Scalable Trajectory-based Protocol for RFID Tags Identification

Rolando Trujillo-Rasua
Universitat Rovira i Virgili
Department of Computer Engineering and Mathematics
Tarragona, Catalonia, Spain.
e-mail: rolando.trujillo@urv.cat

Agusti Solanas
Universitat Rovira i Virgili
Department of Computer Engineering and Mathematics
Tarragona, Catalonia, Spain.
e-mail: agusti.solanas@urv.cat

***Abstract*—RFID systems allow the fast and automatic identification of items through a wireless channel. Items' information like name, model, purpose, and expiration date can be easily stored and retrieved from RFID tags attached to items. Consequently, in the near future, RFID tags might be an active part of our every-day life when interacting with items around us. However, important security and privacy concerns arise from the wireless nature of this technology because tags are resource-constrained devices that respond to any reader interrogation. Although these concerns have been successfully faced by symmetric cryptography schemes, managing a large number of tags is still cumbersome. Therefore, application-dependant solutions seem to be better for the secure, private and efficient identification of tags. In this paper, we propose a new scheme for the fast identification of tags based on readers strategically distributed throughout the system. Using the spatial location of tags, our scheme is able to expedite the identification of tags regardless of the identification protocol used. Furthermore, differently to previous proposals, our scheme is flexible and adaptable to any scenario and to any movement's pattern of tags.**

I. Introduction

Radio frequency Identification (RFID) is a wireless technology aimed at identifying items automatically. Similar to barcodes systems, each item is tagged with one or more tags containing the item identifier. Then, the RFID reader is able to extract the RFID tag's data wirelessly and sends them to a server in order to identify the item. The wireless nature of RFID systems removes the need for a line-of-sight contact between tags and readers and therefore, improves the efficiency in several applications such as supply chain management, inventory control, etc.

An RFID tag can be classified according to its power source in active or passive. Active tags come with an on-board battery that provides the required energy for larger reading ranges and higher computational resource consumption. On the other hand, passive RFID tags are less powerful devices that do not require on-board battery because they use the reader signal interrogation strength as their energy source. Whether to use an active tag or a passive one depends on the application and the company budget, both offer pros and cons in terms of price, lifespan time, reading distance and computational power.

During the deployment of an RFID system, all these features should be considered together with other major concerns such as security, privacy and scalability. By default, security and privacy are desired features for identification systems, whilst scalability is required just in those applications that should manage a large number of tags. Since RFID tags are resource-constrained devices with limited computational capabilities that respond to any reader interrogation through an insecure channel, ensuring security and privacy in RFID systems are challenging tasks. So far, symmetric key cryptography without key sharing seems to be the most suitable solution for the private and secure identification of RFID tags. However, this approach usually draws a scalability problem for the server. Assume that, looking for private identification, a tag encrypts its identifier using its secret key and sends this encrypted message to the server. Then, in order to determine the tag's identity the server needs to decrypt the message using the tag's key but, direct retrieval of the tag's key is only possible when the server knows the tag's identity. Consequently, the server should perform an exhaustive search looking for the proper key to decrypt the message thus, causing scalability problems.

The balance between privacy and scalability in RFID systems has been extensively studied [1]. Solutions based on key updating [2] or key sharing [3], [4], [5] are shown to be vulnerable against active attacks [6], [7], whilst private solutions based on pseudonyms or symmetric key cryptography [8], [9] are poorly scalable. As a result, application-dependant solutions taking advantage of the intrinsic properties and features of the application, seem to be the most suitable for achieving privacy and scalability at the same time.

Let us consider an RFID system intended for identification and tracking, e.g. tracking of goods inside a supply chain or luggage control inside an airport. In such applications, several RFID readers are distributed over the system in order to identify tags passing through the RFID readers positions [10], [11], [12], [13], [14], [15]. Doing so, it is possible to obtain the trajectory of a tag by concatenating the reader's positions where the tag has been identified. Even in applications without tracking purposes, it makes sense to distribute a set of readers covering strategic points or the whole monitored area [10] in order to identify the tags moving into it. Supermarkets with several output/input doors or department stores are genuine examples of such applications.

Although there are several applications where many tags should be identified using some readers, to the best of our

knowledge, just two protocols exploiting this particular property have been proposed so far [10] and [11]. Furthermore, none of them fully describes how to optimize the identification process in different scenarios. Whilst the protocol in [10] is restricted to be applied in just a few scenarios like open areas, the protocol in [11] still does not scale well enough.

We indicate that the scalability problems of some private protocols can be alleviated not only distributing readers throughout the system, but also by exploiting the spatial location of tags. Indeed, a tagged item usually follows a pre-established life-cycle and then, it could be intelligently identified according to its expected spatial location. In this paper, we propose an adaptive and distributed architecture aimed at efficiently identifying RFID tags based on their expected spatial location. Unlike previous proposals [10], our architecture is suitable for all possible scenarios and adapts itself according to the type of tags' movement. We show empirical results based on synthetic data confirming the superiority of our architecture with respect to the previous proposals [10] and [11].

A. Related Work

Distributed databases have been extensively used for search engines, query systems and inference systems. In this architecture, data can be located according to their demand or characteristics, and the database systems could be parallelized, allowing load on the databases to be balanced amongst servers. Since an RFID system can also generate a large amount of information that may need to be processed as a whole, distributed databases are not an option, but a need for large-scale systems. Some distributed architectures have been proposed for RFID systems [12], [13], [14], [15], but none is designed for the fast identification of tags. Indeed, their concern is how to handle the information coming from a tag after its identification and not the identification itself.

From the scalability point of view, defined as the number of cryptographic operations performed in order to identify a tag, it is not relevant whether a distributed database is composed by several interconnected computers, or it is a computer with several processors, or the database is just a computer that logically distributes the tags' information. What is really important is to guarantee the consistency and synchronization amongst the different databases. In consequence, protocols based on key updating, in general, are not suitable for a distributed architecture. Exceptions are the group-based protocols [2], [4] where each tag belongs to a fixed group and a tag responds not only with its encrypted identifier, but also with the encrypted identifier of the group to which it belongs. Then, after the group identification, the server performs an exhaustive search in the space of identifiers of the identified group, reducing the number of operations in the server side. Note that in these cases, updating the tag's key [2] is not a problem for the distributed architecture because each tag is always authenticated using the same database and hence, synchronization between both parties can be easily achieved. Nevertheless, as tags are not tamper-resistant, compromising one tag leads to the disclosure of a shared key used by other tags during the identification process. Thus, in this type of protocols, scalability is achieved by sacrifying privacy. On the contrary, privacy-friendly protocols based on symmetric cryptography [8], [9], and perfectly suitable for distributed architecture, do not scale well.

To the best of our knowledge, Solanas et al. [10] proposed the first protocol that efficiently uses the spatial locations of tags in a distributed architecture. In this proposal, RFID readers collaborate in order to indentify a tag into the system. To do so, each reader in the system covers a specific squared area, called cell, and the whole monitored area should be covered with these cells guaranteeing that every tag into the system is continuously monitored by at least one reader. In this system, two readers are said to be neighbours if there exits a continuous line between both cells no passing through another cell. The identification process is considerably improved due to the neighbourhood relationships of readers and the fact that a tag always moves through neighbour cells. Although this proposal improves the system's scalability it cannot deal with scenarios that cannot be or do not need to be completely covered by a set of readers. Furthermore, technological challenges arise in the implementation of this proposal because readers must compute the distance to a tag in order to be sure that a tag is in their cells.

Similarly, Fouladgar and Afifi [11] point out that tags are usually queried by the same readers. For instance, tags belonging to people living in the same district will be read, in general, by the readers placed in this district (doors' readers, bus' readers, etc). Therefore, unlike the group-based proposals [2], [4], they propose to cluster tags according to the readers that use them more often. In this way, when a tag responds to a reader R, the reader sends this response to a "Dispatcher" that requests to the database corresponding to R the identification of the tag. If the R's database correctly identifies the tag, depending on the R's rights, the "Dispatcher" decides whether to give the tag's information to R. Otherwise, the "Dispatcher" tries to identify the tag using the database of another reader. The efficiency of this protocol relies on two assumptions: i) It is possible to cluster tags according to their expected spatial locations, and ii) tags are, in general, read by the same subset of readers. However, a tag can have a long life-cycle in which it could be moving through different scenarios. In such case, as the subset of readers assigned to a tag are defined a priori and not dynamically tuned up, this proposal could scale as bad as previous protocols based on symmetric cryptography [8], [9].

II. Our Proposal

As stated in [11], in practice, a reader R probably will identify a tag T several times. In this case, the server overhead can be easily reduced storing T's keys in R's cache [10]. The reader cache is defined as a storage device where a reader saves data concerning only to it. It can be either an external database

securely connected to the reader or a database internally managed by the reader.

If R's cache is small in comparison to the total number of tags into the system, identifying a tag by using R's cache can be considered efficient. For instance, let us assume that tags are static devices and each reader saves in its cache just the keys of the tags inside its reading range. If tags are uniformly distributed, the computational complexity of identifying a tag is $O(\frac{n}{k})$ where n and k are the number of tags and readers in the system. Note that, as the number of readers grows, more efficient the identification process is. However, in practice, neither tags are static devices nor they are distributed uniformly. Therefore, strategies considering the expected spatial locations of tags during their life-cycle should be considered in order to manage readers' cache.

Unlike mobile systems, RFID systems use short communication distances. Thus, after the identification of a tag T by a reader R, T's spatial location can be estimated with a high level of confidence using R's spatial location. By doing so, tags' trajectories can be recorded during their lifespan using several readers in the system. The more readers are scattered in the system, the better the accuracy of the tags' trajectories.

Definition 1 (Trajectory of a tag): Let $\mathcal{R}$ and $\mathcal{T}$ be the set of readers and tags deployed into the system respectively. The trajectory of a tag $T \in \mathcal{T}$ is defined as the sequence of readers $S^T = \{R_1^T, R_2^T, \cdots\}$ such that, for every $i > 0$, $R_i^T \in \mathcal{R}$ and, $t_1^T \leq t_2^T \leq \cdots$ where t_i^T is the time in which the reader R_i^T identified T.

Assuming that tags' trajectories are known before the releasing of the tags into the system, i.e. assuming that tags move according to some patterns then, fast identification of tags is possible by inferring helpful knowledge from these patterns. We distinguish the following cases:

Case 1 (Only one tag in the system): Let us consider now the case where $\mathcal{T} = \{T\}$ and T's trajectory $S^T = \{R_1^T, R_2^T, \cdots\}$ is known before the releasing of T into the system. Then, if R_i^T is trying to identify T, it is because R_{i-1}^T already identified T previously. Therefore, during the identification of T, R_i^T just needs to ask for help to R_{i-1}^T, which probably knows how to identify T. Note that, after the identification, R_i^T should save T's required data in order to identify T without the help of R_{i-1}^T in the future. Using this algorithm, the location of T is always known and, it is possible to efficiently identify it. Unfortunately, this algorithm is useless because a system controlling a single tag does not have scalability problems.

Case 2 (Several tags in the system): Let us consider the case where several tags $\mathcal{T} = \{T_1, T_2, \cdots\}$ are moving into the system and, T_i's trajectory $S^{T_i} = \{R_1^{T_i}, R_2^{T_i}, \cdots\}$ is known, for every i, before the releasing of T_i into the system. When a reader $R \in \mathcal{R}$ receives the response of a tag $T \in \mathcal{T}$, it cannot use T's trajectory to improve the identification process because T's identity is unknown to R. However, R can use helpful knowledge inferred from the tags' trajectories in order to know which readers have higher probabilities of identify it.

Proposition 1: Let $c(R_i, R_j) = |\{< k, l > \text{ s.t. } R_i = R_k^{T_l} \text{ and } R_j = R_{k+1}^{T_l}\}|$ be the number of times that readers R_i and R_j appear consecutively in the set of tags' trajectories and, let $p(R_i, R_j)$ be the probability that R_i and R_j appear consecutively in the trajectory of an unknown tag:

$$p(R_i, R_j) = \frac{c(R_i, R_j)}{\sum_{\forall k, k \neq j} c(R_k, R_j)}$$

When R is trying to identify T, differently to the Case 1, it is not sure about which reader previously identified T. However, using Proposition 1 and the set of tags' trajectories, it is possible to find the reader that most likely identified T previously. By doing so, R might find quickly a reader that can identify T. The problem in this solution is that Proposition 1 can only be used when the set of tags' trajectories is known a priori and, in practice, this is generally not possible.

Case 3 (No movements' patterns of tags): In this case, tags have no movements' patterns. This means that, a priori, it is not possible to know the trajectory followed by a tag during its life-cycle and hence, Proposition 1 is useless for the identification process. For this case, we propose a heuristic algorithm that computes an estimated value of $p(R_i, R_j)$ when needed for any pair of readers R_i and R_j. Using this algorithm and a distributed architecture, we propose a new scheme that efficiently identifies tags according to their expected spatial locations.

A. Protocol Initialization

Our distributed architecture is defined as a weighted, directed, and completed graph $G =< \mathcal{R}, E >$ where $\mathcal{R} = \{R_1, \cdots, R_n\}$ is the set of readers and, E is the set of secure connections amongst the different readers. Initially, all tags' data are distributed amongst the readers' caches. This distribution can be done in different ways: i) storing all tags' data in just one reader's cache, ii) storing each tag's data in a reader's cache randomly chosen, iii) storing each tag's data in the reader's cache corresponding to the reader that should read the tag more often, etc. Actually, this distribution can be done randomly because our protocol balances readers' caches according to the spatial location of tags. Like in an optimization problems, the initial distribution only has influence on how fast the optimal distribution can be found by our protocol. The only constraint of the proposed algorithm is that each tag's data must appear in, at least, one reader's cache.

The weights associated to each connection in G can be also assigned in different ways: i) assigning equal values to all of them, ii) assigning random values to each one, iii) assigning values according to the experience of the systems administrator or iv) assigning to each edge $(R_i \rightarrow R_j)$ the value $p(R_i, R_j)$ (see Proposition 1). Although the latter option is preferred, this is only possible when the set of tags' trajectories is known. In any case, these values will be dynamically tune up during the system life-cycle according to the movements' patterns of tags.

Remark 1: For the sake of simplicity, we denote the edge's weight between readers R_i and R_j as $p(R_i, R_j)$ instead of

using the classical notation $w(R_i, R_j)$.

B. Protocol Execution

When a reader R_i receives an encrypted identification message from a tag T, R_i tries to identify T using its own cache. If this first identification attempt fails, R_i creates a list L of readers such that: $R_i \notin L$ and, for every pair of readers R and R' in L, R is sorted in L before R' if $p(R, R_i) > p(R', R_i)$, i.e. in descending order. Then, for each reader $R \in L$, R_i sends the T's encrypted identification message to R in order to identify T using R's cache. Upon a correct identification of T by a reader R, R_i saves in its cache the information required to identify T and, at the same time, R removes T from its cache. If none of the readers could identify T, it is because T is an invalid tag. More details about our protocol execution can be found in Algorithm 1.

Algorithm 1 Tag identification

Require: $\mathcal{R} = \{R_1, \cdots, R_n\}$ a set of readers and a tag T to be identify by the reader $R_i \in \mathcal{R}$.

1: Let $c : \mathcal{R} \times \mathcal{R} \rightarrow Integer$ be a matrix initialized with non zero values. The matrix c represents the function used in Proposition 1.
2: **if** T is correctly identified by reader R_i **then**
3: **return** The identification process finishes correctly.
4: **end if**
5: Let $L = \{R_i^1, R_i^2, \cdots, R_i^{n-1}\}$ be a permutation of $\mathcal{R}$ such as $R_i \notin L$ and $p(R_i^1, R_i) \geq p(R_i^2, R_i) \geq \cdots \geq p(R_i^{n-1}, R_i)$.
6: **for** $j = 1$ to $n - 1$ **do**
7: **if** T is correctly identified by reader R_i^j **then**
8: T is removed from R_i^j's cache and inserted into R_i's cache.
9: $c(R_i^j, R_i) \leftarrow c(R_i^j, R_i) + 1$.
10: $p(R_i^j, R_i) = \frac{c(R_i^j, R_i)}{\sum_{\forall k} c(R_i^k, R_i)}$.
11: **return** The identification process finishes correctly.
12: **end if**
13: **end for**
14: **return** T could not be identified using any of the readers distributed along the system.

It should be noticed that our protocol does not have neither false positive nor false negative identifications. In the worst case, a tag should be identified using all readers' cache, in which case, like in previous proposals [8], [9], our performance is $O(|\mathcal{T}|)$.

III. Experimental Results

In practice, testing RFID protocols with real data sets of tags movements is complicated, especially because data sets having a significant number of tags movements are hard to obtain. With the aim to overcome this limitation, we define two types of tags' movements and three different scenarios in order to evaluate and compare our protocol with other proposals [10], [11].

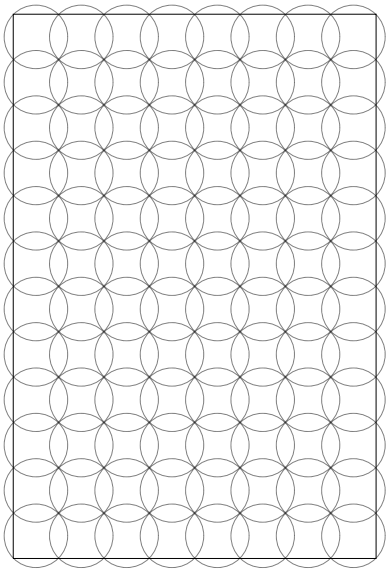

Fig. 1: Open area completely covered by 96 readers .

A. Scenarios

The first scenario is an open area (see Figure 1) where tags can freely move. The area is completely covered by 96 readers uniformly distributed over the whole area. By doing so, we meet the constraints of the Solanas et. al. protocol w.r.t. the readers distribution [10] and hence, comparisons between all the proposals are possible.

The second and third scenarios are representations of the seven Bridges of Königsberg[1]. In these scenarios, people's movements are constrained by the river and thus, they can only use bridges in order to move to different sides of the city. Like people, we assume that tags should not be on the river and then, we design the second and third scenarios using two different readers' distribution. The second scenario (see Figure 2), is a representation of the Königsberg city where 14 readers cover the entire city. This scenario meets the constraints of the Solanas et. al. protocol w.r.t. the reader distribution [10]. Since covering a city by 14 RFID readers can be not practical, we design a third scenario (see Figure 3) that only differs from the previous one in the reading ranges and positions of the readers. Notice that, in the second scenario a tag can be monitored in every part of the city, while in the third scenario a tag can only be read when passing through a bridge. However, due to the movements' constraints in the city and the strategic position of the readers, it is easy to know in which side of the city each tag is located. This is a good example of how intelligently placing readers is possible to obtain accurate tags' trajectories.

B. Tags' Movements

Initially, a tag is located in a valid and random position of the scenario. Later, the tag moves according to two types of

[1]The Seven bridges of Königsberg is a notable historical problem in mathematics. In 1735, Leonhard Euler proved that no Eulerian path exists for the Königsberg city. This result set the foundations of graph theory.

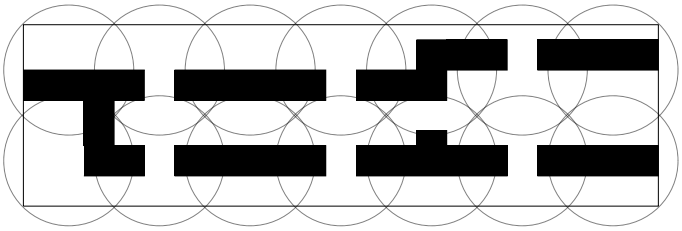

Fig. 2: Königsberg city representation where 14 readers cover the entire city. Black blocks represent the river water and, the seven bridges are represented by the square spaces between black blocks.

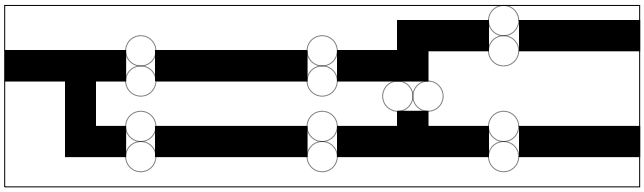

Fig. 3: Königsberg city representation where 14 readers are monitoring the input and output of the bridges. Black blocks represent the river water and, the seven bridges are represented by the square spaces between black blocks.

movement:

- **Random movement:** at each step, a tag chooses a random direction and moves on this direction.
- **Semi-directed movement:** In this movement, a tag always has a target point. Once the tag arrives to its target, it changes the target point to a new random and valid point in the scenario. Then, at each step, with probability 0.5 the tag chooses whether to move randomly or to move on the target's direction.

Between both movements, semi-directed movement can be considered closer to real movement patterns of people. However, unpredictable movement's patterns can be only evaluated using a random movement.

C. Simulations

In order to compare our protocol against the two previous proposals [10], [11], we perform simulations on the three scenarios defined above. For each scenario, different settings defined by the number of tags into the system and the type of movement are used. A simulation process consists of 10^4 tags moving according to some pattern (random or semi-directed) in one of the three scenario. For each simulation process, tags are identified using four different methods:

1) The Fouladgar et. al. method [11] assuming that each tag is in the cache of only one reader. We refer to this method as *Fouladgar one-to-one*.
2) The Fouladgar et. al. method [11] assuming that each tag can be in the cache of several readers. In [11], the authors propose to store the tag's data in the cache of those readers that may read it more often. As this is not possible for the two movement patterns defined above, we make a reasonable assumption: a tag will be in the cache of the readers that have identified it previously. We refer to this method as *Fouladgar one-to-many*.
3) The Solanas et. al. method [10] named as *Solanas*. Due to the constraints of this method, it can only be used within the first two scenarios.
4) The method proposed in this article.

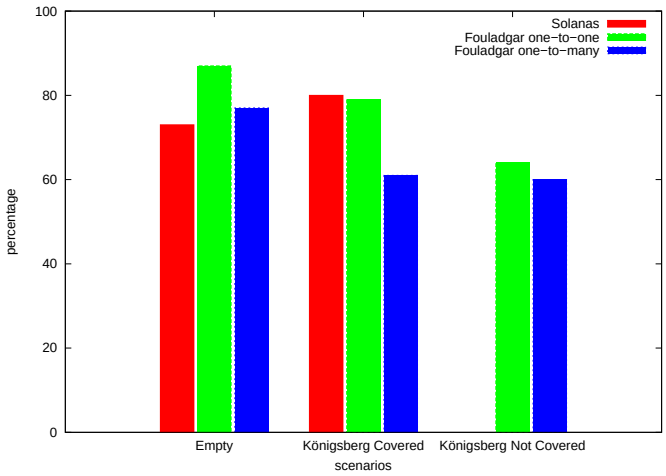

Fig. 4: Percentage of improvement of our proposal w.r.t. previous ones considering the random movement and 10^4 tags.

In order to give statistically sound results, each simulation process is executed 30 times computing the average number of cryptographic operations performed by each method. Figure 4 and Figure 5 show the experimental results obtained for 10^4 tags moving according to the random movement and the semi-directed movement respectively. In both figures, it can be observed that our proposal improves the previous ones in more than 50%. This means that, for any scenario and any type of movement, our proposal executes, in the worst case, the half of the number of cryptographic operations executed by previous proposals.

Our proposal performs better than previous ones mainly due to three reasons: i) after the identification of a tag, the reader saves in its own cache the tag's data in order to identify it faster in the future, ii) the size of readers' caches are minimized in such a way that two readers never share tags' information and, iii) when a reader cannot identify a tag using its own cache, it is able to heuristically find another reader that could identify this tag. This heuristic is one of our main contributions because for the first time tags can be identified according to their type of movement.

In order to check how useful this heuristic is, we perform simulations aimed at comparing our proposal with or without this heuristic. The settings used during the simulations are the same that those defined for the previous simulations. Figure 6 shows, for each scenario and each type of movement, the percentage of improvement of our proposal using our heuristic w.r.t. our proposal without using it. It should be noticed that, the proposed heuristic improves our architecture in all cases.

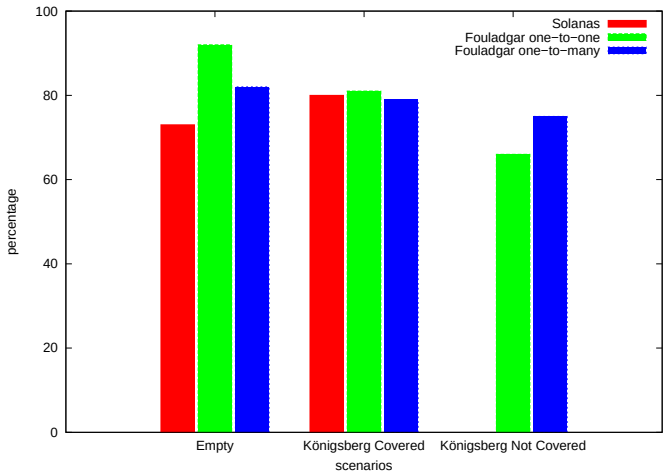

Fig. 5: Percentage of improvement of our proposal w.r.t. previous ones considering the semi-direct movement and 10^4 tags.

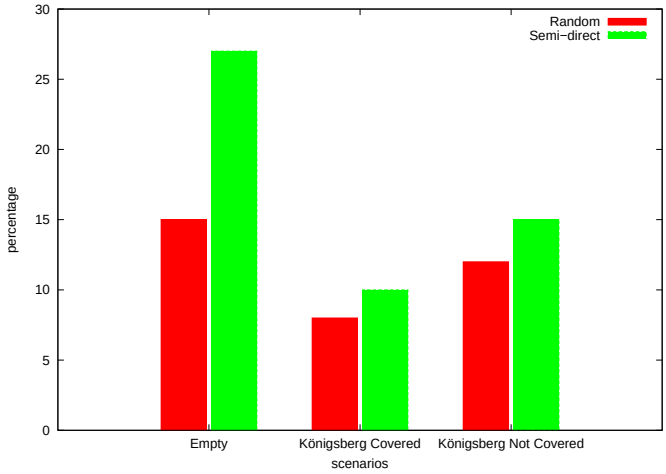

Fig. 6: Percentage of improvement of our proposal with heuristic w.r.t. our proposal without heuristic.

Also, it can be seen in Figure 6 that the heuristic works better when movement's patterns exist (Random vs Semi-directed). This is particularly relevant because, in general, tags move according to some pattern, e.g. the luggage in an airport or the buyers in a shop.

IV. Conclusion and Future Works

In this paper we proposed a new distributed architecture for RFID systems that considerable improves the identification process of tags. Like previous proposals [10], [11], our architecture is based on readers strategically distributed over the monitored area. However, contrary to those proposals, our architecture uses a heuristic that predicts the expected spatial location of tags. By doing so, readers can intelligently collaborate in order to identify tags. Also, our architecture is flexible enough to be applied to any scenario where several readers can be deployed. The experimental results show that our proposal clearly outperforms previous ones [10], [11] in terms of scalability. In the worst case, our proposal is better in more than 50%. As future work, we propose to develop heuristics considering the space-time in the set of tags' trajectories. Note that, movement's patterns of tags might be different in time, e.g. during the day or the night, during the working days or the weekend, etc. Consequently, experiments using real data-set of tags' trajectories must be considered in order to capture the variation of the movement's patterns of tags.

Acknowledgment

This work is partially funded by the Spanish Government through projects TSI2007-65406-C03-01 "E-AEGIS" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", and by the Government of Catalonia under grant 2009 SGR 1135, and by Rovira i Virgili University under project 2010R2B-02.

References

[1] B. Alomair and R. Poovendran, "Privacy versus scalability in radio frequency identification systems," *Computer Communications*, vol. 33, no. 18, pp. 2155 – 2163, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYP-50SXC59-1/2/b10cb573c7e12bb9a063c03900c9491e

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

[3] D. Molnar and D. Wagner, "Privacy and Security in Library RFID: Issues, Practices, and Architectures," in *Conference on Computer and Communications Security – ACM CCS'04*, V. Atluri, B. Pfitzmann, and P. D. McDaniel, Eds., ACM. Washington, DC, USA: ACM Press, October 2004, pp. 210–219.

[4] G. Avoine, L. Buttyant, T. Holczer, and I. Vajda, "Group-based private authentication," in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, 2007, pp. 1 –6.

[5] D. Henrici and P. Muller, "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, 2004, pp. 149–153.

[6] G. Avoine, "Adversary Model for Radio Frequency Identification," Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), Lausanne, Switzerland, Technical Report LASEC-REPORT-2005-001, September 2005.

[7] G. Avoine, E. Dysli, and P. Oechslin, "Reducing Time Complexity in RFID Systems," in *Selected Areas in Cryptography – SAC 2005*, ser. Lecture Notes in Computer Science, B. Preneel and S. Tavares, Eds., vol. 3897. Kingston, Canada: Springer, August 2005, pp. 291–306.

[8] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems." Springer-Verlag, 2003, pp. 201–212.

[9] A. Juels and S. Weis, "Defining Strong Privacy for RFID," in *International Conference on Pervasive Computing and Communications – PerCom 2007*, IEEE. New York City, New York, USA: IEEE Computer Society, March 2007, pp. 342–347.

[10] A. Solanas, J. Domingo-Ferrer, A. Martnez-Ballest, and V. Daza, "A distributed architecture for scalable private rfid tag identification," *Computer Networks*, vol. 51, no. 9, pp. 2268 – 2279, 2007, (1) Advances in Smart Cards and (2) Topics in Wireless Broadband Systems. [Online]. Available: http://www.sciencedirect.com/science/article/B6VRG-4MY6GG8-1/2/a52f3a6f413f045a717a40dceefa398a

[11] S. Fouladgar and H. Afifi, "Scalable privacy protecting scheme through distributed rfid tag identification," in *Proceedings of the workshop on Applications of private and anonymous communications*, ser. AIPACa '08. New York, NY, USA: ACM, 2008, pp. 31–38. [Online]. Available: http://doi.acm.org/10.1145/1461464.1461467

[12] J. Tavares and T. Saraiva, “Elementary Petri net inside RFID distributed database (PNRD) ,” *International Journal of Production Research*, vol. 48, no. 9, pp. 2563 – 2582, January 2010.

[13] C. Bornhövd, T. Lin, S. Haller, and J. Schaper, “Integrating automatic data acquisition with business processes experiences with sap’s auto-id infrastructure,” in *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, ser. VLDB ’04. VLDB Endowment, 2004, pp. 1182–1188. [Online]. Available: http://portal.acm.org/citation.cfm?id=1316689.1316790

[14] R. Agrawal, A. Cheung, K. Kailing, and S. Schonauer, “Towards traceability across sovereign, distributed rfid databases,” *Database Engineering and Applications Symposium, International*, vol. 0, pp. 174–184, 2006.

[15] Z. Cao, Y. Diao, and P. Shenoy, “Architectural considerations for distributed rfid tracking and monitoring.”