A Lightweight RFID Authentication Protocol with Forward Security and Randomized Identifier

Zhicai Shi(✉), Fei Wu, Changzhi Wang, and Shitao Ren

School of Electronic and Electrical Engineering,
Shanghai University of Engineering Science, Shanghai 201620, China
szc1964@163.com

Abstract. The RFID tags only have limited computing and memory resources. This makes it difficult to solve their security and privacy problems. Authentication is considered as an effective approach to protect the security and privacy of RFID systems. Based on Hash function and the randomization of the tag's identifier, a lightweight authentication protocol is proposed. The protocol uses Hash function to ensure the anonymity and confidentiality of the RFID system. It uses a randomization function to randomize the tag's identifier to enhance the difficulty to reveal the secrecy of the RFID system. Time stamp and pseudorandom number generator are combined to prevent replay attack. It also completes the strong authentication of the backend server to the tag by twice authentication. The analysis shows that this protocol provides forward security and it can prevent eavesdropping, tracing, replay and de-synchronize attack. The protocol only uses Hash function and pseudorandom number generator. It is very suitable to the low-cost RFID system.

Keywords: RFID · Authentication protocol · Hash function · Security · Privacy

1 Introduction

With the development and application of the Internet of Things, Radio Frequency IDentification (RFID) technique gets the wide attention from various fields. RFID is a pervasive technology deployed to identify and trace some objects automatically. It uses radiowaves to communicate, without visible light and physical contact. It is considered as a supplementary or replacement technology for traditional barcode technology. Today, RFID systems have been successfully applied to manufacturing, supply chain, agriculture, transportation, health, e-payment, food safety tracing, and some other fields [1]. But the tags of RFID systems only have limited computing and memory resources and they use open wireless channel to communicate. It is easy for the adversary to eavesdrop the session information of an RFID system. Attackers can attack an RFID system by tracing, forging, spoofing, impersonating, tampering and de-synchronizing. So the privacy and security of RFID systems has become one of the main factors to hinder their wide application. Although some physical methods have been proposed to solve the security and privacy problems of RFID systems the research results show that it is the most flexible and effective method to use software encryption and authentication technique. The popular tags are some low-cost passive tags. They have very limited

G. Wang et al. (Eds.): SpaCCS 2016, LNCS 10066, pp. 1–13, 2016.
DOI: 10.1007/978-3-319-49148-6_1

computing and memory resources. They may be limited to hundreds of bits of storage, roughly between 5000 and 10000 logic gates. Within these logic gates, only 250 to 3000 gates can be devoted to security purpose [2]. It is very difficult to implement public key cryptography, even symmetric encryption algorithms for the low-cost passive RFID tags. So some lightweight cryptographic authentication protocols were proposed to satisfy the special requirements of RFID systems. But they usually use some complicated encryption algorithms and they are not suitable for the low-cost RFID tags. Some protocols use Hash function to complete the authentication for RFID systems, but they have some flaws so that they cannot entirely solve the security and privacy of RFID systems [3, 4]. So it is very necessary to design some simple and feasible lightweight authentication protocols for RFID systems, especially for the low-cost RFID systems.

The contribution of this paper is that we use Hash function and pseudorandom number generator to construct a novel lightweight authentication protocol for the low-cost RFID systems. Otherwise, we propose another special function, which is called the randomizing selecting bit function. This function randomly selects some bits of the tag's identifier to generate each session between tag and reader. Hence, each session only includes the partial information of the tag's identifier so as to enhance the difficulty to reveal the secrecy of RFID systems. The protocol provides forward security. It also completes the strong authentication of the backend server to the tag by twice authentication. It can prevent the leakage of the secret information and it implements the anonymous and confidential communication between tag and backend server/reader.

The paper is organized as follows. In Sect. 2, an RFID system's components, its security and privacy are introduced briefly. In Sect. 3, some typical Hash-based lightweight authentication protocols are analyzed and their flaws are pointed out. In Sect. 4, Hash function, a pseudorandom number generator and a randomizing selecting bit function are combined to construct a mutual authentication protocol for the low-cost RFID systems. In Sect. 5, the proposed protocol is analyzed and its security and privacy is proved. The secure performance of the protocol is compared with other similar authentication protocols. In Sect. 6, conclusions are given and the advantages of the proposed protocol are pointed out.

2 The RFID System, Its Security and Privacy

An RFID system consists of three components: Radio Frequency (RF) tag, RF reader and backend server, as shown in Fig. 1. A tag is a silicon chip with antenna and a small storage. There are two types of tags: active tag and passive tag. Active tags include batteries. Passive tags don't have any battery and they are activated by the RF signal from the reader. So they only have limited electric energy to transmit signals over shorter distance. This kind of tags is very cheap and they are usually called the low-cost tags. A reader is a device capable of sending and receiving data in the form of radio frequency signal. This device communicates with tag and reads its identifier. It has electric power enough to transmit signals over longer distance. So the communication channels

between reader and tag are asymmetric. The channel from reader to tag is called forward channel and the channel from tag to reader is called backward channel.

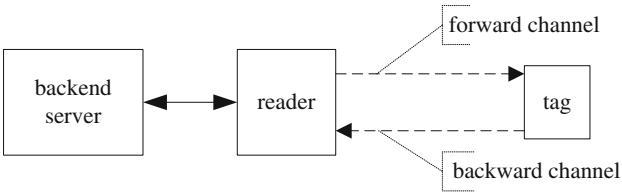

Fig. 1. The component of an RFID System

A backend server is used to store the detail information about the tagged objects, and it cooperates with reader to implement the authentication to tag. It searches the information about the tagged objects according to the tag's identifier and sends the information to the reader.

As an important component of the low-cost RFID system, the tag usually has very limited computing and memory resources and it uses the open wireless channel to communicate. It is difficult for a tag to implement some complicated cryptographic algorithms. So the channel between tag and reader is insecure. Most secure problems of RFID systems are resulted from the insecure wireless channel. But backend server and reader have abundant computing and storage resource. They can implement conventional cryptographic protocols. So the channel between backend server and reader is secure. They can be thought as one part of the RFID system, which is called the backend server/reader.

As a typical resource-constrained system, the low-cost RFID system is very vulnerable to some secure theats. An adversary can eavesdrop, intercept, tamper, block and replay each session between tag and backend sever/reader. It can impersonate a legitimate tag to cheat the backend server/reader. It can start de-synchronization attack by intercepting and blocking the sessions between tag and backend sever/reader. So a secure RFID system can resist against eavesdropping, tracing, replay and de-synchronization attack. Otherwise, it must satisfy forward security and anonymity.

3 Some Typical RFID Authentication Protocols

The cryptographic authentication protocols are thought as an important approach to ensure the privacy and security of RFID systems. They are divided into three categories: general authentication protocols, lightweight authentication protocols and ultra-lightweight authentication protocols. General authentication protocols are suitable for some situations with abundant computing and memory resources. They can use symmetric encryption algorithms, even public key cryptography. Lightweight authentication

protocols use Hash function, CRC function, pseudorandom number generating function, bitwise operations. Ultra-lightweight authentication protocols only use pseudorandom number generating function and bitwise operations. The research results justify that the encryption strength is very limited for ultra-lightweight authentication protocols and they cannot protect the security and privacy of RFID systems. General authentication protocols need abundant computing and storing resources and they are not suitable for the low-cost RFID system. Therefore lightweight authentication protocols become a unique approach to solve the security and privacy of the low-cost RFID system.

Many research works have been done for RFID lightweight authentication in recent years. Some authentication protocols use the one-way property of Hash functions to solve the secure and private problems of RFID systems. But most of them have serious security problems or they are not suitable to the low-cost RFID system. These typical Hash-based authentication protocols are Hash-Lock protocol, Randomized Hash-Lock protocol, Hash-chain protocol, and so on.

Based on the difficulty of inverting to solve an one-way Hash function, S.A. Weis et al. [5] firstly proposed Hash-Lock protocol, which attempts to provide mutual authentication between tag and reader. The protocol uses the pseudonym of the tag, *MetaID*, to replace the actual tag's *ID* to ensure its privacy. During the authenticating process the plaintext of the tag's *ID* is transferred between tag and reader, and *MetaID* is fixed. So an adversary easily compromises mutual authentication by simply eavesdropping and replaying these exchanged sessions between tag and reader. Moreover, an adversary easily traces the tag's holder by the fixed *MetaID*.

In order to overcome the flaws of Hash-Lock protocol, S.A. Weis and S.E. Sarma et al. proposed randomized Hash-Lock protocol [5]. This protocol uses the pseudorandom number generator (PRNG) to randomize the transferred sessions between tag and reader. Tags respond to reader's queries by generating a random number r, then Hashing its *ID* and concatenating the result with r, and sending them to the reader. A legitimate reader identifies one of its tags by performing a brute-force search of its known *IDs*. Then the reader sends the identified tag's *ID* to the tag by plaintext. It is easy for an adversary to eavesdrop and obtain the identity information of the tag. Hence, it is vulnerable to spoofing and replay attack. Moreover, the tag's holder is easily traced and this protocol cannot satisfy forward security.

M. Ohkubo et al. firstly proposed Hash-chain protocol [6, 7]. The aim of their protocol is to provide better protection of the user's privacy by refreshing the identifier of the tag for each authentication. Different from Hash-Lock protocol, Hash-chain protocol uses two different Hash functions, $H()$ and $G()$. This protocol only provides one-way authentication, namely, the reader authenticates the tag while the tag does not authenticate the reader. To achieve forward security, this protocol uses the Hash chain technique to renew the secret information stored in the tag. But this protocol does not use a random number generator and it is vulnerable to spoofing and replay attack. Ohkubo et al.'s scheme has a complexity in terms of Hash computations of $m \times n$, where m is the given maximum limit on the Hash chain length and n is the total number of tags. Thus, when the number of tags n or the chain length m is large the computation becomes unimaginable for the low-cost RFID system. Another similar scheme was provided by Sang-Soo Yeo et al. [8]. The scheme gave a conceptually simple but elegant solution to

defeat the tracing problem and ensure forward security. This scheme requires each tag to support 2 Hash functions. When the tag is queried by a reader, it sends the Hash value of its current identifier by a Hash function $G()$, and then renews its identity information using another different Hash function $H()$. These protocols use two different Hash functions and this makes it not suitable to the low-cost RFID system.

Yong Ki Lee et al. proposed a secure and low-cost authentication protocol for the RFID system, Semi-Randomized Access Control (SRAC) [9]. It also uses a pseudonym, *MetaID*, to replace the tag's *ID* like Hash-Lock protocol. It provides mutual authentication and forward security. It can protect RFID systems from many attacks, such as tracing, cloning and denial of service. However, it is vulnerable to replay attack. The adversary can simply eavesdrop and reuse *MetaID* to be authenticated successfully. Later, Su Mi Lee et al. used the challenge-response mechanism and proposed a low-cost RFID authentication protocol (LCAP) [10]. The aim of their effort is to solve the de-synchronized problem by maintaining a previous identifier in the backend server. This protocol provides mutual authentication and guarantees the location privacy of the tag's holder. It also provides untraceability by changing tag's identification dynamically. Nevertheless, it does not provide forward security, namely, an adversary can infer previous sessions about the tags after it reveals the present secret information of the tags.

Jung-Sik Cho et al. [11, 12] proposed a new Hash-based authentication protocol to solve the secure and private problems for the RFID system. However, Hyunsung Kim [13] demonstrated that this protocol is vulnerable to DOS attack. He pointed out that Jung-Sik Cho et al.'s protocol is vulnerable to traffic analysis and tag/reader impersonation attacks. More precisely, an adversary can impersonate a valid tag or reader with probability 1/4. Finally, an adversary can obtain some information about the secret values of the tag in the next session with probability 3/4. Therefore Hyunsung Kim proposed an improved protocol to offer protection against the attacks described above. But this enhanced version is as insecure as its predecessor. Walid I. Khedr [14] pointed out that an adversary can perform a de-synchronization attack by intercepting and tampering the transferred message. Further, Walid I. Khedr justified that Jung-Sik Cho et al.'s protocol cannot ensure forward security. Masoumeh Safkhani and Pedro Peris-Lopez et al. [15] also constructed three different attacks to demonstrate Jung-Sik Cho et al.'s protocol is vulnerable to de-synchronization attack and tag/reader impersonation attacks. Masoumeh Safkhani and Pedro Peris-Lopez et al. justified that the de-synchronization attack succeeds with probability 1 and the complexity of the attack is only one run of the protocol.

J.H. Ha and S.J. Moon et al. [16] proposed an RFID security protocol using the Hash-based functions and proved that their protocol can provide forward privacy. However, Da-Zhi Sun and Ji-Dong Zhong [17] pointed out that an attacker can track a target tag by observing previous unsuccessful sessions of the tag. Da-Zhi Sun et al. justified that J.H. Ha et al.'s protocol fails to provide forward privacy as they claimed and then they proposed another Hash-based authentication functions to overcome the weaknesses of J.H. Ha et al.'s protocol. But all these protocols use two different Hash functions and they are not suitable for the low-cost RFID system.

Liu Yang, Peng Yu et al. proposed an RFID secure authenticated protocol based on Hash function [18]. Their protocol ensures the privacy of the tag's secret information

and realizes three party mutual authentications among tag, reader and backend server. But, for each authentication process of the protocol, the tag and the reader call Hash function more than five times respectively. So their proposed protocol is so complicated that it is not suitable to the low-cost RFID system.

By analysis as described above, it can be concluded that recent proposed RFID authentication protocols with Hash function failed to solve the security and privacy for the low-cost RFID systems. Especially, many Hash-based authentication protocols cannot ensure forward security, or they use two different Hash functions, which hinders their application to the low-cost RFID system.

4 A Secure Hash-Based Authentication Protocol with Randomized Identifier for the Low-Cost RFID System

Some low-cost tags like EPC Global Class1 Gen2 standard can provide Hash function, pseudorandom number generator and simple bitwise operations [19, 20]. Now, we use these on-chip functions and bitwise operations to complete the mutual authentication between tag and backend server/reader. Moreover, we construct a function to randomly select the tag's partial identifier so that each session only includes the partial secrecy of a tag.

Supposed *ID* is the identifier of a tag and it uniquely identifies the tag. *pID* is the pseudonym of a tag and $pID = PRNG(ID)$. *PRNG*() is a pseudorandom number generator. The length of *ID* and *pID* is *L* bit and $L \in \{64, 96, 128\}$. *ID* and *pID* are stored in the tag. *curID*, *curpID*, *oldID* and *oldpID* are some other parameters, which are stored in the backend server. *curID* and *curpID* are the identifier and pseudonym of a tag used in the current authentication process. *oldID* and *oldpID* are the values of *ID* and *pID* used in the last successful authentication process. The purpose to store *oldID* and *oldpID* is to resist against de-synchronization attack. At the beginning of the authentication, the initial values of *curID* and *oldID* are set to the identifier of the tag. Namely, $curID = oldID = ID$ and $curpID = oldpID = PRNG(ID)$. The tag and the backend server share Hash function *Hash*(), pseudorandom number generator *PRNG*() and a random selecting bit function $f(x, m, n)$. These three functions are defined as follows:

$$\text{Hash}():\{0,1\}* \rightarrow \{0,1\}^L$$

$$PRNG():\{0,1\}* \rightarrow \{0,1\}^L$$

$$f(x,m,n) = x_m x_{m+1} \ldots \ldots x_n$$

Where x is the tag's identifier and $x = x_0 x_1 \ldots \ldots x_{L-1}$, m and n are two random numbers generated by the pseudorandom number generator, $0 \leq m \leq L-1$ and $0 \leq n \leq L-1$.

The function $f(x, m, n)$ randomly selects the partial identifier of a tag and uses it to generate each session between tag and backend server/reader. Hence, each session only includes one part of the tag's identifier and this increases the difficulty to reveal the tag's secrecy. The one-way property of Hash function *Hash*() is used to ensure the integrity

of each session and the confidential transfer of the tag's secrecy. The pseudorandom number generator *PRNG*() is used to keep the freshness of the sessions and to resist against tracing attack. Moreover, the time stamp of the backend server is used to resist against replay attack. The authentication protocol is shown in Fig. 2 and the symbols used by the protocol are described in Table 1.

Fig. 2. The authentication process of the proposed protocol

Table 1. The symbols used in the proposed authentication protocol

Notation	Description
ID, *pID*	The tag's identifier and its pseudonym
curID and *curpID*	The tag's identifier and its pseudonym used for the current authentication process
oldID and *oldpID*	The tag's identifier and its pseudonym used for the prior successful authentication process
L	The length of the tag's identifier
Hash()	A secure cryptographic Hash function
PRNG()	A pseudorandom number generator
$f(x, m, n)$	A randomly selecting bit function and its value is from the m^{th} to n^{th} bits of x
r, s	Two random numbers generated by backend server/reader and tag
t	The time stamp of the backend server
$DATA_k$	The information of the tag k stored in the backend server
%	Modular operation
||	Concatenation operation
$\oplus$	Bitwise exclusive-OR operation

The authentication process of the protocol is described as follows:

Step 1: the backend server/reader to the tag

The backend server calls the pseudorandom number generator *PRNG*() to generate a pseudorandom number r. Then it combines its time stamp t with r by exclusive-OR operation to construct the message $r \oplus t || \ challenge$. It transfers this message to the tag through the reader. Hence, a new authentication process begins.

Step 2: the tag to the backend server/reader

The tag receives the message $r \oplus t$ and it calls *PRNG*() to generate another pseudorandom number s. Then it calls *Hash*(), *PRNG*() and $f(x, m, n)$ to generate the messages as follows:

$$m1 = Hash((f(ID, 0, s\%L) \oplus s)||(r \oplus t)) \tag{1}$$

$$m2 = Hash((r \oplus t)||(f(ID, s\%L, L-1) \oplus s)) \tag{2}$$

$$n1 = PRNG((f(pID, 0, s\%L) \oplus s)||(r \oplus t)) \tag{3}$$

$$n2 = PRNG((r \oplus t)||(f(pID, s\%L, L-1) \oplus s)) \tag{4}$$

$$p = PRNG(pID \oplus r \oplus t) \oplus s \tag{5}$$

The tag constructs the message $m1||n2||p$ and it sends this message to the backend server through the reader.

Step 3: the backend server/reader to the tag

After the backend server receives the message $m1||n2||p$, it searches its backend database to get each record about the tags, (*curID*, *curpID*, *oldID*, *oldpID*). Firstly, it uses *curpID* of the current record to compute $p \oplus PRNG(curpID \oplus r \oplus t)$ and to abstract s.

Secondly, it uses *curID* and *curpID* of the current record to replace *ID* and *pID* in Eqs. (1) to (4) to compute $m1'$, $m2'$, $n1'$ and $n2'$. Then it compares $m1'$ and $n2'$ with $m1$ and $n2$ respectively. If one of them is not equal the backend server uses *oldID* and *oldpID* of the current record to repeat the above procedure to calculate s, $m1'$, $m2'$, $n1'$ and $n2'$ again. The backend server compares $m1'$ and $n2'$ with $m1$ and $n2$. If one of them is not equal yet then next record is picked up from the database to repeat the procedure described above until all records are processed. If $m1'$ does not equal $m1$ or $n2'$ does not equal $n2$ for all records, the authentication to the tag fails and the protocol exits. If there exists one record which satisfies that $m1'$ equals $m1$ and $n2'$ equals $n2$, the first authentication of the backend server to the tag succeeds. Then the backend server sends the message $m2'$ to the tag through the reader. The backend server begins to update its secret keys as follows.

If (*curID*, *curpID*) is used for the above successful authentication the backend server updates its secret keys as follows:

$$oldID = curID \tag{6}$$

$$oldpID = curpID \tag{7}$$

$$curID = PRNG(curID \oplus r \oplus t \oplus s) \tag{8}$$

$$curpID = PRNG(curpID \oplus r \oplus t \oplus s) \tag{9}$$

If (*oldID*, *oldpID*) is used for the above successful authentication the backend server holds its current *oldID* and *oldpID*. It only updates its partial secret keys as follows:

$$curID = PRNG(oldID \oplus r \oplus t \oplus s) \tag{10}$$

$$curpID = PRNG(oldpID \oplus r \oplus t \oplus s) \tag{11}$$

Step 4: the tag to the backend server/reader

After the tag receives the message $m2'$, it compares $m2'$ with $m2$. If they are not equal the authentication to the backend server/reader fails and the protocol exits. Otherwise the authentication to the backend server/reader succeeds. Then the tag begins to update its secret keys as follows:

$$ID = PRNG(ID \oplus r \oplus t \oplus s) \tag{12}$$

$$pID = PRNG(pID \oplus r \oplus t \oplus s) \tag{13}$$

The tag sends *n1* to the backend server through the reader.

Step 5: the backend server to the reader

The backend server receives the message $n1$ from the tag and it compares $n1$ with $n1'$. If they are not equal the authentication fails and the protocol exits. Otherwise the second authentication to the tag is completed successfully.

Then the backend server gets the detail information about the tag, $DATA_k$, from its database and sends the information to the reader. After the reader receives $DATA_k$, it displays $DATA_k$ on its screen.

The procedure described above completes the mutual authentication between backend server/reader and tag. Meanwhile, it also completes the strong authentication of the backend server to the tag by twice authentication.

5 The Analysis to the Privacy and Security of the Proposed Protocol

The authentication process described above shows that the protocol uses the random selecting bit function to make the sessions unpredictable and this increases the difficulty to reveal the secret information of the tag. One-way property of Hash function ensures the integrity of the sessions and the confidential transfer of the secret information of the RFID system. A pseudorandom number generator randomizes the messages sent by the tag so that it is difficult for the adversary to trace and identify a tag. Meanwhile, the time stamp is used to resist against replay attack. The protocol provides forward security and it can also resist against de-synchronization attack.

- Forward security. After each authentication is completed the protocol updates the secrecy of the tag. Therefore the protocol uses some different secret keys to encrypt and generate the sessions for each authentication. There is not any relationship between the previous sessions and the current secret keys. Although an adversary reveals the current secrecy of the tag he cannot decrypt the previous session messages.
- De-synchronization attack. The protocol stores *curID*, *curpID*, *oldID*, and *oldpID* in the backend server. *oldID*, and *oldpID* are the values of *curID* and *curpID* for the last successful authentication. If the tag cannot synchronously update its secrecy with the backend server they can use *oldID*, and *oldpID* to complete the later authentication so as to resist against de-synchronization attack.
- Eavesdropping. For the whole authenticating process of the protocol, all session messages are processed by Hash function or the pseudorandom number generator. Although an adversary can eavesdrop all messages transferred between tag and backend server/reader he cannot reveal these message. So the protocol can effectively resist against the leakage of the secret information and it ensures the confidential and anonymous communication between backend server/reader and tag.
- Tracing attack. If a tag repeats to send the same message to the backend server/reader many times an adversary can easily trace and identify the tag. In order to resist against tracing attack, the tag generates a new pseudorandom number for each authentication and the pseudorandom number is used to randomize the session messages. Therefore the freshness of the session messages is ensured. For any different challenge from the backend server/reader the tag will give a different response. An adversary cannot judge which tag sends the session messages eavesdropped by him and it cannot distinguish two different tags. Therefore the protocol can resist against tracing attack.
- Replay attack. This attack means that an adversary re-sends the session messages intercepted by him so as to get the authentication of the RFID system. Because all session messages transferred between backend server/reader and tag are processed by the time stamp of the backend server. An adversary can intercept the session messages and re-sends them later. But these messages are out of time and they are

meaningless for the later authentication. So the protocol can resist against replay attack.

- Anonymity. The protocol uses Hash function and pseudorandom number generator to process the partial identifier of the tag and generate all sessions between tag and backend server/reader. Each session only includes the partial secret information of the tag. Although an adversary can intercept these sessions it is difficult for him to get the whole secrecy of the tag. Hash function is a one-way function. An adversary cannot get the plaintext of these sessions. So the protocol ensures the anonymity of the RFID system.

Compared with other similar protocols, our proposed protocol has many advantages, which are shown by Table 2.

Table 2. The comparison among the different authentication protocols

Protocols	Eaves dropping	Tracing attack	Replay attack	De-synchronized attack	Spoofing attack	Forward security
Hash-Lock	x	x	x	–	x	x
Random Hash-Lock	x	x	x	–	x	x
Hash chain	√	√	x	√	x	√
SRAC	√	√	x	√	x	√
LCAP	√	√	√	√	√	x
Our protocol	√	√	√	√	√	√

6 Conclusions

The privacy and security of the RFID system is one of the important factors to decide whether it can be applied widely. The current popular tags are some low-cost passive tags and they have very limited computing and storing resources. It is very difficult for these tags to complete some complicated cryptographic protocols. In order to ensure the security and privacy of the RFID systems with low-cost tags, we propose a strong lightweight authentication protocol. This protocol provides forward security and anonymity. It uses Hash function and random selecting bit function to process the session messages so as to increase the difficulty to reveal the secret information of the tag. Meanwhile, twice authentication to the tag also increases the secure strength of the protocol. The analysis to the proposed protocol proves that the protocol can provide forward security and it can resist against eavesdropping, tracing, replay and de-synchronization attacks. It completes the mutual authentication between tag and backend server/reader. The protocol only uses Hash function, pseudorandom number generator and some simple bitwise operations. So the protocol is very suitable to some resource-constrained environment like the low-cost RFID systems.

Acknowledgments. We are appreciated to anonymous reviewers for their constructive suggestion to this paper. The relative work about this paper is supported by National Natural Science Foundation of China (No. 61272097).

References

1. Chen, M., Luo, W., Mo, Z., Chen, S., Fang, Y.: An efficient tag search protocol in large-scale RFID systems with noisy channel. IEEE/ACM Trans. Netw. **24**(2), 703–716 (2016)
2. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: RFID systems: a survey on security threats and proposed solutions. In: Cuenca, P., Orozco-Barbosa, L. (eds.) PWC 2006. LNCS, vol. 4217, pp. 159–170. Springer, Heidelberg (2006). doi: 10.1007/11872153_14
3. Chikouche, N., Cherif, F., Cayrel, P.-L.: Weaknesses in two RFID authentication weaknesses. In: El Hajji, S., et al. (eds.) C2SI 2015, LNCS, vol. 9084, pp. 162–172. Springer, Heidelberg (2015)
4. Deng, R.H., Li, Y., Yung, M., Zhao, Y.: A new framework for RFID privacy. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 1–18. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15497-3_1
5. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and privacy aspects of low-cost radio frequency identification systems. In: Proceedings of the 1st International Conference on Security in Pervasive Computing, Boppard, Germany, pp. 201–212 (2003)
6. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic approach to "Privacy-Friendly" tags. In: RFID Privacy Workshop. MIT Press, Cambridge (2003)
7. Ohkubo, M., Suzuki, K., Kinoshita, S.: Hash-chain based forward secure privacy protection scheme for low-cost RFID. In: Proceedings of the 2004 Symposium on Cryptography and Information Security, Sendai, Japan, pp. 719–724 (2004)
8. Yeo, S.-S., Kim, S.K.: Scalable and flexible privacy protection scheme for RFID systems. In: Molva, R., Tsudik, G., Westhoff, D. (eds.) ESAS 2005. LNCS, vol. 3813, pp. 153–163. Springer, Heidelberg (2005). doi:10.1007/11601494_13
9. Lee, Y.K., Verbauwhede, I.: Secure and low-cost RFID authentication protocols. In: Proceedings of the 2nd IEEE Workshop on Adaptive Wireless Networks, St. Louis, USA, pp. 1–5 (2005)
10. Lee, S.M., Hwang, Y.J., Lee, D.H., Lim, J.I.: Efficient authentication for low-cost RFID systems. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3480, pp. 619–627. Springer, Heidelberg (2005). doi:10.1007/11424758_65
11. Cho, J.-S., Yeo, S.S., Kim, S.K.: Securing against brute-force attack: a hash-based RFID mutual authentication protocol using a secret value. Comput. Commun. **34**(3), 391–397 (2011)
12. Cho, J.-S., Jeong, Y.-S., Sang, O.-P.: Consideration on the brute-force attack cost and retrieval cost: a hash-based radio-frequency identification (RFID) tag mutual authentication protocol. Comput. Math. Appl. **3**, 1–8 (2012)
13. Kim, H.: Desynchronization attack on hash-based RFID mutual authentication protocol. J. Secur. Eng. **9**(4), 357–365 (2012)
14. Khedr, W.I.: SRFID: a hash-based secure scheme for low cost RFID systems. Egypt. Inf. J. **14**, 89–98 (2013)
15. Safkhani, M., Peris-Lopez, P., Hernandez-Castro, J.C., Bagheri, N.: Cryptanalysis of the Cho et al. protocol: a hash-based RFID tag mutual authentication protocol. J. Comput. Appl. Math. **259**, 571–577 (2014)

16. Ha, J., Moon, S., Zhou, J., Ha, J.: A new formal proof model for RFID location privacy. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 267–281. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88313-5_18
17. Sun, D.-Z., Zhong, J.-D.: A hash-based RFID security protocol for strong privacy protection. IEEE Trans. Consum. Electron. **58**(4), 1246–1252 (2012)
18. Yang, L., Yu, P., Bailing, W., Yun, Q., Xuefeng, B.: Hash-based RFID mutual authentication protocol. Int. J. Secur. Appl. **7**(3), 183–194 (2013)
19. Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., Verbauwhede, I.: Spongent: a lightweight hash function. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 312–325. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23951-9_21
20. Gao, S., Wang, H.: Forward private RFID authentication protocol based on universal hash function. J. Inf. Comput. Sci. **10**(11), 3477–3488 (2013)