Journal of Computer and Communications, 2016, 4, 1-9
Published Online June 2016 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2016.48001

The Vulnerability Analysis of Some Typical Hash-Based RFID Authentication Protocols

Zhicai Shi, Shitao Ren, Fei Wu, Changzhi Wang

School of Electronic & Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China
Email: szc1964@163.com

Received 17 April 2016; accepted 14 June 2016; published 17 June 2016

Abstract

The low-cost RFID tags have very limited computing and storage resources and this makes it difficult to completely solve their security and privacy problems. Lightweight authentication is considered as one of the most effective methods to ensure the security in the RFID system. Many lightweight authentication protocols use Hash function and pseudorandom generator to ensure the anonymity and confidential communication of the RFID system. But these protocols do not provide such security as they claimed. By analyzing some typical Hash-based RFID authentication protocols, it is found that they are vulnerable to some common attacks. Many protocols cannot resist tracing attack and de-synchronization attack. Some protocols cannot provide forward security. Győző Gódor and Sándor Imre proposed a Hash-based authentication protocol and they claimed their protocol could resist the well-known attacks. But by constructing some different attack scenarios, their protocol is shown to be vulnerable to tracing attack and de-synchronization attack. Based on the analysis for the Hash-based authentication protocols, some feasible suggestions are proposed to improve the security of the RFID authentication protocols.

Keywords

RFID, Authentication Protocol, Vulnerability, Hash Function, Security and Privacy

1. Introduction

With the development and application of the Internet of Things (IoT), 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 radio-waves to communicate, without visible light and physical contact. It is considered as a supplementary or replacement technology for traditional barcode technology. Today, the RFID system has been successfully applied to manufacturing, supply chain, agriculture, communication and

How to cite this paper: Shi, Z.C., Ren, S.T., Wu, F. and Wang, C.Z. (2016) The Vulnerability Analysis of Some Typical Hash-Based RFID Authentication Protocols. *Journal of Computer and Communications*, **4**, 1-9.
http://dx.doi.org/10.4236/jcc.2016.48001

transportation, health, e-payment, food safety tracing, and some other fields [1]. But the RFID tags only have limited computing and storage resources and they use open wireless channel to communicate. It is easy for an attacker to eavesdrop, intercept and tamper the sessions of the RFID system. Attackers can attack the RFID system by tracing, forging, spoofing, tampering and de-synchronizing. So the privacy and security of the RFID system has become one of the main factors to hinder its wide application. Although some physical methods have been proposed to solve the security and privacy of the RFID system, the research results show that it is the most flexible and effective method to use software encryption and authentication technique. The current popular tags are some low-cost passive tags and they have very limited computing and storage resources. They may be limited to hundreds of bits of storage, roughly between 5000 and 10,000 logic gates. Within these logic gates, only 250 to 3000 gates can be devoted to security functions [2]. So it is very difficult for a low-cost passive RFID tag to implement some complicated encryption algorithms. Therefore some lightweight authentication protocols are proposed to satisfy the special requirements of the RFID system. These protocols only use Hash function, CRC function, pseudorandom generating function, and some bitwise operations to complete the authentication of the RFID system. But they have still some flaws so that they cannot completely solve the security and privacy of the RFID system [3] [4]. So it is very necessary to research and analyze the current typical RFID authentication protocols so as to improve their security. We review some typical Hash-based lightweight authentication protocols. Then we focus on analyzing the authentication protocol proposed by Győző Gódor and Sándor Imre, which is simply called the G-I protocol.

Our main contributions are that we firstly find out the vulnerability of the G-I protocol. Then we construct two different attack scenarios to complete tracing attack and de-synchronization attack to the G-I protocol. We point out the reasons to result in the vulnerability of the G-I protocol. Finally, we propose some suggestions to overcome the weakness and vulnerability of the RFID authentication protocols so as to improve their security.

The paper is organized as follows. In Section 2, RFID systems and their security issues are introduced. In Section 3, some typical Hash-based lightweight authentication protocols are analyzed and their flaws are pointed out. In Section 4, the G-I protocol is analyzed in detail. Some attack scenarios are constructed. The analyzing results show the G-I protocol cannot resist tracing attack and de-synchronization attack. In Section 5, some suggestions are given out so as to improve the security of the RFID authentication protocols.

2. The RFID System, Its Security and Privacy

2.1. The Component of an RFID System

An RFID system usually consists of three components: Radio Frequency (RF) tag, RF reader and backend server, as shown in **Figure 1** [5]. A tag is basically a silicon chip with antenna and a small storage. For an RFID system, a tag is a special device. Its computing and storage resource is very limited. There are two main types of tags: active tag and passive tag. Active tags include miniature batteries used to power the tags and they are capable to transmit data over longer distance. Passive tags don’t have any battery and they are activated by the RF signal beamed from the reader. So passive tags are used for shorter range communication. This kind of tags is very cheap and they are usually called low-cost tags. These low-cost tags have become the most popular tags and they are widely used in many different fields.

A reader is a device capable of sending and receiving data in the form of radio frequency signal. This device is used to communicate with the tag and reads the identifier of the tag. A backend server is used to store the detail information about the tagged objects, and it cooperates with reader to implement the mutual authentication

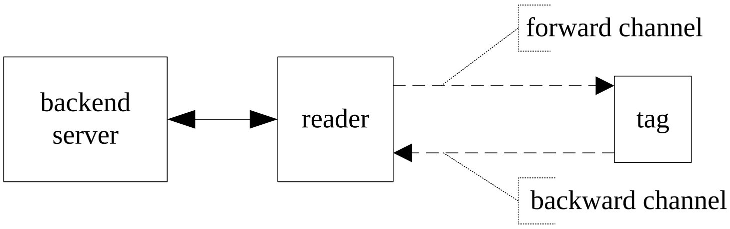

Figure 1. The component of an RFID system.

to tags. It searches the information about the tagged objects according to the tag's identifier. And it sends the information to the reader.

2.2. The Security and Privacy Issue of RFID Systems

As an important component of the RFID system, the tag usually has very limited computing and storage resources and it is difficult to implement some complicated cryptographic algorithms. But backend server and reader are usually considered to be resource-abundant and they can implement conventional cryptographic protocols effectively. So the channels between backend server and reader are secure and they are usually considered as a single entity, which is simply called the backend server/reader. However, because of the limited resources and the open wireless communication mode it has to assume that the channel between tag and reader is insecure. Readers have electric power enough to transmit signals over longer distance and tags only have limited electric energy to transmit signals over shorter 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. These two channels are open and insecure. Most secure problems of the RFID system are resulted from these insecure channels.

As a typical resource-constrained system, the RFID system is very vulnerable to some secure threats. Eavesdropping, impersonating, tracing, replaying and de-synchronization are some popular secure threats. Eavesdropping means that an adversary can intercept sessions between tag and reader by eavesdropping open wireless channel. These sessions are analyzed to reveal the secrecy about the tag. Once an adversary reveals the secrecy of the tag he can impersonate a legitimate tag to get the authentication from the backend server/reader. Tracing attack means that an adversary can identify which tag sends the messages intercepted by him and then he can trace the tag, even the objects or persons carried the tag. If a tag repeats to send some same messages during the authenticating process it is easy to be traced. Replay attack means that an adversary re-sends the previous sessions to impersonate a legitimate tag so as to get the authentication from the backend server/reader. De-synchronization attack means the backend server/reader and the tag cannot update their secret keys synchronously so that they possess different secret keys. This makes future authentication impossible. An adversary can implement de-synchronization attack by tampering, malicious blocking or abnormal closing the sessions between backend server/reader and tag.

Otherwise, a secure RFID system must satisfy forward security and anonymity. Forward security describes the capability to trace the events occurred during the past authentication process. Forward security guarantees that all authentication sessions, which happened before the tag's secret key is revealed, remain irrelevant. Once the current secret key is revealed, the previous authentication sessions are not deduced. So an RFID system has to regularly update its secret keys so as to satisfy forward security. Unfortunately, updating the secret keys of an RFID system often results in de-synchronization attack.

3. Some Typical Hash-Based RFID Authentication Protocols and Their Vulnerability

In order to solve the security and privacy problems of the RFID system, many RFID lightweight authentication protocols have been proposed in recent years. These authentication protocols usually use the one-way property of Hash functions to implement the confidentiality and anonymity of the RFID system. But most of them have serious security problems. 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, and S. E. Sarma *et al*. [6] 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. An adversary easily compromises mutual authentication by simply eavesdropping and replaying the sessions between tag and reader. So Hash-Lock protocol is vulnerable to spoofing attack and replay attack. Moreover, an adversary easily traces the tag's holder by the tag's identifier ID and its fixed pseudonym $metaID$.

In order to overcome the flaws of Hash-Lock protocol, S. A. Weis and S. E. Sarma *et al*. proposed randomized Hash-Lock protocol [6]. This protocol uses a pseudorandom number generator (PRNG) to randomize the transferred sessions between tag and reader. Tags respond to reader's queries by generating a random value *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 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 [7]. The aim of their protocol is to provide better protection for the user's privacy by refreshing the identifier of the tag. 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 Hash chain technique to renew the secret key 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 Hash chain length and *n* is the total number of tags. Thus, when the number of tags or the chain length is large the computation becomes unimaginable for an 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 to ensure forward security. This scheme requires each tag to support 2 Hash functions. When the tag is queried by a reader, it sends Hash value of its current identifier by using Hash function, $G()$, then renews its identity information by using another different Hash function, $H()$. These protocols use two different Hash functions and it is not suitable to the low-cost RFID tags.

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 secrecy of the tags.

Jung-Sik Cho *et al*. [11] proposed a new Hash-based authentication protocol to solve the secure and private problems for the RFID system. However, Hyunsung Kim [12] 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 attack. More precisely, an adversary can impersonate a valid tag or reader with probability 1/4. Finally, an adversary can obtain some information about the secrecy 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 [13] pointed out that an adversary can perform a de-synchronization attack by intercepting and tampering the transferred messages between tag and reader. 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.* [14] also constructed three different attacks to demonstrate Jung-Sik Cho *et al*.'s protocol is vulnerable to de-synchronization attack and tag/reader impersonation attack. 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*. [15] proposed a Hash-based RFID security protocol and proved that their protocol can provide forward privacy. However, Da-Zhi Sun and Ji-Dong Zhong [16] pointed out that an attacker can track a target tag by observing previous unsuccessful sessions. Da-Zhi Sun *et al*. justified that J. H. Ha *et. al*.'s protocol fails to provide forward privacy as they claimed. 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. They require more computing and storage cost. They are not suitable for the low-cost RFID system.

Liu Yang, Peng Yu *et al*. [17] proposed an RFID secure authentication protocol based on Hash function. 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 authenticating 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.

Győző Gódor and Sándor Imre [18] analyzed the typical Hash-based authentication protocols as described above. Then they proposed a Hash-based mutual authentication protocol for the low-cost RFID system, which is the G-I protocol. They claimed that their protocol provides an efficient mutual authentication. It can defy the well-known attacks and it provides stronger security than these protocols described above. But by analyzing, their protocol cannot prevent tracing attack and de-synchronization attack. We will focus on analyzing the G-I protocol in next section.

4. Review and Analysis of the G-I Protocol

4.1. The G-I Protocol

For the G-I protocol, the tag stores its secret keys $k1$ and $k2$. The backend server/reader stores the secret keys of all tags: $k1c$, $k1p$, $k2$ and their Hash values: $h(k1c)$, $h(k1p)$. $k1c$ is the current secret key. $k1p$ is the secret key of the last successful authentication. $h(\ )$ is a Hash function. The backend server/reader and the tag can implement Hash function and pseudorandom number generating operation. The used symbols in the G-I protocol are listed in **Table 1**. This protocol is shown in **Figure 2** and it is described as follows:

1. The backend server/reader sends a message, Query, to the tag.

2. After receiving the message, Query, the tag computes $h(k1)$ and sends it to the backend server/reader.

3. The backend server/reader tries to look for the received $h(k1)$ in its database by replacing $k1$ with $k1c$ and $k1p$ respectively.

In case it is found, the backend server generates a random number $r1$ and computes $t1 = h(k1 \oplus k2 \oplus r1)$, then it sends $r1$ and $t1$ to the tag.

4. After the tag receives $r1$ and $t1$ it computes $t1' = h(k1 \oplus k2 \oplus r1)$. If $t1 = t1'$ then it authenticates the backend server/reader. Then the tag generates another random number $r2$ and computes $t2 = h(k2 \oplus r1 \oplus r2)$. The tag sends $r2$ and $t2$ to the backend server/reader.

5. The backend server/reader receives $r2$ and $t2$. It computes $t2' = h(k2 \oplus r1 \oplus r2)$. If $t2 = t2'$ then it authenticates the tag. After completing the authentication to the tag the backend server/reader updates its secrecy as follows:

If $h(k1c) = h(k1)$, then:

$$k1p = k1c, k1c = h(k1c \oplus r1).$$
$$k2 = h(k2 \oplus r2).$$

If $h(k1p) = h(k1)$, then:

$$k1c = h(k1p \oplus r1).$$
$$k2 = h(k2 \oplus r2).$$

6. After the backend server/reader has updated its secret keys, it sends "$\text{Update} - \text{key}$" to the tag. The tag receives "$\text{Update} - \text{key}$" and it updates its secret keys as follows:

Table 1. The symbols used in the G-I protocol.

Notation	Description
$k1, k2$	The tag's secret keys stored in the tag
$k1c, k1p$	The current secret keys of the tag and its last secret keys stored in the backend server/reader
$h(\ )$	A secure cryptographic Hash function
$r1, r2$	Two random numbers generated by the backend server/reader and the tag
$\oplus$	Bitwise XOR operation

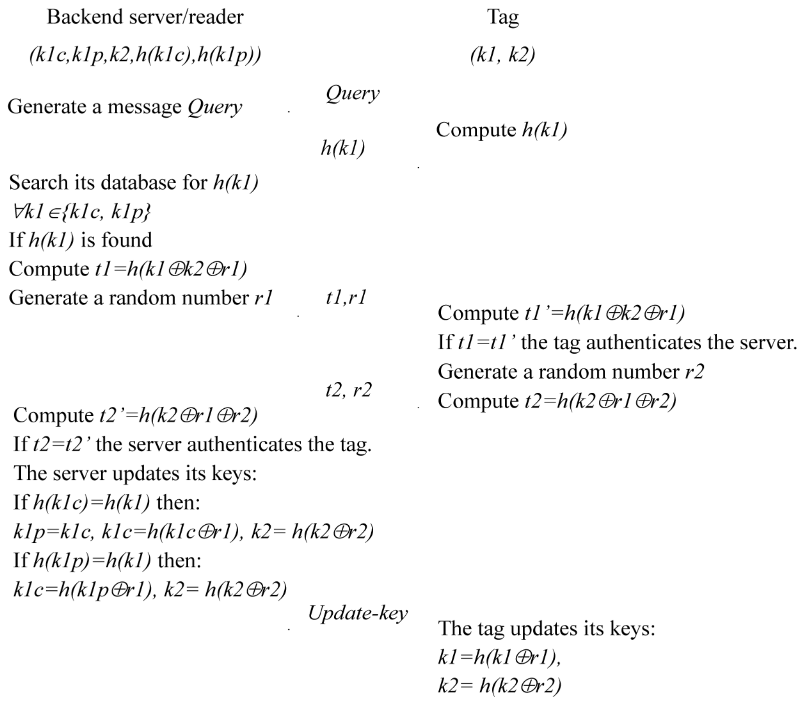
Backend server/reader
(k1c,k1p,k2,h(k1c),h(k1p))
Tag
(k1, k2)
Generate a message Query
Query
Compute h(k1)
h(k1)
Search its database for h(k1)
∀k1 ∈{k1c, k1p}
If h(k1) is found
Compute t1=h(k1⊕k2⊕r1)
Generate a random number r1
t1,r1
Compute t1'=h(k1⊕k2⊕r1)
If t1=t1' the tag authenticates the server.
Generate a random number r2
Compute t2=h(k2⊕r1⊕r2)
t2, r2
Compute t2'=h(k2⊕r1⊕r2)
If t2=t2' the server authenticates the tag.
The server updates its keys:
If h(k1c)=h(k1) then:
k1p=k1c, k1c=h(k1c⊕r1), k2= h(k2⊕r2)
If h(k1p)=h(k1) then:
k1c=h(k1p⊕r1), k2= h(k2⊕r2)
Update-key
The tag updates its keys:
k1=h(k1⊕r1),
k2= h(k2⊕r2)

Figure 2. The diagram of the G-I protocol.

$$k1 = h(k1 \oplus r1).$$
$$k2 = h(k2 \oplus r2).$$

4.2. The Vulnerability Analysis of the G-I Protocol

Győző Gódor and Sándor Imre claimed that their protocol can resist eavesdropping, replaying, tracing and spoofing. It is very strong against de-synchronization attack and it provides forward and backward security. But by analyzing, it is found that their protocol is vulnerable to de-synchronization attack and tracing attack. The G-I protocol doesn't provide their claimed security. One reason, which results in the vulnerability of the protocol, is that the protocol cannot keep the freshness of the sessions between backend sever/reader and tag. Another reason is the worse property of exclusive OR operation and the messages, Query and $\text{Update} - \text{key}$, are not signed by their sender before they are sent.

- Tracing attack.

In order to enhance the scalability and anonymity of the G-I protocol, $h(k1)$ is used as a pseudonym to be sent to the backend server/reader so as to declare the identity of the tag. But this makes the protocol vulnerable to tracing attack. The process of tracing attack is described as follows:

(1) The attacker masquerades a legitimate backend server/reader and sends Query to the tag.

(2) After the tag receives Query it computes $h(k1)$ and sends $h(k1)$ to the attacker.

(3) The attacker blocks the later authentication process or the last step, namely, the attacker prevents "$\text{Update} - \text{key}$" to be sent to the tag. So the tag cannot update its secret keys.

(4) The attacker masquerades a legitimate backend server/reader again, and sends Query to the tag.

(5) After the tag receives Query it will compute and return the same $h(k1)$ to the attacker.

(6) Repeating the above process, the attacker can locate the tag which sends the same $h(k1)$. Therefore tracing attack happens.

The vulnerability of the G-I protocol to tracing attack is that the tag cannot keep the freshness of the sessions which it sends to the backend server/reader. If the tag cannot update its secrecy in time or it does not randomize the response to the backend server/readeran attacker can easily trace it by the fixed $h(k1)$.

- De-synchronization attack.

Győző Gódor and Sándor Imre claimed that their protocol is very strong against de-synchronization attack. But by analyzing, it is found that the G-I protocol cannot resist de-synchronization attack [19]. An attack scenario is constructed as follows:

(1) The attacker masquerades a legitimate backend server/reader and sends Query to the tag. Then it gets $h(k1)$ from the tag.

(2) The attacker masquerades a legitimate tag and sends $h(k1)$ to the backend server/reader. Then it gets $r1$ and $t1$ from the backend server/reader.

(3) The attacker masquerades a legitimate backend server/reader again, and sends $r1$ and $t1$ to the tag. Then it gets $r2$ and $t2$ from the tag. It keeps $r2$, $t2$ and does not send them to the backend server/reader. The backend server/reader does not update its secret keys because it does not receive $r2$ and $t2$. So its current secret keys are kept. Then the attacker sends the message "$\text{Update}-\text{key}$" to the tag.

(4) After the tag receives "$\text{Update}-\text{key}$" from the attacker it begins to update its secret keys as follows:

$$k1 = h(k1 \oplus r1).$$
$$k2 = h(k2 \oplus r2).$$

(5) Later, once the attacker receives Query from the backend server/reader he masquerades a legitimate tag and replays $h(k1)$ to the backend server/reader. The backend server/reader can find the matched $h(k1)$ in its database because its secret keys are not updated. Then it generates $r1'$, $t1'$ and sends them to the attacker.

(6) The attacker receives $r1'$ and $t1'$, then it constructs $r2'$ and $t2'$ as follows:

$$r2' = r1' \oplus r1 \oplus r2.$$
$$t2' = h(k2 \oplus r1' \oplus r2') = h(k2 \oplus r1' \oplus r1' \oplus r1 \oplus r2) = h(k2 \oplus r1 \oplus r2) = t2.$$

(7) The attacker sends $r2'$ and $t2'$ to the backend server/reader. After the backend server/readerproves that $r2'$ and $t2'$ are legitimate it begins to update its secrecy as follows:

$$k1p = k1c, k1c = h(k1c \oplus r1').$$
$$k2 = h(k2 \oplus r2').$$

It is obvious that the secrecy between the backend server/reader and the tag are different. De-synchronization attack occurs.

Moreover, there is another simple attack scenario to result in de-synchronization attack for the G-I protocol, which is that an attacker intercepts "$\text{Update}-\text{key}$" and he does not send it to the tag. Because the tag does not receive "$\text{Update}-\text{key}$" it cannot update its secret keys, $k1$ and $k2$. But the backend server/reader updates its secret keys, $k1c$, $k1p$ and $k2$. In this case, $k2$ of the backend server/reader is updated and it is different from $k2$ of the tag. This makes the protocol cannot complete the later authentication. So de-synchronization attack occurs.

5. Conclusions

It is a great challenge to design a lightweight authentication protocol which is secure and efficient for the low-cost RFID system. In this paper, we analyze some typical Hash-based lightweight authentication protocols and the G-I protocol, and find these protocols are not as secure as they claimed. For the G-I protocol, we demonstrate that an adversary can trace a tag by repeating to send "Query" and blocking the later authentication process. An adversary can masquerade a legitimate tag or a backend server/reader to tamper or counterfeit some sessions and to replay them so that the tag and the backend server/reader cannot update their secret keys synchronously. For overcoming the weakness of the RFID authentication protocols, some feasible suggestions are given out:

(1) In order to resist tracing attack, the response of a tag to the backend server/reader must be randomized by a random number, which is generated by the tag. When a tag receives a different query from the backend server/reader it should give a different response. Therefore the freshness of the sessions between tag and backend server/reader is kept so that an adversary cannot distinguish a tag by the intercepted sessions.

(2) In order to resist de-synchronization attack, the tag or the backend server/reader begins to update its

secrecy if and only if it successfully implements the authentication to its partner. Otherwise, the tag begins to update its secrecy if and only if the backend server/reader has updated its secrecy. It is avoided for a tag to update its secret keys before the backend server/reader updates its secrecy.

Acknowledgements

We are appreciated to anonymous reviewers for their constructive suggestion to this paper so that we can improve it. This research work was supported by the National Natural Science Foundation of China under Grant No. 61272097.

References

[1] Eetu, P.-S., Karri, R. and Ville, H. (2014) The European Approach to Addressing RFID Privacy. *International Journal of Radio Frequency Identification Technology and Applications*, **4**, 260-271. http://dx.doi.org/10.1504/IJRFITA.2014.063923

[2] Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M. and Ribagorda, A. (2006) RFID Systems: A Survey on Security Threats and Proposed Solutions. *IFIP International Federation for Information Processing* 2006 (*PWC* 2006), **4217**, 159-170. http://dx.doi.org/10.1007/11872153_14

[3] Dehkordi, M.-H. (2014) Improvement of the Hash-Based RFID Mutual Authentication Protocol. *Wireless Personal Communications*, **75**, 219-232. http://dx.doi.org/10.1007/s11277-013-1358-7

[4] Hermans, J. and Peeters, R. (2014) Proper RFID Privacy: Model and Protocols. *IEEE Transactions on Mobile Computing*, **13**, 2888-2902. http://dx.doi.org/10.1109/TMC.2014.2314127

[5] Dimitriou, T. (2016) Key Evolving RFID Systems: Forward/Backward Privacy and Ownership Transfer of RFID Tags. *Ad Hoc Networks*, **37**, 195-208.

[6] Weis, S.A., Sarma, S.E., Rivest, R.L. and Engels, D.W. (2003) Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. *Proc. of the* 1*st International Conference on Security in Pervasive Computing*, Boppard, 12-14 March 2003, 201-212.

[7] Ohkubo, M., Suzuki, K. and Kinoshita, S. (2004) Hash-Chain Based Forward Secure Privacy Protection Scheme for Low-Cost RFID. *Proc. of the* 2004 *Symposium on Cryptography and Information Security*, Sendai, 27-30 January 2004, 719-724.

[8] Yeo, S.-S. and Kim, S.-K. (2005) Scalable and Flexible Privacy Protection Scheme for RFID Systems. *European Workshop on Security and Privacy in Ad Hoc and Sensor Networks ESAS*'05, Visegrad, 13-14 July 2005, 153-163. http://dx.doi.org/10.1007/11601494_13

[9] Lee, Y.K. and Verbauwhede, I. (2005) Secure and Low-Cost RFID Authentication Protocols. *Proc. of the* 2*nd IEEE Workshop on Adaptive Wireless Networks*, St. Louis, 28 November-1 December 2005, 1-5.

[10] Lee, S.M., Hwang, Y.J., Lee, D.H. and Lim, J.I. (2005) Efficient Authentication for Low-Cost RFID Systems. *LNCS*, **3480**, 619-627.

[11] Cho, J.-S., Jeong, Y.-S. and Sang, O.P. (2012) Consideration on the Brute-Force Attack Cost and Retrieval Cost: A Hash-Based Radio-Frequency Identification (RFID) Tag Mutual Authentication Protocol. *Computers and Mathematics with Applications*, **3**, 1-8.

[12] Kim, H. (2012) Desynchronization Attack on Hash-Based RFID Mutual Authentication Protocol. *Journal of Security Engineering*, **9**, 357-365.

[13] Khedr, W.I. (2013) SRFID: A Hash-Based Secure Scheme for Low Cost RFID Systems. *Egyptian Informatics Journal*, **14**, 89-98. http://dx.doi.org/10.1016/j.eij.2013.02.001

[14] Safkhani, M., Peris-Lopez, P., Hernandez-Castro, J.C. and Bagheri, N. (2014) Cryptanalysis of the Cho *et al*. Protocol: A Hash-Based RFID Tag Mutual Authentication Protocol. *Journal of Computational and Applied Mathematics*, **259**, 571-577. http://dx.doi.org/10.1016/j.cam.2013.09.073

[15] Ha, J.H., Moon, S.J., Zhou, J.Y. and Ha, J.C. (2008) A New Formal Proof Model for RFID Location Privacy. *Proc. of the* 13*th European Symposium on Research in Computer Security—ESORICS*'08, Malaya, 6-8 October 2008, 267-281. http://dx.doi.org/10.1007/978-3-540-88313-5_18

[16] Sun, D.-Z. and Zhong, J.-D. (2012) A Hash-Based RFID Security Protocol for Strong Privacy Protection. *IEEE Transactions on Consumer Electronics*, **58**, 1246-1252. http://dx.doi.org/10.1109/TCE.2012.6414992

[17] Liu, Y., Peng, Y., Wang, B.L., Qu, Y. and Bai, X.F. (2013) Hash-Based RFID Mutual Authentication Protocol. *International Journal of Security and Its Applications*, **7**, 183-194.

[18] Gódor, G. and Imre, S. (2012) Hash-Based Mutual Authentication Protocol for Low-Cost RFID Systems. *LNCS*, **7479**, 76-87.

[19] Wang, S. and Liu, S.J. (2013) Scalable RFID Mutual Authenticatio Protocol with Backward Privacy. *Journal of Computer Research and Development*, **50**, 1276-1284.