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-conterfeiting 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 reminder 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 b 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.

DatabaseServer (h())	Reader (PRNG, h())	Tag (h())
	r = PRNG(α) $t_1 = h(r)$	
	Request, t_1 →	check $t_1 \neq \overbrace{\{0,...,0\}}^{\beta}$ set $x_1 = h_{k \oplus t_1}(k \oplus t_1)$
		← x_1
← x_1, r		
Identifying k through computing $g = h(k \oplus h(r))$ Checking whether $x_1 = g$, then setting $y_1 = h(g \oplus k \oplus r)$, Checking the monitor M If $M < \theta$, then		
y_1→		
	r, y_1→	
		Set $t_2 = h(r)$ Check
whether $t_1 = t_2$		
		If yes, then set $y_2 = h(k \oplus r \oplus x_1)$ If $y_1 = y_2$, then Replace k with $h_{k \oplus r}(k \oplus r)$

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 $t_1 \neq \overbrace{\{0,...,0\}}^{\beta}$. 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	$h_{k\oplus r}(k \oplus r)$	0
...	...	...	...
...	...	...	...

Fig. 2. The structure of the database server. We use one RFID tag instance T to demonstrate the components of the field of the database server. ID_T is the unique identification number of T. ID_T keeps stable in any authentication process. Suppose in the (i+1)-th authentication process the synchronized secret element of T is k, then for the (i+2)-th authentication process the secret element in K_{OLD} is k while in K_{NEW} the secret element is $h_{k\oplus r}(k \oplus r)$ after a successful authentication process. Here r is a random element that was generated by an authorized RFID reader involving the authentication process.

Remark 1. We suppose the first i+2 times authentication process are all successfully finished. Then we can set the M value for ID_T as 0 in the above table. Otherwise, the M value will be j (where $j<\theta$). This implies that the RFID tag T has not updated its secret element k for j times.

4 Security Analysis and Comparison

This section will provide the security analysis and a comparison of security characteristics between our protocol and some previous protocols.

4.1 Security Analysis

Attack on user privacy: From the construction of our protocol, we can see that no identity was released by an RFID tag participating in the authentication process. Therefore, eavesdropping is not an issue as long as the hash function h() is one-way hash function so that no usable information on its pre-image is revealed. As a result, user privacy is guaranteed.

Attack on the RFID tag: If an adversary tries to impersonate an authorized RFID reader and attack on the RFID tag, then the adversary does not know the secret information k. As a result, the adversary cannot compute g and y_1. Therefore, the adversary cannot let the RFID tag accept its access. On the other hand, the communication channel between S and R is secure. This prevents the adversary from interacting with the database server S.

Mutual authentication: The authentication between R and T is mutual authentication. R is authenticated to T by checking whether $t_1 = t_2$ and $y_1 = y_2$. T is authenticated to R by checking whether $x_1 = g$. This is actually done by the database server.

Forward security and untraceable property: If an adversary obtains the secret information k by compromising the RFID tag, then the adversary cannot trace back the previous authentication the tag involved. This is because for each authentication run, the RFID reader generates a random number r. And the secret information k is mixed and

hashed with r. Therefore, forward security as well as untraceable property is preserved for successive and fulfilled authentication runs.

Anti-cloning: If an adversary tries to make cloning by replaying intercepted interactions, then the adversary cannot success cloning because r is different for each authentication run. On the other hand, the adversary cannot work out a valid x_1 without knowing the secret information k.

In order to compare the security characteristics in different RFID authentication protocols, we give the following definition:

Definition 1. **DeSynchronization Detection Security:** In the proposed RFID tag authentication protocol, there is a monitor M in the database field which can inform the RFID reader that the targeted RFID tag may be prevented from updating its secret information by adversaries if $M \geq \theta$. We name this characteristic as the **DeSynchronization Detection Security (DDS)**.

4.2 Security Characteristic Comparison

By the comparison studied in the following table, we can know that the protocol in our paper offers the most enhanced security in RFID mutual authentication protocols [5], [7], [11], [12], [14] and our paper.

Table 2: Security characteristic comparison

Protocol / Characteristic	[7]	Our protocol	[11]	[14]	[5]	[12]
Forward security	No	Partially satisfy	Partially satisfy	No	Partially satisfy	Partially satisfy
Mutual authentication	Yes	Yes	Yes	Yes	Yes	Yes
DDS	Not satisfy	Satisfy	Not satisfy	Not satisfy	Not satisfy	Not satisfy
Indistinguishable	Satisfy	Satisfy	Satisfy	Partially satisfy	Not satisfy	Not satisfy
Anti-cloning	Satisfy	Satisfy	Satisfy	Satisfy	Satisfy	Satisfy

5 Efficiency and Comparison

In this section, we will measure the computational efficiency of tag T. Following that, we will then compare our protocol with the existing protocols from the computational efficiency point of view.

5.1 Efficiency of Tag

The RFID tag in our protocol does not need the pseudorandom number generator PRNG. For a successful authentication process, the RFID tag needs four hash operations. However, our protocol does not need the strong assumption Hash Chain Requirement. The communication complexity and the non-volatile memory of our protocol are identical to those of [11].

5.2 Efficiency Comparison

In order to compare the efficiency of RFID tag authentication of our protocol with previous protocols, we derive a definition for hash chain requirement from a previous protocol in [11].

Definition 2. **Hash Chain Requirement:** Let s be the total number of RFID tags in an RFID system. Let t be the maximum number of authentication times of each RFID tag T. H() is a one-way hash function. The hash chain can be constructed: $H_1 = H(K_0)$, where K_0 is the a secret element; the second value in the hash chain is $H_2 = H(H_1)$; for any $3 \leq j \leq st$, the j-th value in the hash chain is $H_j = H(H_{j-1})$. If the secret element K_0 needs be chosen such that the hash chain length is longer than st, then we say the hash function H() together with K_0 satisfy the *Hash Chain Requirement.*

By this definition, it is easy to see that the authentication protocol in [11] employed a one-way hash function and an initial secret element which need to meet the Hash Chain Requirement. On the contrary, our protocol does not need to use a hash function which meets the strong assumption. This is because in our authentication protocol the secret element has been mixed with a different random number once whenever an authorized RFID reader interacts with the tag. Therefore, in the hash chain, the preimage of the current hash value is not the previous hash value adjunct to the current hash value.

Table 3: Efficiency in RFID tag

Protocol / Efficiency	[14]	Our protocol	[11]	[7]
PRNG operation		0	1	1

Hash Operation	2	4	3	2
Communication complexity	3γ	4γ	4γ	4γ
Non-volatile memory	2γ	γ	γ	2γ
Hash chain requirement	–	–	+	–

Remark 2. In the above table, we use the similar measure as that of [11]. γ denotes the number of bits of a data unit. In the last row of the table, '+' represents the protocol in [11] should meet the Hash Chain Requirement; '-' represents our protocol is not restricted by the requirement. In other words, the hash function and the initial secret information in our protocol do not need to satisfy the Hash Chain Requirement.

From the above table, we can see that the protocol [11] and our protocol induces more hash operations than the protocols in [7] and [14]. The non-volatile memory cost in our protocol and [11] is identical. From the above table, both of the protocol [11] and our protocol require the non-volatile memory (50% off) less than the protocols in [7] and [14]. Notice that the non-volatile memory is a comparative expensive unit in RFID tags. Therefore, the protocols of [11] and ours have economic effective merit. On the other hand, although our protocol needs one more hash operation than that of [11], the RFID tag in our protocol does not need a PRNG and the strong assumption-Hash Chain Requirement. Therefore, based on the above comparison, our protocol is better desirable than the protocols [11].

6 Conclusions

In this paper, we proposed a mutual authentication protocol for RFID tags based on synchronized secret information with a monitor. Some characteristics are summarized as follows:

- The authentication between tag and reader is mutual authentication.
- The memory of the tag is portioned into two parts: volatile memory and non-volatile memory. Thus, it reduces the cost of tags.
- The synchronized secret information between the tag and the database is monitored so that some attacks can be detected.
- The tag does not need a pseudorandom number generator.
- The synchronized secret information is masked with a random number. Thus, the hash chain requirement is on the rampage.

References

1. Avoine, G., Oechslin, P.: A Scalable and Provably Secure Hash Based RFID Protocol. The 2nd IEEE International Workshop on Pervasive Computing and Communication Security, Kauai Island, Hawaii, USA, March 8th, (2005) 110–114
2. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Approach to Privacy-friendly Tags. in *RFID Privacy Workshop*, MIT, 2003.
3. Juels, A.: Minimalist Cryptography for RFID Tags. In C. Blundo, ed., *Security of Communication Networks (SCN)*, 2004.
4. Vajda, I., Butty´an, L.: Lightweight authentication protocols for low-cost RFID tags. *2nd Workshop on Security in Ubiquitous Computing*, 2003.
5. Henrici, D., M¨uller, P.: Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers. Workshop on Pervasive Computing and Communications Security, 2004.
6. Avoine, G., Oechslin, P.: RFID Traceability: A Multilayer Problem. *Financial Cryptography*, 2005
7. Molnar, D, Wagner, D.: Privacy and Security in Library RFID Issues, Practices, and Architectures. ACM *Conference on Computer and Communication Security*, 2004.
8. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong Authentication for RFID Systems Using the AES Algorithm. International *Workshop on CryptographicHardware and Embedded Systems (CHES)*, 2004.
9. Sean, W., Thomas, L.: Automatic identification and data collection technologies in the transportation industry: Barcode and RFID. Technical Report, 2001.
10. Sarma, S. E., Weis, S. A., Engels, D. W.: RFID systems, security and privacy implications. Technicalreport MIT-AUTOID-WH-014, AutoID center, MIT, 2002
11. Lee, S., Lee, H., Asano, T., Kim, K.: Enhanced RFID Mutual Authentication Scheme based on Synchronized Secret Information. AUTO-ID Labs White Paper, WP-HARDWARE-032, 2006.
12. Dimitriou, T.: A Lightweight RFID Protocol to Protect against Traceability and Cloning Attacks. International Conference on Security and Privacy for Emerging Areas in Communication Networks- SecComm, September 2005.
13. Jules, A., Rivest, R., Szydlo, M.: The Blocker Tag: Selective Blocking of RFID tags for Consumer Privacy. ACM Conference on Computer and Communication Security – ACM CCS, October 2003, 103-111.
14. Lee, S.M., Hwang, Y.J., Lee, D.H., Lim, J.I.: Efficient Authentication for Low-Cost RFID Systems. International Conference on Computational Science and Its Applications – ICCSA, May 2005, 619-627.
15. Choi, E.Y., Lee, S.M., Lee, D.H.: Efficient RFID Authentication Protocol for Ubiquitous Computing Environment. In Proceedings of International Workshop on Security in Ubiquitous Computing Systems, December 2005.
16. Tuyls, P., Batina, L.: RFID-tags for Anti-Counterfeiting. In D. Pointcheval, editor, Topics in Cryptology - CT-RSA 2006, February 13-17 2006, volume 3860 of LNCS, Springer Verlag, 115-131.