Supporting Recovery, Privacy and Security in RFID Systems Using a Robust Authentication Protocol

Md. Endadul Hoque
MSCS Dept., Marquette University, Milwaukee, Wisconsin, USA.
mhoque@mscs.mu.edu

Farzana Rahman
MSCS Dept., Marquette University, Milwaukee, Wisconsin, USA.
frahman@mscs.mu.edu

Sheikh Iqbal Ahamed
MSCS Dept., Marquette University, Milwaukee, Wisconsin, USA.
iq@mscs.mu.edu

ABSTRACT
RFID systems have been scrutinized nowadays as one of the emerging technologies in pervasive environment. And authentication becomes indispensible in applications where security and privacy are major concerns. Besides thwarting some major attacks, RFID systems need to be able to recover from unexpected conditions during operation. In this paper, we propose a Robust Authentication Protocol (RoAP) that supports not only security and privacy, but also recovery in RFID systems. The protocol can get back the desynchronized tags and readers to their normal state, and thus provides robustness. We also present a "safety ring" consisted of six major goals that have to ensure by each RFID system to be secured. This paper illustrates security and robustness analysis of the protocol. Finally, we present the implementation of our authentication protocol.

Categories and Subject Descriptors
K.6.5 [**Security and Protection**]: *Authentication.*

General Terms
Security, Verification.

Keywords
RFID; Authentication; Recovery; Privacy; Security; Robust.

1. INTRODUCTION
RFID systems have been studied actively and frequently in pervasive computing environment as one of the technologies for last few years. The fundamental architecture of RFID technology involves a tag, a reader (or scanning device), and a back end database. A reader scans a tag (or multiple tags simultaneously) and relays the information to a database. Other than the backend database, not even a reader is able to infer any information from tag's reply as it is encrypted. Database returns tag's data to the reader only after verifying both the tag and the reader. In such a system, a malicious reader could hardly obtain precious information from tags.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SAC'09, March 8-12, 2009, Honolulu, Hawaii, U.S.A.
Copyright 2009 ACM 978-1-60558-166-8/09/03…$5.00.

RFID technology, though not very new, is finding applications in myriad fields - ranging from inventory tracking to payment systems and from prevention of pharmaceuticals stealing to e-passports. The initial developments in this field were mainly confined to some simple application such as asset tracking and supply chain management. However, as this technology started to intermingle into various other complex applications, security and privacy risks became more evident.

Security and privacy aspects should be addressed before mass deployment of RFID tags in omnipresent environment. However, conventional security primitives cannot be integrated in RFID tags as they have inadequate computation capabilities with extremely limited resources. Consequently, research community proposed several authentication protocols [2, 5, and 6] that are secured against major attack models including tracking, cloning, eavesdropping etc. One such attack is Denial of Service (DoS), the complete removal of which is almost impossible. In this attack, a tag is attacked with queries from an illegitimate reader. As a result, that tag is not able to respond to a further query from a legitimate reader. In other words, a genuine reader cannot communicate with its legitimate tags. A similar attack is also possible on the reader, but since the tag is much more resource constrained than the reader, they are more susceptible to such attacks than the readers.

DoS attack is not addressed by most of the authentication protocols because it is not possible to cope with all kinds of DoS attacks. DoS attack may also occur because of some communication failure. For example, due to the radio jamming of the channel between the tag and the reader, a communication failure may happen, which may eventually result in DoS attack. Therefore, RFID authentication protocols should at least figure out some methods to detect DoS attack and recover from such attack. Hence, authentication protocol should be designed in such a way that it can detect malicious action taken by the adversary in order to launch DoS attack and recover from them.

In this paper we propose a robust authentication protocol RoAP that is secure against most attack models including DoS. This protocol detects DoS and recovers from the attack so that the tag and the reader are not de-synchronized. Therefore, both tag and reader can communicate successfully with each other even if the adversary launches DoS attack.

The remainder of the paper is organized as follows. Security and privacy related goals are present in section 2 followed by an overview of related works in section 3. The protocol is explained in section 4. Next two sections cover the robustness analysis and security analysis of the protocol. In section 7 the protocol is

evaluated. Finally section 8 concludes the paper along with our future work.

2. SECURITY AND PRIVACY RELATED GOALS

RFID systems may face security attacks because of the proliferation of RFID tags. Several real life applications of RFID systems require them to be secure and protective against privacy attacks. Considering those example scenarios and analyzing their security requirements the following security goals can be identified. All these security goals compose the safety ring which is depicted in Figure 1. An RFID system ensuring all six elements of this safety ring is considered to be secured and protected against all major attacks. The elements of the safety ring are explained in the following.

1) Protect Privacy: RFID technology raises privacy concerns in some situations. For example consumers' privacy is hampered when the use of RFID enables different parties to obtain personally identifiable information, including location information, about particular individuals that those parties otherwise would be unauthorized to obtain. So it should be guaranteed that a tag or its secret data cannot be distinguished without tampering it and realizing all its stored data.

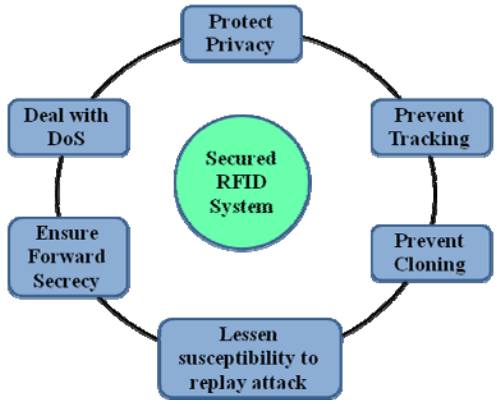

Figure 1. Six elements of "Safety Ring"

2) Prevent Tracking: Consumer community never wants to be tracked. Therefore, preventing tracking is another major goal of authentication protocol. If an adversary does not have any information about the tag then it cannot be tracked. But if the tag replies with a constant response each time it is queried then it becomes a signature of that tag. And this signature allows an adversary to track the tag. So it should be guaranteed by the protocol that a tag neither reveals its id, nor replies with a constant response.

3) Deal with Denial of Service attack: DoS attack means that an authorized entity is prevented from accessing its authorized entities. Therefore, the availability of RFID system mainly depends on the assurance of this goal. An RFID system should continue running and provide service to its authorized users even if an adversary launches DoS attack. As it is not possible to detect all kinds of DoS attack, authentication protocols should at least provide a way to deal with them. Protocols should be able to take measure against vulnerable action of the adversaries and recover from them.

4) Ensure Forward Secrecy: Forward secrecy means that if an adversary compromises a tag and learns the secret key shared between tag and reader, she will be unable to identify the previous outputs of the tag. In order to maintain RFID system security forward secrecy should be ensured by authentication protocol.

5) Lessen susceptibility to replay attack: Authentication protocols must ensure that an attacker cannot impersonate a legitimate tag by replaying an eavesdropped message.

6) Prevent Cloning: One important application of RFID systems is to detect counterfeit products. And in order to avoid counterfeiting, RFID tags need to be unclonable. An adversary can clone a tag if it knows 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.

3. RELATED WORK

The research area of RFID security is mainly divided in two categories. The first one is the protocol based category. The second category is hardware based category. Our paper falls in the first category. Within the area of protocol based category varieties of protocols have been proposed and the assortment of authentication protocols is quite extensive. Moreover [2] lists all relevant works related to the security and privacy of RFID system. But only some of them provide new and groundbreaking ideas. Therefore we shall refrain from a prevalent overview. Further interested readers may go through [5] and [6].

YA-TRAP [9] is a famous authentication protocol that places little burden on the back end server and uses monotonically increasing timestamp which makes it secure against tracking but unsecure against DoS attack. Another hash chain based RFID identification protocol is RIPP-FS [4]. Here Mauro et al. proposed that each tag shares a private symmetric key with the server. After each successful authentication, both the tag and the server update the symmetric key to maintain synchronization. RIPP-FS is resilient to a specific DoS attack where the adversary attempts to exhaust the hash chain. Another lightweight protocol is OSK [7]. Ohkubo, Suzuki and Kinoshita proposed that only two hash function is sufficient to provide indistinguishability and forward secrecy. OSK does not ensure high scalability. In [3], Avoine and Oechslin modified OSK which removed the scalability problem. Another problem of OSK is that a malicious reader may easily desynchronize a tag which results in DoS attack. Seo et al. [8] proposed a scalable and untraceable authentication protocol based on hash function. In [1], authors proposed two serverless authentication protocols. However, authentication protocol 2 is secured against all major attacks. But the major flaw of this protocol lacks recovery support.

DoS is one of the major flaws of all the protocols discussed above. At the time of communication between the tag and the reader, when a message does not reach the opposite party, two entities may become desynchronized with each other. As a result, a legitimate reader cannot communicate with its legitimate tag. So RFID authentication protocols should at least provide some means to recover from the DoS attack.

4. THE ROBUST AUTHENTICATION PROTOCOL (RoAP)

In this section, we illustrate the robust authentication protocol. Before we delve into the protocol, the respective RFID systems need to be defined.

An RFID system consists of three components: tags, readers and a backend server. Tags are wireless transponder embedded in physical objects for detection and prevention of product counterfeiting. Readers are transceivers– they can query tags for identification of objects and/or subjects. To protect privacy, one of the goals we mention earlier, all data of tags that may be privacy sensitive are stored in a tag database in the backend server. Tags contain a limited amount of data to prove itself legitimate. Readers not only interact with tags but also communicate with backend server while identifying tags. For simplicity, we presume a reader and the backend server to be a single entity and refer it as a reader.

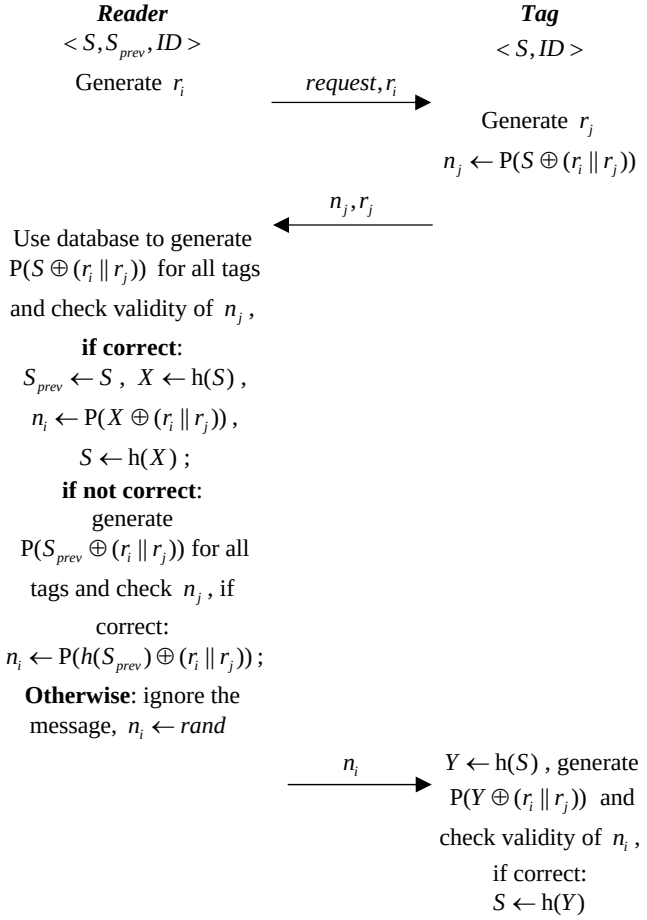

Figure 2. The Robust Authentication Protocol

Each tag contains a 2-tuple consisting of a secret number *S* and an identifier *ID*. Tag gets the data from the reader (in fact, the backend server) at the time of deployment. On the other hand, for each tag, the reader has a 3-tuple composed of the secret number *S*, the secret number of the last successful session S_{prev}, and the tag identifier *ID*. In reader side, a tag *ID* points to all the data associated with the respective tag. All the entities of the system can generate pseudorandom number by generator P(.) based on its seed. Initially, the data of tag and reader are in sync, and S_{prev} equals *S*.

The protocol operates as shown in Figure 2. At first, the reader sends a request accompanied by a random number r_i. Upon receiving the request, the tag computes n_j with another random number r_j generated by itself. The tag replies with n_j for authenticating itself, and r_j to help the reader to produce the same pseudorandom number. Now, the reader checks the validity of n_j by computing $\mathrm{P}(S \oplus (r_i \| r_j))$ for each tag in the database. If the reader finds a match, it can be sure of the validity of the tag. Then the reader updates S_{prev} and n_i is generated by using the next seed that is the hashed secret number h(*S*). If the reader fails to find any match in the first search strategy, it changes the scheme of search by replacing the *S* with the S_{prev} of all the tags in the database. Upon realizing any match, the reader only generates n_i. In fact, this step is to provide the robustness to the protocol by recovering any tag from out of order to in sync with the reader. In both the cases, the reader replies with n_i. If n_j is not valid, the reader simply ignores the message and replies with a random number *rand*. However, this *rand* keeps the protocol consistent by preventing an eavesdropper to acquire any knowledge about this session.

Finally, it is tag's turn to authenticate the reader by verifying n_i. If n_i is valid, the tag updates its secret number accordingly. Otherwise the tag discards the message.

5. ROBUSTNESS ANALYSIS OF RoAP

In this section, we present a detailed example to explain how RoAP exemplifies robustness– how RoAP can recover RFID systems. After describing a successful tag query, we illustrate how a tag is recovered if a message loss (due to communication failure, message intercept, etc.) was happened in earlier interaction. In our example, to make analysis simple, we demonstrate the interaction between a single tag and the reader.

Reader	***Tag***
$<S_0, S_0, ID>$	$<S_0, ID>$

Figure 3. Initial state

Initially, the reader and the tag are both in sync, as shown in Figure 3. Now tag replies to authenticate itself upon receiving the request from the reader. By being a valid tag, the reader finds a match with the entry in the tag database. Now it's time to authenticate reader and it does so. After completing the updates, the reader proves its validity to the tag. The subsequent state after this successful interaction is depicted in Figure 4.

Reader	***Tag***
$<S_2, S_0, ID>$	$<S_2, ID>$

Figure 4. State after the successful interaction

A tag can be out of order in final step of RoAP where the tag waits for n_i from the reader. Suppose the aforementioned tag again interacts with the reader after a while. Every message but the final one, for instance, is effectively received. The very last message containing n_i is damaged or lost (due to communication failure, message modification or intercept). Since the tag does not update its secret numbers, it becomes desynchronized with the reader. The internal state of the reader and the tag after this unexpected situation is shown in Figure 5.

Reader	*Tag*
$<S_4,S_2,ID>$	$<S_2,ID>$

Figure 5. State after the message lost

Now if this tag again comes to vicinity of the reader, the tag starts interaction with the reader. However, the tag still has $<S_2,ID>$ as its internal data. Now, the reader fails to find any match with the received response as it tries to validate with S_4's. The reader continues the search with the previous secret numbers, S_2's. After a fruitful search, the reader comes across the validity of the tag. To synchronize both the entities again, the reader takes a prominent step by sending the valid message devoid of any update in its database. When the tag receives this message, it verifies the originality of the reader and updates internal data as well. Thus the robust protocol recovers the system from out of order state. After recovery the state of internal data is shown in Figure 6.

Reader	*Tag*
$<S_4,S_2,ID>$	$<S_4,ID>$

Figure 6. State after recovery

6. SECURITY ANALYSIS OF RoAP

6.1 The Adversary and Attack Model

The major goal of an adversary in any RFID system is to counterfeit a real tag such that it can only be distinguished from the real one with small probability. Evidently, the fake tag embedded within the fake product can let the product to be identified as a legitimate one. In RoAP an adversary is denoted as $\tilde{A}$. The adversary can control a number of readers and tags. Each reader and tag controlled by the adversary are denoted as $\tilde{R}$ and $\tilde{T}$, respectively. $\tilde{R}$ is unauthorized to have access to any real tags as it is not connected with the backend server. Similarly, $\tilde{T}$ is not valid as it has no idea about *S* and *ID*. We presume that the backend server cannot be compromised because the adversary would get total control over the tag database then. Moreover we assume that all the entities such as tags, readers, adversaries, adversarial tags and adversarial readers have polynomially bounded resources.

We assume that $\tilde{A}$ is simply more powerful than a passive attacker. Like a passive attacker it can eavesdrop on the channel between a valid reader and a valid tag. However, like an active attacker, $\tilde{A}$ can install a rouge reader $\tilde{R}$ that can communicate with a valid tag. In addition, $\tilde{A}$ can install a fake tag $\tilde{T}$ to communicate with a legitimate reader. In both cases the ultimate goal of the adversary is to counterfeit a tag with the learned information. Despite of these attacks, $\tilde{A}$ can launch hardware based physical attacks. But we will not study such attacks as hardware based physical attacks are beyond the scope of this paper.

6.2 Detailed Security Analysis

In section 2, we have already mentioned the six elements of safety ring which must be ensured by an authentication protocol in order to keep an RFID system secured and protected. In this section, we explain how RoAP defends the RFID system against those six major attacks and keeps the system within the safety ring.

1) Privacy Protection: Users carrying various tagged items do not want to hamper their privacy. If an adversary comes by any private information of the tag, by querying or eavesdropping, it may cause several vulnerabilities to owner's day to day life. Our protocol protects users' privacy strongly. According to RoAP, a tag never sends its own *ID* to anyone, not even to the authorized reader. The tag sends its responses in disguise so that only an authorized reader can identify the tag.

2) Prevent Tracking: If RFID tags reply with a constant response each time it is queried, it becomes a signature for that particular tag. So it may potential to establish a link between the tag and the owner of the tagged object which leads to tracking. In order to prevent clandestine physical tracking, each entity's response must be scrambled. Our protocol is secured against tracking attack. In RoAP, each entity never replies with a constant response as random number is involved within each computation.

3) Prevent Cloning: To launch this attack, an active adversary queries a real tag and obtains its response. By placing this response in a fake tag $\tilde{T}$, the adversary $\tilde{A}$ attempts to counterfeit the real tag. Now, attacker $\tilde{A}$ becomes successful in her attempts if she can deceive a legitimate reader. In other words, the real reader fails to distinguish the genuine tag from the fake one. According to RoAP, whenever an adversary $\tilde{A}$ queries a real tag, $\tilde{A}$ receives a distinct response each time because of the inclusion of random numbers. Thus RoAP thwarts tag counterfeiting.

4) Ensure Forward Secrecy: Forward secrecy means that an adversary will not be able to realize any previous output transmitted by the entity even if it compromises that entity. RoAP ensures forward secrecy. The secret number *S*, shared between the tag and the reader, is updated each time using irreversible one way hash function. After compromising a valid entity, $\tilde{A}$ cannot realize earlier responses based on the former secret numbers as it cannot derive the former secret numbers from the current one.

5) Lessen Susceptibility to Replay Attack: In order to launch this attack, the adversary eavesdrops on both the communication channel between the tag and the reader. Thus $\tilde{A}$ can learn the challenges-responses between a legitimate tag and the legitimate reader, and later uses these data to create a fake tag (reader) in order to deceive an honest reader (tag). But in order to deceive a legitimate reader (tag), the fake tag $\tilde{T}$ (fake reader $\tilde{R}$) has to generate valid response. However, this is impossible in our protocol as two distinct random numbers are involved in each interaction. Therefore RoAP is not susceptible to replay attack.

6) Deal with Denial of Service: In this attack, the adversary wants neither to derive any information nor to impersonate a tag or a reader. Rather her main target is to ensure that a valid reader cannot access its authorized tags. To launch a DoS attack, $\tilde{A}$ can adopt several means. Though it is not possible to cope with denial of service due to all possible ways, we focus some of those that RoAP can prevent. Message intercept may cause DoS. This problem exacerbates when the backend server and the tag shares a secret key that has to be synchronized after each regular query. Even distorted or damaged message may launch DoS. Certainly, RoAP is susceptible to above mentioned means. However, even

after being desynchronized, the protocol can recover the RFID system to the normal state.

7. EVALUATION

To evaluate our protocol we are implementing a prototype on Pocket PC with RFID Flash card reader. Some of screenshot are presented in Figure 7. Screen 1 depicts the interface of the prototype. When connect button is clicked, the reader gets connected with the backend server, as shown in screen 2. A tag is identified by the reader in screen 3. In screen 4, we purposely click desynchronize button to make the tag out of sync. It shows that the reader performs all the update on database, but sends a random number instead of the valid pseudorandom number. After clearing the textbox, the reader again scans and discovers the desynchronized tag. Screen 5 displays a message when the reader recovers the tag from being out of order. Finally screen 6 present that the valid message to authenticate the reader is sent without any further update on database.

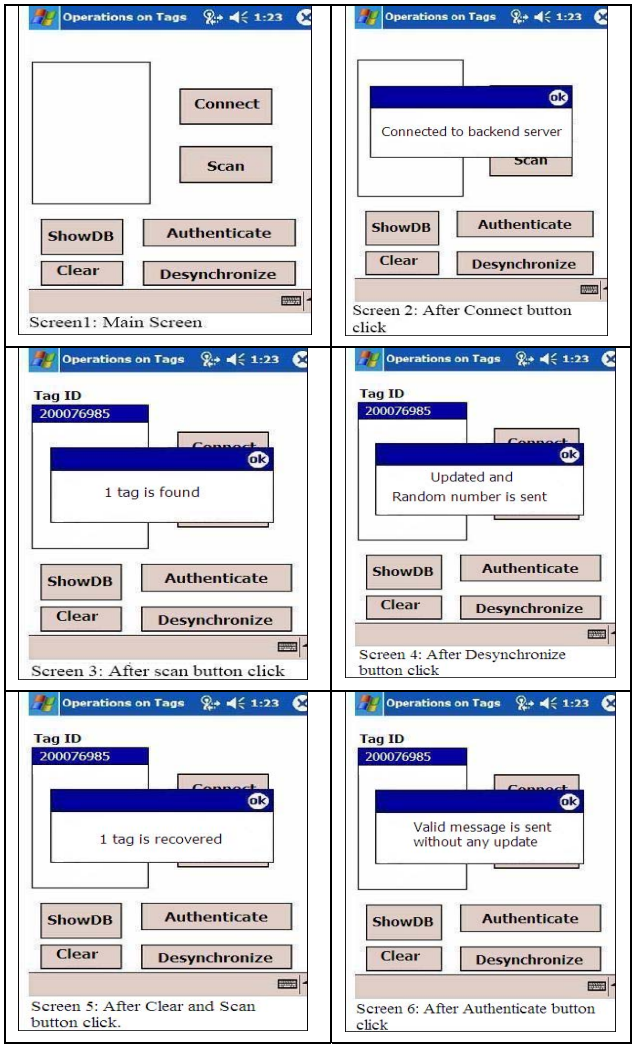

Figure 7. Screenshots of RoAP operation

8. CONCLUSION AND FUTURE WORKS

Prevalent deployment of RFID systems depend on the strength of security against major attacks, protection of private data, and recovery from unanticipated circumstances during operation. Each of the elements of the "Safety Ring" has to be ensured to keep the system secured. In order to cope with these demands, we present a robust authentication protocol (RoAP) in this paper. How RoAP recovers the system is presented in robustness analysis. In addition, security analysis establishes that RoAP keeps the system secured by ensuring the safety ring. Some screenshots of our implemented prototype demonstrate how a tag is recovered in the system.

In the future, we plan to simulate the protocol with a large number of tags to see how it performs. Study of other issues of DoS and making the more robust are other future research issues.

9. REFERENCES

[1] Ahamed, S. I., Rahman. F., Hoque, E., Kawsar, F., and Nakajima, T. YA-SRAP: Yet Another Serverless RFID Authentication Protocol. In *the 4th IET International Conference on Intelligent Environment (IE08)*, Seattle, USA, Jul 2008.

[2] Avoine., G. Security and Privacy in RFID Systems. http://www.avoine.net/rfid/, 2008.

[3] Avoine, G. and Oechslin, P. 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, pp. 110–114, Kauai Island, Hawaii, USA, March 2005.

[4] Conti, M., Pietro, R. D., Mancini, L. V., and Spognardi. A. 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, pp. 229-234, New York, USA, Mar. 2007.

[5] Juels, A. RFID Security and Privacy: A Research Survey. Manuscript, Sep. 2005.

[6] Juels, A. and Weis, S. Defining Strong Privacy for RFID. *Cryptology ePrint Archive, Report 2006/137, IACR*, Apr. 2006.

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

[8] Seo, Y., and Kim, K. Scalable and Untraceable Authentication Protocol for RFID. In *International Workshop on Security in Ubiquitous Computing Systems (Secubiq '06)*, Springer-Verlag , Seoul, Korea, August 2006.

[9] Tsudik, G. YA-TRAP: Yet another Trivial RFID Authentication Protocol. In *International Conference on Pervasive Computing and Communication (PerCom '06)*, IEEE, IEEE Computer Society Press, Pisa, Italy, Mar. 2006.