A New RFID Privacy Model*

Jens Hermans**, Andreas Pashalidis, Frederik Vercauteren***,
and Bart Preneel

Department of Electrical Engineering - COSIC,
Katholieke Universiteit Leuven and IBBT,
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
firstname.lastname@esat.kuleuven.be

Abstract. This paper critically examines some recently proposed RFID privacy models. It shows that some models suffer from weaknesses such as insufficient generality and unrealistic assumptions regarding the adversary's ability to corrupt tags. We propose a new RFID privacy model that is based on the notion of indistinguishability and that does not suffer from the identified drawbacks. We demonstrate the easy applicability of our model by applying it to multiple existing RFID protocols.

Keywords: RFID, authentication, identification, privacy model.

1 Introduction

As Radio Frequency Identification (RFID) systems are becoming more common (for example in access control [10,30], product tracking [10], e-ticketing [27,30], electronic passports [18]), managing the associated privacy and security concerns becomes more important [34]. Since RFID tags are primarily used for authentication purposes, 'security' in this context means that it should be infeasible to 'fake' a legitimate tag. 'Privacy', on the other hand, means that adversaries should not be able to identify, trace, or link tag appearances.

Several models for privacy and security in the context of RFID systems have been proposed in the literature. In this paper, we critically examine some of these models. In particular, we focus on *general* models[1]. For some of these models we show that, despite their intended generality, it remains unclear how to apply

* This work was supported in part by (a) the Research Council K.U.Leuven: GOA TENSE (GOA/11/007), (b) the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy), (c) the 'Trusted Architecture for Securely Shared Services' (TAS3) project, supported by the 7th European Framework Programme with contract number 216287, and (d) the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II.

** Research assistant, sponsored by the Fund for Scientific Research - Flanders (FWO).

*** Postdoctoral Fellow of the Fund for Scientific Research - Flanders (FWO).

[1] We do not discuss some of the early proposals that were made in the context of one specific protocol.

V. Atluri and C. Diaz (Eds.): ESORICS 2011, LNCS 6879, pp. 568–587, 2011.

them to protocols other than the protocol in the context of which they were proposed. Other existing models do not support adversaries that can tamper with tags. However, considering such adversaries is important because, as low-cost devices, tags are hardly protected against physical tampering. In particular, it has been shown that side-channel attacks may enable an adversary to extract secrets from the tag [17, 21, 22, 26], and so-called 'reset' attacks force the tag to re-use old randomness [3, 9, 15]. The adversary can mount reset attacks by inducing power drops or by otherwise influencing the physical environment of the tag. Adversaries that can tamper with tags are therefore realistic.

Subsequently we propose a new model that borrows concepts from previous models, including virtual tag references, the corruption model that Vaudenay [32] introduced and the notion of 'narrow' and 'wide' adversaries to construct a new model. We believe that the new model is easier to apply. Also note that, although presented as a model for RFID privacy, it is not limited to the RFID setting; the model may also apply to other setups, in which the participants should not be identifiable or linkable.

Structure of the paper. Section 2 introduces the basic definitions for RFID systems and some notation. Section 3 discusses a selection of existing models, their underlying assumptions, their usability, and some further technicalities. Section 4 presents our model for RFID privacy which is then applied to some of the stronger existing RFID protocols in Section 5. In the appendices, our model is extended to a multi-indistinguishability setup, which allows multi-bit challenges. Mutual authentication is also discussed there.

2 Definitions

Throughout this paper we use a common model for RFID systems, similar to the definitions introduced in [8, 32]. An RFID system consists of a set of tags $\mathcal{T}$, and a reader R. Each tag is identified by an identifier ID. The memory of the tags contains a state S, which may change during the lifetime of the tag. The tag's ID may or may not be stored in S. Each tag is a transponder with limited memory and computation capability.

Tags can also be corrupted: the adversary has the capability to extract secrets and other parts of the internal state from the tags it chooses. The reader R consists of one or more transceivers and a central database. The reader's task is to identify legitimate tags (i.e. to recover their IDs), and to reject all other incoming communication. The reader has a database that contains for every tag, its ID and a matching secret K.

Definition 1 (RFID Framework [32]). *An RFID scheme consists of the following algorithms:*

- *SetupReader*(1^k)*: setup the reader by generating the necessary keys, depending on the security parameter k. The function returns the public and private keys of the reader. Public keys are assumed to be publicly released by the algorithm, private keys are stored in the reader.*

- *SetupTag(ID): return the tag specific secret K and the initial state S of the tag. The pair (ID, K) will be stored in the reader, the state S in the tag. Note that K is not necessarily stored in the tag, but the definition of the protocol might include K in the state S.*
- *Protocol: a polynomial-time interactive protocol between a reader and a tag. The reader ends with a tape output.*

All the models discussed below fit the above general RFID system definition.

A function $f : \mathbb{N} \to \mathbb{R}$ is called 'polynomial' in the security parameter $k \in \mathbb{N}$ if $f(k) = O(k^n)$, with $n \in \mathbb{N}$. It is called 'negligible' if, for every $c \in \mathbb{N}$ there exists an integer k_c such that $f(k) \leq k^{-c}$ for all $k > k_c$. We denote a negligible function by ϵ.

If $\mathcal{T}$ is a set, $t \in_R \mathcal{T}$ means that t is chosen uniformly at random from $\mathcal{T}$. $|\mathcal{T}|$ denotes the cardinality of the set. If $\mathcal{A}$ is an algorithm, then $\mathcal{A}^{\mathcal{O}}$ denotes the fact that $\mathcal{A}$ has access to the oracle $\mathcal{O}$.

3 Existing Privacy Models

This section discusses certain existing RFID privacy models. Most models feature a correctness (no false negatives), security (no false positives) and privacy definition.

Note that covering all existing models would exceed the scope of this paper by far. Many models, including the ones introduced in [2, 7, 11, 14, 16, 20, 31] do not allow corrupted tags to be traced. We have selected two such models [14, 20] for further discussion, in addition to the stronger models of Vaudenay [32] and Canard et al. [8].

3.1 Vaudenay

Several concepts from the privacy model introduced by Vaudenay [32] are used in our model. We therefore present this in detail.

Adversarial model. The adversary of the Vaudenay model has the ability to influence all communication between a tag and the reader and can therefore perform man-in-the-middle attacks on any tag that is within its range. It may also obtain the result of the authentication of a tag, i.e. whether the reader accepts or rejects the tag. The adversary may also 'draw' (at random) tags and then 'free' them again, moving them inside and outside its range. During these interactions the adversary has to use a virtual identifier (not the tag's real ID) in order to refer to the tags that are inside its range. Finally the adversary may corrupt tags, thereby learning their entire internal state.

The above interactions take place over eight oracles that the adversary may invoke: `CreateTag(ID)`, `DrawTag(distr)` $\rightarrow$ $(vtag)$, `Free(`$vtag$`)`, `Launch` $\rightarrow$ π, `SendReader(`m, π`)` $\rightarrow$ m', `SendTag(`$m, vtag$`)` $\rightarrow$ m', `Result(`π`)` $\rightarrow$ x and `Corrupt(`$vtag$`)`. $vtag$ denotes a virtual tag reference, π a protocol instance, *distr*

a polynomially bounded sampling algorithm, m and m' messages and ID a tag ID. For a complete definition of the oracles the reader is referred to [32].

The Vaudenay model divides adversaries into different classes, depending on restrictions regarding their use of the above the oracles. In particular, a *strong* adversary may use all eight oracles without any restrictions. A *destructive* adversary is not allowed to use a tag after it has been corrupted. This models situations where corrupting a tag leads to the destruction of the tag. A *forward* adversary can only do other corruptions after the first corruption. That is, no protocol interactions are allowed after the first corrupt. A *weak* adversary does not have the ability to corrupt tags. Orthogonal to these four attacker classes there is the notion of *wide* and *narrow* adversary. A *wide* adversary has access to the result of the verification by the server while a *narrow* adversary does not.

Due to their generality, the above restrictions can be used perfectly in other privacy models. Throughout the paper we will frequently refer to strong, destructive, forward, weak and wide/narrow adversaries.

The equations below show the most important relations between the above privacy notions:

$$\begin{array}{ccccccc} \text{Wide Strong} & \Rightarrow & \text{Wide Destructive} & \Rightarrow & \text{Wide Forward} & \Rightarrow & \text{Wide Weak} \\ \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow \\ \text{Narrow Strong} & \Rightarrow & \text{Narrow Destructive} & \Rightarrow & \text{Narrow Forward} & \Rightarrow & \text{Narrow Weak} \end{array}$$

In this case $A \Rightarrow B$ means that if the protocol is A-private it implies that the protocol is B-private. A protocol that is *Wide Strong* private, for example, obviously also belongs to all other privacy classes, that only allow weaker adversaries.

Privacy, security and correctness. In general, an RFID protocol should satisfy (a) correctness (a 'real' tag is always accepted), (b) security (fake tags are rejected) and (c) privacy (tags cannot be identified or traced). Privacy is defined by means of the notion of a 'trivial' adversary. Intuitively, a trivial adversary does not 'use' the communication captured during the protocol run to determine its output.

Definition 2 (Blinder, trivial adversary - Simplified version of Definition 7 from [32]). *A Blinder $\mathcal{B}$ for an adversary $\mathcal{A}$ is a polynomial-time algorithm which sees the messages that $\mathcal{A}$ sends and receives, and simulates the* `Launch`, `SendReader`, `SendTag` *and* `Result` *oracles to $\mathcal{A}$. The blinder does not have access to the reader tapes. A blinded adversary $\mathcal{A}^{\mathcal{B}}$ is an adversary who does not use the* `Launch`, `SendReader`, `SendTag` *and* `Result` *oracles.*

An adversary $\mathcal{A}$ is trivial if there exists a blinder $\mathcal{B}$ such that $|\Pr(\mathcal{A}\, wins) - \Pr(\mathcal{A}^{\mathcal{B}}\, wins)|$ *is negligible.*

Intuitively, an adversary is called trivial if, even when blinded, it still produces the same output. Such an adversary does not 'use' the communication captured during the protocol run in order to determine its output. Note that a blinded adversary is not the same as a simulator typically found in security proofs: the

blinder is separate from the adversary and has no access to the adversary's tape. The blinder just receives incoming queries from the adversary and has to respond either by itself or by forwarding the queries to the system.

We are now ready to present the privacy definition.

Definition 3 (Privacy - Simplified version of Definition 6 from [32]). *The privacy game between the challenger and the adversary consists of two phases:*

1. *Attack phase: the adversary issues oracle queries according to applicable restrictions*
2. *Analysis phase: the adversary receives the table that maps every vtag to a real tag ID. Then it outputs* `true` *or* `false`*.*

The adversary wins if it outputs `true`*. A protocol is called P-private, where P is an adversary class (strong, destructive, ...), if and only if all winning adversaries that belong to the class P are trivial.*

Besides privacy the protocol should also offer authentication of the tag. We refer to this property as the *security* of the protocol.

Definition 4 (Security - Simplified version of Definition 4 from [32]). *We consider any adversary in the class* strong*. The adversary wins if the reader identifies an uncorrupted legitimate tag, but the tag and the reader did not have a matching conversation. The RFID scheme is called secure if the success probability of any such adversary is negligible.*

Definition 5 (Correctness - Definition 1 from [32]). *An RFID scheme is correct if its output is correct except with negligible probability for any polynomial-time experiment which can be described as follows:*

1. *set up the reader*
2. *create a number of tags including a subject one named ID*
3. *execute a complete protocol between reader and tag ID*

The output is correct if and only if $Output = \perp$ *and tag ID is not legitimate or* $Output = ID$ *and tag ID is legitimate.*

In a follow-up paper [25] to the Vaudenay paper, the concept of mutual authentication for RFID is defined. The tag simply outputs a boolean, indicating whether or not the reader was accepted. The authors extend the security definition by adding a criterion for reader authentication.

Discussion. The paper of Vaudenay inspired many authors to formulate derived RFID privacy models or to evaluate the (Paise-)Vaudenay model [6, 8, 12, 13, 23, 24, 25, 28, 29]. Although Vaudenay's privacy model is perhaps the strongest and most complete, it contains some flaws with respect to strong privacy.

Vaudenay's proof of the statement that 'strong privacy is impossible' uncovers some of these flaws. This proof assumes a destructive private protocol. By

definition, for every destructive adversary, there exists a blinder. This includes the adversary that (a) creates one real tag, (b) corrupts this tag right away, (c) starts a protocol using either the state from the corrupted tag or from another fake tag. In the end, the blinder has to answer the `Result` oracle. Obviously, the adversary knows which tag was selected and knows which result to expect. However, since the blinder has no access to this random coin of the adversary, it must be able to distinguish a real and a fake tag just by looking at the protocol run from the side of the reader. The proof then uses this blinder to construct a strong adversary. Since all strong adversaries are also destructive, this proves the impossibility of strong privacy.

Obviously, this proof only works because the blinder is separated from the adversary. In later work [33], Vaudenay corrects the inconsistency in the model and shows that strong privacy is indeed possible. In this new approach, the blinder is given access to the random coin flips of the adversary. The issue with a separate blinder is exploited multiple times by Armknecht et al. in [1]. Using this property the authors show the impossibility of reader authentication combined with respectively narrow forward privacy (if `Corrupt` reveals the temporary state of tags) and narrow strong privacy (if `Corrupt` only reveals the permanent state of tags).

Independent from this correction, Ng et al. [23] also identified the problems with strong privacy. They propose a solution, based on the concept of a 'wise' adversary that does not make any 'irrelevant' queries to the oracles i.e. queries to which it already knows the answer. The authors claim that, if the protocol does not generate false negatives, then a wise adversary never calls the `Result` oracle. Given the vague definition of wise adversaries it is hard to verify these claims. The existence of attacks which exploit false positives [4] however, suggests that the general claim that `Result` is not used by a wise adversary is incorrect. Based on this questionable general claim, the authors further identify an IND-CPA-based protocol as being strong private, without giving a formal proof.[2]

3.2 Canard et al.

Model. The model of Canard et al. [8] builds on the work of Vaudenay, so the definition of oracles is quite similar. For the privacy definition the model requires the adversary to produce a non-obvious link between virtual tags.

Definition 6. *$(vtag_i, vtag_j)$ is a non-obvious link if $vtag_i$ and $vtag_j$ refer to the same ID and if a 'dummy' adversary, who only has access to `CreateTag`, `Draw`, `Free`, `Corrupt`, is not able to output this link with a probability better than $1/2$.*[3]

[2] Note that the original security proof (i.e. no false positives) by Vaudenay requires IND-CCA2 encryption, so using only IND-CPA encryption would require a new security proof. The `Result` may therefore serve as a decryption oracle.

[3] It is unclear why the authors use the probability threshold $1/2$, since one would expect some dependency on the total number of non-obvious links. One slightly different interpretation is that a 'dummy' adversary cannot determine if a given non-obvious candidate link $vtag_i$, $vtag_j$ is a link in reality or not.

One major difference with respect to Vaudenay's model is that a 'dummy' adversary is used instead of a blinded adversary. This avoids some of the issues surrounding the use of a blinder, because a 'dummy' adversary can also access its own random tape, while a blinder cannot access the adversary's random tape.

The definition requires the adversary to output a non-obvious link. A protocol is said to be untraceable if, for every adversary $\mathcal{A}$, it is possible to construct a 'dummy' adversary $\mathcal{A}_d$ such that $|\mathbf{Succ}_{\mathcal{A}}^{Unt}(1^k) - \mathbf{Succ}_{\mathcal{A}_d}^{Unt}(1^k)| \leq \epsilon(k)$.

Discussion. While the work certainly has its merit in formalizing and fixing the Vaudenay model (by using a dummy adversary instead of a blinder), the model of Canard et al. lacks generality because it focuses on non-trivial links. Other relevant properties, which do not imply the leakage of a non-trivial link, are not considered a privacy breach. For example, the cardinality of the set of active tags can be leaked without leaking a non-trivial link. Because of the limited scope of untraceability, we are not using this model.

3.3 Deng, Li, Yung and Zhao

Model. Deng et al. presented their RFID Privacy Framework in [14].

The correctness ('adaptive completeness') definition used by Deng et al. is more elaborate than Vaudenay's definition. In particular, it allows the adversary to execute multiple complete protocol runs. This captures 'desynchronization' attacks where the adversary communicates a number of times with a tag (without involvement of the reader), in order to desynchronize the tag's state such that it will no longer be recognised by the reader.

The security definition considers both tag-to-reader and reader-to-tag authentication. The definition is similar to Vaudenay's since it requires matching sessions at reader and tag side. In Deng et al.'s model the last message is always sent by the reader, so an adversary could just prevent the tag from finishing the protocol by dropping this last message. Deng et al. therefore define the notion of 'matching sessions' such that last message attacks do not breach security. Vaudenay omits an exact definition of 'matching sessions', and therefore issues like the last message attack are not captured.

While the correctness and security definitions of Vaudenay and Deng et al. appear to be, to a large extent, equivalent, there is a significant discrepancy in the privacy definitions. Firstly, there is no notion of virtual tags in Deng et al.'s model; instead the adversary can refer to all tags using their real identifiers. Secondly, the adversary cannot create new tags. Thirdly, Deng et al. apply a zero-knowledge proof instead of Vaudenay's blinder construction. Informally stated, in the zero-knowledge experiment, the adversary (in the real world) consists of these phases:

1. Standard interaction using the oracles.
2. Select one tag at random (the 'challenge' tag) from the set of clean (non-corrupted and non-active) tags.

3. Interaction using the oracles, except that the adversary can only interact with the non-clean tags and the challenge tag. Moreover, the challenge tag cannot be corrupted.
4. Output a view from the previous step and the index of the challenge tag.

The simulated world is the same, except that, in the third phase, the adversary cannot access the challenge tag. If all PPT adversaries can be simulated such that the output of the adversary and simulator are computationally/statistically indistinguishable, then the protocol is considered zk-private. This implies that for all adversaries the output can actually be derived without interacting with the challenge tag (as the simulator does).

Discussion. Because of the very specific restrictions imposed in the third phase, this model is significantly weaker than Vaudenay's. Firstly, the model focuses on deriving information about a specific challenge tag (selected by the adversary), while in Vaudenay's model *any* statement that reveals information on the underlying identity of any of the tags is considered a privacy breach. Secondly, the adversary's ability to corrupt tags is limited. In Vaudenay's (corrected) strong privacy model one could prove that a protocol satisfies the privacy definition even if the 'challenge' tag is corrupted. The restriction that the challenge tag must be clean is, according to the authors, introduced to ensure that the tag is not stuck halfway a protocol run. Otherwise one can trivially distinguish the challenge tag by checking whether or not it responds to the remainder of the protocol run. Since a protocol run takes only a short timespan, obviously linking two protocol messages from the same run to the same tag should not be considered a privacy breach. However, we believe that, for the purposes of excluding this as a privacy breach, the concept of virtual tags is more suitable than overly limiting the adversary's corruption abilities in this manner.

The zero-knowledge private protocol proposed in [14] uses a counter as the tag state. The value of this counter is incremented after each protocol run completed by the tag. Obviously, this protocol does not satisfy the privacy definition if the adversary can corrupt the targeted tag, because the adversary learns the value of the counter (and the key) and, by decrementing the value of the counter, it can identify previous protocol runs of the targeted tag. The model in [14] has however been specifically tuned to disallow corruption of the challenge tag, which is a rather unrealistic assumption and thus undermines the significance of the claims that follow from its application.

The security and correctness definitions are more rigorous than Vaudenay's, so they can be a valuable alternative to them.

3.4 Juels-Weis

Model. The Juels-Weis model [20] is based on the notion of indistinguishability. The model does not feature a `DrawTag` query and the `Corrupt` query is replaced by a `SetKey` query, which returns the current secret of the tag and allows the adversary to set a new secret. Figure 3.4 shows a simplified version of the privacy

Experiment $\mathbf{Exp}^{priv}_{\mathcal{A},\mathcal{S}}$:

1. Setup:
   - Generate n random keys key_i.
   - Initialize the reader with the random key_i.
   - Create n tags, each with a key_i.
2. Phase (1): Learning
   - $\mathcal{A}$ can interact with a polynomial number of calls to the system, but can only issue `SetKey` on $n-2$ tags, leaving at least 2 uncorrupted tags
3. Phase (2): Challenge
   - $\mathcal{A}$ selects two uncorrupted tags T_0 and T_1. Both are removed from the set of tags.
   - One of these tags (T_b, the challenge tag) will be selected at random by the challenger.
   - $\mathcal{A}$ can make a polynomial number of calls to the system, but cannot corrupt the challenge tag T_b.
   - $\mathcal{A}$ outputs a guess bit $g \in \{0, 1\}$.

Fig. 1. Privacy experiment from [20]

game. The protocol is considered private if $\forall \mathcal{A}, \Pr\left[\mathbf{Exp}^{priv}_{\mathcal{A},\mathcal{S}} \text{guesses } b \text{ correctly}\right] \leq \frac{1}{2} + \epsilon$

Discussion. The Juels-Weis model is one of the few models that are based on a simple indistinguishability game instead of the notion of simulatability. The model is limited by the fact that the challenge tags cannot be corrupted. In terms of the model in [32] it would be a Weak adversary with regard to the challenge tags. For example, attacks in which the adversary links together executions of a tag that have taken place prior to its corruption are not possible in the Juels-Weis model because of this.

The model from [16] is very similar, with the difference that the privacy is defined as distinguishing the reply from a real tag from a random reply.

3.5 Bohli-Pashalidis

Model. Unlike the previous models, the Bohli-Pashalidis model [5] is not an RFID-specific model. Unfortunately, it captures only privacy properties; properties like security and correctness are not covered. The model considers a set of users (with unique identifiers) $\mathcal{U}$, whose size is at least polynomial in a security parameter. There is no formal difference between different types of player, like there is with tag and reader in most RFID models. The system $\mathcal{S}$ can be invoked with input batches $(u_1, \alpha_1), (u_2, \alpha_2), \ldots, (u_c, \alpha_c) \in (\mathcal{U}, A)^c$, consisting of pairs of user identifiers and 'parameters' and will output a batch $((e_1, \ldots e_c), \beta)$, with the outputs e_i from each system invocation and a general output β, applying to the batch as a whole. Users can also be corrupted, revealing their internal state to the adversary.

The authors investigate the properties of the function $f \in \mathcal{F}$, where $\mathcal{F} = \{f : \{1, 2, \ldots, n\} \rightarrow \mathcal{U}\}$ is the space of functions that map the serial number of each output element to the user it corresponds to. In the Strong Anonymity (SA) setting, no information should be revealed to the adversary about the function

f, guaranteeing the highest level of privacy. Several weaker notions (which reveal *some* information on f) are defined and the relations among notions are examined.

In the RFID setting the batch properties are currently not considered, although this would be an interesting extension, since some localization protocols are based on batch invocations of a large set of RFID tags. For simplicity we restrict ourselves to the Bohli-Pashalidis model for online systems. For these systems, where all batches have size one (i.e. the system never waits for multiple inputs until it produces some output), the only two applicable distinct notions are Strong Anonymity (SA) and Pseudonymity (PS).

The adversarial model is based on indistinguishability. The adversary can cause different users to invoke the system using different parameters (e.g. messages) in both a left and right world with the $\texttt{Input}((u_0, \alpha_0), (u_1, \alpha_1))$ oracle. Based on a bit b, selected by the challenger, the system will be invoked with the user-data pair (u_b, α_b). That is, the adversary itself defines the functions $f_0, f_1 \in \mathcal{F}$, for respectively the left and right world. The adversary can also corrupt users. At the end of the game the adversary has to output a guess bit g. The adversary wins the game if $g = b$. By imposing restrictions on f_0 and f_1, the authors investigate different levels of privacy.

Definition 7. *A privacy protecting system $\mathcal{S}$ is said to unconditionally provide privacy notion X, if and only if the adversary $\mathcal{A}$ is restricted to invocations (u_0, α_0) and (u_1, α_1) such that f_0 and f_1 are X-indistinguishable for all invocations and for all such adversaries $\mathcal{A}$, it holds that $\mathbf{Adv}^X_{\mathcal{S},\mathcal{A}}(k) = 0$.*

Similar definitions for computational ($\mathcal{A}$ is polytime in k and $\mathbf{Adv}^X_{\mathcal{S},\mathcal{A}}(k) \leq \epsilon(k)$) and statistical privacy are available.

Discussion. Due to its generality, and due to the fact that it is not meant to cover security properties, the Bohli-Pashalidis model needs non-trivial adaptations in order to apply to RFID setting. In its current form, the model does not support multi-pass protocols, where linking two messages from the same protocol run is not a privacy breach. Moreover there is no distinction between tags that need to be protected, and the reader for which privacy is not an issue. An interesting question is whether the strictly binary distinguishing game (only one bit of randomness in the challenge) provides enough flexibility compared to other models, like Vaudenay's, where there are multiple bits of randomness that are to be guessed.

4 Our Model

4.1 Adversarial Model and Privacy

We use the setup from Definition 1. We assume a central reader R and a set of tags $\mathcal{T} = \{T_1, T_2, \ldots, T_i\}$. $\mathcal{T}$ is initially empty, and tags are added dynamically

by the adversary. The reader maintains a database of tuples (ID_i, K_i), one for every tag $T_i \in \mathcal{T}$. Moreover, every tag T_i stores an internal state S_i.

Let $\mathcal{A}$ denote the adversary, which can adaptively control the system $\mathcal{S}$. $\mathcal{A}$ interacts with $\mathcal{S}$ through a set of oracles. The experiment that the challenger sets up for $\mathcal{A}$ (after the security parameter k is fixed) proceeds as follows:

$\mathbf{Exp}^b_{\mathcal{S},\mathcal{A}}(k)$:

1. $b \in_R \{0,1\}$
2. $SetupReader(1^k)$
3. $g \leftarrow \mathcal{A}^{\texttt{CreateTag},\texttt{Launch},\texttt{DrawTag},\texttt{Free},\texttt{SendTag},\texttt{SendReader},\texttt{Result},\texttt{Corrupt}}()$
4. Return $g == b$.

At the beginning of the experiment, the challenger picks a random bit b. The adversary $\mathcal{A}$ subsequently interacts with the challenger by means of the following oracles:

- `CreateTag(ID)` $\rightarrow T_i$: on input a tag identifier ID, this oracle calls $SetupTag(ID)$ and registers the new tag with the server. A reference T_i to the new tag is returned. Note that this does not reject duplicate IDs.
- `Launch()` $\rightarrow \pi, m$: this oracle launches a new protocol run, according to the protocol specification. It returns a session identifier π, generated by the reader, together with the first message m that the reader sends. Note that this implies that our model does not support tag-initiated protocols.
- `DrawTag(`T_i`,`T_j`)` $\rightarrow vtag$: on input a pair of tag references, this oracle generates a virtual tag reference, as a monotonic counter, $vtag$ and stores the triple $(vtag, T_i, T_j)$ in a table $\mathcal{D}$. Depending on the value of b, $vtag$ either refers to T_i or T_j. If one of the two tags T_i or T_j is already referenced in the table (i.e. is already passed to a `DrawTag` without being released with a `Free`), then this oracle returns $\perp$. Otherwise, it returns $vtag$.
- `Free(vtag)`$_b$: on input $vtag$, this oracle retrieves the triple $(vtag, T_i, T_j)$ from the table $\mathcal{D}$. If $b = 0$, it resets the tag T_i. Otherwise, it resets the tag T_j. Then it removes the entry $(vtag, T_i, T_j)$ from $\mathcal{D}$. When a tag is reset, its volatile memory is erased. The non-volatile memory, which contains the state S, is preserved.
- `SendTag(vtag,`m`)`$_b$ $\rightarrow m'$: on input $vtag$, this oracle retrieves the triple $(vtag, T_i, T_j)$ from the table $\mathcal{D}$ and sends the message m to either T_i (if $b = 0$) or T_j (if $b = 1$). It returns the reply from the tag (m'). If the above triple is not found in $\mathcal{D}$, it returns $\perp$.
- `SendReader(`π`,` m`)` $\rightarrow m'$: on input π, m this oracle sends the message m to the reader in session π and returns the reply m' from the reader (if any) is returned by the oracle.[4]
- `Result(`π`)`: on input π, this oracle returns a bit indicating whether or not the reader accepted session π as a protocol run that resulted in successful authentication of a tag. If the session with identifier π is not finished yet, or there exists no session with identifier π, $\perp$ is returned.

[4] If no active session π exists, the reader is likely to return $\perp$.

– `Corrupt`(T_i): on input a tag reference T_i, this oracle returns the complete internal state of T_i.[5] Note that the adversary is not given control over T_i.

According to the above experiment description, the challenger presents to the adversary the system where either the 'left' tags T_i (if $b = 0$) or the 'right' tags T_j (if $b = 1$) are selected when returning a virtual tag reference in `DrawTag`. The function $f_0 \in \mathcal{F}$ (where $\mathcal{F} = \{f : \{1, 2, \ldots, n\} \rightarrow \mathcal{T}\}$, see Section 3.5) maps the `DrawTag` invocations (referenced by an index k) to the tag T_i, which was passed as first argument to `DrawTag`. Similarly, f_1 maps invocation serial numbers to the second argument to `DrawTag`. f_0 and f_1 therefore describe the 'left' and the 'right' world, respectively.

$\mathcal{A}$ queries the oracles a number of times and, subsequently, outputs a guess bit g. We say that $\mathcal{A}$ wins the privacy game if and only if $g = b$, i.e. if it correctly identifies which of the worlds was active. The advantage of the adversary is defined as

$$\mathbf{Adv}_{\mathcal{S},\mathcal{A}}(k) = \left| Pr\left[\mathbf{Exp}^0_{\mathcal{S},\mathcal{A}}(k) = 1\right] + Pr\left[\mathbf{Exp}^1_{\mathcal{S},\mathcal{A}}(k) = 1\right] - 1 \right| \qquad (1)$$

4.2 Security, Correctness, Privacy

Since our model focuses on privacy, the correctness and security property are not discusses further. Both the Vaudenay and Deng et al. security and correctness definition can be used combined with the new privacy definition, without compatibility issues (also see Section 3.1 and Section 3.3).

The adversary restrictions, as defined in Section 3.1, also apply to our privacy definition. Depending on the acceptable usage of the `Corrupt` oracle, an adversary in our model is either Strong, Destructive (`Corrupt` destroys a tag), Forward (after the first `Corrupt` only further corruptions are allowed), or Weak (no `Corrupt` oracle) adversaries. Depending on the allowed usage of the `Result` oracle, there exist Narrow (no `Result` oracle) and Wide adversaries. X is used to denote one of these privacy notions.

Definition 8 (Privacy). *An RFID system $\mathcal{S}$, is said to unconditionally provide privacy notion X, if and only if for all adversaries $\mathcal{A}$ of type X, it holds that $\mathbf{Adv}^X_{\mathcal{S},\mathcal{A}}(k) = 0$. Similarly, we speak of computational privacy if for all polynomial time adversaries, $\mathbf{Adv}^X_{\mathcal{S},\mathcal{A}}(k) \leq \epsilon(k)$*

We also define X^+ privacy notion variants, where X refers to the basic privacy notion and $+$ to the notion that arises when the corruption abilities of the adversary are further restricted (see [5]). Formally, an RFID system is said to be X^+ private if it is X private and if, for all adversaries, $f_0 \approx_{\hat{T}} f_1$. Here, $f_0 \approx_{\hat{T}} f_1$ means that $\forall i$ such that $f_0(i) \in \hat{T}$ or $f_1(i) \in \hat{T}$, it holds that $f_0(i) = f_1(i)$,

[5] Both the volatile and non-volatile state is returned. For multi-pass protocols it might be necessary to relax this to only the non-volatile state; to force the adversary to only corrupt tags T_i that are currently not drawn; or to use the concept of X^+ privacy, as discussed in Section 4.3.

where $\hat{T}$ denotes the set of corrupted tags. This implies that, whenever a tag is corrupted at some point during the privacy game, it always has to be drawn simultaniously in both the left and the right world using a `DrawTag`(T_i, T_i) query with identical arguments.

4.3 Motivation and Comparison

Our proposed model is based on the well-studied notion of (left-or-right) indistinguishability. This avoids the issues with less well-studied concepts such as blinders that the Vaudenay model suffers from (see Section 3.1). Moreover, since several cryptographic schemes have proven security properties based on indistinguishability games (e.g. IND-CPA, IND-CCA, IND-CCA2...), this is likely to simplify the proofs using our model when using these schemes as building blocks.

Note that the Juels-Weis model from Section 3.4 also uses a traditional indistinguishability setup. However, the model requires the adversary to distinguish one out of two selected tags in the final phase. The disadvantage of this approach is that it does not take into account other properties that might leak privacy (e.g. cardinality) and that it limits the use of tag corruption. The Vaudenay model did introduce some crucial tools like virtual tag references and the corruption types that are still required.

Modelling details. There are certain notable differences of our model when compared to the Bohli-Pashalidis model [5] and the other models discussed in Sect. 3:

- The introduction of `CreateTag(·)`: since the set of tags is not predefined we allow the adversary to dynamically create new tags.
- `DrawTag(·,·)` and `Free(·)` are used to introduce the concept of virtual tags. This concept is needed since otherwise `SendTag(·,·)` would have to accept two tag/message pairs (and select one of them based on the value of b). In this case it would be trivial to determine the bit b for multi-pass protocols, simply by using different tags for each pass of the protocol if $b = 0$ and the same tag if $b = 1$. The protocol would only succeed if $b = 1$, thus allowing detection of b. Hence, it is crucial that the same tag is always used within a certain protocol run, which can be ensured by using virtual tag identifiers.
- `Free(·)` clears the volatile memory of tag, in order to avoid attacks that depend on leaving a tag hanging in a temporary state. Such an attack is described in [25].
- A separate communication oracle for tags and reader is used, since the reader is not considered as an entity whose privacy can be compromised.
- `Corrupt(·)`: corruption is done with respect to a tag, not a virtual tag. If `Corrupt(·)` would accept a vtag, then determining the bit b becomes trivial by performing the following attack:
  - $vtag_a \leftarrow$ `DrawTag(`T_1`,`T_2`)`
  - $C_a \leftarrow$ `Corrupt(`$vtag_a$`)`
  - `Free(`$vtag_a$`)`
  - $vtag_b \leftarrow$ `DrawTag(`T_1`,`T_3`)`
  - $C_b \leftarrow$ `Corrupt(`$vtag_b$`)`

If $C_a = C_b$ then $b = 0$, otherwise $b = 1$.
We believe that it is realistic to assume that one has the tag identifier T_i when corrupting a tag, since corruption implies having physical access to a tag.
Note that stateful protocols (which update their state after a protocol run) do not satisfy our privacy definition. By issuing a `Corrupt`(T_i) query before and after a protocol run, one can always identify whether or not the tag has been active. For such protocols, one could use the significantly weaker X^+ privacy notions.

– In the current setup `Corrupt`(T_i) reveals the full internal state of the tag, i.e. both its volatile and non-volatile parts. This follows [1] where it is shown that, if corruptions reveal the volatile state, then the resulting privacy notions are stronger. Single-pass protocols (e.g. challenge-response) do not suffer from any issues, since the volatile memory is typically erased after sending the reply, and hence all computations are confined to the invocation of the `SendTag` oracle. Multi-pass protocols on the contrary, typically require storage of data in between `SendTag` invocations. Because corruption yields the entire internal state, one could make additional assumptions on the corruption abilities of the adversary by restricting corruption to the non-volatile state. An even stronger restriction would be to allow only corruption of tags that are not drawn in either the left or right world; or use the X^+ privacy notions.

5 Evaluating Existing Protocols

This section evaluates several protocols (or classes of protocols) using our privacy model. For security and correctness results we refer to the original papers.

Several protocol 'prototypes' based on symmetric cryptography are evaluated by Ng et al. in [24] with respect to Vaudenay's privacy model. Since none of these protocols attain wide-forward privacy, we expect them to behave the same in our model. For this reason, these protocols are not discussed further.

5.1 Vaudenay's Public Key Protocol

Figure 2 shows the public key protocol presented by Vaudenay. The reader sends out a random number a and the tag encrypts this challenge, combined with the shared secret K and tag ID under the public key K_P of the reader. The reader can decrypt the tag's reply and verify the shared secret K in its database. The protocol relies on the encryption being IND-CPA to achieve narrow-strong Vaudenay-privacy and IND-CCA2 to achieve security and forward privacy. However, this protocol is wide-strong private under our model, if the underlying encryption is IND-CCA2.

Theorem 1. *If the encryption used in the protocol from Figure 2 is IND-CPA, then the protocol is strong private for narrow adversaries (i.e. adversaries that do not use the* `Result` *query).*

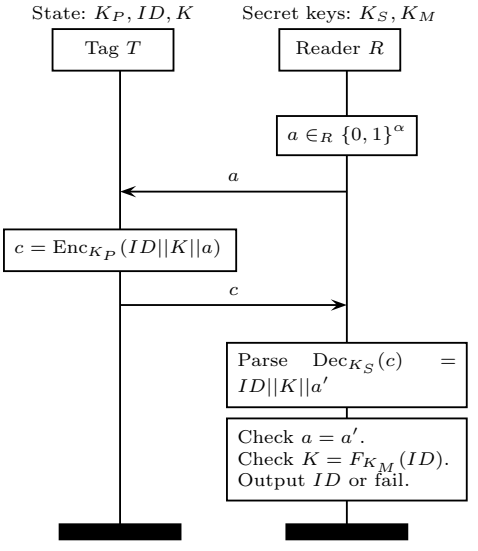

Fig. 2. Public key RFID protocol from [32]

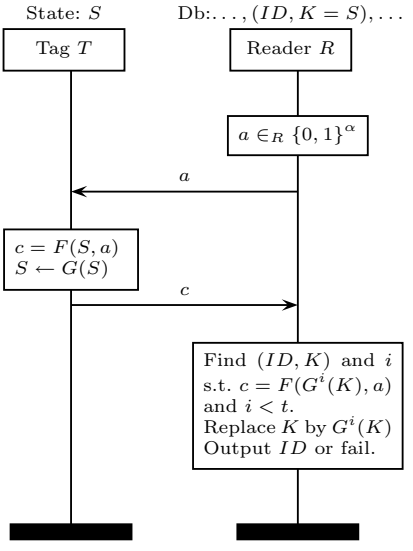

Fig. 3. RO protocol from [32]

Proof. Given an adversary $\mathcal{A}$ that wins the privacy game with non-negligible advantage, we show how to create an adversary $\mathcal{A}'$ that wins the IND-CPA game with non-negligible advantage.

The adversary $\mathcal{A}'$ runs the adversary $\mathcal{A}$ and answers all oracle queries from $\mathcal{A}$ by simply simulating the system $\mathcal{S}$, with the following exceptions:

- The public key K_P of the reader is the public key of the IND-CPA game.
- `SendTag`: retrieve the tag references T_i and T_j from the table using the virtual tag identity $vtag$. For these two tags, it generates the messages $m_0 = ID_i||K_i||a$ and $m_1 = ID_j||K_j||a$. The two messages m_0, m_1 are forwarded to the IND-CPA oracle, which returns the encryption under K_P of one of the messages.

At the end of the game $\mathcal{A}'$ outputs whatever guess $\mathcal{A}$ outputs. The privacy game is perfectly simulated for the inner adversary $\mathcal{A}$.

Assume that $\mathcal{A}$ breaks privacy, i.e. it can distinguish the left and right world, then $\mathcal{A}'$ wins the IND-CPA game. Since IND-CPA with only one call to the encryption oracle is equivalent to IND-CPA with multiple calls to the encryption oracle, this proves the (narrow) privacy of the protocol. □

The results from Lemma 8 in [32] still hold, provided the security and correctness definitions from Vaudenay are used. So, based on these results, the protocol above is also wide forward private.

Theorem 2. *If the encryption used in the protocol from Figure 2 is IND-CCA2, then the protocol is strong private for wide adversaries.*

Proof. The proof is similar to the proof for Theorem 1 above. When receiving a `Result` query, the adversary proceeds as follows. It first compares the ciphertext c to a list of outputs generated by the encryption oracle from the IND-CPA game (which are used in the `SendTag` oracle). If it matches one of these, `true` is returned. Otherwise, the result oracle forwards the ciphertext to the IND-CCA decryption oracle and receives the matching plaintext m. The plaintext is then parsed and verified, just as the reader would do. This game gives the same result as the IND-CPA game described in Theorem 1. □

5.2 RO-Based Protocol

Another (weaker) protocol from [32], shown in Figure 3, makes use of two random oracles F and G. The protocol uses an updating state S, which is shared by both tag and reader. The reader sends out a random number a and the tag computes a reply by applying F on the state S and a. The state is afterwards updated using G. Obviously, such a protocol cannot be (narrow) strong private, since the tag can trivially be traced after being corrupted.

Theorem 3. *The protocol shown in Figure 3 is narrow-destructive private.*

Proof. Assume that the challenge bit $b = 0$. We simulate the `SendTag` oracle by returning a random value c. There will never be a `SendTag` query to a corrupted tag, since tags are destroyed after corruption. This way we obtain a 'random' world that is indistinguishable from the 'left' world obtained when $b = 0$, provided the adversary makes no calls to F and G identical to the queries inside the `SendTag` oracle when $b = 0$. The probability of this happening is however negligible. By applying the same argument to the adversary execution when $b = 1$, we show that the adversary cannot distinguish between the two worlds. □

6 Conclusion

Several RFID privacy models were critically examined with respect to their assumptions, practical usability and other issues that arise when applying their privacy definition to concrete protocols. We have shown that, while some models are based on unrealistic assumptions, others are impractical to apply. We presented a new RFID privacy model, that, based on the classic notion of indistinguishability, combines the benefits of existing models while avoiding their identified drawbacks. By proving it for a concrete protocol, we show that the notion of (wide) strong privacy can be achieved under our model. Since the privacy model is based on an indistinguishability game, we can fall back on a wide range of existing proof techniques, making the model quite straightforward to use in practice.

Acknowledgements. The authors would like to thank Elena Andreeva, Junfeng Fan, Sebastian Faust, and Roel Peeters for the frequent meetings and discussions; and the anonymous reviewers for their comments and suggestions.

References

1. Armknecht, F., Sadeghi, A.-R., Scafuro, A., Visconti, I., Wachsmann, C.: Impossibility Results for RFID Privacy Notions. Transactions on Computational Science 11, 39–63 (2010)
2. Avoine, G., Dysli, E., Oechslin, P.: Reducing Time Complexity in RFID Systems. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 291–306. Springer, Heidelberg (2006)
3. Bellare, M., Fischlin, M., Goldwasser, S., Micali, S.: Identification Protocols Secure against Reset Attacks. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 495–511. Springer, Heidelberg (2001)
4. Bleichenbacher, D.: Chosen Ciphertext Attacks against Protocols Based on the RSA Encryption Standard PKCS #1. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 1–12. Springer, Heidelberg (1998)
5. Bohli, J.-M., Pashalidis, A.: Relations Among Privacy Notions. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 362–380. Springer, Heidelberg (2009)
6. Bringer, J., Chabanne, H., Icart, T.: Efficient zero-knowledge identification schemes which respect privacy. In: Li, W., Susilo, W., Tupakula, U.K., Safavi-Naini, R., Varadharajan, V. (eds.) ASIACCS, pp. 195–205. ACM, New York (2009)
7. Burmester, M., Le, T., Medeiros, B.: Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In: Proceedings of the 2nd IEEE/CreateNet International Conference on Security and Privacy in Communication Networks (SECURECOMM). IEEE Press, Los Alamitos (2006)
8. Canard, S., Coisel, I., Etrog, J., Girault, M.: Privacy-preserving rfid systems: Model and constructions. Cryptology ePrint Archive, Report 2010/405 (2010), `http://eprint.iacr.org/`
9. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: STOC, pp. 235–244 (2000)
10. Atmel Corporation. Innovative Silicon IDIC solutions (2007), `http://www.atmel.com/dyn/resources/prod_documents/doc4602.pdf`
11. Damgård, I., Østergaard, M.: RFID Security: Tradeoffs between Security and Efficiency. Cryptology ePrint Archive, Report 2006/234 (2006), `http://eprint.iacr.org/`
12. D'Arco, P., Scafuro, A., Visconti, I.: Revisiting DoS Attacks and Privacy in RFID-Enabled Networks. In: Dolev, S. (ed.) ALGOSENSORS 2009. LNCS, vol. 5804, pp. 76–87. Springer, Heidelberg (2009)
13. D'Arco, P., Scafuro, A., Visconti, I.: Semi-Destructive Privacy in DoS-Enabled RFID systems. In: RFIDSec (2009)
14. Deng, R.H., Li, Y., Yung, M., Zhao, Y.: A New Framework for RFID Privacy. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 1–18. Springer, Heidelberg (2010)
15. Goyal, V., Sahai, A.: Resettably Secure Computation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 54–71. Springer, Heidelberg (2009)
16. Ha, J., Moon, S.-J., Zhou, J., Ha, J.: A New Formal Proof Model for RFID Location Privacy. In: Jajodia, S., López, J. (eds.) [19], pp. 267–281
17. Hutter, M., Schmidt, J.-M., Plos, T.: RFID and Its Vulnerability to Faults. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 363–379. Springer, Heidelberg (2008)
18. I.C.A. Organization. Machine Readable Travel Documents, Doc 9303, Part 1 Machine Readable Passports, 5th edn. (2003)

19. Nali, D., van Oorschot, P.C.: CROO: A Universal Infrastructure and Protocol to Detect Identity Fraud. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 130–145. Springer, Heidelberg (2008)
20. Juels, A., Weis, S.A.: Defining Strong Privacy for RFID. In: PerCom Workshops, pp. 342–347. IEEE Computer Society, Los Alamitos (2007)
21. Kasper, T., Oswald, D., Paar, C.: New Methods for Cost-Effective Side-Channel Attacks on Cryptographic RFIDs. In: RFIDSec (2009)
22. Mangard, S., Oswald, E., Popp, T.: Power analysis attacks - revealing the secrets of smart cards. Springer, Heidelberg (2007)
23. Ng, C.Y., Susilo, W., Mu, Y., Safavi-Naini, R.: RFID Privacy Models Revisited. In: Jajodia, S., López, J. (eds.) [19], pp. 251–266
24. Ng, C.Y., Susilo, W., Mu, Y., Safavi-Naini, R.: New Privacy Results on Synchronized RFID Authentication Protocols against Tag Tracing. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 321–336. Springer, Heidelberg (2009)
25. Paise, R.-I., Vaudenay, S.: Mutual Authentication in RFID: Security and Privacy. In: ASIACCS 2008, pp. 292–299. ACM Press, New York (2008)
26. Plos, T.: Evaluation of the Detached Power Supply as Side-Channel Analysis Countermeasure for Passive UHF RFID Tags. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 444–458. Springer, Heidelberg (2009)
27. Sadeghi, A.-R., Visconti, I., Wachsmann, C.: User Privacy in Transport Systems Based on RFID E-Tickets. In: Bettini, C., Jajodia, S., Samarati, P., Wang, X.S. (eds.) PiLBA. CEUR Workshop Proceedings, vol. 397 (2008), `CEUR-WS.org`
28. Sadeghi, A.-R., Visconti, I., Wachsmann, C.: Anonymizer-Enabled Security and Privacy for RFID. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 134–153. Springer, Heidelberg (2009)
29. Sadeghi, A.-R., Visconti, I., Wachsmann, C.: Efficient RFID security and privacy with anonymizers. In: RFIDSec (2009)
30. NXP Semiconductors. MIFARE, `http://www.mifare.net/`
31. Van Le, T., Burmester, M., de Medeiros, B.: Universally composable and forward-secure RFID authentication and authenticated key exchange. In: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, ASIACCS 2007, pp. 242–252. ACM Press, New York (2007)
32. Vaudenay, S.: On Privacy Models for RFID. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)
33. Vaudenay, S.: Invited talk at RFIDSec 2010 (2010)
34. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 201–212. Springer, Heidelberg (2004)

A Extending the Model

In a typical indistinguishability-based security/privacy definition, a challenger picks a random bit b and then offers a set of well-defined interfaces over which an adversary $\mathcal{A}$ can interact with the challenger. In 'left-or-right' security/privacy definitions, in particular, the interface specification requires that $\mathcal{A}$ provides *a pair* of identically formatted inputs to the challenger. The value of b can be interpreted as indicating in which of two possible configurations the challenger

operates, namely the 'left' or the 'right' configuration, and $\mathcal{A}$'s job is to determine this configuration.

It is possible to generalise left-or-right indistinguishability such that, the challenger picks one out of 2^n possible configurations, giving us an n- indistinguishability game, with adversary $\mathcal{A}_n$. Suppose there is a system $\mathcal{S}$ that, if invoked with some parameter α (taken from a system-specific parameter space A), produces an output $\mathcal{S}(\alpha)$. The challenger chooses a positive number n, such that n is polynomial in k and generates an n-bit vector $\hat{b} = (\hat{b}_1, \ldots, \hat{b}_n)$ uniformly at random. Finally, it offers an interface over which $\mathcal{A}_n$ may query the challenger with triplets of the form $(i, \alpha_0, \alpha_1) \in \{1, \ldots, n\} \times A \times A$. On input such a triple, the challenger outputs $\mathcal{S}(\alpha_{\hat{b}_i})$.

At the end of the game, $\mathcal{A}_n$ outputs a guess $\hat{g}$ for $\hat{b}$, and we say that it wins the game if $\hat{g} = \hat{b}$. If there exists some $\mathcal{A}_n$ such that $\Pr(\mathcal{A}_n\text{wins}) > 1/2^n + \epsilon$, where ϵ is any function that is non-negligible in k, then we say that $\mathcal{A}_n$ has 'non-negligible advantage' and that $\mathcal{S}$ is not secure.

In general, it is unclear whether or not n-indistinguishability implies 1-indistinguishability. In principle, a system could be secure if the adversary has to identify a string from a space that is exponentially large in k, but may fail security if the adversary just needs to identify a single hidden bit.

Lemma 1 (1-indistinguishability implies n-indistinguishability). *If a system $\mathcal{S}$ satisfies 1-indistinguishability then $\mathcal{S}$ also satisfies n-indistinguishability.*

Proof. We construct an 1-indistinguishability adversary $\mathcal{A}$ that uses an n- indistinguishability adversary $\mathcal{A}_n$ as a black box. $\mathcal{A}$ proceeds as follows. First, it uniformly at random chooses two n-bit vector κ and λ such that $\kappa \neq \lambda$. Then it offers the interface (i, α_0, α_1) to $\mathcal{A}_n$. For each (i, α_0, α_1) received from $\mathcal{A}_n$, $\mathcal{A}$ forwards the query $(\alpha_{\kappa_i}, \alpha_{\lambda_i})$ to the challenger, and returns the challenger's output. By forwarding the queries this way, $\mathcal{A}$ simulates $\hat{b} = \kappa$ if $b = 0$, and $\hat{b} = \lambda$ if $b = 1$ for $\mathcal{A}_n$. In the rest of the proof $\hat{b}$ will denote the κ if $b = 0$ and λ if $b = 1$, $\bar{\hat{b}}$ will denote the κ if $b = 1$ and λ if $b = 0$. Accordingly, and given $\mathcal{A}_n$'s guess $\hat{g}$, $\mathcal{A}$ outputs the guess $b = 0$ if $\hat{g} = \kappa$, $b = 1$ if $\hat{g} = \lambda$, or simply a uniformly at random selected bit otherwise.

Consider the $2^n \times 2^n$ matrix P with elements $p_{i,j} = \Pr(\mathcal{A}_n \text{ outputs } j \mid \hat{b} = i)$. That is, P contains the probabilities that $\mathcal{A}_n$ outputs any possible value $\hat{g}$, conditional on the value of $\hat{b}$; the element at row number i and column number j is the probability that $\mathcal{A}_n$ outputs $\hat{g} = j$ (encoded as a bit vector), given the challenge bit vector has the value $\hat{b} = i$ (encoded as a bit vector). Note that, for all $0 \leq i \leq 2^n$, $\sum_j p_{i,j} = 1$.

For any given choice of a pair (κ, λ), the probability that $\mathcal{A}_n$ wins (i.e. that it outputs $\hat{g} = \hat{b}$) is $1/2(p_{\kappa,\kappa} + p_{\lambda,\lambda})$. Similarly, the probability that it ouputs $\hat{g} = \bar{\hat{b}}$ is $1/2(p_{\kappa,\lambda} + p_{\lambda,\kappa})$. Averaging over all possible choices of (κ, λ) we obtain

$$\Pr(\mathcal{A}_n \text{ wins}) = \frac{1}{2^n(2^n - 1)} \sum_{\substack{\kappa,\lambda \in \{0,1\}^n \\ \kappa \neq \lambda}} \frac{1}{2}(p_{\kappa,\kappa} + p_{\lambda,\lambda}) = \frac{\mathcal{D}}{2^n} \tag{2}$$

$$\Pr(\text{err}) = \frac{1}{2^n(2^n-1)} \sum_{\substack{\kappa,\lambda \in \{0,1\}^n \\ \kappa \neq \lambda}} \frac{1}{2}(p_{\kappa,\lambda} + p_{\lambda,\kappa}) = \frac{2^n - \mathcal{D}}{2^n(2^n-1)}, \tag{3}$$

where $\mathcal{D} = \sum_{i=1}^{2^n} p_{i,i}$ is the trace of P. By construction of our $\mathcal{A}$, we have

$$\Pr(\mathcal{A} \text{ wins}) = \Pr(\mathcal{A}_n \text{ wins}) + 1/2(1 - \Pr(\mathcal{A}_n \text{ wins}) - \Pr(\text{err})) \tag{4}$$

and substituting Equations 2 and 3 into Equation 4, we obtain

$$\Pr(\mathcal{A} \text{ wins}) = \frac{1}{2} + \frac{2^n(\mathcal{D}-1)}{2^{n+1}(2^n-1)}. \tag{5}$$

By assumption we have that $\Pr(\mathcal{A}_n \text{ wins}) > 1/2^n + \epsilon$ for all functions ϵ that are negligible in k. Hence, $\Pr(\mathcal{A}_n \text{ wins}) = 1/2^n + \delta$ for some non-negligible positive $\delta \leq 1 - 1/2^n$. In terms of the elements in P, we have $\mathcal{D} = 1 + 2^n\delta$ and when substituting this into Equation 5 we obtain $\Pr(\mathcal{A} \text{ wins}) = \frac{1}{2} + \frac{2^n\delta}{2(2^n-1)} > 1/2 + \delta/2$. Hence, $\mathcal{A}$'s advantage is non-negligible. □

Unlike standard hybrid arguments, the advantage δ is at most divided by 2, when going from an n-bit distinguisher to a 1-bit distinguisher.

B Mutual Authentication

Since our model is not based anymore on the blinder construction of Paise-Vaudenay [25], none of the impossibility results of [1] apply. It is straightforward to modify the proof from Section 5.1 to the mutual authentication protocol based on IND-CCA encryption from Section 6.3 in [25].