An Improved Lightweight RFID Protocol to Protect against Desynchronization Attack

HE Lei[1], LU Zhongning[1], WANG Pengyuan[1], DANG Jianliang[2], YU Miao[3]

1. School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, China

2. College of Sciences, Henan Agricultural University, Zhengzhou, China

3. School of Electronic and Information Engineering, Zhongyuan Institute of Technology, Zhengzhou, China

Abstract: With the RFID system widely used, lightweight authentication protocols between the tag and backend database are necessary because this channel is insecure. Moreover, the low-cost tags only have limited computational resources and does not implement symmetric or asymmetric encryption algorithm. Many researchers proposed some lightweight authentication protocols which only use lightweight operations, such as hash function, XOR etc. In this paper, we analyze two protocols' security proposed by Dimitriou and find they can not provide against desynchronization attack. Afterwards, we propose an improved lightweight protocol and analyze its property. The result indicates that this protocol does not increase the computation complexity and provide two-way authentication, intractability, forward security and prevent against replay attack, clone attack, desynchronization attack.

Keywords: authentication protocol; hash operation; desynchronization attack; RFID

1. Introduction

Radio frequency identification is a technology which is used to access the information about the object without physical contact. In a typical RFID system, it consists of three major parts: tags, reader and backend database. The tags are attached to merchandise and store information including identification, manufacturing location, and so on. The reader is responsible for transmitting messages between the tag and backend database. The backend database provides various services and is powerful in computational capacity.

RFID technology is considered to be a replacement of bar-codes because it has many merits, such as longer access distance, longer life time, larger memory etc. It can increase the item management efficiency during the process of manufacture, logistics and retail. For example, logistics providers can track the merchandise location and retailers can check how many merchandise in its warehouse through broadcasting a query signal.

As tags have been pervasive computing equipment, more and more people concern about information security issues in the RFID system. Generally speaking, we consider the channel between tags and reader insecure while the channel between the reader and backend database secure because the former is wireless channel and low-cost tags only have limited computational resource and can implement simple operation, such as hash function, XOR operation etc. instead of implement symmetric or asymmetric encryption algorithm. An adversary can eavesdrop the communication between the tag and reader. Moreover, the tag may leak carrier's location privacy because it can be attached to product, animal or person.

Many scholars engage in the research of protecting the information security and privacy in the tag-reader channel. They proposed some lightweight security protocols used in RFID system, such as Hash-Lock protocol [1,2] and randomized Hash-Lock protocol [3] etc.

In 2005, Dimitriou proposes a simplified and an enhanced lightweight RFID protocols to protect against traceability and cloning attack and analyzes their properties. It is found the simplified protocol is vulnerable to replay attack and desynchronization attack. Hence Dimitriou proposes another enhanced lightweight protocol in order to protect against these attacks. It is pointed out that the secret information, namely, tag's ID, needs to update after every times authentication. Nevertheless, it does not

This paper is sponsored by the Science and Technology Plan of Henan Province (No.0624220084), Zhengzhou (No.074SCCG23109-2), Education Department of Henan province (No. 2008A120011, 2008A510018) and Science Research Fund of Zhengzhou University of Light Industry.

978-1-935068-06-8

explain explicitly the ID updating scheme. In this paper, we analyze the security of protocols proposed by Dimitriou and find its security vulnerabilities. It does not define explicitly ID update scheme and is vulnerable to desynchronization attack. Afterwards, we proposed an improved lightweight RFID protocol to protect against desynchronization attack.

2. Security Protocols Proposed by Dimitriou

In this paper, these notations and operations below are used.

H(x): a secure hash functions of x
a, b: conjunction of variable a and b
ID: a confidential identification of tag
$H_{ID}(x)$: a keyed hash function with ID

2.1 A Simplified Protocol

2.1.1 Protocol Description

During system initialization, the tag is assigned a unique identification ID_0 which is a random number. Similarly, the backend database also stores this identification ID_0 and $H(ID_0)$. Afterwards, this protocol is described as Figure 1 and follows these steps below [4].

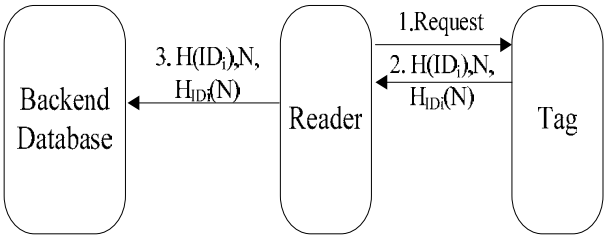

Figure 1. A simplified protocol

1) The reader sends a query request to the tag.

2) After receiving the request message, the tag generates a new nonce N and sends $H(ID_i)$, N and $H_{IDi}(N)$ to the reader. Afterwards, the tag updates its ID_i to ID_{i+1}.

3) The reader forwards $H(ID_i)$, N and $H_{IDi}(N)$ to backend database.

4) The backend database searches a proper ID whose hash value is equal to $H(ID_i)$ received. If such ID exists, the backend database computes $H_{IDi}(N)$ using it and check whether it is equal to the received one. If it is, the backend database authenticates the tag successfully and updates the ID_i to ID_{i+1}. Otherwise, this protocol stops.

2.1.2 Security Analysis

We analyze its security as follows.

•Information leakage

In this protocol, the tag's ID is confidential. An adversary does not deduce ID even if he eavesdrops all messages transmitted between the tag and reader because hash function is not invertible.

•Authentication

Obviously, this protocol does not provide mutual authentication. In this protocol, the backend database authenticates the tag while the tag does not authenticate backend database.

•Intractability

In this protocol, the adversary does not obtain any ID information about the tag even though it eavesdrops all messages. Moreover, the value of H(ID) is different in every communication because tag's ID updates after every communication. Hence the adversary can not trace the tag's holder.

•Replay attack

This protocol uses a nonce N which is not repeating to protect against replay attack. If the adversary simply replays these messages he eavesdropped, the backend database will find there is a replay attack by checking the value of N and $H_{IDi}(N)$.

•Forward security

The definition of forward security is competing in some papers [5,6]. Here we think forward security means it does not compromise the previous confidential information even if the adversary obtains the current confidential information. In essence, most of forward security issues come from the confidential information which is never updated. In this paper, the confidential information ID_i is updated to ID_{i+1} after every time authentication successfully through irreversible computation. However, it does not explicitly explain how to update ID in this paper.

•Desynchronization attack

This protocol is vulnerable to desynchronization attack because there is not a synchronization scheme of the ID stored in the tag and backend database. The adversary can interfere with the communication between the tag and database or impersonate a valid tag to make the ID stored in the tag and database desynchronized.

•Clone attack

In this protocol, the adversary can eavesdrop the messages transmitted between the tag and backend database. However, it does not deduce the tag's ID_i from $H(ID_i)$ because hash operation is nonreversible. Accordingly, the adversary does not obtain the tag's ID and duplicate this tag.

2.2 An Enhanced Protocol

2.2.1 Protocol Description

Considering the security weakness above mentioned, Dimitriou proposed an enhanced protocol which is illustrated in Figure 2.

1) The reader sends a query request and a nonce N_R to the tag.

2) After receiving the request message, the tag generates a new nonce N_T and sends $H(ID_i)$, N_T and $H_{IDi}(N_T,N_R)$ to the reader.

3) The reader forwards $H(ID_i)$, N_T and $H_{IDi}(N_T,N_R)$ to backend database.

4) The backend database searches a proper ID whose hash value is equal to $H(ID_i)$ received. If such ID exists, the backend database computes $H_{IDi}(N_T,N_R)$ using it and check whether it is equal to the received one. If it is, the backend database authentication this tag successfully and updates the ID_i to ID_{i+1}. Afterwards, the backend database sends $H_{ID(i+1)}(N_T,N_R)$ to the tag through reader. Otherwise, this protocol stops.

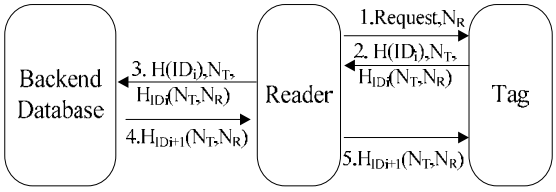

Figure 2. An enhanced protocol

5) The tag computes $H_{ID(i+1)}(N_T,N_R)$ and checks whether it is equal to the received one. If it is equal, the tag authentications the backend database successfully and updates its ID_i to ID_{i+1}. Otherwise, this protocol stops.

2.2.2 Security Analysis

We analyze this enhanced protocol's security in the same way with the simplified protocol. Obviously, this enhanced protocol can provide mutual authentication, forward security, and intractability and prevent against replay attack, clone attack. Additional, there is not information leakage. However, it also can not prevent against desynchronization attack; namely, it is lack of a synchronization scheme. If the backend database authenticates the tag successfully while the tag does not think this backend database authentic, the IDs stored in the both sides are desynchronizing.

3. An Improved Protocol to Protect against Desynchronization Attack

3.1 An Improved Lightweight Authentication Protocol

In this protocol, we add two steps which are responsible for providing tag's feedback about ID updating. Actually, we divide the entire process into two parts. One is the authentication process which include step (1) to (5), the other is updating process which include step (6) and (7). This protocol is presented in Figure 3 and follows these steps below.

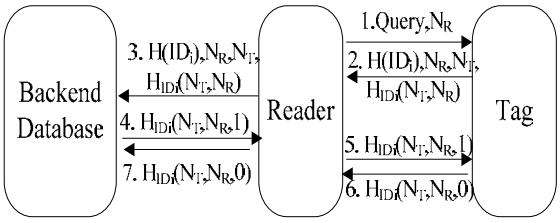

Figure 3. Improved lightweight protocol

1) The reader generates a nonce N_R and sends query request and N_R to tag.

2) After receiving the query message, the tag computes $H(ID_i)$ and $H_{IDi}(N_T, N_R)$. Afterwards, it transmits $H(ID_i)$, N_R, N_T and $H_{IDi}(N_T, N_R)$ to the reader. Simultaneously, it stores N_R and N_T in its memory.

3) The reader forwards $H(ID_i)$, N_R, N_T and $H_{IDi}(N_T, N_R)$ to the backend database.

4) After receiving these messages, the backend database stores N_T. Afterwards, it determines whether this tag is authentic or not in the two ways below.

a) It checks whether there is an ID_i stored which is used to compute $H(ID_i)$ and $H_{IDi}(N_T, N_R)$ and makes the computational result is equal to the received messages.

b) It computes $ID_i = H(ID_i, N_T)$ and checks whether this

978-1-935068-06-8 © 2009 SciRes.

new ID_i meets the same requirement above mentioned.

The backend database believes this tag is authentic provided that it finds an ID which meets one of requirements mentioned in a) and b). Next it computes $H_{IDi}(N_T, N_R, 1)$ and sends the computation result to the reader.

Otherwise, it does not authenticate this tag and terminates this protocol.

5) The reader forwards $H_{IDi}(N_T, N_R, 1)$ to the tag.

6) The tag computes whether the $H_{IDi}(N_T, N_R, 1)$ received is correct or not. If it is correct, the tag computes $H_{IDi}(N_T, N_R, 0)$ and transmits it to the backend database through the reader. Afterwards, it updates $ID_{i+1}=H(ID_i, N_T)$.

7) After receiving $H_{IDi}(N_T, N_R, 0)$, the backend database check whether it is correct or not. If it is correct, it updates the corresponding ID_i as $ID_{i+1}=H(ID_i, N_T)$. Otherwise, it does not update this ID_i.

3.2 Protocol Analysis

We think this protocol is efficient because it is only used hash function in order to keep low-cost and does not increase computation complexity obviously compared with the two protocols proposed by Dimitriou.

In the field of security, it is used seven steps to accomplish the authentication and ID updating process. Obviously, this improved protocol can provide mutual authentication, forward security, and intractability and prevent against replay attack, clone attack, which is like the enhanced protocol aforementioned. The biggest difference between the two protocols is the former can protect against desynchronization attack.

In this protocol, the tag updates its ID in advance and sends a feedback to the backend database. After receiving this feedback, the backend database determines whether to updates its corresponding ID_i or not. If the adversary interferes the communication in step (6) or backend database think the feedback is not correct, the ID stored in the backend database will not be updated, namely, the two IDs stored in the backend database and tag are not synchronous. In this case, the backend database can use the second rule in the step (4) to update its corresponding ID and decide whether to authenticate this tag successfully or not if this tag asks for authentication once again. In a word, the backend database can authenticate a valid tag and provide service even if this protocol is subjected to desynchronization attack.

4. Conclusions

In this paper, we analyze the security of a simplified protocol and an enhanced protocol proposed by Dimitriou. The former provide one-way authentication, intractability, forward security and prevent against replay attack, clone attack. Moreover, there is not information leakage. However, desynchronization attack comprises this protocol. The latter can provide mutual authentication except these security properties above mentioned. Nevertheless, it is vulnerable to desynchronization attack, too. Both of the two protocols are lack of an ID synchronization scheme. We propose an improved protocol with an ID synchronization scheme which can protect from desynchronization attack and analyze its security. The result indicates that this improved protocol only used hash function in order to keep low-cost and does not increase computation complexity obviously. In the field of security, it can provide two-way authentication, intractability, forward security and prevent against replay attack, clone attack, desynchronization attack. Next our work is to research how to further reduce the computation load in this improved protocol.

References

[1] S. E. Sarma, S. A. Weis, and D. W. Engels, RFID systems and security and privacy implications [J], Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems. Lectures Notes in Computer Science. Berlin, 2003, 2523, 454-469.

[2] S. E. Sarma, S. A. Weis, and D. W. Engels, Radio-Frequency identification: Secure risks and challenges [J], RSA Laboratories Cryptobytes, 2003, 6, 2-9.

[3] S. A. Weis , S. E. Sarma , R. L. Rivest , D. W. Engels, Security and privacy aspects of low-cost radio frequency identification systems [C], Proceedings of the 1st International Conference on Security in Pervasive Computing. Lectures Notes in Computer Science. Berlin, 2004, 2802, 201-212.

[4] Tassos Dimitriou, A Lightweight RFID protocol to protect against traceability and cloning attacks [J], Security and Privacy for Emerging Areas in Communications Networks, 2005, 59-66.

[5] Chae Hoon Lim, Taekyoung Kwon, Strong and Robust RFID Authentication Enabling Perfect Ownership Transfer [J], Information and Communication Security, 2006, 4307, 1-20.

[6] R. Shirley, Internet Security Glossary, Version 2[S], 2007, RFC 4949.