

# PUF-Enhanced RFID Security and Privacy

Ahmad-Reza Sadeghi<sup>1</sup>, Ivan Visconti<sup>2</sup>, and Christian Wachsmann<sup>1</sup>

<sup>1</sup> Horst Görtz Institute for IT-Security (HGI), Ruhr-University Bochum, Germany  
{ahmad.sadeghi, christian.wachsmann}@trust.rub.de

<sup>2</sup> Dipartimento di Informatica ed Applicazioni, University of Salerno, Italy  
visconti@dia.unisa.it

**Abstract** RFID-enabled systems allow fully automatic wireless identification of objects and are rapidly becoming a pervasive technology with various applications. However, despite their benefits, RFID-based systems also pose challenging risks, in particular concerning user privacy. Indeed, most RFID chips are computationally and memory constrained devices without protection against physical tampering. Thus, existing computationally demanding privacy-protecting schemes cannot be applied for RFID while physical attacks that reveal the tag secrets impede the use of symmetric-key based techniques. Hence, defining and designing *usable* and privacy-preserving RFID protocols is a challenging open problem.

Recently, Vaudenay presented a comprehensive RFID security and privacy framework that captures authentication of tags to readers and anonymity aspects. This framework defines eight privacy notions that correspond to adversaries of different strength, i.e., that differ in their ability to access the secrets of (i.e., to corrupt) tags and to obtain auxiliary information from tag to reader communication.

In this paper, we present an efficient privacy-preserving RFID protocol that addresses Vaudenay's open question on the feasibility of *destructive privacy*, i.e., privacy of tags that are destroyed during corruption. Our protocol is based on the use of Physically Unclonable Functions (PUFs), which provide cost-efficient means to fingerprint chips based on their physical properties and can be used to realize tamper-evident storage for cryptographic secrets.

## 1 Introduction

Radio Frequency Identification (RFID) is a technology that enables RFID *readers* to perform fully automatic wireless identification of objects that are labeled with RFID *tags*. Initially, this technology was mainly used for electronic labeling of pallets, cartons and products to enable seamless supervision of supply chains. Today, RFID technology is widely deployed to many other applications as well, including animal and product identification [19,1], access control [1,22], electronic tickets [22] and passports [15], and even human implantation [16].

As pointed out in previous publications (see, e.g., [33,16,26,29,28]), this prevalence of RFID technology introduces various risks, in particular concerning the privacy of its users and holders. The most deterrent privacy risk concerns the tracking of users, which allows the creation and misuse of detailed user profiles. Thus, an RFID system should provide *anonymity* (confidentiality of the tag identity) as well as *untraceability* (unlinkability of the communication of a tag) even in case the state of a tag has been disclosed. Despite these privacy risks, classical threats to authentication and identification systems must be considered as well. Indeed, potential threats to RFID systems are attacks, where the adversary tries to impersonate or copy a legitimate tag. By legitimate we mean a tag created by an accredited tag issuer. Thus, appropriate countermeasures must be provided (*authentication* and *unclonability*). Moreover, there are some other risks such as denial-of-service attacks [5], which must also be prevented (*availability*).

The design of a secure privacy-preserving RFID scheme requires a careful analysis in an appropriate formal security and privacy model. Existing security and privacy models for RFID (see, e.g., [2,17,6,5]) often do not consider important aspects like adversaries with access to auxiliary information or the privacy of corrupted tags (whose secrets have been disclosed). Recently, a comprehensive security and privacy model that generalizes and improves

many previous works in a single concise framework has been proposed in [32] and refined in [20,24,8,7,27]. In the following, we refer to the privacy model of [32] as the *Vaudenay-Model*. The Vaudenay-Model introduces eight privacy notions, which correspond to adversaries of different strength. The strongest *achievable* privacy notion in this model (*narrow-strong privacy*) allows the adversary to arbitrarily corrupt tags but does not capture the availability of auxiliary information. If auxiliary information is of concern, the weaker notions of *destructive* and *forward privacy* must be considered while *weak privacy* does not adequately model the capabilities of real-world adversaries since weak privacy does not allow tag corruption. It has been shown that narrow-strong privacy requires the use of public-key cryptography [32], which in general clearly exceeds the capabilities of current cost-efficient RFIDs [1,22]. Moreover, [32] shows that forward privacy can be achieved at the cost of using public-key cryptography while the feasibility of the stronger notion of destructive privacy has been and still is the most challenging open problem [32].

**Contribution.** In this paper, we propose a new privacy-preserving tag authentication protocol for RFID that can be proven to be destructive private in the Vaudenay-Model. This means that our protocol provides untraceability of tags against adversaries that permanently destroy a tag by physically attacking (i.e., corrupting) it. Our protocol is based on the weak private protocol proposed in [32] and uses Physically Unclonable Functions (PUFs) as tamper-evident key storage in a similar way as described in [30]. This means that the tag authentication key is not stored on the tag but reconstructed from the physical characteristics of the RFID chip each time it is needed. The properties of the PUF ensure that any attempt to physically tamper with the PUF to obtain the authentication secret of the tag result in destruction of the PUF and the tag secret, which corresponds to the definition of a destructive adversary in the Vaudenay-Model.

Note that in the Vaudenay-Model, the only information that differentiates a tag from another tag is a binary state  $S$  that is stored on each tag during its creation. However, the use of PUFs implies placing on the tag a physical (non-digital) object, thus differentiating tags also from a physical point of view. Hence, we solve the problem of achieving destructive privacy in a very mild variant of the Vaudenay-Model that includes the possibility of physically differentiating tags during their creation, in our case through the use of PUFs. For the sake of simplifying the exposition in the remaining part of the paper we will not insist in claiming the *revisited* Vaudenay-Model, but we will stick with Vaudenay-Model.

**Outline.** We first informally discuss the general RFID scenario on a high level in Section 2. Then we give an overview on existing privacy-preserving RFID protocols based on Physically Unclonable Functions (PUFs) in Section 3. We introduce our notation in Section 4 and describe the technical details of the Vaudenay-Model in Section 5. In Section 6 we present our destructive private RFID protocol based on PUFs and prove its security properties in Section 7.

## 2 RFID System and Requirement Analysis

**System Model.** An RFID system consists of at least an operator  $\mathcal{I}$ , a reader  $\mathcal{R}$  and a tag  $\mathcal{T}$ .  $\mathcal{I}$  initializes  $\mathcal{T}$  and  $\mathcal{R}$  before they are deployed in the system.  $\mathcal{T}$  and  $\mathcal{R}$  are called *legitimate* if they have been initialized by  $\mathcal{I}$ . In many applications  $\mathcal{T}$  is a hardware token with constrained computing and memory capabilities that is equipped with a radio interface [1,22]. All information (e.g., secrets and data) that is stored on  $\mathcal{T}$  is denoted as the *state* of  $\mathcal{T}$ . Usually  $\mathcal{T}$  is attached to some object or carried by a user of the RFID system [12,21].  $\mathcal{R}$  is a stationary or mobile computing device that interacts with  $\mathcal{T}$  when  $\mathcal{T}$  gets into the reading range of  $\mathcal{R}$ . The main purpose of this interaction usually is the authentication of  $\mathcal{T}$  to  $\mathcal{R}$ . Depending on the use case,  $\mathcal{R}$  may also authenticate to  $\mathcal{T}$  and/or  $\mathcal{R}$  may obtain additional information like

the identity of  $\mathcal{T}$ .  $\mathcal{R}$  can have a sporadic or permanent online connection to some backend system  $\mathcal{D}$ , which typically is a database maintaining detailed information on all tags in the system.  $\mathcal{D}$  is initialized and maintained by  $\mathcal{I}$  and can be read and updated by  $\mathcal{R}$ .

**Trust and Adversary Model.** The operator  $\mathcal{I}$  maintains the RFID system, and thus is considered to behave correctly. However,  $\mathcal{I}$  may be curious since he may collect user information. Since  $\mathcal{T}$  and  $\mathcal{R}$  communicate over a radio link, any entity can eavesdrop and manipulate this communication, even from outside the nominal reading range of  $\mathcal{R}$  and  $\mathcal{T}$  [18]. Thus, the adversary  $\mathcal{A}$  can be every (potentially unknown) entity. Besides the communication between  $\mathcal{T}$  and  $\mathcal{R}$ ,  $\mathcal{A}$  can also obtain useful auxiliary information (e.g., by visual observation) on whether  $\mathcal{R}$  accepted  $\mathcal{T}$  as a legitimate tag [17,32]. Most commercial RFID tags are cost-efficient devices without expensive protection mechanisms against physical tampering [1,22]. Hence,  $\mathcal{A}$  can physically attack (*corrupt*)  $\mathcal{T}$  and obtain its state (e.g., its secrets). In practice, RFID readers are embedded devices that can be integrated into mobile devices (e.g., mobile phones or PDAs) or computers. The resulting complexity exposes them to sophisticated hard- and software attacks (e.g., viruses and Trojans). Hence,  $\mathcal{A}$  can get full control over  $\mathcal{R}$  [3].

**Security and Privacy Objectives.** The most deterrent privacy risk concerns the *tracking* of tag users, which allows the creation and misuse of detailed user profiles in an RFID system [16]. For instance, detailed movement profiles can leak sensitive information on the personal habits and interests of the tag user. The major security threats are to create illegitimate (*forge*) tags that are accepted by honest readers, to simulate (*impersonate*) or copy (*clone*) legitimate tags, and to permanently prevent users from using the RFID system (*denial-of-service*) [5]. Thus, an RFID system should provide *anonymity* as well as *untraceability* of tags even when their states have been disclosed. Anonymity means the confidentiality of the identity of a tag whereas untraceability refers to the unlinkability of the communication of a tag. The main security objective is to ensure that only legitimate tags are accepted by honest readers (*tag authentication*). Most use cases (like access control systems) additionally require the reader to determine the authentic tag identity (*tag identification*). Moreover, there are several applications where reader authentication is a fundamental security property. However, most use cases do not require reader authentication.

### 3 Related Work

Physically Unclonable Functions (PUFs) are a very interesting and promising approach to increase the security of existing RFID systems. Moreover, they open new directions towards cost-efficient privacy-preserving protocols based on physical assumptions. A PUF consists of an inherently unclonable noisy function  $P$  that is embedded into a physical object [31]. The unclonability of a PUF comes from randomness generated during its manufacturing processes. A PUF maps challenges to responses. A *challenge*  $c$  is a stimulus signal input to the PUF that makes the PUF to return a *response*  $r' \leftarrow P(c)$  that is specific for that PUF with respect to the stimulus  $c$ . This response  $r'$  relies on the physical properties of the corresponding physical object, which, however, is subject to environmental noise (e.g., temperature or supply voltage variations). Thus, the PUF will always return slightly different responses  $r'$  to the same stimulus  $c$ . These slight deviations can be removed by a small circuit, called *Fuzzy Extractor* [11], that (up to a certain threshold) maps different responses  $r'$  to a unique value  $r$  for each specific challenge  $c$ . The Fuzzy Extractor needs some additional input  $w$  (called *helper data*) to remove the effects of noise on the PUF. Moreover, two different PUFs that are challenged with the same stimulus will return seemingly independent responses with overwhelming probability. A PUF can be embedded into a microchip, e.g., by exploiting statistical variations of delays of gates and wires within the chip [13]. These deviations are

unique for every sample from a set of chips (even from the same lot or wafer) that implement the same circuit.

One of the first proposals of using PUFs in RFID systems is introduced by [25]. It proposes the manufacturer of a tag to store a set of challenge-response pairs in a database, which can later be used by RFID readers that are connected to this database to identify a tag. The idea is that the reader chooses a challenge from the database, queries the tag and checks whether the database contains a tuple that matches the response received from the tag. One problem of this approach is that challenge-response pairs cannot be reused since this would enable replay attacks and allow tracing of tags. Hence, the number of tag authentications is limited by the number of challenge-response pairs in the database. This scheme has been implemented on an RFID tag and its security and usability has been analyzed in [9]. The authors of [14] propose a similar approach based on the physical characteristics of SRAM cells. The advantage of [14] is that SRAM-PUFs can be implemented using the existing SRAM memory cells of the RFID chip without the need for additional hardware.

In [30], the authors propose to use a PUF as secure key storage for the secret authentication key of the RFID tag. This means that instead of storing the key in some protected memory, a PUF is used to reconstruct the key whenever it is needed. Since the key is inherently hidden within the physical structure of the PUF, obtaining this secret by hardware-related attacks is supposed to be intractable for real-world adversaries [13]. According to [30], a PUF-based key storage can be implemented with less than 1000 gates. However, the authentication scheme proposed in [30] relies on public-key cryptography, which is still much too expensive for current low-cost RFID tags.

The authors of [4] propose to frequently update the identity of tags to provide privacy. They suggest to equip each RFID tag with a PUF  $P$  that is used to derive new tag identifiers. Since readers cannot recompute these identifiers, the authors propose the readers to access a database that stores a tuple  $(ID_0, ID_1, \dots, ID_m)$  for each legitimate tag  $\mathcal{T}$  where  $ID_0$  is a random tag identifier and  $ID_{i+1} = P(ID_i)$  for  $i \in \{0, \dots, m-1\}$ . To authenticate to a reader, a tag first sends its current identifier  $ID_i$  and then updates its identity to  $ID_{i+1} \leftarrow P(ID_i)$ . The reader then checks whether there is a tuple that contains a value  $ID_i$  in the database. In case the reader finds  $ID_i$ , it accepts the tag and invalidates all previous database entries  $ID_j$  where  $j \leq i$  to prevent replay attacks. A major drawback of this scheme is that a tag can only be authenticated  $m$  times without being re-initialized, which, as the authors mention, allows an adversary to perform denial-of-service attacks.

## 4 Preliminaries and Notation

**General Notation.** For a finite set  $S$ ,  $|S|$  denotes the size of set  $S$  whereas for an integer (or a bitstring)  $n$  the term  $|n|$  means the bit-length of  $n$ . The term  $s \in_R S$  means the assignment of a uniformly chosen element of  $S$  to variable  $s$ . With  $\emptyset$  we denote both the empty set as well as the empty string. Let  $A$  be a probabilistic algorithm. Then  $y \leftarrow A(x)$  means that on input  $x$ , algorithm  $A$  assigns its output to variable  $y$ . The term  $[A(x)]$  denotes the set of all possible outputs of  $A$  on input  $x$ .  $A_K(x)$  means that the output of  $A$  depends on  $x$  and some additional parameter  $K$  (e.g., a secret key). The term  $\text{Prot}[A : x_A; B : x_B; * : x_{pub}] \rightarrow [A : y_A; B : y_B]$  denotes an interactive protocol  $\text{Prot}$  between two probabilistic algorithms  $A$  and  $B$ . Hereby,  $A$  (resp.  $B$ ) gets a private input  $x_A$  (resp.  $x_B$ ) and a public input  $x_{pub}$ . While  $A$  (resp.  $B$ ) is operating, it can interact with  $B$  (resp.  $A$ ). After the protocol terminates,  $A$  (resp.  $B$ ) returns  $y_A$  (resp.  $y_B$ ). Let  $E$  be some event (e.g., the result of a security experiment), then  $\Pr[E]$  denotes the probability that  $E$  occurs. Probability  $\epsilon(l)$  is called *negligible* if for all polynomials  $f$  it holds that  $\epsilon(l) \leq 1/f(l)$  for all sufficiently large  $l$ . Probability  $1 - \epsilon(l)$  is called *overwhelming* if  $\epsilon(l)$  is negligible.

**Pseudo-Random Function (PRF).** Let  $l \in \mathbb{N}$  be a security parameter,  $\kappa, \alpha, \beta \in \mathbb{N}$  be polynomially bounded in  $l$  and  $F : \{0, 1\}^{\kappa+\alpha} \rightarrow \{0, 1\}^\beta$  be a family of functions. Consider the following security experiment  $\mathbf{Exp}_{\mathcal{A}_{\text{prf}}}^{\text{prf}-b}$ , where an adversary  $\mathcal{A}_{\text{prf}}$  interacts with a *PRF-challenger*  $\mathcal{C}_{\text{prf}}$ : When initialized with  $l, \kappa, \alpha, \beta$  and  $b \in_R \{0, 1\}$ ,  $\mathcal{C}_{\text{prf}}$  chooses  $K \in_R \{0, 1\}^\kappa$  and initializes an oracle  $\mathcal{O}^{F_K}$  that on input  $x \in \{0, 1\}^\alpha$  returns  $y \leftarrow F_K(x)$  if  $b = 1$  and  $y \in_R \{0, 1\}^\beta$  otherwise. After a polynomial number of queries to oracle  $\mathcal{O}^{F_K}$ ,  $\mathcal{A}_{\text{prf}}$  then must return a bit  $b'$ .  $\mathcal{A}_{\text{prf}}$  wins the security experiment if  $b = b'$ .

**Definition 1 (Pseudo-Random Function [23]).** A pseudo random function (PRF) is a family of functions  $F$  with the following properties:

1. Each function  $F_K \in F$  can be identified by a unique index  $K \in \{0, 1\}^\kappa$ .
2. There is a polynomial time algorithm that given an index  $K \in \{0, 1\}^\kappa$  and input  $x \in \{0, 1\}^\alpha$  computes  $F_K(x)$ .
3. Each probabilistic polynomial time (p.p.t.) adversary  $\mathcal{A}_{\text{prf}}$  has at most negligible advantage

$$\text{Adv}_{\mathcal{A}_{\text{prf}}}^{\text{prf}} = |\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prf}}}^{\text{prf}-1} = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prf}}}^{\text{prf}-0} = 1]|.$$

**Physically Unclonable Function (PUF).** To the best of our knowledge, currently there is no widely accepted security model for PUFs. Moreover, setting up a model that reflects the properties of real PUFs requires precise physical evaluation results to determine the capabilities of an adversary against PUFs in practice. However, industry considers this data as trade secret while academia usually is restricted to prototype implementations of PUFs (e.g., on FPGAs) that do not reflect the properties of real product-quality PUF implementations (e.g., on ASICs). Hence, in this paper, we fall back to an idealized model of PUFs that does *not* reflect *real* PUF implementations but captures the *desired* properties of an *ideal* PUF component.

Let  $l \in \mathbb{N}$  be a security parameter,  $\gamma, \kappa \in \mathbb{N}$  be polynomially bounded in  $l$  and  $\mathsf{P} : \{0, 1\}^\gamma \rightarrow \{0, 1\}^\kappa$  be an ideal PUF. Consider the following security experiment  $\mathbf{Exp}_{\mathcal{A}_{\text{puf}}}^{\text{puf}-b}$  that is similar to  $\mathbf{Exp}_{\mathcal{A}_{\text{prf}}}^{\text{prf}-b}$  described above. The difference is that, when initialized with  $l, \gamma, \kappa$  and  $b \in_R \{0, 1\}$ , the PUF-challenger  $\mathcal{C}_{\text{puf}}$  initializes an oracle  $\mathcal{O}^{\mathsf{P}}$  that on input  $x \in \{0, 1\}^\gamma$  returns  $y \leftarrow \mathsf{P}(x)$  if  $b = 1$  and  $y \in_R \{0, 1\}^\kappa$  otherwise. After a polynomial number of queries to  $\mathcal{O}^{\mathsf{P}}$ ,  $\mathcal{A}_{\text{puf}}$  must return a bit  $b'$ .  $\mathcal{A}_{\text{puf}}$  wins the security experiment if  $b = b'$ .

**Definition 2 (Ideal PUF).** An ideal Physically Unclonable Function (PUF) is a function  $\mathsf{P}$  with the following properties:

1. For all  $c \in \{0, 1\}^\gamma$  and all pairs  $(r_i, r_j) \in [\mathsf{P}(c)]^2$  it holds that probability  $\Pr[r_i = r_j] = 1$ .
2. Each probabilistic polynomial time (p.p.t.) adversary  $\mathcal{A}_{\text{puf}}$  has at most negligible advantage

$$\text{Adv}_{\mathcal{A}_{\text{puf}}}^{\text{puf}} = |\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{puf}}}^{\text{puf}-1} = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}_{\text{puf}}}^{\text{puf}-0} = 1]|$$

3. Any attempt to physically tamper with the object implementing  $\mathsf{P}$  results in destruction of  $\mathsf{P}$ , i.e.,  $\mathsf{P}$  cannot be evaluated any more.

Note that the second property of Definition 2 is similar to the pseudo-randomness property of a PRF (see Definition 1). Hence, the output of an ideal PUF is pseudo-random, which can be achieved by Fuzzy Extractors [10,11]. In addition, the second property of Definition 2 implies that the adversary cannot compute the output of the PUF for an adaptively chosen challenge even after adaptively querying the PUF for a polynomial number of times. In return, this means that the adversary cannot emulate (i.e., impersonate or clone) the PUF

based on its input/output behaviour. The third property of Definition 2 ensures that the adversary cannot obtain any information about the PUF by physical means, which prevents cloning of the PUF. Moreover, the capabilities of the adversary are not limited concerning the creation and querying of other PUFs, which means that different ideal PUFs are independent pseudo-random functions.

## 5 RFID Security and Privacy Model by Vaudenay

In this section, we give a brief formal specification of the RFID security and privacy framework proposed by Vaudenay (Vaudenay-Model) [32], which is one of the most comprehensive RFID privacy and security models up to date.

### 5.1 System Model

The Vaudenay-Model considers RFID systems that consist of a single operator  $\mathcal{I}$ , a single reader  $\mathcal{R}$  and a polynomial number of tags  $\mathcal{T}$ .  $\mathcal{R}$  is assumed to be capable of performing public-key cryptography and of handling multiple instances of the mutual authentication protocol with different tags in parallel. Each tag  $\mathcal{T}$  is a passive device, i.e., it does not have its own power supply but is powered by the electromagnetic field of  $\mathcal{R}$ . Hence,  $\mathcal{T}$  cannot initiate communication, has a narrow communication range (i.e., a few centimeters to meters) and erases its temporary state (i.e., all session-specific information and randomness) after it gets out of the reading range of  $\mathcal{R}$ . Each  $\mathcal{T}$  is assumed to be capable of performing basic cryptographic functions like hashing, random number generation and symmetric-key encryption. The authors of [32,24] also use public-key encryption, although it exceeds the capabilities of currently available commercial RFID tags [1,22].

In the Vaudenay-Model the operator  $\mathcal{I}$  sets up the reader  $\mathcal{R}$  and all tags  $\mathcal{T}$ . Hence, there are two setup protocols where  $\mathcal{R}$  and  $\mathcal{T}$  are initialized and their system parameters (e.g., keys) are generated and defined. A third protocol between  $\mathcal{T}$  and  $\mathcal{R}$  covers mutual authentication.

**Definition 3 (RFID System [24]).** *An RFID system is a tuple of probabilistic polynomial time (p.p.t.) algorithms  $(\mathcal{R}, \mathcal{T}, \text{SetupReader}, \text{SetupTag}, \text{Ident})$  that are defined as follows:*

$\text{SetupReader}(1^l) \rightarrow (sk_{\mathcal{R}}, pk_{\mathcal{R}}, \text{DB})$  *On input of a security parameter  $l$ , this algorithm initializes the reader algorithm  $\mathcal{R}$  by creating some public parameters  $pk_{\mathcal{R}}$  that are known to all entities and some secret parameters  $sk_{\mathcal{R}}$  that are only known to  $\mathcal{R}$ . This algorithm also initializes a credentials database  $\text{DB}$  that can only be accessed by  $\mathcal{R}$  and that stores the identities and the authentication secrets of all legitimate tags.*

$\text{SetupTag}_{pk_{\mathcal{R}}}(\text{ID}) \rightarrow (K, S)$  *Creates a tag  $\mathcal{T}_{\text{ID}}$ , which is an instance of the tag algorithm  $\mathcal{T}$ . Hereby,  $pk_{\mathcal{R}}$  is used to generate a secret  $K$  and an initial tag state  $S$ .  $\mathcal{T}_{\text{ID}}$  is initialized with  $S$  and  $(\text{ID}, K)$  is stored in  $\text{DB}$ .*

$\text{Ident}[\mathcal{T}_{\text{ID}}:S; \mathcal{R}:sk_{\mathcal{R}}, \text{DB}; *:pk_{\mathcal{R}}] \rightarrow [\mathcal{T}_{\text{ID}}:-; \mathcal{R}:out_{\mathcal{R}}]$  *is an interactive protocol between  $\mathcal{T}_{\text{ID}}$  and  $\mathcal{R}$ .  $\mathcal{T}_{\text{ID}}$  takes as input its current state  $S$  while  $\mathcal{R}$  has as input  $sk_{\mathcal{R}}$  and  $\text{DB}$ . The common input to all parties is  $pk_{\mathcal{R}}$ . After the protocol terminates,  $\mathcal{R}$  returns either the identity  $\text{ID}$  of  $\mathcal{T}_{\text{ID}}$  or  $\perp$  to indicate that  $\mathcal{T}_{\text{ID}}$  is not a legitimate tag.  $\mathcal{T}_{\text{ID}}$  has no output.*

### 5.2 Trust and Adversary Model

The Vaudenay-Model assumes the issuer  $\mathcal{I}$ , the backend database  $\mathcal{D}$  and the readers to be trusted whereas a tag  $\mathcal{T}$  can be compromised. All readers and  $\mathcal{D}$  are subsumed to *one single* reader entity  $\mathcal{R}$  that cannot be corrupted. The privacy and security objectives are based on security experiments, where a p.p.t. adversary  $\mathcal{A}$  interacts with a set of oracles that model the capabilities of  $\mathcal{A}$ . These oracles are:

- CreateTag<sup>b</sup>(ID)** Allows  $\mathcal{A}$  to set up a tag  $\mathcal{T}_{\text{ID}}$  with identifier  $\text{ID}$  by calling  $\text{SetupTag}_{pk_{\mathcal{R}}}(\text{ID})$  to create  $(K, S)$  for  $\mathcal{T}_{\text{ID}}$ . If input  $b = 1$ ,  $\mathcal{A}$  chooses  $\mathcal{T}_{\text{ID}}$  to be legitimate, which means that  $(\text{ID}, K)$  is added to the credentials database  $\text{DB}$  of  $\mathcal{R}$ . For input  $b = 0$ ,  $\mathcal{A}$  chooses  $\mathcal{T}_{\text{ID}}$  to be illegitimate and  $(\text{ID}, K)$  is *not* added to  $\text{DB}$ .
- Draw( $\delta$ )**  $\rightarrow (vtag_1, b_1, \dots, vtag_n, b_n)$  Initially,  $\mathcal{A}$  cannot interact with any tag but must query the **Draw** oracle to get access to a set of tags that has been chosen according to a given tag distribution  $\delta$ .  $\mathcal{A}$  knows the tags he can interact with by some temporary tag identifiers  $vtag_1, \dots, vtag_n$ . The **Draw** oracle manages a secret look-up table  $\Gamma$  that keeps track of the real tag identifier  $\text{ID}_i$  that is associated with each temporary tag identifier  $vtag_i$  (i.e.,  $\Gamma[vtag_i] = \text{ID}_i$ ). Moreover, the **Draw** oracle also provides  $\mathcal{A}$  with information on whether the tags are legitimate ( $b_i = 1$ ) or not ( $b_i = 0$ ).
- Free( $vtag$ )** Makes tag  $vtag$  inaccessible to  $\mathcal{A}$ . This means that  $\mathcal{A}$  cannot interact with  $vtag$  any longer until it is made accessible again (under a new temporary identifier  $vtag'$ ) by another **Draw** query.
- Launch()**  $\rightarrow \pi$  Makes  $\mathcal{R}$  to start a new instance  $\pi$  of the **Ident** protocol, which allows  $\mathcal{A}$  to start different concurrent **Ident** protocol instances with  $\mathcal{R}$ .
- SendReader( $m, \pi$ )**  $\rightarrow m'$  Sends a message  $m$  to instance  $\pi$  of the **Ident** protocol that is run by  $\mathcal{R}$ .  $\mathcal{R}$  interprets  $m$  as a protocol message of instance  $\pi$  of **Ident** and responds with a message  $m'$ .
- SendTag( $m, vtag$ )**  $\rightarrow m'$  Sends a message  $m$  to the tag  $\mathcal{T}_{\text{ID}}$  that is known as  $vtag$  to  $\mathcal{A}$ .  $\mathcal{T}_{\text{ID}}$  interprets  $m$  as a protocol message of the **Ident** protocol and responds with a message  $m'$ .
- Result( $\pi$ )** Returns 1 if instance  $\pi$  of the **Ident** protocol has been completed and the tag  $\mathcal{T}_{\text{ID}}$  that participated in instance  $\pi$  has been accepted by  $\mathcal{R}$ . Otherwise **Result** returns 0.
- Corrupt( $vtag$ )**  $\rightarrow S$  Returns the current state  $S$  of the tag  $\mathcal{T}_{\text{ID}}$  that is known as  $vtag$  to  $\mathcal{A}$ .

The Vaudenay-Model distinguishes the following adversary classes, which differ in (i) their ability to corrupt tags and (ii) the availability of auxiliary information (i.e., the ability to access the **Corrupt** and **Result** oracle).

**Definition 4 (Adversary Classes [24]).** *An adversary is a p.p.t. algorithm that has arbitrary access to the oracles described above. Weak adversaries cannot access the **Corrupt** oracle. Forward adversaries can no longer query any other oracle than **Corrupt** after they made the first **Corrupt** query. Destructive adversaries cannot query any oracle for  $vtag$  again after they made a **Corrupt**( $vtag$ ) query. Strong adversaries have no restriction on the use of the **Corrupt** oracle. Narrow adversaries cannot access the **Result** oracle.*

According to the above notation and definitions, we now recall the definitions of correctness, security and privacy of the Vaudenay-Model.

### 5.3 Definition of Correctness, Security and Privacy

The main security objective of the Vaudenay-Model is tag authentication. Availability and protection against cloning are not captured by the Vaudenay-Model. The privacy objectives are anonymity and unlinkability.

**Correctness.** Correctness describes the honest behavior of the tags  $\mathcal{T}$  and the reader  $\mathcal{R}$ .

**Definition 5 (Correctness [24]).** *An RFID system (Definition 3) is correct if for all  $l \in \mathbb{N}$ , for all  $(sk_{\mathcal{R}}, pk_{\mathcal{R}}, \text{DB}) \in [\text{SetupReader}(1^l)]$ , and for all  $(K, S) \in [\text{SetupTag}_{pk_{\mathcal{R}}}(\text{ID})]$  it holds with overwhelming probability that  $\text{Ident}[\mathcal{T}_{\text{ID}}:S; \mathcal{R}:sk_{\mathcal{R}}, \text{DB}; *:pk_{\mathcal{R}}] \rightarrow [\mathcal{T}_{\text{ID}}:-; \mathcal{R}:\text{ID}]$ .*

**Security.** The security definition given by the Vaudenay-Model focuses on attacks where the adversary aims to impersonate or forge a legitimate tag. It does *not* capture security against cloning and availability. The definition of tag authentication is based on a security experiment  $\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\mathcal{T}\text{-auth}}$  where a strong adversary  $\mathcal{A}_{\text{sec}}$  (Definition 4) must make the reader  $\mathcal{R}$  to identify some tag  $\mathcal{T}_{\text{ID}}$  in some instance  $\pi$  of the `Ident` protocol. To exclude trivial attacks (e.g., relay attacks),  $\mathcal{A}_{\text{sec}}$  is not allowed to simply forward all the messages from  $\mathcal{T}_{\text{ID}}$  to  $\mathcal{R}$  in instance  $\pi$  nor to corrupt  $\mathcal{T}_{\text{ID}}$ . This means that at least some of the protocol messages that made  $\mathcal{R}$  to return ID must have been computed by  $\mathcal{A}_{\text{sec}}$  without knowing the secrets of  $\mathcal{T}_{\text{ID}}$ . With  $\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\mathcal{T}\text{-auth}} = 1$  we denote the case where  $\mathcal{A}_{\text{sec}}$  wins the security experiment.

**Definition 6 (Tag Authentication [24]).** *An RFID system (Definition 3) achieves tag authentication if for every strong adversary  $\mathcal{A}_{\text{sec}}$  (Definition 4)  $\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\mathcal{T}\text{-auth}} = 1]$  is negligible.*

Note that tag authentication is a critical security property and hence must be preserved even against strong adversaries.

**Privacy.** The privacy definition of the Vaudenay-Model is very flexible and, dependent on the adversary class (see Definition 4), it covers different notions of privacy. It captures anonymity and unlinkability and focuses on the privacy leakage of the communication of tags with the reader. It is based on the existence of a simulator  $\mathcal{B}$ , called *blinder*, that can simulate any tag  $\mathcal{T}$  and the reader  $\mathcal{R}$  without knowing their secrets such that an adversary  $\mathcal{A}_{\text{prv}}$  cannot distinguish whether it is interacting with the real or the simulated RFID system. The rationale behind this simulation-based definition is that the communication of  $\mathcal{T}$  with  $\mathcal{R}$  does not leak any information about  $\mathcal{T}$ . Hence, everything  $\mathcal{A}_{\text{prv}}$  observes from the interaction with  $\mathcal{T}$  and  $\mathcal{R}$  appears to be independent of  $\mathcal{T}$  and consequently,  $\mathcal{A}_{\text{prv}}$  cannot distinguish different tags based on their communication, which corresponds to unlinkability.

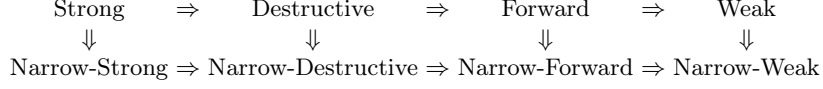
This privacy definition can be formalized by the following privacy experiment  $\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-}b}$ : Let  $\mathcal{A}_{\text{prv}}$  be an adversary according to Definition 4,  $l \in \mathbb{N}$  be a given security parameter and  $b \in_R \{0, 1\}$ . In the first phase of the experiment,  $\mathcal{R}$  is initialized with  $(sk_{\mathcal{R}}, pk_{\mathcal{R}}, \text{DB}) \leftarrow \text{SetupReader}(1^l)$ .  $pk_{\mathcal{R}}$  is given to  $\mathcal{A}_{\text{prv}}$  and  $\mathcal{B}$ . First,  $\mathcal{A}_{\text{prv}}$  is allowed to arbitrarily interact with all oracles defined in Section 5.2. Hereby,  $\mathcal{A}_{\text{prv}}$  is subject to the restrictions of its corresponding adversary class (see Definition 4). If  $b = 1$ , all queries to the `Launch`, `SendReader`, `SendTag` and `Result` oracles are redirected to and answered by  $\mathcal{B}$ . Hereby,  $\mathcal{B}$  can observe all queries  $\mathcal{A}_{\text{prv}}$  makes to all other oracles that are not simulated by  $\mathcal{B}$  and the corresponding responses (“ $\mathcal{B}$  sees what  $\mathcal{A}_{\text{prv}}$  sees”). After a polynomial number of oracle queries, the second phase of the experiment starts. In this second stage,  $\mathcal{A}_{\text{prv}}$  can no longer interact with the oracles but is given the secret table  $\Gamma$  of the `Draw` oracle. Finally,  $\mathcal{A}_{\text{prv}}$  returns a bit  $b'$ , which we denote with  $\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-}b} = b'$ .

**Definition 7 (Privacy [32]).** *Let  $C$  be one of the adversary classes according to Definition 4. An RFID system (Definition 3) is  $C$ -private if for every adversary  $\mathcal{A}_{\text{prv}}$  of  $C$  there exists a p.p.t. algorithm  $\mathcal{B}$  (blinder) such that*

$$\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}} = |\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-}0} = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-}1} = 1]|$$

*is negligible.  $\mathcal{B}$  simulates the `Launch`, `SendReader`, `SendTag` and `Result` oracles to  $\mathcal{A}_{\text{prv}}$  without having access to  $sk_{\mathcal{R}}$  and  $\text{DB}$ . Hereby, all oracle queries  $\mathcal{A}_{\text{prv}}$  makes and their corresponding responses are also sent to  $\mathcal{B}$ .*

All privacy notions defined in the PV-Model [24] are summarized in Figure 1, which also shows the relations among them. It has been shown that strong privacy is impossible [32] while the technical feasibility of destructive privacy has been an open problem.



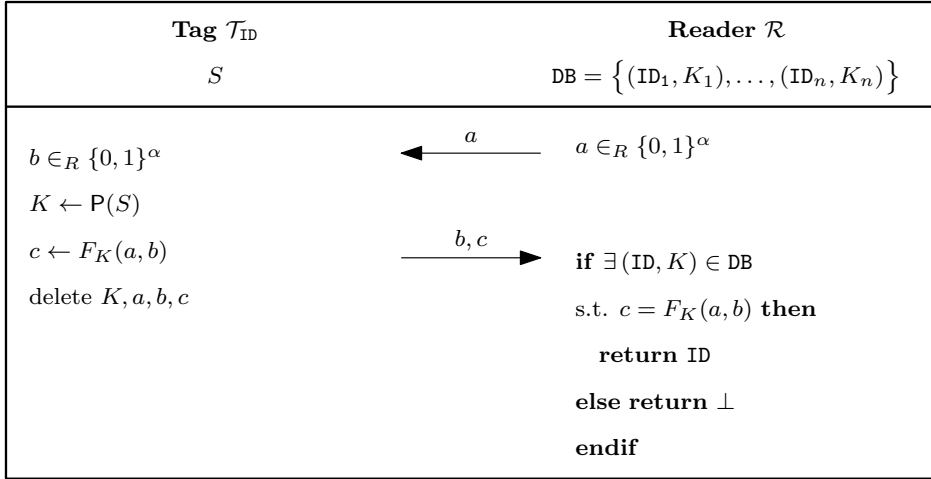
**Figure 1.** Privacy notions defined in the PV-Model [24] and their relations.

## 6 PUF-based Destructive-Private RFID Protocol

In this section, we address an open problem of [32] by presenting the first destructive-private RFID protocol. Our protocol is based on the weak-private protocol of [32], which is a simple challenge-response protocol. To achieve destructive-privacy, in our protocol, the tag  $\mathcal{T}$  does not directly use its state  $S$  as authentication key  $K$ . Instead,  $K$  is derived by evaluating a physically unclonable function  $P$  on input  $S$  each time  $K$  is needed. Hence, the properties of the PUF ensure that the adversary cannot access the tag secret  $K$  but destroys the tag  $\mathcal{T}$  by any attempt to corrupt it.

Let  $l \in \mathbb{N}$  be a given security parameter,  $\alpha, \beta, \gamma, \kappa \in \mathbb{N}$  be polynomial in  $l$  and  $F : \{0, 1\}^\kappa \times \{0, 1\}^{2\alpha} \rightarrow \{0, 1\}^\beta$  be a family of pseudorandom functions. Each tag  $\mathcal{T}$  is equipped with a (unique) PUF  $P : \{0, 1\}^\gamma \rightarrow \{0, 1\}^\kappa$  and is initialized by a random state  $S \in_R \{0, 1\}^\gamma$ . The credentials database  $\text{DB}$  of the reader  $\mathcal{R}$  contains a tuple  $(\text{ID}, K)$  for each legitimate tag  $\mathcal{T}_{\text{ID}}$  where  $K \leftarrow P(S)$ .

Our destructive-private tag authentication protocol is illustrated in Figure 2.  $\mathcal{R}$  starts by



**Figure 2.** Destructive-private PUF-based RFID protocol.

sending a random challenge  $a$  to  $\mathcal{T}_{\text{ID}}$ , which first chooses a random  $b$  and then queries its PUF  $P$  with  $S$  to reconstruct  $K$ . Next,  $\mathcal{T}_{\text{ID}}$  evaluates  $F_K(a, b)$ , sends the result  $c$  and  $b$  to  $\mathcal{R}$  and immediately erases  $K, a, b$  and  $c$  from its temporary memory. On receipt of  $c$ ,  $\mathcal{R}$  recomputes  $F_K(a, b)$  for each tuple  $(\text{ID}, K)$  in  $\text{DB}$  until it finds a match. If  $\mathcal{R}$  finds a matching  $(\text{ID}, K)$ , it accepts  $\mathcal{T}_{\text{ID}}$  by returning  $\text{ID}$ . Otherwise,  $\mathcal{R}$  rejects  $\mathcal{T}_{\text{ID}}$  and returns  $\perp$ .

**Correctness.** Clearly, if both  $\mathcal{T}_{\text{ID}}$  and  $\mathcal{R}$  are legitimate, then the correctness of the `Ident` protocol shown in Figure 2 follows directly from the properties of the PRF  $F$  (see Definition 1) and the correctness of the PUF  $P$  (see Definition 2).

## 7 Security Analysis

Due to space restrictions, we only give proof sketches. The full proofs can be found in Appendix A and Appendix B.

**Theorem 1.** *The RFID protocol illustrated in Figure 2 achieves tag authentication (Definition 6) if  $F$  is a PRF (Definition 1).*

*Proof (Sketch).* Assume by contradiction that there is an adversary  $\mathcal{A}_{\text{sec}}$  against the protocol shown in Figure 2 who violates tag authentication (Definition 6). We show that  $\mathcal{A}_{\text{sec}}$  can be transformed into an algorithm  $\mathcal{A}_{\text{prf}}$  that contradicts the security property of the underlying PRF  $F$  (Definition 1). The main idea of the proof is as follows:

$\mathcal{A}_{\text{prf}}$  uses  $\mathcal{O}^{F_{\tilde{K}}}$  to simulate the oracles defined in Section 5.2 to  $\mathcal{A}_{\text{sec}}$ . After a polynomial number of interactions with the oracles,  $\mathcal{A}_{\text{sec}}$  returns a *new* protocol message  $(\tilde{b}, \tilde{c})$  for a given value  $\tilde{a}$ . (Note that  $\mathcal{A}_{\text{sec}}$  is not allowed to make a `SendTag`( $\tilde{a}, \cdot$ ) query to the tag  $\mathcal{T}_{\text{ID}}$ , which ensures that  $\mathcal{O}^{F_{\tilde{K}}}$  has not been queried with  $(\tilde{a}, \tilde{b})$  before.) Now,  $\mathcal{A}_{\text{prf}}$  sends  $x \leftarrow (\tilde{a}, \tilde{b})$  to  $\mathcal{C}_{\text{prf}}$  who responds with a challenge  $y$ . Note that in case  $\mathcal{O}^{F_{\tilde{K}}}$  simulates  $F_{\tilde{K}}$ , the simulation of the oracles to  $\mathcal{A}_{\text{sec}}$  is perfect. Hence, in this case, by assumption with non-negligible probability it holds that  $\tilde{c} = F_{\tilde{K}}(\tilde{a}, \tilde{b})$ . This means that if  $\mathcal{O}^{F_{\tilde{K}}}$  simulates  $F_{\tilde{K}}$ , then  $\tilde{c} = y$  must hold with non-negligible probability. Clearly, this allows  $\mathcal{A}_{\text{prf}}$  to distinguish between  $F_{\tilde{K}}$  and a randomly chosen value, which contradicts the pseudo-randomness of the PRF  $F$  (Definition 1).  $\square$

**Theorem 2.** *The RFID protocol illustrated in Figure 2 achieves destructive privacy (Definition 7) if the protocol achieves tag authentication (Definition 6),  $\mathsf{P}$  is a PUF (Definition 2) and  $F$  is a PRF (Definition 1).*

*Proof (Sketch).* According to Definition 7, destructive privacy means that there is a blinder  $\mathcal{B}$  that simulates the `Launch`, `SendTag`, `SendReader` and `Result` oracle such that no destructive adversary  $\mathcal{A}_{\text{prv}}$  (Definition 4) can distinguish between the blinder  $\mathcal{B}$  and the real oracles. Hence, we first give the construction of  $\mathcal{B}$  and then show that it cannot be distinguished from real oracle by any destructive adversary  $\mathcal{A}_{\text{prv}}$ .

The simulation of the `Launch` oracle is trivial.  $\mathcal{B}$  simulates the `SendTag` and `SendReader` oracle queries by returning random numbers of the specific output domain. To simulate `Result`,  $\mathcal{B}$  returns 1 only if the corresponding protocol transcript has been generated by a `SendReader` and `SendTag` query (i.e., the transcript has been generated by an “honest” tag and reader) and 0 otherwise.

We show by hybrid arguments that if  $\mathcal{A}_{\text{prv}}$  can distinguish  $\mathcal{B}$  from the real oracles, then we can use  $\mathcal{A}_{\text{prv}}$  to construct a polynomial time algorithm that violates either tag authentication or the security properties of the underlying PUF  $\mathsf{P}$ . Let game  $\mathsf{G}^{(0)}$  be the game where  $\mathcal{A}_{\text{prv}}$  interacts with the real oracles as defined in Section 5.2. Then we consider the hybrid game  $\mathsf{G}^{(1)}$  that is exactly as  $\mathsf{G}^{(0)}$  with the only difference that the states  $S$  and the authentication secrets  $K$  of all tags are simulated by randomly chosen values. We show that if  $\mathcal{A}_{\text{prv}}$  can distinguish between  $\mathsf{G}^{(0)}$  and  $\mathsf{G}^{(1)}$ , then we can use  $\mathcal{A}_{\text{prv}}$  to construct a polynomial time algorithm  $\mathcal{A}_{\text{puf}}$  that contradicts the security property of the PUF  $\mathsf{P}$  (Definition 2). The idea is that  $\mathcal{A}_{\text{puf}}$  uses the PUF-challenger  $\mathcal{C}_{\text{puf}}$  to simulate the oracles defined in Section 5.2 to  $\mathcal{A}_{\text{prv}}$ . By the contradicting assumption  $\mathcal{A}_{\text{prv}}$  detects  $\mathcal{B}$  with non-negligible probability if the oracle  $\mathcal{O}^{\mathsf{P}}$  provided by  $\mathcal{C}_{\text{puf}}$  simulates a random function. Hence, the output of  $\mathcal{A}_{\text{prv}}$  can be used to distinguish between the output of  $\mathsf{P}$  and a random value, which contradicts the security of the PUF  $\mathsf{P}$  (Definition 2).

Next, we consider the hybrid game  $\mathsf{G}^{(2)}$  that is exactly as  $\mathsf{G}^{(1)}$  with the only difference that the `SendTag` oracle is simulated by  $\mathcal{B}$  as described above. We show that if  $\mathcal{A}_{\text{prv}}$  can distinguish between  $\mathsf{G}^{(1)}$  and  $\mathsf{G}^{(2)}$ , then we can use  $\mathcal{A}_{\text{prv}}$  to construct a polynomial time algorithm  $\mathcal{A}_{\text{prf}}$  that contradicts the security property of the PRF  $F$  (Definition 1). Therefore,  $\mathcal{A}_{\text{prf}}$  uses the PRF-challenger  $\mathcal{C}_{\text{prf}}$  to simulate the oracles defined in Section 5.2 to  $\mathcal{A}_{\text{prv}}$ . Since  $\mathcal{A}_{\text{prv}}$  is assumed to detect  $\mathcal{B}$  with non-negligible probability if the oracle  $\mathcal{O}^{F_K}$  provided by  $\mathcal{C}_{\text{prf}}$

simulates a random function,  $\mathcal{A}_{\text{prf}}$  can use the output of  $\mathcal{A}_{\text{prv}}$  to distinguish between the output of  $F_K$  and a random value with non-negligible probability. Clearly, this violates the security property of the PRF  $F$  (Definition 1).

We finally consider the hybrid game  $\mathsf{G}^{(3)}$  that is exactly as  $\mathsf{G}^{(2)}$  with the only difference that the Result oracle is simulated by  $\mathcal{B}$  as described above. We show that if  $\mathcal{A}_{\text{prv}}$  can distinguish between  $\mathsf{G}^{(2)}$  and  $\mathsf{G}^{(3)}$ , then  $\mathcal{A}_{\text{prv}}$  can be used to construct a polynomial time algorithm  $\mathcal{A}_{\text{sec}}$  that contradicts tag authentication (Definition 6). Note that the simulation of Result is perfect except for the case where  $\mathcal{A}_{\text{prf}}$  can generate a protocol transcript (without just forwarding the messages of an uncorrupted honest tag to the reader) that makes the real Result oracle to return 1. However, as shown in the proof of Theorem 1 this can only happen with negligible probability. Note that  $\mathsf{G}^{(3)}$  corresponds to the game where  $\mathcal{A}_{\text{prv}}$  interacts with a full blinder  $\mathcal{B}$ . Hence,  $\mathcal{A}_{\text{prv}}$  cannot distinguish between the real oracles and the full blinder  $\mathcal{B}$ , which completes the proof of Theorem 2.  $\square$

## 8 Conclusion

In this paper, we have shown that PUFs are a very interesting and promising approach to improve the security and privacy of existing RFID systems. However, several aspects of PUFs and their deployment to RFID require further research. Since PUFs are bound to the device in which they are embedded, no other entity can verify the output of a PUF to a given challenge without knowing the correct output value in advance. Another problem with PUFs is that their realizations require careful statistical testing before they can be safely deployed to real security-critical products. Moreover, to our knowledge, there is no complete security and adversary model for PUFs yet.

**Acknowledgments.** We wish to thank Frederik Armknecht, Paolo D’Arco, and Alessandra Scafuro for several useful discussions about RFID privacy notions. This work has been supported in part by the European Commission through the EU ICT program under Contract ICT-2007-216646 ECRYPT II, in part through the FP7 Information and Communication Technologies programme under Contract FET-238811 UNIQUE, in part through the FP7 Information and Communication Technologies programme under Contract FET-215270 FRONTS, and in part by the Ateneo Italo-Tedesco under Program Vigoni.

## References

1. Atmel Corporation. Innovative IDIC solutions. [http://www.atmel.com/dyn/resources/prod\\_documents/doc4602.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc4602.pdf), 2007.
2. Gildas Avoine. Adversarial model for radio frequency identification. Cryptology ePrint Archive, Report 2005/049, 2005.
3. Gildas Avoine, Cedric Lauradoux, and Tania Martin. When compromised readers meet RFID. In *RFID-Sec*, 2009.
4. Leonid Bolotny and Gabriel Robins. Physically unclonable function-based security and privacy in RFID systems. In *Proceedings of PERCOM*, pages 211–220. IEEE Computer Society, 2007.
5. Mike Burmester, Tri van Le, and Breno de Medeiros. Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In *Proceedings of Second International Conference on Security and Privacy in Communication Networks (SecureComm)*, pages 1–9. IEEE Computer Society, 2006.
6. Ivan Damgård and Michael Østergaard. RFID security: Tradeoffs between security and efficiency. Cryptology ePrint Archive, Report 2006/234, 2006.
7. Paolo D’Arco, Alessandra Scafuro, and Ivan Visconti. Revisiting DoS Attacks and Privacy in RFID-Enabled Networks. In *Proceedings of ALGOSENSORS 2009*, volume 5804 of *LNCS*, pages 76–87. Springer-Verlag, July 2009.
8. Paolo D’Arco, Alessandra Scafuro, and Ivan Visconti. Semi-Destructive Privacy in DoS-Enabled RFID Systems. In *Proceedings of RFIDSec 2009*, July 2009.

9. Srinivas Devadas, Edward Suh, Sid Paral, Richard Sowell, Tom Ziola, and Vivek Khandelwal. Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications. In *IEEE International Conference on RFID*, pages 58–64. IEEE Computer Society, 2008.
10. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Proceedings of EUROCRYPT*, volume 3027 of *LNCS*, pages 523–540. Springer Verlag, 2004.
11. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. *Security with Noisy Data*, chapter Fuzzy Extractors, pages 79–99. Springer-Verlag, 2007.
12. EPCglobal Inc. Web site of EPCglobal Inc. <http://www.epcglobalinc.org/>, April 2008.
13. Blaise Gassend, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. Controlled physical random functions. In *Proceedings of ACSAC*, pages 149–160. IEEE Computer Society, 2002.
14. Daniel E. Holcomb, Wayne P. Bursleson, and Kevin Fu. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In *RFIDSec*, 2007.
15. I.C.A. Organization. Machine Readable Travel Documents, Doc 9303, Part 1 Machine Readable Passports, Fifth Edition, 2003.
16. Ari Juels. RFID security and privacy: A research survey. *Journal of Selected Areas in Communication*, 24(2):381–395, February 2006.
17. Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. Cryptology ePrint Archive, Report 2006/137, 2006.
18. Ilan Kirschenbaum and Avishai Wool. How to build a low-cost, extended-range RFID skimmer. Cryptology ePrint Archive, Report 2006/054, 2006.
19. David Molnar and David Wagner. Privacy and security in library RFID: Issues, practices, and architectures. In *Proceedings of ACMCCS*, pages 210–219. ACM Press, 2004.
20. Ching Yu Ng, Willy Susilo, Yi Mu, and Rei Safavi-Naini. RFID privacy models revisited. In *Proceedings of ESORICS*, volume 5283 of *LNCS*, pages 251–256. Springer Verlag, 2008.
21. NXP Semiconductors. MIFARE Application Directory (MAD) — List of Registered Applications. [http://www.nxp.com/acrobat/other/identification/mad\\_overview\\_042008.pdf](http://www.nxp.com/acrobat/other/identification/mad_overview_042008.pdf), April 2008.
22. NXP Semiconductors. MIFARE Smartcard ICs. <http://www.mifare.net/products/smartcardics/>, September 2008.
23. Silvio Micali Oded Goldreich, Shafi Goldwasser. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.
24. Radu-Ioan Païse and Serge Vaudenay. Mutual authentication in RFID: Security and privacy. In *Proceedings of ASIACCS*, pages 292–299. ACM Press, 2008.
25. Damith C. Ranasinghe, Daniel W. Engels, and Peter H. Cole. Security and privacy: Modest proposals for low-cost RFID systems. Auto-ID Labs Research Workshop, September 2004.
26. Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. User privacy in transport systems based on RFID e-tickets. International Workshop on Privacy in Location-Based Applications (PiLBA), Malaga, Spain, October 9, 2008, 2008.
27. Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Anonymizer-enabled security and privacy for rfid. In *The 8th International Conference in Cryptography and Network Security, December 12–14, 2009, Kanazawa, Ishikawa, Japan*, volume 5888 of *LNCS*, pages 134–153. Springer-Verlag, 2009.
28. Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Efficient RFID security and privacy with anonymizers. In *RFIDSec*, July 2009.
29. Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Location privacy in RFID applications. In *Privacy in Location-Based Applications — Research Issues and Emerging Trends*, volume 5599 of *LNCS*, pages 127–150. Springer-Verlag, August 2009.
30. Pim Tuyls and Lejla Batina. RFID-tags for anti-counterfeiting. In *proceedings of CT-RSA*, volume 3860 of *LNCS*, pages 115–131. Springer Verlag, 2006.
31. Pim Tuyls, Boris Škorić, and Tom Kevenaar, editors. *Security with Noisy Data — On Private Biometrics, Secure Key Storage, and Anti-Counterfeiting*. Springer-Verlag, 2007.
32. Serge Vaudenay. On privacy models for RFID. In *Proceedings of ASIACRYPT*, volume 4833 of *LNCS*, pages 68–87. Springer Verlag, 2007.
33. Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Security in Pervasive Computing, Revised Papers*, volume 2802 of *LNCS*, pages 50–59. Springer Verlag, 2003.

## A Proof of Theorem 1

Assume by contradiction that the protocol shown in Figure 2 does not achieve tag authentication. This means that there is an adversary  $\mathcal{A}_{\text{sec}}$  who can generate, with non-negligible

probability  $p$ , a protocol message  $(\tilde{b}, \tilde{c})$  for a given  $\tilde{a}$  such that  $\tilde{c} = F_{\tilde{K}}(\tilde{a}, \tilde{b})$  where  $(\tilde{\text{ID}}, \tilde{K}) \in \text{DB}$  without having made a **Corrupt** or **SendTag** $(\tilde{a}, \cdot)$  query to the tag  $\mathcal{T}_{\tilde{\text{ID}}}$ . In the following, we show that  $\mathcal{A}_{\text{sec}}$  can be transformed into a probabilistic polynomial time algorithm  $\mathcal{A}_{\text{prf}}$  that contradicts the security property of the underlying PRF  $F$  (Definition 1). Hence, the pseudo-randomness of  $F$  ensures that there is no such adversary  $\mathcal{A}_{\text{sec}}$ .

The construction of  $\mathcal{A}_{\text{prf}}$  is as follows: Given the security parameters  $l, \kappa, \alpha, \beta$  and a description of the PRF  $F$  from the PRF-challenger  $\mathcal{C}_{\text{prf}}$ ,  $\mathcal{A}_{\text{prf}}$  initializes the RFID system by first choosing  $\gamma$  polynomial in  $l$  and then setting  $sk_{\mathcal{R}} \leftarrow \emptyset, pk_{\mathcal{R}} \leftarrow (l, \gamma, \kappa, \alpha, \beta, F)$  and  $\text{DB} \leftarrow \emptyset$ . Then  $\mathcal{A}_{\text{prf}}$  guesses the identifier  $\tilde{\text{ID}}$  of the tag  $\mathcal{T}_{\tilde{\text{ID}}}$  that will be impersonated by  $\mathcal{A}_{\text{sec}}$ . Note that the probability of correctly guessing  $\tilde{\text{ID}}$  is polynomial since  $\mathcal{A}_{\text{sec}}$  can create at most a polynomial number of tags. Next,  $\mathcal{A}_{\text{prf}}$  initializes  $\mathcal{A}_{\text{sec}}$  with  $(l, \gamma, \kappa, \alpha, \beta, F)$  and simulates all the oracles defined in Section 5.2 to  $\mathcal{A}_{\text{sec}}$ :

- **CreateTag**(ID) If there already is a tuple  $(\text{ID}, \cdot, \cdot) \in \text{DB}$  or if  $\text{ID} = \tilde{\text{ID}}$ , then  $\mathcal{A}_{\text{prf}}$  aborts. Otherwise,  $\mathcal{A}_{\text{prf}}$  chooses  $S \in_R \{0, 1\}^\gamma$  and  $K \in_R \{0, 1\}^\kappa$  and updates  $\text{DB} \leftarrow \text{DB} \cup \{(\text{ID}, K, S)\}$ .
- **Draw, Free, Launch** The simulation of the **Draw, Free** and **Launch** oracle is straightforward. Note that  $\mathcal{A}_{\text{prf}}$  knows the secret look-up table  $\Gamma$  of the **Draw** oracle.
- **SendTag** $(a, vtag)$  If  $\Gamma[vtag] = \tilde{\text{ID}}$ , then  $\mathcal{A}_{\text{prf}}$  responds with  $b \in_R \{0, 1\}^\alpha$  and  $c \leftarrow \mathcal{O}^{F_{\tilde{K}}}(a, b)$ . Else,  $\mathcal{A}_{\text{prf}}$  gets  $(\Gamma[vtag], K, S)$  from  $\text{DB}$  and responds with  $b \in_R \{0, 1\}^\alpha$  and  $c \leftarrow F_K(a, b)$ .
- **SendReader** $(\emptyset, \pi)$  If  $\pi$  has been previously generated by a **Launch** oracle query and the corresponding protocol transcript is  $\text{tr}_\pi = \emptyset$ , then  $\mathcal{A}_{\text{prf}}$  returns  $a \in_R \{0, 1\}^\alpha$  and updates  $\text{tr}_\pi \leftarrow a$ .
- **SendReader** $((b, c), \pi)$  If  $\pi$  has been previously generated by a **Launch** oracle query and the corresponding protocol transcript is  $\text{tr}_\pi = a$ , then  $\mathcal{A}_{\text{prf}}$  updates the protocol transcript  $\text{tr}_\pi \leftarrow (a, b, c)$  and aborts otherwise.
- **Result** $(\pi)$  If  $\pi$  has been previously generated by a **Launch** oracle query and the corresponding protocol transcript  $\text{tr}_\pi = (a, b, c)$  has been obtained through  $a \leftarrow \text{SendReader}(\emptyset, \pi)$ , then  $\mathcal{A}_{\text{prf}}$  computes  $c' \leftarrow F_K(a, b)$  for each tuple  $(\text{ID}, K)$  in  $\text{DB}$ . If a  $c' = c$  for some  $(\text{ID}, K)$ ,  $\mathcal{A}_{\text{prf}}$  returns 1. If there is no  $c' = c$ , then  $\mathcal{A}_{\text{prf}}$  returns 0.
- **Corrupt** $(vtag)$  If there is a tuple  $(\Gamma[vtag], K, S)$  in  $\text{DB}$ ,  $\mathcal{A}_{\text{prf}}$  returns  $S$ . Note that according to Definition 6,  $\mathcal{A}_{\text{sec}}$  is not allowed to corrupt the tag  $\mathcal{T}_{\tilde{\text{ID}}}$  and hence,  $\mathcal{A}_{\text{prf}}$  needs not to simulate the **Corrupt** oracle for the tag  $\mathcal{T}_{\tilde{\text{ID}}}$ .

With non-negligible probability, after a polynomial number of oracle queries,  $\mathcal{A}_{\text{sec}}$  returns a protocol message  $(\tilde{b}, \tilde{c})$  for a given  $\tilde{a}$ . Next,  $\mathcal{A}_{\text{prf}}$  sends  $x \leftarrow (\tilde{a}, \tilde{b})$  to  $\mathcal{C}_{\text{prf}}$  who responds with a challenge  $y$ , which is either  $y = F_{\tilde{K}}(x)$  or  $y \in_R \{0, 1\}^\beta$ . In case  $y = \tilde{c}$ ,  $\mathcal{A}_{\text{prf}}$  returns 0 and 1 otherwise.

Note that in case  $b = 1$ ,  $\mathcal{A}_{\text{prf}}$  perfectly simulates all oracles defined in Section 5.2 to  $\mathcal{A}_{\text{sec}}$ . Hence, in case  $b = 1$ , by assumption  $\mathcal{A}_{\text{sec}}$  generates  $(\tilde{b}, \tilde{c})$  for any given  $\tilde{a}$  such that  $\tilde{c} = F_{\tilde{K}}(\tilde{a}, \tilde{b})$  holds with non-negligible probability. In return, this means that  $\mathcal{A}_{\text{prf}}$  has a non-negligible advantage of distinguishing the output of  $F$  and a randomly chosen value. Clearly, this contradicts the pseudo-randomness of the PRF  $F$  (Definition 1), which proves Theorem 1.

## B Proof of Theorem 2

According to Definition 7, destructive privacy means that there is a blinder  $\mathcal{B}$  that simulates the **Launch, SendTag, SendReader** and **Result** oracle such that no destructive adversary  $\mathcal{A}_{\text{prv}}$  (Definition 4) can distinguish between the blinder  $\mathcal{B}$  and the real oracles. Hence, to prove Theorem 2, we first give the construction of the blinder  $\mathcal{B}$  and then show that it cannot be distinguished from real oracles by any destructive adversary  $\mathcal{A}_{\text{prv}}$ .

The blinder  $\mathcal{B}$  is initialized with the security parameters  $l, \gamma, \kappa, \alpha, \beta$  and the public key  $pk_{\mathcal{R}}$  of the reader  $\mathcal{R}$  and works as follows:

- **Launch**() The simulation of the **Launch** oracle is straightforward.
- **SendTag**( $a, vtag$ ) Return  $b \in_R \{0, 1\}^\alpha$  and  $c \in_R \{0, 1\}^\beta$ .
- **SendReader**( $\pi$ ) Return  $a \in_R \{0, 1\}^\alpha$ .
- **SendReader**( $(b, c), \pi$ ) Since oracle queries of this form do not generate any output nor change the state of the tag and the reader, the blinder  $\mathcal{B}$  needs not to simulate their responses.
- **Result**( $\pi$ ) If  $\pi$  has been previously generated by a **Launch** oracle query and the corresponding protocol transcript  $\text{tr}_\pi = (a, b, c)$  has been generated by  $a \leftarrow \text{SendReader}(\emptyset, \pi)$  and  $(b, c) \leftarrow \text{SendTag}(a, vtag)$ , return 1 and 0 otherwise.

In the following, we show that if there is a destructive adversary  $\mathcal{A}_{\text{prv}}$  who can distinguish the blinder  $\mathcal{B}$  from the real oracles, then we can use  $\mathcal{A}_{\text{prv}}$  to construct a polynomial time algorithm that violates either tag authentication or the security properties of the underlying PRF  $F$  or the PUF  $P$ .

Let game  $\mathbf{G}^{(0)}$  be the game where the adversary  $\mathcal{A}_{\text{prv}}$  interacts with the real oracles as defined in Section 5.2. Now consider the following hybrid game  $\mathbf{G}^{(1)}$  that is exactly as  $\mathbf{G}^{(0)}$  with the only difference that the states  $S$  and the authentication secrets  $K$  of all tags are simulated by randomly chosen values. In the following, we show that if  $\mathcal{A}_{\text{prv}}$  can distinguish between  $\mathbf{G}^{(0)}$  and  $\mathbf{G}^{(1)}$ , then we can use  $\mathcal{A}_{\text{prv}}$  to construct a polynomial time algorithm  $\mathcal{A}_{\text{puf}}$  that contradicts the security property of the PUF  $P$  (Definition 2).

According to the protocol specification given in Section 6, the states and PUFs of different tags are chosen independently. Moreover,  $\mathcal{A}_{\text{puf}}$  can trivially simulate different tags by following the protocol specifications. Hence, we assume w.l.o.g. that  $\mathcal{A}_{\text{prv}}$  creates just one single tag  $\mathcal{T}_{\text{ID}}$  during his attack. To create this tag  $\mathcal{T}_{\text{ID}}$ ,  $\mathcal{A}_{\text{puf}}$  chooses  $S \in_R \{0, 1\}^\gamma$  and sets  $K \leftarrow \mathcal{O}^P(S)$ . Note that  $\mathcal{O}^P(S)$  either returns  $K \leftarrow P(S)$  as in  $\mathbf{G}^{(0)}$  or  $K \in_R \{0, 1\}^\kappa$  as in  $\mathbf{G}^{(1)}$ . Now,  $\mathcal{A}_{\text{puf}}$  can interact with all the oracles defined in Section 5.2 that are simulated by  $\mathcal{A}_{\text{puf}}$  based on the input of  $\mathcal{C}_{\text{puf}}$ . The simulation of the **Draw**, **Free** and **Launch** oracle is straightforward. Note that the output of the **Result** and **Corrupt** oracle is independent of the PUF of tag  $\mathcal{T}_{\text{ID}}$  and hence, these oracles can be simulated in a trivial way. Since **SendReader** queries generate no output and do not change the state  $S$  of the tag  $\mathcal{T}_{\text{ID}}$ , they need not be simulated by  $\mathcal{A}_{\text{puf}}$ . On a **SendTag**( $a, vtag$ ) oracle query,  $\mathcal{A}_{\text{puf}}$  responds with  $b \in_R \{0, 1\}^\alpha$  and  $c \leftarrow F_K(a, b)$ .

Note that  $\mathcal{A}_{\text{prv}}$  is a destructive adversary and hence, by making a **Corrupt**( $vtag$ ) query,  $\mathcal{A}_{\text{prv}}$  can obtain the state  $S$  of the tag  $vtag$  but he can no longer send any query that involves the tag  $vtag$  afterwards. After a polynomial number of oracle queries,  $\mathcal{A}_{\text{prv}}$  returns a bit  $b'$ . In case  $b' = 1$  (which indicates that  $\mathcal{A}_{\text{prv}}$  detected  $\mathcal{B}$ ), with non-negligible probability  $\mathcal{O}^P$  must have returned a random  $K \in_R \{0, 1\}^\kappa$ . Hence,  $\mathcal{A}_{\text{puf}}$  can distinguish the between the output of a PUF and a randomly chosen value, which contradicts the security property of the PUF (Definition 2). As a result, the following is negligible

$$|\Pr[\mathbf{G}^{(0)} = 1] - \Pr[\mathbf{G}^{(1)} = 1]|. \quad (1)$$

Next, consider the hybrid game  $\mathbf{G}^{(2)}$  that is exactly as  $\mathbf{G}^{(1)}$  with the only difference that the **SendTag** oracle is simulated by the blinder  $\mathcal{B}$  as described above. In the following, we show that if  $\mathcal{A}_{\text{prv}}$  can distinguish between  $\mathbf{G}^{(1)}$  and  $\mathbf{G}^{(2)}$ , then we can use  $\mathcal{A}_{\text{prv}}$  to construct a polynomial time algorithm  $\mathcal{A}_{\text{prf}}$  that contradicts the security property of the PRF  $F$  (Definition 1).

Let  $q \in \mathbb{N}$  be the number of **SendTag** queries made by  $\mathcal{A}_{\text{prv}}$ , which is polynomial in  $l$ . Moreover, let  $i \in \{0, \dots, q\}$ . Now consider the following hybrid game  $\mathbf{G}_i$  with  $\mathcal{A}_{\text{prv}}$ : The first  $i$  **SendTag** queries of  $\mathcal{A}_{\text{prv}}$  are answered by the blinder  $\mathcal{B}$  (as in  $\mathbf{G}^{(2)}$ ), while the remaining

$q - i$  queries are forwarded and answered by the real **SendTag** oracle (as in  $\mathbf{G}^{(1)}$ ). Note that  $\mathbf{G}_0$  corresponds to  $\mathbf{G}^{(1)}$  whereas  $\mathbf{G}_q$  corresponds to game  $\mathbf{G}^{(2)}$ . Hence, and due to the contradicting assumption made at the beginning of the proof, it holds that  $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}} = |\Pr[\mathbf{G}_0 = 1] - \Pr[\mathbf{G}_q = 1]|$  is non-negligible. Therefore, there must be some index  $i \in \{1, \dots, q\}$  such that

$$|\Pr[\mathbf{G}_{i-1} = 1] - \Pr[\mathbf{G}_i = 1]| \quad (2)$$

is non-negligible. Note that Equation 2 implies w.l.o.g. that  $\mathcal{A}_{\text{prv}}$  detects  $\mathcal{B}$  in game  $\mathbf{G}_i$  with non-negligible probability while only with negligible probability it can detect  $\mathcal{B}$  in game  $\mathbf{G}_{i-1}$ .

We can use  $\mathcal{A}_{\text{prv}}$  to construct the following polynomial time algorithm  $\mathcal{A}_{\text{prf}}$  that violates the security property of the PRF  $F$  (Definition 1). Therefore,  $\mathcal{A}_{\text{prf}}$  plays the hybrid game  $\mathbf{G}'_i$  with  $\mathcal{A}_{\text{prv}}$ , which is like  $\mathbf{G}_i$  except that the  $i$ -th **SendTag**( $a, vtag$ ) query is answered as follows:  $\mathcal{A}_{\text{prf}}$  chooses  $b \in_R \{0, 1\}^\alpha$  and sends  $x \leftarrow (a, b)$  to the PRF-challenger  $\mathcal{C}_{\text{prf}}$ , which responds with  $y \leftarrow \mathcal{O}^F(x)$  that is either  $y = F_K(x)$  or  $y \in_R \{0, 1\}^{2\alpha}$ . Then,  $\mathcal{A}_{\text{prf}}$  sends  $(b, c)$  to  $\mathcal{A}_{\text{prv}}$ . Note that, in case  $\mathcal{C}_{\text{prf}}$  sends  $y = F_K(x)$  then  $\mathbf{G}'_i = \mathbf{G}_{i-1}$  and  $\mathbf{G}'_i = \mathbf{G}_i$  otherwise. Hence, if  $\mathcal{A}_{\text{prv}}$  returns 1 (which indicates that  $\mathcal{A}_{\text{prv}}$  detected  $\mathcal{B}$ ) then  $\mathcal{A}_{\text{prf}}$  must have played  $\mathbf{G}_i$ . Clearly, this allows  $\mathcal{A}_{\text{prf}}$  to distinguish the output of the PRF  $F$  from a random value, which contradicts the security property of the PRF (Definition 1). Hence, the PRF ensures that Equation 2 is negligible and, as thus the following is negligible:

$$|\Pr[\mathbf{G}^{(1)} = 1] - \Pr[\mathbf{G}^{(2)} = 1]|. \quad (3)$$

Next, consider the hybrid game  $\mathbf{G}^{(3)}$  that is exactly as  $\mathbf{G}^{(2)}$  with the only difference that the **Result** oracle is simulated by the blinder  $\mathcal{B}$  as described above. In the following, we show that if there is an adversary  $\mathcal{A}_{\text{prv}}$  who can distinguish between  $\mathbf{G}^{(2)}$  and  $\mathbf{G}^{(3)}$ , then  $\mathcal{A}_{\text{prv}}$  can be used to construct a polynomial time algorithm  $\mathcal{A}_{\text{sec}}$  that contradicts tag authentication (Definition 6).

In the following, let  $p \in \mathbb{N}$  be the number of **Result** queries made by  $\mathcal{A}_{\text{prv}}$ , which is polynomial in  $l$ . Moreover, let  $i \in \{0, \dots, p\}$ . Now consider the following hybrid game  $\mathbf{G}^*_i$ : The first  $i$  **Result** queries of  $\mathcal{A}_{\text{prv}}$  are answered by the blinder  $\mathcal{B}$  (as in  $\mathbf{G}^{(3)}$ ), while the remaining  $p - i$  queries are forwarded and answered by the real **Result** oracle (as in  $\mathbf{G}^{(2)}$ ). Note that  $\mathbf{G}^*_0$  corresponds to  $\mathbf{G}^{(2)}$  whereas  $\mathbf{G}^*_p$  is equivalent to  $\mathbf{G}^{(3)}$ . Hence, and due to the contradicting assumption made at the beginning of the proof, it holds that  $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}} = |\Pr[\mathbf{G}^*_0 = 1] - \Pr[\mathbf{G}^*_p = 1]|$  is non-negligible. Therefore, there must be some index  $i \in \{1, \dots, p\}$  such that

$$|\Pr[\mathbf{G}^*_{i-1} = 1] - \Pr[\mathbf{G}^*_i = 1]| \quad (4)$$

is non-negligible. Note that Equation 4 implies that w.l.o.g.  $\mathcal{A}_{\text{prv}}$  detects  $\mathcal{B}$  in game  $\mathbf{G}^*_i$  with non-negligible probability while he has at most negligible probability to detect  $\mathcal{B}$  in game  $\mathbf{G}^*_{i-1}$ . This means that in  $\mathbf{G}^*_i$   $\mathcal{A}_{\text{prv}}$  runs a protocol instance  $\pi$  where the **Result** oracle simulated by  $\mathcal{B}$  returns a different output than the real **Result** oracle. According to the description of  $\mathcal{B}$  given at the beginning of this proof and the definition of the **Result** oracle in Section 5.2, this can only happen if  $\mathcal{A}_{\text{prv}}$  generates a protocol transcript  $\mathbf{tr}_\pi = (a, b, c)$  such that  $c = F_K(a, b)$  where  $(\text{ID}, K) \in \text{DB}$  and tag  $\mathcal{T}_{\text{ID}}$  has not been corrupted by  $\mathcal{A}_{\text{prv}}$ . However, as shown in the proof of Theorem 1 this can only happen with negligible probability. Hence, tag authentication ensures that Equation 5 is negligible and thus the following is negligible as well:

$$|\Pr[\mathbf{G}^{(2)} = 1] - \Pr[\mathbf{G}^{(3)} = 1]|. \quad (5)$$

Note that  $\mathbf{G}^{(3)}$  corresponds to the game where  $\mathcal{A}_{\text{prv}}$  interacts with a full blinder  $\mathcal{B}$ . Hence, from Equation 1, Equation 3 and Equation 5 it follows that  $|\Pr[\mathbf{G}^{(0)} = 1] - \Pr[\mathbf{G}^{(3)} = 1]|$  is negligible. This means that  $\mathcal{A}_{\text{prv}}$  cannot distinguish between the real oracles and the full blinder  $\mathcal{B}$ , which completes the proof of Theorem 2.