An Improved Proof for RFID Tags

Thiti Nuamcherm[1], Piya Kovintavewat[2], Charturong Tantibundhit[3], Urachada Ketprom[4], Chaichana Mitrpant[5]

[1,3] *Electrical and Computer Engineering Department, Thammasat University, Thailand*

[2] *RFID Technology and Application Research Unit, Nakhon Pathom Rajabhat University, Nakhon Pathom 73000, Thailand.*

[4,5] *RFID Program, National Electronics and Computer Technology Center (NECTEC), Thailand*

Email : [1]tnuamcherm@hotmail.com, [2]piya@npru.ac.th, [3]charturong@engr.tu.ac.th,
[4] urachada.ketprom@nectec.or.th , [5]chaichana.mitrpant@nectec.or.th

Abstract— **An RFID system consists of an RFID reader, a server connected to a database, and RFID tags attached to the objects needed to be identified. We are interested in the case where two tags are needed to be simultaneously scanned in a reader's field, for which many protocols have been proposed in literature. They however are all insecure for the special case, where an adversary, acting as a reader, can penetrate the server and access data stored in the server. In this paper, we propose a protocol to remedy this problem for two-tag environment. In addition, it can be extended to more-than-two-tag environment without much increasing total amount of time to scan all the tags.**

I. INTRODUCTION

Radio Frequency Identification (RFID) is a technology for automated identification. An RFID system typically consists of servers, RFID readers, and RFID tags, where the readers and the tags communicate with one another via radio frequency waves. Its advantages over a barcode system [1], include multiple RFID tags reading without line-of-sight, and high data storage capacity in an RFID tag. Due to its capability, RFID has been widely used in several applications, such as transport systems, electronic ticketing, access control, animal identification, logistics and supply chain management [1]. Although RFID has many benefits, but in practice it has some privacy and security issues needed to be addressed [2]. A lot of research works has been conducted to solve such problems.

Normally, before data transmission between a reader and a tag begins, the reader needs to authenticate the tag to ensure that the tag is known to the reader. In this paper, we are interested in the case where two tags are needed to be simultaneously scanned in a reader's field, and verified by the server. This case is very crucial in pharmaceutica distribution, especially in the case where some medications are required to be dispensed with leaflet describing its side-effects [3]. Many protocols to scan two tags simultaneously have been proposed in the literature. The "Yoking proof" was proposed by Juels [3] for scanning two tags simultaneously. Nonetheless, "Yoking proof" is not immune against a "replay attack" as described in [4]. Therefore, Saito and Sakurai proposed a "Yoking proof using time stamp" or a "Grouping proof" [4] to prevent a "replay attack." However, "Grouping proof" is also not immune to a "replay attack" if the time stamp used is somehow known to an adversary. As a result, Piramuthu proposed a "Modified proof" [5] to solve this problem. This modified proof is still vulnerable to the attack where an adversary acting as a reader can penetrate the server and gain access the information needed in the process of simultaneous tags scanning. We propose a protocol that addresses this vulnerability and can be extended to the case where there are more than two tags to be scanned simultaneously in a parallel manner.

The rest of this paper is organized as follows. Section II briefly describes the existing proofs for scanning two tags simultaneously. We propose a new protocol called "Proposed proof" in Section III. Section IV discusses some security issues related to the "Proposed proof." Finally, Section IV concludes this paper.

II. RELATED WORKS

In this section, we briefly describe three protocols used to scan two tags simultaneously in a reader's field, namely, "Yoking proof," "Grouping proof," and "Modified proof." Before explaining these protocols, we define some notations that will be used throughout this paper.

Notation:

- T_A, T_B : RFID tags (i.e., Tag A and Tag B)
- r, r_A, r_B, r_T : random numbers
- TS : a time stamp
- x_R, x_A, x_B : secret keys of the reader, T_A, and T_B
- MAC: a message authentication code
- $MAC_x(m)$: MAC applying a secret key *x* on a message *m*
- P_{AB} : a proof for verification in a server/verifier that T_A and T_B are simultaneously present

A. *Yoking Proof*

Juels proposed a "Yoking proof" protocol [3] for an RFID reader to scan two RFID tags simultaneously, and the result of the scan is sent to a server for verification, as shown in Fig. 1. This protocol works as follows: First, T_A and T_B are initialized with secret keys x_A and x_B, respectively, known to the server. The protocol starts with the reader

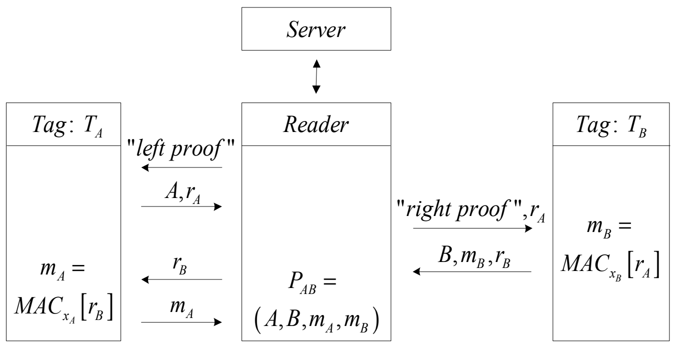


Fig. 1. A "Yoking proof" protocol [3].

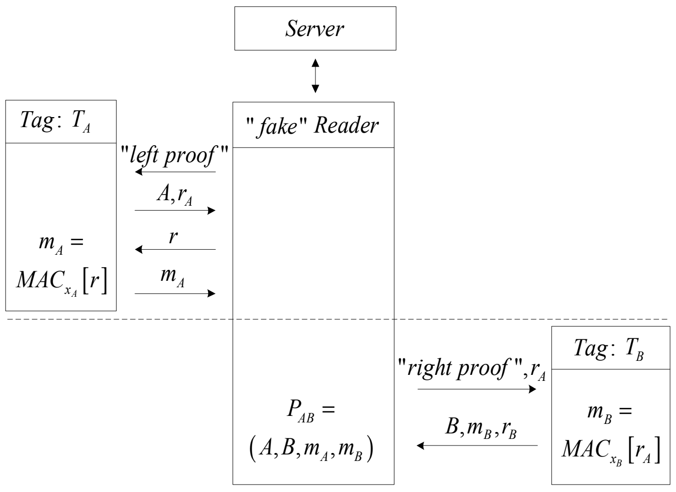


Fig. 2. A replay attack against "Yoking proof" [4].

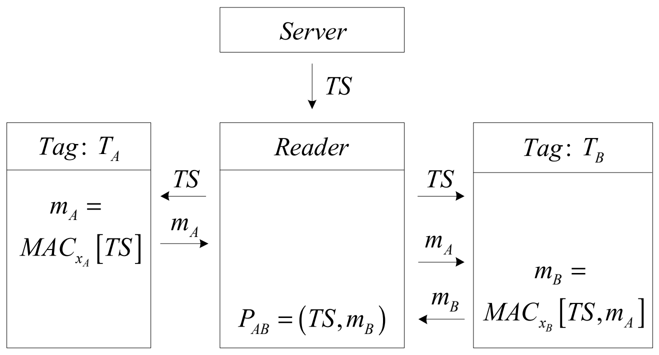


Fig. 3. A "Grouping proof" protocol [4].

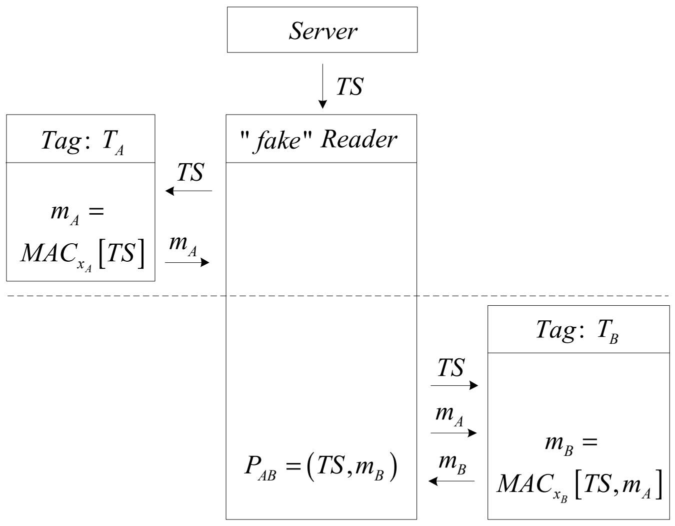


Fig. 4. A replay attack against "Grouping proof" [5].

sending a "left proof" to one of the two tags, says T_A. Then, T_A generates a random number r_A and transmits it to the reader. Upon receiving r_A, the reader sends a "right proof" and r_A to the other tag, i.e., T_B. Thus, T_B generates a random number r_B and the MAC m_B based on x_B and r_A, and sends r_B and m_B back to the reader. Again, the reader sends r_B to T_A in order to generate the MAC m_A based on x_A and r_B, and then sends m_A to the reader. Finally, the reader compute P_{AB} = (A, B, m_A, m_B) and sends it to the server to verify the simultaneous presence of T_A and T_B.

B. Grouping Proof

Saito and Sakurai [4] have shown that the Yoking proof can be attacked by a "replay attack," where two tags can be correctly verified although they are not present at the same time. Figure 2 illustrates how the "replay attack" works, where the dashed line indicates the differences in time and place. Because m_A and m_B are solely generated from r and r_A, respectively, and m_B is independent of T_A, the adversary can pretend to be a legitimate reader that has ability to get information from T_A and T_B, i.e., r_A, r, m_A, and m_B. Then, the "fake" reader can compute P_{AB} and send it to the server for verification since the adversary has complete control over the content of what is submitted to the verifier [5].

Therefore, Saito and Sakurai proposed a "Grouping proof" or a "Yoking proof using time stamp (*TS*)" [4], as depicted in Fig. 3. The idea is that because *TS* changes every time, it will thus be used to generate a MAC instead of using a random number. The protocol starts with the server sending *TS* to T_A and T_B via a reader. Then, T_A generates and sends the MAC m_A, applying x_A on *TS*, to the reader. Next, the reader sends m_A to T_B, and T_B reacts by generating and sending the MAC m_B, applying x_B on *TS* and m_A, to the reader. Eventually, the reader computes P_{AB} and sends it to the server for verification.

C. Modified Proof

Piramuthu has shown that "Grouping proof" can also be attacked by a "replay attack" for a specific case [5], as illustrated in Fig. 4. That is, if the adversary somehow knows *TS*, he/she can then perform a replay attack when both tags are separated. This is because m_A is independent of T_B, although

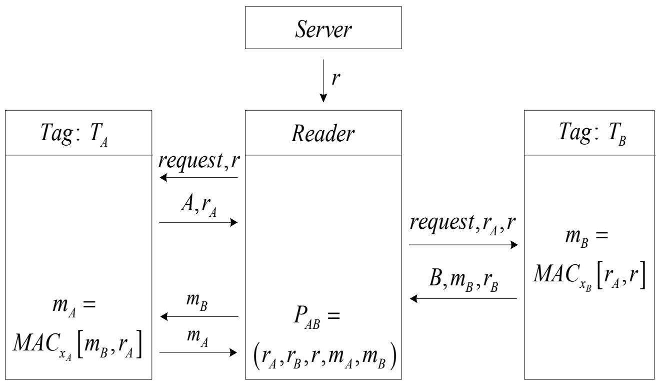
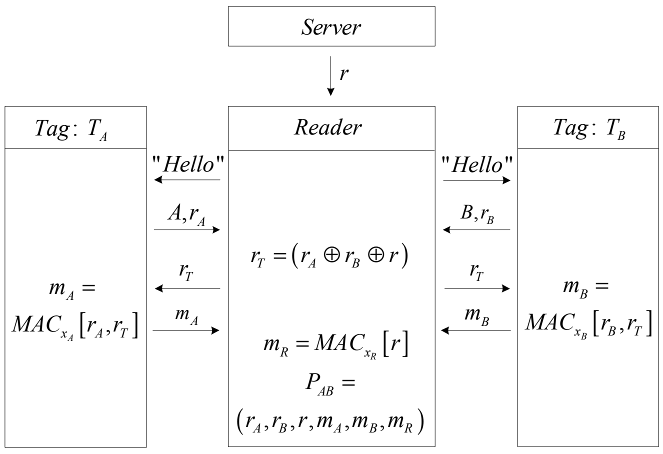


Fig. 5. A "Modified proof" protocol [5].



Server
r
Tag: T_A
Reader
Tag: T_B
"Hello"
A, r_A
r_T
m_A
"Hello"
B, r_B
r_T
m_B
$r_T = (r_A \oplus r_B \oplus r)$
$m_R = MAC_{x_R}[r]$
$P_{AB} = (r_A, r_B, r, m_A, m_B, m_R)$
$m_A = MAC_{x_A}[r_A, r_T]$
$m_B = MAC_{x_B}[r_B, r_T]$

Fig. 6. A "Proposed proof" protocol.

m_B depends on m_A (see how to compute m_A and m_B in Figs. 3 and 4).

Accordingly, Piramuthu proposed a "Modified proof" [5], as shown in Fig. 5. The protocol begins when the reader receives a random number r from a server. Then, the reader sends a request and r to T_A, and T_A reacts by generating and sending r_A to the reader. The reader queries T_B by sending a request, r_A, and r. T_B reacts by generating and sending r_B and the MAC m_B, applying x_B on r_A and r. The reader again sends m_B to T_A, and T_A generates and sends the MAC m_A, applying x_A on m_B and r_A, back to the reader. Finally, the reader computes $P_{AB} = (r_A, r_B, r, m_A, m_B)$ and sends it to the server for verification. It is now clear that this protocol requires the presence of the two tags for verification, because the tags are related to each other.

III. Proposed proof

Our goal is to propose a protocol that can prevent the adversary generating a proof, denoted as "fake proof", and a replay attack under the assumption that a reader has limited time to create a valid proof. Therefore, if the time used to generate a proof exceeds an allowable time, the server will reject that proof, and the reader has to restart the proof generating process.

We propose two mechanisms: one to authenticate a reader to the server and the other one to prevent a replay attack. For the server to authenticate a reader, both should have a shared secret key that can be used for MAC generation. To prevent a replay attack, a reader creates a proof based on a random number, collaboratively generated by all the tags and the server, and secret keys shared between each tag and the server.

Figure 6 illustrates the proposed protocol, which can be explained as follows:

1) The reader queries the server, and the server reacts by sending a random number r to the reader.
2) The reader queries T_A and T_B by sending "Hello." T_A and T_B react by generating and transmitting r_A and r_B to the reader, respectively. Then, the reader generates r_T using r_A, r_B, and r, based on an XOR operation, i.e., $r_T = r_A \oplus r_B \oplus r$.
3) The reader sends r_T to T_A, and T_A reacts by generating and sending the MAC m_A, applying x_A on r_A and r_T, to the reader.
4) The reader sends r_T to T_B, and T_B reacts by generating and transmitting the MAC m_B, applying x_B on r_B and r_T, to the reader.
5) The reader generates the MAC m_R, applying x_R on r, and computes P_{AB} using r_A, r_B, r, m_A, m_B, and m_R.
6) The reader sends P_{AB} to the server for verification.

Additionally, the proposed proof can also be extended to scan more than two tags simultaneously. That is, the reader gets r from the server and sends it to all the tags, the tags then react by generating and transmitting r_i (i is a tag number from 1 to n) to reader. The reader computes r_T using r_1, r_2, …, r_n, and r, based on XOR. Each tag then generates m_i, applying a secret key x_i on r_i and r_T. Thus, the reader computes $P_{1,2,\ldots,n}(r_1, r_2, \ldots, r_n, r, m_1, m_2, \ldots, m_n, m_R)$ and sends it to the server for verification.

IV. Security analysis

The proposed protocol can scan two tags simultaneously and requires the combined presence of the two tags in a reader's field. Therefore, it can prevent a fake proof and a replay attack. Based on security analysis from Dimitriou [6], below are some issues of our proposed proof.

1) **Attack on a tag**: In this case, the adversary pretends to be a trusted reader and generates a proof to get verified by a server. However, the server will not accept this proof because the adversary does not know the secret key x_R used to generate m_R. Therefore, the adversary cannot get

verified by the server. Likewise, a replay attack does not work in this case because m_A and m_B all depend on r_A, r_B, and r.

2) **Attack on the reader**: The adversary pretends to be an RFID tag. An attack on the reader does not succeed because the server knows and the secret keys of the two tags (x_A and x_B).

3) **Attack on the communication between tag and reader**: The adversary can eavesdrop a communication between the reader and tags to get information necessary to generate a proof. Nonetheless, the server rejects this proof because the value of *r* changes all times and cannot be reused. In addition, the adversary cannot generate m_R because he/she does not know the secret key (x_A and x_B).

4) **Attack on user privacy**: In this paper, we consider the public communication, where the adversary can eavesdrop a communication between the reader and tags. Thus, we will not mention about this security issue.

5) **Attack on location privacy**: Because the data used in communication changes every times, the adversary cannot reuse that data to generate a proof. Then, the proposed protocol immunes to this attack.

6) **Attack against the key**: The adversary wants to know the key used in data communication. This problem can be ignored if the key is chosen carefully [7]. Thus, we disregard this attack.

7) **Attack against implementation**: Similarly, if the keys and the random numbers used in the protocol are selected carefully, this problem is negligible. Then, we will not mention about this attack.

8) **Disassembling the tags**: When the two tags disassemble, the adversary cannot generate a proof for verification because the two tags depend on each other. Thus, the proposed protocol is immune to this attack.

V. Conclusion

We briefly described the authentication protocols to scan two tags simultaneously. Several protocols have been proposed in the literature. However, for a specific case, where the adversary, acting as a reader, can penetrate the server and gain access to data stored on the server, the existing protocols are not immune to a replay attack and a fake proof. Here, we proposed a protocol that addresses this problem. The idea is that the proposed proof must have a secret key known to the server. The reader generates the MAC to identify itself to the server, using that secret key. Based on security analysis, we found that our proposed proof is immune to a fake proof and a replay attack. Although, the security becomes strong in our situation, the reader requires high computation to compute the MAC.

Acknowledgment

This work was supported by National Science and Technology Development Agency (NSTDA) and the RFID Program, National Electronics and Computer Technology Center (NECTEC), Thailand, under grant TG-44-21-50-097M. The authors also would like to thank Assist. Prof. Dr. Chanathip Namprempre for insightful advice.

References

[1] K. Finkenzeller, *RFID Handbook*, 2nd ed., Wiley & Sons, 2002.

[2] A. Juels. "RFID Security and Privacy: A Research Survey," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 381-394, 2006.

[3] A. Juels. "Yoking Proofs" for RFID Tags," in *Proc. of the First International Workshop on Pervasive Computing and Communication Security*. IEEE Press. 2004.

[4] J. Saito and K. Sakurai. "Grouping Proof for RFID Tags," in *Proc. of the 19th International Conference on Advanced Information Networking and Applications* (AINA'05), pp. 621-624, 2005.

[5] S. Piramuthu. "On Existence Proofs for Multiple RFID Tags," *IEEE International Conference on Pervasive Services* (ICPS'06), pp. 26-29, June 2006, Lyon, France.

[6] T. Dimitriou. "A Lightweight RFID Protocol to Protect Against Traceability and Cloning Attacks," in *Proc. of the IEEE International Conference on Security and Privacy for Emerging Areas in Communication Networks – SECURECOMM*, 2005.

[7] A. Lenstra and E. Verheul. "Selecting Cryptographic Key Sizes," *Journal of Cryptography*, vol. 14, no. 4, pp. 255 – 293, 2001.