

# RFID Security: Tradeoffs between Security and Efficiency

Ivan Damgård and Michael Østergaard Pedersen

Dept. of Computer Science, University of Aarhus

**Abstract.** Recently, Juels and Weis defined strong privacy for RFID tags. We add to this definition a completeness and a soundness requirement, i.e., a reader should accept valid tags and only such tags. 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 et al. 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, strong privacy is lost if an adversary corrupts only 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 security, as long as we assume suitable limitations on the adversary.

## 1 Introduction

RFID tags are small wireless devices that react to electromagnetic fields; they can emit some prestored information and can also in some cases do a limited amount of computation. 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 humans 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 [3], Juels and Weis propose a definition of what they call “strong privacy” (based on earlier work by Avoine[1]). Strong privacy is indeed a strong notion, primarily because the adversary is given a lot of power: he can corrupt any number of tags and read their contents, and he can eavesdrop and schedule the tag/reader communication any way he wants. In independent work, Burmester et al.[2] propose a security definition based on Canetti’s Universal Composability framework.

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. This would trivially be private, but would of course lead to a useless system.

In this paper, we propose a simple extension to the strong privacy definition so one can require also completeness and soundness, with the intuitive meaning that the reader accepts valid tags and only valid tags. 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.

Juels and Weis suggest a system that satisfies their definition. 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, the reader must access all, or at least a large fraction of the keys in the system. Here, we prove this conjecture, for the case where tags have independent keys. We need to assume that the system is complete and sound, but this is a very natural requirement and is necessary anyway to exclude "pathological" cases, such as when tags send only random information.

The limitation to schemes with independent keys is not surprising. It follows from work by Molnar et al[4] 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. This does not mean that all privacy is lost in practice in the system of Molnar et al. – the definition of strong privacy gives a lot of power to the adversary that he may not have in practice. But still, it is natural to ask if we can get strong 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 security, as long as we assume suitable limitations on the adversary.

## 2 Model and Definition

Juels and Weis[3] define *strong privacy* for RFID systems using the following model: the system consists of tags  $\mathcal{T}_i, i = 1..n$  and a reader  $\mathcal{R}$ , we assume there is only one, for simplicity.

Tags can receive SETKEY messages which will cause the tag to reveal its secret key, 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 learn 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 [3], 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 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 [3], 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 id of a tag. The reader keeps an internal log of all challenges and responses 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,  $t$  is the number of TAGINIT messages it generates. Finally,  $k$  is a cryptographic security parameter. Juels and Weis do not treat the number of TAGINIT messages, i.e., the number of corrupted tags, as a separate parameter, but simply say it is 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.

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

Setting  $\mathcal{S} = Gen, \mathcal{R}, \{\mathcal{T}_i\}$ , strong privacy is defined via an experiment called  $\mathbf{Exp}_{\mathcal{A}, \mathcal{S}}^{priv}[k, n, r, s, t]$ . Here, we run the system where the adversary may corrupt tags, initiate sessions, etc., 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 the adversary starts a session with the reader, 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 the 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.

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 is polynomial time in  $k$ .

It is natural to expect a system as described here to also have the properties that valid tags are accepted, and the adversary cannot impersonate a tag unless he corrupts it. This aspect was not treated in [3] (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 state of the reader 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 the reader outputs Accept with probability 1 for any such session.

**Strong Soundness** Consider the following experiment similar to the privacy experiment of Juels and Weis. We start with the same setup phase, and then the experiment is as follows:

$\text{Exp}_{A,S}^{strongsound}[k,n,r,s,t,u]$ :

The adversary may do the following in any order:

1. Make READERINIT calls, without exceeding  $r$  overall calls.
2. Make TAGINIT calls, without exceeding  $t$  overall calls.
3. Make SETKEY calls without exceeding  $u$  calls. Tags that receive a *SetKey* call, are marked as corrupted.
4. Communicate and compute, without exceeding  $s$  overall steps

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

**Weak Soundness** Weak  $(r, s, t)$ -soundness is defined by the same experiment as above, except that the reader now only has to output Accept or Reject at the end of a session, the adversary is not allowed to corrupt tags, and the error event  $E_{fail}$  is now defined to be that the reader 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 basically first encrypt its identity under the reader's public key, and then show possession of the corresponding secret.

However, 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 crypto and nothing else is used”, we will give the system access to a pseudorandom function,  $\psi(\cdot)$ , 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  $\psi_{K_i}(\cdot)$  as a black box. This means that we can equivalently give tags and reader oracle access to  $\psi_{K_i}(\cdot)$  for any key they need to use.

Now, to model that the pseudorandom function is the *only* cryptographic resource used, we will use a standard technique, namely replace the oracle access to the resource by access to an ideal information theoretically secure resource, and require that the resulting system is secure against an unlimited adversary. The point here is that if the system uses some other form of cryptography, this can now be broken, so in this way we force the system to rely only on the target resource.

Concretely, in our case, we replace the calls to the pseudorandom function  $\psi_{K_i}(\cdot)$  by calls to a random oracle  $R_{K_i}$ , so we have a set of independent oracles indexed by the keys in the system. These oracles can be accessed in the same way as the keys in real life, i.e.,  $R_{K_i}$  can be accessed by  $\mathcal{T}_i$  and the reader, but not by the adversary, unless he corrupts  $\mathcal{T}_i$ .

The requirement now is that when using these oracles, the system remains complete, sound and strongly private against an unlimited adversary. For short, we say that the system is *secure in the independent oracle model*. Note that the hash-lock systems that were shown to be strongly private in [3] are indeed secure in the independent oracles model.

The first lemma summarizes the rather obvious 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  $(1, \text{poly}(k), 0)$ -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  $R_{K_i}$ . 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.

Of course, if we replace the pseudorandom functions by oracles, i.e., we go to the independent oracle model, the lemma is still true, if not, this would contradict pseudorandomness of the functions.

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

**Theorem 1.** *Assume an RFID system is complete and weakly  $(1, \text{poly}(k), 0)$ -sound. Such a system cannot have strong  $(0, \text{poly}(k), 1, n - 2)$ -privacy in the independent oracle model, if the reader algorithm accesses at most  $\alpha n$  of the oracles, for a constant  $\alpha < 1/2$ .*

*Proof.* We describe an adversary that will break strong privacy for any system that is complete and weakly sound and where only  $\alpha n$  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 an oracle  $R_{K_t}$ , the adversary corrupts  $\mathcal{T}_t$ , this gives access to key  $K_t$  and therefore in our model access to  $R_{K_t}$ . This goes on until the reader algorithm wants to access  $R_{K_t}$  where  $t = i$  or  $j$ . In this case the adversary outputs 0 if  $t = i$  and 1 otherwise and then stops.

To analyse 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  $R_{K_i}$  when talking to  $\mathcal{T}_b^*$  with probability essentially 1. On the other hand, the probability that it will not access  $R_{K_j}$  is greater than  $1 - \alpha$  because only  $\alpha n$  oracles are accessed (one of which is  $R_{K_i}$ ), 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.

## 4 Correlated Keys

We have shown in the previous section that if we want strong privacy, the reader has to access least half of the keys in the worst case. This does obviously not scale very well, so we investigate how much privacy and soundness we will lose if we allow the keys to be correlated.

It was already known from the work of Molnar et al.[4] that using correlated keys, one can obtain the property that the reader only needs to access a logarithmic number of keys. 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  $K_0$ , it is not hard to find two tags holding different keys, and clearly using two such tags as the target in the privacy experiment, the adversary can identify with certainty which tag he talks to.

Of course, this attack is based on the adversary's ability to choose himself which tags he wants to be challenged on. This is part of the model, but on the other hand the adversary may not be in such a strong position in real life, so the above does not mean that all privacy is lost in practice in the system of Molnar et al. But still, it is natural to ask if we can get strong privacy with a larger number of corruptions without going back to systems where the reader does exhaustive search over all keys.

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 systems we propose below, as well as the systems proposed by Molnar et al., 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 [3, 4], 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 a constant number of interactions with 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 here an example of the second approach.

Our construction depends on two parameter,  $v, c$ . Typically,  $v$  will be quite large, say  $v = \lceil \sqrt{n} \rceil$  for a large system, while  $c$  may be something small, say constant or logarithmic in  $n$ . We will assume that we have a pseudorandom function  $\phi(\cdot)$ . 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 be possible.

The key generation involves generating  $c$  lists of keys to the pseudorandom function  $\phi$ ,

$$K^j = (k_1^j, k_2^j, \dots, k_v^j)$$

for  $j = 1..c$ .

We assign to each tag  $\mathcal{T}_i$  a random string  $str_i = (s_{i,1}, \dots, s_{i,c}) \in Z_v^c$ ,  $c$  keys  $(k_{s_{i,1}}^1, \dots, k_{s_{i,c}}^c)$ , and a key  $k_i$  that is unique to  $\mathcal{T}_i$ . 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. Then the protocol between the tag  $\mathcal{T}_i$  and reader is:

1.  $\mathcal{T}_i \leftarrow \mathcal{R}: n_R$
2.  $\mathcal{R}_i \leftarrow \mathcal{T}: n_T, \phi_{k_{s_{i,j}}} (n_T, n_R)$ , for  $j = 1, \dots, c$ , 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 the identity of the tag.

For the  $j$ 'th pseudorandom function value received,  $j = 1..c$ , the reader searches through the  $t$  keys in  $K^j$  and checks if one of these will generate the value received. If this is not the case, reject and stop. Otherwise note the index of the key. The indices noted form a string  $(s_1, \dots, s_c)$ . 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  $\phi$  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 hashfunction used in the proposed construction by a random oracle, then an adversary playing the privacy or soundness game is exactly working in the oracle model just described. We will therefore analyze the system in this model.

**Lemma 2.** *In both the privacy and soundness games, the adversary's access to initiate sessions with the reader can be simulated without access to the reader. The simulation is perfect, except with probability negligible in  $k$ .*

*Proof.* In any session, the reader defines a nonce  $n_R$ , the message that is returned must consist of a nonce  $n_T$  and  $c + 1$  values  $r_1, \dots, r_c, s$ . Since the reader checks these values against oracle outputs generated from the fresh input  $n_R, n_T$ , each of the  $c + 1$  values must have been generated by calling one of the oracles in the system on this input. If not, the adversary already knows the reader will

reject except with negligible probability – we assume that oracle answers are sufficiently long so they cannot be guessed except with negligible probability. We can therefore assign oracle identities to the  $c + 1$  values according to which oracle generated the value. Let these be  $k_1, \dots, k_c, k'$ . If the call to oracle  $O_{k'}$  was made by an uncorrupted tag, this has to be because that tag received  $n_R$  as a challenge and produced a correct answer for nonces  $n_R, n_T$ . If the adversary forwards this to the reader, it will accept. If he has replaced any of the first  $c$  values with other oracle responses, the reader will reject, 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 the other  $c$  oracles that this tag possesses. Therefore, having generated the message sent to the reader, he can check whether this is a correct response from  $\mathcal{T}_i$ . If this is not the case, he knows the reader will reject. Otherwise, the reader will output (accept,  $i$ ).

This lemma essentially already implies soundness. This is because a session where the reader outputs (accept,  $i$ ), but tag  $\mathcal{T}_i$  did not participate, must be a session where the adversary called the oracle generating the last part of the tag-message. But the argument in the Lemma says that, except with negligible probability, this can only happen if the tag was corrupted – in which case this does not allow the adversary to win the soundness game.

The following lemma turns out to be essential:

**Lemma 3.** *Let  $M$  be the set of oracles that the adversary gains access to during the privacy or soundness 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. The probability that  $E$  occurs is at most*

$$\frac{ctu}{v} + \frac{ctu}{v - u}$$

*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  $x \in Z_t$  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.

It is straightforward to see that if the event  $E$  does not occur in either the privacy or the soundness experiment, then the adversary cannot win. For soundness this follows from the remarks after Lemma 2. For privacy, it is clear that if  $E$  did not occur, the adversary has no information on which keys are held by the target tags and hence he cannot distinguish between them at all. In summary, we have

**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  $(r, poly(k), t, u)$ -private, if parameters are chosen such that*

$$\frac{ctu}{v} + \frac{ctu}{v-u} + \text{negl}(k)$$

*is negligible, where  $\text{negl}(k)$  is a negligible function of  $k$  (which enters because of the reduction in Lemma 2).*

The interest in this result is that it shows a possibility for new tradeoffs between security and efficiency for very 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. This means we can choose parameters such that  $r, t, u \ll v \ll n$ , but still  $n \ll v^c$ . This will make the probability in the theorem be small, and yet the total number of keys in the system is  $cv$  which is can be much smaller than  $n$ , and each tag only has to hold  $c$  keys.

We emphasize that this is only a preliminary result, and numeric examples do not give very favorable results. However, the analysis is rather pessimistic and not very precise.

One characteristic thing about this result, however, is that the reader does work proportional to  $v$ , whereas the adversary's probability of breaking privacy is proportional to  $1/v$ . If we want the reader to do work that is not proportional to  $n$  but is much smaller, we believe a tradeoff of this type cannot be avoided without going to a completely different type of system using public-key, for instance – although the precise result can probably be improved.

Note that although the total number of keys in the system is greater than  $n$ , this does not mean the reader has to store this many keys – they can be generated pseudorandomly from a single key when they are needed.

## References

1. Avoine: *Adversarial Model for Radio Frequency identification*, the Eprint archive, [www.iacr.org](http://www.iacr.org).
2. Burmester, van Le and de Medeiros: *Provable Ubiquitous Systems: Universally Composable RFID Authentication protocols*, the Eprint archive, [www.iacr.org](http://www.iacr.org).
3. Juels and Weis: *Defining Strong Privacy for RFID*, the Eprint archive, [www.iacr.org](http://www.iacr.org).
4. Molnar, Soppera and Wagner: *A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID tags*, the Eprint archive, [www.iacr.org](http://www.iacr.org).