RFID Security: Tradeoffs between Security and Efficiency

Ivan Damgård and Michael Østergaard Pedersen

University of Aarhus
{ivan,michael}@daimi.au.dk

Abstract. We propose a model and definition for anonymous (group) identification that is well suited for RFID systems. This is based on the definition of Juels and Weis of strong privacy for RFID tags, where we add requirements for completeness and soundness. We also propose a weaker and more realistic definition of privacy. For the case where tags hold independent keys, we prove a conjecture by Juels and Weis, namely in a strongly private and sound RFID system using only symmetric cryptography, a reader must access virtually all keys in the system when reading a tag. It was already known from work by Molnar, Soppera and Wagner that when keys are dependent, the reader only needs to access a logarithmic number of keys, but at a cost in terms of privacy: For that system, privacy is lost if an adversary corrupts just a single tag. We propose protocols offering a new range of tradeoffs between security and efficiency. For instance, the number of keys accessed by a reader to read a tag can be significantly smaller than the number of tags while retaining soundness and privacy, as long as we assume suitable limitations on the adversary.

1 Introduction

RFID tags are small wireless devices that react to electromagnetic fields generated by an RFID reader; they can emit some prestored information and can also do computation. The computing power one can assume an RFID tag to have, however, is severely limited in many applications by requirements for extremely low price tags. RFID technology holds great promise in many scenarios, but can also lead to serious privacy problems, for instance because it becomes possible to track the behavior and whereabouts of people carrying tagged items.

Several research works have proposed protocols for addressing the privacy problem in RFID systems. However, until recently, not much work has addressed formal definitions of security for RFID systems. In [9], Juels and Weis propose a definition of what they call "strong privacy" (based on earlier work by Avoine [2]). Strong privacy is indeed a strong notion, primarily because the adversary is given a lot of power: He can corrupt any number of tags (but not the reader) and read their contents, he can eavesdrop and schedule the tag/reader communication any way he wants, and he can himself select the tags whose privacy he wants to break. In independent work, Burmester, Le and Medeiros

T. Malkin (Ed.): CT-RSA 2008, LNCS 4964, pp. 318–332, 2008.

propose a security definition based on Canetti's Universal Composability framework [6] and they also propose a protocol secure in their model [14].

The work of Juels and Weis only addresses privacy, that is, making sure that the communication of a tag does not allow an external adversary to determine the identity of the tag. Of course, another natural requirement is that a reader should be able to determine whether the tag it reads is valid and not fabricated by an adversary, for instance. Indeed, if this was not required, tags could just return random information all the time or just not reply at all. This would trivially be private, but would of course lead to a useless system.

In this paper, we propose an extension to the strong privacy definition so one can also require completeness and soundness, with the intuitive meaning that the reader accepts valid tags and valid tags only. More specifically, soundness in the weakest sense means that we assume the adversary cannot corrupt tags, and when the reader accepts an instance of the read protocol, an (uncorrupted) tag has been involved in that instance at some point. So in this weak flavor, it is not required that the reader knows which tag it has been talking to. We also suggest a stronger version where corruptions are allowed and the reader must output the identity of the (honest) tag that was involved.

The concept of strongly private and sound systems is closely related to existing concepts for anonymous identification schemes, such as identity escrow schemes [11] or group signature schemes [1,7,10]. They are not the same, however: Our system model is designed to model RFID systems, and where identity escrow and group signature schemes are by definition public-key techniques, we want to cover techniques based on secret-key algorithms only.

The most important privacy issue regarding RFID tags is the issue of being able to systematically track individuals as they carry RFID enabled goods from the supermarket, embedded in the their clothes, etc. In this scenario, it is reasonable to assume that the adversary cannot himself choose the tags he wants to track. Strong privacy is therefore more than we need in this scenario, so we introduce a weaker, but more suitable, definition called *benign-selection privacy.*

Juels and Weis suggest a system that satisfies their definition, building on earlier work by Weis, Sarma and Rivest [15]. In this scheme, each tag is given an independently chosen key, and the reader must search exhaustively through all keys every time a tag is read. This of course does not scale well, but Juels and Weis conjecture that this is, in a certain sense, unavoidable: In strongly private systems that use only symmetric cryptography, and where tags are independently keyed, the reader must access all, or at least a large fraction of the keys in the system. Here, we prove this conjecture. We need to assume that the system is complete and sound, but this is of course a natural requirement and is necessary anyway to exclude degenerate cases, such as when tags only send random information.

The limitation to symmetric cryptography is clearly necessary: With public-key technology, a tag could send its identity encrypted under the reader's public-key, and then prove its identity using some shared-key technique, for instance. This does not require the reader to look at any information that is not related

to the relevant tag. There has in fact been recent work in the direction of implementing public-key on very small devices [5], but even if public-key enabled RFID tags are only slightly more expensive than symmetric-key only tags, this will still inhibit the use of public-key technology in large scenarios that require millions of tags, in order to maximize profit. Therefore we believe the question of what can be done with symmetric techniques is of interest, both theoretically and in practice.

The limitation to schemes with independent keys is not surprising. It follows from work by Molnar, Soppera and Wagner [12] that when dependent keys are allowed, we can have a system where the reader only needs to look at a logarithmic (in the number of tags) number of keys. This comes at the price that strong privacy only holds if the adversary is "radio-only", i.e., he does not corrupt any tags. If the adversary corrupts even a single tag, strong privacy is lost, and benign-selection privacy is lost with large probability. This makes it natural to ask if there are alternative solutions where we can get some amount of privacy with a larger number of corruptions without going back to systems where the reader does exhaustive search over all keys.

In this paper, we first argue that for a wide range of RFID systems, there has to be a tradeoff between the efficiency of the reader and the resources we can allow the adversary to have. We then propose a class of protocols offering a new range of tradeoffs between security and efficiency. For instance, the number of keys accessed by a reader to read a tag can be significantly smaller than the number of tags while retaining soundness and privacy, as long as we assume suitable limitations on the adversary.

2 Model and Definition

Juels and Weis define *strong privacy* for RFID systems using a model of which we give a summary here, for details refer to [9].

The system consists of tags $\mathcal{T}_i, i = 1..n$ and a reader $\mathcal{R}$. For simplicity, we assume that there is only one reader. Tags can receive SetKey messages which will cause the tag to reveal its secret key, and the caller may then send a new key to the tag. This can be used to initialize the system and also models an attacker corrupting a tag to learns its key. A tag may receive a (TagInit, *sid*) message (where *sid* is a session id), which is used in the start of a session. The tag will forget any previous value of *sid*, so a tag may only run a single session at a time. Finally, the tag may respond to a protocol message c_i, called a challenge in [9], by a response r_i. A protocol may consist of several rounds of challenges and responses.

A Reader may receive ReaderInit messages, causing it to generate a fresh session identifier *sid* and a first protocol message c_0 to be sent to a tag. It may also receive pairs of the form (sid, r_i). It will then return either a new message c_{i+1} to be sent to the tag or *Accept* or *Reject*. In [9], a reader, if it returns *Accept*, is not required to say which tag it thinks it has been talking to. We assume here that it may also return the identity of a tag. The reader keeps an

internal log of all challenge and response pairs for each session id that is active, and decides based on this whether to accept or reject. A reader may be involved in several sessions simultaneously, but its behavior in a session only depends on messages it receives in that session and the fixed key material it holds.

We allow the adversary $\mathcal{A}$ to schedule all messages as it wants, and generate its own messages. The adversary is parameterized as follows: r is the number of ReaderInit messages it generates, s is the number of computational steps and t is the number of TagInit messages it generates. Finally, k is a cryptographic security parameter. Juels and Weis do not treat the number of SetKey messages, i.e., the number of corrupted tags, as a separate parameter, but simply say it has to be at most $n-2$. As we shall see, however, the number of corrupted tags is a very important parameter, so we will define u to be the number of tags corrupted by the adversary. A summary of these parameters can be found in Figure 1. Note that this model also captures an adversary that passively listens to a session between reader and tag, namely he starts a session with the reader and one with the tag and simply relays messages between the parties.

k: security parameter	n: number of tags in the RFID system $\mathcal{S}$
r: number of ReaderInit messages allowed	s: number of computational steps allowed
t: number of TagInit messages allowed	u: number of SetKey messages allowed

Fig. 1. Description of parameters

The system is initially setup by running a probabilistic key generation algorithm $\text{Gen}(1^k)$ which produces a set of keys $key_1, ..., key_n$ to be assigned to the tags. Of course, $\mathcal{A}$ does not know these keys initially.

Let $\mathcal{S} = (\text{Gen}, \mathcal{R}, \{\mathcal{T}_i\})$ denote an RFID system. Strong privacy is defined via an experiment called $\mathbf{Exp}^{priv}_{\mathcal{A},\mathcal{S}}[k, n, r, s, t]$. Here, we run the system where the adversary may corrupt tags, initiate sessions, etc., observing the limitations put on him. This ends by the adversary selecting two uncorrupted tags, called $\mathcal{T}_0^*, \mathcal{T}_1^*$. He is then given oracle access to $\mathcal{T}_b^*$ where b is a random bit. He may now again corrupt other tags and initiate sessions, and must finally guess the value of b. However, we have to assume that in this last phase, when using the reader to interact with $\mathcal{T}_b^*$, he only learns whether the reader outputs accept or reject and not the identity found by the reader. Otherwise, he could just let the reader identify $\mathcal{T}_b^*$. The system is said to be (r, s, t)-private if any (r, s, t)-adversary's advantage over $1/2$ in guessing b is negligible as a function of k. We propose here to define also (r, s, t, u)-privacy, which is the same, except that the adversary may only corrupt at most u tags. However, for some systems, the advantage that can be achieved depends not only k, but on all the parameters, and does not tend to 0 as we increase k, if other parameters are constant. We will therefore use a variant of strong privacy here:

Definition 1. *Strong $(k, r, s, t, u, n, \epsilon)$-privacy is defined via the experiment $\mathbf{Exp}^{priv}_{\mathcal{A},\mathcal{S}}[k, n, r, s, t, u]$ which is the same as Juels and Weis' except that the*

adversary can only corrupt up to u tags. We say that the system is strongly $(k, r, s, t, u, n, \epsilon)$-private if any adversary observing the limitations in the experiment has advantage at most ϵ.

Experiment $\mathbf{Exp}^{priv}_{\mathcal{A},\mathcal{S}}[k, n, r, s, t, u]$ **Setup:**

1. $\textsc{Gen}(1^k) \rightarrow (key_0, ..., key_n)$
2. Initialize $\mathcal{R}$ with $(key_0, ..., key_n)$
3. Set each $\mathcal{T}_i$'s key to key_i with a SetKey call

Phase 1 (Learning):

4. $\mathcal{A}$ may do the following in any interleaved order:
   (a) Make ReaderInit calls, without exceeding r overall calls
   (b) Make TagInit calls, without exceeding t overall calls
   (c) Make SetKey calls, without exceeding u overall calls
   (d) Communicate and compute without exceeding s overall steps

Phase 2 (Challenge):

5. $\mathcal{A}$ selects two tags $\mathcal{T}_i$ and $\mathcal{T}_j$ to which it did *not* send SetKey messages
6. Let $\mathcal{T}_0^* = \mathcal{T}_i$ and $\mathcal{T}_1^* = \mathcal{T}_j$ and remove both of these from the current tag set
7. Choose a random bit $b \in \{0, 1\}$ and provide $\mathcal{A}$ access to $\mathcal{T}_b^*$
8. $\mathcal{A}$ may do the following in any interleaved order:
   (a) Make ReaderInit calls, without exceeding r overall calls
   (b) Make TagInit calls, without exceeding t overall calls
   (c) Make SetKey calls, without exceeding u overall calls to any tag in the current tag set
   (d) Communicate and compute without exceeding s overall steps
9. $\mathcal{A}$ outputs a guess bit b'

$\mathcal{A}$ succeeds if $b = b'$

As Juels and Weis note in [9], strong privacy may be too strong a notion for many real world applications. In particular, the adversary can freely choose the target tags he wants to be challenged on. He may not have that much power in real life, where the choice may be forced on him by the environment he operates in. One may try to model this by having the target tags be chosen from some distribution independently of the adversary – this ideas is already present in the work of Avoine [2]. But it is very difficult to single out a distribution that realistically models the environment. We therefore propose a new model called *benign-selection privacy* where we allow any distribution as long as it only selects uncorrupted tags.

Definition 2. *Benign-selection privacy is defined via an experiment called $\mathbf{Exp}^{bspriv}_{\mathcal{A},\mathcal{S},\mathcal{D}}[k, n, r, s, t, u]$ which is the same as $\mathbf{Exp}^{priv}_{\mathcal{A},\mathcal{S}}[k, n, r, s, t, u]$, except that the adversary does not select the two tags $\mathcal{T}_0^*, \mathcal{T}_1^*$. Instead they are chosen at random from distribution D among all uncorrupted tags. We think of D as a probabilistic algorithm that only gets the set of corrupted tags as input, and outputs*

the index of the target tags, i.e., the choice is uncorrelated to adversarial activity other than corruptions. We say that the system is $(k, r, s, t, u, n, \epsilon)$*-private with benign D-selection if any adversary observing the limitations in the experiment has advantage at most* ϵ*.*

In the following, it will often be cumbersome and unnecessarily complicated to specify s, the number of computational steps, exactly. We will often replace s by a $poly(k)$, meaning that the statement involved holds for any adversary that uses time polynomial in k.

It is natural to expect a system as described here to also have the properties that valid tags are accepted, and that the adversary cannot impersonate a tag unless he corrupts it. This aspect was not treated in [9] (but was also not the main goal there). We propose to define this as follows:

Completeness. Assume that at the end of session *sid* the internal log of the reader $\mathcal{R}$ for that session contains pairs (c_j, r_j) where all $r_j's$ were generated by an honest tag in correct order. Completeness means that $\mathcal{R}$ outputs Accept with probability 1 for any such session.

Strong Soundness. Consider the following experiment similar to the privacy experiment of Juels and Weis:

Experiment $\mathbf{Exp}^{sound}_{\mathcal{A},\mathcal{S}}[k, n, r, s, t, u]$:

Setup:

1. $\textsc{Gen}(1^k) \rightarrow (key_0, ..., key_n)$
2. Initialize $\mathcal{R}$ with $(key_0, ..., key_n)$
3. Set each $\mathcal{T}_i$'s key to key_i with a SetKey call

Attack:

4. $\mathcal{A}$ may do the following in any interleaved order:
   (a) Make ReaderInit calls, without exceeding r overall calls
   (b) Make TagInit calls, without exceeding t overall calls
   (c) Make SetKey calls, without exceeding u overall calls
   (d) Communicate and compute without exceeding s overall steps

Let E be the event that occurs if $\mathcal{R}$ at some point outputs $(Accept, i)$ at the end of session *sid* where $\mathcal{T}_i$ is not corrupted, yet $\mathcal{R}$'s internal entry for *sid* only contains pairs (c_j, r_j) where r_j was not sent by T_i as a response to c_j, i.e., $\mathcal{T}_i$ has not been involved in the session. We say that the system provides strong (r, s, t, u)-soundness if the probability that E occurs is negligible in k.

Weak Soundness. Weak (r, s, t)-soundness is defined by the same experiment as above, except that $\mathcal{R}$ now only has to output Accept or Reject at the end of a session, $\mathcal{A}$ is not allowed to corrupt tags, and the error event E is now defined to be that $\mathcal{R}$ outputs Accept, and yet no tag has been involved in the session.

3 Independent Keys

As mentioned earlier, our goal in this section is to prove the speculation by Juels and Weis: In any strongly private, complete and sound RFID system, the reader must access a key for every tag, or at least a large fraction of them, when reading a tag. This can only be expected to hold, however, when keys for different tags are independently chosen, and the system "only" uses symmetric cryptography. If public-key cryptography was allowed, a tag could first encrypt its identity under the reader's public-key, and then show possession of some secret that is shared between reader and this tag only.

To prove something, we need to formalize the constraints on the system. For the independence of keys, this is easy, we simply assume that each tag $\mathcal{T}_i$ gets a key K_i chosen independently from all other keys by a key generation algorithm G_i, i.e., $K_i \leftarrow G_i(1^k)$ where k is the security parameter. As for the constraint that "symmetric cryptography and nothing else is used", we will give the system access to a pseudorandom function, $\phi.(\cdot)$, and we will assume that every key K_i in the system is used only as a key to this function, i.e., tag $\mathcal{T}_i$ or reader use $\phi_{K_i}(\cdot)$ as a black box. This means that we can equivalently give tags and reader oracle access to $\phi_{K_i}(\cdot)$ for any key they need to use. Therefore, when in the following we say that "the reader accesses a key", this means it calls the oracle that holds that key.

Now, to model that the pseudorandom functions are the essential cryptographic resource used, we will simply assume that the keys $\{K_i\}$ held by the reader and tags are the only secret data in the system. More precisely, we think of the reader's algorithm as an interactive Turing machine that takes no private input, but may make oracle calls to $\phi_{K_i}(\cdot)$ for any K_i. Similarly, a tag may only call its own pseudorandom function, whereas the adversary may only call $\phi_{K_i}(\cdot)$ if he has corrupted $\mathcal{T}_i$. We will say that such a system is *essentially symmetric*.

Note that an essentially symmetric system is not prevented from using public-key, or using secret-key techniques in a non-blackbox way – the reader could try to do a Diffie-Hellman key exchange with a tag, for instance, or generate a key for a pseudorandom function and use this key in any way it wants. Nevertheless, the constraints we have defined are sufficient to show what we are after. To get better intuition for why this is the case, one may note that, while the reader is free to generate a public encryption key and send it to a tag, the tag cannot immediately verify that the key comes from the reader and not the adversary. Thus it would not be secure to send the tag's id encrypted under the public key.

The first lemma formalizes the straightforward intuition that if keys are independent, a reader cannot determine if it is talking to a valid tag unless it accesses the key for that tag. More formally:

Lemma 1. *Consider an RFID system that is complete, weakly* $(1, poly(k), 0)$*-sound, and uses independent keys. Consider a session between reader and a tag where the adversary does not modify the traffic. In any such session, the algorithm executed by the reader when reading a tag* $\mathcal{T}_i$ *will access* K_i*, except with negligible probability.*

Proof. We consider all probabilities as taken over the choice of keys and the random coins used by tag and reader in the session. Let E be the event that the reader does not access ϕ_{K_i}. By completeness, the reader should accept with probability 1, so the probability that the reader accepts and E occurs equals $Pr(E)$. Assume for contradiction that $Pr(E)$ is non-negligible. Then an adversary could fabricate his own tag $\mathcal{T}_i'$ with a key K_i' generated by G_i, and start a session between this tag and the reader, while simply following the protocol. Now by independence of keys, as long as E occurs, conversations with $\mathcal{T}_i'$ and $\mathcal{T}_i$ are perfectly indistinguishable. Hence, the reader accepts with probability at least $Pr(E)$, which contradicts weak soundness. □

The next theorem uses the observation that in an essentially symmetric system, the *only* difference between the honest reader and an adversary is that the reader has access to all keys, while the adversary initially does not. He can, however, corrupt tags and get access to (some of) the keys. He can therefore potentially run the same algorithm that the reader uses when reading a tag.

Theorem 1. *Assume an essentially symmetric RFID system is complete and weakly* $(1, poly(k), 0)$*-sound. Assume also that the reader algorithm accesses at most* αn *of the keys, for a constant* $\alpha < 1/2$*. Such a system cannot have strong* $(k, 0, poly(k), 1, \alpha n, n, 1/2 - \alpha)$*-privacy*

Proof. We describe an adversary that will break strong privacy for any system that is complete and weakly sound and where only αn oracles are accessed. The adversary picks uniformly a pair of tags $\mathcal{T}_i, \mathcal{T}_j$, and uses these two as the challenge pair $(\mathcal{T}_0^*, \mathcal{T}_1^*)$ from the strong privacy definition. It then gets oracle access to $\mathcal{T}_b^*$, where $b = 0$ or 1 and should try to guess which of the two it is talking to. To do this, it executes the read protocol with $\mathcal{T}_b^*$, and while doing so, it emulates the reader's algorithm. Whenever the reader algorithm wants to access K_t, the adversary corrupts $\mathcal{T}_t$, and may now call the pseudorandom function with key K_t. This goes on until the reader algorithm wants to access K_t where $t = i$ or j. In this case the adversary outputs 0 if $t = i$ and 1 otherwise and then stops.

To analyze the probability that this adversary has success, suppose, for instance, that $b = 0$. Since our adversary follows the protocol when talking to $\mathcal{T}_b^*$, we can apply Lemma 1 to conclude that the reader will access K_i when talking to $\mathcal{T}_b^*$ with probability essentially 1. On the other hand, the probability that it will not access K_j is greater than $1-\alpha$ because only αn keys are accessed (one of which is K_i), and given i, j is uniform over all values different from i. It follows that the adversary's guess is correct with probability $1 - \alpha$ which is a constant greater than $1/2$ and hence we contradict strong privacy. □

Note that since the adversary we construct in the proof selects target tags uniformly, this same argument also shows that a system as specified in the theorem cannot even have benign D-selection privacy where D is the uniform distribution.

One might use some form of pre-computation to perform key lookups more efficiently. For example Avoine, Dysli, and Oechslin [4,3] propose to use Hellman tables [8] in the protocol of Ohkubo, Suzuki and, Kinoshita, to reduce key lookup

time to $O(n^{2/3})$ at the cost of using an additional $O(n^{2/3})$ space [13]. Since the construction of the table requires accessing all keys in the system, methods using Hellman tables do not immediately contradict the lower bound. We can, however, argue that such methods cannot provide both soundness and privacy: To initialize such a table one must predict all possible outputs from the tag, which in turn means that the tag can only have a fixed number of outputs, m. Juels and Weis show how to break strong privacy for such a scheme, simply by querying a tag m times, and use the reader to distinguish it from another tag that has been queried less than m times [9]. Note that the reader can only accept having the same conversation once with a tag, otherwise a simply replay attack could break the soundness.

4 Correlated Keys

We have shown in the previous section that if we want strong privacy and tags have independent keys, the reader has to access least half of the keys in the worst case. This obviously does not scale well, so we now look at how much privacy and soundness we will loose in return for efficiency if we allow the keys to be correlated.

It was already known from the work of Molnar, Soppera and Wagner that using correlated keys, one can obtain the property that the reader only needs to access a logarithmic number of keys [12]. Unfortunately, this comes at the price that strong privacy is lost already if the adversary corrupts a single tag. This is due to the fact that the system works with a pair of keys (K_0, K_1), where half the tags hold K_0, the other half hold K_1 - as well as many other keys, arranged in a tree structure, which is not important here, however. Corrupting a single tag tells the adversary one of the keys, say K_0. The protocol is such that one can easily extract from the responses tags give, a part that is computed only from K_0 or K_1. This gives the adversary a way to compute from the responses of an uncorrupted tag which of the two keys it holds. Since half the tags hold K_0, 2 sessions with random chosen tags will locate two tags holding different keys with probability 1/2. Clearly, using two such tags as the target in the privacy experiment, the adversary can identify with certainty which tag he talks to. It is not even private with benign selection, no matter which distribution is used: the distribution is by definition independent of which keys are held by uncorrupted tags, so we again have that the target tags hold different keys with probability 1/2. Of course, an error probability of 1/2 is too large in practice.

This makes it natural to ask if we can get privacy with a larger number of corruptions without going back to systems where the reader does exhaustive search over all keys.

4.1 A Necessary Tradeoff

First, it is useful to observe that in the kind of systems we look at here, some tradeoff between efficiency of the reader and privacy is unavoidable: suppose the key generation algorithm works by generating independently a number of keys, and then assigning to each tag a subset of these keys. The system we propose

below, as well as the systems proposed by Molnar, Soppera and Wagner, and by Juels and Weis, are all of this type.

Let K be one of the keys used. We will say that K is *efficiently decidable* if there is an efficient algorithm that, when given K and a session between a tag $\mathcal{T}$ and the reader, can decide whether $\mathcal{T}$ holds K or not. For instance, it may be that the tag, if indeed it holds K, computes a particular part of its response only from K. One can then from K compute what the tag should say if it knows K and compare to what it actually said. In the systems from [9,12], all keys are efficiently decidable.

An efficiently decidable key can be used by the reader towards identifying the tag it is reading, because it can tell whether the tag is in the set of tags that know K or in the complement. However, such a key can also be used by the adversary, who may learn K by corrupting a tag, and can now also distinguish tags that know K from those who do not. Clearly, if the adversary can locate two tags, of which one holds K and the other doesn't, then he can break strong privacy. Let $p(K)$ be the number of tags that hold the key K. The case where $p(K) = n/2$ is the case where the reader gets maximal information from knowing K, namely one bit of information on the identity of the tag. Unfortunately, this is also the optimal case for the adversary, since interactions with a constant number of tags will be sufficient to locate two target tags that can be used to break the privacy.

One may treat this problem either by letting every part of the tag response depend on several keys, or make sure that $p(K)$ is small for every efficiently decidable key K. Both approaches make life harder for the adversary as well as for the reader. We give below an example of the second approach.

4.2 A Tradeoff Construction

Our construction depends on two parameter, v, c. Typically, v will be quite large, say $v = n^d$ for some constant $d < 1$, while c may be something small, say constant or logarithmic in n. We will assume that we have a pseudorandom function $\phi_\cdot(\cdot)$. It is straightforward to construct such functions from a cryptographic hash function by simply hashing the key together with the input, this is provably secure in the random oracle model. Other constructions based on, e.g. AES are also possible.

The key generation involves generating c lists of keys to the pseudorandom function ϕ, $K^j = (k_1^j, k_2^j, ..., k_v^j)$ for $j = 1..c$.

We assign to each tag $\mathcal{T}_i$ a random string $str_i = (s_{i,1}, ..., s_{i,c}) \in Z_v^c$, c keys $(k^1_{s_{i,1}}, ..., k^c_{s_{i,c}})$, and a key k_i that is unique to $\mathcal{T}_i$ (see Figure 2). The probability that two tags will be assigned the same string is at most n^2/v^c, we assume v, c are chosen such that this is negligible. Let n_T, n_R be nonces chosen by tag, respectively reader, such that these values do not repeat. Then the protocol between the tag $\mathcal{T}_i$ and reader is:

1. $\mathcal{R}_i \longrightarrow \mathcal{T}_i$: n_R
2. $\mathcal{R} \longleftarrow \mathcal{T}_i$: n_T, $\phi_{k_{s_{i,j}}}(n_T, n_R)$, for $j = 1, .., c$, and $\phi_{k_i}(n_T, n_R)$. The intuition is that the first c values allow the reader to identify the tag, while the final value proves that the tag is who it claims to be.

$$K^1 = k_1^1, \boxed{k_2^1}, k_3^1, k_4^1, \ldots, k_v^1$$
$$K^2 = \boxed{k_1^2}, k_2^2, k_3^2, k_4^2, \ldots, k_v^2$$
$$K^3 = k_1^3, k_2^3, \boxed{k_3^3}, k_4^3, \ldots, k_v^3$$
$$\ldots$$
$$K^c = k_1^c, \boxed{k_2^c}, k_3^c, k_4^c, \ldots, k_v^c \quad \boxed{k_i}$$

Fig. 2. Example: Keys assigned to a tag $\mathcal{T}_i$ with string $str_i = (2, 1, 3, \ldots, 2)$

For the j'th pseudorandom function value received, $j = 1 \ldots c\}$, the reader searches through the v keys in K^j and checks if one of these will generate the value received, i.e., for each $k \in K^j$ one checks if $\phi_k(n_T, n_R) = \phi_{k_{s_{i,j}}}(n_T, n_R)$. If this is not the case, reject and stop. Otherwise note the index of the key. The indices noted form a string $(s_1, .., s_c)$. If this string matches the string assigned to some tag $\mathcal{T}_i$, and the final pseudorandom value received is equal to $\phi_{k_i}(n_T, n_R)$, output (accept, i). Else output reject.

To show security of the system, we first go to the *independent oracles model*, i.e., we replace each call to ϕ using key k by a call to a random oracle O_k, using independent oracles for different keys. The adversary can only call an oracle O_k if he corrupts a tag that holds k.

It is straightforward to see that if we model the hash function used in the proposed construction of ϕ by a random oracle, then an adversary playing the privacy or soundness game is exactly working in the oracle model just described. For this reason and for simplicity, we will analyze the system in this model. At the cost of a more complicated proof, it is also possible to argue security based only on pseudo-randomness of ϕ, i.e., without using the random oracle model.

The first result on our system shows that, without loss of generality, we may consider only adversaries who do no talk to the reader:

Lemma 2. *In both the privacy and soundness games, sessions that the adversary initiates with the reader can be simulated without access to the reader, but with access to those oracles that the adversary can access. The simulation is perfect, except with probability negligible in k.*

Proof. We describe an algorithm for simulating the sessions in question: In any session, the reader first sends a nonce n_R, this can be simulated by simply following the reader's algorithm for selecting nonces. The message that the adversary returns must consist of a nonce n_T and $c + 1$ values $r_1, ..., r_c, s$. Note that the reader checks these values against oracle outputs generated from the fresh input n_R, n_T, and that we may assume that oracle answers are sufficiently long so they cannot be guessed except with negligible probability. For these reasons, the adversary can only hope to have the reader accept if he generated each of the $c+1$ response values by either using an oracle he has direct access to, or by starting a session with an uncorrupted tag and using (part of) the tag's response.

If this is not the case, we can return *reject* to the adversary: in real life the reader will reject such a response except with negligible probability. But if the adversary has indeed generated the entire response by calling oracles (directly or indirectly), we know the identity k' of the oracle the generated the last value in the response. If the call to oracle $O_{k'}$ was made by an uncorrupted tag $\mathcal{T}_j$, this has to be because that tag received n_R as a challenge and therefore produced a correct response for nonces n_R, n_T. If we see that the adversary forwards this response to the reader, we return $(accept, j)$ as the real reader would have done. If the adversary has replaced any of the first c values with other oracle responses, we return *reject*, which is correct except with negligible probability.

The only remaining possibility is that it was the adversary who called $O_{k'}$. This means he must have corrupted the tag $\mathcal{T}_i$ giving access to this oracle, and so he also has access to to the other c oracles that this tag possesses. Therefore, having generated the message sent to the reader, we can check whether this is a correct response from $\mathcal{T}_i$. If this is not the case, we return reject to the adversary. Otherwise, we return $(accept, i)$. □

The following lemma turns out to be essential for privacy:

Lemma 3. *Consider an adversary that does not start any session with the reader. Let M be the set of oracles that the adversary gains access to during the privacy game. Let E be the event that the following condition is satisfied after the game: the adversary has started at least one session with some uncorrupted tag $\mathcal{T}$, and one of the oracles assigned to $\mathcal{T}$ is in M. In the privacy game, by convention, the adversary selecting the two target tags counts as starting a session with both tags. Let t' be the number of different tags the adversary talks to during the game. The probability that E occurs is at most*

$$\frac{ct'u}{v} + \frac{ct'u}{v-u}$$

Proof. Suppose we are at some point in the game where E has not occurred yet. This means that for all uncorrupted tags the adversary has talked to, he knows that they only have oracles he has no access to, but due to the randomness of the oracles, he has no information on their identity.

The adversary may now start a session with a new tag he did not talk to before, or corrupt a tag. For each of these moves, we bound the probability that E will occur after the move:

Start new session: Since the adversary has not previously talked to the tag $\mathcal{T}_i$, given what he knows, str_i is uniform. We can therefore model what goes on as follows: look at one of the c positions in str_i, and let $x \in Z_v$ be the number in this position. Now, x is uniform over v possibilities, and the adversary has success, if x happens to be one of the $\leq u$ values corresponding to oracles he can access. So the adversary has success in one position with probability at most u/v, and therefore has success in any position with probability at most $\frac{cu}{v}$

Corrupt new tag: For the previously uncorrupted tag $\mathcal{T}_i$, consider again x, the number at some position in str_i. Then given what the adversary knows, before he corrupts $\mathcal{T}_i$, x is uniform over at least $v - u$ possibilities, if the adversary talked to $\mathcal{T}_i$ before, he knows x does not match any of the $\leq u$ possibilities he knows from already corrupted tags. The adversary hopes x will hit one of the $\leq t'$ possibilities for tags he talked to, so the probability of success is at most $t'/(v-u)$ for one position and $\frac{ct'}{v-u}$ for all positions.

Finally, since there are at most t' respectively u steps that could cause the first respectively second kind of event, the lemma follows. □

We are now ready to prove security of our construction.

Theorem 2. *For the RFID system described above, we have that if the hash function used in the construction is modeled by a random oracle, then the system is* $(poly(k), poly(k), poly(k), n)$*-strongly sound, and is strongly* $(k, r, poly(k), t, u, n, \epsilon)$*-private, where*

$$\epsilon = \frac{ct'u}{v} + \frac{ct'u}{v-u} + negl(k)$$

and where $negl(k)$ *is a negligible function of* k*.*

Proof. Completeness is obvious from the fact that the strings assigned to tags are unique except with negligible probability.

For soundness, recall that the adversary wins the soundness game if a session is generated where the reader outputs $(accept, i)$, but the (uncorrupted) tag $\mathcal{T}_i$ did not participate. Since the input nonces are fresh and oracles answers cannot be guessed in advance except with negligible probability, the oracle O_{k_i} must have been called to generated the last part of the response. But this is impossible since $\mathcal{T}_i$ did not participate and the adversary does not have access to O_{k_i} as long as $\mathcal{T}_i$ is uncorrupted.

Finally, for privacy, note that by Lemma 2, any adversary $\mathcal{A}$ playing the privacy game can be replaced by a new adversary $\mathcal{A}'$, who does not start sessions with the reader, and such that the advantage of $\mathcal{A}'$ is smaller than that of $\mathcal{A}$ by at most a negligible amount. This, together with Lemma 3 immediately implies the privacy result. □

Finally, we show that the adversary's advantage in the benign selection privacy game is much smaller:

Theorem 3. *Our system is* $(k, r, poly(k), t, u, n, \epsilon)$*-private with benign* D*-selection for any* D*, and where* $\epsilon = 2cu/v + negl(k)$

Proof. As above, we can assume that the adversary does not talk to the reader, at the cost of adding a negligible amount to the advantage. Now consider the situation when the target tags are chosen. For each of the c positions in the strings assigned to tags, the adversary can access at most u of the v oracles assigned to this position. Hence, when an uncorrupted tag is chosen, no matter how this is

done, the probability that its oracle for this position is known to the adversary is u/v since "names" of tags are assigned uniformly and independently. Since the two target tags hold a total of $2c$ oracles that could be used to distinguish them, the probability that at least one of them is known to the adversary is at most $2cu/v$. On the other hand, if the adversary has no oracles in common with the target tags, he cannot distinguish them at all. □

5 Efficiency

The interest in this result is that it shows a possibility for a new tradeoff between security and efficiency for large systems, where the adversary can be expected to only corrupt or talk to a number of the tags that is very small compared to the total number of tags in the system. More precisely, for parameter values such that $r, t', u << v << n$, but still $n^2 < v^c$. However, for particular values of r, t', u and c, v and hence n must very large to make the privacy advantage be small. This has to do with the fact that we are asking for strong privacy and this is a very strong demand. Below, we show that the systems performs much better under the privacy definition with benign selection. On the practical side, note that the reader needs to look at only cv keys which can be much smaller than n. Also, each tag only has to hold $c + 1$ keys. Although the total number of keys in the system is greater than n, this does not mean that the reader has to store this many keys – they can be generated pseudorandomly from a single key when they are needed.

Let us look at a concrete example of parameters in the benign selection model for any distribution. Suppose we choose $v = 2^{16}$ and $c = 4$. Then we can accommodate over 33 million tags, say $n = 2^{25}$, and each tag only needs to store 5 keys. If the adversary can corrupt 100 tags, the above says that his chance of distinguishing two tags that are chosen for him is at most 1/100. Note that even if the adversary is lucky with one pair of tags, his chance against another pair is still only 1/100, so we think this can be quite reasonable in practice. In other words, even though a probability of 1/100 is not negligible in the usual sense, this is not necessary, if the "bad event" does not imply a complete break of the system. With these parameters, the reader must search through at most 2^{18} keys to identify a tag, which is clearly better than 2^{25}, which was needed to get strong privacy. We can even increase n without increasing the number of keys to search through, as long as we keep the probability that two tags will be assigned the same key n^2/v^c reasonably small.

6 Conclusion

We have proposed a new definition of security for RFID systems, incorporating both strong privacy, soundness and completeness, and also a weaker but more realistic variant of privacy, with benign selection. We have shown that in sound, complete and essentially symmetric RFID system where tags are independently keyed, the reader must access at least half of all keys when reading a tag, or

privacy is violated. Finally, we have proposed a new RFID system based on symmetric cryptography offering a tradeoff between reader efficiency and privacy.

References

1. Ateniese, G., Camenisch, J., Joye, M., Tsudik, G.: A practical and provably secure coalition-resistant group signature scheme. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 255–270. Springer, Heidelberg (2000)
2. Avoine, G.: Adversarial model for radio frequency identification. Cryptology ePrint Archive, Report 2005/049 (2005)
3. 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)
4. Avoine, G., Oechslin, P.: A scalable and provably secure hash-based rfid protocol. In: Stajano, F., Thomas, R. (eds.) PerSec 2005, vol. 00, pp. 110–114. IEEE Computer Society Press, Los Alamitos (2005)
5. Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., Verbauwhede, I.: An elliptic curve processor suitable for RFID-tags. Cryptology ePrint Archive, Report 2006/227 (2006)
6. Burmester, M., van Le, T., de Medeiros, B.: Provably secure ubiquitous systems: Universally composable rfid authentication protocols. Cryptology ePrint Archive, Report 2006/131 (2006)
7. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
8. Hellman, M.E.: A cryptanalytic time-memory tradeoff. IEEE Transactions on Information Theory 26(6), 401–406 (1980)
9. Juels, A., Weis, S.A.: Defining strong privacy for rfid. In: PERCOMW 2007, vol. 1462, pp. 342–347. IEEE Computer Society Press, Los Alamitos (2007)
10. Kiayias, A., Yung, M.: Group signatures with efficient concurrent join. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 198–214. Springer, Heidelberg (2005)
11. Kilian, J., Petrank, E.: Identity escrow. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 169–185. Springer, Heidelberg (1998)
12. Molnar, D., Soppera, A., Wagner, D.: A scalable, delegatable pseudonym protocol enabling ownership transfer of rfid tags. Cryptology ePrint Archive, Report 2005/315 (2005)
13. Ohkubo, M., Suzuki, K., Kinoshita, S.: Efficient hash-chain based rfid privacy protection scheme. In: Ubicomp, Privacy Workshop: Current Status and Future Directions (2004)
14. de Medeiros, B., van Le, T., Burmester, M.: Universally composable and forward secure rfid authentication and key exchange. Cryptology ePrint Archive, Report 2006/448 (2006)
15. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and privacy aspects of low-cost radio frequency identification systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 201–212. Springer, Heidelberg (2004)