A Lightweight Hash-Based Mutual Authentication Protocol for RFID

Zhangbing Li[1,2(✉)], Xiaoyong Zhong[1], Xiaochun Chen[1], and Jianxun Liu[1,2]

[1] School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, Hunan, China
lzb_xt@126.com, chen_xiaochuns@126.com, zxyhnust@163.com, jx529@gmail.com

[2] Key Lab of Knowledge Processing and Networked Manufacturing, College of Hunan Province, Xiangtan, Hunan, China

Abstract. For the RFID authentication protocols based on Hash functions, there are some shortcomings, such as imperfect defense on the attacks, intensive calculation, time-consuming authentication process, and so on. By using of the dynamic-shared key and one-way feature of Hash function, a lightweight Hash-based mutual authentication protocol has been proposed and proved by SVO logic in this paper. It avoids an exhaustive search in the back-end database, and supports the transfer of ownership of the tag and the scalability of RFID system. Besides resisting the common attacks, the protocol is suitable for the RFID system that needs to be low-cost, lightweight computing and large numbers of tags, which is of significant merit for RFID application.

Keywords: RFID · Mutual authentication · Hash function · SVO logic · Security protocol

1 Introduction

In recent years, radio frequency identification (RFID) technology has been widely used in supply chain management, target detection and tracking, electronic payment, environmental monitoring and so on, but the information leakage and other security issues are also increasingly highlighted [1]. A complete RFID system is composed of three parts: a reader, tags and back-end database. It may suffer from the main attacks as follows: message replay, denial of service, tag clone camouflage, reader fake, unauthorized access and track, desynchronization etc. Therefore a RFID system needs a strong security protocol that can withstand the above attacks to meet the security demands and protect the data privacy. RFID authentication is designed to make mutual authentication between the reader and tag, but does not allow any attacker to recover the intimate information in the course of authentication process. Thus, to design a secure authentication protocol is becoming a research hotspot [2].

Authentication protocols need to encrypt the sensitive data for preventing information leakage or tampering with hackers. Encryption function should be able to ensure the data integrity and guarantee the confidentiality and factuality of RFID

J. Cao and J. Liu (Eds.): MiPAC 2016, CCIS 686, pp. 87–98, 2017.
DOI: 10.1007/978-981-10-3996-6_7

system. In view of the limited resources and storage space in RFID tag, the RFID authentication process will become very difficult since some encryption algorithm has a complicated computation in the practical application. Hash function is a frequently used algorithm for the RFID protocol with reliable safety and acceptable computing cost [1, 2]. Some typical authentication protocols based on hash function are: random Hash-Lock protocol, Hash-Chain protocol, RFID Library protocol, LCAP (Low-cost RFID Authentication Protocol) [3–9], and some improved protocols based on them [1, 2, 10–20]. These protocols all assume that the channel between the back-end database and the reader is secure but insecure between the reader and Tags, and requires authentication.

In 2002 Sarma [3] proposed Hash-Lock protocol which uses metaID(equal to hash (key)) to replace the true ID of the tag to protect the data privacy. But it will easily suffer from replaying and spoofing attack since the metaID value is unchanged during each communication, and the protocol also does not prevent tracking. Then Weis [4] mended the Hash-Lock protocol using the unpredictability of the random number. This protocol is called RHL protocol which ensures the indistinguishability of the session data and resists the position tracking, but the plaintext transmission for the tag ID still does not resist the counterfeiting and replay attacks. The Hash-Chain protocol proposed by Ohkubo [5] uses two different Hash functions to refresh the ID dynamically, and is also with strong ability of anti tracking. But it can only achieve a one-way authentication, and is vulnerable to the man-in-the-middle and replay attacks. Henrici [6] proposed a protocol based on hash ID-changed, which introduces the identification information to prevent man-in-the-middle attack, but there are some risks of desynchronization between the tag and database. The LCAP protocol based on distributed inquiry-response mode is proposed by Rhee et al. [7], which imports the random number in both the reader and tag. But there exists of the problem for forward security if the attacker gets the tag ID, and also the risk of losing synchronization between the tag and database. Molnar et al. [8] proposed David digital library protocol, which is a mutual authentication protocol and different from the hash ID-changed protocol. It makes use of a static ID and the shared secret value S to achieve the authentication between the server and tag, but the authentication is time-consuming and has of intensive calculation and high cost. In 2006, Tsudik [9] proposed the YA-TRAP's authentication protocol which introduces the time stamp, but it is vulnerable to denial of service attacks (unable to distinguish from illegal and legal tags).

Some domestic scholars have also made the design and improvement of RFID security protocols [10–17]. Li [10] introduces the random number to prevent replay attacks in the improved Hash-Chain protocol, but it will appear Dos attack when the number of illegal tags is more than of (M + T − 1). Hash-Chain protocol based on two-dimensional interval are proposed by Xiong [11], which increases the index (Ai, Bi) for each tag, but there are threats of replay attack and impersonation attack. Yuan improved protocol [12] hides the tag's ID for transmission, but needs the traversal calculation to find the destination tag with hash function, and doesn't resist the asynchronous attack. The location index of the tags uses a plaintext in the Chen Shaowei's protocol, which is vulnerable to the tracking attack and denial of service attack. Zhou [14] introduces the pseudo random number based on Hash-chain, and the update cost of the key is large. Liu Peng et al. make use of the random numbers

produced by the reader and tag, and assemble them with ID of tags as the input of hash function, and transmit the values to the back-end database for calculation and comparison by exhaustive search, but it may lead to poor performance of the system and does not resist DOS attack. The RP and RSP authentication protocols are proposed by Zheng [1]. RSP utilizes the random number generator, the exclusive OR function, the same OR function and hash function respectively to enshroud the interaction information between the tag and reader, and the security is formally proved by using BAN logic, but the transmission of hash value for single ID of the tag will lead to replay attack. The HSASILC protocol for RFID authentication is proposed by Si [17] with GNY logic proving, which introduces the time stamp in each certification step, but it does not resist man-in-the-middle attack.

In short, there are still some problems in the existing RFID authentication protocols based on Hash function, and it is of great practical significance to design efficient, secure and reliable RFID Hash-based protocol with limited cost. So, in this paper a lightweight mutual authentication protocol based on dynamic shared keys is proposed, which is suitable for the RFID system with low cost, low computational cost, and large numbers of Tags.

2 Lightweight RFID Mutual Authentication Protocol Based on Dynamic Shared-Key

In this protocol, the query-and-response mechanism is used and the mutual authentication process is based on the improvement of the storage information in the RFID tags.

2.1 Initial Condition

Initially, the parameters including the location index as *ki*, identification as *tagID* and a dynamic shared-key as *key* are stored in the tag, which is embedded with a Hash function (SHA-1, MD4) and a random number generator. The reader has a random number generator, and the back-end database stores all the records for all tags and readers. A record of a tag should fully include following parameters such as *Ki*, ID_T, key_{old} and key_{new}, and the backend system can carry out a variety of complex computing. Assume as follows:

H(x) is a one-way Hash function; R_R is a random number generated by the reader, R_{Ti} is a random number generated by the tag; *Rot(A, B)* realizes circularly the left shift of binary number A with n bit, and n is the binary 1 number of B (Hamming weight). The variables key_{old} and key_{new} own the same value as key by initialization.

2.2 Authentication Steps

The authentication process of this protocol is shown in Fig. 1. The process is specially described as follows:

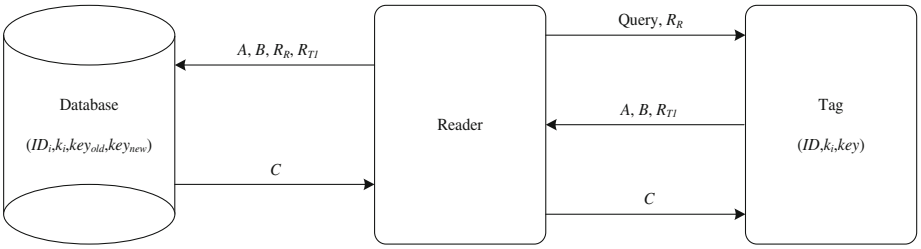

Fig. 1. Certification process of the protocol

(1) The reader, together with the random number R_R generated by itself, sends a Query to the tag as an authentication request.
(2) Within the range of effective communication, there may be more than one tag to respond the reader at the same time, which may lead to the collision of the Radio frequency signal, and cause the failure for tag identification. So the anti-collision protocol will be lunched to ensure that the suitable tag is selected for the response.
(3) The authentication process starts between the selected tag and the reader. The tag generates the random number R_T, and calculates as follows:

$$A = \mathrm{H}(\mathrm{Rot}(R_T \oplus R_R,\ R_R)) \oplus k_i \tag{2.1}$$

$$B = \mathrm{Rot}(\mathrm{H}(ID \oplus key),\ R_{Ti}) \tag{2.2}$$

where, A is used for encrypting the Ki value to transmit the tag's location index in back-end database, B is used for transmitting the dynamic shared-key secretly. Then the variables A, B and R_{Ti} are sent to the reader by the tag.
(4) The reader will transmit the received variables like A, B and R_{Ti} to the back-end database, as well as the random number RR generated by its own self. The transport way may be through the serial port with wired way or other net way.
(5) The back-end database system receives the information from the reader, and calculates the location index of the record for the response tag in the database.

$$k_i = A \oplus \mathrm{H}(\mathrm{Rot}(R_{Ti} \oplus R_R,\ R_R)) \tag{2.3}$$

According to the k_i value the system locates the record of the tag in the database and read the corresponding variables such as ID_i, f_i, keyold and keynew. If the record search fails or the variable f_i equals to 5 (over 5 times failure), the tag verification is failed and have to turn to (7).
(6) The calculations as following will be done in the back-end database system:

$$B_1 = \mathrm{Rot}(\mathrm{H}(ID_i \oplus key_{new}),\ R_{Ti}) \tag{2.4}$$

$$B_2 = \mathrm{Rot}(\mathrm{H}(ID_i \oplus key_{old}),\ R_{Ti}) \tag{2.5}$$

If $B_1 = B$ then makes $key_i = key_{new}$ and $f_i = 0$, where, it is to say that both the previous and this authentication are successful;

If $B_2 = B$ then makes $key_i = key_{old}$ and $f_i = 0$, which explains this authentication is successful but the previous is failed;

Otherwise set $key_i = key_{old}$ and $f_i = f_i + 1$, and what is illustrated that both the variables key_{new} and key_{old} in the back-end database is different from the key value in the tag, and both the previous and this authentication are failed. The current tag is considered as illegal.

The calculation will be done as follows:

$$key_{old} = key_i \tag{2.6}$$

$$key_{new} = \mathrm{Rot}(ID_i \oplus key_{old}, R_R \oplus R_{Ti}) \tag{2.7}$$

$$C = \mathrm{H}(\mathrm{Rot}(ID_i \oplus key_{old}, R_{Ti})) \tag{2.8}$$

The system updates the k_i-th record of database with parameters f_i, key_{old} and key_{new}.

If $B = B_1$ or $B = B_2$ then turn to (8).

(7) Set

$$C = \mathrm{H}(ID_i \oplus R_{Ti} \oplus R_R) \oplus f_i \tag{2.9}$$

(8) The back-end database transfers the C value to the reader, which forwards it to the tag.

(9) The tag receives the C value from the reader and then calculates:

$$C_1 = \mathrm{H}(\mathrm{Rot}(ID \oplus key, R_{Ti})) \tag{2.10}$$

If $C_1 = C$ then the reader and tag are legitimate, and the key value will be calculated and updated in the tag:

$$key = \mathrm{Rot}(ID \oplus key, R_R \oplus R_{Ti}) \tag{2.11}$$

Otherwise the authentication fails, the agreement is terminated.

2.3 Protocol Characteristics

This protocol has the following characteristics:

(1) Hide the location index the tag ID in the database to avoid the exhaustive search for each tag ID and comparison;
(2) Hide the authentication key through the transformation of the hash value of the communication;
(3) Record the last two certified keys in the database, and the key_{old} is the final key;
(4) Record the number of failures to prevent unrestricted attacks for authentication;
(5) Support the ownership transfer of tags. After the tag and reader finish the mutual authentication, the key value shared by the tag and back-end database is updated dynamically, and can be normally used after the tag ownership is transferred;

(6) Support the system scalability. Increase or decrease in the number of tags will not significantly affect the system performance.

3 Safety and Performance Analysis

3.1 Security Analysis

(1) Confidentiality. Since the tag interior is safe, it is difficult to obtain the internal key and the identifier of tag unless the attacker makes the reverse engineering analysis of the tag's internal circuit. Even though the current session information of the tag Ti is known, the attacker can not obtain the tag ID because the communication information only includes A, B, C, R_R and R_{Ti} between reader and tag, which are packed by using the unidirectional Hash function except the random numbers, so the protocol can guarantee the anonymity of the tag. After each successful authentication, the shared key in the tag and database is synchronously updated. For each authentication request, the tag responses include the A and B values are calculated by using the shared key and random numbers of a new round, as well as the Hash function. So each response from the tag to reader is not the same, i.e. the tag has the indistinguishability.
(2) Integrity. All the received datum will be calculated and verified by use of the one-way characteristic of Hash function, any modification on the data will lead to the failure of the authentication, which can guarantee the integrity of the data.
(3) Forward security. Each authentication request makes use of a random number of new round to calculate the A and B values. The tag ID only can be used by the tag own, thus an attacker is unable to figure out the tag ID from the hash value, and cannot work out the last key to decrypt the last message from this key value yet. So the attacker does not recognize the last session of the Ti tag, and it's past behavior cannot be traced.
(4) Backward security. The each response of A and B values from the tag are worked out of the random number R_R and R_{Ti} by hash function and Rot-function in tag. The shared-key between the tag and back-end database is calculated with the key (key_{old}) of current session and hash function for update, and the attacker is unable to get the update parameters of the key only by eavesdropping. In the case of R_R and R_{Ti}, the attacker cannot obtain the key information needed for the next authentication by self-calculation.
(5) Anti replay attack. The tag and the reader respectively have new random numbers in each certification process. These random numbers ensure the freshness of the transport message for the authentication based on challenge-response mode, and each successful authentication makes the new shared key updated synchronously between the tag and the database. Therefore, though the attacker repeatedly sends authentication request to the tag with the same random number R_R, the responses will be different by hash encryption, and the different tags response different messages because of different random numbers, so the tag will not be tracked.

(6) Anti desynchronization attack. The shared-key value in the tag will not be updated because of unsuccessful authentication, but the key_{old} in the back-end database all the time keeps the shared-key value for the successful authentication right, which can make sure of using the right key in the next authentication. So the synchronization of secret information can be kept between the server and the tags.
(7) Anti DOS attacks. This protocol does not limit the number of access tags instead of the number of failed authentication. If the third session between reader and tag in the certification process is blocked, that leads to the dynamic shared-key in back-end database updated but the corresponding key in tag not updated synchronously. However, $key_{old} = key$ in the database will be the right key for the authentication next time, the updated key value keynew will be invalid. While the reader launches the next authentication, the equation $key = key_{old}$ in the back-end database is still set up, the tag can still be certified. So the protocol has a good resistance to denial of service(DOS) attacks.

According to the security of seven aspects: indistinguishability, forward security, replay attack, spoofing attack, non traceability, can not track of key, dynamic Key update, and anti desynchronization attacks, the proposed protocol is compared with Hash-Lock protocol (HL), Random-Hash-Lock protocol (RHL), Hash-Chain protocol (HC) and the two improved protocol in 12th reference (Ref. 12) and 1st reference (Ref. 1). By comparison as shown in Table 1, it is found that the proposed protocol has better security than other protocols.

Table 1. Comparison of the security of the protocols

Security	HL	RHL	HC	Ref. 12	Ref. 1	This protocol
Indistinguishability	×	✓	✓	✓	✓	✓
Forward security	✓	✓	✓	✓	✓	✓
Replay attack	×	×	×	✓	×	✓
Spoofing attack	×	×	×	✓	✓	✓
Non traceability	×	×	✓	✓	✓	✓
Dynamic key update	×	×	×	×	×	✓
Anti desynchronization attack	○	○	○	×	○	✓

The protocol has security with the case of: X: does not; ✓: has; ○: leaves out of account.

3.2 Computational Performance Analysis

The Hash value and the shared key are required to calculate in the tag and the back-end database for Hash-based RFID authentication protocol, but the storage capacity and the amount of computation will affect the efficiency of the implementation of the protocol and the production cost of the tags. The performance of each protocol is analyzed from two aspects: the calculation amount and the storage capacity of the tag and the back-end database. Comparisons are as shown in Tables 2 and 3, where N denotes the number of tags, H says hash function, L shows logic operations, M figures Hash-chain length, $O(x)$ is the complexity of the calculation for searching tags in back-end database.

Table 2. Comparison of calculation

Calculation	HL	RHL	HC	Ref. 12	Ref. 1	This Protocol
Backend DB	$o(1)H$	$o(N)H$	$o(MN)H$	$o(N)H$	$o(1)H + o(1)L$	$o(1)H + o(1)L$
Tag	H	H	$2H$	$3H$	$2H$	H

Table 3. Comparison of storage capacity

Storage cap.	HL	RHL	HC	Ref. 12	Ref. 1	This protocol
Backend DB	$4l * N$	$l * N$	$2l * N$	$2l * N$	$3l * N$	$4l * N$
Tag	$2l$	l	l	$3l$	$3l$	$3l$

As can be seen from Table 2, compared with other protocols, the amount of computation for this protocol no matter on the tag or in the back-end database is correspondingly less. So it improves the efficiency of the authentication. However, as Table 3 shown, the storage capacity of this protocol in tags and back-end database storage is a bit more (where l is the length of the shared key) than the others, which has almost no impact on the calculation.

3.3 Proof of the Protocol with SVO Logic

The security of this protocol is proved by SVO logic [21, 22]. SVO logic is proposed by Syverson and Van Oorshot, which is optimized and derived from four kinds of logics including BAN, GNY, AT and VO. With very simple inference rules and axioms, SVO Logic repairs the defects and deficiencies of other Logics like BAN.

In the course of proof, R represents the reader (with database), T represents the tag. The axiom A1, A2, A3, A4 are shown as in the references [21, 22]. During SVO logical reasoning, those symbols "$|\equiv$", "$\triangleleft$", "$|:$", "$|\approx$", "$|\Rightarrow$", "$\ni$", "#" and "$\equiv$" are still used to express "believe", "received", "said", "say", "control", "has", "fresh" and "equivalent" respectively. The analysis of the RFID mutual authentication protocol is as follows:

1. Initial hypothesis

P1: $R|\equiv\#R_R$, $T|\equiv\#R_{Ti}$
P2: $R|\equiv R \ni K, R|\equiv R \ni ID$, $T|\equiv T \ni ID$, $T|\equiv T \ni K$
P3: $T|\equiv T \overset{K}{\leftrightarrow} R, R|\equiv T \overset{K}{\leftrightarrow} R$
P4: $T|\equiv((K, ID, R_R, R_{Ti})|\Rightarrow(A, B, C))$, $R|\equiv((K, ID, R_R, R_{Ti})|\Rightarrow(A, B, C))$
P5: $T \triangleleft R_R$, $T \triangleleft C$
P6: $R \triangleleft (A, B, R_{Ti})$
P7: $R|\equiv R \triangleleft *1$, $T|\equiv T \triangleleft *2$ (An understanding of the received message by the subject, unknown message)
P8: $R|\equiv(R \triangleleft *1 \supset R \triangleleft (A, B, R_{Ti}))$ (Interpretation of the received messages by the subject)
P9: $T|\equiv(T \triangleleft *2 \supset T \triangleleft C)$

P10: $R \mid\equiv (R \triangleleft \{X^T\}_K \wedge R \overset{K}{\leftrightarrow} T \supset T \mid\sim X)$

P11: $T \mid\equiv (T \triangleleft \{X^R\}_K \wedge R \overset{K}{\leftrightarrow} T \supset R \mid\sim X)$

2. Proof goal

G1: $R \mid\equiv (T \ni K)$
G2: $R \mid\equiv (T \ni ID)$
G3: $T \mid\equiv (R \ni K)$
G4: $T \mid\equiv (R \ni ID)$
G5: $R \mid\equiv \#R_{Ti}$
G6: $T \mid\equiv \#R_R$

3. Derivation by using SVO Logic

SVO logic has 20 axioms and 2 derivation rules, see References [21, 22]. NEC rule is that $\mid\!-P \mid\equiv \Phi$can be derived by $\mid\!-\Phi$; MP rule is that ψ can be derived by Φ and $\Phi \supset \psi$.

Firstly, an inference can be made by P6, P8 and Trust axiom ($P \mid\equiv \varphi \wedge P \mid\equiv (\varphi \supset \psi) \supset P \mid\equiv \psi$, which is denoted by A1 in this paper):

$$R \mid\equiv R \triangleleft (A, B, R_{Ti})\square \tag{3.1}$$

Secondly, an inference can be made by P3, P10, formula (3.1) and A1:

$$R \mid\equiv T \mid\sim \{A, B, R_{Ti}\}_K \tag{3.2}$$

So the formula "$R \mid\equiv (T \ni K)$" is established, and the goal G1 has to be permitted.

The following formula can be deduced by P1, P4, A1, NEC rule and Message-freshness axiom (#(Xi) $\supset$ #(X1, X2, …, Xn), which is denoted by A2 in this paper):

$$R \mid\equiv \#\{A, B, R_{Ti}\}_K \tag{3.3}$$

It can be reasoned out from the formulae (3.2) and (3.3), the rule NEC and the temporary-value-verification axioms ((#(Xi) $\wedge$ P$\mid\sim$X) $\supset$ P$\mid\approx$ X, which is denoted by A3 in this paper):

$$R \mid\equiv T \mid\approx \{A, B, R_{Ti}\}_K \tag{3.4}$$

Furthermore, the inference can be worked out by the formula (3.2) and the message sending axiom (P $\mid\approx$ (X1, X2, …, Xn) $\supset$ P $\mid\sim$ (X1, X2, …, Xn) $\wedge$ P $\ni$ Xi, which is denoted by A1 in this paper):

$$R \mid\equiv (T \ni (A, B, R_{Ti})) \tag{3.5}$$

So the formula "R |≡(T ∋ ID)" is established according to the formula (3.5), P4, A1 and the message understanding axiom (P |≡(P ∋ F(X)) ⊃ P |≡(P ∋ X), which is denoted by A5 in this paper), and the goal G2 gets permit.

In succession, the inference can be made by P5, P7, P9 and A1 as follows:

$$T \mid \equiv T \triangleleft \{C\}_K \tag{3.6}$$

It can be easily inferred out by P3, P11, the formula (3.6) and A1 as follows:

$$T \mid \equiv R \mid : \{C\}_K \tag{3.7}$$

So the formula "T |≡(R ∋ K)" is set up, and the goal G3 gets permit.

Similarly, it can be deduced by the P1, A2, A1 and NEC rule as follows:

$$T \mid \equiv \#\{C\}_K \tag{3.8}$$

The following formula can be reasoned out from the formulae (3.7) and (3.8), A3, A1and NEC rule:

$$T \mid \equiv R \mid \approx \{C\}_K \tag{3.9}$$

An inference can be made by the formula (3.9), A4, A1and NEC rule as follows:

$$T \mid \equiv (R \ni C) \tag{3.10}$$

So the formula "T |≡(R ∋ ID)" is established and the goal G4 gets permit.

The formula "R |≡#R_{Ti}" can be referred out by the formulae (3.4) and (3.5), A2 and A1, therefore the goal G5 gets permit.

The formula "T |≡#R_R" can be referred out by the formulae (3.9) and (3.10), A2 and A1, therefore the goal G6 gets permit.

The formal proof of G1 to G6 shows that after successful implementation of this protocol, reader R and tag T with its ID would both trust the shared-key between them. Furthermore, the tag T trusts the random number RR which is sent by the reader is fresh, and the reader R trusts that the random number RTi which is sent by the tag is fresh.

4 Conclusions

RFID authentication protocol is the key guarantee for the safe and stable operation of RFID system. In the light of the analysis of the Hash-based RFID authentication protocols and those improved protocols, a novel lightweight RFID mutual authentication protocol based on Hash function is proposed, and the SVO logic verification and performance analysis of the protocol are carried out. The new protocol uses the random numbers and the hash function to transfer secret authentication information with a limited number for invalid authentication. Compared with the existing protocols, it

supports ownership transfer and quantity scalability of Tags, and has the characteristics of resisting spoofing attack, replay attack, tracking attack, anti asynchronous attack and privacy protection. So it offers good security and high application value. However, storage space of the Tag in the new protocol is slightly larger, and the computational load on the tag side will further be reduced so as to reduce costs of tags in the future work.

Acknowledgments. This research is supported by Natural Science Foundation of China (NSFC), under grant number 61370227, and by Union NSF of Hunan Province & Xiangtan City of China, under grant number: 2015JJ5034.

References

1. Zheng, Z., Mo, H.: Research and implication of RFID security authentication protocol. Master dissertation, Beijing Jiaotong University, Beijing, April 2014
2. Sun, X., Zhao, Z.: A Hash-based mutual authentication protocol for the RFID system. J. Hangzhou Dianzi Univ. **32**(6), 29–32 (2012)
3. Sarma, S.E., Weis, S.A., Engels, D.W.: RFID systems and security and privacy implications. In: Kaliski, B.S., Koç, ç.K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 454–469. Springer, Heidelberg (2003). doi:10.1007/3-540-36400-5_33
4. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and privacy aspects of low-cost radio frequency identification systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 201–212. Springer, Heidelberg (2004). doi:10.1007/978-3-540-39881-3_18
5. Ohkubo, D., 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 (SCIS 2004), Sendai, pp. 719–724 (2004)
6. Henrici, D.,Muller, P.: Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, pp. 149–153. IEEE (2004)
7. Rhee, K., Kwak, J., Kim, S., Won, D.: Challenge-response based RFID authentication protocol for distributed database environment. In: Hutter, D., Ullmann, M. (eds.) SPC 2005. LNCS, vol. 3450, pp. 70–84. Springer, Heidelberg (2005). doi:10.1007/978-3-540-32004-3_9
8. Molnar, D., Wagner, D.: Privacy and security in library RFID: issues, practices, and architectures. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington, DC, pp. 210–219 (2004)
9. Tsudik, G.: YA-TRAP: yet another trivial RFID authentication protocol. In: Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops. PerCom Workshops 2006, pp. 640–643. IEEE (2006)
10. Li, Z., Lu, G., Xin, Y.W.: A extensible authentication protocol based on Hash chain. Comput. Eng. **34**(4), 173–175 (2008)
11. Xiong, W., Xue, K., Hong, P., et al.: A RFID security protocol based on Hash chain in two-dimensional interval. J. China Univ. Sci. Technol. **41**(007), 594–598 (2011)
12. Yuan, S.-G., Dai, H.-Y., Lai, S.-L.: Hash-based RFID authentication protocol. Comput. Eng. **34**(12), 141–143 (2008)

13. Chen, S., Chen, R., Ling, L.: An improved Hash-function security protocol for RFID bidirectional authentication. Comput. Syst. Appl. **19**(3), 67–70 (2010)
14. Zhou, Y.: Research on RFID mutual authentication protocol based on Hash chain. Master dissertation, South West Jiaotong University (2012)
15. Liu, P., Zhang, C., Ou, Q.Y.: A Hash-based f mutual authentication security protocol for the mobile RFID. Design. Comput. Appl. **33**(5), 1350–1352 (2013)
16. Ding, Z., Li, J., Feng, B.: Research on Hash-based RFID security authentication protocol. J. Comput. Res. Dev. **46**(4), 583–592 (2009)
17. Si, C., Wen, G.: A design and implementation of RFID security authentication protocol based on Hash function. Master dissertation, University of Electronic Science and technology, Chengdu, December 2013
18. Song, B., Mitchell, C.J.: Scalable RFID security protocols supporting tag ownership transfer. Comput. Commun. **34**(4), 556–566 (2011)
19. Huang, Y.J., Yuan, C.C., Chen, M.K., et al.: Hardware implementation of RFID mutual authentication protocol. IEEE Trans. Ind. Electron. **57**(5), 1573–1582 (2010)
20. Kardas, S., Akgu, M., Kiraz, M.S., et al.: Cryptanalysis of lightweight mutual-authentication and ownership transfer for RFID systems. In: 2011 Workshop on Lightweight Security & Privacy: Devices, Protocols and Applications, pp. 20–25 (2011)
21. Syverson, P.F., van Oorschot, P.C.: On unifying some cryptographic protocol. In: Proceedings of the IEEE 1994 Computer Society Symposium on Security & Privacy. IEEE Computer Society, USA, pp. 14–28 (1994)
22. Syverson, P.F., van Oorschot P.C.: A unified cryptographic protocol logic. Technical report, NRL Publication 5540-227