Anonymizer-Enabled Security and Privacy for RFID

Ahmad-Reza Sadeghi[1], Ivan Visconti[2], and Christian Wachsmann[1]

[1] Horst Görtz Institute for IT-Security, Ruhr-University Bochum, Germany
{ahmad.sadeghi,christian.wachsmann}@trust.rub.de
[2] Dipartimento di Informatica ed Applicazioni, University of Salerno, Italy
visconti@dia.unisa.it

Abstract. RFID-based systems are becoming a widely deployed pervasive technology that is more and more used in applications where privacy-sensitive information is entrusted to RFID tags. Thus, a careful analysis in appropriate security and privacy models is needed before deployment to practice.

Recently, Vaudenay presented a comprehensive security and privacy model for RFID that captures most previously proposed privacy models. The strongest achievable notion of privacy in this model (*narrow-strong privacy*) requires public-key cryptography, which in general exceeds the computational capabilities of current cost-efficient RFIDs. Other privacy notions achievable without public-key cryptography heavily restrict the power of the adversary and thus are not suitable to realistically model the real world.

In this paper, we extend and improve the current state-of-the art for privacy-protecting RFID by introducing a security and privacy model for *anonymizer*-enabled RFID systems. Our model builds on top of Vaudenay's model and supports anonymizers, which are separate devices specifically designated to ensure the privacy of tags. We present a privacy-preserving RFID protocol that uses anonymizers and achieves narrow-strong privacy without requiring tags to perform expensive public-key operations (i.e., modular exponentiations), thus providing a satisfying notion of privacy for cost-efficient tags.

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 and studied for its applications, including animal identification [1], library management [2], access control [1,3,4,5], electronic tickets [4,3,5,6] and passports [7] and even human implantation [8].

As pointed out in previous publications (see, e.g., [9,8]), 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

J.A. Garay, A. Miyaji, and A. Otsuka (Eds.): CANS 2009, LNCS 5888, pp. 134–153, 2009.

users. Thus, an RFID system should provide *anonymity* (i.e., confidentiality of the tag identity) as well as *untraceability* (i.e., unlinkability of the communication of a tag) even in case the state of (i.e., the information stored on) the tag has been disclosed.[1] RFID applications in practice must also achieve various security and functional goals. The security goals include *authentication*, which prevents an adversary from impersonating and forging tags and *availability*, which means the resilience to remote tampering that allows denial-of-service attacks. The functional goals include *efficiency* (e.g., fast verification of cost-efficient tags) and *scalability* (i.e., support of a huge number of tags).

Most currently used RFID systems do not offer privacy at all (e.g., [11,12,4,5]). This is mainly because current cost-efficient tags do not provide the necessary computational resources to run privacy-preserving protocols [1,4], which heavily rely on public-key cryptography. Moreover, as pointed out in Section 2, privacy-preserving solutions without public-key cryptography do not fulfill important security or functional requirements and thus, are inapplicable to real-world applications.

As elaborated in related work (see Section 2), a promising approach towards solving these problems and our focus in this paper, are *anonymizers*. These are special devices that take off the computational workload (i.e., the public-key operations) from the tags and enable privacy-preserving protocols with cost-efficient tags. Note that an anonymizer-based RFID system is not equivalent to a straight forward extension of a resource constrained RFID system to one with higher capabilities (such as public-key cryptography). The anonymizer-enabled approach in general requires an additional protocol between tags and anonymizers that opens attack surfaces and thus, must be carefully considered. Indeed, an anonymizer shall not be able to impersonate or to copy the tags it anonymizes since this would violate authentication. Moreover, to ensure availability, the protocol between a tag and the anonymizer must be secure against attacks where an adversary aims to manipulate the tag.

Anonymizers can be incorporated into the standard RFID system model in different ways. One approach is to provide public anonymizers that can be controlled by the operator of the RFID system or by one of several independent anonymizing service providers the user may choose from. Alternatively, each user may have his/her own personal anonymizer that could be implemented as a software running on the user's mobile phone or PDA[2], allowing a very cost-efficient implementation of anonymizers. The main advantage of anonymizer-enabled protocols is that they allow operators of RFID systems to enable privacy for the concerned users (who may buy his/her own personal anonymizer) with no or only minor extra costs.

[1] To distinguish tracing in past or future protocol-runs, the notions of *forward untraceability* (i.e., unlinkability of the communication of the tag that has been recorded *before* disclosure) and *backward untraceability* (i.e., unlinkability of the communication of the tag that takes place *after* disclosure) are defined in [10].

[2] An increasing number of mobile phones and PDAs support the Near Field Communication (NFC) standard, which enables them to communicate to RFID devices.

However, as pointed out in Section 2, current anonymizer-enabled solutions are vulnerable to impersonation attacks. Hence, the design of a secure and privacy-preserving RFID system requires an appropriate security and privacy model to enable a careful analysis of the underlying schemes. On the other hand, existing security and privacy models for RFID (e.g., [13,14,15,16]) suffer from various shortcomings. As discussed in Section 2, these models do not consider important aspects like adversaries with access to *auxiliary information* (i.e., whether the identification of a tag was successful or not) or the privacy of *corrupted* tags (i.e., whose state has been disclosed). Both are essential to ensure anonymity and untraceability in practice. Another drawback is that most of these models are incomparable, which leads to the problem that a protocol can be proven secure in one privacy model while being insecure in another model.[3] Therefore, it is crucial to develop a widely accepted security and privacy model for RFID.

Recently, a comprehensive security and privacy model that generalizes and improves many previous works has been proposed in [18] and refined in [19,20]. 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, narrow-strong privacy requires the use of public-key cryptography [18], which in general clearly exceeds the capabilities of current cost-efficient RFIDs [1,4]. Moreover, it has been shown that forward privacy can be achieved but at the cost of using public-key cryptography [18] (which in general is too expensive).

We observe that the model of [18] does not include anonymizers, which play a critical role for going beyond the barrier of simultaneously achieving a strong privacy notion with protocols that are suitable for cost-efficient tags. Therefore, we investigate the use of anonymizers in the model of [18] and show an anonymizer-enabled scheme that provides important security and privacy properties while fulfilling the functional requirements of real-world applications.

Contribution. We introduce a formal framework for privacy-preserving RFID systems, which extends the security and privacy model of [18] to support anonymizers and at the same time is backwards-compatible to it. Given the granularity of the different security and privacy notions of [18], our anonymizer-based model is the first universal security and privacy model for anonymizer-enabled RFID systems. Moreover, we propose a privacy-preserving RFID protocol that can be proven secure and private in the anonymizer-enabled model (with random oracles). The protocol that we propose enjoys several appealing features that were not simultaneously achieved by any previous proposal. Indeed, our protocol is very efficient for all involved entities, in particular for tags that only have

[3] For instance, the OSK protocol [17] can be proven secure in the model of [13] although a tracing attack can be shown in the model of [14].

to perform minimal computations. Further, the protocol enjoys the strongest achievable[4] privacy notion defined in [18], which is narrow-strong privacy. Our protocol also provides forward privacy, which restricts the adversary's capability to corrupt tags but instead allows him to access auxiliary information. We finally stress that our protocol is provably secure against impersonation attacks and forgeries even if the adversary can corrupt the anonymizers. Therefore, we require the existence of (honest) anonymizers in the system only to guarantee privacy (anonymity and untraceability) of the tags. This assumption gracefully matches the realistic scenario where many anonymizers are spread in the system and an adversary can be successful in corrupting many of them with the purpose of violating the security of the system. At the same time, privacy is guaranteed as long as tags are frequently anonymized by an uncorrupted anonymizer.

2 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.[5] Essentially there are two 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 [9]. Solutions to optimize this approach (see, e.g., [2,30]) 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. In the second approach, a tag updates its identity after each interaction such that the 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., [31,32,33,10,34]). This approach requires each tag to be always synchronized with all readers in the system. However, in general, it is easy to mount denial-of-service attacks that desynchronize the tag and the readers (see e.g., [31,33]). For a broad overview about privacy issues in RFID systems, see also [35].

Anonymizer-Enabled RFID Protocols. A promising approach to enhance privacy of RFID without lifting the computational requirements on tags are anonymizer-enabled protocols, where external devices (*anonymizers*) are in charge of providing anonymity of tags. Anonymizer-enabled RFID protocols are very

[4] Note that the impossibility of achieving strong privacy [18] trivially holds in our anonymizer-enabled model since any protocol in the anonymizer-enabled model also works in the model of [18] by simply requiring that the anonymization protocol (i.e., the protocol run between tags and anonymizers) is played locally inside tags.

[5] A prominent family of lightweight authentication protocols proposed in the context of RFID are the HB protocols (see e.g., [21,22,23,24]). However, these protocols are subject to man-in-the-middle attacks [25,26,27,28], require the reader to perform an exhaustive search for the (shared) authentication secret of the authenticating tag and have a high communication complexity (many rounds of interaction) [29]. Moreover, tag corruption is usually not considered in the security evaluation of the HB protocols.

suitable for many practical scenarios with privacy needs that use cost-efficient tags. The main concept of existing anonymizer-enabled protocols [36,37,38,39] 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 sent to the reader each time the tag authenticates. Since this ciphertext is static data and can be used to track and to identify the tag, it must be frequently changed to provide anonymity and unlinkability. However, current RFIDs [1,4] are not capable of updating their ciphertext on their own and thus, privacy in these protocols relies on anonymizers that frequently refresh the ciphertexts stored on the tags. The first proposal to use anonymizers [36] considers a plan by the European Central Bank to embed RFID tags into Euro banknotes to aggravate forgeries [40]. It proposes to store a ciphertext of the serial number of a banknote on the RFID tag that is attached to the banknote. Each time the banknote is spent, anonymizers in shops or banks re-encrypt the ciphertext stored on the tag. The drawback of this scheme is that the serial number of a banknote must be optically scanned before its ciphertext can be re-encrypted. In [37], the authors introduce a primitive called *universal re-encryption*, which is an extension of the El Gamal encryption scheme where re-encryption is possible without knowledge of the corresponding (private and public) keys. In this approach, an adversary can "mark" tags such that he can recognize them even after they have been re-encrypted. This issue has been addressed in [38] that shows tracing attacks and proposes solutions. In [39], the authors improve the ideas of [37] and [38] by introducing the notion of *insubvertible encryption*, which adds a signature on the blinded public key of the reader that is linked to the ciphertexts stored on the tags. Re-randomization involves this signature in a way that prevents the adversary from marking tags.

All known anonymizer-enabled schemes are subject to impersonation attacks since 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.

Privacy Models for RFID. One of the first privacy models for RFID [17] defines anonymity and backward untraceability based on a security game where an adversary must distinguish a random value from the output of a tag. It does not consider forward untraceability. A privacy model specific for RFIDs that cannot perform any cryptographic operations [41] 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 [13,42] 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 [14] by the notion of auxiliary information. In [15], a *completeness* and *soundness* requirement is added to the definition of [14], which means that a reader must accept *all* but *only* valid tags. The definition of [14] has been further improved in [43] to cover backwards untraceability. Another privacy model [16] 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 thus does not capture backwards untraceability.

Recently, [18] 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 [18] captures anonymity, backward and forward untraceability and adversaries with access to auxiliary information. Moreover, it provides a security definition equivalent to [15] that covers authentication. The model of [18] has been extended in [19] to consider reader authentication whereas [20] aims at reducing the mentioned eight privacy classes to three privacy classes. Recently in [44,45] other privacy notions have been considered along with denial of service attacks. Since [18] classifies the most significant RFID privacy notions, we focus on this security model and extend it to support anonymizers.

3 Our Anonymizer-Enabled RFID System

3.1 Trust Relations and Assumptions

Before presenting our anonymizer-enabled RFID system, we first give an informal description of the underlying trust relations that are formalized in Section 4.1.

Roles and Trust Relations. An anonymizer-enabled RFID system consists of readers R, anonymizers A and tags T. The readers R set up tags that can later be identified by all the readers R in he system. A tag T that has been set up by an honest R is called *legitimate*. The task of the anonymizers A is to enforce the privacy goals of legitimate tags.

As most RFID privacy models, we assume the readers R to be trusted. This means that the readers R will behave as intended, which means that they do nothing that violates the security and privacy goals of legitimate tags. Tags are considered to be untrusted since an adversary can obtain full control of the tags and the data stored on them. Similar to tags, we consider anonymizers to be untrusted and an adversary can get full control over many anonymizers and their secrets.

Assumptions. Following the majority of existing RFID models, we make the following assumptions.

Reader. We assume that all readers R are connected to the same backend system (e.g., a database d). Thus, all honest readers R have access to the same information and thus can be subsumed as *one single* reader entity R. Moreover, the reader R can perform public-key cryptography and can handle multiple instances of the identification protocol with different tags in parallel.

Tags. The tags considered in this paper are passive devices, which means that they do not have own power supply but are powered by the electromagnetic field of the reader R. Thus, tags cannot initiate communication, have a narrow communication range (e.g., a few centimeters to meters) and are constrained in

their computational and storage capabilities, which limits them to basic cryptographic functions like hashing, random number generation and symmetric-key encryption [1,4].

Anonymizer. Anonymizers can perform public-key cryptography and can handle multiple parallel instances of the anonymization protocol with different tags. Since a tag T does not posses the required computational resources to update its state, it can always be tracked between two anonymizations. Therefore, to provide anonymity and unlinkability, it must be guaranteed that each tag T is frequently anonymized by an honest anonymizer (e.g., every few minutes). In practice, this is achieved by a dense network of public anonymizers or a personal anonymizer. At this point we stress that in order to eavesdrop on every interaction of a tag with a reader or an anonymizer, an adversary must always be within reading range of the tag. Due to the limited communication range of RFID this implies that the adversary is following the user of the tag, which obviously violates the tag user's privacy even if he would not carry an RFID tag. Thus, a privacy-preserving RFID system can at most offer privacy guarantees against adversaries that do not have permanent access to the tags. Moreover, an adversary in practice can at most corrupt a limited number of anonymizers, which ensures that there is at least one honest anonymizer in the system.

3.2 Notation and Preliminaries

General Notation. For a finite set S, $|S|$ denotes the size of set S whereas for an integer n the term $|n|$ means the bit-length of n. The term $s \in_R S$ means the assignment of a uniformly chosen element from S to variable s. 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. $\mathsf{A}_K(x)$ means that the output of A depends on x and some additional parameter K (e.g., a secret key). Probability $\epsilon(l)$ is called *negligible* if for all polynomials $f(\cdot)$ it holds that $\epsilon(l) \leq 1/f(l)$ for all sufficiently large l. Moreover, probability $1 - \epsilon(l)$ is called *overwhelming* if $\epsilon(l)$ is negligible.

Encryption Schemes. An encryption scheme ES is a tuple of algorithms $(\mathsf{Genkey}, \mathsf{Enc}, \mathsf{Dec})$ where Genkey is the key generation, Enc is the encryption and Dec is the decryption algorithm. ES is called *homomorphic* if there are two operations $(\circ, \bullet)$ such that for every pair of ciphertexts $c_1 = \mathsf{Enc}(m_1)$ and $c_2 = \mathsf{Enc}(m_2)$ it holds that $c_1 \bullet c_2 = \mathsf{Enc}(m_1 \circ m_2)$ (see, e.g., [46,37,47]). We indicate homomorphic encryption schemes by $\mathsf{ES}^{\mathfrak{h}} = (\mathsf{Genkey}^{\mathfrak{h}}, \mathsf{Enc}^{\mathfrak{h}}, \mathsf{Dec}^{\mathfrak{h}})$. A public-key encryption scheme is said to be *CPA-secure* [48,49] if every probabilistic polynomial time (p.p.t.) adversary $\mathcal{A}$ has at most negligible advantage of winning the following security experiment. An algorithm $\mathcal{S}^{\mathrm{cpa}}$ (called *CPA-challenger*), generates an encryption key pair $(sk, pk) \leftarrow \mathsf{Genkey}(1^l)$ and gives the public encryption key pk to $\mathcal{A}$. Now, $\mathcal{A}$ must respond with two messages m_0 and m_1. $\mathcal{S}^{\mathrm{cpa}}$ then randomly chooses a bit $b \in_R \{0, 1\}$, encrypts $c_b \leftarrow \mathsf{Enc}_{pk}(m_b)$ and returns the resulting ciphertext c_b to $\mathcal{A}$, who now must return a bit b' that indicates whether c_b encrypts m_0 or m_1. $\mathcal{A}$ wins if $b' = b$.

Random Oracles. A random oracle RO [50] is an oracle that responds with a random output for each given input. More precisely, RO starts with an empty look-up table τ. When queried with an input m, RO first checks if it already knows a value $\tau[m]$. If this is not the case, RO chooses a random value r and updates τ such that $\tau[m] = r$. Finally, RO returns $\tau[m]$.

3.3 Protocol Description and Specification

Our Goals. Our scheme combines and extends some of the schemes proposed in [18] and employs anonymizers, which brings several improvements that are important for practical applications. Our protocol achieves both narrow-strong and forward privacy, allows tags to be verified in constant time and provides basic protection against denial-of-service attacks. Therefore, our protocol achieves the most important security, privacy and functional requirements of practical RFID systems for both adversaries with and without access to auxiliary information.

We stress that our scheme only considers anonymity and untraceability of the communication between tags and the reader that takes place when a tag is used to access some service. Therefore, our protocol does not consider privacy of the communication between tags and anonymizers. Notice that all tags access anonymizers and thus from a rerandomization there is no special information given to the adversary about the use of a given tag obtaining access to a given service (i.e., when the tag communicates with a reader). Moreover, the use of services can be selective, since only some tags can have access to some services and thus privacy is critical in this phase. Finally note that the crucial issue is that an adversary must not be able to obtain any information about which tag accessed any service and about whether the same tag has obtained access to some services.

Our protocol provides basic availability, which means that an adversary cannot manipulate (i.e., invalidate) legitimate tags without physically attacking an anonymizer (and thus criminalizing himself). However, this is sufficient for most practical scenarios since a stolen or damaged public anonymizer can be detected and thus such attacks are unlikely to happen just to violate privacy. Further, public anonymizers can be physically secured (e.g., by a robust housing as it is used for surveillance cameras). Moreover, in the scenario of personal anonymizers, the damage that can be done by a corrupted anonymizer is limited only to the tags of one single user (since only the key of this single user's anonymizer is revealed). Obviously, a potential success in a security violation (i.e., in impersonating a legitimate tag) could motivate an adversary since he would obtain unauthorized access to services, which in turn means that he would get some economic advantages. However, our protocols turn out to be secure against impersonation attacks even against adversaries that corrupt anonymizers.

We do not consider unclonability of tags since this seems to be infeasible to achieve without hardware assumptions for the tags (which would significantly increase the costs of the tags). Further, we do not consider tracing or identification attacks based on the physical characteristics of tags, which in practice seems to be a problem that cannot be prevented by protocols on the logical layer [51].

One of the main features of our scheme is that we give a generic structure that allows one to instantiate our scheme using various cryptographic primitives (i.e., any CPA-secure homomorphic encryption scheme) based on different number-theoretic assumptions with different performance properties. Our protocol does not require tags to perform public-key cryptography (beyond the homomorphic operation that usually does not resort to a modular exponentiation) and thus, is not limited to the use of special lightweight public-key encryption schemes. This opens the possibility to employ optimized schemes, e.g., with short keys (in particular when using a prime as modulus) and ciphertexts to reduce the memory requirements to tags and to decrease the size of the protocol messages.

Protocol Overview. Our RFID scheme consists of two protocols: The tag identification and the tag anonymization protocol. The former protocol is executed by the reader R and a tag T and allows R to check if T is legitimate. The latter protocol ensures anonymity and untraceability of T in the identification protocol by updating the authentication secrets of T.

System Setup. The reader R and the anonymizers A are initialized as follows.

Reader Setup. Given a security parameter $l_{\mathsf{R}} = (l_{\mathfrak{h}}, l_{\mathfrak{s}})$, the reader R generates a key pair $(sk_{\mathsf{R}}, pk_{\mathsf{R}}) \leftarrow \mathsf{Genkey}^{\mathfrak{h}}(1^{l_{\mathfrak{h}}})$ for a CPA-secure homomorphic public-key encryption scheme. Moreover, R initializes a secret database $d \leftarrow \{\,\}$ that later stores the identities and authentication secrets of all legitimate tags. The secret key of R is sk_{R} whereas the corresponding public key is $(l_{\mathfrak{h}}, l_{\mathfrak{s}}, pk_{\mathsf{R}})$. For brevity, we write pk_{R} to mean the complete tuple.

Anonymizer Setup. Given a security parameter $l_{\mathsf{A}} = (l_{\mathfrak{a}}, l_{\mathfrak{s}})$, the anonymizer A generates a key pair $(sk_{\mathsf{A}}, pk_{\mathsf{A}}) \leftarrow \mathsf{Genkey}(1^{l_{\mathfrak{a}}})$ for the CPA-secure public-key encryption scheme. The secret key of A is sk_{A} whereas the corresponding public key is the tuple $(l_{\mathfrak{a}}, l_{\mathfrak{s}}, pk_{\mathsf{A}})$. We write pk_{A} to mean the complete tuple.[6]

Tag Creation. A tag T with identifier $\mathtt{ID}$ is initialized by the reader R as follows: first, R generates a random long-term secret K and an ephemeral secret T, that are used later in the authentication protocol to authenticate T to R. Moreover, R generates a symmetric encryption key $A \leftarrow \mathsf{Genkey}(1^{l_{\mathfrak{s}}})$, which is used later by T to encrypt the communication of the anonymization protocol. Moreover, R computes three public-key encryptions $E \leftarrow \mathsf{Enc}_{pk_{\mathsf{A}}}(A)$, $F \leftarrow \mathsf{Enc}^{\mathfrak{h}}_{pk_{\mathsf{R}}}(T)$ and $G \leftarrow \mathsf{Enc}^{\mathfrak{h}}_{pk_{\mathsf{R}}}(\mathtt{ID})$. The ciphertext E is used to transport the symmetric key A from T to A in the anonymization protocol whereas F and G are used to transport the ephemeral secret T and the identifier $\mathtt{ID}$ from T to R in the identification protocol. Finally, R updates its database $d \leftarrow d \cup \{(\mathtt{ID}, K)\}$ and initializes T with the state $S \leftarrow (A, T, E, F, G, \mathtt{ID}, K)$.

[6] Note that personal anonymizers (i.e., those running on the users' mobile phone or PDA) can have different user-specific keys. However, this requires the user of a personal anonymizer to indicate to the tag issuing entity which anonymizer shall be used later to anonymize the newly created tag.

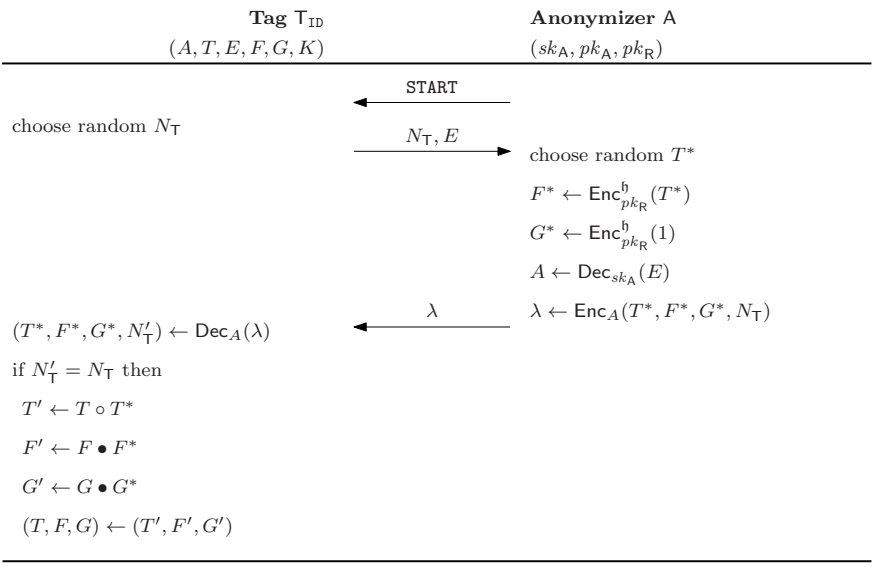

Fig. 1. The anonymization protocol

Anonymization Protocol. The anonymization protocol is illustrated in Figure 1. It is a protocol between a tag T with identifier $\mathtt{ID}$ and an anonymizer A and aims at updating the state S of T. First, T randomly chooses N_{T} and sends (N_{T}, E) to A. Then, A chooses a new ephemeral tag secret T^* and encrypts it to $F^* \leftarrow \mathsf{Enc}^{\mathfrak{h}}_{pk_{\mathsf{R}}}(T^*)$. Moreover, A encrypts $G^* \leftarrow \mathsf{Enc}^{\mathfrak{h}}_{pk_{\mathsf{R}}}(1)$ of the identity w.r.t. to the homomorphic operation $\circ$ of the public-key encryption scheme. Finally, A decrypts $A \leftarrow \mathsf{Dec}_{sk_{\mathsf{A}}}(E)$, encrypts $\lambda \leftarrow \mathsf{Enc}_A(T^*, F^*, G^*, N_{\mathsf{T}})$ and sends λ to T. Then, T decrypts $(T^*, F^*, G^*, N'_{\mathsf{T}}) \leftarrow \mathsf{Dec}_A(\lambda)$ and checks if $N'_{\mathsf{T}} = N_{\mathsf{T}}$. If this is the case, T computes a new ephemeral authentication secret $T' \leftarrow T \circ T^*$, the (homomorphic) public-key encryption $F' \leftarrow F \bullet F^*$ of the new ephemeral key T' and a new (re-randomized) encryption $G' \leftarrow G \bullet G^*$ of the tag identifier $\mathtt{ID}$. Finally, T updates its state $(T, F, G) \leftarrow (T', F', G')$. If $N'_{\mathsf{T}} \neq N_{\mathsf{T}}$, T aborts the anonymization protocol without updating its state.

Identification Protocol. Figure 2 illustrates the identification protocol, which takes place between a tag T with identifier $\mathtt{ID}$ and the reader R with the goal to identify T on the reader side. R sends a random N_{R} to T, which then computes $D \leftarrow \mathsf{RO}(N_{\mathsf{R}}, F, G, T, K)$ and responds with (D, F, G). Then, R decrypts $\mathtt{ID}' \leftarrow \mathsf{Dec}^{\mathfrak{h}}_{sk_{\mathsf{R}}}(G)$ and checks if its secret database d contains a tuple $(\mathtt{ID}', K')$. If this is the case, R decrypts $T' \leftarrow \mathsf{Dec}^{\mathfrak{h}}_{sk_{\mathsf{R}}}(F)$ and accepts T by returning $\mathtt{ID}'$ if $D = \mathsf{RO}(N_{\mathsf{R}}, F, G, T', K')$. Otherwise, R rejects T and returns $\perp$.

Technical Feasibility. Using the (homomorphic) El Gamal public-key encryption scheme, our protocol requires tags to provide about 0.6 KBytes of non-volatile memory. Anonymization requires the tag to generate a random number,

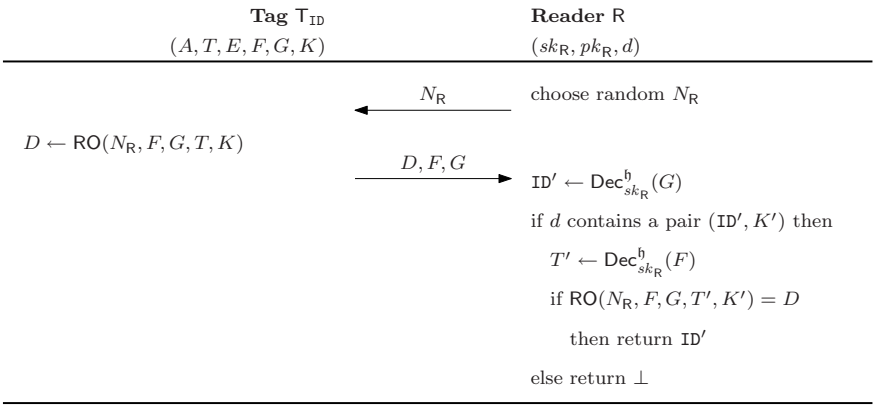

Fig. 2. The identification protocol

decrypt one symmetric ciphertext and to perform five modular multiplications. Identification only requires the tag to evaluate a hash function. Note that the anonymization protocol is completely transparent to the user whereas identification usually requires the user to wait (e.g., at a door) until the authentication protocol completes. Thus, in contrast to the anonymization protocol, most practical applications have strict time constraints on the identification protocol. Our scheme should be implementable with widely available RFID tags.

4 The Anonymizer-Enabled RFID Model

To prove the security and privacy properties claimed in Section 3.3, an appropriate security and privacy model is needed. Since existing RFID security and privacy models do not capture anonymizer-enabled protocols (see Section 2), we extend the model of [18] to the first universal security and privacy model for anonymizer-enabled RFID systems.

4.1 System Model

To form the anonymizer-enabled model, the original system model of [18] must additionally consider the anonymizers A and the corresponding protocols. This means that there must be a procedure to set up A and an interactive protocol where A updates the state of the tags. Following [18], we now define an anonymizer-enabled RFID system.

$\mathsf{SetupReader}(1^{l_{\mathsf{R}}}) \rightarrow (sk_{\mathsf{R}}, pk_{\mathsf{R}}, d)$ on input of a security parameter l_{R}, this function initializes the reader R by creating some public parameters pk_{R} that are known to all entities and some secret parameters sk_{R} that are only known to R. This function also creates a secret database d that can only be accessed by R and that stores the identities and authentication secrets of all legitimate tags.

$\mathsf{SetupAnon}(1^{l_\mathsf{A}}, pk_\mathsf{R}) \rightarrow (sk_\mathsf{A}, pk_\mathsf{A})$ on input of a security parameter l_A and the public key pk_R of R, this function initializes the anonymizer A by creating some public parameters pk_A that are known to all entities and some secret parameters sk_A that are only known to A.

$\mathsf{SetupTag}_{pk_\mathsf{R}}(\mathtt{ID}, pk_\mathsf{A}) \rightarrow (K, S)$ generates a tag-specific secret K and uses the public key pk_R of R to create an initial state S for the tag T with identifier $\mathtt{ID}$. T is initialized with S and $(\mathtt{ID}, K)$ is stored in the secret database d of R. Since T must identify the anonymizer A in the anonymization protocol, this procedure involves pk_A.

$\mathsf{AnonTag}[\mathsf{T}_\mathtt{ID}(S) \leftrightarrow \mathsf{A}(sk_\mathsf{A}, pk_\mathsf{A}, pk_\mathsf{R})] \rightarrow S'$ is an interactive protocol that is (frequently) run between the tag T with identifier $\mathtt{ID}$ and the anonymizer A. The goal of this protocol is to update the state S of T to a new indistinguishable state S'.

$\mathsf{IdentTag}[\mathsf{T}_\mathtt{ID}(S) \leftrightarrow \mathsf{R}(sk_\mathsf{R}, pk_\mathsf{R}, d)] \rightarrow out$ is an interactive protocol between the tag T with identifier $\mathtt{ID}$ and the reader R. The goal of this protocol is to identify T and to verify whether T is legitimate. With overwhelming probability, R returns $out = \mathtt{ID}$ if T is legitimate and $out = \perp$ otherwise.[7]

4.2 Adversary Model

The adversary model of [18] defines the privacy and security objectives as a security experiment, where a polynomially bounded adversary can interact with a set of oracles that model the capabilities of the adversary.

In the anonymizer-enabled model, an adversary may obtain information from the anonymization protocol. This ability is modeled by allowing the adversary to launch new anonymization protocol sessions and to interact with the anonymizer. To consider the case where the adversary controls a set of anonymizers, we allow the adversary to obtain the secrets of the anonymizers by corrupting them. However, as discussed in Section 3.1 and stated in Assumption 1, we assume that there is at least one honest anonymizer in the system whose communication cannot be eavesdropped or manipulated by the adversary. In the anonymizer-enabled model, the adversary has access to the oracles described below.

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

$\mathsf{DrawTag}(\Delta) \rightarrow (\mathtt{vtag}_1, b_1, \ldots, \mathtt{vtag}_n, b_n)$ Initially, the adversary cannot interact with any tag but must query the $\mathsf{DrawTag}$ oracle to get access to a set of tags that has been chosen according to a given tag distribution Δ. This

[7] A *false negative* occurs when T is legitimate but $out = \perp$, a *false positive* happens if T is not legitimate and $out \neq \perp$. An *incorrect identification* occurs if the tag T with identifier $\mathtt{ID}$ is legitimate but $out \notin \{\mathtt{ID}, \perp\}$.

models the fact that the adversary can only interact with the tags within his reading range. The adversary usually only knows the tags he can interact with by some temporary identifiers $\mathtt{vtag}_1, \ldots, \mathtt{vtag}_n$ (e.g., in our protocol the tuple (F, G) can be seen as virtual identifier). The DrawTag oracle manages a secret look-up table $\mathcal{T}$ that keeps track of the real identifier $\mathtt{ID}_i$ that is associated with each temporary identifier $\mathtt{vtag}_i$, i.e., $\mathcal{T}(\mathtt{vtag}_i) = \mathtt{ID}_i$. Moreover, the DrawTag oracle also provides the adversary with information on whether the corresponding tags are legitimate ($b_i = 1$) or not ($b_i = 0$). This models the availability of auxiliary information to the adversary.[8]

FreeTag(vtag) Contrary to the DrawTag oracle, the FreeTag oracle makes a tag vtag inaccessible to the adversary, which means that the adversary cannot interact with vtag any longer until it is made accessible again (under a new temporary identifier vtag′) by another DrawTag query. This models the fact that a tag can get out of the reading range of the adversary.

LaunchIdent( ) $\rightarrow \pi_{\mathsf{R}}$ makes the reader to start a new instance π_{R} of the IdentTag protocol, which allows the adversary to start different parallel IdentTag protocol instances with the reader R.

LaunchAnon( ) $\rightarrow \pi_{\mathsf{A}}$ makes the anonymizer to start a new instance π_{A} of the AnonTag protocol, which allows the adversary to start different parallel AnonTag protocol instances with an honest anonymizer.

SendTag$(m, \mathtt{vtag}) \rightarrow m'$ sends a message m to the tag T that is known as vtag to the adversary. The tag T responds with message m'. This allows the adversary to perform active attacks against both the AnonTag and the IdentTag protocol.

SendReader$(m, \pi_{\mathsf{R}}) \rightarrow m'$ sends a message m to the instance π_{R} of the IdentTag protocol that is executed by the reader R, which responds with message m'. This allows the adversary to perform active attacks against the IdentTag protocol.

SendAnon$(m, \pi_{\mathsf{A}}) \rightarrow m'$ sends a message m to the instance π_{A} of the AnonTag protocol that is executed by an honest anonymizer A, which responds with message m'. This allows the adversary to perform active attacks against the AnonTag protocol.

Result(π_{R}) returns 1 if the instance π_{R} of the IdentTag protocol has been completed but the tag T that participates in the protocol has not been accepted by the reader R. In case R identified a legitimate tag, Result returns 0. This allows the adversary to obtain auxiliary information on whether the authentication of T was successful or not.

CorruptTag(vtag) $\rightarrow S$ returns the current state S of the tag T that is known as vtag to the adversary. This models (physical) attacks on tags that disclose the current tag state.

CorruptAnon(A) $\rightarrow (sk_{\mathsf{A}})$ returns the secret parameter sk_{A} of anonymizer A. This models (physical) attacks against honest anonymizers that disclose the secret sk_{A} of anonymizer A.

[8] For instance, in an access control scenario, the adversary may notice that a tag $\mathtt{vtag}_i$ is legitimate by observing its communication with a reader at a locked door and then watching whether the door opens or not.

As discussed in Section 3.1, we make the following assumption:

Assumption 1. *A tag* T *with identifier* $\mathtt{ID}$ *always runs* $\mathsf{AnonTag}[\mathsf{T}_{\mathtt{ID}} \leftrightarrow \mathsf{A}]$ *with an honest anonymizer* A *at least once before each execution of* $\mathsf{IdentTag}[\mathsf{T}_{\mathtt{ID}} \leftrightarrow \mathsf{R}^*]$ *with a (potentially malicious) reader* R^* *and before each* $\mathsf{CorruptTag}(\mathtt{vtag})$ *query where* $\mathcal{T}(\mathtt{vtag}) = \mathtt{ID}$.

Adversary Classes. The original model of [18] distinguishes the following four major adversary classes that represent adversaries of different strength:

- *Weak adversaries* cannot corrupt tags and are limited to active attacks on the protocols. This assumes that corruption of tags is infeasible (e.g., due to tamper-resistant hardware), which is clearly not the case for low-cost RFIDs.
- *Forward adversaries* cannot interact with the RFID system (i.e., all the oracles described above) any longer after corrupting any of the tags for the first time but they can still make CorruptTag queries to all other tags. This models the case where the secrets of the tags become known when the life of the system is over.
- *Destructive adversaries* can never use a tag again after it has been corrupted but can still query all oracles for any of the remaining non-corrupted tags. This assumes that tags are destroyed when they are corrupted (e.g., due to tamper-evident hardware).
- *Strong adversaries* have full access to all of the oracles at any time.

Moreover, [18] defines *narrow* variants of the four adversary classes described above. A *narrow adversary* cannot obtain auxiliary information (i.e., on whether a tag is legitimate or not). This may be the case in application scenarios where the result of the identification protocol cannot be observed by the adversary. Therefore, a narrow adversary cannot query the Result oracle and is not given the values $(b_1, \ldots, b_n)$ from the DrawTag oracle, which both are the only sources of auxiliary information.

4.3 Security Definition

The security definition of [18] considers attacks where the adversary aims to impersonate or to forge a legitimate tag. More precisely, the definition is based on a security experiment $\mathbf{Exp}^{\mathrm{sec}}_{\mathcal{A}_{\mathrm{sec}}}$ where a *strong adversary* must create an instance $\mathsf{LaunchIdent}(\,) \rightarrow \pi_{\mathsf{R}}$ of the IdentTag protocol with the reader R and finish this protocol instance π_{R} with a query $\mathsf{SendReader}(m, \pi_{\mathsf{R}})$. Note that $\mathcal{A}_{\mathrm{sec}}$ can arbitrarily interact with all of the oracles defined in Section 4.2 at any time during the experiment. The adversary $\mathcal{A}_{\mathrm{sec}}$ wins if (i) R identifies a legitimate tag $\mathtt{ID}$ in the instance π_{R} of the IdentTag protocol, (ii) tag $\mathtt{ID}$ has not been corrupted and (iii) tag $\mathtt{ID}$ and R have not run any instance $\pi_{\mathsf{R}}{}'$ of the IdentTag protocol that generated the same messages as instance π_{R} (i.e., π_{R} is not a *replay* of an old transcript $\pi_{\mathsf{R}}{}'$). Let $\mathbf{Exp}^{\mathrm{sec}}_{\mathcal{A}_{\mathrm{sec}}} = 1$ denote the case where the adversary $\mathcal{A}_{\mathrm{sec}}$ wins this security experiment.

Definition 1 (Security [18]). *An RFID system (as defined in Section 4.1) is secure if for any strong adversary* $\mathcal{A}_{\text{sec}}$ *the probability* $\Pr[\mathbf{Exp}^{\text{sec}}_{\mathcal{A}_{\text{sec}}} = 1]$ *is negligible.*

Definition 1 can be used in the anonymizer-enabled model as it is. Note that the adversary $\mathcal{A}_{\text{sec}}$ is allowed to corrupt all the anonymizers when playing the security experiment described above. This models the fact that anonymizers should not be able to clone or to forge tags.

4.4 Privacy Definition

The privacy definition of [18] is very flexible and, dependent on the class of adversaries considered (in Section 4.2), it covers different notions of privacy. For strong adversaries the definition considers anonymity, backward and forward untraceability.

The privacy definition requires the communication of a tag T to not reveal any information that helps an adversary $\mathcal{A}_{\text{prv}}$ to trace or to identify T. It is based on the existence of a simulator $\mathcal{B}$ that can simulate the communication of T to $\mathcal{A}_{\text{prv}}$ without using any of the secrets of the RFID system. $\mathcal{B}$ must answer all queries of $\mathcal{A}_{\text{prv}}$ by only using the inputs and outputs of the oracle queries that $\mathcal{A}_{\text{prv}}$ previously made (i.e., $\mathcal{B}$ "sees" what $\mathcal{A}_{\text{prv}}$ "sees"). In case the success probability of $\mathcal{A}_{\text{prv}}$ does not change significantly when interacting with $\mathcal{B}$ instead of the real RFID system, the communication of T does not help $\mathcal{A}_{\text{prv}}$ to break the privacy properties of the RFID scheme. In [18], $\mathcal{B}$ is called *blinder* and an adversary $\mathcal{A}^{\mathcal{B}}_{\text{prv}}$ who interacts with $\mathcal{B}$ is called *blinded adversary.*

More formally, the privacy definition considers a security game $\mathbf{Exp}^{\text{prv}}_{\mathcal{A}_{\text{prv}}}$ where an adversary $\mathcal{A}_{\text{prv}}$ must distinguish whether he interacts with the real RFID system or a blinder $\mathcal{B}$. Therefore, $\mathcal{A}_{\text{prv}}$ first performs an attack phase that is followed by an analysis phase. In the attack phase, $\mathcal{A}_{\text{prv}}$ is allowed to interact with the oracles described in Section 4.2 in an arbitrary way. In the analysis phase, $\mathcal{A}_{\text{prv}}$ cannot access the oracles any more but is given access to the secret table $\mathcal{T}$ of the DrawTag oracle, which allows $\mathcal{A}_{\text{prv}}$ to link the temporary identifiers `vtag` of all the tags he interacted with to their corresponding real identities `ID`. Finally, $\mathcal{A}_{\text{prv}}$ must return a bit b to indicate whether he interacted with a blinder $\mathcal{B}$ ($b = 1$) or the real RFID system ($b = 0$). This leads to the privacy definition described below.

Definition 2 (Privacy [18]). *Let P be one of the adversary classes defined in Section 4.2. An RFID system (as defined in Section 4.1) is P-private if for any adversary* $\mathcal{A}_{\text{prv}}$ *of class P there exists a blinder $\mathcal{B}$ such that* $|\Pr[\mathbf{Exp}^{\text{prv}}_{\mathcal{A}_{\text{prv}}} = 1] - \Pr[\mathbf{Exp}^{\text{prv}}_{\mathcal{A}^{\mathcal{B}}_{\text{prv}}} = 1]|$ *is negligible.*

The communication of a tag is modeled by the LaunchIdent, SendReader, SendTag and the Result oracle. Thus, a blinder $\mathcal{B}$ must simulate these oracles. In the anonymizer-enabled model, we additionally have the LaunchAnon and the SendAnon oracles that model the interaction of a tag with the anonymizer. However, as

discussed in Section 3.3, we are not concerned of the privacy of the communication between tags and the anonymizer. Thus, the LaunchAnon and the SendAnon oracle need not to be simulated by $\mathcal{B}$. Note that the CorruptTag query is not simulated by $\mathcal{B}$ because Definition 2 only captures the privacy loss of the wireless communication of tags.

5 Security Analysis

Theorem 1. *The RFID system presented in Section 3.3 is correct, secure in the random oracle model, narrow-strong and forward private in the random oracle model under Assumption 1 if the homomorphic public-key encryption scheme is CPA-secure.*

Note that Assumption 1 is *only* required to ensure the privacy properties of our scheme. Security (against impersonation attacks) also holds if there is no (honest) anonymizer in the system.

Due to space restrictions, we only give proof sketches and provide full proofs in the full version of the paper [52].

Correctness. No false negative can be produced since each legitimate tag T will always be accepted by the reader R. A false positive cannot be produced since the decryption of G outputs a unique ID and, if ID is not in the database d, R immediately rejects the identification. □

Security. The idea of the security proof is as follows: by contradiction, we assume that there is a narrow-strong adversary $\mathcal{A}_{\text{sec}}$ (as defined in Section 4.2), who wins the security game of Definition 1. Given $\mathcal{A}_{\text{sec}}$, one can construct a p.p.t. algorithm that finds a collision to the random oracle with non-negligible probability. However, by the pseudorandomness of the random oracle, this can happen with at most negligible probability. □

Narrow-Strong Privacy. The idea of the privacy proof is as follows: by contradiction, we assume that there is a narrow-strong adversary $\mathcal{A}_{\text{prv}}$ (as defined in Section 4.2), who wins with non-negligible probability the game of Definition 2. Given such an adversary $\mathcal{A}_{\text{prv}}$, one can construct a p.p.t. algorithm that breaks the CPA-security of the homomorphic public key encryption scheme with non-negligible probability. However, since the encryption scheme is assumed to be CPA-secure, this can happen with at most negligible probability, which is a contradiction. □

Forward-Privacy. To prove forward-privacy, we can use the following lemma from [18]:

Lemma 1. *For every secure RFID scheme that has the property that, whenever a legitimate tag* T *and the reader* R *have executed a complete run of the* IdentTag *protocol in a secure environment (i.e., where no adversary can manipulate the protocol-run), the output out of* R *is never* $\perp$ *(i.e.,* R *does never reject a legitimate tag), it holds that narrow-forward privacy implies forward-privacy.*

Our scheme is narrow-strong private, which implies narrow-forward privacy [18]. Moreover, it is correct and secure, which means that it fulfills all requirements to apply Lemma 1. Since the original proof of Lemma 1 is also valid in the anonymizer-enabled model, we can apply Lemma 1 to prove that our scheme achieves forward privacy. □

Acknowledgments

The second author wishes to thank Paolo D'Arco and Alessandra Scafuro for several useful discussions about RFID privacy notions.

The work of the authors has been supported in part by the European Commission through the EU ICT program under Contract ICT-2007-216646 ECRYPT II. The work of the second author has also been supported in part by the European Commission through the FP7 Information Communication Technologies programme, under Contract FET-215270 FRONTS.

References

1. Atmel Corporation: Innovative IDIC solutions (2007), http://www.atmel.com/dyn/resources/prod_documents/doc4602.pdf
2. Molnar, D., Wagner, D.: Privacy and security in library RFID: Issues, practices, and architectures. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, pp. 210–219. ACM Press, New York (2004)
3. Calypso Networks Association: Web site of Calypso Networks Association. (May 2007), http://www.calypsonet-asso.org/
4. NXP Semiconductors: MIFARE smartcard ICs (September 2008), http://www.mifare.net/products/smartcardics/
5. Sony Global: Web site of Sony FeliCa (June 2008), http://www.sony.net/Products/felica/
6. Sadeghi, A.R., Visconti, I., Wachsmann, C.: User privacy in transport systems based on RFID e-tickets. In: International Workshop on Privacy in Location-Based Applications (PiLBA), Malaga, Spain, October 9 (2008)
7. I.C.A. Organization: Machine Readable Travel Documents, Doc 9303, Part 1 Machine Readable Passports, Fifth Edition (2003)
8. Juels, A.: RFID security and privacy: A research survey. Journal of Selected Areas in Communication 24(2), 381–395 (2006)
9. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and privacy aspects of low-cost radio frequency identification systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 50–59. Springer, Heidelberg (2004)
10. Lim, C.H., Kwon, T.: Strong and robust RFID authentication enabling perfect ownership transfer. In: Ning, P., Qing, S., Li, N. (eds.) ICICS 2006. LNCS, vol. 4307, pp. 1–20. Springer, Heidelberg (2006)
11. Spirtech: CALYPSO functional specification: Card application, version 1.3. (October 2005), http://calypso.spirtech.net/

12. Octopus Holdings: Web site of Octopus Holdings (June 2008), http://www.octopus.com.hk/en/
13. Avoine, G.: Adversarial model for radio frequency identification. Cryptology ePrint Archive, Report 2005/049 (2005)
14. Juels, A., Weis, S.A.: Defining strong privacy for RFID. Cryptology ePrint Archive, Report 2006/137 (2006)
15. Damgård, I., Østergaard, M.: RFID security: Tradeoffs between security and efficiency. Cryptology ePrint Archive, Report 2006/234 (2006)
16. Burmester, M., van Le, T., de Medeiros, B.: Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In: Proceedings of Second International Conference on Security and Privacy in Communication Networks (SecureComm), pp. 1–9. IEEE Computer Society, Los Alamitos (2006)
17. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic approach to privacy-friendly tags (November 2003)
18. Vaudenay, S.: On privacy models for RFID. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)
19. Paise, R.I., Vaudenay, S.: Mutual authentication in RFID: Security and privacy. In: ASIACCS 2008: Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, pp. 292–299. ACM Press, New York (2008)
20. Ng, C.Y., Susilo, W., Mu, Y., Safavi-Naini, R.: RFID privacy models revisited. In: [53], pp. 251–256
21. Juels, A., Weis, S.A.: Authenticating pervasive devices with human protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005)
22. Katz, J., Shin, J.S.: Parallel and Concurrent Security of the HB and HB+ Protocols. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 73–87. Springer, Heidelberg (2006)
23. Katz, J., Smith, A.: Analyzing the HB and HB+ protocols in the large error case. Cryptology ePrint Archive, Report 2006/326 (2006)
24. Katz, J.: Efficient Cryptographic Protocols Based on the Hardness of Learning Parity with Noise. In: Galbraith, S.D. (ed.) Cryptography and Coding 2007. LNCS, vol. 4887, pp. 1–15. Springer, Heidelberg (2007)
25. Gilbert, H., Robshaw, M., Silbert, H.: An active attack against HB+ — A provable secure leightweight authentication protocol. Cryptology ePrint Archive, Report 2007/237 (2007)
26. Gilbert, H., Robshaw, M.J.B., Seurin, Y.: Good Variants of HB+ Are Hard to Find. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143, pp. 156–170. Springer, Heidelberg (2008)
27. Ouafi, K., Overbeck, R., Vaudenay, S.: On the Security of HB# against a Man-in-the-Middle Attack. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 108–124. Springer, Heidelberg (2008)
28. Frumkin, D., Shamir, A.: Un-Trusted-HB: Security Vulnerabilities of Trusted-HB. Cryptology ePrint Archive, Report 2009/044 (2009)
29. Levieil, E., Fouque, P.A.: An Improved LPN Algorithm. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 348–359. Springer, Heidelberg (2006)
30. Tsudik, G.: YA-TRAP: Yet Another Trivial RFID Authentication Protocol. In: Security in Pervasive Computing. LNCS, vol. 2802, pp. 640–643. IEEE Computer Society, Los Alamitos (2006)

31. Henrici, D., Müller, P.: 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, pp. 149–153. IEEE Computer Society, Los Alamitos (2004)
32. Ohkubo, M., Suzuki, K., Kinoshita, S.: Efficient hash-chain based RFID privacy protection scheme. In: International Conference on Ubiquitous Computing (UbiComp), Workshop Privacy: Current Status and Future Directions (September 2004)
33. Dimitriou, T.: 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), pp. 59–66. IEEE Computer Society, Los Alamitos (2005)
34. Song, B., Mitchell, C.J.: RFID authentication protocol for low-cost tags. In: Proceedings of the First ACM Conference on Wireless Network Security, pp. 140–147. ACM Press, New York (2008)
35. Sadeghi, A.R., Visconti, I., Wachsmann, C.: Location privacy in RFID applications. In: Bettini, C., et al. (eds.) Privacy in Location-Based Applications: Research Issues and Emerging Trends. LNCS, vol. 5599, pp. 127–150. Springer, Heidelberg (2009)
36. Juels, A., Pappu, R.: Squealing Euros: Privacy protection in RFID-enabled banknotes. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 103–121. Springer, Heidelberg (2003)
37. Golle, P., Jakobsson, M., Juels, A., Syverson, P.: Universal re-encryption for mixnets. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 163–178. Springer, Heidelberg (2004)
38. Saito, J., Ryou, J.C., Sakurai, K.: Enhancing privacy of universal re-encryption scheme for RFID tags. In: Yang, L.T., Guo, M., Gao, G.R., Jha, N.K. (eds.) EUC 2004. LNCS, vol. 3207, pp. 879–890. Springer, Heidelberg (2004)
39. Ateniese, G., Camenisch, J., de Medeiros, B.: Untraceable RFID tags via insubvertible encryption. In: Proceedings of the 12th ACM Conference on Computer and Communications Security, pp. 92–101. ACM Press, New York (2005)
40. Economist: Security technology: Where's the smart money? The Economist, 69–70 (February 2002)
41. Juels, A.: Minimalist cryptography for low-cost RFID tags (extended abstract). In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 149–164. Springer, Heidelberg (2005)
42. Avoine, G., Dysli, E., Oechslin, P.: Reducing time complexity in RFID systems. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 291–306. Springer, Heidelberg (2006)
43. Ha, J.H., Moon, S.J., Zhou, J., Ha, J.C.: A new formal proof model for RFID location privacy. In: [53], pp. 267–281.
44. D'Arco, P., Scafuro, A., Visconti, I.: Semi-Destructive Privacy in DoS-Enabled RFID systems. In: Proceedings of RFIDSec 2009 (July 2009)
45. D'Arco, P., Scafuro, A., Visconti, I.: Revisiting DoS attacks and privacy in rfid-enabled networks. In: Dolev, S. (ed.) ALGOSENSORS 2009. LNCS, vol. 5804, p. 263. Springer, Heidelberg (2009)
46. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
47. Prabhakaran, M., Rosulek, M.: Homomorphic encryption with CCA security. Cryptology ePrint Archive, Report 2005/079 (2008)

48. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28, 270–299 (1984)
49. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer, Heidelberg (1998)
50. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing-efficient protocols. In: Proceedings of the Annual Conference on Computer and Communications Security (CCS) (1994)
51. Danev, B., Heydt-Benjamin, T.S., Capkun, S.: Physical-layer Identification of RFID Devices. In: 18th USENIX Security Symposium, Montreal, Canada, August 10-14, pp. 199–214 (2009)
52. Sadeghi, A.R., Visconti, I., Wachsmann, C.: Efficient RFID security and privacy with anonymizers. In: Proceedings of RFIDSec 2009 (July 2009)
53. Jajodia, S., Lopez, J. (eds.): ESORICS 2008. LNCS, vol. 5283, p. 602. Springer, Heidelberg (2008)