

n-Time Protocol for Multiple RFID Tags Authentication

Chang-Hyun Cho*, Jaesik Lee*, Jae-Woo Kim*, Moon-Seog Jun*

* Dept. of Computer Science, SoongSil University, Seoul, South Korea

hist0001@naver.com, j30231@ssu.ac.kr, saypeace@ssu.ac.kr, mjun@ssu.ac.kr

Abstract— Ari Juels proposed yoking-proof protocol for authenticating multiple tags using RFID system. Then, one-time yoking-proof protocol, which is a lightweight version using minimalist MACs was proposed. However, Ari Juels's protocol is vulnerable to replay attack during the handshake process for authentication between tags. Some solutions have been proposed for making up for the vulnerability but those also had some weaknesses in some ways. In this study, a protocol has been proposed for making up for the vulnerability of one-time yoking-proof protocol while allowing multiple authentications instead of one, depending on the tag capacity.

Keywords— n-Time, RFID, Authentication, Multiple, MAC

I. INTRODUCTION

RFID (Radio Frequency Identification) is a technology for recognizing number of objects using radio frequency. It consists of a tag with ID, a reader that reads the tag and a back-end DB that manages ID information. RFID tag is classified into active tag and passive tag, and this study is related to passive tag. RFID tag is vulnerable if its authentication process is not secured. Also, due to limited capability of tag, the authentication of tag is difficult in terms of technology and cost.

There have been many solutions proposed for multiple tags authentication. One of them was from Ari Juels who proposed a way to efficiently authenticate multiple tags using yoking-proof. However, the standard version of the yoking-proof requires a tag to deploy MAC (Message Authentication Code) and it is cost ineffective. Then, Juels proposed one-time yoking-proof, which is the lightweight version. It has simplified the handshake process between tag and reader and it uses one-time authentication using minimalist MAC.

However, the one-time yoking-proof protocol using minimalist MAC is very vulnerable to replay attack. Because only one authentication is allowed, if the first authentication fails, the second one will be vulnerable in terms of security. In effort to make up for such vulnerability, a study for using timestamp was proposed but in the structure of timestamp, time value will be increased and it is still vulnerable to replay attack.

In this study, a protocol has been proposed where it allows multiple authentication depending on the type of tag

while being well protected against replay attack. The composition of this study is as follows; Chapter 2 will briefly describe the existing studies related to RFID and Chapter 3 will illustrate the protocol proposed in this study, whereas Chapter 4 will analyze the security of the proposed protocol and Chapter 5 will be a conclusion.

II. EXISTING STUDIES RELATED TO RFID

A. Terms and symbols

The following terms and symbols will be used to describe the protocol.

- T_A : Refers to the RFID Tag A
- T_B : Refers to the RFID Tag B
- C_A : A counter value of T_A
- C_B : A counter value of T_B
- X_A : A secretkey value of T_A
- X_B : A secretkey value of T_B
- f : Hash function
- $MAC_{key}[v1, v2, \dots]$: Use key hash values($v1, v2, \dots$) to generate a message authentication code, similar to the function f
- $mMAC_{key}[v1, v2, \dots]$: minimalist MAC
- α / β : Intermediate result values
- P_{AB} : Finally, when comparing the DB, the data transmitted includes information for authentication

B. Related studies

1) Yoking-proof protocol standard cryptographic

Ari Jules's yoking-proof standard cryptographic protocol is shown in Figure 1 and its authentication process is as follows;

Reader requests authentication to T_A through 'left proof'. Then, T_A hashes C_A , the current counter value to X_A , the secret key, to produce r_A , a random value. A , the T_A identification value, r_A , the random value and C_A , the counter value are gathered into α and sent to the reader.

T_B sends to the reader the newly produced m_B and C_B , the counter value and B , the identification value of T_B , where T_B increases C_B by 1.

The reader requests authentication to T_B together with α received from T_A through 'right proof'. From α and C_B , T_B

produces m_B , the MAC value using X_B . The reader sends β to T_A and T_A uses X_A to produce m_{AB} , which is the MAC value of α and β . And T_A increases C_A by 1.

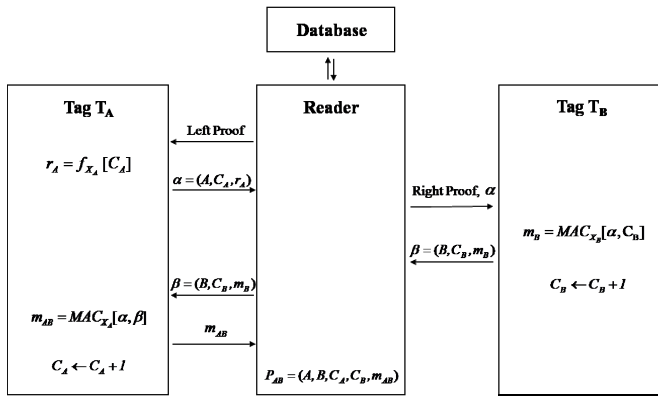


Figure 1 Yoking-proof protocol standard cryptographic

The newly produced m_{AB} is sent to the reader again and P_{AB} , the authentication value is sent proving that T_A and T_B had been recognized simultaneously. P_{AB} consists of A and B , which are the identification values of T_A and T_B , respectively, and C_A and C_B , which are the current counter value of the tags, and m_{AB} , the message authentication code value.

In yoking-proof protocol standard cryptographic, MAC function is used for message authentication. However, because deployment of MAC function in RFID is costly, Ari Juels proposed one-time yoking-proof protocol with minimalist MACs.

2) One-time yoking-proof protocol minimalist MACs

One-time yoking-proof protocol using minimalist MACs is shown in Figure 2 and its authentication process is as follows;

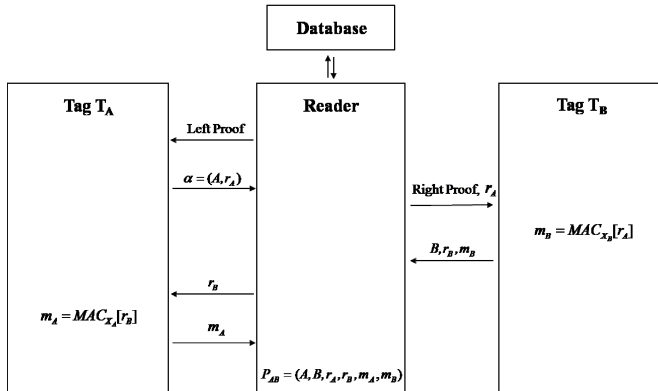


Figure 2 One-time yoking-proof protocol minimalist MACs

When the reader requests authentication to T_A through 'left proof', T_A sends r_A to the reader. Here, unlike 2.2.1, r_A is not produced by T_A , but the random value produced previously is stored in the tag.

The reader requests authentication to T_B together with r_A value through 'right proof'. T_B generates an authentication code by using the private key X_B for $mMAC$ function.

$mMAC$ uses key bits 0 and 1 to produce the previously defined authentication code value.

For instance, if there is $mMAC$ function that expresses the output of 8 bits, X_i will be as shown as Table 1 and when "11001001" is input as a function value, the output will be "11111010". If a vicious user inputs the function value where all the bits are "0" and "1", MAC table of X_i can be known easily. Therefore, when producing authentication code using $mMAC$, the authentication code should be produced and used only for once.

Table 1 MAC Table of X_i

X_i	1	2	3	4	5	6	7	8
bit 0	1	0	1	1	0	0	1	1
bit 1	1	1	0	1	1	1	1	0

T_B is produced previously together with m_B and it sends r_B , the one-time value produced by T_B to the reader. And the reader sends r_B to T_A . T_A generates an authentication code m_A by using the private key X_A for $mMAC$ function.

Finally, T_A sends P_{AB} value to the reader proving that T_A and T_B had been recognized simultaneously. P_{AB} value consists of A and B which are the identification values of T_A and T_B , respectively, and C_A and C_B , the current counter value of the tags, and m_{AB} , the message authentication code values.

However, One-time yoking-proof protocol minimalist MACs is vulnerable to replay attacks and the verifier is unable to know if r_A and r_B , the one-time production values are actually produced by the tag.

The replay attack on one-time yoking-proof protocol minimalist MACs is shown in Figure 3.

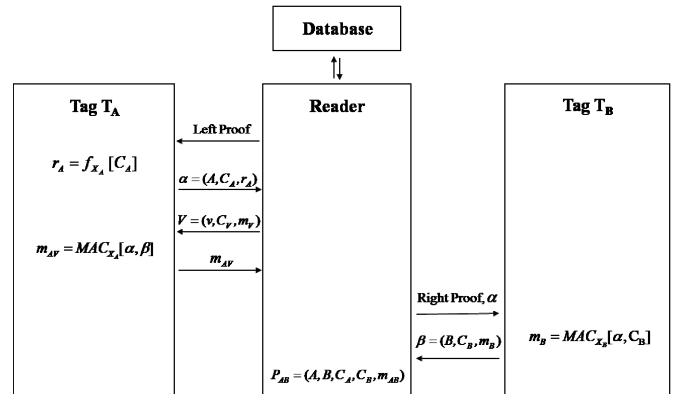


Figure 3 One-time yoking-proof protocol minimalist MACs Replay Attack

When the attacking reader with a vicious intent requests authentication to T_A through 'left proof', he will receive r_A value. Then, he will send r , the random number produced by himself to T_A as if it is r_B that he received from T_B . He will eventually receive m_A of T_A using r .

Being unable to know the authenticity of r_A makes it vulnerable to replay attacks.

r_A received from T_A is sent to T_B through 'right proof'. T_B produce normal m_B and receives m_B and normal r_B from the attacking reader. Replay attack will be successful where both T_A and T_B are recognized simultaneously.

III. THE PROPOSED PROTOCOL

In this study, a protocol is proposed where it allows multiple authentications while being well-protected against the replay attacks from Chapter 2. The proposed protocol is an improved version of the standard yoking-proof protocol proposed by Ari Juels.

It is assumed that in the proposed protocol, the key values used for MAC arithmetics are stored in the tag and that the keys are entered into database in the process of tag production and already being shared.

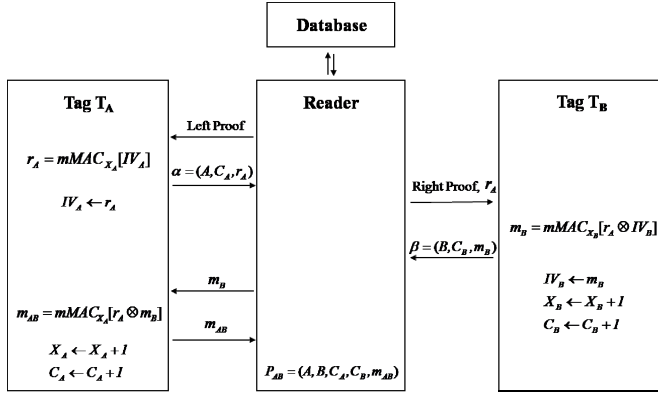


Figure 4 Proposed protocol

The proposed n-Time yoking-proof protocol is largely divided into 4 steps.

Step 1 : It is 'left proof' process where a partial verification takes place between the reader and Tag A.

Step 2 : It is 'right proof' process where the result of 'left proof' process is used and the reader executes another partial verification with Tag B.

Step 3 : It is a process where 'left' and 'right' are both verified.

Step 4 : Using the verification result of Step 3, the reader can execute an actual verification of the tag through communication with DB.

Looking into the details of each step, it can be divided into another 10 steps.

Step 1 : The reader requests 'left proof' to T_A

Step 2 : T_A produces a random value, r_A using IV_A as follows.

$$r_A = mMAC_{X_A}[IV_A]$$

Step 3 : T_A sends to the reader its ID, A along with C_A , the counter value of the current tag and r_A , the random value produced in Step 2.

$$\alpha = (A, C_A, r_A)$$

Then it renews IV_A value to r_A .

$$IV_A \leftarrow r_A$$

Step 4 : The reader, from the values it received from T_A , sends r_A to T_B and requests for 'right proof'.

Step 5 : Using the values of r_A and IV_B it received from Step 4, T_B produces m_B , which is its partial authentication.

$$m_B = mMAC_{X_B}[r_A \otimes IV_B]$$

Step 6 : T_B sends to the reader its ID B along with C_B , the counter value of the current tag and m_B , which was produced in Step 5.

$$\beta = (B, C_B, m_B)$$

Then it renews IV_B value to m_B and by increasing the counter value by 1, it renews C_B , the key value.

$$IV_B \leftarrow m_B$$

$$X_B \leftarrow X_B + I$$

$$C_B \leftarrow C_B + I$$

Step 7 : From the values it received from the tag in Step 6, the reader sends m_B to T_A .

Step 8 : T_A uses r_A sent in Step 3 and m_B to calculate m_{AB} .

$$m_{AB} = mMAC_{X_A}[r_A \otimes m_B]$$

Step 9 : T_A sends m_{AB} to the reader and by increasing the counter value by 1, it renews X_A , the key value.

$$X_A \leftarrow X_A + I$$

$$C_A \leftarrow C_A + I$$

Step 10 : The reader sends to the DB, T_A including m_{AB} , and A and B , the identification values of T_A and T_B , respectively, and C_A and C_B the counter values of T_A and T_B to get verification.

$$P_{AB} = (A, B, C_A, C_B, m_{AB})$$

The proposed protocol uses different key values for every authentication, which makes it able to allow multiple authentication while being well-protected against replay attacks.

IV. SECURITY ANALYSIS

The proposed protocol uses different key values for every authentication session using counter. It is well-protected from replay attacks and by preventing the re-use of key values, which is one of the weaknesses of the MAC function to prevent key exposure.

The proposed protocol is safe from replay attacks in the following attacking process.

Attack 1 : The attacker obtains $\alpha = (A, C_A, r_A)$ from T_A through 'left proof'.

Attack 2 : The attacker sends its randomly produced m_B value to T_A

Attack 3 : The attacker obtains m_{AB} value produced by T_A through m_B

Then, using the information it has obtained in Attack 1~3, the attacker exchanges authentication information with T_B without T_A .

Attack 4 : The attacker sends r_A that it obtained from T_A to T_B .

Attack 5 : The attacker receives normal $\beta = (B, C_B, m_B)$ from T_B .

Attack 6 : The attacker produces the authentication value $P_{AB} = (A, B, C_A, C_B, m_{AB})$ and requests for verification.

In the verification process, because the value sent from the attacker was calculated with wrong key values, the verification will fail. Also, because the key values are renewed everytime, it is safe even when the key value is exposed.

V. CONCLUSION

Ari Juels has proposed 2 versions of yoking-proof for authenticating multiple tags simultaneously, and both of them have strengths and weaknesses. The standard version provides maximum security but it is cost ineffective because it deploys MAC function with keys in the tag. On the other hand, the one-time version provides a simple and powerful MAC function using minimalist MAC but when it is used more than twice, it is very vulnerable in terms of security. Also, the one-time version is very vulnerable to replay attacks. Therefore, in this study, a new version called n-time authentication protocol has been proposed where it made the best of both the standard and one-time versions.

ACKNOWLEDGMENT

This work was supported by the SoongSil University Research Fund.

REFERENCES

- [1] A. Juels, "Yoking Proofs for RFID Tags", IEEE Computer Society, Proceedings of the First International Workshop on Pervasive Computing and Communication Security - PerSec 2004, pp.138-143, 2004
- [2] Ari Juels, "Generalized yoking-proofs for a group of RFID tags", MOBIQUITOUS 2006, 2006.
- [3] H.Krawczyk, M.Bellare, and R.Canetti.HMAC: Keyedhashing for message authentication, Internet Engineering Task Force RFC 2104, February 1997.
- [4] J. Saito and K. Sakurai. "Grouping Proof for RFID Tags," Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA' 05), pp. 621-624, 2005.
- [5] L. Bolotnyy and G. Robins, Generalized "Yoking-Proofs" for a Group of Radio Frequency Identification Tags, Technical Report CS-2006-12, Department of Computer Science, University of Virginia, 2006.
- [6] Selwyn Piramuthu. On existence proofs for multiple RFID tags. In IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing - SecPerU 2006, Lyon, France, June 2006. IEEE, IEEE Computer Society Press.

[7] Jung-Sik Cho, Suchul Hwang, Sang-Soo Yeo and Sang-Yong Rhee, "Enhanced Yoking Proof Protocols for RFID Tags and Tag Groups", 22nd International Conference on Advanced Information Networking and Applications - Workshops, pp.1591 – 1596, 2008