

Mutual Authentication Protocol for RFID tags based on Synchronized Secret Information with Monitor

Song Han, Vidyasagar Potdar, and Elizabeth Chang

DEBII Institute
Curtin University of Technology
GPO Box U1987, WA
s.han@curtin.edu.au

Abstract. RFID, as an anti-counterfeiting technology, is pushing huge potential industrial, medical, business and social applications. RFID-based identification is an example of emerging technology which requires authentication. In this paper, we will propose a new mutual authentication protocol for RFID tags. The RFID reader and tag will carry out the authentication based on their synchronized secret information. The synchronized secret information will be monitored by a component of the database server. Our protocol also supports the low-cost non-volatile memory of RFID tags. This is desirable since non-volatile memory is an expensive unit in RFID tags.

1 Introduction

Radio frequency identification (RFID) is a method of remotely storing and retrieving data using small devices called RFID tags. RFID system could lower the cost of tagged items in the process of identification [9], [16]. Therefore, RFID technology is expected to diffuse into our everyday life. In a server-based scenario, an RFID system is composed of three players: the database server, the RFID tag(s), and the RFID reader(s).

Tag: is comprised of an IC chip and antenna, and sends information to the RFID reader in response to a wireless probe.

Reader: is a device that transmits a radio frequency probe signal to RFID tag, receives the information sent by tag, and sends the information to the database server.

Database Server: is a secure server that has a database and manages various types of information related to each tag, e.g., ID, reader location, read time, and temperature of sensor. The server resolves the ID of tag from the information sent by tag through authenticated RFIDreader.

The RFID reader makes an access request to the RFID tag and then returns the reply from the RFID tag to the database server. The identification and authentication are carried out under the help of the database server. After the RFID tag and reader authenticate each other, the database server returns the information of the RFID tag to

the reader. Generally, most works assume the communication channel between the database server and the RFID reader is secure.

In this paper, we will design an RFID authentication protocol for RFID tags and readers. This protocol is based on synchronized secret information shared between the database server and the RFID tag. Our method is based on the framework of the previous works [1], [2], [5], [7], [11], [12], and [14].

1.1 RFID Security and Privacy Issues

The central issue the RFID technology raised is the privacy issue [7], [8], [10], [13], [15] as well as the cost [3], [4], [12]. In this paper we will focus on the privacy issue and RFID tag authentication. In fact, the mass deployment and acceptance of RFID technology is nowadays mainly limited by privacy concerns [9]. Products labeled with RFID tags reveal sensitive information when queried by readers, and they do it indiscriminately. This may induce the violation of location privacy, i.e. tracking. Two security requirements related to privacy are:

Indistinguishability: Tag output must be indistinguishable from truly random values. Moreover, they should be unlinkable to ID of the tag. If the adversary can distinguish that a particular output is from a target tag, he can trace the tag.

Forward security: Even if the adversary acquires the secret tag data stored in the tag, he cannot trace the data back through past events in which the tag was involved. Needless to say, the adversary who only eavesdrops on the tag output, cannot associate the current output with past output.

Besides the privacy and tracking issues, there are some other worth mentioning: impersonating, spoofing, eavesdropping, traffic analysis, etc.. In order to address the cost issue, our paper will design an authentication protocol which can reduce the cost of RFID tags through trimming down the non-volatile memory. This is similar to the protocols in [11] and [14].

1.2 Summary of the Attributes of the Proposed Protocol

In this paper, we will propose a new mutual authentication protocol for RFID tags. Our method is based on one-way hash function. Our protocol has the following attributes:

- The protocol is based on synchronized secret information.
- The synchronized secret information is monitored by an internal configuration of the database server. Thus, desynchronization attack can be detected and diminished.
- Hash chain requirement is not necessary in our protocol.
- The memory of the RFID tag is portioned into two parts: volatile memory and non-volatile memory.
- The structure of the tag memory can reduce the cost of RFID tags.
- The protocol supports user privacy protection.
- The protocol supports forward security and untracking property.

1.3 Organization of the rest of the Paper

The remainder of our paper is organized as follows: In Section 2, some related works will be introduced. In Section 3, some notations will be first presented; then some basic assumptions related to RFID tags and readers as well as database server will be provided; following that, the mutual authentication protocol based on synchronized secret information with monitor will be proposed; thereafter, the structure of the database server will be explained. In Section 4, the security analysis and comparison will be provided. In Section 5, the efficiency and its comparison will be investigated. Finally, we will conclude our paper.

2 Related Works

The authentication protocol we will propose in this paper is a hash-based authentication method. Therefore, the related works introduced here focus on the hash-based authentication protocols for RFID tags.

Ohkubo et al. [2] proposed a hash-based authentication protocol. The protocol changes RFID identities on each read based on hash chains. Their method also requires a hash function on the RFID tag, but does not require a random number generator. However, the protocol in [2] is vulnerable to certain re-play attacks.

The proposed modification in [1] address the replay attack problem but does not consider the issue of availability, and their protocol is vulnerable to attacks where the attacker forces an honest tag to fall out of synchronization with the server so that it can no longer authenticate itself successfully.

The scheme by Jules [3] only provides security against “fly-by” attacks where the attacker is allowed to interact with the tag for a fixed time budget but does not provide protection in the case of tag capture.

Molnar et al. [7] proposed a hash-tree based authentication protocol for RFID tags. They exposed privacy issues related to RFID in libraries, described current deployments, and suggested novel architectures for library RFID. However, the amount of computation required per tag is not constant, but logarithmic with the number of tags in the hash-tree. Also, if a tag is lost, anonymity for the rest of the hash-tree group may be compromised. Finally, the protocol does not provide for forward-anonymity.

Another hash-based authentication protocol was introduced by Henrici et al. [5]. Their solution does not provide full privacy guarantees, in particular, the tag is vulnerable to tracing when the attacker interrupts the authentication protocol mid-way.

Lee et al. [11] proposed an RFID authentication scheme which uses a hash function and and synchronized secret information. However, the RFID tag needs a PRNG. Moreover, the server does not know how many times an RFID tag may have not yet

updated its secret information. Thus, desynchronization attacks may be conducted by adversaries.

Dimitriou [12] also proposed an anonymous RFID protocol against tag cloing. Their protocol is based on the use of a secret shared between tag and database that is refreshed to avoid tag tracing. However, their protocol is vulnerable to desynchronization attacks against availability.

Lee et al. [14] proposed a protocol to solve the desynchronization problem by maintaining a previous ID in the database server. However, an adversary who queries T actively without updating ID can trace the RFID tag because the hashed ID is continually identical. Moreover, the adversary can trace the previous event of tag because ID is updated by XORing the previous ID with a random number emitted through radio frequency (RF).

3 Mutual Authentication Protocol with Monitor

In this section, we will develop the mutual authentication protocol for RFID tags based on synchronized secret information. Some notations will be first introduced and then used throughout the rest of our paper. Next, we will give the basic assumptions that related to what capability the RFID tags and readers as well as the database server have. Thirdly, we will present the new authentication protocol for RFID tags. Following that, we will explain the structure of the database of the server in detail. This is important since some characteristics of the structure can detect some adversaries that may compromise or prevent RFID tags from updating their secret elements. This point will be demonstrated in the Section 4.

3.1 Notations Used in Our Protocol

The following table provides the notations used in the proposed authentication protocol.

Table 1. Notations used in the proposed RFID tag and reader authentication protocol

Symbol	Representation
$h()$	One hash function available to all parties
T	An RFID tag
S	RFID system database server
R	An RFID reader
i, j	Non-negative integers

M	Monitor for checking RFID tag whether updates its secret element after a successful authentication
ID	RFID identification number
PRNG	A pseudorandom number generator
β	The bit-length of the output of $h()$
\oplus	Exclusive-OR function (XOR)
θ	A threshold for M
α	A seed for PRNG
VM	Volatile memory of RFID tag
NVM	Non-volatile memory of RFID tag

3.2 Basic Assumptions

The RFID tag T has a one-way hash function $h()$. T can calculate the hash value for any input to this function. T can also carry out the XOR calculation. In fact, carrying out the XOR calculation is an affordable capability for various RFID tags, especially for low-cost RFID tags. In our protocol, the RFID tag does not need a pseudorandom number generator PRNG. This will further reduce the cost of the RFID tags. Generally, the more power of calculating the RFID tag has, the more cost the RFID tag induces. Therefore, in our authentication protocol, the most expensive capability of the RFID tag needs is to carry out the hash function evaluation and the XOR calculation. For each RFID tag in our system, it has two memories: one is volatile memory (VM) with bit-length β , the other is non-volatile memory (NVM) with the same bit-length β . The volatile memory is used to store metadata in an RFID tag and reader authentication process. Here the metadata stored in the volatile memory of the RFID tag represents the data the tag T calculated by itself and received from reader R. The metadata stored in the VM will be automatically deleted as soon as the authentication process is fulfilled. In fact, some metadata only stays in the VM for a temporal state even before the authentication process is fulfilled. The data stored in the NVM is the synchronized secret information whose bit-length is β . For initial value of the secret element k , it is shared between the database server S and the RFID tag T. In fact, it is embedded in T when T was registered to the database server S by its business owner. After a successful authentication process, T will update k with $h_{k \oplus r}(k \oplus r)$ where r is a random element generated by R's pseudorandom number generator. For any $i > 2$, if the synchronized secret information in NVM is k_i after the successful i -th authentication process, then after the $(i+1)$ -th successful authentication process the synchronized secret information in NVM is $k_{i+1} = h_{k_i \oplus r_{i+1}}(k_i \oplus r_{i+1})$ where r_{i+1} is a random number generated by R's pseudorandom number generator PRNG.

In our authentication protocol, we set up a monitor M in the database server S. M is used to scrutinize whether the RFID tag T updated its secret information after interacting with an authorized RFID reader R. The detail of the structure for the database server will be introduced in subsection 3.4. The communication channel between

the RFID tag T and reader R is insecure while the channel between R and the database server S is secure.

3.3 The Proposed Protocol

The mutual authentication protocol for RFID tags based on synchronized secret information with monitor is summarized in the following figure.

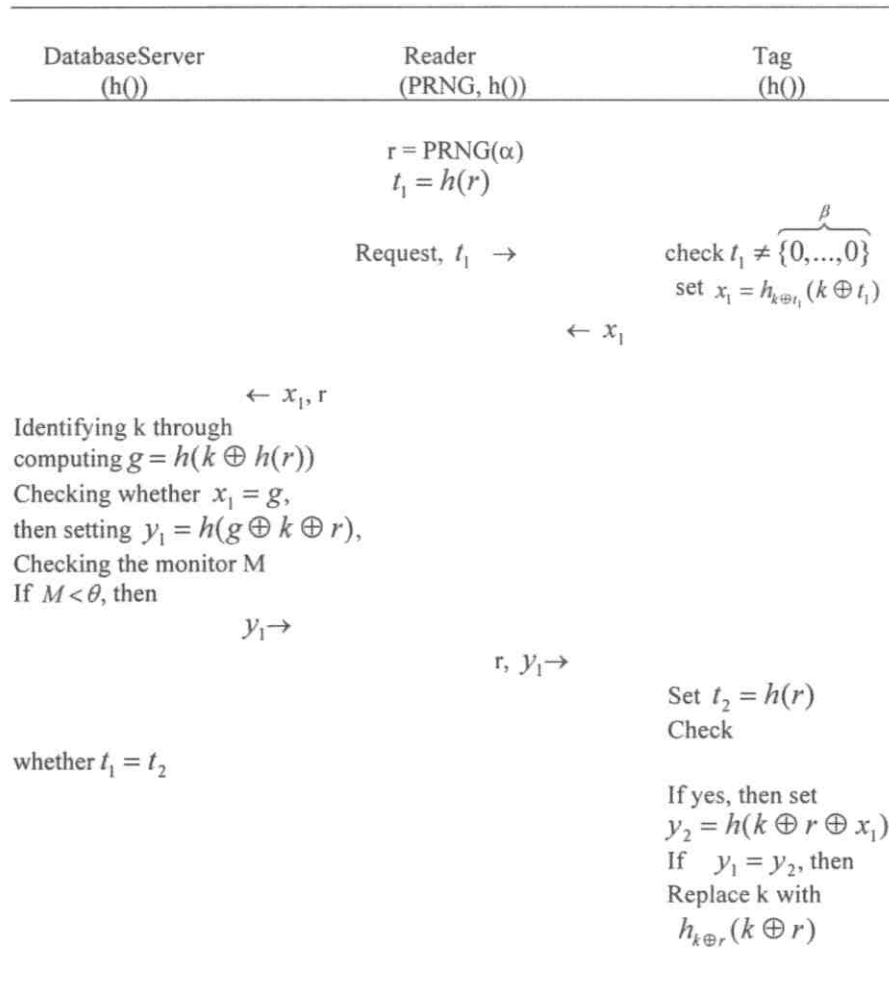


Fig. 1. Mutual authentication protocol for RFID tags based on synchronized secret information with monitor. In this figure, β is the bit-length of the output of the hash function $h()$. After the successful mutual authentication, the RFID tag will delete the old private element k and update the non-volatile memory using the new private element $h_{k \oplus r}(k \oplus r)$, where the hash value is the keyed hash evaluation with $k \oplus r$.

The details of the mutual authentication protocol are presented as:

1. RFID reader R uses the PRNG and a seed α to generate a random number r , then uses the hash function $h()$ to get $t_1 = h(r)$. R then sends Request and t_1 to RFID tag T.
2. T first checks $t_1 \neq \overbrace{\{0, \dots, 0\}}^{\beta}$. If yes, then T uses its secret element k and the received t_1 to get $x_1 = h_{k \oplus t_1}(k \oplus t_1)$. T then sends x_1 back to R.
3. After receiving x_1 , R sends x_1 and its random number r to the database server S.
4. S first uses hash function $h()$ to get $h(r)$, then identify ID of tag T through (1) finding a value k in the database to compute $g = h(k \oplus h(r))$; (2) checking whether $x_1 = g$. If there is only value k after the process, then the identification ID in the list of k is the ID of the tag T.
5. S then computes $y_1 = h(x_1 \oplus k \oplus r)$ using x_1 , k and r . Before sending y_1 back to R, S checks whether the monitor M in the list of k is less than θ . If yes, then S sends y_1 back to R. Otherwise, S will inform R that the tag T may be prevented from updating its secret information k by adversaries. An order will be instantly sent to the system administrator (e.g. the business owner) to inspect the tag.
6. After receiving y_1 , R forwards y_1 and its random element r to T.
7. T first sets $t_2 = h(r)$ and then checks $t_1 = t_2$. If they are not equal, then T stops here; otherwise, T sets $y_2 = h(k \oplus r \oplus x_1)$ and compares $y_1 = y_2$. If yes, then T replaces k with $h_{k \oplus r}(k \oplus r)$.
8. For the database server S, if k is found in K_{OLD} , then the monitor M will be updated as $M+1$ while the secret information k will keep stable for this authentication process. If k is found in K_{NEW} , then the monitor M will not be updated while the secret information k in K_{NEW} will be replaced with $h_{k \oplus r}(k \oplus r)$ and the value of k will be copied to the field K_{OLD} .

3.4 The Structure of the Database Server

The database server plays an important role in the mutual authentication protocol between the RFID tag and reader. One the one hand, the database server maintains the identification information for each registered RFID tag and stores some private

elements which can be used to authenticate RFID reader to tag. On the other hand, the database server maintains a secure channel for authorized RFID readers. This enables to authenticate RFID tag to reader. For each registered RFID tag in the system, the database server needs to maintain a list. The list is composed of four components: the identification information part (ID), the old secret information part (K_{OLD}), the new secret information part (K_{NEW}), and the authentication monitor part (M). Here M is an integer that will augment if the secret element k is found in K_{OLD} ; otherwise, it will keep stable. That is,

$$M = \begin{cases} M + 1, & \text{if } k \text{ is found in } K_{OLD} \\ M, & \text{if } k \text{ is found in } K_{NEW} \end{cases}$$

There is also a threshold θ for M. If for the (i+1)-th authentication process, the database server identifies that $M \geq \theta$,

then the server will inform the RFID reader of the the (i+1)-th authentication process that the tag (that the reader interacts with) may be compromised or prevented from updating the secret information k by malicious parties. The initial value of M is 0.

The internal components of the database server in the (i+3)-th authentication process for tag T

ID	K_{OLD}	K_{NEW}	M
...
ID_T	k	$h_{ver}(k \oplus r)$	0
...
...
