Secure and Efficient Tag Searching in RFID Systems using Serverless Search Protocol

Sheikh I. Ahamed[1], Farzana Rahman[1], Endadul Hoque[1], Fahim Kawsar[2], and Tatsuo Nakajima[2]

[1]Ubicomp lab, MSCS Dept., Marquette University, USA
[2]Dept. of Computer Science, Waseda University, Japan

iq@mscs.mu.edu, {bondhons084, endadulhoque }@yahoo.com
{fahim,tatsuo}@dcl.info.waseda.ac.jp

Abstract

In the coming pervasive society, Radio Frequency Identification(RFID) Tags will be affixed within every product and object including human. This technology is anticipated to be a major technology which will be utilized by several pervasive services where these tags will be used to identify various objects. However, the use of RFID tags may create new threats to the security and privacy of individuals holding RFID tags. Therefore, widespread deployment of RFID systems preserving users' privacy and data integrity is a major security challenge of the coming year. That is why research related to privacy preserving authentication is growing. And the envision is that: RFID systems can intermingle into human lives if they can offer practical, low cost and secured mechanisms for tag authentication which has been in the midst of researcher's interest for almost a decade. One extension of RFID authentication is RFID tag searching. Any RFID authentication protocol which provides adequate security and privacy can be used for RFID tag searching. However, when the number of tags within a system will increase, the overall data collection cost will also increase. Therefore, more efficient tag searching method is needed. RFID search protocol can play a major role for tag searching which has not been given much attention so far. But we firmly believe that in near future tag searching will be a significant issue. In this paper we propose a lightweight and serverless RFID tag searching protocol. This protocol can search a particular tag efficiently without server's intervention. Furthermore they are secured against major security threats.

Keywords: RFID, security, serverless search protocol

1. Introduction

RFID systems (referring to Radio Frequency IDentification) embrace one important development track in the framework of ubiquitous or pervasive computing. RFID allows effective identification of a large number of tagged objects without actual physical or visual contact. The use of RFID system is appropriate basically everywhere that has to be automatically labeled, identified, stored, or monitored. RFID systems have been displaying a continual market development in selected market segments for decades now. Depending on various application conditions, some of which are sector-specific, RFID systems are being used over the whole range of possible technological complexity. In other segments RFID's means of automatic and contactless identification is being tested in numerous pilot studies. It is such a technology whose potential application can be found in practically all areas of daily life and business. Theoretically the application areas of RFID systems are unlimited. From industry viewpoint, they are applicable in various fields such as supply chain management, employee identification, product maintenance etc.

An RFID system is composed of three main components; tag, reader and Back-end database. RFID tag carries an object identifying data. When a tag receives a query from a reader, the tag transmits information to the reader using RF signals. RFID reader reads and sometimes re-writes the stored data in a tag. After a reader queries to a tag and receives information from the tag, the reader forwards the information to a Back-end database. Back-end server is powerful in computational capacity and manages lots of information related to each tag. Actually in server based system, back-end server plays an essential role and it is quite easy to check validity of tags or reader, which is very important for privacy protection and security issues. Consequently a malicious reader can hardly obtain precious information from tags in such a system.

But, the major drawback of central server based system is that the readers always have to be connected to the server, which limits usage of RFID systems in remote locations where connectivity with server cannot be ensured. Besides having a single database makes the whole system more vulnerable to privacy attacks. Central server has knowledge of all tag secrets and tag information. So if the database is collapsed by an adversary, entire user community's privacy is jeopardized.

The expansion of RFID technology is limited because of security and privacy concerns. Conventional security primitives cannot be integrated in RFID tags as they have inadequate computation capabilities with extremely limited resources. So security and privacy issues must be addressed before the enormous deployment of RFID tags in omnipresent environment. That is why research community devoted themselves in search of appropriate authentication protocols that will ensure RFID privacy and security without compromising the cost.

Security and privacy protection is a major issue in another situation where a single reader and multiple tags are present. In all such practical situation, often a reader needs to determine whether a particular tag exists within a group of tags. This is referred to as RFID searching. Tag searching with the help of central database is not a challenging issue. But without the help of server, the reader has to search a tag entirely by itself. This is a critical task because it is vulnerable to privacy and security threats [5]. For example, through the broadcast of a search query, a reader in a warehouse wants to search for a tag which belongs to a precious object. Now if the tag exists, it will reply and an adversary will become sure that a valuable object exists around it. Such security threats are very common while searching. So introducing straightforward, secure and practical RFID searching is one of the major goals of researchers now a day.

However, RFID searching can be thought as an extension of RFID authentication. By authenticating every tag within a group, we can find out the desired tag. But as the number of tags increase, the ability to search RFID tags becomes invaluable when the reader requires data from few RFID tags rather than all the tags in the collection. If the reader has to authenticate each tag one at a time then the entire searching process will become very time consuming. Though tag searching is very useful and necessary in many RFID applications, secure searching methods have not received enough attention in research literature. So in this paper, we suggest efficient search protocol which ensures security and privacy. A preliminary version of this protocol has been presented in [7].

So far serverless searching is discussed only in [5]. In serverless system, reader has to search, authenticate as well as provide security without server's intervention. This departure from server based system will also reduce cost for RFID system deployment in many areas where tag searching is done frequently like supply chain management and E-passport.

In this paper, we tried to find solutions to the following questions: a) how readers can search a particular tag without the help of server? b) how a tag identifies that the communicating reader is legitimate? Here, we propose a low cost, secured, serverless search protocol that provides solutions to the preceding questions. And all these characteristics are ensured without a back end server which makes our proposal suitable for various application areas.

1.1 Our major contribution

I. In this paper we are proposing serverless, forward secure, anonymous searching protocols for RFID tags.

II. We have considered all the major attacks and our search protocols are secure against those attacks. We considered security of both tags and readers as both can be attacked by adversaries

III. In this paper, we discussed some real life application challenges and their solutions using our proposed serverless search protocols.

The remainder of the paper is organized as follows. The next section presents related work. Some major security requirements for RFID search protocols are reflected in section 3. Section 4.1 provides some preliminaries for the rest of the paper. Section 4.2 provides search protocols and their security analysis is discussed in section 4.3. Some real life challenges and solutions are discussed in section 5. And finally in section 6 some concluding remarks are reported.

2. Related works

The assortment of research literature on RFID searching is inadequate although it is a major issue in its real life implementation. We stated in section 1 that RFID searching is an extension of RFID authentication. So we will go through some relevant literatures on RFID authentication. But we will mainly concentrate on the single serverless searching protocols proposed so far [5].

RFID systems are severely vulnerable to many security and privacy threats. That is why numbers of techniques have been proposed for ensuring RFID security and the assortment of authentication protocols is quite extensive [3]. Most of the authentication protocols proposed so far is backed by central database. One such famous authentication protocol is YA-TRAP [6] which is not secured against DOS attack. Another hash chain based RFID identification protocol is RIPP-FS [2], which shares a private symmetric key with server. Another famous lightweight authentication protocol is OSK [4], which suffers from the problem of desynchronization. In [1], Avoine and Oechslin modified OSK which removed the scalability problem. Serverless authentication protocols are proposed for the first time in [5]. In this paper, Chiu et al. proposed a challenge response based mutual authentication protocol. But the reader has to do lot of computation to find out id of the required tag. And their protocol 2 is not purely and strongly anonymous.

Serverless RFID searching protocols were also proposed in [5] for the first time. According to this protocol, a reader wishes to find out whether a specific tag is within its vicinity by broadcasting $h(f(r_i, t_j)||n_r) \oplus id_j, n_r$ and r_i . Based on this search query, only the intended tag, if exists, reply with its encrypted id. Other tags within the reader's vicinity reply a random number based on certain probability. Tags authenticate the reader based on the search query and reader authenticates tags based on the reply "string". Both valid query and valid replies are generated by legitimate parties.

3. Security requirements

A number of research literatures have dealt with several privacy and security issues of RFID. Some of which are discussed in section 2. RFID searching should also be secured because an adversary may want to find out whether precious objects exist by querying tags. So we point out the following security attacks which must be addressed by a search protocol in order to ensure security:

Tracking: It is tough for an adversary to track a tag if it does not have any information about the tag. But the adversary can track a tag, if the tag replies with a constant response each time it is queried. So protocols should be designed such that a tag neither reveals its id nor replies with constant response.

Cloning: In order to clone a tag, an adversary needs to know the secret key shared by the tag with its authorized reader. So, to be secured against cloning attack, protocols should never reveal the shared secret key.

Eavesdropping: Security must be ensured against eavesdropping attacks so that an adversary cannot impersonate a legitimate tag by replaying an eavesdropped message.

4. Search protocols

In practical implementation of RFID, a reader often wants to find out whether a particular tag exists around him within a group of tags. One solution can be to perform authentication protocol for each of the tags of that group. But this is an inefficient approach as the number of tags within a system is likely to be huge. So another solution for RFID searching can be: reader will search for a tag and only that particular tag, if it exists, will reply in return. So the objective of secure RFID searching should be: the reader will search a specific RFID tag which he is authorized to access. And tags will reply with valid answers only if the reader is legitimate.

In this paper, we present different search protocols. According to the protocols, tag identifier is not passed to the reader in response to a reader's query. Whereas the tag sends certifying information to the reader in such a way that only the authorized reader is able to find out whether this is the desired tag. In this way, the reader can become sure about the existence of the tag that he is searching for.

4.1. Notation and assumption

We refer an RFID reader denoted as R. Each R has a unique identifier r and a contact list $\mathcal{L}$. We will describe the contents of $\mathcal{L}$ a little later. R obtains r and $\mathcal{L}$ from a trusted center, TC, after authenticating itself. The TC is a trusted party who deploys all the RFID tags and authorizes any RFID reader. For the sake of simplicity we assume that R and TC communicate through a secure channel.

According to our proposal, Each RFID tag T contains a unique value id, a unique secret t in its nonvolatile memory. All readers and tags also have knowledge of a pseudorandom number generator $\mathcal{P}(.)$ which takes a $seed$ as an argument and outputs a pseudorandom number according to its distribution. After generating a pseudorandom number, $\mathcal{P}(.)$ makes use of a function $\mathcal{M}(.)$ that generates next $seed$ of the pseudorandom number generator. For each authorized tag, the current *seed* is stored in the reader in its nonvolatile memory. And in case of tag, a current $seed$ is stored for the authenticated reader in its nonvolatile memory. The initial $seed$ is computed by TC and stored in the tag and the reader by TC. The $seed$ stored in both the reader and tag, is defined in the following manner:

$$seed_i^0 = f(r\,,t) = h(r \parallel t)$$

$$seed_i^{k+1} = \mathcal{M}\left(seed_i^k\right)$$

where, $h(.)$ is a one way hash function and i represents i^{th} tag or reader. Superscript k is used to represent the $seed$ after generating $(k-1)^{st}$ pseudorandom number from the distribution of pseudorandom number generator. $seed_i^k$ is used to generate k^{th} number according to its distribution. From now on, we will refer to k as the step k. In fact, both the $seed$ s in tag and reader become same after each authentication and searching.

Subscripts are used to describe a particular R or T and their respective variables. Thus a particular RFID reader i will be R_i, with an identifier r_i and contact list $\mathcal{L}_i$. A tag j is T_j and has a secret t_j. The contact list $\mathcal{L}$ contains information about the RFID tags which a particular R has access to. $\mathcal{L}$ has a list of all $seed^0 = f(r, t)$ that TC has authorized R to access. So reader i, R_i authorized to access tags $T_1, \cdots, T_n$ will have $\mathcal{L}_i$ after authenticating itself to TC where,

$$\mathcal{L}_i = \left\{ \begin{matrix} seed_1^0 : id_1 \\ seed_2^0 : id_2 \\ \cdot \\ \cdot \\ seed_n^0 : id_n \end{matrix} \right\}$$

Note that R_i does not know any of the tags secret t. It only knows the outcome of the function $f(r, t)$ as $seed^0$. We assume that the TC cannot be compromised, and that all readers once authenticated by the TC are trusted. We denote an adversary as ϱ .

$seed_i^k$ and $seed_j^k$ represents current $seed$ of pseudo random number distribution of R_i and T_j respectively, where superscript k bears aforementioned meaning. n_j^k is a pseudorandom number generated by i^{th} reader for the j^{th} tag using $seed_j^k$ at step k. Similarly, n_i^k is another pseudorandom number generated by j^{th} tag for the i^{th} reader using $seed_i^k$ at step k.

We can assume $\mathcal{M}(.)$ as an irreversible one way hash function. Therefore a $seed$ can't be linked to the previous $seed$. Although we haven't explicitly shown the use of $\mathcal{M}(.)$ by $\mathcal{P}(.)$ in the protocols, after generating a new pseudorandom number $\mathcal{P}(.)$ executes $\mathcal{M}(.)$ to update the $seed$ which will be stored in a nonvolatile memory of reader or tag. For example, T_j generates a pseudorandom number n_i^k for R_i using $seed_i^k$ stored in T_j and at the same time next seed $seed_i^{k+1}$ is also generated. Therefore $\mathcal{P}(.)$ performs like:

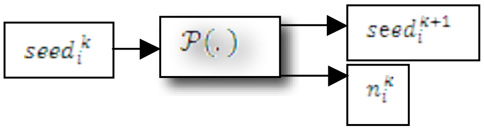

Figure 1. Seed refreshing and pseudorandom number generation mechanism.

Correspondingly, in case of a reader R_i, $\mathcal{P}(.)$ performs in the same way by replacing i with j. Whenever the $seed$ needs to be stored in nonvolatile memory it is explicitly mentioned in the protocols. For example, $seed_j = seed_j^k$ represents that $seed_j^k$ is stored in $seed_j$ along with id_j in contact list, $\mathcal{L}$ of reader.

R_i	RFID reader i which wishes to execute search
$T_{desired}$	Desired RFID tag that the reader is searching for
$seed_{desired}^k$	seed residing in the contact list of R_i for the RFID tag $T_{desired}$
$n_{desired}^k$	Pseudorandom number generated by the reader R_i for tag $T_{desired}$, based on $seed_{desired}^k$
$T*$	All tags within the vicinity of the reader R_i

Table 1: Summary of notations for Search Protocols

4.2. Protocols

Suppose, a reader R_i is searching for a tag which we are referring to as $T_{desired}$. One way of searching may be according to Search Protocol 1. Here R_i broadcasts its r_i wishing to find $T_{desired}$. Before getting reply from the tags, R_i computes next random number ($n_{desired}^k$) for the desired tag using $seed_{desired}^k$. Now, all the tags receiving r_i will reply with next random number (n_i^k) for this particular reader. Reader compares computed random number with those received from the tags. If a match occurs, reader becomes sure that the $T_{desired}$ is present.

Search Protocol 1

```
$R_i \rightarrow T*$ : $Broadcast\ r_i$
$R_i$        : $Compute\ n_{desired}^k = \mathcal{P}(seed_{desired}^k)$
$T*$         : $n_i^k = \mathcal{P}(seed_i^k)$
$R_i \leftarrow T*$ : $n_i^k$
$R_i$        : $If\ (n_i^k = n_{desired}^k)\ then$
                 $T_{desired}\ found$
             $Else$
                 $T_{desired}\ not\ found$
```

One main problem of this protocol is that it is a one side authenticated search protocol. Here tags do not authenticate the readers before replying. So they cannot know whether they are replying to an adversary or to a valid reader. Tags should only reply to authorize the reader. But here tags reply whenever they see a query. Sometimes even an adversary may query a group of tags to find out if a particular valuable tag is present. So the tag needs to authenticate the reader before replying. It means that when R_i broadcasts the search query, every tag, not only the tag that satisfies this query, needs to authenticate R_i before replying.

Another issue is, as seeds are not updated in both parties after each search, tags will reply to the same reader with the same answers in subsequent queries. If an adversary queries with a previously listened r_i, tags will reply with the exact same values as before. Although the adversary will not be able to find out which tag the reader was searching for, it will become sure that the same search is taking place. Querying several times with different r_i, adversary can get a pattern for queries and replies.

The problem of replying with the fixed answer for the same reader can be solved if we update the seed in both parties after each search, which is specified in search protocol 2.

Search Protocol 2

```
$R_i \rightarrow T*$ : $Broadcast\ r_i$
$R_i$       : $Compute\ n_{desired}^{k} = \mathcal{P}(seed_{desired}^{k})$
$T*$        : $n_i^k = \mathcal{P}(seed_i^k)$
              $seed_i = seed_i^{k+1}$
$R_i \leftarrow T*$ : $n_i^k$
$R_i$       : $seed_j = seed_j^{k+1}\ for\ each\ tag$
                $T_j\ replying\ with\ n_i^k\ ,where$
                $1 \leq j \leq n$
              $If\ (n_i^k = n_{desired}^{k})\ then$
                $T_{desired}\ found$
              $Else$
                $T_{desired}\ not\ found$
```

In this protocol, after replying to the search query each tag will update its seed. A reader will update the seeds of only those tags, which have replied. But here the problem is that reader has to update $O(n)$ seeds in worst case scenario. Therefore, the reader is burdened with more computations.

Another problem of this protocol is synchronization. By querying tags, an adversary can desynchronize the tags and reader very easily. As a result after de-synchronization, in spite of the presence of the desired tag, a legitimate reader cannot access it.

Therefore, we can set up our goals for searching as follows. Tags should only respond to authenticated readers. The reader should only query authenticated tags. And both parties should update their seeds after authentication. All these properties are incorporated in our final search protocol which is search protocol 3. Her, the reader issues a query in a way that only an authenticated tag can understand and the tag replies in such a manner that only an authenticated reader can understand.

Search Protocol 3

```
$R_i$        : $Compute\ n_{desired}^{k} = \mathcal{P}(seed_{desired}^{k})$
$R_i \rightarrow T*$: $Broadcast\ n_{desired}^{k}$
$T*$         : $n_i^k = \mathcal{P}(seed_i^k)$
             $If\ (n_i^k = n_{desired}^{k})\ then$
                 $n_i^{k+1} = \mathcal{P}(seed_i^{k+1})$
                 $seed_i = seed_i^{k+2}$
                 $R_i \leftarrow T_j : n_i^{k+1}$
             $Else$
                 $R_i \leftarrow T_j : rand\ with\ probablity\ \lambda$
```

```
$R_i$    : $n^{k+1}_{desired} = \mathcal{P}(seed^{k+1}_{desired})$
           $If\ (n_i^{k+1} = n^{k+1}_{desired})\ then$
               $seed_{desired} = seed^{k+2}_{desired}$
               $T_{desired}\ found$
           $Else$
               $T_{desired}\ not\ found$
```

In this protocol, R_i computes $n^k_{desired}$ and broadcasts it to find out $T_{desired}$. All tags receiving $n^k_{desired}$ will compare this with their own individual n_i^k. If a match occurs, the tag will know that it is an authorized reader. A match can occur only in $T_{desired}$ because only a legitimate reader can know its seed. Therefore only a valid reader can generate valid $n^k_{desired}$. Hence after authenticating the reader in this way, $T_{desired}$ will reply with next number (n_i^{k+1}) for this reader and update its seed. And for those tags in which a match doesn't occur, they will reply with a random number with probability λ. Reader now computes $n^{k+1}_{desired}$ and compares it with n_i^{k+1}. If a match occurs, then reader can be sure that it is a valid tag as only a legitimate tag can generate this. Therefore, the reader now updates its seed for $T_{desired}$. This protocol is resistant against almost all the attacks. Security analysis for this protocol is discussed in the next subsection.

In search protocol 3 we let some other tags also reply in addition to the desired tag to put the actual reply in disguise. Each tag receiving a search query that does not match with the request will have some probability λ of replying. So by observing tag replies, an adversary cannot reveal a particular tag that the reader is searching for.

4.3. Security analysis of search protocols

Tracking: Our final protocol is resistant against tracking. Tracking attack in searching is slightly different from the one found in security literature. Here adversary cannot pick a particular tag to track. Rather, he can only track a tag that has been searched for by a legitimate reader. Consider the following attack. ϱ eavesdrops on the transaction between a reader R_i and tags. So he knows the queries and replies. He will not be able to reverse compute the replies or learn the query but he can certainly be sure that a searching has taken place. However he cannot be sure, which tag $T_{desired}$ reader was searching for, as besides the desired tag other tags also replied with probability λ. Now ϱ can replay previously listened $n^k_{desired}$ to track $T_{desired}$. But after the previous successful searching between R_i and $T_{desired}$, both parties have changed their seeds. So $n^k_{desired}$, send by the adversary, will not match with the one computed by $T_{desired}$. As a result $T_{desired}$ will reply will a random number. At the same time other tags will also reply a random number. If ϱ continues to query with different $n^k_{desired}$, all tags including the desired tag will reply randomly. Therefore ϱ will not be able to track a tag.

Cloning: Consider the following cloning attack. R_i queries to search a tag $T_{desired}$. If $T_{desired}$ is present it will reply. At the same time other tags will also reply. Suppose, ϱ finds out the tag the reader was searching for. Now if he is able to clone $T_{desired}$, then he can fool R_i by not replying or even giving a false reply. As a result, R_i will assume that the desired tag $T_{desired}$ does not exist in this group. In our protocol, this attack is impossible. Because ϱ is unable to find out, which tag the reader was searching for.

Eavesdropping: Here ϱ observes all the queries between a reader and tags. And his goal is to use the data to impersonate a fake reader R_i or a fake tag T_j. Our protocol is powerful against this attack. In our protocol ϱ will not be able to find out the expected reply of the reader as more than one tag will reply. He can only observe $n^k_{desired}$ send by the reader. With his little knowledge he cannot impersonate R_i or T_j, because after the last successful searching between R_i and $T_{desired}$, both of them have updated their seeds. So both of them are now expecting new values which are not known by ϱ. Therefore by eavesdropping ϱ cannot launch a replay attack by using previous values.

5. Illustrative examples

In this section we have drawn a couple of application scenarios that can be directly benefited from our approach presented in this paper.

1. User Interactions in a smart space: A smart space typically contains multiple smart objects offering several invisible services. Users' personal devices are usually used to interact with the smart space. Discovering invisible services securely and authenticating the users are interesting research problems in the smart space domain. Our approach offers promising solutions to both of these problems. Iconic images embedded with RFID tags can advertise invisible services and user terminals can be equipped with an RFID reader. A user can search for a specific service (tags in this case) or can initiate a service by touching the tag. Considering the pre-negotiation between the reader and the tags, secure discovery and authentication mechanism can be easily achieved applying our protocol.

***2. Emergency Evacuation System*:** Safety at the workplace and saving human lives in emergency situations has always been one of the highest priorities in all civilized countries. Fast and efficient evacuation of building complexes, and keeping account of all involved in unpredictable circumstances with hundreds or even thousands of people escaping from danger zones, is an essential component of any emergency system. In the case of emergency, conventional evacuation strategies rely on emergency authority (Fire Brigade, Police etc) to check each and every floor and to direct the personnel to come out of the building in the case of emergency situation. This approach has experienced limited success for safe and effective evacuation operation. A better mechanism or process is needed. In an emergency evacuation process, one major task is to identify whether a certain person is still within the danger zone. For this purpose, our search protocol can be used efficiently to search a particular person within the emergency area. Our search protocol is also perfect for such situations because it is likely not to have a central server in an emergency situation.

3. Container search within seaports: There are hundreds and thousands of containers within a seaport. Containers are parked and stacked by hundreds of employees and countless drivers who deliver containers from remote locations. Moreover, containers are also unloaded from ships in order to deliver them to different customers and locations. Whether a particular container has already been unloaded from the ship or not, whether a specific container has arrived at the seaport for shipment or not, are some of the major tasks performed within seaports. But it is quite impossible to search for a particular container manually. That is why seaports in different countries have long been searching for technologies that can identify specific containers and that can confirm the existence of containers within seaports. One solution to the aforementioned problem can be to use RFID tags for container identification. Now through the use of our serverless search protocols, it will be quite easy to search for a particular container by searching the tag. If a container's tag id (in fact *seed*) is known, then we can invoke a search operation with this id within the seaport. If the container is present within the seaport then according to our protocol, definitely that particular tag will reply. Thus we can be sure about the container's existence.

4. Mishandled bag search within Airports: Passengers suffer a lot due to inefficient bag handling system in the airports. Passengers have to deal with customer service representative in search of their lost baggage. The industry refers to this as "Mishandled bag". Every missing or mishandled bag costs the responsible airline approximately $80 to $120, or an average of $100 per bag [8]. And yearly this figure rises to approximately $146 million. Moreover, this type of events degrades the reputation of the responsible airline. However a simple, cost-effective, efficient solution to Mishandled Bag can be achieved using our search protocol. Whenever a passenger arrives to customer service representative to report about missing bags, the representative can get the tag IDs of bags from airport operations database (AODB) and can request a search operation. Mobile readers can be used to identify the exact location of the missing bag by directing those readers to different location within airport.

6. Conclusions

RFID systems have been developing continuously in selected areas for decades now. It is still a potential technology which can be applied in practically all areas of daily life. Theoretically the application areas of RFID systems are unlimited. In spite of this, secure RFID searching has not gathered much attention till now. But we firmly believe that it will become very important when RFID will be deployed at a larger scale. In this paper we introduce various problems incurred while performing secure RFID tag search. Moreover, we analyzed different attack models of which tag searching is severely vulnerable. And finally we proposed secure serverless RFID tag searching protocols that can safeguard against those major attacks without server's intervention. We also discussed a couple of applications of our proposed serverless protocol in a real life scenario. We are currently working on realizing these scenarios through actual implementations. The application of our protocol is not limited to these examples only, but it can also be applied to some other real life circumstances.

7. References

[1] G. Avoine, and P. Oechslin, "A Scalable and Provably Secure Hash Based RFID Protocol", In *International Workshop on Pervasive Computing and Communication Security (PerSec '05)*, IEEE, IEEE Computer Society Press, Kauai Island, Hawaii, USA, March 2005, pp. 110-114.

[2] M.Conti, R. D. Pietro, L. V. Mancini, and A. Spognardi, "RIPP-FS: an RFID Identification, Privacy Preserving Protocol with Forward Secrecy", In *International Workshop on Pervasive Computing and Communication Security (PerSec '07)*, IEEE, IEEE Computer Society Press, New York, USA, March 2007, pp. 229-234.

[3] A. Juels, "RFID Security and Privacy: A Research Survey", RSA *Laboratories*, September 2005.

[4] M.Ohkubo,K. Suzuki, and S. Kinoshita, "Cryptographic Approach to "Privacy-Friendly" Tags", In *RFID Privacy Workshop*, MIT, MA, USA, November 2003.

[5] C. C.Tan, B. Sheng, and Q. Li, "Severless Search and Authentication Protocols for RFID", In *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '07)*, New York, USA, March 2007, pp. 3-12.

[6] G. Tsudik, "YA-TRAP: Yet another Trivial RFID Authentication Protocol", *In Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '06)*, IEEE, IEEE Computer Society Press, Pisa, Italy, March 2006, pp. 640-643.

[7] S. I. Ahamed, F. Rahman, E. Hoque, F. Kawsar, and T. Nakajima. S^{3}PR: Secure Serverless Search Protocols for RFID. *In the Proceedings of Second IEEE International Conference on Information Security and Assurance (ISA 2008)*, Busan, Korea, April 2008, pp. 187-192.

[8] Improving airport operation and security with RFID. Industry Brief, June 2005. http://www.exploit-tech.com/industries/RFID/Other/Improving airport operation and security with RFID.pdf

Authors

Endadul Hoque is a member of Ubicomp lab, Marquette University, USA. Currently, he is a graduate student of Computer Science at Marquette University, USA. He received his B.Sc degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Bangladesh in 2008. He has joined Ubicomp research lab of Marquette University in 2008. His field of interest encompasses RFID security, privacy in pervasive environment and trust model in pervasive computing, and wireless sensor network. His contact address is mhoque@mscs.mu.edu; http://www.mscs.mu.edu/~mhoque

Farzana Rahman is a member of Ubicomp lab, Marquette University, USA. Currently, she is a graduate student of Computer Science at Marquette University, USA. She received her B.Sc degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Bangladesh in 2008. She has joined Ubicomp research lab of Marquette University in 2008. Her field of interest encompasses pervasive security, RFID security, and trust models in pervasive computing. Her contact address is frahman@mscs.mu.edu; http://www.mscs.mu.edu/~frahman

Sheikh I. Ahamed is an assistant professor in the department of Math., Stat., and Computer Science at Marquette University, USA. He is a member of the IEEE, ACM, and the IEEE Computer Society. Dr. Ahamed received the B.Sc. in computer science and engineering from the Bangladesh University of Engineering and Technology, Bangladesh in 1995. He completed his Ph.D in Computer Science from Arizona State University, USA in 2003. His research interests are security in ad hoc networks, middleware for ubiquitous/pervasive computing, sensor networks, and component-based software development. He serves regularly on international conference program committees in software engineering and pervasive computing such as COMPSAC 08, PERCOM 08, SAC 08, and UIC 08. He is the Workshop Program Co-Chair of International Workshop on Security, Privacy, and Trust for Software Applications (SPTSA 08). He also directs the Ubicomp research lab (www.mscs.mu.edu/~ubicomp) in the department of Math., Stat., and Computer Science at Marquette University, USA.. Dr. Ahamed can be contacted at iq@mscs.mu.edu; http://www.mscs.mu.edu/~iq.

Fahim Kawsar is a Ph.D. candidate at the Distributed Computing Lab of Waseda University. His research interests evolve around ubiquitous computing with specific interest in smart object systems, human-centric system infrastructures and tangible interfaces. He has published in the areas of pervasive middleware, smart objects, personalization, and physical interfaces. He received his M. Engg. in Computer Science from Waseda University in 2006. He is a Microsoft Research (Asia) fellow and a student member of ACM and IEEE. Mr. Kawsar can be contacted at fahim.kawsar@gmail.com, http://www.fahim-kawsar.net.