PRIVACY FOR RFID-ENABLED DISTRIBUTED APPLICATIONS
Design Notes

Mikaël Ates, Jacques Fayolle, Christophe Gravier, Jeremy Lardon
Université de Lyon, SATIn Team, DIOM Laboratory, Telecoms Saint-Etienne
Jean Monnet University of Saint-Etienne, France

Rahul Garg
LNM Institute of Information Technology, Jaipur-302012, India

Keywords: RFID, Privacy, Modelling.

Abstract: The concern of this paper is RFID systems coupled with distributed applications. We do not treat known RFID attacks, we rather focus on the best way to protect the identity mapping, i.e. the association of a tag identifier, which can be obtained or deduced from the tags or communications including the tags, and the real identity of its carrier. We rely on a common use case of a distributed application and a modelling approach.

1 INTRODUCTION

A Radio Frequency Identification (RFID) system is interesting for many existing applications, and would be considered for new ones, since it is wanted to trigger automated operations encompassing users or items. As a matter of fact, user tracking systems are useful to avoid redundant administrative tasks, and to make human interactions smoother, in multiple identity checking environments. For instance, to manage flows of passengers in airports, patients in hospitals, or employees in access-controlled buildings. Moreover, most of the services based on the diffusion of personalized information depend on the media employed and the accuracy of information. Both depends on the mean for updating the information, spreading information updates, and making accesses to this information as simple as possible. RFID systems contribute to enhance those purposes.

A RFID system(Finkenzeller, 2003)(Glover and Bhatt, 2006) is based on transponders, also known as tags, readers and identity backends. Readers broadcast radio frequency signals to query tags which respond with identifying information. Many people fears of RFID technologies. It often results from misunderstanding and over-considering threats as clandestine scanning, eavesdropping or data leakage. RFID deployments must obviously rely on an architectural design preventing privacy threats (Westin, 1967), but also, on an advertising plan explaining the stakes. However, this fear is also often justified. The privacy of tag's carrier can be very threaten by covert tracking, also known as skimming. More generally, with RFID systems, private information can be seamlessly leaked and multiple readers can collude to track the movements of a person.

The concern of this paper is RFID systems coupled with distributed application dedicated to user services. We do not treat known RFID attacks, as identity theft by cloning or faking, or as denial of service by tag disruption. We rather focus on protecting the identity mapping, i.e. the association of a tag identifier, which can be obtained or deduced from the tags or communications including the tags, and the real identity of its carrier. We depict through the paper a common use case of a distributed application to endorse our proposition.

2 PRIVACY CONCERNS

2.1 Overall RFID Privacy Concerns

The benefits of RFID systems are also often synonym of privacy threats (Langheinrich, 2007):

- *Automation i.e. no user intervention is required to read a RFID tag.*
- *Identification of a tag carrier.*

Ates M., Fayolle J., Gravier C., Lardon J. and Garg R. (2009).
PRIVACY FOR RFID-ENABLED DISTRIBUTED APPLICATIONS - Design Notes.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 324-328
DOI: 10.5220/0001862003240328

- *Integration i.e. it can be difficult to visually ascertain the presence of a tag.*
- *Retrieve information, other than identification, carried by, or linked to, RFID tags.*

It is always a better choice to not keep sensitive information on devices as tags if it possible to do differently. But even if there is no other sensitive information than an identifier on the tag, this is enough to be a real concern. Covert tracking for tags carried by human people threaten their privacy enabling their localization and the tracking of their activities. Moreover, a tag may reveal the users' membership to their organization which delivered their tags. A set of tags may represent multiple memberships and constitute a personal profile, e.g. identifying a person as being customer of some transport companies, of media and clothes stores, etc.

2.2 Our Privacy Concerns

We are concerned with an RFID system which can operate with many parts of the information system, as opposed to closed applications like access building software which could be, for most of them, isolated from the rest of the information system. The obvious primary question is 'who represent the threat?'. Attack from the outside would mean that somebody want to track one of our member, or maybe, if it knows his real identity, link it with a tag identifier. So he wants to make one of our tag to leak its identifier. This means two requirements, our tags should not respond to readers other than ours, and, to prevent eavesdropping, the tag identifier should not be revealed in clear. Readers authentication by tags and anonymous communication, hidding the tag identifier, between tags and readers is the ideal. But for know it is not the common case. We thus make here a study when tags have no such capabilities. Hence we have to take in account the leakings of the communications between tags and readers. We have also to take care of attacks against RFID information in the rest of the information system. This means to monitor carefully the RFID backend mapping tag identifiers and user identifiers. And also, the communications and applications logging records which could allow to link the real identity with a tag identifier as well.

2.3 RFID Cryptographic Protocols

We should be able to encrypt and authenticate the communication between the tags and the readers with cryptographic protocols (Lee et al., 2006; Song and Mitchell, 2008). A public key infrastructure with an authority certificate embedded in tags, and readers broadcasting their certificate seems relevant. However, most of the RFID tags have limited computation capabilities which, for now, prevent from spreading asymmetric cryptography in the RFID domain. Moreover, it requires to implement mechanisms which authenticate without compromising anonymity, i.e revealing the tag identifier, which, with symmetric cryptography, could be resume to the key search issue(Juels, 2006). Finally, the tag identifier should not be a single static data string. In cryptographic scheme as the Song's one (Song and Mitchell, 2008), the tag identifier changes at each authentication.

3 FOCUS ON THE DISTRIBUTED APPLICATION

We here rely on a use case which is the deployment of a trivial application of agenda consultation. This application consists in a fast and easy way to inform people of their agendas and updates, thus making the people inner-organization life easier and lightening some of the administrative tasks of the bureau. Users are provided with RFID tags allowing them to trigger the display of their agendas with the help of a RFID reader standing close to a large screen. We have chosen a simple application to focus on the privacy concerns implied by the RFID system, and not on access control questions. Our members and their multiple group memberships (section, language, options, etc...) are all registered in a central identity registry (henceforth idregistry). Their agendas are registered in a database (henceforth agendadb), and only depend on group memberships, not on their own identity, personal agenda are not concerned by this application. Moreover, the agendas are already publicly available for insiders. Hence, we consider that the service of displaying agenda is not a privacy threat by itself.

3.1 Overview of the Use Case

As a matter of fact, the main privacy threat comes from the identity mapping, i.e. the association of a RFID tag and a person. As a consequence, we have to take care of:

- *User identifiable information written on tags;*
- *Records in the information system linking tags identifiers with users identifiers;*
- *Communications linking tag's tid with user's id.*

We can however deduce from the agenda depending only to group memberships, that a tag identifier has only to be linked to a set of groups, not to a user. The user anonymity should thus be implicitly preserved. We consider the anonymity as the state of being not identifiable within a set of subjects, the anonymity set (Pfitmann and Kohntopp, 2001). Hence, the set of group memberships is an anonymity set which does not allow user identification, i.e. it does not exist in our information system a unique combination of group memberships identifying a particular user. We could thus only associate tags with a list of group membership, and provide users with a tag corresponding to its agenda. As a consequence, it means that, for a tag, we would not be able to determine which is its carrier. It is obviously a wise privacy choice. But it is totally inefficient if we want to update user information linked to tags, especially if only a subset of users are concerned. It would be necessary to update and redistribute all tags. For this reason, it is necessary to record the mapping between a tag identifier (henceforth *tid*) and a user identifier (henceforth *userid*) in the information system. These recordings are also synonym with communication containing these mappings. And these mappings will be explicitly performed during the administration of the RFID system. Considering tags as the most sensitive part of the system, and also as one of the less controllable, we do not record on them this association, or any other identifying information. This mapping information is recorded in a dedicated database, the RFID database (henceforth *RFIDdb*).

3.2 Elements For Modelling

The goal is to represent easily where information about identity is issued and registered in a distributed application, and thus, to highlight where it is necessary to use encryption. This representation must encompass events to allow their correlation with records to map identifiers. We have choose to not represent the time to keep this informal model as simple as possible. However, the notion of event is intuitive. If someone is currently scanned by a reader, events are triggered in the distributed application, hence, either the intruder can capture communications, or look at the records written at this time. If all the communications are encrypted and the logs are not time stamped or encrypted, the problem is solved. So the goal of this approach is to highlight these communications and log records, ant then, to design a safe common architecture for RFID-enabled distributed applications.

We transcribe any communication and recording with respectively the primitives *com()* and *rec()*. The real user identity can be deduced from the physical presence of a human near a reader. We transcribe this as *hum(userid)*. We represent by *map()* the information of identity mapping. We note *linkid* an identifier of a network segment of the distributed application. We note *locid* an identifier of a record location.

A mapping can be either directly accessible, and will then be noted taking userid and tid as parameters *map(userid,tid)*, or deduced from the correlation of two events, and represented by *map(linkid,locid)* when, respectively, a segment and a location are identifiers of the events which may be correlated.

We note as follow the events which reveal the *userid*:

- *a user interact with the RFID system and she is humanly identifiable (hum(userid));*
- *a communication containing a userid (com(userid,linkid));*
- *a record containing a userid (rec(userid,locid)).*

We note as follows the events which reveal the *tid*:

- *a communication containing a tid (com(tid,linkid));*
- *a record containing a tid (rec(tid,locid)).*

Two of each kind of these events may be correlated to reveal a *map(userid,tid)*. For instance, the correlation can be done by deduction from the time the events happened. Mappings issued from the correlation of distinct events are as follows[1]:

- *map(linkid,linkid)* $\leftarrow$ *com(tid,linkid)* $\wedge$ *com(userid,linkid)*
- *map(linkid,locid)* $\leftarrow$ *com(tid,linkid)* $\wedge$ *rec(userid,locid)*
- *map(linkid,hum(userid))* $\leftarrow$ *com(tid,linkid)* $\wedge$ *hum(userid)*
- *map(locid,linkid)* $\leftarrow$ *rec(tid,locid)* $\wedge$ *com(userid,linkid)*
- *map(locid,locid)* $\leftarrow$ *rec(tid,locid)* $\wedge$ *rec(userid,locid)*
- *map(locid,hum(userid))* $\leftarrow$ *rec(tid,locid)* $\wedge$ *hum(userid)*

Two elements, one picked from the set *TIDS* of events containing *tids* and one picked from the set *USERIDS* of events containing *userids*.

[1] Although we have use logic symbols, there is here no more than what have been said, i.e the correlation of two events to deduce a mapping. In other words, no one should expect establish a proof with this formalism.

3.3 Architecture Design

3.3.1 Administration Application

The *RFIDdb*, which contains all the past and current carriers of each tag, is managed by a dedicated administration application (henceforth *adminApp*). *adminApp* is used by administrators to entry the *map(tid,userid)* in the RFIDdb. These recordings, and the events triggered, must be secured. An other functionality of *adminApp* is to review the historic of tag carrying. Both require the following steps:

- *userids are obtained from the idregistry.*
- *tids are obtained from the tags by RFID readers.*
- *maps are recorded in, or read from, the RFIDdb.*

The Figure 1 depicts these events, concerning *tids*, *userids* and *mappings*, triggered by adminApp. We can identify the mappings directly reachable in single events, and the mappings issued from the correlation of two distinct events. There are four mappings directly reachable in single events(rec(map(userid,tid),db1), rec(map(userid,tid),log1), rec(map(userid,tid),log4) and com(map(userid,tid),eth3)). There are nine mappings issued from the correlation of two distinct events. For instance, for the link *rf1*:

- *map(rf1,eth2) ← com(tid,rf1) ∧ com(userid,eth2)*
- *map(rf1,hum(userid)) ← com(tid,rf1) ∧ hum(userid)*
- *map(rf1,log3) ← com(tid,rf1) ∧ rec(userid,log3)*

3.3.2 RFID-Enabled Distributed Applications

The RFID-enabled distributed applications offering user services must access to *RFIDdb* to obtain the *userid* from the *tid*, for next retrieving information about the tags' carriers from the information system. In the agenda application (henceforth agendaApp) users present their tag in the reader near-field region to display their agendas, which requires the following steps:

- *tids are obtained from tags by RFID readers.*
- *userids are obtained from RFIDdb thanks to the recordings containing the tids.*
- *group memberships are obtained thanks to the userids from idregistry.*
- *agendas are obtained thanks to the group memberships from agendadb.*

The privacy threats are thus exactly the same as the one described in Section 3.3.1. The only difference is that with *adminApp*, a uni-directional communication from *adminApp* to *RFIDdb* contain the event *com(map(userid,tid),eth3)*. Whereas, for *agendaApp*, the mapping requires a bi-directional communication, *com(tid,eth3)* from *agendaApp* to *RFIDdb*, and *com(userid,eth3)* from *RFIDdb* to *agendaApp*. Then, for any application based on this distributed applicative scheme, the threats are the same. It should be taken into account architectures with multiple RFID readers which only add events *com(tid,[rfx])* between tags and readers. *[rfx]* is the notation for the set of identifiers of these links. Furthermore, we can generalize the architecture to any RFID-enabled distributed applications if we consider that, in the part of the distributed application not being directly in charge of RFID processes, no event containing *tids* should happen. We can thus split the RFID-enabled distributed application in two parts. The part with RFID can be made of multiple distinct agents in charge of readers. We assume that there is only one RFIDdb per RFID-enabled distributed application. Hence, the AdminApp should only reside in the RFID side and its communications only authorized to the RFIDdb. For RFID-enabled distributed applications, it means to identify communications required to satisfy the RFID concerns. Then, communications containing RFID information between those two parts should only be authorized from the host performing the requests of mapping and only to the RFIDdb.

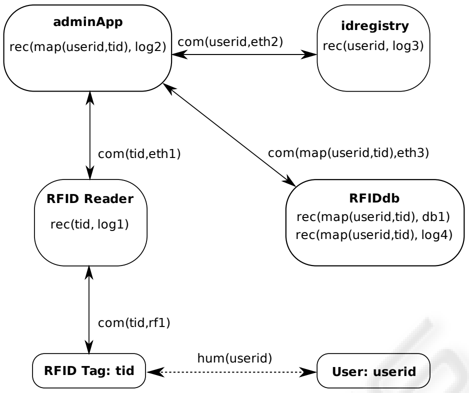

Figure 1: Events triggered by the adminApp.

3.3.3 Security Mechanisms

According to the mappings identified in Section 3.3.1, the sensitive events are known. All the communications and time stamped log records should be encrypted, transcribed with *enc()*.

We consider that it is unfeasible to act on the event hum(userid).

The access control, transcribed with authz(), relies on only two roles granted for accessing the *RFIDdb*: the former with read/write rights for the adminApp role, and the latter with only read rights for the application role. A safe implementation should rely on a single agent allowing to assume the adminApp role. Any other authorized distinct agent should assume the application role with its own identity. *enc(authz(com()))* means that the communication is encrypted and that the requestor must authenticate and be authorized.

The resulting architecture for *agendaApp* is depicted in Figure 2.

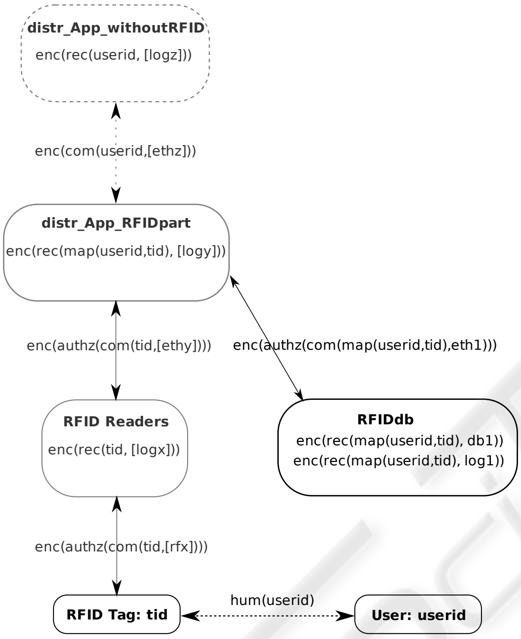
distr_App_withoutRFID
enc(rec(userid, [logz]))
enc(com(userid,[ethz]))
distr_App_RFIDpart
enc(rec(map(userid,tid), [logy]))
enc(authz(com(tid,[ethy])))
enc(authz(com(map(userid,tid),eth1)))
RFID Readers
enc(rec(tid, [logx]))
RFIDdb
enc(rec(map(userid,tid), db1))
enc(rec(map(userid,tid), log1))
enc(authz(com(tid,[rfx])))
hum(userid)
RFID Tag: tid
User: userid

Figure 2: Implementing privacy for any RFID-enabled distributed application.

4 CONCLUSIONS

We have depicted a RFID system coupled with a distributed application through a simple use case, enough to highlight privacy concerns. We have mainly taken care to the mapping between a tag's carrier and its real identity. We have then proposed an informal method, and its associated model, for this privacy threat analysis in a distributed environment. In a first time, to specify the functionalities of the expected application revealing each event containing identifiers. Then, to model the application thanks to the elements of modelling highlighting possible correlations leaking the identity mappings. Once the threats are known, it is a trivial implementation concern to identify where it is required to secure the application. And, splitting the distributed application in two parts, one with RFID, and one without, highlights the sensitive information flows which should be monitored.

REFERENCES

Finkenzeller, K. (2003). *RFID Handbook: Fundamentals and Applications in Contactless smartcards and Identification, 2nd edt.* John Wiley and Sons.

Glover, B. and Bhatt, H. (2006). *RFID Essentials*. O'Reilly.

Juels, A. (2006). Rfid security and privacy: A research survey. *IEEE Journal on Selected Areas in Communication*.

Langheinrich, M. (2007). Privacy and rfid. In *Security, Privacy, and Trust in Modern Data Management*, pages 433–450. Springer.

Lee, H., Yang, J., and Kim, K. (2006). Enhanced mutual authentication protocol for low-cost rfid. Technical report, Auto-ID Labs.

Pfitmann, A. and Kohntopp, M. (2001). Anonymity, unobservability, and pseudonymity - a proposal for terminology. In *Lecture Notes in Computer Science: Designing Privacy Enhancing Technologies*, volume 2009, pages 1–9. Springer Berlin / Heidelberg.

Song, B. and Mitchell, C. J. (2008). Rfid authentication protocol for low-cost tags. In *WiSec'08: Proceedings of the 2008 ACM Conference of Wireless Security*. ACM.

Westin, A. F. (1967). *Privacy and freedom*. Atheneum Publishers.