Enhancing RFID Security and Privacy by Physically Unclonable Functions

Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann

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. Moreover, 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.

Ahmad-Reza Sadeghi and Christian Wachsmann
Horst Görtz Institute for IT-Security (HGI), Ruhr-University Bochum, Germany
e-mail: {ahmad.sadeghi,christian.wachsmann}@trust.rub.de

Ivan Visconti
Dipartimento di Informatica ed Applicazioni, University of Salerno, Italy
e-mail: visconti@dia.unisa.it

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 [42, 2], access control [2, 47], electronic tickets [47] and passports [27], and even human implantation [30].

As pointed out in previous publications (see, e.g., [67, 30]), 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 (e.g., the secret) 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*). However, there are some other risks such as denial-of-service attacks, where an adversary unnoticeably interacts with tags and exploits deficiencies of the underlying protocols to permanently disable legitimate tags remotely [7], which must also be prevented (*availability*). In addition to the privacy and security requirements discussed above, RFID systems in practice must achieve various functional goals, including fast verification of cost-efficient tags (*efficiency*) and support of a huge number of tags (*scalability*). However, depending on the underlying application scenario and the given technological constraints, practical realizations may not be able to fulfill all of these requirements. In particular, the security and functional requirements often contradict the privacy requirements.

Most currently used RFID systems do not offer privacy at all (see, e.g., [62, 48, 47, 61]). This is mainly because current cost-efficient tags do not provide the necessary computational resources to run privacy-preserving protocols [2, 47], which heavily rely on public-key cryptography. Moreover, as pointed out in Section 3, privacy-preserving solutions without public-key cryptography do not fulfill important security or functional requirements and thus, are inapplicable to real-world applications.

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., [3, 33, 8, 7]) often do not consider important aspects like adversaries with access to auxiliary information (on whether the identification of a tag was successful or not) 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 [66] and refined in [45, 53]. In the following, we refer to the

privacy model of [66] as the *V-Model* (Vaudenay Model). The V-Model [66] 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. However, [66] showed that narrow-strong privacy requires the use of public-key cryptography [66], which in general clearly exceeds the capabilities of current cost-efficient RFIDs [2, 47]. Moreover, it has been shown that forward privacy can be achieved but at the cost of using public-key cryptography while the feasibility of the stronger notion of destructive privacy currently is an open question [66].

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 V-Model [66]. 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 [66] and uses Physically Unclonable Functions (PUFs) as tamper-evident key storage in a similar way as described in [64]. 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 V-Model [66].

2 High-Level RFID System and Requirement Analysis

We first informally analyze the general scenario of Radio Frequency Identification (RFID) on a very high level.

System Model

An RFID system consists of at least an operator $\mathscr{I}$, a reader $\mathscr{R}$ and a tag $\mathscr{T}$ [17]. The operator $\mathscr{I}$ is the entity that enrolls and maintains the RFID system. Hence, $\mathscr{I}$ initializes each tag $\mathscr{T}$ and reader $\mathscr{R}$ before it is deployed in the system. A tag $\mathscr{T}$ or reader $\mathscr{R}$ that has been initialized by the operator $\mathscr{I}$ is called *legitimate*. A tag $\mathscr{T}$ is a hardware token with constrained computing and memory capabilities that is equipped with a radio interface [17, 2, 47]. All information (e.g., secrets and data) that is stored on a tag $\mathscr{T}$ is denoted as the *state* of $\mathscr{T}$. Usually, tags are attached to objects or carried by the users of the RFID system [16, 46]. A reader $\mathscr{R}$ is a stationary or mobile computing device that interacts with all tags within its read-

ing range to authenticate them. Depending on the specific use case (e.g., electronic passports [27]), the reader $\mathscr{R}$ may obtain additional information like the tag identity or some data stored on the tag $\mathscr{T}$. Readers can have a sporadic or permanent online connection to some backend system $\mathscr{D}$, which typically is a database maintaining detailed information on all tags in the system [15]. The backend is initialized and maintained by the operator $\mathscr{I}$ and can be read and updated by the readers $\mathscr{R}$.

Trust and Adversary Model

The operator $\mathscr{I}$ is the entity that maintains the RFID system, and thus can be considered to be honest. However, $\mathscr{I}$ may be curious since he may collect user information (see, e.g., [67, 29]) while in general at the same time nobody can blame him for cheating.

Since RFID tags and readers communicate over a radio link, every entity can eavesdrop or manipulate this communication, even from outside the nominal reading range [37]. Thus, the adversary can be every (potentially unknown) entity that needs not to be a member of the RFID system. Besides the communication between a tag $\mathscr{T}$ and a reader $\mathscr{R}$, an adversary can also obtain useful auxiliary information (e.g., by visual observation) on whether the reader $\mathscr{R}$ accepted the tag $\mathscr{T}$ [33, 66]. Most commercial RFID tags are cost-efficient devices without (expensive) protection mechanisms against physical tampering [2, 47]. Hence, an adversary in practice can physically attack (*corrupt*) a tag to access its state (e.g., its secrets) [38, 39, 41, 26].

RFID readers are embedded devices that can be integrated into mobile devices (e.g., mobile phones or PDAs) or laptops and personal computers. The resulting complexity exposes readers to sophisticated hard- and software attacks (e.g., viruses or Trojans). Hence, an adversary in practice can get full control of (*corrupt*) an RFID reader [5]. This problem aggravates for mobile readers that can easily be lost or stolen.

Security and Privacy Threats

The most deterrent privacy risk concerns the *tracking* of users, which allows the creation and misuse of detailed profiles of a user of the RFID system [30]. For instance, tracking or identification of a tag enables the creation of detailed movement profiles, which can leak sensitive information on the personal habits and interests of the tag user.

A major security risk concerns adversaries who trick an honest reader to accept illegitimate tags. The main threats are to create faked (illegitimate) tags that are accepted by legitimate readers (*forgery*) and to simulate (*impersonate*) or to copy (*clone*) legitimate tags. Another threat concerns attacks that permanently prevent users from using the RFID system (*denial-of-service*) [7].

Security and Privacy Objectives

Based on the discussion in the previous paragraphs, we consider RFID systems that provide *anonymity* as well as *untraceability* even when the state of (i.e., the data stored on) a tag has been disclosed. Anonymity means the confidentiality of the tag identity whereas untraceability refers to the unlinkability of the tag communication. To distinguish tracing in past or future protocol-runs, the notions of *forward untraceability* and *backward untraceability* are defined in [40]. In use cases like electronic passports, where tags store privacy-sensitive data, *reader authentication* is an additional goal to prevent disclosure of this data to illegitimate readers.

The major security objective of an RFID system is to ensure that only legitimate tags are accepted by legitimate readers (*tag authentication*). Hence, the reader must be able to distinguish between legitimate and illegitimate tags. Most use cases additionally require the reader to be capable of determining the (authentic) tag identity (*tag identification*).

3 Related Work

Privacy-Preserving RFID Protocols

A general problem with privacy-preserving authentication of low-cost tags that are incapable of public-key operations is how to inform the reader which key should be used for the authentication. Indeed, a tag cannot disclose its identity before the reader has been authenticated since this would violate untraceability. However, a reader cannot authenticate a tag unless it knows the identity (i.e., the key) of that tag. Essentially there are three approaches that address this problem.

The first approach is that the reader performs an exhaustive search for the secret key that is used by the authenticating tag [67]. Solutions to optimize this approach (see, e.g., [42, 63]) suffer from inefficiency since tag verification depends on the total number of tags in the system. Clearly, this violates the efficiency and scalability requirements of most practical RFID systems. A prominent family of lightweight authentication protocols in this context are the HB protocols (see, e.g., [32, 35, 36, 34]). These protocols are subject to man-in-the-middle attacks [20, 21, 52, 18], require the reader to perform an exhaustive search for the authentication secret of the authenticating tag and usually require many rounds of interaction [68]. Moreover, tag corruption is usually not considered in the security evaluation of the HB protocols.

In the second approach, a tag updates its identity after each interaction such that its new identity is unlinkable and only known to the tag and the authorized readers, which allows readers to identify tags in constant time (see, e.g., [24, 51, 12, 40, 60]). However, this approach requires each tag to be always synchronized with all readers in the system. In general, it is easy to mount denial-of-service attacks that desynchronize the tag and the readers (see e.g., [24, 12]).

Another approach to enhance the privacy of RFID systems without lifting the computational requirements on tags are anonymizer-enabled protocols, where external devices (*anonymizers*) are in charge of providing anonymity of tags (see, e.g., [31, 22, 58, 1, 55, 56]). The main concept of anonymizer-enabled protocols is that each tag stores a ciphertext that encrypts the information carried by the tag (e.g., the tag identifier) under the public key of the reader. This ciphertext is transmitted to the reader in the tag authentication protocol. Since this ciphertext is static data that can be used to track and to identify the tag, it must be frequently changed to provide anonymity and untraceability. However, current RFIDs [2, 47] are not capable of updating this public-key encrypted ciphertext on their own and thus, privacy in these protocols relies on third parties, called anonymizers, that frequently refresh the ciphertexts stored on the tags. Most anonymizer-enabled RFID systems are subject to impersonation attacks since tag authentication is only based on the ciphertext that the tag sends to the reader. Moreover, existing security models do not capture RFID systems that use anonymizers. The authors of [57] address these issues and propose an anonymizer-enabled RFID system that provides untraceability, tag authentication and basic availability along with a general security and privacy framework for anonymizer-enabled RFID systems that is based on the security and privacy model of [66].

For a broad overview about privacy issues in RFID systems, see also [57].

RFID Protocols based on Physically Unclonable Functions

To prevent cloning of a tag it must be infeasible to determine its authentication secret by both attacking the corresponding authentication protocols as well as by physically attacking the tag. One solution to counterfeit cloning attacks is to employ physical protection mechanisms that aggravate reading out the memory of a tag [59, 43]. However, this would dramatically increase the price of tags and render them inappropriate for most commercial applications. A more economic solution to prevent cloning can be implemented by using physically unclonable functions (PUFs) [64, 19].

A PUF consists of an inherently unclonable noisy function P that is embedded into a physical object [65]. 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* $r' = \mathsf{P}(c)$ 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*, that (up to a certain threshold) maps different responses r' to a unique value r for each specific challenge c [14]. 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 [19]. These deviations are unique for every sample from a set of chips (even from the same lot or wafer) that implement the same circuit.

Physically unclonable functions 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. They provide cost-effective and practical tamper-evident storage for cryptographic secrets that even cannot be learned or reproduced by the manufacturer of the corresponding PUF (as long as the manufacturer produced the PUF following the prescribed procedure).

One of the first proposals of using PUFs in RFID systems is introduced by [54]. 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 database and the time required to measure the reference responses for the database. This scheme has been implemented by [11] who provide a realization of PUF-enabled RFID tags and analyze their security and usability. The authors of [25] propose a similar approach based on the physical characteristics of SRAM cells. The advantage of this approach is that SRAM-PUFs can be implemented using the existing SRAM memory cells of the RFID chip without the need for additional hardware.

In [64], 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 [19]. According to [64], a PUF-based key storage can be implemented with less than 1000 gates. However, their authentication scheme relies on public-key cryptography, which is still much too expensive for current low-cost RFID tags.

The authors of [6] follow the approach of frequently updating the identity of tags to provide privacy (see Section 3) and suggest to use PUFs instead of pseudorandom functions. They propose to equip each 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 $(\mathscr{T}_0, \mathscr{T}_1, \ldots, \mathscr{T}_m)$ for each legitimate tag $\mathscr{T}$ where $\mathscr{T}_0$ is a random tag identifier and $\mathscr{T}_{i+1} = \mathsf{P}(\mathscr{T}_i)$ for $i \in \{0, \ldots, m-1\}$. To authenticate to a reader, a tag first sends its current identifier $\mathscr{T}_i$ and then updates its identity to $\mathscr{T}_{i+1} \leftarrow \mathsf{P}(\mathscr{T}_i)$. The reader then checks whether there is a tuple that contains a value $\mathscr{T}_i$ in the database. In case the reader finds $\mathscr{T}_i$, it accepts the tag and

invalidates all previous database entries $\mathscr{T}_j$ 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.

Privacy Models for RFID

One of the first privacy models for RFID [50] defines anonymity and backward untraceability based on a security game where an adversary must distinguish a random value from the output of a tag. However, it does not consider forward untraceability. A privacy model specific for RFIDs that cannot perform any cryptographic operations [29] is based on assumptions on the number of queries an adversary can make to a tag but does not capture adversaries who can corrupt tags. Thus, it does not cover backward and forward untraceability, which is required to realistically model adversaries against cost-efficient tags in practice. Another privacy model [3, 4] provides various flexible definitions for different levels of privacy based on a security experiment where an adversary must distinguish two known tags. This model is extended in [33] by the notion of auxiliary information. In [8], a *completeness* and *soundness* requirement is added to the definition of [33], which means that a reader must accept *all* but *only* valid tags. The definition of [33] has been further improved in [23] to cover backwards untraceability. Another privacy model [7] is based on the universal composability (UC) framework and claims to be the first model that considers availability. However, it does not allow the adversary to corrupt tags and does not capture backwards untraceability. Recently, [66] presented a privacy model that generalizes and classifies previous RFID privacy models by defining eight levels of privacy that correspond to real-world adversaries of different strength. The strongest privacy notion of [66] captures anonymity, backward and forward untraceability and adversaries with access to auxiliary information. Moreover, it provides a security definition equivalent to [8] that covers tag authentication. The model of [66] has been extended in [53] to consider reader authentication whereas [45] aims at reducing the mentioned eight privacy classes to three privacy classes. Recently in [10, 9] other privacy notions have been considered along with denial of service attacks. The authors of [44] use the framework of [66, 53] to classify and to examine the privacy properties of various existing symmetric-key-based authentication protocols for RFID and show several impossibility results for this class of protocols.

4 RFID Security and Privacy Model of Vaudenay [66]

In this section, we review the RFID security and privacy model proposed by Vaudenay (V-Model) [66], which is one of the most comprehensive RFID privacy and security models up to date. We start by setting the notation that will be used later

and then give a fairly detailed and at the same time more formal specification of the V-Model [66].

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 $y \leftarrow \mathsf{A}(x)$ means that on input x, algorithm A assigns its output to variable y. The term $[\mathsf{A}(x)]$ denotes the set of all possible outputs of A on input x. $\mathsf{A}_K(x)$ means that the output of A depends on x and some additional parameter K (e.g., a secret key). The term $\mathsf{Prot}[\mathsf{A}: x_\mathsf{A};\ \mathsf{B}: x_\mathsf{B};\ *: x_{pub}] \rightarrow [\mathsf{A}: y_\mathsf{A};\ \mathsf{B}: y_\mathsf{B}]$ denotes an interactive protocol 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 $\varepsilon(l)$ is called *negligible* if for all polynomials $f(\cdot)$ it holds that $\varepsilon(l) \leq 1/f(l)$ for all sufficiently large l. Probability $1-\varepsilon(l)$ is called *overwhelming* if $\varepsilon(l)$ 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 $F : \{0,1\}^{\kappa+\alpha} \longrightarrow \{0,1\}^{\beta}$ be a family of functions. Consider the following security experiment $\mathbf{Exp}^{\text{prf-}b}_{\mathscr{A}_{\text{prf}}}$, where an adversary $\mathscr{A}_{\text{prf}}$ interacts with a *PRF-challenger* $\mathscr{C}_{\text{prf}}$: When initialized with l, κ, α, β and $b \in_R \{0,1\}$, $\mathscr{C}_{\text{prf}}$ chooses $K \in_R \{0,1\}^{\kappa}$ and initializes an oracle $\mathscr{O}^{F_K}$ 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 $\mathscr{O}^{F_K}$, $\mathscr{A}_{\text{prf}}$ then must return a bit b'. $\mathscr{A}_{\text{prf}}$ wins the security experiment if $b = b'$.

Definition 1 (Pseudo-Random Function [49]). 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 adversary $\mathscr{A}_{\text{prf}}$ has at most negligible advantage: $\mathbf{Adv}^{\text{prf}}_{\mathscr{A}_{\text{prf}}} = \left| \Pr\left[\mathbf{Exp}^{\text{prf-1}}_{\mathscr{A}_{\text{prf}}} = 1\right] - \Pr\left[\mathbf{Exp}^{\text{prf-0}}_{\mathscr{A}_{\text{prf}}} = 1\right] \right|$.

Physicaly Unclonable Function (PUF)

A physically unclonable function is an inherently unclonable noisy function that is embedded into a physical object (e.g., an integrated circuit) [65]. When challenged with a stimulus (*challenge*), a PUF generates an output (*response*) that depends on both the challenge and the physical properties of the object containing the PUF. However, the physical object is subject to noise (e.g., temperature and/or supply voltage variations) and hence, when queried with the same challenge multiple times, the PUF will always return slightly different responses. To eliminate these output variations *Fuzzy Extractors* [13, 14] can be used.

To the best of our knowledge, currently there is no widely accepted security model for PUFs. Moreover, setting up a model that realistically 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 $\mathsf{P} : \{0,1\}^{\gamma} \longrightarrow \{0,1\}^{\kappa}$ be a function. Consider the following security experiment $\mathbf{Exp}^{\text{puf-}b}_{\mathscr{A}_{\text{puf}}}$ that is similar to $\mathbf{Exp}^{\text{prf-}b}_{\mathscr{A}_{\text{prf}}}$ described above. The difference is that, when initialized with l, γ, κ and $b \in_R \{0,1\}$, the PUF-challenger $\mathscr{C}_{\text{puf}}$ initializes an oracle $\mathscr{O}^{\mathsf{P}}$ 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 $\mathscr{O}^{\mathsf{P}}$, $\mathscr{A}_{\text{puf}}$ must return a bit b'. $\mathscr{A}_{\text{puf}}$ wins the security experiment if $b = b'$.

Definition 2 (Ideal PUF). An ideal PUF is a function P with the following properties:

1. For all $c \in \{0,1\}^{\gamma}$ and all tuples $(r_i, r_j) \in \left[\mathsf{P}(c)\right]^2$, probability $\Pr[r_i = r_j] = 1$.
2. In the above experiment, any probabilistic polynomial time adversary $\mathscr{A}_{\text{puf}}$ has at most negligible advantage: $\mathbf{Adv}^{\text{puf}}_{\mathscr{A}_{\text{puf}}} = \left|\Pr\left[\mathbf{Exp}^{\text{puf-1}}_{\mathscr{A}_{\text{puf}}} = 1\right] - \Pr\left[\mathbf{Exp}^{\text{puf-0}}_{\mathscr{A}_{\text{puf}}} = 1\right]\right|$.
3. Any attempt to physically tamper with the object implementing P results in destruction of P, i.e., 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 in practice by using Fuzzy Extractors [13, 14]. 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. According to the third property of Definition 2, the adversary cannot obtain any information about the PUF by physical means, which entirely

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.

4.1 System Model

As most RFID privacy models, the V-Model [66] considers RFID systems that consist of one single operator $\mathscr{I}$, one single reader $\mathscr{R}$ and a polynomial number of tags $\mathscr{T}$. The reader $\mathscr{R}$ is assumed to be capable of performing public-key cryptography and of handling multiple instances of the tag identification protocol with different tags in parallel. Tags are passive devices, i.e., they do not have own power supply but are powered by the electromagnetic field of the reader $\mathscr{R}$. Hence, tags cannot initiate communication and have a narrow communication range (of a few centimeters to meters). Tags are assumed to be capable of performing basic cryptographic operations like hashing, random number generation and symmetric-key encryption.

Trust and Adversary Model

In the V-Model [66], the issuer $\mathscr{I}$, the backend $\mathscr{D}$ and the readers $\mathscr{R}$ are assumed to be trusted. Therefore, these entities will behave as intended. All the readers $\mathscr{R}$ and the backend $\mathscr{D}$ are subsumed to *one single* reader entity $\mathscr{R}$. This implies that all readers $\mathscr{R}$ are assumed to be tamper-resistant devices that have a permanent secure online connection to a database $\mathscr{D}$. Tags are considered to be untrusted, which means that the adversary can obtain their state (i.e., all the data stored on them).

The adversary can eavesdrop and manipulate the communication channel between a tag $\mathscr{T}$ and the reader $\mathscr{R}$. The V-Model [66] defines eight adversary classes that differ in their ability to corrupt tags and the availability of auxiliary information (see Section 4.2). Hence, depending on the adversary class, the adversary is subject to different restrictions concerning tag corruption. At this point we would like to stress that the V-Model [66] does not pose any limitation regarding corruption of a tag $\mathscr{T}$ while the tag $\mathscr{T}_{\mathtt{ID}}$ is involved in the authentication protocol with the reader $\mathscr{R}$. However, the adversary is not allowed to corrupt the reader $\mathscr{R}$.

Security and Privacy Objectives

The main security goal of the V-Model [66] is tag authentication. More precisely, a legitimate reader $\mathscr{R}$ should only accept legitimate tags and must be able to identity them. Reader authentication, availability and protection against cloning are not captured by the V-Model [66]. The privacy objectives are anonymity and unlinkability.

Protocol Definitions

The operator $\mathscr{I}$ sets up the reader $\mathscr{R}$ and all tags $\mathscr{T}$. Hence, there are two setup protocols where $\mathscr{R}$ and the tags $\mathscr{T}$ are initialized and their system parameters (e.g., keys) are generated and defined. A third protocol between a tag $\mathscr{T}$ and $\mathscr{R}$ covers tag authentication. More formally, the RFID system model of [53] is defined as follows:

Definition 3 (RFID System [53]). An RFID system RFID is a tuple of probabilistic polynomial time algorithms $(\mathscr{R}, \mathscr{T}, \mathsf{SetupReader}, \mathsf{SetupTag}, \mathsf{Ident})$ that are specified as follows:

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

$\mathsf{SetupTag}_{pk_{\mathscr{R}}}(\mathtt{ID}) \to (K, S)$ Creates a tag $\mathscr{T}_{\mathtt{ID}}$, which is an instance of the tag algorithm $\mathscr{T}$. Hereby, the public key $pk_{\mathscr{R}}$ of $\mathscr{R}$ is used to generate a secret K and an initial tag state S. $\mathscr{T}_{\mathtt{ID}}$ is initialized with S and $(\mathtt{ID}, K)$ is stored in the credentials database $\mathtt{DB}$ of $\mathscr{R}$.

$\mathsf{Ident}[\mathscr{T}_{\mathtt{ID}} : S;\ \mathscr{R} : sk_{\mathscr{R}}, \mathtt{DB};\ * : pk_{\mathscr{R}}] \to [\mathscr{T}_{\mathtt{ID}} : -;\ \mathscr{R} : out_{\mathscr{R}}]$ This is an interactive protocol between a tag $\mathscr{T}_{\mathtt{ID}}$ and the reader $\mathscr{R}$. $\mathscr{T}_{\mathtt{ID}}$ takes as input its current state S while $\mathscr{R}$ has as input its secret key $sk_{\mathscr{R}}$ and the credentials database $\mathtt{DB}$. The common input to all parties is the public key $pk_{\mathscr{R}}$ of $\mathscr{R}$. After the protocol terminates, $\mathscr{R}$ returns either the identity $\mathtt{ID}$ of $\mathscr{T}_{\mathtt{ID}}$ or $\perp$ to indicate that $\mathscr{T}_{\mathtt{ID}}$ is not a legitimate tag.

4.2 Adversary Model

In the V-Model [66], the privacy and security objectives are defined as security experiments, where a polynomially bounded adversary can interact with a set of oracles that model the capabilities of the adversary. These oracles are:

$\mathsf{CreateTag}^b(\mathtt{ID})$ This oracle allows the adversary to set up a tag $\mathscr{T}_{\mathtt{ID}}$ with identifier $\mathtt{ID}$ by internally calling $\mathsf{SetupTag}_{pk_{\mathscr{R}}}(\mathtt{ID})$ to create (K, S) for $\mathscr{T}_{\mathtt{ID}}$. If input $b = 1$, the adversary chooses $\mathscr{T}_{\mathtt{ID}}$ to be legitimate, which means that $(\mathtt{ID}, K)$ is added to the credentials database $\mathtt{DB}$ of $\mathscr{R}$. For input $b = 0$, the adversary chooses $\mathscr{T}_{\mathtt{ID}}$ to be illegitimate and $(\mathtt{ID}, K)$ is *not* added to $\mathtt{DB}$.[1] This models the fact that an adversary can obtain (e.g., buy) legitimate tags and create forgeries.

[1] Note that illegitimate tags created by the $\mathsf{CreateTag}$ oracle are initialized in the same way as legitimate tags with the only difference that their identifier $\mathtt{ID}$ and secret K is not added to the credentials database $\mathtt{DB}$ of $\mathscr{R}$. As shown in [66], an adversary can use such tags to violate the privacy objectives.

$\mathsf{Draw}(\delta) \to (vtag_1, b_1, \ldots, vtag_n, b_n)$ Initially, the adversary 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 δ. This models the fact that the adversary can only interact with the tags within his reading range. The adversary usually only knows the tags it can interact with by some temporary tag identifiers $vtag_1, \ldots, vtag_n$. The Draw oracle manages a secret look-up table Γ that keeps track of the real tag identifier $\mathtt{ID}_i$ that is associated with each temporary tag identifier $vtag_i$ (i.e., $\Gamma[vtag_i] = \mathtt{ID}_i$). Moreover, the Draw oracle also provides to the adversary the information on whether the tags are legitimate ($b_i = 1$) or not ($b_i = 0$).

$\mathsf{Free}(vtag)$ Contrary to Draw, the Free oracle makes a tag *vtag* inaccessible to the adversary. This means that the adversary cannot interact with the tag *vtag* any longer until it is made accessible again (under a new temporary identifier $vtag'$) by another Draw query. This models the fact that a tag can get out of the reading range of the adversary.

$\mathsf{Launch}() \to \pi$ Makes the reader $\mathscr{R}$ to start a new instance π of the Ident protocol, which allows the adversary to start different concurrent Ident protocol instances with the reader $\mathscr{R}$.

$\mathsf{SendReader}(m, \pi) \to m'$ Sends a message m to the instance π of the Ident protocol that is running on the reader $\mathscr{R}$. The reader $\mathscr{R}$ interprets m as a protocol message of instance π of the Ident protocol and responds with a message m'. This allows the adversary to perform active attacks on the Ident protocol.

$\mathsf{SendTag}(m, vtag) \to m'$ Sends a message m to the tag $\mathscr{T}_{\mathtt{ID}}$ that is known as *vtag* to the adversary. $\mathscr{T}_{\mathtt{ID}}$ interprets m as a protocol message of the Ident protocol and responds with a message m'. This allows the adversary to perform active attacks on the Ident protocol.

$\mathsf{Result}(\pi)$ Returns 1 if the instance π of the Ident protocol has been completed and the tag $\mathscr{T}_{\mathtt{ID}}$ that participated in this instance π has been accepted by the reader $\mathscr{R}$. In case $\mathscr{R}$ identified an illegitimate tag, Result returns 0. This allows the adversary to obtain auxiliary information on whether the authentication of $\mathscr{T}_{\mathtt{ID}}$ was successful or not.

$\mathsf{Corrupt}(vtag) \to S$ Returns the current state S of the tag $\mathscr{T}_{\mathtt{ID}}$ that is known as *vtag* to the adversary. This models (physical) attacks on $\mathscr{T}_{\mathtt{ID}}$ that disclose the current tag state S (i.e., all information stored on or used by $\mathscr{T}_{\mathtt{ID}}$ at the time of corruption) to the adversary.

The V-Model [66] distinguishes the following adversary classes, which differ in their ability to corrupt tags and the availability of auxiliary information (i.e., the ability to call the Corrupt and the Result oracle):

- *Weak adversaries* cannot corrupt tags and are limited to eavesdropping and manipulating the communication between the tags and the reader.
- *Forward adversaries* can obtain the state of the tags only as the last interaction with the oracles defined above. This means that after having corrupted a tag for the first time, a forward adversary can no longer observe any protocol execution

or interact with any tag or reader. However, he can still corrupt all remaining non-corrupted tags.
- *Destructive adversaries* cannot reuse a tag after corrupting it. This means that a destructive adversary cannot observe or interact with a corrupted tag nor can he impersonate the corrupted tag to the reader. However, he can still observe and interact with any non-corrupted tag.[2]
- *Strong adversaries* are not restricted in their ability to corrupt tags.

Moreover, the V-Model [66] defines *narrow* variants of the four adversary classes described above (i.e., narrow-weak, narrow-forward, narrow-destructive and narrow-strong). In addition to the restrictions concerning tag corruption of the corresponding adversary class, a narrow adversary cannot obtain auxiliary information from the communication between the tags and the reader.

Definition 4 (Adversary Classes [66]). An adversary is a probabilistic polynomial time algorithm that has arbitrary access to all of the oracles described in Section 4.2. Weak adversaries cannot access the Corrupt oracle. Forward adversaries can no longer query any other oracle than Corrupt after they made the first query to the Corrupt oracle. Destructive adversaries cannot query any oracle for *vtag* again after they made a $\mathsf{Corrupt}(vtag)$ query. Strong adversaries have no restrictions 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 V-Model [66].

4.3 Definition of Correctness, Security and Privacy

Correctness

Correctness describes the honest behavior of legitimate tags $\mathscr{T}$ and the reader $\mathscr{R}$. With overwhelming probability, the reader $\mathscr{R}$ returns $out_{\mathscr{R}} = \mathtt{ID}$ when interacting with a legitimate tag $\mathscr{T}_{\mathtt{ID}}$ and $out_{\mathscr{R}} = \perp$ otherwise. More formally:

Definition 5 (Correctness [53]). An RFID system RFID as defined in Definition 3 is correct if for every $l \in \mathbb{N}$, every $(sk_{\mathscr{R}}, pk_{\mathscr{R}}, \mathtt{DB}) \in [\mathsf{SetupReader}(1^l)]$ and every $(K,S) \in [\mathsf{SetupTag}_{pk_{\mathscr{R}}}(\mathtt{ID})]$ it holds with overwhelming probability that

$$\mathsf{Ident}[\mathscr{T}_{\mathtt{ID}} : S;\ \mathscr{R} : sk_{\mathscr{R}}, \mathtt{DB};\ * : pk_{\mathscr{R}}] \rightarrow [\mathscr{T}_{\mathtt{ID}} : -;\ \mathscr{R} : \mathtt{ID}]\,.$$

[2] Note that, in case of PUF-enabled RFID tags, a destructive adversary can corrupt the tag and read out its memory whereas the properties of the PUF ensure that the PUF is destroyed and the adversary does not obtain any information on the PUF.

Security Definition

The security definition given by the V-Model [66] 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 $\mathbf{Exp}^{\mathscr{T}\text{-auth}}_{\mathscr{A}_{\text{sec}}}$ where a strong adversary $\mathscr{A}_{\text{sec}}$ must make the reader $\mathscr{R}$ to identify some tag $\mathscr{T}_{\mathtt{ID}}$ in some instance π of the Ident protocol. To exclude trivial attacks (e.g., relay attacks), $\mathscr{A}_{\text{sec}}$ is not allowed to simply forward all the messages from $\mathscr{T}_{\mathtt{ID}}$ to $\mathscr{R}$ in instance π nor to corrupt $\mathscr{T}_{\mathtt{ID}}$. This means that at least some of the protocol messages that made $\mathscr{R}$ to return ID must have been partly computed by $\mathscr{A}_{\text{sec}}$ without knowing the secrets of $\mathscr{T}_{\mathtt{ID}}$. With $\mathbf{Exp}^{\mathscr{T}\text{-auth}}_{\mathscr{A}_{\text{sec}}} = 1$ we denote the case where $\mathscr{A}_{\text{sec}}$ wins the security experiment.

Definition 6 (Tag Authentication [53]). An RFID system (Definition 3) achieves tag authentication if for every strong adversary $\mathscr{A}_{\text{sec}}$ $\Pr[\mathbf{Exp}^{\mathscr{T}\text{-auth}}_{\mathscr{A}_{\text{sec}}} = 1]$ is negligible.

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

Privacy Definition

The privacy definition of the V-Model [66] is very flexible and, dependent on the adversary class considered (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 $\mathscr{R}$. It is based on the existence of a simulator $\mathscr{B}$, called *blinder*, that can simulate the tags and the reader $\mathscr{R}$ without knowing any of their secrets such that an adversary $\mathscr{A}_{\text{prv}}$ 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 the tags with the reader $\mathscr{R}$ does not leak any information about the tags. Hence, everything the adversary $\mathscr{A}_{\text{prv}}$ observes from the interaction with the tags and the reader $\mathscr{R}$ appears to be independent of the tags and consequently, $\mathscr{A}_{\text{prv}}$ cannot distinguish different tags based on their communication, which corresponds to unlinkability.

In the following, we express this privacy definition in a more formal way by a privacy experiment $\mathbf{Exp}^{\text{prv-}b}_{\mathscr{A}_{\text{prv}}}$. Let $\mathscr{A}_{\text{prv}}$ 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, the reader $\mathscr{R}$ is initialized with $(sk_{\mathscr{R}}, pk_{\mathscr{R}}, \mathtt{DB}) \leftarrow \mathsf{SetupReader}(1^l)$. The public key $pk_{\mathscr{R}}$ is given to $\mathscr{A}_{\text{prv}}$ and to the simulator $\mathscr{B}$. Now, $\mathscr{A}_{\text{prv}}$ is allowed to arbitrarily interact with all oracles defined in Section 4.2. Hereby, $\mathscr{A}_{\text{prv}}$ 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 the simulator $\mathscr{B}$. Hereby, $\mathscr{B}$ can observe all queries $\mathscr{A}_{\text{prv}}$ makes to all the other oracles that are not simulated by $\mathscr{B}$ and the corresponding responses ("$\mathscr{B}$ sees what $\mathscr{A}_{\text{prv}}$ sees"). After a polynomial number of oracle queries, the second

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

Fig. 1 Privacy notions defined in the PV-Model [53] and their relations.

phase of the experiment starts. In this second stage, $\mathscr{A}_{\text{prv}}$ can no longer interact with the oracles but is given the hidden table Γ of the Draw oracle. Finally, $\mathscr{A}_{\text{prv}}$ returns a bit b', which we denote with $\mathbf{Exp}^{\text{prv-}b}_{\mathscr{A}_{\text{prv}}} = b'$.

Definition 7 (Privacy [66]). Let C be one of the adversary classes according to Definition 4. An RFID system (Definition 3) is C-private if for every adversary $\mathscr{A}_{\text{prv}}$ of C there is a probabilistic polynomial time algorithm $\mathscr{B}$ (blinder) such that

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

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

All privacy notions defined in the PV-Model [53] are summarized in Figure 1, which also shows the relations among them. It has been shown that strong privacy is impossible [66] while the technical feasibility of destructive privacy has been an open problem.

5 A PUF-based Destructive-Private RFID Protocol

In this section, we address an open problem of [66] by presenting the first destructive-private RFID protocol. Our protocol is based on the weak-private protocol of [66], which is a simple challenge-response protocol. To achieve destructive-privacy, in our protocol, the tag $\mathscr{T}$ 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 $\mathscr{T}$ 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 $F : \{0,1\}^{\kappa} \times \{0,1\}^{2\alpha} \longrightarrow \{0,1\}^{\beta}$ be a family of pseudorandom functions. Each tag $\mathscr{T}$ is equipped with a PUF $\mathsf{P} : \{0,1\}^{\gamma} \longrightarrow \{0,1\}^{\kappa}$ and is initialized by a random state $S \in_R \{0,1\}^{\gamma}$. The credentials database DB of the reader $\mathscr{R}$ contains a tuple $(\mathtt{ID}, K)$ for each legitimate tag $\mathscr{T}_{\mathtt{ID}}$ where $K \leftarrow \mathsf{P}(S)$.

Our destructive-private tag authentication protocol is illustrated in Figure 2. The reader $\mathscr{R}$ starts by sending a random challenge a to the tag $\mathscr{T}_{\mathtt{ID}}$, which first chooses a random value b and then queries its PUF with its state S to reconstruct its tag authentication secret K. Next, the tag $\mathscr{T}_{\mathtt{ID}}$ evaluates $F_K(a,b)$, sends the result c and b to the reader $\mathscr{R}$ and immediately erases K, a, b and c from its temporary memory. On

recept of c the reader $\mathscr{R}$ recomputes $F_K(a,b)$ for each tuple $(\mathtt{ID},K)$ in its credential database $\mathtt{DB}$ until it finds a match. If the reader $\mathscr{R}$ finds a matching $(\mathtt{ID},K)$, it accepts the tag $\mathscr{T}_{\mathtt{ID}}$ by returning $\mathtt{ID}$. Otherwise, the reader $\mathscr{R}$ rejects the tag $\mathscr{T}_{\mathtt{ID}}$ and returns $\perp$.

Correctness

Clearly, if both tag $\mathscr{T}_{\mathtt{ID}}$ and reader $\mathscr{R}$ 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).

6 Security Analysis

6.1 Tag Authentication

Theorem 1. *The RFID protocol illustrated in Figure 2 achieves tag authentication (Definition 6).*

Proof. Assume by contradiction that the protocol shown in Figure 2 does not achieve tag authentication. This means that there is an adversary $\mathscr{A}_{\text{sec}}$ 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 $(\tilde{\mathtt{ID}},\tilde{K}) \in \mathtt{DB}$ without having made a $\mathsf{Corrupt}$ or $\mathsf{SendTag}(\tilde{a},\cdot)$ query to the tag $\mathscr{T}_{\tilde{\mathtt{ID}}}$. In the following, we show that $\mathscr{A}_{\text{sec}}$ can be transformed into a probabilistic polynomial time algorithm $\mathscr{A}_{\text{prf}}$ 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 $\mathscr{A}_{\text{sec}}$.

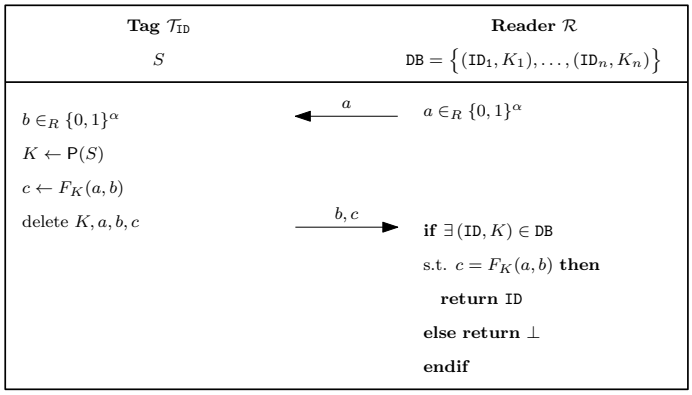

Fig. 2 Destructive-private PUF-based RFID protocol.

The construction of $\mathscr{A}_{\text{prf}}$ is as follows: Given the security parameters l, κ, α, β and a description of the PRF F from the PRF-challenger $\mathscr{C}_{\text{prf}}$, $\mathscr{A}_{\text{prf}}$ initializes the RFID system by first choosing γ polynomial in l and then setting $sk_{\mathscr{R}} \leftarrow \emptyset$, $pk_{\mathscr{R}} \leftarrow (l,\gamma,\kappa,\alpha,\beta,F)$ and $\mathtt{DB} \leftarrow \emptyset$. Then $\mathscr{A}_{\text{prf}}$ guesses the identifier $\tilde{\mathtt{ID}}$ of the tag $\mathscr{T}_{\tilde{\mathtt{ID}}}$ that will be impersonated by $\mathscr{A}_{\text{sec}}$. Note that the probability of correctly guessing $\tilde{\mathtt{ID}}$ is polynomial since $\mathscr{A}_{\text{sec}}$ can create at most a polynomial number of tags. Next, $\mathscr{A}_{\text{prf}}$ initializes $\mathscr{A}_{\text{sec}}$ with $(l,\gamma,\kappa,\alpha,\beta,F)$ and simulates all the oracles defined in Section 4.2 to $\mathscr{A}_{\text{sec}}$:

- $\mathsf{CreateTag}(\mathtt{ID})$ If there already is a tuple $(\mathtt{ID},\cdot,\cdot) \in \mathtt{DB}$ or if $\mathtt{ID} = \tilde{\mathtt{ID}}$, then $\mathscr{A}_{\text{prf}}$ aborts. Otherwise, $\mathscr{A}_{\text{prf}}$ chooses $S \in_R \{0,1\}^{\gamma}$ and $K \in_R \{0,1\}^{\kappa}$ and updates $\mathtt{DB} \leftarrow \mathtt{DB} \cup \{(\mathtt{ID},K,S)\}$.
- Draw, Free, Launch The simulation of the Draw, Free and Launch oracle is straightforward. Note that $\mathscr{A}_{\text{prf}}$ knows the secret look-up table Γ of the Draw oracle.
- $\mathsf{SendTag}(a,vtag)$ If $\Gamma[vtag] = \tilde{\mathtt{ID}}$, then $\mathscr{A}_{\text{prf}}$ responds with $b \in_R \{0,1\}^{\alpha}$ and $c \leftarrow \mathscr{O}^{F_K}(a,b)$. Else, $\mathscr{A}_{\text{prf}}$ gets $(\Gamma[vtag],K,S)$ from $\mathtt{DB}$ and responds with $b \in_R \{0,1\}^{\alpha}$ and $c \leftarrow F_K(a,b)$.
- $\mathsf{SendReader}(\emptyset,\pi)$ If π has been previously generated by a Launch oracle query and the corresponding protocol transcript is $\mathtt{tr}_{\pi} = \emptyset$, then $\mathscr{A}_{\text{prf}}$ returns $a \in_R \{0,1\}^{\alpha}$ and updates $\mathtt{tr}_{\pi} \leftarrow a$.
- $\mathsf{SendReader}((b,c),\pi)$ If π has been previously generated by a Launch oracle query and the corresponding protocol transcript is $\mathtt{tr}_{\pi} = a$, then $\mathscr{A}_{\text{prf}}$ updates the protocol transcript $\mathtt{tr}_{\pi} \leftarrow (a,b,c)$ and aborts otherwise.
- $\mathsf{Result}(\pi)$ If π has been previously generated by a Launch oracle query and the corresponding protocol transcript $\mathtt{tr}_{\pi} = (a,b,c)$ has been obtained through $a \leftarrow \mathsf{SendReader}(\emptyset,\pi)$, then computes $c' \leftarrow F_K(a,b)$ for each tuple $(\mathtt{ID},K)$ in $\mathtt{DB}$. If a $c' = c$ for some $(\mathtt{ID},K)$ then returns 1, otherwise returns 0.
- $\mathsf{Corrupt}(vtag)$ If there is a tuple $(\Gamma[vtag],K,S)$ in $\mathtt{DB}$, $\mathscr{A}_{\text{prf}}$ returns S. Note that according to Definition 6, $\mathscr{A}_{\text{sec}}$ is not allowed to corrupt the tag $\mathscr{T}_{\tilde{\mathtt{ID}}}$ and hence, $\mathscr{A}_{\text{prf}}$ needs not to simulate the Corrupt oracle for the tag $\mathscr{T}_{\tilde{\mathtt{ID}}}$.

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

Note that in case $b = 1$, $\mathscr{A}_{\text{prf}}$ perfectly simulates all oracles defined in Section 4.2 to $\mathscr{A}_{\text{sec}}$. Hence, in case $b = 1$, by assumption $\mathscr{A}_{\text{sec}}$ 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 $\mathscr{A}_{\text{prf}}$ 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. □

6.2 Destructive Privacy

Theorem 2. *The RFID protocol illustrated in Figure 2 achieves destructive privacy (Definition 7).*

Proof. According to Definition 7, destructive privacy means that there is a blinder $\mathscr{B}$ that simulates the Launch, SendTag, SendReader and Result oracle such that no destructive adversary $\mathscr{A}_{\text{prv}}$ (Definition 4) can distinguish between the blinder $\mathscr{B}$ and the real oracles. Hence, to prove Theorem 2, we first give the construction of the blinder $\mathscr{B}$ and then show that every destructive adversary $\mathscr{A}_{\text{prv}}$ has at most negligible probability to distinguish the blinder $\mathscr{B}$ from the real oracles.

The blinder $\mathscr{B}$ is initialized with the security parameters l, γ, κ, α, β and the public key $pk_{\mathscr{R}}$ of the reader $\mathscr{R}$ 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(π) Return $a \in_R \{0,1\}^{\alpha}$.
- SendReader$\big((b,c),\pi\big)$ Since oracle queries of this form do not generate any output nor change the state of the tag and the reader, the blinder $\mathscr{B}$ needs not to simulate their responses.
- Result(π) If π has been previously generated by a Launch oracle query and the corresponding protocol transcript $\mathtt{tr}_{\pi} = (a,b,c)$ has been generated by $a \leftarrow$ SendReader$\big(\emptyset,\pi\big)$ and $(b,c) \leftarrow$ SendTag$\big(a, vtag\big)$, return 1 and 0 otherwise.

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

Let game $\mathtt{game}^{(0)}$ be the game where the adversary $\mathscr{A}_{\text{prv}}$ interacts with the real oracles as defined in Section 4.2. Now consider the following hybrid game $\mathtt{game}^{(1)}$ that is exactly as $\mathtt{game}^{(0)}$ 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 $\mathscr{A}_{\text{prv}}$ can distinguish between $\mathtt{game}^{(0)}$ and $\mathtt{game}^{(1)}$, then we can use $\mathscr{A}_{\text{prv}}$ to construct a polynomial time algorithm $\mathscr{A}_{\text{puf}}$ that contradicts the security property of the PUF P (Definition 2).

According to the protocol specification given in Section 5, the states and PUFs of different tags are chosen independently. Moreover, $\mathscr{A}_{\text{puf}}$ can trivially simulate different tags by following the protocol specifications. Hence, we assume w.l.o.g. that $\mathscr{A}_{\text{prv}}$ creates just one single tag $\mathscr{T}_{\mathtt{ID}}$ during his attack. To create this tag $\mathscr{T}_{\mathtt{ID}}$, $\mathscr{A}_{\text{puf}}$ chooses $S \in_R \{0,1\}^{\gamma}$ and sets $K \leftarrow \mathscr{O}^{\mathsf{P}}(S)$. Note that $\mathscr{O}^{\mathsf{P}}(S)$ either returns $K \leftarrow \mathsf{P}(S)$ as in $\mathtt{game}^{(0)}$ or $K \in_R \{0,1\}^{\kappa}$ as in $\mathtt{game}^{(1)}$. Now, $\mathscr{A}_{\text{puf}}$ can interact with all the oracles defined in Section 4.2 that are simulated by $\mathscr{A}_{\text{puf}}$ based on the input of $\mathscr{C}_{\text{puf}}$. 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 $\mathscr{T}_{\mathtt{ID}}$ 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 $\mathscr{T}_{\mathtt{ID}}$, they need

not be simulated by $\mathscr{A}_{\text{puf}}$. On a $\mathsf{SendTag}(a, vtag)$ oracle query, $\mathscr{A}_{\text{puf}}$ responds with $b \in_R \{0,1\}^{\alpha}$ and $c \leftarrow F_K(a,b)$.

Note that $\mathscr{A}_{\text{prv}}$ is a destructive adversary and hence, by making a $\mathsf{Corrupt}(vtag)$ query, $\mathscr{A}_{\text{prv}}$ 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, $\mathscr{A}_{\text{prv}}$ returns a bit b'. In case $b' = 1$ (which indicates that $\mathscr{A}_{\text{prv}}$ detected $\mathscr{B}$), with non-negligible probability $\mathscr{O}^{\mathsf{P}}$ must have returned a random $K \in_R \{0,1\}^{\kappa}$. Hence, $\mathscr{A}_{\text{puf}}$ 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, it follows that

$$\left| \Pr\left[\mathtt{game}^{(0)} = 1\right] - \Pr\left[\mathtt{game}^{(1)} = 1\right] \right| \tag{1}$$

is negligible.

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

Let $q \in \mathbb{N}$ be the number of $\mathsf{SendTag}$ queries made by $\mathscr{A}_{\text{prv}}$, which is polynomial in l. Moreover, let $i \in \{0,\ldots,q\}$. Now consider the following hybrid game $\mathtt{game}_i$ with $\mathscr{A}_{\text{prv}}$: The first i $\mathsf{SendTag}$ queries of $\mathscr{A}_{\text{prv}}$ are answered by the blinder $\mathscr{B}$ (as in $\mathtt{game}^{(2)}$), while the remaining $q-i$ queries are forwarded and answered by the real $\mathsf{SendTag}$ oracle (as in $\mathtt{game}^{(1)}$). Note that $\mathtt{game}_0$ corresponds to $\mathtt{game}^{(1)}$ whereas $\mathtt{game}_q$ corresponds to game $\mathtt{game}^{(2)}$. Hence, and due to the contradicting assumption made at the beginning of the proof, it holds that

$$\mathbf{Adv}^{\text{prv}}_{\mathscr{A}_{\text{prv}}} = \left| \Pr[\mathtt{game}_0 = 1] - \Pr[\mathtt{game}_q = 1] \right|$$

is non-negligible. In return, this means that there must be some index $i \in \{1,\ldots,q\}$ such that

$$\left| \Pr[\mathtt{game}_{i-1} = 1] - \Pr[\mathtt{game}_i = 1] \right| \tag{2}$$

is non-negligible. Note that Equation 2 implies that w.l.o.g. $\mathscr{A}_{\text{prv}}$ detects $\mathscr{B}$ in game $\mathtt{game}_i$ with non-negligible probability while he has at most negligible probability to detect $\mathscr{B}$ in game $\mathtt{game}_{i-1}$.

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

$$\left|\Pr\left[\mathtt{game}^{(1)}=1\right]-\Pr\left[\mathtt{game}^{(2)}=1\right]\right| \tag{3}$$

is negligible.

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

In the following, let $p \in \mathbb{N}$ be the number of Result queries made by $\mathscr{A}_{\text{prv}}$, which is polynomial in l. Moreover, let $i \in \{0,\ldots,p\}$. Now consider the following hybrid game $\mathtt{game}_i^*$: The first i Result queries of $\mathscr{A}_{\text{prv}}$ are answered by the blinder $\mathscr{B}$ (as in $\mathtt{game}^{(3)}$), while the remaining $p-i$ queries are forwarded and answered by the real Result oracle (as in $\mathtt{game}^{(2)}$). Note that $\mathtt{game}_0^*$ corresponds to $\mathtt{game}^{(2)}$ whereas $\mathtt{game}_p^*$ is equivalent to $\mathtt{game}^{(3)}$. Hence, and due to the contradicting assumption made at the beginning of the proof, it holds that

$$\mathbf{Adv}^{\text{prv}}_{\mathscr{A}_{\text{prv}}} = \left|\Pr[\mathtt{game}_0^*=1]-\Pr[\mathtt{game}_p^*=1]\right|$$

is non-negligible. In return, this means that there must be some index $i \in \{1,\ldots,p\}$ such that

$$\left|\Pr[\mathtt{game}_{i-1}^*=1]-\Pr[\mathtt{game}_i^*=1]\right| \tag{4}$$

is non-negligible. Note that Equation 4 implies that w.l.o.g. $\mathscr{A}_{\text{prv}}$ detects $\mathscr{B}$ in game $\mathtt{game}_i^*$ with non-negligible probability while he has at most negligible probability to detect $\mathscr{B}$ in game $\mathtt{game}_{i-1}^*$. This means that in $\mathtt{game}_i^*$ $\mathscr{A}_{\text{prv}}$ runs a protocol instance π where the Result oracle simulated by $\mathscr{B}$ returns a different output then the real Result oracle. According to the description of $\mathscr{B}$ given at the beginning of this proof and the definition of the Result oracle in Section 4.2, this can only happen if $\mathscr{A}_{\text{prv}}$ generates a protocol transcript $\mathtt{tr}_\pi = (a,b,c)$ such that $c = F_K(a,b)$ where $(\mathtt{ID},K) \in \mathtt{DB}$ and tag $\mathscr{T}_{\mathtt{ID}}$ has not been corrupted by $\mathscr{A}_{\text{prv}}$. 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

$$\left|\Pr\left[\mathtt{game}^{(2)}=1\right]-\Pr\left[\mathtt{game}^{(3)}=1\right]\right| \tag{5}$$

is negligible as well.

Note that $\mathtt{game}^{(3)}$ corresponds to the game where $\mathscr{A}_{\text{prv}}$ interacts with a full blinder $\mathscr{B}$. Hence, from Equation 1, Equation 3 and Equation 5 it follows that

$$\left|\Pr\left[\mathtt{game}^{(0)}=1\right]-\Pr\left[\mathtt{game}^{(3)}=1\right]\right|$$

is negligible. This means that $\mathscr{A}_{\text{prv}}$ cannot distinguish between the real oracles and the full blinder $\mathscr{B}$, which completes the proof of Theorem 2. □

7 Conclusion

In this paper, we have shown that physically unclonable functions 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 FP7 programme under contract 216646 ECRYPT II, 238811 UNIQUE, and 215270 FRONTS, in part by the Ateneo Italo-Tedesco under Program Vigoni and by the MIUR Project PRIN 2008 "PEPPER: Privacy E Protezione di dati PERsonali" (prot. 2008SY2PH4).

References

1. Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pages 92–101. ACM Press, 2005.
2. Atmel Corporation. Innovative IDIC solutions. `http://www.atmel.com/dyn/resources/prod_documents/doc4602.pdf`, 2007.
3. Gildas Avoine. Adversarial model for radio frequency identification. Cryptology ePrint Archive, Report 2005/049, 2005.
4. Gildas Avoine, Etienne Dysli, and Philippe Oechslin. Reducing time complexity in RFID systems. In *12th International Workshop on Selected Areas in Cryptography (SAC) 2005*, volume 3897 of *LNCS*, pages 291–306. Springer Verlag, 2005.
5. Gildas Avoine, Cedric Lauradoux, and Tania Martin. When compromised readers meet RFID. In *The 5th Workshop on RFID Security 2009, Leuven, Belgium, June 30 – July 2*, 2009.
6. Leonid Bolotnyy and Gabriel Robins. Physically unclonable function-based security and privacy in RFID systems. In *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications*, pages 211–220. IEEE Computer Society, 2007.
7. 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.
8. Ivan Damgård and Michael Østergaard. RFID security: Tradeoffs between security and efficiency. Cryptology ePrint Archive, Report 2006/234, 2006.

9. Paolo D'Arco, Alessandra Scafuro, and Ivan Visconti. Revisiting DoS Attacks and Privacy in RFID-Enabled Networks. In *Proceedings of ALGOSENSORS 2009*, LNCS. Springer-Verlag, July 2009.
10. Paolo D'Arco, Alessandra Scafuro, and Ivan Visconti. Semi-Destructive Privacy in DoS-Enabled RFID Systems. In *Proceedings of RFIDSec 2009*, July 2009.
11. 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 2008, Las Vegas, NV, USA, 16–17 April, 2008*, pages 58–64. IEEE Computer Society, 2008.
12. Tassos Dimitriou. A lightweight RFID protocol to protect against traceability and cloning attacks. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm)*, pages 59–66. IEEE Computer Society, 2005.
13. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2–6, 2004, Proceedings*, volume 3027 of *LNCS*, pages 523–540. Springer Verlag, 2004.
14. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. *Security with Noisy Data*, chapter Fuzzy Extractors, pages 79–99. Springer-Verlag, 2007.
15. EPCglobal Inc. Object Naming Service (ONS), version 1.0, October 2005.
16. EPCglobal Inc. Web site of EPCglobal Inc. `http://www.epcglobalinc.org/`, April 2008.
17. Klaus Finkenzeller. *RFID-Handbook*. Wiley & Sons, 2nd edition, April 2003. Translated from the 3rd German edition by Rachel Waddington, Swadlincote, UK.
18. Dmitry Frumkin and Adi Shamir. Un-Trusted-HB: Security Vulnerabilities of Trusted-HB. Cryptology ePrint Archive, Report 2009/044, 2009.
19. Blaise Gassend, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. Controlled physical random functions. In *Proceedings of the 18th Annual Computer Security Applications Conference, December 9–13, 2002*, pages 149–160. IEEE Computer Society, 2002.
20. Henri Gilbert, Matthew Robshaw, and Hervé Silbert. An active attack against HB+ — A provable secure leightweight authentication protocol. Cryptology ePrint Archive, Report 2007/237, 2007.
21. Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. Good Variants of HB+ Are Hard to Find. In Gene Tsudik, editor, *Financial Cryptography and Data Security, 12th International Conference, FC 2008, Cozumel, Mexico, January 28–31, 2008, Revised Selected Papers*, volume 5143 of *LNCS*, pages 156–170. Springer-Verlag, 2008.
22. Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson. Universal re-encryption for mixnets. In *The Cryptographers' Track at the RSA Conference 2004, Proceedings*, volume 2964 of *LNCS*, pages 163–178. Springer Verlag, 2004.
23. Jung Hoon Ha, Sang Jae Moon, Jianying Zhou, and Jae Cheol Ha. A new formal proof model for RFID location privacy. In Jajodia and Lopez [28], pages 267–281.
24. Dirk Henrici and Paul Müller. Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 149–153. IEEE Computer Society, 2004.
25. Daniel E. Holcomb, Wayne P. Burleson, and Kevin Fu. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In *Conference on RFID Security 2007, Malaga, Spain, July 11–13*, 2007.
26. Michael Hutter, Jörn-Marc Schmidt, and Thomas Plos. RFID and its vulnerability to faults. In *10th International Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2008, Washington, D.C., USA, August 10–13, 2008, Proceedings*, volume 5154 of *LNCS*, pages 363–379. Springer-Verlag, 2008.
27. I.C.A. Organization. Machine Readable Travel Documents, Doc 9303, Part 1 Machine Readable Passports, Fifth Edition, 2003.

28. Sushil Jajodia and Javier Lopez, editors. *Computer Security — ESORICS 2008*, volume 5283 of *LNCS*. Springer Verlag, 2008.
29. Ari Juels. Minimalist cryptography for low-cost RFID tags (extended abstract). In *4th International Conference on Security in Communication Networks (SCN) 2004, Revised Selected Papers*, volume 3352 of *LNCS*, pages 149–164. Springer Verlag, 2004.
30. Ari Juels. RFID security and privacy: A research survey. *Journal of Selected Areas in Communication*, 24(2):381–395, February 2006.
31. Ari Juels and Ravikanth Pappu. Squealing Euros: Privacy protection in RFID-enabled banknotes. In *7th International Conference on Financial Cryptography (FC) 2003, Revised Papers*, volume 2742 of *LNCS*, pages 103–121. Springer Verlag, 2003.
32. Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In Victor Shoup, editor, *Advances in Cryptology — CRYPTO 2005, 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14–18, 2005, Proceedings*, volume 3621 of *LNCS*, pages 293–308. Springer-Verlag, 2005.
33. Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. Cryptology ePrint Archive, Report 2006/137, 2006.
34. Jonathan Katz. Efficient Cryptographic Protocols Based on the Hardness of Learning Parity with Noise. In Steven D. Galbraith, editor, *Cryptography and Coding, 11th IMA International Conference, Cirencester, UK, December 18–20, 2007, Proceedings*, volume 4887 of *LNCS*, pages 1–15. Springer-Verlag, 2007.
35. Jonathan Katz and Ji Sun Shin. Parallel and Concurrent Security of the HB and HB+ Protocols. In Serge Vaudenay, editor, *Advances in Cryptology — EUROCRYPT 2006, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 – June 1, 2006, Proceedings*, volume 4004 of *LNCS*, pages 73–87. Springer-Verlag, 2006.
36. Jonathan Katz and Adam Smith. Analyzing the HB and HB+ protocols in the "large error" case. Cryptology ePrint Archive, Report 2006/326, 2006.
37. Ilan Kirschenbaum and Avishai Wool. How to build a low-cost, extended-range RFID skimmer. Cryptology ePrint Archive, Report 2006/054, 2006.
38. Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *16th Annual International Cryptology Conference, Santa Barbara, California, USA, Proceedings, August 18–22, 1996*, volume 1109 of *LNCS*, pages 104–113. Springer Verlag, 1996.
39. Oliver Kömmerling and Markus G. Kuhn. Design principles for tamper-resistant smartcard processors. In *Proceedings of the USENIX Workshop on Smartcard Technology on USENIX Workshop on Smartcard Technology, Chicago, Illinois, May 10–11, 1999*, 1999.
40. Chae Hoon Lim and Taekyoung Kwon. Strong and robust RFID authentication enabling perfect ownership transfer. In *8th International Conference on Information and Communications Security (ICICS)*, volume 4307 of *LNCS*, pages 1–20. Springer Verlag, 2006.
41. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks Revealing the Secrets of Smart Cards*. Springer-Verlag, 2007.
42. David Molnar and David Wagner. Privacy and security in library RFID: Issues, practices, and architectures. In *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 210–219. ACM Press, 2004.
43. Michael Neve, Eric Peeters, David Samyde, and Jean-Jacques Quisquater. Memories: A survey of their secure uses in smart cards. In *Proceedings of the Second IEEE International Security in Storage Workshop, October 31, 2003*, pages 62–72. IEEE Computer Society, 2003.
44. Ching Yu Ng, Willy Susilo, Yi Mu, and Safavi-Naini. New privacy results on synchronized RFID authentication protocols against tag tracing. In *Proc. of ESORICS*, volume 5789 of *LNCS*, pages 321–336. Springer, 2009.
45. Ching Yu Ng, Willy Susilo, Yi Mu, and Rei Safavi-Naini. RFID privacy models revisited. In Jajodia and Lopez [28], pages 251–256.
46. NXP Semiconductors. MIFARE Application Directory (MAD) — List of Registered Applications. `http://www.nxp.com/acrobat/other/identification/mad_overview_042008.pdf`, April 2008.

47. NXP Semiconductors. MIFARE Smartcard ICs. http://www.mifare.net/products/smartcardics/, September 2008.
48. Octopus Holdings. Web site of Octopus Holdings. http://www.octopus.com.hk/en/, June 2008.
49. Silvio Micali Oded Goldreich, Shafi Goldwasser. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.
50. Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic approach to "privacy-friendly" tags, November 2003.
51. Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Efficient hash-chain based RFID privacy protection scheme. International Conference on Ubiquitous Computing (UbiComp), Workshop Privacy: Current Status and Future Directions, September 2004.
52. Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the Security of HB# against a Man-in-the-Middle Attack. In Josef Pieprzyk, editor, *Advances in Cryptology — ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7–11, 2008, Proceedings*, volume 5350 of *LNCS*, pages 108–124. Springer-Verlag, 2008.
53. Radu-Ioan Paise and Serge Vaudenay. Mutual authentication in RFID: Security and privacy. In *ASIACCS'08: Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*, pages 292–299. ACM Press, 2008.
54. 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.
55. 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.
56. 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*, LNCS. Springer-Verlag, 2009.
57. 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.
58. Junichiro Saito, Jae-Cheol Ryou, and Kouichi Sakurai. Enhancing privacy of universal re-encryption scheme for RFID tags. In *International Conference on Embedded and Ubiquitous Computing (EUC) 2004, Proceedings*, volume 3207 of *LNCS*, pages 879–890. Springer Verlag, 2004.
59. Sergei P. Skorobogatov and Ross J. Anderson. Optical fault induction attacks. In *4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002), Redwood Shores, CA, USA, August 13–15, 2002, Revised Papers*, volume 2523 of *LNCS*, pages 31–48. Springer Verlag, 2002.
60. Boyeon Song and Chris J. Mitchell. RFID authentication protocol for low-cost tags. In *Proceedings of the First ACM Conference on Wireless Network Security*, pages 140–147. ACM Press, 2008.
61. Sony Global. Web site of Sony FeliCa. http://www.sony.net/Products/felica/, June 2008.
62. Spirtech. CALYPSO functional specification: Card application, version 1.3. http://calypso.spirtech.net/, October 2005.
63. Gene Tsudik. YA-TRAP: Yet Another Trivial RFID Authentication Protocol. In *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, volume 2802 of *LNCS*, pages 640–643. IEEE Computer Society, 2006.
64. Pim Tuyls and Lejla Batina. RFID-tags for anti-counterfeiting. In *The Cryptographers' Track at the RSA Conference 2006, Proceedings*, volume 3860 of *Lecture Notes on Computer Science (LNCS)*, pages 115–131. Springer Verlag, 2006.
65. Pim Tuyls, Boris Škoriç, and Tom Kevenaar, editors. *Security with Noisy Data — On Private Biometrics, Secure Key Storage, and Anti-Counterfeiting*. Springer-Verlag, 2007.

66. Serge Vaudenay. On privacy models for RFID. In *13th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings*, volume 4833 of *LNCS*, pages 68–87. Springer Verlag, 2007.
67. 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 *First International Conference on Security in Pervasive Computing, Revised Papers*, volume 2802 of *LNCS*, pages 50–59. Springer Verlag, 2003.
68. Éric Levieil and Pierre-Alain Fouque. An Improved LPN Algorithm. In *Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, September 6–8, 2006, Proceedings*, volume 4116 of *LNCS*, pages 348–359. Springer-Verlag, 2006.