Efficient probabilistic communication protocol for the private identification of RFID tags by means of Collaborative Readers

Rolando Trujillo and Agusti Solanas

Department of Computer Engineering and Mathematics
Rovira i Virgili University
Catalonia, Spain.
E-mail: {rolando.trujillo,agusti.solanas} @ urv.cat

Abstract

There is a need for efficient communication protocols that allow the private and scalable deployment of RFID systems with a large number of tags. In this paper, we leverage the idea of using distributed, collaborative readers to identify RFID tags and propose a new probabilistic communication protocol for those readers to privately identify RFID tags more efficiently in terms of computational cost and bandwidth usage. Our protocol, which is based on Hash-locks, allows readers to exchange information so as to reduce the amount of tag IDs stored in their caches. Consequently, our proposal improves the scalability of the system, and allows the easy management of large amounts of tags. We provide simulation results showing that our proposal is more efficient and flexible than previous ones in terms of computational cost and bandwidth usage.

Keywords: RFID, protocol, scalability, privacy, security.

1. Introduction

Radio frequency identification (RFID) technology allows the cheap, rapid, and efficient management of goods and people. It has gained importance in a variety of areas, namely transportation, retailing, access control, etc.

Tags and readers are the basic elements of RFID systems. Readers are the active components. They are active in the sense that they do not require others to power them up, this is, they have their own source of energy and they can initiate communications. On the other hand, tags are the passive components of the RFID system. They have neither an energy source nor capability to initiate communications (Note that active tags also exist, however they are far more expensive than passive tags. Thus, the latter are more common regardless of their computational and storage limitations). Readers are used to wirelessly power up tags and retrieve information from them - generally an electronic product code (EPC). Tags can be attached to almost everything, and due to the ability of readers to query tags without visual contact, logistics providers and large retailers find these systems very convenient. Notwithstanding, due to the fact that tags can be accessed remotely, if the proper privacy measures are not taken, unauthorised people could obtain private information (e.g. a company could gauge the inventory of its competitors to modify the prices of its own products accordingly). To cope with this privacy/security problem efficiently, several solutions have been proposed. Most of them might be classified into two main families: i) **tree-based solutions** and ii) **hash-based solutions**.

Tree-based solutions aim at reducing the cost of identifying tags in the readers' side by using a labelled tree of a given depth d, where the label of each node is a unique key. These trees contain N leaves, and each leaf corresponds to an RFID tag. Each label in the path from the tree root to a given leaf belongs to the set of keys that identifies the tag assigned to that leaf. With this set of keys, it is possible for a reader to identify a tag in logarithmic time ($O(\log N)$) (1). However, tree-based protocols suffer from three main shortcomings: i) the authentication process requires several rounds, ii) the size of the sent messages may be too large, and iii) they are vulnerable to compromising attacks (2, 3).

On the other hand, hash-based protocols usually operate on a single round and aim at reducing the identification cost in the side of the tag by using lightweight cryptography (i.e. one-way hash functions and random numbers generators). The improved randomised hash-locks (IRHL) (4) are one of the most interesting proposals in terms of computational requirements in the tag side. The basic identification operation between tags and readers performed by using IRHL consists of three steps (cf. Figure 1):

1. A reader (R) generates a random number r_1 and sends it to a tag (T).
2. T receives r_1 and generates its own random number r_2. With these random numbers and its identification number (ID_T), T computes a response $r = (r_2, h(r_1||r_2||ID_T))$, where $h(.)$ is a one-way hash function

Reader (R)		**Tag (T)**
Generates r_1	$\xrightarrow{r_1}$	
		Generates r_2
	$\xleftarrow{r}$	$r = (r_2, h(r_1 \Vert r_2 \Vert ID_T))$
Computes r^{T_i}		
$r^{T_i} = (r_2, h(r_1 \Vert r_2 \Vert ID_{T_i}))$		
Identifies T as T_i when $r^{T_i} = r$		

Figure 1: Scheme of the Improved Randomized Hash-locks Protocol

and $(||)$ is the concatenation operator.

3. Finally, when R receives r, it has to generate all possible responses $r^{T_i} = h(r_1||r_2||ID_{T_i})$ for all the tag IDs in its database and compare the result with r. When $r^{T_i} = r$ the tag is identified as T_i.

Although the computational cost in the tag side is considerably low, readers must perform a number of operations that linearly grows with the number of tags. Due to the large number of tags that might be managed in, for example, manufacturing processes, readers must hand over this task to centralised mainframes, thus generating bottlenecks and undesired delays and extra costs. With the aim to improve the scalability of the system, Henrici and Müller proposed another hash-based protocol (5). In that protocol each tag belongs to a group and has two keys: a group key and an individual key. Thus, the reader first identifies the group and later identifies the tag within the group. This way, identifying the tag is faster because there are less groups than tags. However, if a tag T is tampered with by an attacker, every tag T' belonging to the group of T could be traced using the group key. Tsudik proposed the YA-TRAP protocol (6) that also uses lightweight hash functions, but it is vulnerable to denial-of-service attacks, and although some improvements of YA-TRAP have been proposed (7), they cannot cope with active adversaries. In (8), the authors propose a new probabilistic and lightweight RFID identification protocol. Sending a series of verification values from tags to readers, readers are able to quickly identify tags with a certain level of confidence. The longer the size of the series of verification values, the higher the level of confidence.

In a different line, Ohkubo, Suzuki and Kinoshita (9), and Conti et al. (10) (11) use hash chain techniques to achieve forward security at the cost of some performance decrease. Thus, although they succeed in obtaining forward security, their proposals can hardly be used in RFID systems with large amounts of tags.

The idea of making tags and/or readers to collaborate has been proposed and tested. With regard to tags, in (12) and (13) a distribution of tags is used to guide mobile robots equipped with RFID readers and to perform precise indoor positioning respectively. Also, in (14) tags cooperate in order to detect when and for how long a tag has been tampered with. With regard to readers, to improve the scalability of hash-based solutions without increasing the number of rounds of the protocol, Solanas et al. proposed an approach that used collaborative readers deployed in a grid structure (15). Instead of having a centralised database with all the tag IDs, each reader maintains a local database (e.g. in a local cache) in which it stores the IDs of the tags located in its cover area and the ones in its adjacent neighbours' area. By doing so, readers no longer need to check all possible IDs to identify a tag but only a smaller subset of IDs in their local cache. Although the proposal in (15) is a step forward in terms of scalability, it replicates too many tag IDs and imposes several constraints to the system (e.g. readers must know the exact distance to the tags and the reader distribution is very rigid). In (16), Ahamed, Rahman and Hoque modified the proposal of Solanas et al. and proposed a more natural neighbourhood structure using a hexagonal grid (cf. Figure 2). Note that this solution reduces the number of neighbours from nine (in the squared grid) to six (in the hexagonal grid). However, this proposal has the same limitations of (15).

The idea of distributing tags amongst a number of readers distributed in a grid or in hexagonal cells might resemble the antenna structure of the well-known GSM system for mobile communications. In fact, readers store information about tags similarly to what visitor location registers (VLR) do with cell phones in GSM. However, there are some fundamental differences that make our problem different:

- In GSM cell phones are active and they are responsible for the registration of their ID in the VLR.
- Visitor location registers (generally) do not exchange information between them. They mainly communicate with a centralised database known as the home location register (HLR).
- We do not consider the existence of a centralised database such as the HLR. Thus, our system might be consider fully decentralised.

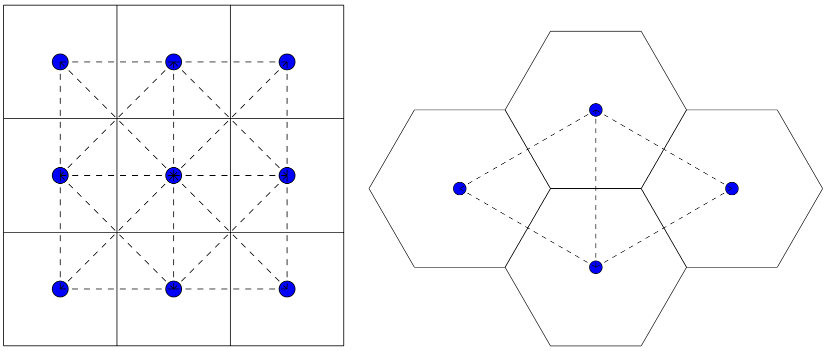


Figure 2: **Left:** Scheme of nine collaborative readers using a squared grid neighbourhood (15). **Right:** Scheme of four readers using a hexagonal grid neighbourhood (16). Dashed lines represent neighbourhood relations amongst readers.

1.1. Contributions and Plan of the Article

In this article, we propose a new probabilistic protocol for the private identification of RFID tags. This protocol drastically reduces redundant information in the caches of the readers and improves the efficiency of previous proposals in terms of computational cost and bandwidth usage. In addition, this new protocol imposes neither constraints on the readers' distribution nor in their neighbourhood relations (unlike (15) and (16)). Thanks to the probabilistic nature of the proposed protocol and its parametrisation capabilities, it can be tuned to balance the relation between computational cost in the reader's side and bandwidth usage.

The rest of the article is organised as follows: In Section 2, we describe our protocol. First, in Subsection 2.1 we summarise our motivation for proposing this new protocol, then in Subsection 2.2 we provide the reader with some definitions and assumptions that are used throughout the article. Next, in Subsection 2.3 the probabilistic nature of our protocol is explained and justified. Section 2 finishes with Subsection 2.4 in which we describe in detail the different messages used in our protocol, and Subsection 2.5 that shows the protocol execution flow. Section 3 shows the results of the simulations, and Section 4 contains some conclusions and further research lines.

2. The Protocol

2.1. Motivation

Hash-based identification protocols for RFID tags have shown to be private and secure but they require a significant computational effort on the readers' side that is generally overcome by using a centralised mainframe, which can lead to bottlenecks and delays. Specifically, the number of operations performed by the mainframe to identify a single tag is a function of the number of tags (n) in the system (i.e. $f(n)$).

An alternative to the centralised solution is the collaborative approach that was first described in (15), whose main idea is to distribute the list of tag IDs amongst all the readers in the system and allow them to identify tags within their cover range without contacting a central mainframe. The solution proposed by Solanas et al. improves the scalability of the system with regard to the centralised solution. Ideally, if we consider a number of readers (m) and a number of tags (n), the number of operations that must be performed by a reader to identify a tag is a function of ($\frac{n}{m}$) (i.e. $f(\frac{n}{m})$). Unfortunately, the protocol proposed by Solanas et al. requires the readers to store the IDs of the tags controlled by neighbour readers, this leads to a significant increase of redundant IDs. If we assume that the redundancy can be expressed by a factor (k), the number of operations that a reader performs to identify a tag using the protocol described in (15) is $f(\frac{k \times n}{m})$, where

$$f(\frac{n}{m}) < f(\frac{k \times n}{m}) < f(n)$$

Our protocol leverages the idea of collaboration from (15), but implements a new set of messages that permit the reduction of redundant information. Ideally, we want $k \to 1$. To do so, thanks to our protocol, readers can be initialised with a parameter $p \in [0, 1]$ that represents the probability for a reader of storing tag IDs from its neighbours. Note that when $p = 0$, the number of redundant IDs is zero and we reach the optimal situation where the number of operations required to identify a tag is $f(\frac{n}{m})$.

In addition, network designers/engineers can balance the reader's computational cost and its bandwidth usage by tuning p. The smaller p is, the lower the number of operations are, but the bandwidth requirements are higher.

2.2. Assumptions and Definitions

In our proposal, instead of using the concept of *unshared cover area*, as described in (15), we use the more general concept of *shared cover area.*

Definition 1 (Unshared Cover Area (A^u))**.** *The unshared cover area of a reader R is the set of locations controlled by R, from which tags can communicate* ***only*** *with R.*

Definition 2 (Shared Cover Area (A^s))**.** *The shared cover area of a reader R is the set of locations from which tags in the system can communicate with R and possibly with other readers.*

From these definitions it can be derived that given two shared cover areas A^s_i and A^s_j, $A^s_i \cap A^s_j$ might be different from the $\emptyset$, whilst given two unshared cover areas A^u_i and A^u_j, $A^u_i \cap A^u_j$ is always $\emptyset$. Although this property of the unshared cover areas might be theoretically useful, it is extremely hard to realise it in practise. Thus, from now on, when we use the term *cover area* we will refer to the more realistic concept of *shared cover area* described in Definition 2 and, for the sake of clarity, we avoid using the superscript s.

Let A_i be the cover area of a reader R_i and let A be the area covered by all the readers in the system. We assume that $A \subseteq \bigcup^i A_i, \forall i$.

Considering our definition of shared cover area, we define the neighbourhood relation as follows:

Definition 3 (Neighbourhood relation)**.** *Two readers R_i and R_j are neighbours if their cover areas A_i and A_j are not disjoint i.e. $A_i \cap A_j \neq \emptyset$.*

Our notions of cover area and neighbourhood are more flexible and realistic than those proposed in (15) and (16). Also, they lead to a simple criterion for connecting readers, i.e., only neighbour readers will share a communication link to exchange protocol messages. We assume that each reader in the system is connected to its neighbours (e.g. using WLAN + SSL) and maintains a local database with a list of pairs (ID_T, ID_R), where ID_T is the identifier of a given tag and ID_R is the identifier of the reader. We also assume that each tag is controlled by a single reader, which is its owner.

Note that by using the notion of shared cover areas the tags moving in a region shared by two readers are controlled by only one of them. On the contrary, if unshared cover areas are used, a tag moving from one unshared cover area to another leads to the change of owner from one reader to another. In Figure 3 an example of this behaviour is shown. If we use shared cover areas, the tag T is controlled by $R2$ throughout its way. However, if we consider the notion of unshared cover area, the tag T is controlled by $R2$ at locations $(1), (3)$ and (5); and it is controlled by $R1$ at locations (2) and (4). This unnecessary change of ownership requires communication between readers and increases the bandwidth usage. Consequently, using shared cover areas may decrease the utilised bandwidth.

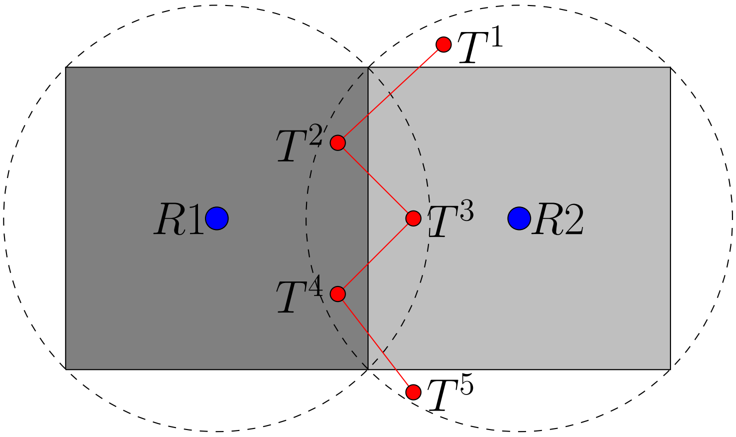


Figure 3: Graphical example of two readers $R1$, $R2$ and a tag T moving. The tag T is captured in different positions at different instants T^1, T^2, T^3, T^4, and T^5 (T^x indicates the position of tag T at time x). The squares represent the unshared areas of $R1$ and $R2$. The circles represent the shared areas of $R1$ and $R2$.

2.3. The Role of p

The number of operations performed in a reader to identify a tag is linear with the number of tag IDs stored in its cache. A reader stores the IDs of the tags in its cover area (for which it is responsible) – we say that that reader is the **owner** of those tags. In addition, a reader may store the IDs of tags located in the cover area of its neighbours. This way, if a tag moves from the cover area of one of its neighbours, it can identify that tag without querying its neighbours.

Each reader is initialised with a parameter p. This parameter defines the probability for a given reader of storing neighbour tag's IDs in its cache. If $p = 1$ the reader stores all the IDs of its neighbour tags, on the contrary if $p = 0$ the reader stores no information about its neighbours' tags. If p takes a value in $(0, 1)$ the reader stores a number of IDs *proportional* to that value. The main goal of p is to reduce the number of redundant IDs stored in the cache of neighbour readers.

The number of IDs stored by a reader i ($\#ID^i$) can be computed as follows:

$$\#ID^i = n_i + p_i \sum_{j=1}^{b_i} n_j^i$$

where n_i is the number of tags owned by i, b_i is the number of neighbours

2	2	2
2	2	2
2	2	2

$p = 0$

5	7	5
7	10	7
5	7	5

$p = 0.5$

8	12	8
12	18	12
8	12	8

$p = 1$

Figure 4: Number of IDs stored in the readers for different values of p considering that each reader is the owner of 2 tags. The neighbourhood relations are the ones described in Figure 2-left.

of reader i, n^i_j is the number of tags owned by the j-th neighbour of reader i, and p_i is the probability for the reader i of storing IDs of tags owned by its neighbours. The total number of IDs stored in the system ($\#ID$) can be computed as $\sum_{i=1}^{m} \#ID_i$, where m is the total number of readers.

In the example shown in Figure 4, it is apparent that reducing the value of p, the number of IDs stored in the caches of the readers is also reduced. Consequently, the number of operations required to identify a tag is also reduced and the whole process of identifying tags scales better.

Note that the protocols described in (15) and (16) do no support the addition of this probabilistic property. Thus, the main goal of the proposed protocol, explained in the following sections, is to allow the use of the parameter p and, as a result, to improve the scalability of the identification process in the readers' side.

2.4. Messages

In our protocol, readers use a number of messages to exchange information about the ownership of tags and collaborate to identify them. Each message sent by a source reader (R_{ID_S}) to a destination reader (R_{ID_D}) makes the latter perform an action regarding a tag (ID_T) (cf. to Figure 5 for a graphical scheme of the message format, and its flow). Depending on the message, the information sent about the tag can be:

- **The tag ID – (ID_T)**: If R_{ID_S} can identify the tag because it has the required information in its cache, it can send ID_T to R_{ID_D}. This might happen for the following messages of the protocol: *Delete*, *I am the owner*, *You are the owner*, and *Search* messages.

- **The response of the tag** $r = (r_2, h(r_1||r_2||ID_T))$ **and the challenge** $r1$: If R_{ID_S} is not able to identify the tag, it sends to R_{ID_D} the challenge

Message			
Operation	**Source**	**Destination**	**Tag**
3 bits	32 bits	32 bits	128 bits
	ID_S	ID_D	ID_T
	ID_S	ID_D	r,r_1

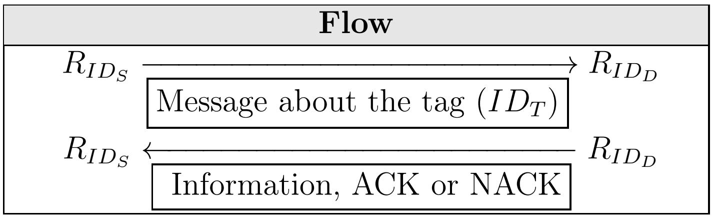


Figure 5: Message format and flow.

$r1$ that it sent to the tag and the answer r received from the tag. This happens for the *Identify* message.

The messages of the protocol are explained in more detail below:

- **Delete -** (ID_T) When R_{ID_D} receives this message, it removes the identifier ID_T from its local cache.
- **I am the owner -** (ID_T) When R_{ID_D} receives this message, it realises that R_{ID_S} claims the ownership of the tag ID_T. If R_{ID_D} was the former owner, it sends a *Delete* message to its neighbours, excepting R_{ID_S} and its neighbours, to let them know that it is no longer the owner of that tag. If R_{ID_D} was not the former owner, then it would generate a random number $x \in [0, 1]$, and if $x \geq p$ it would update its cache with the new ownership information.
- **You are the owner -** (ID_T) When R_{ID_D} receives this message, it takes control over the tag ID_T. It stores the new ownership information in its cache and sends an *I am the owner* message to all its neighbours, so as to propagate the new ownership information.
- **Identify -** (r, r_1) This message is sent by R_{ID_S} when it is not able to determine the ID of a tag (using the Hash Lock protocol). With this message, R_{ID_S} asks R_{ID_D} to identify the tag and return the ownership information stored in its cache. If R_{ID_D} identifies the tag and finds its owner, it sends the ID of the owner back to R_{ID_S}, otherwise it responds with a NACK message.
- **Search -** (ID_T) When R_{ID_D} receives this message it checks whether the tag ID_T is in its cover area. If it finds the tag, it sends an ACK message

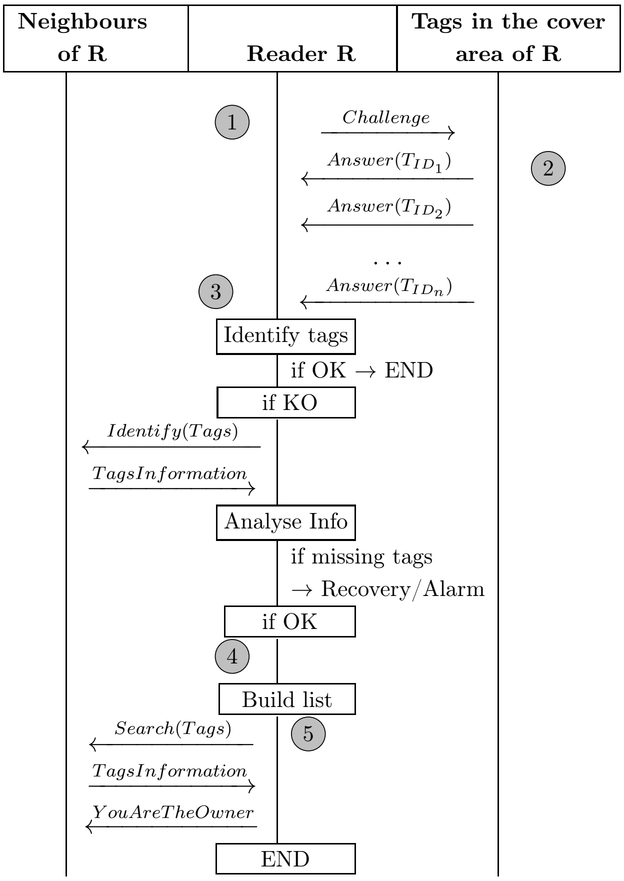


Figure 6: Scheme of the flow of the identification protocol.

back to R_{ID_S}, otherwise it responds with a NACK.

2.5. Protocol Execution

Thanks to the probabilistic nature of our protocol, the number of IDs stored in the local caches of the readers can be reduced with respect to the protocols presented in (15) and (16), however, the flow of messages is a bit more complex. The identification protocol proposed in this article considers three main actors: (i) the tags in the system, (ii) a reader, and (iii) the neighbours of that reader. The protocol schemed in Figure 6 works as follows:

1. A reader (R) sends a challenge (r_1) to the tags in its cover range.
2. All tags in the cover range of R answer the challenge.

3. For each tag (T) responding to the challenge, R tries to identify it using the hash-locks scheme (4) applied to its local cache.
   (a) If it identifies the responding tag, the process finishes.
   (b) Otherwise, R sends an *Identify* message to its neighbours and stores their answers in its cache. If any of its neighbours identifies the tag, R executes a recovery procedure described in the next section.
4. Then, R builds a list (L) containing all the tags that it owns (i.e. which are under its control) and that have not responded to the challenge (e.g. those tags that have left its cover range).
5. For each tag $T \in L$, R sends a *Search* message to its neighbours. After receiving the answers from its neighbours, R sends a *You are the owner* message to the first neighbour that responded positively (i.e. ACK) to the search message.

All the readers in the system periodically use this protocol. By doing so, all tags can be controlled without the intervention of a centralised database. In addition, due to the fact that readers only store information about the tags of their neighbours with a given probability p, the number of redundant IDs is reduced with respect to (15)(16) and, therefore, the computational effort performed by the readers is also reduced.

2.6. Alarm/Recovery Protocol

When a reader is not able to identify a tag and its neighbours do not have information about this tag neither, two possible situation might be tacking place:

- An unauthorised tag has entered the system.
- A tag has been covered (so as to hide it from the readers) and uncovered in a different location controlled by another reader whose neighbours have no information about.

When this situation arises, we propose two possible solutions:

- A centralised solution: This solution is based on maintaining a backup of all tag's IDs in a centralised server. Doing so, when neither a reader nor its neighbours could identify a tag, that reader could request the identification of this tag to the centralised server. Note that, this solution has a high computational cost but does not create bottlenecks because the centralised server is supposed to be used in exceptional cases only.

- A fully decentralised solution: In this case readers can iteratively query their neighbours so as to find the previous owner of the tag in the system. First the reader queries its adjacent neighbours (located at one hop), then it queries the neighbours located at two hops, etc... This procedure finishes when the tag ID is found or when all readers have been queried. In the first case, our protocol keeps working normally, in the second case, an alarm is raised. This procedure is depicted in Figure 7. Note that in the worse case, in which all readers in the system were to be queried, the computational cost would be linear on the number of tags n. Although the computational cost is high and the communication overhead might be significant, this situation should happen rarely, thus, it should not affect the overall efficiency of the proposed protocol.

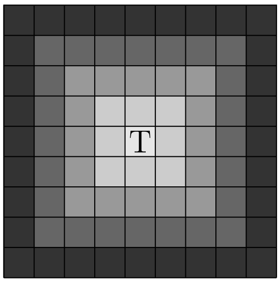


Figure 7: A representation of cells covering the monitored area. An unidentified tag is located in the central cell. The reader in that cell will iteratively query other readers to identify T. Readers in lighter coloured cells are queried first.

2.7. Our protocol in a centralised back-end

Although our protocol has been designed to work in a distributed way. It could be "simulated" by a centralised database (i.e. a back-end) connected to a set of readers properly distributed. By doing so, the back-end would be able to identify tags and "logically" cluster them in regions (e.g. virtually covered by the readers). Thus, intelligently search a tag into these regions might be scalable in terms of computational cost. In addition, this approach averts the communication overhead associated with the exchange of messages between readers because all the communication might be "simulated" within the back-end.

The main problems of using this approach are: (i) using a single centralised database leads to a single point of failure and, (ii) the communication of a single back-end with a (possibly) large number of readers, might create bottle-necks and undesired delays.

It might be said that depending on the special characteristics of the environment in which the RFID system is to be deployed, engineers may decide whether to use our protocol "simulated" within a back-end, or use it as a fully distributed non-centralised protocol.

3. Simulation results

We have developed a simulator to analyse the number of operations performed by the collaborative readers during the execution of our probabilistic protocol, and their bandwidth usage. The simulator allows the deployment of readers without constraints. The number of readers, their cover range, their location, the number of moving tags, and the scenario in which they move can be defined in the beginning of the simulation.

We have concentrated on simulations to analyse the theoretical properties of our protocol and we have left for the future the implementation and testing of a practical prototype. Although there are some limitations in the off-the-shelves RFID tags, there exist some EPC UHF Gen 2 tags that are able to compute hash functions and random numbers (using ARMADILLO (17), and can be read at distances of up to 1 meter. Currently, newer versions with larger reading distances (i.e. 3 m) are under development (cf. `www.oridao.com`).

With the aim to evaluate our probabilistic protocol, we compare it with the one presented in (15), which is referred hereafter as "*original*". Although our protocol has no limitations related to the deployment and range of the readers, the *original* protocol has some. Consequently, we simulate the regular distribution of 24 readers (4 × 6) depicted in Figure 9 that the *original* protocol can handle.

We have considered five different scenarios[1]: (i) An empty scenario in which tags can freely move, (ii) a scenario with narrow corridors, (iii) a

[1] Some of these scenarios were already used in (15)

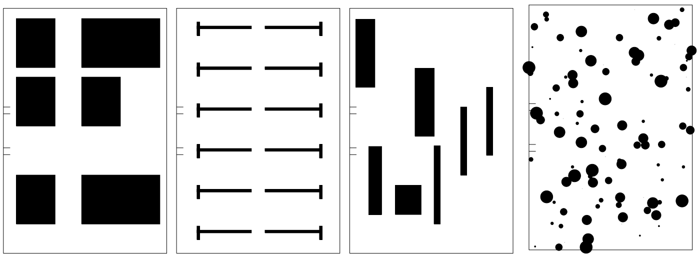


Figure 8: Graphical scheme of the simulated scenarios. (From left to right) Scenario with narrow corridors, scenario with wide corridors, scenario with random large obstacles, scenario with random small obstacles.

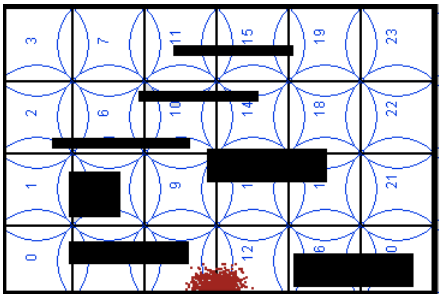
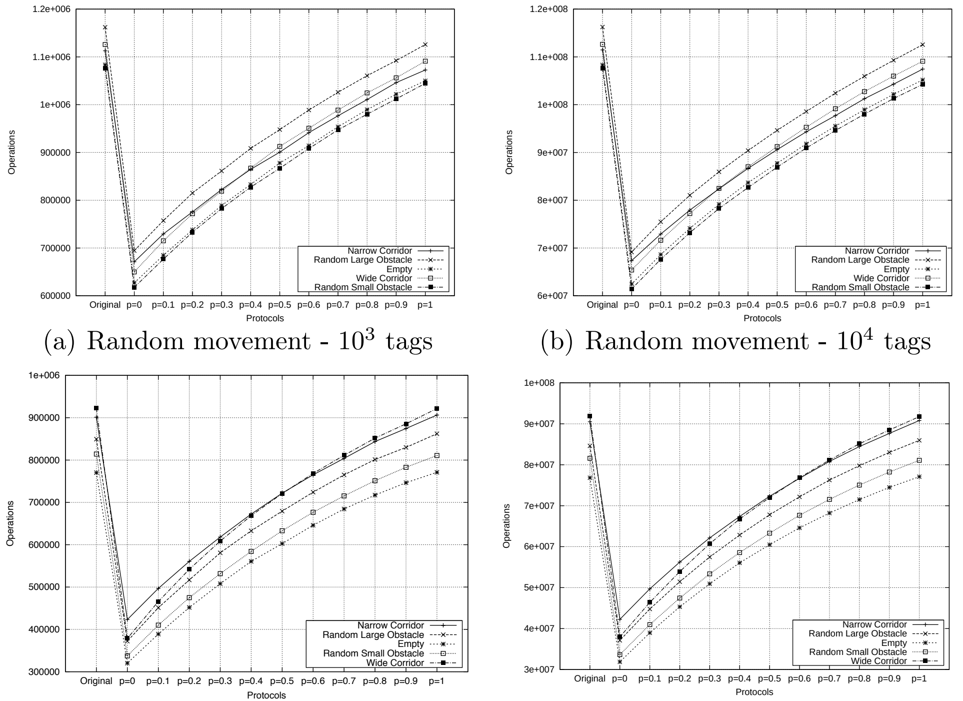


Figure 9: Screenshot of the simulator. The cloud of red dots represents the tags entering the system. Blue circles represent the **shared** cover area of the readers, which are identified by a number. Thick black lines represent obstacles. Finally, thin black lines represent the **unshared** cover areas that the protocol in (15) would use.

scenario with wide corridors, (iv) a scenario with randomly placed large obstacles and, (v) a scenario with randomly placed small obstacles (cf. Figure 9 for a screenshot of the simulator and Figure 8 for a graphical scheme of the four non-empty scenarios). For each scenario we have simulated the movement of 10^3 and 10^4 tags. We have considered two different tags' behaviour: (i) a random movement and, (ii) a semi-directed movement: tags move half of the times randomly and half of the times toward a far, randomly-selected point within the scenario. Each simulation has been repeated 30 times for each value of p in $(0, 1)$ with increments of 0.1. Globally a total of 7.200 simulations[2].

For each scenario we have concentrated in analysing the computational

[2] 2 types of movement × 5 different scenarios × 12 protocols (11 different p + the original) × 30 repetitions × 2 different tags' populations (10^3, 10^4)



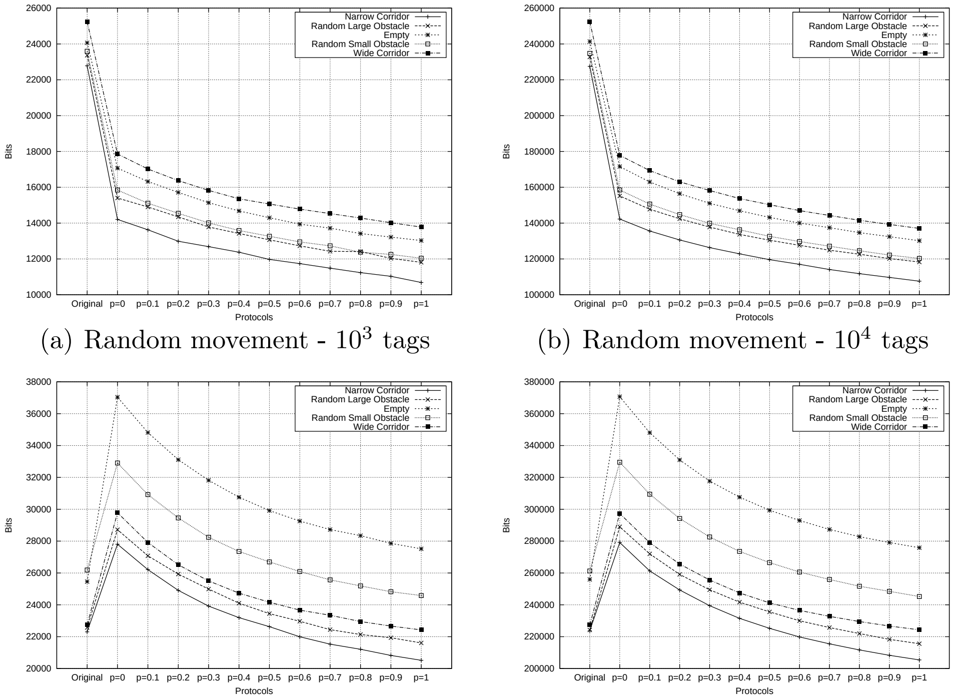
(a) Random movement - 10^3 tags

(b) Random movement - 10^4 tags

(c) Semi-directed movement - 10^3 tags

(d) Semi-directed movement - 10^4 tags

Figure 10: Operations performed by the readers controlling 10^3 and 10^4 tags for different values of p in all scenarios and with two different movement patterns (random and semi-directed) **The lower the better.**



(a) Random movement - 10^3 tags

(b) Random movement - 10^4 tags

(c) Semi-directed movement - 10^3 tags

(d) Semi-directed movement - 10^4 tags

Figure 11: Total number of bits transmitted by the readers controlling 10^3 and 10^4 tags for different values of p in all scenarios and with two different movement patterns (random and semi-directed) **The lower the better.**

cost (in terms of number of operations performed by readers) and the bandwidth usage (in terms of total number of bits sent). Figure 10 shown the results for the computational cost and Figure 11 shows the results for the bandwidth usage. It can be observed that our protocol has a significantly lower computation cost than the *original* protocol. This is specially apparent when the probability p is low[3].

Regarding the bandwidth usage two different behaviours can be observed:

- With random movements: Tags change from a cell to another with low probability (in our protocol). Thus, the number of required messages to update the state of the readers' caches is smaller. In this situation our protocol is clearly more efficient than the original one.

- With semi-directed movements: Tags follow a clear path and change from one cell to another with a higher probability. In this case, our protocol requires more messages (specially in the case of using a low p). Thus, in this situation the original protocol is more efficient for smaller p.

In general, the computational cost is the main concern in RFID identification protocols and, as we have shown above, our proposal clearly outperforms the original protocol in this regard for all scenarios. Indeed, if bandwidth usage is not a concern at all, our proposal with $p = 0$ is the optimal solution. However, our protocol requires more bandwidth to improve the computational cost.

Capturing the trade-off between computational cost and bandwidth is not trivial. Note that, the computational cost and bandwidth usage use different measurement units. However, it is possible to define a measure in order to compare our proposal with regard to the original protocol in terms of both, computational cost and bandwidth usage.

Definition 4 (Trade-off measure)**.** *Let α be a real value in the range* [0..1]*. Let c and b be the computational cost and the bandwidth usage respectively of the original protocol for a given configuration*[4]*. Let c_p and b_p be the com-*

[3] Note that when the probability p tends to 1, our protocol tends to resemble the original protocol in terms of computational cost. However, it is still better in most cases.

[4] A configuration will be defined by the number of tags in the system, the number of readers and their distribution, the scenario, etc.

putational cost and the bandwidth usage of our protocol using the same configuration and p the probability value. Then, the trade-off measure that we propose is computed as follows:

$$d(\alpha, p) = \left(\left(\frac{c_p}{c} - 1\right) \times 100\right) \times \alpha + \left(\left(\frac{b_p}{b} - 1\right) \times 100\right) \times (1 - \alpha)$$

Intuitively, the proposed trade-off measure $d(\alpha, p)$ represents the performance of the original protocol with regard to our protocol using p as the probability value and, considering α the weight given to the computational cost and $1 - \alpha$ the weight given to the bandwidth usage. Note that, when $\alpha = 0$ the bandwidth usage is the only concern whilst when $\alpha = 1$ only the computational cost is considered.

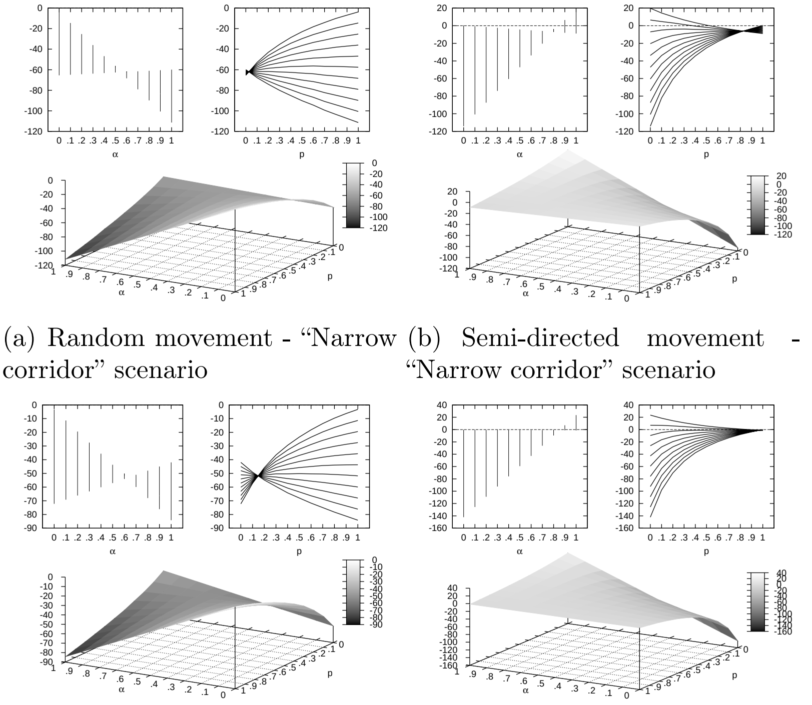
Figures 12, 13 and 14 depict the performance of the original protocol with regard to our protocol using the trade-off measure described above. At the bottom of each figure there is a three dimensional chart showing the values of $d(\alpha, p)$ for each $\alpha \in \{0, 0.1, \cdots, 0.9, 1\}$ and each $p \in \{0, 0.1, \cdots, 0.9, 1\}$. Also, at the top left side and at the top right side of the figure there are the projections of the three dimensional charts for the x-axis and y-axis respectively. In the x-axis projection, for each value of α the values of $d(\alpha, p) \forall p \in [0, 1]$ are shown, whilst in the y-axis projection the plot of the linear functions $d(\alpha, p)$ with α fixed is shown.

It can be observed that our protocol outperforms the previous proposal in most cases. When the movement of the tags is random, our protocol is always better for all possible configurations. When the movement of the tags is semi-directed our proposal is better in 81% of the cases. That leads to a global improvement in more than 90% of all configurations.

4. Conclusion

We have presented an efficient communications protocol for collaborative RFID readers to privately identify RFID tags. With the presented protocol, the centralised management of tags can be avoided, along with bottlenecks and undesired delays.

Our protocol is not a simple modification of previous proposals but a completely different approach that clearly improves the efficiency and flexibility of the whole system. In addition, due to the probabilistic nature of our protocol, the system becomes very flexible (i.e. the relation between computational cost and communications overhead can be easily tuned by means of



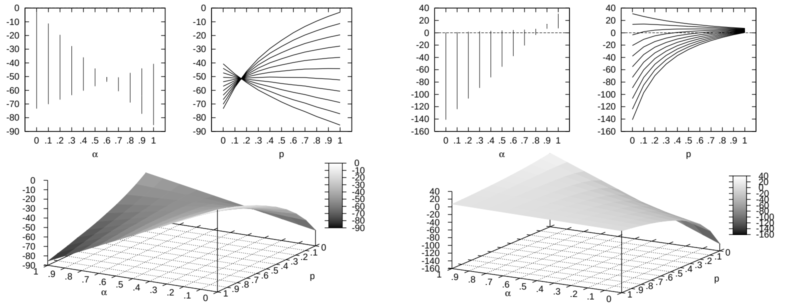
(a) Random movement - "Narrow corridor" scenario

(b) Semi-directed movement - "Narrow corridor" scenario

(c) Random movement - "Wide corridor" scenario

(d) Semi-directed movement - "Wide corridor" scenario

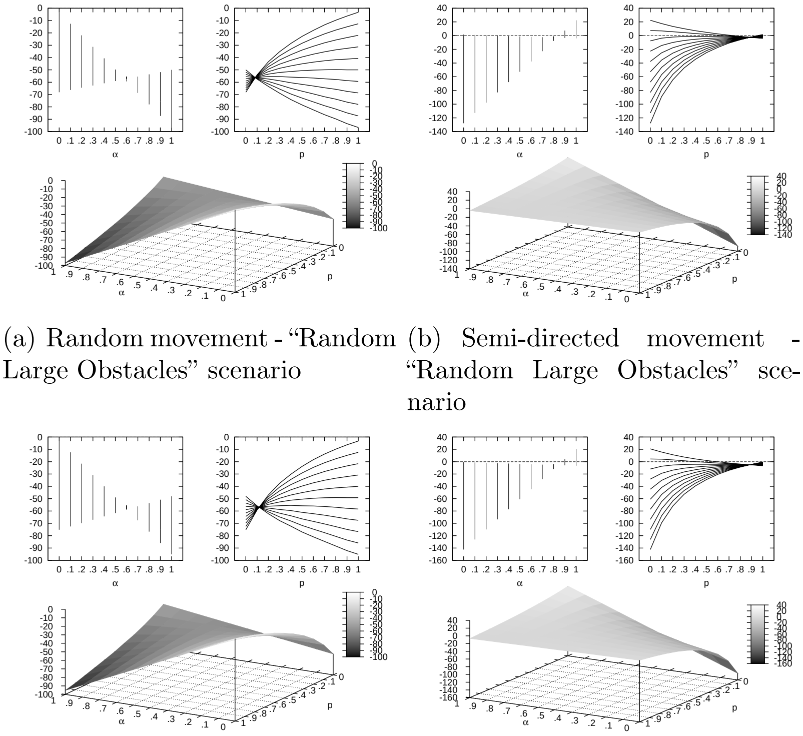
Figure 12: $d(\alpha, p)$ results for 10^4 tags and different values of p and α in the scenarios with corridors. **Values below zero indicate that our protocol is better with respect to the "original"**.



(a) Random movement

(b) Semi-directed movement

Figure 13: $d(\alpha, p)$ results for 10^4 tags and different values of p and α in the empty scenario. **Values below zero indicate that our protocol is better with respect to the "original"**.



(a) Random movement - “Random Large Obstacles” scenario

(b) Semi-directed movement - “Random Large Obstacles” scenario

(c) Random movement - “Random Small Obstacles” scenario

(d) Semi-directed movement - “Random Small Obstacles” scenario

Figure 14: $d(\alpha, p)$ results for 10^4 tags and different values of p and α in the scenarios with random obstacles. **Values below zero indicate that our protocol is better with respect to the “original”**.

p). The simulation results confirm that our protocol outperforms previous approaches like (15).

Although the presented protocol is an improvement, there are some open issues that should be considered in the future, namely (i) study the effect of the number of neighbours, (ii) propose methods to dynamically vary p so as to adapt it to tags' movements, (iii) propose hybrid methods that mix hash-based solutions and tree-based solutions with collaborative readers, etc.

Appendix A. Brief Recap of the "Original" Protocol

The protocol described in (15), that we call "original", was designed to allow multiple readers to collaborate in order to exchange information about tags so as to improve the scalability of the improved randomised hash-locks (IRHL) identification procedure.

In the *original* protocol each reader was responsible for a squared cell and they were all distributed in a grid structure. Note that, using this distribution, the areas covered by each reader were disjoint and, by construction, a tag in a given location could only be queried by a single reader. (This is an important difference with regard to the protocol described in this paper).

In the *original* protocol three main procedures/subprotocols were described:

1. Tag arrival protocol: This protocol is applied when a tag enters the system through a System Access Point or SAP. A reader controlling this SAP identifies the tag using IRHL and communicates to **all** its neighbours the ID of that tag. Then if that tag moves to any of the cells controlled by these neighbours, they will be able to identify it.
2. Roaming protocol: This protocol is used when a tag changes its location from a cell controlled by a reader to another cell. In this case, the reader controlling the destination cell, informs all its neighbours that he is the new owner of the tag and forwards the ID information of the tag to all its neighbours. Also, the previous owner sends a message to its neighbours so as to inform that it is no longer the owner of the tag.
3. Departure protocol: This protocol is used when a tag leaves the system. In this case a reader controlling a System Exit Point (SEP) simply forward to its neighbours the message of deleting that tag from their caches.

Appendix B. On the relation of p and the neighbours' topology

In our protocol, the cache size of each reader linearly grows with p for any readers' distribution. However, neither the number of operations nor the number of messages linearly change with p because they depend on the neighbours' topology. In particular, the number of "Identify" messages sent by a reader R strongly depends on its number of neighbours.

Theorem 1. *Let R be a reader receiving a response from a tag T, whose ID is not stored in its cache. Let O be the owner of T and let $N_{R,O}$ be the set of readers that are neighbours of R and O. Then, for $|N_{R,O}| > 0$, the number of "Identify" messages sent by R during the identification process of T is not linear with regard to p.*

Proof. "Identify" messages are sent when R needs the information of T in order to identify it. These messages can be sent to any of R's neighbours but only O or those that are neighbours of O could answer positively to this message. Therefore, without loss of generality we assume that R sends "Identify" messages only to the readers in $N_{R,O} \cup \{O\}$.

Let $S_{R'}$ be the event that represents a successfully "Identify" message sent by R to the reader R'. Then, $\Pr(S_{R'}) = \Pr(S_{R'}|R' = O)\Pr(R' = O) + \Pr(S_{R'}|R' \neq O)\Pr(R' \neq O)$ and therefore:

$$\Pr(S_{R'}) = \frac{1}{|N_{R,O}|+1} + \frac{|N_{R,O}|}{|N_{R,O}|+1}p \ . \tag{B.1}$$

Let M be a random variable that represents the number of "Identify" messages sent by R during the identification of T. Then, $\Pr(M = i) = (1 - \Pr(S_{R'}))^{i-1}\Pr(S_{R'})$ therefore:

$$E(M) = \sum_{i=1}^{|N_{R,O}|+1} i \Pr(M = i) = \sum_{i=1}^{|N_{R,O}|+1} i(1 - \Pr(S_{R'}))^{i-1}\Pr(S_{R'}) \tag{B.2}$$

Where $E(M)$ is the expected value of M. Considering that $\Pr(S_{R'})$ is a polynomial of degree 1 w.r.t. p (see Equation B.1), then $E(M)$ is a polynomial of degree $|N_{R,O}|+1$ w.r.t. p (see Equation B.2). Therefore, we conclude that, only when $|N_{R,O}| = 0$ the number of "Identify" messages sent by R during the identification process of T is linear regarding to p. □

Corollary 1. *Neither the number of messages nor the number of operations are linear regarding to p when $|N_{R,O}| > 0$.*

Proof. We have shown that the number of "Identify" messages is not linear with p. Thus, due to the fact that the total number of messages includes the "Identify" messages, it is also no-linear w.r.t. p. Regarding the computational cost, it is enough to consider that each "Identify" message causes an exhaustive search in the cache of the reader that receives the message. Thus, the computational cost is also non-linear w.r.t. p. □

References

[1] D. Molnar, D. Wagner, Privacy and Security in Library RFID: Issues, Practices, and Architectures, in: B. Pfitzmann, P. Liu (Eds.), Conference on Computer and Communications Security – ACM CCS, ACM, ACM Press, Washington, DC, USA, 2004, pp. 210–219.

[2] L. Lu, Y. Liu, L. Hu, J. Han, L. Ni, A Dynamic Key-Updating Private Authentication Protocol for RFID Systems, in: International Conference on Pervasive Computing and Communications – PerCom 2007, IEEE, IEEE Computer Society Press, New York City, New York, USA, 2007, pp. 13–22.

[3] G. Avoine, B. Martin, T. Martin, Tree-Based RFID Authentication Protocols Are Definitively Not Privacy-Friendly, in: Workshop on RFID Security – RFIDSec'10, Istanbul, Turkey, 2010.

[4] S. Weis, S. Sarma, R. Rivest, D. Engels, Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems 2802 (2003) 454–469.

[5] D. Henrici, P. Müller, Hash-Based Enhancement of Location Privacy for Radio-Frequency Identification Devices Using Varying Identifiers (2004) 149–153.

[6] G. Tsudik, YA-TRAP: Yet Another Trivial RFID Authentication Protocol, in: International Conference on Pervasive Computing and Communications – PerCom 2006, IEEE, IEEE Computer Society, Pisa, Italy, 2006, pp. 640–643.

[7] C. Chatmon, T. van Le, M. Burmester, Secure Anonymous RFID Authentication Protocols, Technical Report TR-060112, Florida State University, Department of Computer Science, Tallahassee, Florida, USA (2006).

[8] R. Di Pietro, R. Molva, An optimal probabilistic solution for information confinement, privacy, and security in RFID systems, Journal of Network and Computer Applications, Elsevier 34 (3) (2011) 853 - 863.

[9] M. Ohkubo, K. Suzuki, S. Kinoshita, Cryptographic approach to "privacy-friendly" tags, RFID Privacy Workshop. (2003)

[10] M. Conti, R. D. Pietro, L. V. Mancini, A. Spognardi, RIPP-FS: an RFID Identification, Privacy Preserving Protocol with Forward Secrecy, in: International Workshop on Pervasive Computing and Communication Security – PerSec 2007, IEEE, IEEE Computer Society Press, New York City, New York, USA, 2007, pp. 229–234.

[11] M. Conti, R. Di Pietro, L. V. Mancini, A. Spognardi, FastRIPP: RFID Privacy Preserving protocol with Forward Secrecy and Fast Resynchronization, in: 33th Annual Conference of the IEEE Industrial Electronics Society (IEEE IECON 07), Taipei, Taiwan, 2007, pp. 52–57.

[12] J. Bohn,F. Mattern, Super-distributed RFID Tag Infrastrucutres, in: EUSSAI2004. LNCS, 3295, 2004, pp. 1-12.

[13] A. Lim, K. Zhang, A Robust RFID-Based Method for Precise Indoor Positioning, in: Advances in Applied Artificial Intelligence. LNCS, 4031, 2006, pp. 1189-1199.

[14] M. Conti, R. Di Pietro, A. Spognardi, "Who Counterfeited My Viagra?" Probabilistic Item Removal Detection via RFID Tag Cooperation, EURASIP Journal on Wireless Communications and Networking, Hindawi Publishing Corporation (2011) 1–13, doi:10.1155/2011/575171.

[15] A. Solanas, J. Domingo-Ferrer, A. Martínez-Ballesté, V. Daza, A Distributed Architecture for Scalable Private RFID Tag Identification, Computer Networks, Elsevier 51 (9) (2007) 2268–2279.

[16] S. I. Ahamed, F. Rahman, M. E. Hoque, Secured tag identification using edsa (enhanced distributed scalable architecture), in: SAC '08: Proceedings of the 2008 ACM symposium on Applied computing, ACM, New York, NY, USA, 2008, pp. 1902–1907.

[17] S. Badel, N. Dägtekin, J. Nakahara, K. Ouafi, N. Reffé, P. Sepehrdad, P. Susil and S. Vaudenaye, ARMADILLO: A Multi-purpose Cryptographic Primitive Dedicated to Hardware in: Proceedings of Workshop on Cryptographic Hardware and Embedded Systems 2010 (CHES 2010), vol. 6225, 2011, p. 398-412 Springer-Verlog, 2011.