsf{tk_{ID}} := \mathsf{PRF}(\mathsf{ID}, \mathsf{rk})$; $\mathcal{F}ree \leftarrow_+ \mathsf{ID}$; $\mathsf{DB} \leftarrow_+ (\mathsf{ID}, \mathsf{tk_{ID}})$.

The set $\mathcal{ID}$ here corresponds to the set of all possible identifiers which can be created by the system.

- $\mathcal{O}^{\textsc{Draw}}(i)$:
  - $*$ if $i \leq \#(\mathcal{F}ree)$, then repeat i times:
    $\mathsf{ID} \in_R \mathcal{F}ree$; $\mathcal{F}ree \leftarrow_- \mathsf{ID}$; $nbt++$; $\mathsf{t}_{nbt} \in_R \mathbb{Z}$;
    $\mathsf{tk}_{\mathsf{t}_{nbt}} := \mathsf{DB}(\mathsf{ID}).\mathsf{tk_{ID}}$; $\mathsf{Tab} \leftarrow_+ (\mathsf{t}_{nbt}, \mathsf{ID}, \textsc{Actual})$;
    return t_{nbt};
  - $*$ else return $\perp$.

A unique pseudonym t_{nbt} is attributed to each new selected tag. We here introduce the key $\mathsf{tk}_{\mathsf{t}_{nbt}}$ initially equal to $\mathsf{tk_{ID}}$, where ID is the identifier associated to the pseudonym.

- $\mathcal{O}^{\textsc{Free}}(\mathsf{t})$:
  - $*$ $\mathsf{Tab}(\mathsf{t}).status := \textsc{Old}$; $\mathcal{F}ree \leftarrow_+ \mathsf{Tab}(\mathsf{t}).\mathsf{ID}$.

- $\mathcal{O}^{\textsc{Corrupt}}(\mathsf{t})$:
  - $*$ $\mathsf{ID} \leftarrow \mathsf{Tab}(\mathsf{t}).\mathsf{ID}$;
  - $*$ $\mathcal{C}orrupt \leftarrow_+ (\mathsf{t}, \mathsf{ID})$; return $\mathcal{A}(\mathsf{ID}, \mathsf{tk_{ID}})$.

- $\mathcal{O}^{\textsc{Execute}}(\mathsf{t})$:
  - $*$ $\mathsf{ID} \leftarrow \mathsf{Tab}(\mathsf{t}).\mathsf{ID}$;
  - $*$ $req++$; $a \in_R \{0,1\}^\lambda$; $c := \mathsf{Enc}(\mathsf{rpk}, \mathsf{tk_{ID}}||a)$;

```
    * $\mathcal{I}nst \leftarrow_+ (req, a, (1, \mathsf{t}, c, 1))$;
    * return $(req, (a, c), 1)$.
```

This oracle realizes a complete execution of the protocol between the reader and the tag represented by the pseudonym t. The adversary obtain the whole transcript of this protocol, the identifier of the instance req and the bit 1 corresponding to the success of the authentication. This instance is added in the table $\mathcal{I}nst$.

- $\mathcal{O}^{\textsc{Launch}}$:

```
    * $req++$;$a \in_R \{0,1\}^\lambda$; $\mathcal{I}nst \leftarrow_+ (req, a, \perp)$;
    * return $(a, req)$.
```

This oracle generate a new instance which initially contain the instance identifier req and a random value a, transmitted to the adversary.

- $\mathcal{O}^{\textsc{SendTag}}(a, \mathsf{t})$:

```
    * $\mathsf{ID} \leftarrow \mathsf{Tab}(\mathsf{t}).\mathsf{ID}$;
    * $c := \mathsf{Enc}(\mathsf{rpk}, \mathsf{tk}_{\mathsf{ID}}||a)$; return $c$;
    * if $\exists\ \pi$ such that $(\mathcal{I}nst(\pi).m_R = a)$, then:
          if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).result \neq \perp)$,
              then $\mathcal{I}nst(\pi).ssinst \leftarrow_+ (\rho_{new}, \mathsf{t}, c, 0)$;
          else $\mathcal{I}nst(\pi).ssinst \leftarrow_+ (\rho_{new}, \mathsf{t}, c, \perp)$;
    * else $req++$; $\mathcal{I}nst \leftarrow_+ (req, a, (1, \mathsf{t}, c, 0))$;
```

When a tag's message is created, a new sub-instance is added in $\mathcal{I}nst$. Depending of the state of the corresponding instance in $\mathcal{I}nst$, the sub-instance will not be created identically.

- If there is an instance π associated to the value a, two cases are possible:
  - if there is a sub-instance π such that $result \neq \perp$, then the tag given in input to the oracle will be rejected by the reader. Thus, the added sub-instance has to be $(\rho_{new}, \mathsf{t}, c, 0)$.
  - if $result = \perp$ for all existing sub-instances, then the corresponding instance is not yet closed. Thus, the added sub-instance has to be $(\rho_{new}, \mathsf{t}, c, \perp)$.
- Else, the input value a has been generated by the adversary. As we consider that a reader will create a new instance with this nonce with a negligible probability, $(req, a, (1, \mathsf{t}, c, 0))$ is added in $\mathcal{I}nst$.

- $\mathcal{O}^{\textsc{SendReader}}(c, \pi)$:

```
    * if $forall\ \rho$ $(\mathcal{I}nst(\pi).ssinst(\rho).result = \perp)$ then:
        - if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).m_{\mathcal{T}} = c)$, then $(\mathcal{I}nst(\pi).ssinst(\rho).result = 1)$;
        - else:
              $\mathsf{tk}_{\mathsf{t}}||a' := \mathsf{Dec}(\mathsf{rsk}, c)$;
              if $a' = \mathcal{I}nst(\pi).m_{\mathcal{R}}$, then:
                  if $\mathsf{tk}_{\mathsf{ID}} = \mathsf{PRF}(\mathsf{ID}, \mathsf{rk})$, then $\mathcal{I}nst(\pi).ssinst \leftarrow_+ (\rho_{new}, \mathsf{ID}, c, 1)$;
              else $\mathcal{I}nst(\pi).ssinst \leftarrow_+ (\rho_{new}, \perp, c, 0)$;
```

If the instance π is complete or if it has been initialized by the adversary, then it exists a sub-instance where the result is 1 or 0. In this case, the oracle cannot answer to this request and stops.

Else the two following cases are possible.

- It exists a sub-instance which contains the encrypted message c, necessarily corresponding to the answer of a legitimate tag in this instance (by construction of $\mathcal{I}nst$). The oracle replace the field $result$ of this sub-instance by 1, meaning that the associated tag has been authenticated in the instance π.

– It does not exist any sub-instance containing c. Then this value as been generated by the adversary. The oracle decrypts the message in order to verify its validity. Whatever the result of the identification is, the oracle generates the appropriate sub-instances with the corresponding result.

- $\mathcal{O}^{\textsc{Return}}(\pi)$:

```
* if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).result = 1)$ then return 1;
* else
      if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).result = 0)$ then return 0;
      else return $\perp$.
```

- Adversary's response: $(\mathsf{t}_i, \mathsf{t}_j)$

In the untraceability experiment, the goal of an adversary is to output a non-obvious link $(\mathsf{t}_i, \mathsf{t}_j)$. We define the event S_0 as: "$(\mathsf{t}_i, \mathsf{t}_j)$ is a non-obvious link". We thus obviously have:

$$Succ^{\mathsf{Unt}}_{\mathcal{S},\mathcal{A}}(1^\lambda) = Pr[S_0]$$

We now define the final game where all tag's message are replaced by encryption of nonce. As all messages are now simulated, the adversary in the final game is a dummy adversary. Then to prove that the scheme is untraceable, we will have to demonstrate that the success probability of an adversary and the one of the dummy adversary are equal.

For clarity reasons, we only redefine oracles which are different of those in the initial game, and modifications are written in bold.

Final Game

- $\textsc{Setup}(1^\lambda)$ (unchanged).
- $\mathcal{O}^{\textsc{CreateTag}}$ (unchanged).
- $\mathcal{O}^{\textsc{Affect}}(i)$ (unchanged).
- $\mathcal{O}^{\textsc{Free}}(t)$ (unchanged).
- $\mathcal{O}^{\textsc{Launch}}$ (unchanged).
- $\mathcal{O}^{\textsc{SendReader}}(c, \pi)$ (unchanged).

- $\mathcal{O}^{\textsc{Execute}}(\mathsf{t})$:

```
  * $req{+}{+}$; $a \in_R \{0,1\}^\lambda$; $\mathbf{n_T \in_R \{0,1\}^{\lambda'}}$; $c := \mathsf{Enc}(\mathsf{epk}, \mathbf{n_T}||a)$;
  * $\mathcal{I}nst \leftarrow_+ (req, a, (1, \mathsf{t}, c, 1))$;
  * return $(req, (a, c), 1)$.
```

k' is defined by $k' := |\mathsf{tk_{ID}}| + |\mathsf{ID}|$.

- $\mathcal{O}^{\textsc{SendTag}}(a, \mathsf{t})$:

```
  * $\mathbf{n_T \in_R \{0,1\}^{\lambda'}}$; $c := \mathsf{Enc}(\mathsf{rpk}, \mathbf{n_T}||a)$;
  * if $\exists\ \pi$ such that $(\mathcal{I}nst(\pi).m_R = a)$ then:
        if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).result \neq \perp)$
            then $\mathcal{I}nst(\pi).ssinst \leftarrow_+ (\rho_{new}, \mathsf{t}, c, 0)$;
        else $\mathcal{I}nst(\pi).ssinst \leftarrow_+ (\rho_{new}, \mathsf{t}, c, \perp)$.
  * else $req{+}{+}$; $\mathcal{I}nst \leftarrow_+ (req, a, (1, \mathsf{t}, c, 0))$;
  * return $c$.
```

The only things changed in this final game are the messages send by tags. The recording in $\mathcal{I}nst$ is unchanged as legitimate tags must still be accepted. The event S_{final} is also defined by "$(\mathsf{t}_i, \mathsf{t}_j)$ is a non-obvious link". As all transcripts are simulated in the final game, the outputted link by the adversary in the final game is non-obvious only if she correctly guess it. It is thus obvious that this adversary is a dummy adversary and consequently, we have:

$$Succ^{\mathsf{Unt}}_{\mathcal{S},\mathcal{A}_d}(1^\lambda) = Pr[S_{final}]$$

In order to conclude, we have now to exhibit all the transitions games which will permit us to rely the success of an adversary to the success of the dummy adversary, and thus to conclude. We have to introduce as much intermediate games (including the final game) as there are request to oracles $\mathcal{O}^{\text{Execute}}$ and $\mathcal{O}^{\text{SendTag}}$. Let q be this number. Each of these games will replace one more "correct" encryption made by one of these two oracles by the encryption of a nonce. Each event S_i are still defined by "$(\mathsf{t}_i, \mathsf{t}_j)$ is a non-obvious link". We consider that we have ordered all the "encryption" requests (done during requests to $\mathcal{O}^{\text{Execute}}$ and $\mathcal{O}^{\text{SendTag}}$). The i-th game is defined as follows:

- the i first requests return an encryption of $(\mathsf{rpk}, n_T||a)$ where n_T is randomly chooses in $\{0,1\}^{\lambda'}$ for each request.
- the $q-i$ next requests return an encryption of $(\mathsf{rpk}, \mathsf{tk}_{\mathsf{ID}}||a)$, where t is the pseudonym given in input of the oracle and $\mathsf{ID} = \mathsf{Tab}(\mathsf{t}).\mathsf{ID}$.

Note that only one encrypted message is modified between two games (even between game 0 and 1, or $q-1$ and the final game). If $|Pr[S_i] - Pr[S_{i-1}]|$ is not negligible, it is possible to trivially define a distinguisher for the IND-CCA2 property of the encryption scheme.

Remark 7. It is important to notice that during the requests to $\mathcal{O}^{\text{SendReader}}$, the simulator has to decrypt the value c if it has been transmitted by an adversary (*i.e.* which is not in $\mathcal{I}nst$). As the adversary can transmit this kind of message at any moment of the experiment (before or after the i-th encryption request), the simulator must have access to a decryption oracle during the whole game. Thus it is necessary, in this proof, that the encryption scheme is IND-CCA2 secure.

The distinguisher $\mathcal{D}$ can be defined as follows. He first computes the i-th cipher correctly using the keys $\mathsf{tk}_{\mathsf{ID}}$. When the adversary makes the $(i+1)$-th requests, $\mathcal{D}$ defines two messages $m_0 = \mathsf{tk}_{\mathsf{ID}}||a$ and $m_1 = n_T||a$, and sends these messages to the challenger of $\mathsf{Exp}_{IND-CCA2}$. The distinguisher sends the received encryption $c = \mathsf{Enc}(\mathsf{rpk}, m_b)$ to the adversary. Depending of b, the adversary will play the game i, if $b = 0$, and the game $i+1$ otherwise. More formally we have:

$$Pr[\mathcal{D} \to 1 | b = 0] = Pr[S_i] \quad \text{and} \quad Pr[\mathcal{D} \to 1 | b = 1] = Pr[S_{i+1}].$$

Thus:

$$\begin{aligned} |Pr[S_i] - Pr[S_{i+1}]| &= |Pr[\mathcal{D} \to 1 | b = 1] - Pr[\mathcal{D} \to 1 | b = 1]| \\ &= Adv_{\mathcal{D},\mathcal{S}_E}^{\mathsf{Exp}_{IND-CCA2}} \end{aligned}$$

In conclusion,

$$\begin{aligned} \left| Succ_{\mathcal{S},\mathcal{A}}^{\mathsf{Unt}}(1^\lambda) - Succ_{\mathcal{S},\mathcal{A}_d}^{\mathsf{Unt}}(1^\lambda) \right| &= |Pr[S_0] - Pr[S_{final}]| \\ &= |Pr[S_0] - Pr[S_1] + Pr[S_1] - \ldots + Pr[S_{q-1}] - Pr[S_{final}]| \\ &\leq \sum_{i=0}^{q-1} |Pr[S_i] - Pr[S_{i+1}]| \\ &\leq q.Adv_{\mathcal{D},\mathcal{S}_E}^{\mathsf{Exp}_{IND-CCA2}} \end{aligned}$$

As q is polynomial, the adversary's advantage in the untraceability experiment is negligible in λ which prove the theorem. □

We have thus proved that our future untraceability can be reached by an RFID authentication and identification scheme, and thus that the full strong adversary (*i.e.* not narrow) can be used to prove the security of such schemes. We now give a concrete instantiation of the generic construction from Vaudenay, using an IND-CCA secure encryption scheme.

9.3 A Very Practical Instantiation: the DHAES Case

The DHAES has been introduced in [1] by Abdalla, Bellare and Rogaway and has been submitted to the IEEE P1363a standard. Its aim is to propose a method to encrypt strings using the Diffie-Hellman assumption, since the standard El Gamal encryption scheme has some flaws when regarding the message as a string. It is as efficient as the standard El Gamal encryption but has more and better security properties since it has been proved to have the indistinguishability property against adaptive chosen ciphertext attacks with unlimited access to the decryption oracle (IND-CCA2). It is thus possible to directly use it in the above generic construction to obtain the security of the underlying privacy-preserving RFID identification scheme (see 7 and [32]).

Let $\mathbb{G}$ be a cyclic group of prime order q. The reader's private key to decrypt a message is $\mathsf{rsk} \in \mathbb{Z}_q^*$ and the corresponding encryption public key is $\mathsf{rpk} = g^{\mathsf{rsk}}$ where g is a generator of $\mathbb{G}$. Moreover, let $\mathsf{tk}_{\mathsf{ID}}$ be the λ-bit key of a tag ID, which is known by both the tag and the reader. The DHAES encryption scheme can be used to obtain and RFID identification scheme as described in Figure 5, where $\mathcal{H}$ is a cryptographically secure hash function.

$$\begin{array}{rcl}
\mathcal{R} & & \mathcal{T} \\
a \in \{0,1\}^{\lambda} & \xrightarrow{\qquad\quad a \qquad\quad} & T_0 = g^r; R = \mathsf{rpk}^r \\
 & & \mathsf{k}_1 \| \mathsf{k}_2 = \mathcal{H}(T_0 \| R) \\
 & & T_1 = \textsc{Mac}(\mathsf{tk}_{\mathsf{ID}} \| a, \mathsf{k}_1) \\
\mathsf{k}_1 \| \mathsf{k}_2 = \mathcal{H}(T_0 \| T_0^{\mathsf{rsk}}) & \xleftarrow{\qquad T_0, T_1, T_2 \qquad} & T_2 = \textsc{SymEnc}(\mathsf{tk}_{\mathsf{ID}} \| a, \mathsf{k}_2) \\
\mathsf{tk}_{\mathsf{ID}} \| a = \textsc{SymDec}(T_2, \mathsf{k}_2) & & \\
\text{Verify } T_1 \text{ (using } \mathsf{k}_1\text{), } a \text{ and } \mathsf{tk}_{\mathsf{ID}} & &
\end{array}$$

Fig. 5. DHAES based protocol

The IND-CCA property of an encryption scheme is a strong property which need adapted schemes. In the following, we try to use weaker encryption scheme, while trying to keep the same security level for the resulting RFID identification and authentication scheme.

10 Constant Fixed Non Malleability ⇒ Untraceability

In this section, we study the case of encryption schemes which are not IND-CCA. In fact, we can consider IND-CPA schemes (with various results), or something between IND-CCA and IND-CPA.

10.1 Insecure Scheme: the Hash El Gamal Case

The Hash El Gamal encryption scheme [14] consists in computing $T_0 = m \oplus \mathcal{H}(\mathsf{epk}^r)$ and $T_1 = g^r$ for the encryption of the message m, and is known to be IND-CPA, but not IND-CCA.

Using the hash El Gamal encryption scheme in the Vaudenay's construction, it is trivially possible to break the soundness of the resulting scheme. Concretely, from one successful authentication $T_0 = (\mathsf{tk}_{\mathsf{ID}} \| a) \oplus \mathcal{H}(\mathsf{epk}^r)$ and $T_1 = g^r$, one can fake the valid tag by simply computing, on reception of the new random $\tilde{a}$, $\tilde{T_0} = T_0 \oplus (0 \cdots 0 \| (a \oplus \tilde{a}))$ which is obviously equal to $(\mathsf{tk}_{\mathsf{ID}} \| \tilde{a}) \oplus \mathcal{H}(\mathsf{epk}^r)$. Thus, $(\tilde{T_0}, T_1)$ is a valid authentication of ID under the request $\tilde{a}$. One possibility to avoid this attack is to keep all received successful authentications and checks that the received T_1 has not previously been used. But we do not want the reader to perform so many comparisons and store so much data in its database.

10.2 Secure Scheme: the Rabin Case

The Rabin cryptosystem [29] is a public key cryptosystem introduced by Rabin whose security is related to the factorization problem. The main disadvantage of this system is that each ciphertext can be generated by any of four possible inputs so that some extra computation is needed during the decryption phase to output the correct corresponding plaintext.

Rabin in RFID Systems. The Rabin encryption scheme can be described as follows.

- KeyGen: let p and q be two large prime numbers and let $n = pq$. The private key rsk is the factorization (p, q) of n and the corresponding public key rpk is n.
- Enc: when someone wants to encrypt a message m in the range $[1, n[$, he has to compute the ciphertext $c = m^2 \pmod{n}$. To overcome the problem of selecting the correct plaintext from among four possibilities, one solution is to add prespecified redundancy to the original plaintext m prior to encryption. Then, with high probability, exactly one of the four square roots will possess the right redundancy. Note that if this randomness is always modified, this transform the cryptosystem to a probabilistic one but it is however not possible to prove that it is IND-CCA secure.
- Dec: the owner of the decryption key $\mathsf{sk} = (p, q)$ retrieves m by computing $c^{1/2} \pmod{n}$ (which is possible only when knowing the factorization of n) and uses the prespecified redundancy to find the correct plaintext.

In the RFID setting, this cryptosystem has been used by Shamir to describe a MAC scheme [30]. In [27], Oren and Feldhofer also use this cryptosystem in the design of their privacy-preserving RFID identification scheme named WIPR. As the Rabin cryptosystem is deterministic (and thus not IND-CPA), the protocol needs to be modified so as to include some randomness coming from the tag, as it is described in [27].

The WIPR System. Let p and q be two large prime numbers and let $n = pq$. The reader's private key rsk is the factorization (p, q) of n and the corresponding public key rpk is n. The scheme is described in Figure 6, where ByteMix is a publicly known byte-interleaving operation used to ensure that neither the tag nor the reader fully dominates a large element of the plaintext. Moreover, reduction modulo n is replaced by an addition of a multiple of the divisor n. Moreover, ByteMix algorithm is a publicly known byte-interleaving operation used to ensure that neither the tag nor the reader fully dominates a large element of the plaintext. It takes as input a message and outputs another message. Finally, $\mathsf{tk}_{\mathsf{ID}}$ is again a λ-bit key of a tag ID, which is known by both the tag and the reader.

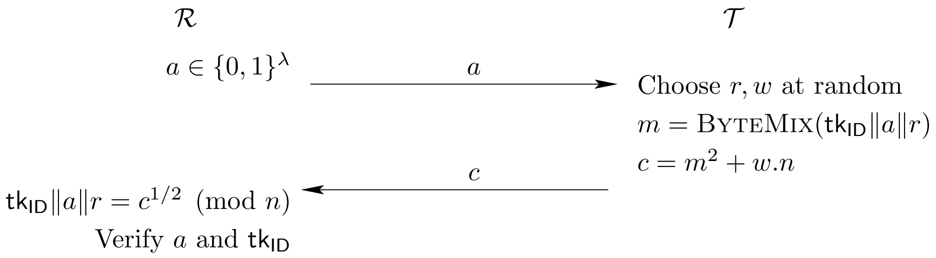
$\mathcal{R}$
$\mathcal{T}$
$a \in \{0,1\}^\lambda$
a
Choose r, w at random
$m = \textsc{ByteMix}(\mathsf{tk}_{\mathsf{ID}} \| a \| r)$
$c = m^2 + w.n$
c
$\mathsf{tk}_{\mathsf{ID}} \| a \| r = c^{1/2} \pmod{n}$
Verify a and $\mathsf{tk}_{\mathsf{ID}}$

Fig. 6. The WIPR protocol

Security Considerations. As said above, it is well-known that the Rabin cryptosystem is not IND-CCA. The preprocessing step which consists in adding some redundancy permits to overcome some known chosen ciphertext attacks but no security proof can be done. However, it is not possible to prove that the resulting encryption scheme is IND-CCA secure.

In [34], the authors show that without a good preprocessing step (*e.g.* a weak ByteMix), the scheme is insecure. They use the preprocessing step SAEP (Simple OAEP) so as to prove the security in a simple model where, unfortunately, strong privacy is not taken into account.

This should be impossible, even if the adversary can obtain several encryptions of this kind where the challenge a is known. But what we need is that an adversary, taken on input the public key and being able to send some a and obtain in response the Rabin encryption of $\mathsf{tk}_{\mathsf{ID}}\|a\|r$ where $\mathsf{tk}_{\mathsf{ID}}$ is unknown and fixed and r is random, is not able to output the encryption of $\mathsf{tk}_{\mathsf{ID}}\|\tilde{a}\|\tilde{r}$ for a given $\tilde{a}$. As Oren and Feldhofer, we are not able to prove the "partial non-malleability" of this scheme. However, as no such attack already exists, it seems that this system and the WIPR one are secure.

10.3 Secure Scheme: the El Gamal Case

The El Gamal encryption scheme has been introduced in [19] and is now largely used in many cryptographic papers. The El Gamal encryption scheme can be used either in groups of prime order or in groups of unknown order. In the following, we use a group of prime order. It has been shown that this scheme is IND-CPA but not IND-CCA (as malleable).

Description of the System. Let $\mathbb{G}$ be a cyclic group of prime order q. The reader's private key to decrypt a message is $\mathsf{rsk} \in \mathbb{Z}_q^*$ and the corresponding public key is $\mathsf{rpk} = g^{\mathsf{rsk}}$ where g is a generator of $\mathbb{G}$. Finally, $\mathsf{tk}_{\mathsf{ID}}$ is again a λ-bit key of a tag ID, which is known by both the tag and the reader. We next obtain the RFID identification scheme described in Figure 7.

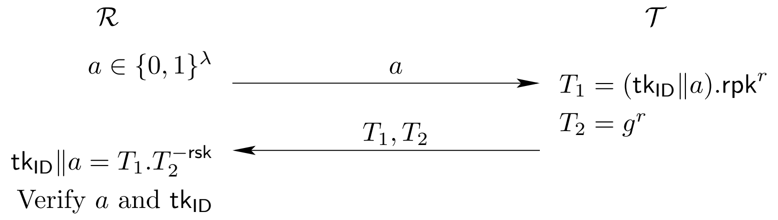
$\mathcal{R}$
$\mathcal{T}$
$a \in \{0,1\}^\lambda$
a
$T_1 = (\mathsf{tk}_{\mathsf{ID}}\|a).\mathsf{rpk}^r$
$T_2 = g^r$
T_1, T_2
$\mathsf{tk}_{\mathsf{ID}}\|a = T_1.T_2^{-\mathsf{rsk}}$
Verify a and $\mathsf{tk}_{\mathsf{ID}}$

Fig. 7. El Gamal based protocol

Security Considerations. As for the Rabin case, we are unable to provide a proof that the construction based on El Gamal is secure but again, it would seem that this is the case.

In addition to what has been said for the Rabin case, the El Gamal opens a new problem. In fact, we should be careful here that the message $\mathsf{tk}_{\mathsf{ID}}\|a$ truly belongs to the right working group. This should be done by using a good preprocessing step. Note however that this may imply some additional computations for the RFID tag. This is for example the case if the implementation is done using elliptic curves [10].

10.4 The Constant Fixed Non Malleability Property

In [7], Bellare *et al.* have shown that the IND-CCA property is equivalent to the NM-CCA one. Moreover, the soundness property of the Vaudenay's generic scheme intuitively comes from the non-malleability of the public key cryptosystem while the privacy property comes from the indistinguishability property. But the non-malleability property may be too strong for our purpose and, as we need lightweight computation, this may be not a good choice. In fact, most of existing IND-CCA secure cryptosystems are not relevant in the RFID setting and thus, cannot be used in practice.

We first notice that in the Vaudenay's generic construction, the RFID tag does not simply encrypt a message but the concatenation of some secret values $\mathsf{tk}_{\mathsf{ID}}$ that are always the same for a particular tag together with some randomness a that are "publicly" known, since they are sent in clear by the reader. We thus introduce the following security definition for encryption schemes.

Definition 9 (Constant Fixed Non Malleability). *A public key encryption scheme verifies the* constant fixed non malleability *if given the encryption public key and having access to an oracle which on input a value a, outputs the encryption of* $\mathsf{tk}_{ID}\|a$*, where* tk_{ID} *is secret, an adversary is unable to output the encryption of* $\mathsf{tk}_{ID}\|\tilde{a}$ *on input* $\tilde{a}$ *with non-negligible probability.*

As a conclusion, if we are able to find a public key cryptosystem not necessarily IND-CCA but having the constant fixed non malleability property, then we have the following result on privacy-preserving RFID systems.

Theorem 2. *The Vaudenay's generic construction given in Figure 2 using a constant fixed non malleable and IND-CPA encryption scheme is secure and forward private.*

11 IND-CPA Cryptosystem + MAC ⇒ (Future) Untraceability

In this section, we first provide a generic construction of a privacy-preserving RFID identification system which make use of any IND-CPA public key cryptosystem and a MAC function. Next, we provide a practical implementation using the Hash El Gamal encryption scheme.

11.1 MAC function

A cryptographic message authentication code (MAC) is a cryptographic tool used to authenticate a message and belongs to the family of symmetric cryptography. A *MAC scheme* denoted $\mathcal{M}$ is composed of the following procedures: KeyGen is the key generation algorithm which permits to generate the MAC key denoted k; Mac is the code generation algorithm which accepts as input an arbitrary-length message m and the secret key k and outputs the MAC σ for message m, under the secret key k; VerMac is the code verification algorithm which takes as input a message m, the secret key k and a message authentication code σ and outputs 1 if $\sigma = \textsc{Mac}(m, \mathsf{k})$ and 0 otherwise.

To be considered as secure, a MAC scheme should resist to existential forgery under chosen-plaintext attacks (EF-CPA). This means that even if an adversary $\mathcal{A}$ has access to an oracle which possesses the secret key and generates MACs for messages chosen by the adversary, $\mathcal{A}$ is unable to guess the MAC for a message it did not query to the oracle.

11.2 Our New Generic Construction

Our generic construction needs a public key cryptosystem and a MAC scheme as defined above.

Proposed construction. Let $\mathcal{E}$ be a public-key encryption scheme with the IND-CPA property and a MAC scheme $\mathcal{M}$ such as defined above, we next introduce our new RFID identification scheme in Figure 8, where each tag ID shares with the reader a unique key denoted $\mathsf{tk}_{\mathsf{ID}}$, and where the reader key pair $(\mathsf{rsk}, \mathsf{rpk})$ corresponds to the public key cryptosystem key pair, that is, rsk is a secret decryption key and rpk is the corresponding encryption public key.

Security Considerations. Assume an adversary able to impersonate an uncorrupted tag. As she has no control over the nonce a chosen by the reader, the returned values will correspond, with a significant probability, to a new message $\mathsf{tk}_{\mathsf{ID}}\|a$, which contradict the EF-CPA property of the MAC. Consequently, under the EF-CPA property, our new generic construction is sound.

Regarding the untraceability property, we have to prove that for every adversary $\mathcal{A}$ of this protocol, there exists a blinded adversary $\mathcal{A}^{\mathcal{B}}$ such that whatever $\mathcal{A}$ do, $\mathcal{A}^{\mathcal{B}}$ can obtain the same result by interacting

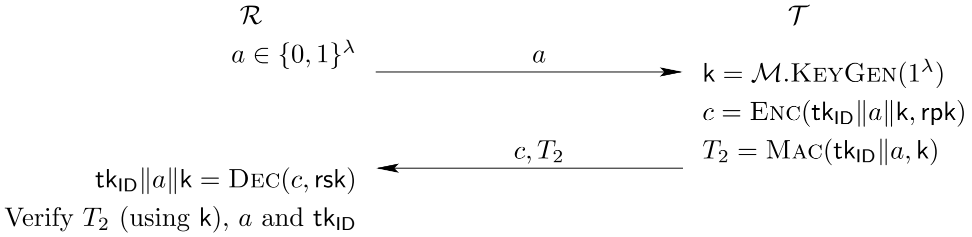
$\mathcal{R}$
$\mathcal{T}$
$a \in \{0,1\}^\lambda$
a
$\mathsf{k} = \mathcal{M}.\textsc{KeyGen}(1^\lambda)$
$c = \textsc{Enc}(\mathsf{tk_{ID}} \| a \| \mathsf{k}, \mathsf{rpk})$
c, T_2
$T_2 = \textsc{Mac}(\mathsf{tk_{ID}} \| a, \mathsf{k})$
$\mathsf{tk_{ID}} \| a \| \mathsf{k} = \textsc{Dec}(c, \mathsf{rsk})$
Verify T_2 (using k), a and $\mathsf{tk_{ID}}$

Fig. 8. A generic scheme based on an IND-CPA encryption scheme and a MAC

with the simulator. The game technique, presented by Shoup is perfectly adapted to obtain this result. The purpose is to replace every interactions with oracles of $\mathcal{A}$ by an answer of the simulator. The success of each game is the experiment that perform the adversary, for example : find a non-trivial link between two pseudonyms. If the difference between the success probabilities of two successive games is negligible, then it follows that the difference between the success probability of the adversary and the one of the blinded adversary is negligible.

We give here some details about this proof. It is possible to replace one by one every plaintexts of the public key cryptography by random messages. As detailed in [31], these operations cannot influenced the success probability of the adversary, otherwise it is possible to exhibit a distinguisher for the IND-CPA experiment. In order to obtain a perfect simulation of all messages exchanged during the experiment, it is also necessary to modify inputs of the MAC function. For this purpose, the MAC scheme must be a pseudo random function, which is also required to avoid attacks as those presented in [8]. This is not restrictive in practice as most of MAC schemes verifies this property. In conclusion, as we use the game technique, the difference between the success probabilities of $\mathcal{A}$ and $\mathcal{A}^{\mathcal{B}}$ is increased by the advantage of an adversary against the IND-CPA property of the encryption scheme plus the advantage of an adversary against the pseudo-random property. As both of these advantages are negligible by definition, the success probability of $\mathcal{A}$ must be negligible which demonstrates the untreacability property of our scheme.

11.3 The Hash El Gamal Case

The Hash El Gamal encryption scheme [14] is a variant of the classical El Gamal encryption scheme which uses a hash function. It allows a compact ciphertext and avoids problems with messages whose orders are not the ones of the group. We have shown in Sectionr̃efss:insecurehasheg that using this scheme alone is not enough to obtain a secure scheme. However, we can use it with the above generic proposal to obtain a secure and efficient scheme. But, even if this is easily possible, we describe in this section a slightly different scheme which permits us to obtain a more efficient scheme than the “simple” applying of the Hash El Gamal encryption scheme in the above generic construction.

Description of the System. Let $\mathbb{G}$ be a cyclic group of prime order q. The reader’s private key to decrypt a message is $\mathsf{rsk} \in \mathbb{Z}_q^*$ and the corresponding encryption public key is $\mathsf{rpk} = g^{\mathsf{rsk}}$, where g is a generator of $\mathbb{G}$. We thus obtain the RFID identification scheme described in Figure 9, where $\mathsf{tk_{ID}}$ and k_{ID} are secret λ-bits values shared by the tag ID and the reader.

In a nutshell, we have described an efficient authentication scheme based on an IND-CPA public-key cryptosystem and a MAC scheme. It is sound and untraceable as the DHAES scheme (see section 9.3) and seems to be efficient. In the next section we give some implementation estimation for all presented schemes. We will then be able to conclude about the relevancy of an authentication scheme based on an IND-CPA public-key cryptosystem. First of all, we formally prove that this scheme is secure, *i.e.* sound and untraceable.

Security considerations. We now prove that our solution is sound and future untraceable.

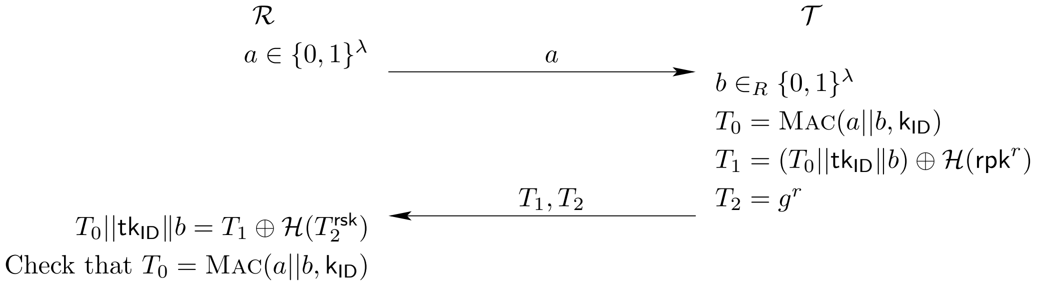
$\mathcal{R}$
$\mathcal{T}$
$a \in \{0,1\}^\lambda$
a
$b \in_R \{0,1\}^\lambda$
$T_0 = \text{MAC}(a||b, \mathsf{k_{ID}})$
$T_1 = (T_0||\mathsf{tk_{ID}}||b) \oplus \mathcal{H}(\mathsf{rpk}^r)$
$T_2 = g^r$
T_1, T_2
$T_0||\mathsf{tk_{ID}}||b = T_1 \oplus \mathcal{H}(T_2^{\mathsf{rsk}})$
Check that $T_0 = \text{MAC}(a||b, \mathsf{k_{ID}})$

Fig. 9. Hash El Gamal based protocol

Theorem 3. *The Hash El Gamal based solution given in Figure 9is sound.*

Proof. We prove that our scheme $\mathcal{S}$ is sound by reduction to the existential unforgeability of the MAC function f. We assume that there exists a polynomial time adversary $\mathcal{A}$ able to win the soundness experiment of our scheme with a non negligible probability. More precisely, $\mathcal{A}$ is able to successfully participate in an authentication protocol with the reader. Then we construct a polynomial time machine $\mathcal{M}$ which emulates the challenger in the soundness experiment and interacts with $\mathcal{A}$ in order to output, with a non negligible probability, an existential forgery of the function MAC. We denote by $\epsilon_{\mathcal{A}}$ the probability of success $Succ^{\mathsf{sound}}_{\mathcal{S},\mathcal{A}}(1^\lambda)$ of the adversary $\mathcal{A}$, where λ is the security parameter. We construct a polynomial adversary $\mathcal{M}$ which plays the EF-CMA experiment with a challenger $\mathcal{C}$.

First, $\mathcal{C}$ randomly generates the secret key sk used in MAC and gives access to $\mathcal{M}$ to the

- $\mathcal{O}^{\mathsf{sk}}$ oracle which on input a message m outputs $\text{MAC}(m, \mathsf{sk})$, and to the
- $\mathcal{O}^{V\mathsf{sk}}$ oracle which on input m and σ outputs 1 if $\sigma = \text{MAC}(m, \mathsf{sk})$ and 0 otherwise.

$\mathcal{M}$ generates the system of the authentication scheme as follows. It first generates at random two distinct prime numbers p and q such that $q|p-1$ and then randomly chooses $g \leftarrow_R \mathbb{Z}_p^*$, $\mathsf{rsk} \leftarrow_R \mathbb{Z}_q$, it sets $\mathsf{rpk} = g^{\mathsf{rsk}}$, and sends (p, q, g, rpk) to $\mathcal{A}$.

Let n be the number of times the adversary $\mathcal{A}$ requests the $\mathcal{O}^{\text{CREATETAG}}()$ oracle. Without loss of generality, we assume that $\mathcal{A}$ makes the n requests at the beginning of the experiment, during which $\mathcal{M}$ randomly generates n identifiers, denoted by $\mathsf{ID}_1, \ldots, \mathsf{ID}_n$. $\mathcal{M}$ next selects a value $\mathsf{ID}_{\tilde{i}} \in \{\mathsf{ID}_1, \ldots, \mathsf{ID}_n\}$, which corresponds to the *target* tag. Finally, for all i such that $i \neq \tilde{i}$, $\mathcal{M}$ randomly chooses $\mathsf{k}_i \in \{0,1\}^\ell$, $\mathsf{tk}_{\mathsf{ID}_i} \in \mathcal{SK}$ and associates in a database all couple $(\mathsf{ID}_i, \mathsf{k}_i, \mathsf{tk}_{\mathsf{ID}_i})$. Next, $\mathcal{M}$ chooses at random $\mathsf{k}_{\tilde{i}} \in \{0,1\}^\ell$. After that, $\mathcal{M}$ acts as follows:

- On reception of a request to the $\mathcal{O}^{\text{EXECUTE}}$ (resp. $\mathcal{O}^{\text{SENDTAG}}$) oracle, $\mathcal{M}$ chooses $a \in_R \{0,1\}^\lambda$ (resp. receives a from $\mathcal{A}$), $b \in_R \{0,1\}^\lambda$ and $w \in_R \mathbb{Z}_q$. There are then two cases:
  - if $i \neq \tilde{i}$, it computes and sends to the adversary the values $T_1 = (\text{MAC}(a||b, \mathsf{k}_i)||\mathsf{tk}_{\mathsf{ID}_i}||b) \oplus \mathcal{H}(y^w)$ and $T_2 = g^w$.
  - if $i = \tilde{i}$, it first asks the oracle $\mathcal{O}^{\mathsf{sk}}(a||b)$ to obtain $\sigma = \text{MAC}(a||b, \mathsf{sk})$ and then sends $T_1 = (\sigma||\mathsf{tk}_{\mathsf{ID}_i}||b)) \oplus \mathcal{H}(y^w)$ and $T_2 = g^w$ to $\mathcal{A}$.
- On reception of a request $\mathcal{O}^{\text{SENDREADER}}(T_1, T_2)$ related to a challenge a, the corresponding tag is necessarily corrupted since otherwise, the adversary has win the game and only consider this case during Step 3 of the soundness experiment.
- On reception of a request to the $\mathcal{O}^{\text{CORRUPT}}$ oracle related to the tag ID_i, there are two cases. If $i \neq \tilde{i}$, $\mathcal{M}$ returns $(\mathsf{ID}, \mathsf{tk}_{\mathsf{ID}_i}, \mathsf{k}_i)$ and if $i = \tilde{i}$, $\mathcal{M}$ stops the experiment.

Finally, $\mathcal{A}$ launches an IDENT protocol, $\mathcal{M}$ randomly chooses $\tilde{a}$, such that $\tilde{a}$ is different from all the previously sent a, and sends it to $\mathcal{A}$, which outputs (T_1, T_2). $\mathcal{M}$ first computes $\sigma||\mathsf{tk}_{\mathsf{ID}_i}||b = T_1 \oplus \mathcal{H}(T_2^x)$. There are then two cases:

- if $i \neq \tilde{i}$, $\mathcal{M}$ tests whether or not $\textsc{Mac}(a||b, \mathsf{k}_i) = \sigma$. If yes, the adversary has won the game but its response is not useful and the game is aborted. Otherwise, the adversary is rejected and $\mathcal{M}$ outputs 0.
- if $i = \tilde{i}$, $\mathcal{M}$ outputs $(\sigma, a||b)$ and wins the experiment.

Finally, the probability of success for $\mathcal{M}$ depends on the cases where the game above is aborted. More precisely, we have:

$$\begin{aligned} Succ^{\mathsf{ef\text{-}cma}}_{\textsc{Mac},\mathcal{M}}(1^\lambda) = Pr[Exp^{\mathsf{ef\text{-}cma}}_{\textsc{Mac},\mathcal{M}} = 1] &= \Big(1 - Pr[\mathcal{O}^{\textsc{Corrupt}}(\tilde{P}) \leftarrow \mathcal{A}]\Big).Pr[P = \tilde{P}].Pr[Exp^{\mathsf{sound}}_{\mathcal{S},\mathcal{A}} = 1] \\ &= \frac{n - q_c}{n}.\frac{1}{n - q_c}.\epsilon_{\mathcal{A}} = \frac{\epsilon_{\mathcal{A}}}{n} \end{aligned}$$

and thus

$$Succ^{\mathsf{sound}}_{\mathcal{S},\mathcal{A}}(1^\lambda) = n \cdot Succ^{\mathsf{ef\text{-}cma}}_{\textsc{Mac},\mathcal{A}}(1^\lambda).$$

As $Succ^{\mathsf{ef\text{-}cma}}_{\textsc{Mac},\mathcal{A}}(1^\lambda)$ is negligible in λ and n is polynomial, then the success of an adversary in the soundness experiment is negligible in λ which concludes the proof. □

Theorem 4. *The Hash El Gamal based solution given in Figure 9is future untraceable.*

Proof. To prove that our main scheme verifies the future-untraceability property, we again use the game technique introduced by Shoup [31], as for the untraceability proof of Vaudenay's generic scheme (see section 9.2).

- **Game 0:** this first game corresponds to the untraceability experiment described in Section 6.2.
  - $\textsc{Setup}(1^\lambda)$:
    - $*$ $nbt := 0$; $\mathcal{F}ree := \emptyset$
    - $*$ System: $g \in_R \mathbb{Z}_p^*$;
    - $*$ Reader: $\mathsf{rsk} \in_R \mathbb{Z}_q$; $\mathsf{rpk} \in g^{\mathsf{rsk}}$;
  - $\mathcal{O}^{\textsc{CreateTag}}$:
    - $*$ $\mathsf{ID} \in_R \mathcal{ID}$; $\mathsf{tk_{ID}} \in_R \{0,1\}^{\lambda'}$; $\mathsf{k_{ID}} \in_R \{0,1\}^l$; $\mathcal{F}ree \leftarrow_+ \{\mathsf{ID}\}$; $\mathsf{DB} \leftarrow_+ (\mathsf{ID}, \mathsf{tk_{ID}}, \mathsf{k_{ID}})$
  - $\mathcal{O}^{\textsc{Draw}}(i)$:
    - $*$ if $i \leq \#(\mathcal{F}ree)$, then repeat i times:
      $\mathsf{ID} \in_R \mathcal{F}ree$; $\mathcal{F}ree \leftarrow_- \mathsf{ID}$; $nbt++$; $\mathsf{t}_{nbt} \in_R \mathbb{Z}$;
      $\mathsf{tk}_{\mathsf{t}_{nbt}} := \mathsf{DB(ID).tk_{ID}}$; $\mathsf{Tab} \leftarrow_+ (\mathsf{t}_{nbt}, \mathsf{ID}, \textsc{Actual})$;
      return t_{nbt};
    - $*$ else return $\perp$.
  - $\mathcal{O}^{\textsc{Free}}(\mathsf{t})$:
    - $*$ $\mathsf{Tab(t)}.status := \textsc{Old}$; $\mathcal{F}ree \leftarrow_+ \mathsf{Tab(t).ID}$.
  - $\mathcal{O}^{\textsc{Corrupt}}(\mathsf{t})$:
    - $*$ $\mathcal{C}orrupt \leftarrow_+ (\mathsf{t}, \mathsf{ID})$; return $(\mathsf{Tab(t).ID}, \mathsf{tk_t}, \mathsf{k_t})$.
  - $\mathcal{O}^{\textsc{Execute}}(\mathsf{t})$:
    - $*$ $req++$; $a \in_R \{0,1\}^\lambda$; $b \in_R \{0,1\}^\lambda$; $r \in_R \mathbb{Z}_q$;
    - $*$ $T_0 := \textsc{Mac}(a||b, \mathsf{k_t})$; $T_1 := (T_0||\mathsf{tk_t}||b) \oplus \mathcal{H}(\mathsf{rpk}^r)$; $T_2 := g^r$;
    - $*$ $c := (T_1, T_2)$; return c;
    - $*$ $\mathcal{I}nst \leftarrow_+ (req, a, (1, \mathsf{t}, c, 1))$;
    - $*$ return $(req, (a, c), 1)$.
  - $\mathcal{O}^{\textsc{Launch}}$:
    - $*$ $req++$;$a \in_R \{0,1\}^\lambda$; $\mathcal{I}nst \leftarrow_+ (req, a, \perp)$;
    - $*$ return (a, req).
  - $\mathcal{O}^{\textsc{SendTag}}(a, \mathsf{t})$:
    - $*$ $b \in_R \{0,1\}^\lambda$; $r \in_R \mathbb{Z}_q$;
    - $*$ $T_0 := \textsc{Mac}(a||b, \mathsf{k_t})$; $T_1 := (T_0||\mathsf{tk_t}||b) \oplus \mathcal{H}(\mathsf{rpk}^r)$; $T_2 := g^r$;

```
    * $c := (T_1, T_2)$; return $c$;
    * if $\exists\ \pi$ such that $(\mathcal{I}nst(\pi).m_R = a)$, then:
          if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).result \neq \perp)$,
             then $\mathcal{I}nst(\pi).ssinst \leftarrow_{+} (\rho_{new}, \mathsf{t}, c, 0)$;
          else $\mathcal{I}nst(\pi).ssinst \leftarrow_{+} (\rho_{new}, \mathsf{t}, c, \perp)$;
    * else $req++$; $\mathcal{I}nst \leftarrow_{+} (req, a, (1, \mathsf{t}, c, 0))$;
• $\mathcal{O}^{\textsc{SendReader}}(c, \pi)$ :
    * if $forall\ \rho\ (\mathcal{I}nst(\pi).ssinst(\rho).result = \perp)$ then:
        - if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).m_{\mathcal{T}} = c)$, then $(\mathcal{I}nst(\pi).ssinst(\rho).result = 1)$;
        - else:
            $T_0||\mathsf{tk}||b := \mathsf{Dec}(\mathsf{rsk}, c)$;
            if $\exists\ \mathsf{ID} \in \mathsf{DB}$ such that $\mathsf{DB}(\mathsf{ID}).\mathsf{tk}_{\mathsf{ID}} = \mathsf{tk}$ and $T_0 = \textsc{Mac}(\mathcal{I}nst(\pi).m_{\mathcal{R}}||b, \mathsf{k}_{\mathsf{ID}})$, then:
               $\mathcal{I}nst(\pi).ssinst \leftarrow_{+} (\rho_{new}, \mathsf{ID}, c, 1)$;
            else $\mathcal{I}nst(\pi).ssinst \leftarrow_{+} (\rho_{new}, \perp, c, 0)$;
• $\mathcal{O}^{\textsc{Return}}(\pi)$ :
* if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).result = 1)$ then return 1;
* else
      if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).result = 0)$ then return 0;
      else return $\perp$.
• Output of the untraceability experiment:
      $(\mathsf{t}_i, \mathsf{t}_j) \leftarrow \mathcal{A}$.
```

The event S_0 is defined by: "$(\mathsf{t}_i, \mathsf{t}_j)$ is a non-obvious link". We thus obviously have:

$$Succ^{\mathsf{Unt}}_{\mathcal{S},\mathcal{A}}(1^\lambda) = Pr[S_0]$$

– **Game 1:** in this game, we first use the random oracle, denoted $\mathcal{O}^{\mathrm{RO}}$, to replace outputs of the hash function by random values. To avoid mistakes produced by several requests to $\mathcal{O}^{\mathsf{RO}}$ on the same input, this procedure stores couple $(value, output)$ in a hash table (initially empty) and looks for each request if this one has already been queried. If this is the case, $\mathcal{O}^{\mathsf{RO}}$ outputs the corresponding value, else it selects a new random value, stores the new couple in the table, and outputs this random value. As a consequence, we also modify the $\mathcal{O}^{\textsc{SendReader}}$ oracle which first test all entries h outputted by the random oracle. This does not modify the result of the game but prepare the next modifications.

```
• $\mathcal{O}^{\textsc{Execute}}(\mathsf{t})$ :
    * $req++$; $a \in_R \{0,1\}^\lambda$; $b \in_R \{0,1\}^\lambda$; $r \in_R \mathbb{Z}_q$;
    * $h \leftarrow \mathcal{O}^{\mathsf{RO}}(\mathsf{rpk}^r)$;
    * $T_0 := \textsc{Mac}(a||b, \mathsf{k}_{\mathsf{t}})$; $T_1 := (T_0||\mathsf{tk}_{\mathsf{t}}||b) \oplus h$; $T_2 := g^r$;
    * $c := (T_1, T_2)$; return $c$;
    * $\mathcal{I}nst \leftarrow_{+} (req, a, (1, \mathsf{t}, c, 1))$;
    * return $(req, (a, c), 1)$.
• $\mathcal{O}^{\textsc{SendTag}}(a, \mathsf{t})$ :
    * $b \in_R \{0,1\}^\lambda$; $r \in_R \mathbb{Z}_q$;
    * $h \leftarrow \mathcal{O}^{\mathsf{RO}}(\mathsf{rpk}^r)$;
    * $T_0 := \textsc{Mac}(a||b, \mathsf{k}_{\mathsf{t}})$; $T_1 := (T_0||\mathsf{tk}_{\mathsf{t}}||b) \oplus h$; $T_2 := g^r$;
    * $c := (T_1, T_2)$; return $c$;
    * if $\exists\ \pi$ such that $(\mathcal{I}nst(\pi).m_R = a)$, then:
          if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).result \neq \perp)$,
             then $\mathcal{I}nst(\pi).ssinst \leftarrow_{+} (\rho_{new}, \mathsf{t}, c, 0)$;
          else $\mathcal{I}nst(\pi).ssinst \leftarrow_{+} (\rho_{new}, \mathsf{t}, c, \perp)$;
    * else $req++$; $\mathcal{I}nst \leftarrow_{+} (req, a, (1, \mathsf{t}, c, 0))$;
```

```
• $\mathcal{O}^{\textsc{SendReader}}(c, \pi)$ :
    ∗ if $forall\ \rho\ (\mathcal{I}nst(\pi).ssinst(\rho).result = \bot)$ then:
        - if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).m_{\mathcal{T}} = c)$, then $(\mathcal{I}nst(\pi).ssinst(\rho).result = 1)$;
        - else:
            search $h$ of $\mathcal{O}^{\mathsf{RO}}$ and $\mathsf{ID} \in \mathsf{DB}$ such that $\exists \mathsf{tk}_{\mathsf{ID}}$ verifying $T_0||\mathsf{tk}_{\mathsf{ID}}||b := \mathsf{Dec}(\mathsf{rsk}, c)$;
            if $T_0 = \textsc{Mac}(\mathcal{I}nst(\pi).m_{\mathcal{R}}||b, \mathsf{k}_{\mathsf{ID}})$, then:
                $\mathcal{I}nst(\pi).ssinst \leftarrow_+ (\rho_{new}, \mathsf{ID}, c, 1)$;
            else $\mathcal{I}nst(\pi).ssinst \leftarrow_+ (\rho_{new}, \bot, c, 0)$;
```

S_1 is defined to be the event that "$(\mathsf{t}_i, \mathsf{t}_j)$ is a non-obvious link". Considering we are in the random oracle, we thus have $Pr[S_0] = Pr[S_1]$. The only difference is that, in this game, we need in addition $q_H = \sum_{j=0}^{current} q_{E_j}$ requests to the random oracle, where q_{E_j} is the number of requests to the Execute and the SendTag oracles for the tag ID_j. We need to verify that the modification we have done do not change the $\mathcal{O}^{\textsc{Result}}$ oracle. In fact, this is only the case when the adversary sends a true random value which is accepted but this is only possible with negligible property, for obvious reasons.

- **Game 2:** in this game, we only modify the way to compute the public parameters.

```
• $\textsc{Setup}(1^\lambda)$:
    ∗ $nbt := 0$; $\mathcal{F}ree := \emptyset$
    ∗ System: $g \in_R \mathbb{Z}_p^*; \theta \in_R \mathbb{Z}_q; G := g^\theta$;
    ∗ Reader: $\mathsf{rsk} \in_R \mathbb{Z}_q$; $\mathsf{rpk} \in g^{\mathsf{rsk}}; Y := \mathsf{rpk}^\theta$;
```

The rest of the game is unchanged (except that g and rpk are changed into G and Y) and not detailed anymore here. S_2 is defined to be the event that"$(\mathsf{t}_i, \mathsf{t}_j)$ is a non-obvious link" and thus we trivially obtain $Pr[S_1] = Pr[S_2]$ because of the preservation of the relation $Y = G^{\mathsf{rsk}}$.

- **Game 3:** we now make one small change to the above game. Namely, instead of computing Y as G^{rsk}, we compute it as $\mathsf{rpk}^{\theta'}$ for randomly chosen $\theta' \in \mathbb{Z}_q$. This is a transition based on the indistinguishability of the Hashed El Gamal encryption scheme (and corresponds to the security proof for the Hashed El Gamal indistinguishability).

```
• $\textsc{Setup}(1^\lambda)$:
    ∗ $nbt := 0$; $\mathcal{F}ree := \emptyset$
    ∗ System: $g \in_R \mathbb{Z}_p^*; \theta \in_R \mathbb{Z}_q, G := g^\theta$;
    ∗ Reader: $\mathsf{rsk} \in_R \mathbb{Z}_q$; $\theta' \in_R \mathbb{Z}_q; \mathsf{rpk} \in g^{\mathsf{rsk}}; Y := \mathsf{rpk}^{\theta'}$;
```

The rest of the game is unchanged and not detailed anymore here. S_3 is defined to be the event that "$(\mathsf{t}_i, \mathsf{t}_j)$ is a non-obvious link". It is not possible to conclude by giving the exact value of $\Pr[S_3]$ but the difference between Game 2 and Game 3 is given by the following lemma.

Lemma 3. *The difference* $|\Pr[S_2] - \Pr[S_3]|$ *is equal to the DDH-advantage of a distinguisher.*

Proof. We do not give the proof for this lemma. An interested reader can refer to [31] for this proof. □

- **Game 4:** in this game, we only perform a modification during requests to $\mathcal{O}^{\textsc{Execute}}$ and $\mathcal{O}^{\textsc{SendTag}}$. We need that each output of the random oracle in these steps are uniformly distributed. For this, each of the corresponding input must be different from all previous ones. If this is not the case, we stop the experiment. In order to represent this, we create a second procedure, denoted $\mathcal{O}^{\mathsf{RO}'}$ which stops the game if one input has already been queried. We use this new procedure only in the last step.

```
• $\mathcal{O}^{\textsc{Execute}}(\mathsf{t})$ :
    ∗ $req++$; $a \in_R \{0,1\}^\lambda$; $b \in_R \{0,1\}^\lambda$; $r \in_R \mathbb{Z}_q$;
    ∗ $h \leftarrow \mathcal{O}^{\mathsf{RO}}(\mathsf{rpk}^r)$ or STOP;
    ∗ $T_0 := \textsc{Mac}(a||b, \mathsf{k}_\mathsf{t})$; $T_1 := (T_0||\mathsf{tk}_\mathsf{t}||b) \oplus h$; $T_2 := g^r$;
    * $c := (T_1, T_2)$; return $c$;
    * $\mathcal{I}nst \leftarrow_+ (req, a, (1, \mathsf{t}, c, 1))$;
    * return $(req, (a, c), 1)$.
  • $\mathcal{O}^{\text{SendTag}}(a, \mathsf{t})$ :
    * $b \in_R \{0,1\}^\lambda$; $r \in_R \mathbb{Z}_q$;
    * $h \leftarrow \mathcal{O}^{\mathsf{RO}}(\mathsf{rpk}^r)$ or STOP;
    * $T_0 := \text{Mac}(a||b, \mathsf{k_t})$; $T_1 := (T_0||\mathsf{tk_t}||b) \oplus h$; $T_2 := g^r$;
    * $c := (T_1, T_2)$; return $c$;
    * if $\exists\ \pi$ such that $(\mathcal{I}nst(\pi).m_R = a)$, then:
          if $\exists\ \rho$ such that $(\mathcal{I}nst(\pi).ssinst(\rho).result \neq \perp)$,
             then $\mathcal{I}nst(\pi).ssinst \leftarrow_+ (\rho_{new}, \mathsf{t}, c, 0)$;
          else $\mathcal{I}nst(\pi).ssinst \leftarrow_+ (\rho_{new}, \mathsf{t}, c, \perp)$;
    * else $req++$; $\mathcal{I}nst \leftarrow_+ (req, a, (1, \mathsf{t}, c, 0))$;
```

S_4 is defined to be the event that "$(\mathsf{t}_i, \mathsf{t}_j)$ is a non-obvious link". We denote by F the event STOP. It is obvious that S_4 is equivalent to S_3 if F does not occur. Consequently, using result of [31] we have:

$$|Pr[S_3] - Pr[S_4]| \leq Pr[F]$$

The adversary has no control on the random values ω and thus, the values Y^ω are also considered as randomly chosen in $\mathbb{Z}_p^*$. Using the birthday paradox, we obtain that the probability that F occurs is:

$$Pr[F] = 1 - \frac{|\mathbb{Z}_p^*|!}{(|\mathbb{Z}_p^*| - q_E)!} \frac{1}{|\mathbb{Z}_p^*|^{q_E}} \leq 1 - \frac{(p-1)!}{(p-1-q_E)!(p-1)^{q_E}}$$

where $q_E = \sum_j q_{E_j}$.

- **Game 5:** in this game, each pseudonym obtains a true random secret key sk for the MAC scheme and a random key k used in the Hash El Gamal encryption, instead of the ones of a true tag.

```
  • $\mathcal{O}^{\text{Draw}}(i)$ ($i$ times):
    * if $i \leq \#(\mathcal{F}ree)$, then repeat $i$ times:
          $\mathsf{ID} \in_R \mathcal{F}ree$; $\mathcal{F}ree \leftarrow_- \mathsf{ID}$; $nbt++$; $\mathsf{t}_{nbt} \in_R \mathbb{Z}$;
          $\mathsf{tk}_{\mathsf{t}_{nbt}} \in_R \{0,1\}^{\lambda'}$; $\mathsf{k}_{\mathsf{t}_{nbt}} \in_R \{0,1\}^l$; $\mathsf{Tab} \leftarrow_+ (\mathsf{t}_{nbt}, \mathsf{ID}, \text{Actual})$;
          return $\mathsf{t}_{nbt}$;
    * else return $\perp$.
```

S_5 is the event that "$(\mathsf{t}_i, \mathsf{t}_j)$ is a non-obvious link". The adversary has no way to detect if the keys refer to true tags or not and we thus have that $Pr[S_5] = Pr[S_4]$. In conclusion, the adversary in this game can not obtain any information about a possible relation between the different Ident protocols she has seen as it is a One-Time Pad, which is unconditionally secure. As a consequence, it is obvious that the adversary in this game is the dummy adversary, and consequently:

$$Succ^{\mathsf{Unt}}_{\mathcal{S},\mathcal{A}_d}(1^\lambda) = Pr[S_5]$$

- **Conclusion**:

Finally, we obtain the following advantage for the adversary against the untraceability of our scheme:

$$\begin{aligned}\left|Succ^{\mathsf{Unt}}_{\mathcal{S},\mathcal{A}}(1^\lambda) - Succ^{\mathsf{Unt}}_{\mathcal{S},\mathcal{A}_d}(1^\lambda)\right| &= |Pr[S_0] - Pr[S_5]| \\ &= |Pr[S_2] - Pr[S_3] + Pr[S_3] - Pr[S_4]| \\ &\leq |Pr[S_2] - Pr[S_3]| + |Pr[S_3] - Pr[S_4]| \\ &\leq Adv^{\mathsf{DDH}}_{\mathcal{D}} + 1 - \frac{(p-1)!}{(p-1-q_E)!(p-1)^{q_E}}\end{aligned}$$

As a conclusion, in the random oracle model, under the DDH assumption, the advantage of an adversary against the privacy property of our scheme is negligible, which concludes the proof. □

12 Efficiency and Security Comparison

In this section, we finally make a global comparison of all the schemes described in this paper, regarding security and efficiency.

12.1 Security Comparison

We have presented several solutions, based on different encryption schemes. We now make a summary of all we have obtained in Figure 10. As a first conclusion, we notice that there are currently two practical schemes which are interesting from the security point of view, as they verify all expected properties, namely the DHAES based scheme from Section 9.3 and our new Hash El Gamal based scheme described in Section 11.3.

Scheme	soundness	forward privacy	narrow-strong privacy	destructive privacy	untraceability	future untraceability
IND-CCA (generic+DHAES)	Y	Y	Y	Y	Y	Y
IND-CPA (Hash El Gamal)	N	N	N	N	N	N
CF-NM + IND-CPA (generic)	Y	N	Y	N	Y	N
IND-CPA + MAC (generic)	Y	Y	Y	Y	Y	N
IND-CPA + MAC (Hash El Gamal)	Y	Y	Y	Y	Y	Y

Fig. 10. Security comparison

12.2 Efficiency Comparison

It is notoriously difficult to make implementation estimates without going through the implementation process and so, by necessity, our estimates offer a rough guide only. In particular, since there are so many implementation variables (space, power, speed...) and so we have concentrated our efforts on getting an estimate for the space required, using as our data-points established reference points in the literature. Of course power consumption and timing are vital considerations, however our goal has been to give a first-order comparison between the schemes described in this paper. Throughout, we will use *gate equivalents* (GEs) as the unit of comparison. We're aiming for an 80-bit security level which is typically of interest and we will use approximately 160-bit elliptic curves.

The case of DHAES. To reach our security model we choose the parameters $\mathsf{tk}_{\mathsf{ID}}$, a, k_1 and k_2 to all be 80-bits in length. We might consider using coupons and pre-computing a set of 320-bit valid coupons of the form $(T_0, \mathsf{k}_1||\mathsf{k}_2)$ where $T_0 = g^r$ and $\mathsf{k}_1||\mathsf{k}_2 = \mathcal{H}(T_0||\mathsf{epk}^r)$. These would be stored on the tag.

In terms of computational operations, the tag computes SymEnc over a 160-bit input as well as a MAC with a 160-bit input.

An efficient option would probably be to build the symmetric primitives out of a block cipher. One could use AES for SymEnc and a corresponding MAC-construction which could all be done for around 3600 GE [18], though some significant overheads to deal with different modes should be anticipated. A more lightweight possibility would be to use Present [9] to construct both SymEnc and the corresponding MAC. A range of implementations suggests that 1500 GE would be a good estimate for the basic core, with a range of overheads suggesting that 2000-3000 GE could be enough. Finally the last possibility is to store the 160-bit key k_3 generated by a pseudo random generator and k_2 and to don't store k_2 in the tag as a coupon. This means using 400-bit coupons $(T_0, \mathsf{k}_1 || \mathsf{k}_3)$. As the exclusive-or on the tag of two 160-bit numbers requires around 400 GE, this increases slightly the number of gates but requires half less PRESENT computations so it appears as the most efficient in term of implementation.

The case of WIPR. In [27], Oren and Feldhofer propose a hardware implementation of WIPR and obtain a total chip area of 5705 GEs. Note that this implementation does not use elliptic curves and coupons, and so this offers some additional storage and usage advantages over the schemes that do.

The case of El Gamal. As in the case of DHAES, it is interesting to consider the use of coupons. In this scheme the 320-bit coupons are of the form $(\mathsf{epk}^r, T_2 = g^r)$. However even though we use coupons, the computation that remains on the tag is an elliptic curve addition. Depending on the elliptic curve and the underlying field arithmetic, there are a vast range of different elliptic curve implementations available. The most striking are those of Batina *et al* [5] where we might expect an elliptic curve addition to take a few thousand GEs.

The case of Hash El Gamal. Again, coupons are likely to make the most efficient implementations. In this scheme, the 480-bit coupons are of the form $(k, \mathcal{H}(\mathsf{epk}^r), T_1 = g^r)$. It is possible to generalize the scheme by replacing the computation of T_0 via the exclusive-or to encryption using any symmetric scheme. However, the use of the exclusive-or would perhaps offer the best implementation opportunities. In this case in term of implementation the situation is like the last possibility for DHAES with the difference than the tag has to store bigger coupons and to perform an exclusive-or between two 240-bit numbers instead of two 160-bit numbers so it requires approximately 200 GE more.

Summary. While coupons carry a storage and usage cost, they are often the best technique available to make a serious reduction in the cost of an on-tag RFID computation. With these in place, most of the rest of the functionality can be provided using lightweight primitives such as PRESENT. This tend to all lead to roughly the same space cost for the cryptographic operations (except for the case of El Gamal) with a slightly edge for DHAES.

Used encryption scheme	Size implementation [GE]	Time [ms]	Size of coupons
WIPR [26] (Section 10.2)	5706	≈ 600	0
WIPR-SAEP [26, 35] (Section 10.2)	8035	?	0
El Gamal (Section 10.3)	< 8000	≈ 1000	320
DHAES (Section 9.3)	< 3000	< 100	400
Hash El Gamal with coupons (Section 11.3)	< 3000	< 100	400
Hash El Gamal w/o coupons (Section 11.3)	≈ 18000	≈ 300	0

Fig. 11. Estimate performances

Table 11 shows an estimate on the performances of the instantiations given in this paper. It is obvious that in terms of security and efficiency, both the DHAES scheme and our main Hash El Gamal based scheme are the most interesting ones. However, in terms of time execution, our main Hash El Gamal based scheme seems better since the generation of the key k can be pre-computed while the execution of the hash function cannot.

Nevertheless, we have proved in this paper that this is possible to reach the highest security level for an RFID authentication scheme from an IND-CPA encryption scheme, with correct performances (but with coupons). Then, it may be possible to develop really efficient schemes by using our constructions.

13 Conclusion

In this paper, we have introduced a new security model for privacy-preserving authentication/identification RFID schemes. We have also proved that the strongest property of our model is stronger than those (achievable) of Vaudenay's model. Although it seems hard to prove that our future-untraceability implies the Vaudenay's destructive privacy, we have proved that the opposite is false. One remaining interesting goal is to prove that future-untraceability implies the narrow-strong privacy.

Regarding efficiency, we have proposed several concrete constructions with the conclusion that this is today possible to efficiently reach the best possible security level. It may be interesting now to make concrete implementations of such schemes to really know which one (between the DHAES and the Hash El Gamal based) is the most interesting one. It seems also important to study efficient solutions without the use of the coupon technique.

References

1. M. Abdalla, M. Bellare, and P. Rogaway. DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem. Technical report, UC Davis Computer Science, 1998.
2. Gildas Avoine. Adversarial model for radio frequency identification. Cryptology ePrint Archive, Report 2005/049, 2005.
3. Gildas Avoine, Christian Floerkemeier, and Benjamin Martin. RFID Distance Bounding Multistate Enhancement. In *Proceedings of the 10th International Conference on Cryptology in India – Indocrypt 2009*, New Delhi, India, December 2009.
4. Gildas Avoine and Philippe Oechslin. A scalable and provably secure hash based RFID protocol. In *PerSec 2005*. IEEE Computer Society Press, 2005.
5. Lejla Batina, Jorge Guajardo, Tim Kerins, Nele Mentens, Pim Tuyls, and Ingrid Verbauwhede. An Elliptic Curve Processor Suitable for RFID-Tags. In *IACR eprint. Available at http://eprint.iacr.org/2006/227*, 2006.
6. Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2004.
7. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, 1998.
8. Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In *ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2000.
9. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In *CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
10. Xavier Boyen and L. Martin. Identity-Based Cryptography Standard (IBCS) #1. In *Request for Comments: 5091*. IETF, 2007.
11. Sébastien Canard and Iwen Coisel. Data synchronization in privacy-preserving rfid authentication schemes. In *Proceedings of RFIDSec'08*, 2008.
12. Sébastien Canard, Iwen Coisel, and Jonathan Etrog. Lighten Encryption Schemes for Secure and Private RFID Systems. In *1st International Workshop on Lightweight Cryptography for Resource-Constrained Devices – WLC'10*, Lecture Notes in Computer Science, Tenerife, Canary Islands, Spain, January 2010. Springer.
13. Sébastien Canard, Iwen Coisel, and Marc Girault. Security of privacy-preserving rfid systems. In *Proc. IEEE International Conference on RFID-Technology and Applications (RFID-TA 2010)*, pages 269–274, 2010.
14. Benoît Chevallier-Mames, Pascal Paillier, and David Pointcheval. Encoding-free elgamal encryption without random oracles. In *Public Key Cryptography, PKC 2006*, volume 3958 of *Lecture Notes in Computer Science*, pages 91–104. Springer, 2006.
15. Ivan Damgård and Michael Østergaard Pedersen. Rfid security: Tradeoffs between security and efficiency. In *CT-RSA 2008*, volume 4964 of *Lecture Notes in Computer Science*, pages 318–332. Springer, 2008.
16. Robert H. Deng, Yingjiu Li, Andrew C. Yao, Moti Yung, and Yunlei Zhao. A New Framework for RFID Privacy. Cryptology ePrint Archive, Report 2010/059, 2010.

17. Tassos Dimitriou. A lightweight RFID protocol to protect against traceability and cloning attacks. In *SecureComm 2005*. IEEE Computer Society Press, 2005.
18. Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong authentication for rfid systems using the aes algorithm. In *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370. Springer, 2004.
19. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.
20. Gerhard Hancke and Markus Kuhn. An RFID Distance Bounding Protocol. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*, pages 67–73, Athens, Greece, September 2005. IEEE, IEEE Computer Society.
21. Ari Juels and Stephen A. Weis. Defining strong privacy for rfid. In *PERCOM'07*, pages 342–347, Washington, DC, USA, 2007. IEEE Computer Society.
22. Tri Van Le, Mike Burmester, and Breno de Medeiros. Universally composable and forward-secure rfid authentication and authenticated key exchange. In *ASIACCS 2007*, pages 242–252. ACM, 2007.
23. Jorge Munilla, Andres Ortiz, and Alberto Peinado. Distance Bounding Protocols with Void-Challenges for RFID. In *Workshop on RFID Security – RFIDSec'06*, Graz, Austria, July 2006. Ecrypt.
24. Jorge Munilla and Alberto Peinado. Distance Bounding Protocols for RFID Enhanced by using Void-Challenges and Analysis in Noisy Channels. *Wireless Communications and Mobile Computing*, 8(9):1227–1232, January 2008.
25. Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic approach to "privacy-friendly" tags. In *RFID Privacy Workshop 2003*, 2003.
26. Yossef Oren and Martin Feldhofer. WIPR - public key identification on two grains and sans. In *RFID Sec 2008*. Manfred Aigner editor, 2008.
27. Yossef Oren and Martin Feldhofer. Wipr - public key identification on two grains of sand. In *Proceedings of RFIDSec'08*, 2008.
28. Khaled Ouafi and Raphael C.-W. Phan. Traceable privacy of recent provably-secure rfid protocols. In *ACNS 2008*, volume 5037 of *Lecture Notes in Computer Science*, pages 479–489, 2008.
29. Michael O. Rabin. Digitalized Signatures and Public-Key Functions as Intractable as Factorization. In *MIT Laboratory for Computer Science*. MIT, 1979.
30. Adi Shamir. Squash - a new mac with provable security properties for highly constrained devices such as rfid tags. In *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 2008.
31. V. Shoup. Sequences of games: a tool for taming complexity in security proofs, 2004.
32. Serge Vaudenay. On privacy models for rfid. In *ASIACRYPT 2007*, pages 68–87, 2007.
33. Stephen Weis, Sanjay Sarma, Ronald Rivest, and Daniel Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *SPC 2003*, volume 2802 of *Lecture Notes in Computer Science*, pages 454–469. Springer-Verlag, 2003.
34. Jiang Wu and Doug Stinson. How to Improve Security and Reduce Hardware Demands of the WIPR RFID Protocol. In *IEEE International Conference on RFID – RFID 2009*, Orlando, Florida, USA, April 2009.
35. Jiang Wu and Doug Stinson. How to improve security and reduce hardware demands of the WIPR RFID protocol. In *IEEE RFID 2009*. IEEE Computer Society, 2009.
36. Ching Yu Ng, Willy Susilo, Yi Mu, and Rei Safavi-Naini. Practical RFID Ownership Transfer Scheme. In *Workshop on RFID Security – RFIDSec Asia'10*, volume 4 of *Cryptology and Information Security*, Singapore, Republic of Singapore, February 2010. IOS Press.