Ad hoc subgroup proofs for RFID

Mike Burmester and Daniel Miller

Department of Computer Science

Florida State University, Tallahassee, Florida 32306-4530, U.S.A.

Abstract. In many RFID applications it is necessary to be able to identify the presence of (all) the tags of a predefined group, e.g., to establish that all components of a kit are present. If the interrogation is offline, by RFID readers that do not share any secret keys with the tags, then a record of simultaneous presence, or a *grouping proof*, has to be generated by the tags themselves. This type of proof is typically obtained by assigning to each tag in the group a unique group identifier. In certain situations, however, the entire group may not be present, and it may be sufficient that a (an ad hoc) subgroup proof is generated.

In this paper we consider *ad hoc subgroup proofs*. We first describe a generic ad hoc subgroup proof by extending a previously proposed grouping proof to an ad hoc setting. We then propose a novel ad hoc subgroup proof for RFID that supports privacy (unlinkability).

1 Introduction

Radio-frequency identification (RFID) is a field that continues to grow in popularity due to its many applications, in particular warehouse inventory, supply chain management, highway toll payments, etc. Several RFID protocols that address security issues have been proposed in the literature–we refer the reader to a comprehensive repository available online at the RFID Security and Privacy Lounge [1]. Ari Juels introduced in 2004 the security context of RFID readers generating evidence of simultaneous presence of tags which he called a *yoking-proof* [7], and noted that interesting security engineering challenges arise when the trusted server (or Verifier) is not online during the scan activity. Yoking proofs uses message authentication codes (MAC) to generate a proof. Each tag stores a secret key (shared with the Verifier) and a counter. Saito and Sakurai [13] analyzed a minimalist version (not containing a counter) of Juels' proof and found that it was vulnerable to replay attacks and DoS attacks, and proposed a variant that addresses these vulnerabilities by using timestamps [13]. Piramuthu [12] showed that this variant was vulnerable to replay attacks and proposed an improved protocol in which the values generated by the verifier and sent to the tags were randomized. Lin *et al.* [10] pointed out that the Piramuthu protocol suffers from an interference problem when multiple readers are within range of each other. Duc-Kim observed in [6] that all grouping proofs are vulnerable to a man-in-the-middle attack because a malicious reader can relay messages from a reader to tags outside of the reader's range. Burmester *et al.*[4] proposed three protocols that only require a pseudo-random number generator: each protocol builds on the previous one with the last one providing anonymity and forward secrecy. Duc-Kim [6] observed that all grouping proof protocols proposed so far use the reader to relay data between the tags and thus suffer from scalability. Burmester *et al.* [3] analyze recent protocols that claim to support the EPC Gen2 standard and point out their vulnerabilities. They also propose two protocols that are within the constraints of this standard.

2 RFID deployments and ad hoc subgroup proofs

A typical deployment of an RFID system involves three types of legitimate entities: a Verifier (back-end server) RFID readers (interrogators) and RFID tags. The tags are attached to, or embedded in, objects to be identified. They consist of a transponder and an RF coupling element. The coupling element has an antenna coil to capture RF power, clock pulses and data from the RFID reader. The readers typically contain a transceiver, a control unit, and a coupling element, to interrogate tags. They implement a radio interface to the tags and also a high level interface to a back-end server that processes captured data.

The Verifier is a trusted entities that maintain a database containing information needed to identify tags. Since the integrity of an RFID system is entirely dependent on the proper behavior of the Verifier, it is assumed that the Verifier is physically secure and not attackable. On the other hand readers are not trusted, and may be compromised Consequently any secret keys that the tags share with readers must not make it possible to impersonate tags.

Grouping proofs. An RFID *grouping proof* is digital evidence corroborating the simultaneous scanning of a group of tags by an RFID reader. A grouping proof with group identifier ID_{group} is *valid* if: (i) all of the tags of ID_{group} are present during a single scanning by an RFID reader, and (ii) no tag of ID_{group} is compromised. The second requirement is needed because even if one tag is compromised, the adversary can use a proxy of that tag to impersonate it. The resulting proof will be a forged proof, since the real tag_1 was not scanned.

Ad hoc subgroup proofs. A *subgroup proof* is digital evidence corroborating the simultaneous scanning (or presence) of a subgroup of tags by an RFID reader. Validity for such proofs requires only that no tag of the scanned subgroup is compromised. There are important differences between the grouping and subgroup proofs primitives related to: (a) the labeling of ad hoc subgroups (scalability, when the group is large), and (b) having the tags of an ad hoc subgroup establish a subgroup pseudonym, when privacy is required.

3 Adversarial model and design requirements

We adopt the Byzantine threat model. All entities: the verifier, the readers, the tags, including the adversary (the attackers) have polynomially bounded resources. The adversary controls the delivery schedule of all communication channels, and may eavesdrop into, or modify, their contents. The adversary may also instantiate new communication channels and directly interact with honest (non-compromised) parties. However, since the reader-server channels are assumed secure, we do not need model adversarial interactions with reader-server channels.

Threats and attacks. The goal of the adversary is to forge ad hoc subgroup proofs, and more generally, to undermine the procedure of generating a proof. There are several general types of attack on RFID systems that have been discussed in the literature. Below we identify the most important one.

1. *Availability/Denial of Service*: the adversary causes a tag to be unable to respond correctly to an RFID reader.
2. *Privacy*: The adversary traces subgroups of tags from protocol flows.
3. *Integrity.*
   - *Replay/Forgery*
   - *Tag cloning*
   - *Interleaving and Reflection attacks*
   - *Offline man-in-the-middle attacks*

There are also attacks that are usually excluded from the security model, such as:

- *Online man-in-the-middle relay attacks* [2, 8]
- *Side Channel and Power Analysis* [11] attacks

These attacks are usually prevented by using "out of system" protection mechanism.

Design Requirements. RFID systems must be efficient and lightweight. This is because passive tags do not have battery-power and rely on scavenging power from RFID readers. Furthermore the overhead of the RFID system should be minimal. In particular, when the system is under attack there should not be additional cost incurred by the tags: additional costs, if any, should be borne by the Verifier and the RFID readers. In many applications, privacy (anonymity/unlinkability) is an important requirement: for example in applications where tagged objects (or subjects) should not be traceable. Finally, RFID protocols should provide modularity and reusability. Due to the vast number of applications, RFID security protocols must be able to run parallel with other protocols without interfering. Often security protocols are analyzed with the assumption that they are operating alone. With such protocols there is no security assurance when resource are shared with other protocols.

4 Offline vs online interrogation for grouping proofs

An ad hoc subgroup proof can have online access to the Verifier or offline access. Online access proofs can easily be implemented. It is sufficient that each tag in the group gets authenticated: the set of authenticators of the tags in the subgroup then constitutes a subgroup proof. This argument does not work well for offline proofs because the adversary may inject a large number of bogus authenticators that can only be identified later by the Verifier [7]. This is essentially a DoS attack. An offline ad hoc subgroup proof must provide some assurance that bogus proofs are rejected by RFID readers, who do not share any secret keys with the tags. In particular, a subgroup proof can only be generated by the tags of the subgroup that is scanned by the (authorized) reader—without any interaction with the Verifier. The role of the reader is to enable this by linking the tags in the subgroup, and forwarding their messages. This approach enables greater portability in situations where the reader does not have a direct connection to the Verifier, and broadens the applicability of subgroup proofs.

5 An ad hoc subgroup proof

We present two ad hoc subgroup proofs for RFID. The first one does not provide anonymity: the messages the tags send to the RFID reader include a group identifier ID_{group}. In the second proof no identifier is passed to the reader: the proof uses a pseudonym value that depends on a secret *group key* K_{group}, but the dependency is known only to the tags in the group. Thus, only tags of the group are able to check group membership: this guarantees unlinkability and anonymity. We only consider situations when the Verifier is offline while the tags are being scanned, as this is the most challenging case, and when the Verifier communicates with the RFID readers through authenticated channels.

Irrespective of the number of tags involved, a specific tag in the subgroup always plays the role of "initiator". The task of the initiator is to check that all singulated tags of the subgroup are accounted for, and to generate a confirmation. Each tag of the subgroup stores in non-volatile memory the (shared) group key K_{group} used to establish membership in a subgroup as well as an *identification key* k_{tag} used to authenticate protocol flows. Tags instances are denoted as tag_i, $i = 1, 2, \ldots$, and the key for instance tag_i is written in shorthand as k_i. The Verifier stores for each group the values $(ID_{group}, \{k_i\})$ in a database D.

The protocol starts with an RFID reader broadcasting a random challenge r_{sys} which is obtained from the trusted Verifier at regular intervals. The challenge defines the scanning period, i.e., each group should generate at most one proof between consecutive challenge values. In other words, the Verifier cannot (without further assumptions) determine simultaneity of a group scan to a finer time interval than the scanning period. Each RFID reader receives a set (or one at a time, if online) of specific r_{sys} values from the Verifier. Thus it is not possible for one reader to send to the Verifier a proof that was collected by another reader, because the Verifier would detect an incorrect value of r_{sys} for that specific reader. Furthermore any proof generated by an unauthorized reader will be invalid because it is not linked to an authorized r_{sys}.

The protocol

Let $tag_1, \ldots, tag_n$ be the tags of a group with identifier ID_{group}, and $tag_{i_1}, \ldots, tag_{i_k}$, $i_1 < \cdots < i_k$, be the subgroup of tags singulated (scanned) by an (authorized) reader. Arrange the tags of the subgroup in a logical ring with indices taken $\mathrm{mod} k$, so that $tag_{i_{k+1}} = tag_{i_1}$. In this subgroup tag_{i_1} (with the smallest index) will act as an "initiator".

In the protocol each tag_{i_j} computes two authenticators: $aut_{i_{j-1}}$ and aut_{i_j}. The first is used to authenticate the preceding $tag_{i_{j-1}}$ in the ring, while the second is sent as an authenticator to its successor $tag_{i_{j+1}}$. The authenticators are obtained by evaluating $f(K; r_{sys}||sn_{i_1}||i_t)$, $t = j-1,\ j$, and used by the subgroup tags to make certain that all singulated tags of the subgroup are accounted for. The state of the interrogated subgroup is determined by a session number sn_{i_1} of the initiator tag (the tag with the smallest index in the subgroup). This number is initialized with a random value and updated with each execution of the protocol.

The protocol has three phases—see Figure 1. In the first phase the RFID reader challenges all tags in its range with the random number r_{sys} and each tag_{i_j} responds

```
Parties:   Reader($r_{sys}$);   Subgroup $\{\, tag_{i_j}(ID_{group}, K, k_{i_j}, sn_{i_j}),\ \ j = 1, \ldots, k \,\}$

Phase 1
    1. Reader $\rightarrow * :\ r_{sys}$   (a random number)
    2. For each $1 \leq j \leq k\ :\ tag_{i_j} \ \rightarrow * :\ ID_{group},\ i_j$
Phase 2
    1. The Reader selects one subgroup, and links the tags of that subgroup.
    2. For each $1 \leq j \leq k$
            Reader $\rightarrow tag_{i_j}$ :   the subgroup $i_1, \ldots, i_k$ of $ID_{group}$, is linked
Phase 3 (sequential execution)
    1. $tag_{i_1}$ : Set timer; compute $aut_{i_1} = f(K; r_{sys}||sn_{i_1}||i_1||i_2)$; set $c \leftarrow sn_{i_1}$,
            update $sn_{i_1} \leftarrow aut_{i_1}$
                $tag_{i_1} \ \rightarrow$ Reader $\rightarrow\ tag_{i_2}$ :   $(i_1, c, aut_{i_1})$

    2. For $1 < j \leq k$
       $tag_{i_j}$ : Compute $aut_{i_j} = f(K; r_{sys}||c||i_j||i_{j+1})$
            If $(i_{j-1}, X_{j-1})$ is received and $X_{j-1} = aut_{i_{j-1}}$ then
                $tag_{i_j} \ \rightarrow$ Reader $\rightarrow\ tag_{i_{j+1}}$ :   $(i_j, c, aut_{i_j})$ ;   timeout
            else timeout

    3. $tag_{i_1}$ : Compute $aut_{i_k} = f(K; r_{sys}||c||i_k||i_{k+1})$ and $cnf_{i_1} = f(k_{i_1}; r_{sys}||c)$
            If $(i_k, X_k)$ is received and $aut_{i_k} = X_k$ then
                $tag_{i_1} \ \rightarrow$ Reader :   $(i_1, cnf_{i_1})$ ;   timeout
            else timeout

    4. Reader : If a confirmation from the $tag_{i_1}$ is received then generate:

        $P_{sgp} = (r_{sys}, ID_{group}, c, i_1, \ldots, i_k, cnf_{i_1}, aut_{i_2}, \ldots, aut_{i_k})$
```

Fig. 1. An ad hoc ring subgroup proof.

with its group identifier $ID_{group},\ i_j$. In the second phase—which takes place at the data-link layer—the subgroup of tags of ID_{group} in range of the reader gets linked by channels through the reader. The third phase has two parts: first tag_{i_1} challenges tag_{i_2} with (i_1, c, aut_{i_1}) that contains the authenticator aut_{i_1}—obtained by evaluating $f(K; r_{sys}||c||i_1||i_2)$, where f is a pseudorandom function. Then each tag_{i_j} in the ring checks that the preceding tag is accounted for by verifying its authenticator $aut_{i_{j-1}}$ and if so, sends to the next tag in the ring its authenticator aut_{i_j}. Finally the initiator tag checks the authenticator aut_{i_k} and if this is valid sends to the reader the subgroup confirmation cnf_{i_1}—obtained by evaluating $f(k_{i_1}; r_{sys}||c)$. Then the RFID reader generates the proof $P_{sgp} = (r_{sys}, ID_{group}, c, i_1, \ldots, i_k, cnf_{i_1}, aut_{i_2}, \ldots, aut_{i_k})$.

Each phase can be executed concurrently by the tags in the subgroup—this includes all computations: *e.g.*, tag_{i_2} can evaluate $f(K; r_{sys}||c)$ immediately after being linked, except the third phase in which each tag checks the authenticator of the preceding tag in the ring. The various phases cannot be consolidated without loss of some security feature, or worse, of determinate outcome. If we remove the first phase (r_{sys}) the protocol would be subject to a full-replay attack (Section 3). If we remove the second phase (the

Parties: READER(r_{sys}); SUBGROUP $\{ tag_{i_j}(ID_{group}, K, k_{i_j}, sn_{i_j}),\ \ j = 1, \dots, k \}$

Phase 1 and Phase 2 are as in Figure 1

```
Phase 3 (concurrent execution)
1. $tag_{i_1}$ : Set timer; compute $aut_{i_j} = f(K; r_{sys}||sn_{i_1}||i_1||i_2)$; set $c \leftarrow sn_{i_1}$,
       update $sn_{i_1} \leftarrow aut_{i_1}$
         $tag_{i_1} \rightarrow$ READER $\rightarrow * :$   $(i_1, c, aut_{i_1})$
   For each $1 < j \leq k$
   $tag_{i_j}$ : Set timer; compute $aut_{i_j} = f(K; r_{sys}||c||i_j||i_{j+1})$
         $tag_{i_j} \rightarrow$ READER $\rightarrow * :$   $(i_j, aut_{i_j})$
2. $tag_{i_1}$ : Compute $cnf_{i_1} = f(k_{i_1}; r_{sys}||c)$
       If $(i_k, X_k)$ is received and $X_k = aut_{i_k}$ then $tag_{i_1} \rightarrow$ READER: $(i_1, cnf_{i_1})$
         and timeout
       else timeout
   For each $1 < j \leq k$
   $tag_{i_j}$ : Compute $cnf_{i_j} = f(k_{i_j}; r_{sys}||c)$
       If $(i_{j-1}, X_{j-1})$ is received and $X_{j-1} = aut_{i_{j-1}}$ then $tag_{i_j} \rightarrow$READER: $(i_j, cnf_{i_j})$
         and timeout
       else timeout
3. READER : If $k$ confirmations are received from the scanned tags of $ID_{group}$ then generate:
       $P_{sgp} = (r_{sys}, ID_{group}, c, i_1, \dots, i_k, cnf_{i_1}, \dots, cnf_{i_k})$
```

Fig. 2. A concurrent ad hoc subgroup proof.

linking of the tags via the reader), then the tags would be unable to communicate with each other (the transmitted signals of the tags are very weak). Phase three consists of three rounds of communication, and each is crucial to provide the data for the subgroup proof. If we were to suppress the exchange of authenticators, or did not implement timers, then the adversary may be able to inject adversarial tags (that are not present) in the subgroup and forge a proof. Also, the implementation of the third round enables an authorized reader to detect certain protocol failures immediately, namely those that lead the initiator tag to timeout. The update of the number sn_{i_1} immediately after it is sent by tag_{i_1} allows the state of the interrogation to be updated even if the protocol round should be interrupted. This, along with timers prevents replay attacks. We assume that the challenge r_{sys}, the keys $K, k_{i_1}, k_{i_2}, \dots$, the session number sn_{i_1}, and the strings $aut_{i_1}, aut_{i_2}, \dots, cnf_{i_1}$, all have the same (bit) length κ, which is the *security parameter* of the protocol.

Phase 3 of this protocol is sequential, and therefore the time taken to identify all the tags of the group is linear in k, which could be a drawback and lead to aborted interrogations when k is large. We can modify Phase 3 to get a concurrent version. In Figure 2 we describe the modifications needed for concurrent execution. Note that in the concurrent protocol the authenticators (i_j, aut_{i_j}) are broadcast independently (one step); similarly the confirmations (i_j, cnf_{i_j}) are broadcast independently.

This protocol can be implemented very efficiently, with a footprint of fewer than 2000 Gate-Equivalents. For a discussion on optimized implementations of pseudorandom functions suitable for RFID applications, we refer the reader to [9]. In the following section we shall formally show that a variant of this protocol that supports anonymity is secure.

5.1 DoS attacks on the subgroup proof

Since authorized readers cannot verify subgroup proofs, a determined adversary can substitute the confirmations of tags with bogus confirmations and cause an RFID reader to generate a proof P_{sgp} in which only r_{sys} and ID_{group} have correct values. Such a proof will of course be rejected by the Verifier—see Section 5.1 for a discussion on DoS attacks. It is important to note that P_{sgp} is an actual proof of simultaneous scanning of the tags of a subgroup by a reader only when all the tags of the subgroup are not compromised: as pointed out in Section 2, a compromised tag can proxy for other compromised tags to forge a grouping-proof.

Note that some of the interrogator tags will recognize bogus authenticators, of if some of the singulated tags (in Phaze 2) are skipped, (*e.g.,* the initiator tag in the sequential version), but again whatever the tags do (or undo), the adversary can undo (or do), because the tags do not share any private information with the reader.

It should be pointed out that our security model is rather strict for most RFID deployments, in which tags are typically close to the interrogating reader, and the wireless medium does not readily lend itself to substitution or deletion attacks—at least not to the level required for a DoS attack of the type described. It is for this reason that in our protocols the tags use authenticators aut_i: these assure the tags of each others presence. deal with typical DoS attacks.

We conclude by noting that there is a subtle difference between the sequential and concurrent versions of the basic grouping proof, which although not an issue for this protocol, will impact on the protocol in Section 6. This has to do with the fact that in the sequential version the initiator can check that the scanned group is complete, whereas in the concurrent version the checking is distributed. If the initiator is responsible for maintaining the state of the group (as in the case of the protocol mentioned earlier) then since error messages cannot be used to declare faults (the adversary can delete/substitute these), one has to use additional confirmations to authenticate protocol flows. To prevent de-synchronization this would require the initiator tag to do $O(n)$ computations, which may be prohibitive for lightweight applications when n is large.

6 An anonymous ring subgroup proof

For our second subgroup proof, the identifier ID_{group} is replaced by a randomized subgroup pseudonym PS. Each tag_{i_j} computes a subgroup pseudonym PS_{i_j} by evaluating $f(K; r_{sys}||sn_{i_j})$ where K is the group key, r_{sys} the random challenge of the reader and sn_{i_j} the value of a random counter. The RFID reader identifies a subgroup by selecting selecting a pseudonym with the smallest index i_1 (the ordering of the indexes is increasing): the corresponding tag will be the initiator tag. The reader informs the tags of its selection and links them. On completion, the reader generates the proof $P_{sgp} = (r_{sys}, PS_{i_1}, c_1, i_1, \ldots, i_k, cnf_{i_1}, aut_{i_2}, \ldots, aut_{i_k})$ in which the pseudonym

```
Parties:   READER($r_{sys}$);   SUBGROUP $\{ tag_{i_j}(K, k_{i_j}, sn_{i_j}), \quad j = 1, \ldots, k \}$

Phase 1
  1. READER $\rightarrow *$ : $r_{sys}$ (a random number)
  2. For $1 \leq j \leq k$
     $tag_{i_j}$ : compute $PS_{i_j}||cnt_{i_j} = f(K; r_{sys}||sn_{i_j})$; set $c_j \leftarrow sn_{i_j}$, $sn_{i_j} \leftarrow cnt_{i_j}$
              $tag_{i_j} \rightarrow *$ : $PS_{i_j}, i_j, c_j$
Phase 2
  1. The READER selects an initiator tag with least index $i_1$ and subgroup pseudonym $PS_{i_1}$
              READER $\rightarrow *$ : subgroup of $PS_{i_1}, c_1, i_1$ is selected
  2. For each $1 < j \leq k$
     $tag_{i_j}$ : parse $f(K; r_{sys}||c_1)$; if the value of the first part is $PS_{i_1}$ then
              $tag_{i_j} \rightarrow *$ : tag $i_j$ belongs to subgroup $PS_{i_1}$
  3. The READER links the tags $i_1, \ldots, i_k$ of subgroup $PS_{i_1}$ and sets a timer
              READER $\rightarrow *$ : subgroup $PS_{i_1}, c_1, i_1, \ldots, i_k,$ is linked
Phase 3
  1. $tag_{i_1}$ : Set timer; compute $aut_{i_1} = f(K; PS_{i_1}||i_1||i_2)$
              $tag_{i_1} \rightarrow$ READER $\rightarrow tag_{i_2}$ : $(i_1, c_1, aut_{i_1})$
  2. For $1 < j \leq k$
     $tag_{i_j}$ : If $(i_{j-1}, X_{j-1})$ is received then set timer, compute $aut_{i_t} = f(K; PS||i_j||i_{j+1})$,
          If $X_{j-1} = aut_{i_{j-1}}$ then
              $tag_{i_j} \rightarrow Reader \rightarrow tag_{i_{j+1}}$ : $(i_j, aut_{i_j})$, and timeout
        else timeout
  3. $tag_{i_1}$ : If $(i_k, X_k)$ is received then compute $aut_{i_k} = f(K; PS||i_k||i_1)$.
          If $X_k = aut_{i_k}$ then compute $cnf_{i_1} = f(k_{i_1}; PS_{i_1}||c_1)$ and
                $tag_{i_1} \rightarrow$ READER : $(i_1, cnf_{i_1})$ and timeout
        else timeout
  4. READER : If a confirmation is received from $tag_{i_1}$ then generate:
        $P_{sgp} = (r_{sys}, PS_{i_1}, c_1, i_1, \ldots, i_k, cnf_{i_1}, aut_{i_2}, \ldots, aut_{i_k})$
```

Fig. 3. An anonymous ad hoc ring subgroup proof.

PS_{i_1} is linked to the challenge r_{sys} (via the group key K and the counter sn_{i_1}) and the challenge r_{sys} is linked to the confirmation cnf_{i_1} (via the secret key k_{i_1} and the counter sn_{i_1}). The protocol is presented in Figure 3.

The Verifier keeps in a database D the values $(r_{sys}, PS_{i_j}, K, k_{i_j}, sn_{i_j})$ that link the secret key k_{i_j}, the group key K and a (precomputed) group pseudonym PS_{i_j} for the interrogation session r_{sys}. D is doubly indexed by PS_{i_1} and k_{i_1}. The pseudonyms are updated with each successful execution of the protocol (the group key K and the numbers sn_{i_1} are used for this purpose). The database D is also used to optimize the performance of the protocol (*optimistic performance* [5]): if the adversary has not challenged the tags in the group (e.g., via rogue readers), and protocol flows were not disrupted since the last group interrogation, then the value of the pseudonym in D will be the one that is actually used by the initiator tag, and therefore the corresponding secret keys can be found directly (one lookup) and used to verify the correctness of the confirmation cnf_1 of the initiator tag. If no value in D corresponds to the pseudonym listed in the proof then the Verifier will have to find the secret key of the initiator from its confirmation $cnf_1 = f(k_{i_1}; r_{sys}||c_1)$ by exhaustive search over all the secret keys of initiator tags in D.

It is important to notice that the data transmitted from the tags to the reader depends on the challenge r_{sys}, the group pseudonym PS_{i_1}, and the session number sn_1. Thus, at every round, the values that the reader receives vary, even if a malicious reader attempts to re-use the same value of r_{sys} in multiple rounds. This provides *unlinkability*.

As in the previous protocol, each step is essential. The main difference is that in this protocol, in the second step the tags use pseudonyms PS rather than group identifiers ID_{group}. The functionality provided by this step, however, is analogous in the two protocols and enables the Verifier to identify the group.

6.1 A baiting attack on privacy

Observe that the ad hoc ring subgroup proof is subject to a *baiting attack* in which the adversary lures tags by replaying a challenge r_{sys} and the corresponding responses from other tags (PS_{i_j}, i_j, c_j) obtained earlier from an authorised interrogation by eavesdropping. If the tag responds then it must belong to the same group. This does not identify the tag, but links tags to earlier subgroup interrogations. The only way to address such attacks on privacy is to require that in Phase 2 the initiator tag be authenticated for the new session by responding to a (pseudo) random challenge from each tag in the subgroup. In future research we shall modify this protocol to address this attack.

6.2 DoS attacks on the anonymous subgroup proof

As in Section 5.1 a determined adversary can inject/substitute authenticators and confirmations and cause an RFID reader to generate a proof in which only r_{sys} has the correct value. Such a proof will of course be rejected by the Verifier—see Section 5.1 for a discussion on DoS attacks. It is important to note that P_{sgp} is an actual proof of simultaneous scanning of the tags of a subgroup by a reader *only when all* the tags of the subgroup are not compromised: as pointed out in Section 2, a compromised tag can proxy for other compromised tags to forge a grouping proof. Our model does not distinguish between a fake proof where only one tag is not present from a fake proof with several tags missing—an adversarial tag may impersonate several tags.

7 Conclusion

We have proposed a novel RFID application in which subgroups of a group of tags generate a proof of simultaneous presence in the range of an RFID reader. We proposed two protocols for subgroup proofs. These proofs are robust, with the second one providing partial anonymity. In future work we shall show how to achieve full anonymity (unlinkability) and will also address forward security issues.

References

1. Gidas Avoine. http://www.avoine.net/rfid/, 2010.
2. Samy Bengio, Gilles Brassard, Yvo Desmedt, Claude Goutier, and Jean-Jacques Quisquater. Secure implementations of identification systems. *J. Cryptology*, 4(3):175–183, 1991.
3. Mike Burmester and Breno de Medeiros. The Security of EPC Gen2 Compliant RFID Protocols. In Steven M. Bellovin, Rosario Gennaro, Angelos D. Keromytis, and Moti Yung, editors, *ACNS*, volume 5037 of *Lecture Notes in Computer Science*, pages 490–506, 2008.
4. Mike Burmester, Breno de Medeiros, and Rossana Motta. Provably Secure Grouping-Proofs for RFID Tags. In Gilles Grimaud and François-Xavier Standaert, editors, *CARDIS*, volume 5189 of *Lecture Notes in Computer Science*, pages 176–190. Springer, 2008.
5. Mike Burmester, Tri Van Le, Breno De Medeiros, and Gene Tsudik. Universally Composable RFID Identification and Authentication Protocols. *ACM Trans. Inf. Syst. Secur.*, 12(4):1–33, 2009.
6. Dang Nguyen Duc and Kwangjo Kim. Grouping-proof protocol for rfid tags: Security definition and scalable construction. Cryptology ePrint Archive, Report 2009/609, 2009. `http://eprint.iacr.org/`.
7. Ari Juels. Yoking-proofs for RFID tags. In *PERCOMW '04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 138–142, Washington, DC, USA, 2004. IEEE Computer Society.
8. Chong Hee Kim and Gildas Avoine. Rfid distance bounding protocol with mixed challenges to prevent relay attacks. In Juan A. Garay, Atsuko Miyaji, and Akira Otsuka, editors, *CANS*, volume 5888 of *Lecture Notes in Computer Science*, pages 119–133. Springer, 2009.
9. Tri Van Le, Mike Burmester, and Breno de Medeiros. Universally composable and forward-secure rfid authentication and authenticated key exchange. In Feng Bao and Steven Miller, editors, *ASIACCS*, pages 242–252. ACM, 2007.
10. Chih-Chung Lin, Yuan-Cheng Lai, J. Tygar, Chuan-Kai Yang, and Chi-Lung Chiang. Coexistence proof using chain of timestamps for multiple rfid tags. In Kevin Chang, Wei Wang, Lei Chen, Clarence Ellis, Ching-Hsien Hsu, Ah Tsoi, and Haixun Wang, editors, *Advances in Web and Network Technologies, and Information Management*, volume 4537 of *Lecture Notes in Computer Science*, pages 634–643. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-72909-9_70.
11. Stefan Mangard, Thomas Popp, and Maria Elisabeth Oswald. *Power Analysis Attacks - Revealing the Secrets of Smart Cards*, volume (ISBN: 0-387-30857-1). Springer, 2007.
12. Selwyn Piramuthu. On existence proofs for multiple RFID tags. In *IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing – SecPerU 2006*, Lyon, France, June 2006. IEEE, IEEE Computer Society Press.
13. Junichiro Saito and Kouichi Sakurai. Grouping proof for rfid tags. In *AINA*, pages 621–624. IEEE Computer Society, 2005.