On Existence Proofs for Multiple RFID Tags

Selwyn Piramuthu
Decision and Information Sciences
University of Florida
Gainesville, Florida 32611–7169
Email: selwyn@ufl.edu

***Abstract*— The ability to securely authenticate RFID (Radio-Frequency Identification) tags is of paramount importance in RFID-tagged applications where the integrity of the system as well its security can be compromised by an adversary. RFID tags have resource constraints including limited memory and processing capacity which restrict utilization of cryptographic primitives that have been used in other applications. We consider the case where two or more RFID tags need to be simultaneously scanned. Juels (2004) provided a proof for the simultaneous existence of two RFID tags. Saito and Sakurai (2005) later showed that the proof described in Juels(2004) is vulnerable to 'replay attack,' and proposed new proofs to alleviate this problem. However, the proofs provided by Saiko and Sakurai (2005) are not immune to 'replay attack.' We propose a modified proof and provide its security analysis.**

I. Introduction

RFID (Radio-Frequency Identification) tags are slowly becoming popular as replacements for barcodes in several application areas. Although quite expensive compared to systems using barcodes, the costs associated with attaching RFID tags and implementing relevant systems such as tag readers, back-end database systems, etc., are coming down as the number of implementations increases over time (Finkenzeller, 2002).

RFID tags have several advantages over barcodes including the ability to store more data, contactless readability, being able to be read in batches, no need for line-of-sight to be read, among others. These advantages, of course, come with costs including the obvious monetary ones as well as those in terms of privacy and security. Several studies have approached this issue of privacy and security in RFID tag enabled systems (e.g., Avoine and Oechslin, 2005, Weis et al., 2004) with varying levels of success.

In this study, we are interested in a specific scenario involving RFID tag applications, namely the case where simultaneous presence of two tags in a reader's field is to be proved. This has been studied by Juels (2004) and Saito and Sakurai (2005). Example scenarios where this is relevant are the need for certain medications to be dispensed together with an appropriate leaflet, the need for two parts to leave a factory together, and generally any situation that dictates the combined presence of two entities (Juels, 2004).

This paper is organized as follows: we present an overview of related work and provide a brief evaluation of these in the next section. We also observe conditions under which these proofs could be compromised. In Section III, we describe the proposed proof addressing some of the concerns discussed in Section II. This is followed by brief security analysis of the proposed proof in Section IV. Section V concludes the paper.

IEEE International Conference on Pervasive Services (ICPS'06), 26-29 June 2006, Lyon, France.

II. Related Work

Juels (2004) and Saito and Sakurai (2005) present proofs for simultaneous presence of two RFID tags in reader's field. In this section, we consider each of these in turn. Specifically, in the next subsection, we describe the "Yoking Proof" (Juels, 2004) and its critique by Saito and Sakurai(2005). We then provide a brief discussion on "grouping proof" (Saito and Sakurai, 2005) in the following subsection. In a subsequent subsection, we show yet another complementary scenario where "yoking proof" is not immune to 'replay attack.' We show that the "grouping proof" too is not immune from 'replay attack.'

Notations Used:

- V: verifier for MAC
- r, r_A, r_B: random numbers
- x_A, x_B: secret keys of RFID Tags T_A & T_B
- MAC: Message Authentication Code
- MAC_x[m]: MAC using secret key x on message m
- TS: time stamp
- P_{AB}: proof A&B scanned simultaneously

A. Yoking Proof

Juels(2004) presented the "yoking proof" (Figure 1) for two tags T_A and T_B to be simultaneously scanned. Here, the reader interacts with the two RFID tags (T_A and T_B) and a back-end trusted server/verifier (V). The tags themselves cannot interact with each other, but only with the reader.

The scenario begins when the reader sends a "left proof" to one of the tags indicating its role in the protocol. The 'left' tag reacts by generating and transmitting a random number (r_A). The reader forwards this along with "right proof" to the other ('right') tag. This tag generates the MAC (m_B) using x_B on r_A. The secret keys x_A and x_B are known to the server. The second tag then transmits m_B along with a randomly generated number (r_B). This random number is sent by the reader to T_A, which then uses x_A to generate MAC on r_B resulting in m_A. The tag T_A then sends m_A to the reader which assembles everything necessary for the proof (P_{AB}) and sends them to the back-end server for verification that the tags T_A and T_B were scanned together.

Saito and Sakurai(2005) show how a 'replay attack' can be played against "yoking proof" (Figure 2). Here, the authors separate the interactions between the reader and the tags (T_A and T_B) across time (represented by the horizontal lines across T_A and T_B in Figure 2) and show that the interactions between the reader and the tag T_A can be captured ahead of time and 'replayed' to tag T_B at a later point in time.

Although interactions between the reader and T_A can be captured earlier in time the interactions between the reader and tag T_A are not completely independent of the interactions between the reader and the tag T_B. For example, although an adversary can generate and transmit a random number (r) to T_A, the MAC generated by T_A is not independent of the random number r_B generated by the tag T_B. Hence, when P_{AB} is generated, there will be a mismatch between the random number r that is used by T_A for m_A and the random number (r_B) generated by T_B. This issue can be bypassed by submitting r, r_A, m_A, and m_B to the verifier (V) since the reader (here, the adversary) has complete control over the content of what is submitted to the verifier.

B. Grouping Proof

Saito and Sakurai (2005) present another proof (yoking proof with time stamp) which they call the "grouping proof" (Figure 3) where "yoking proof"'s 'replay attack' can be avoided. The grouping proof proceeds as follows: Initially, the reader sends TS, the time stamp, to both the tags. Tag T_A (the 'left' tag) generates m_A using its secret (x_A) on TS and transmits m_A to the reader. The reader transmits this to the 'right' tag (T_B), which uses its secret (x_B) on TS and m_A to generate m_B, which is then transmitted to the reader. The reader then assembles TS and m_B to generate the proof (P_{AB}).

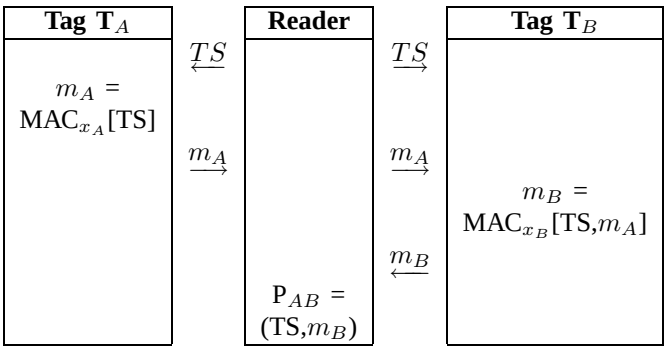

Fig. 3. "Yoking Proof" using time stamp (Saito and Sakurai, 2005)

According to Saito and Sakurai (2005), the reason TS was generated by V and used in the proof is to verify the time at which a given MAC was generated. Since these times are known, 'replay attacks' reusing these MACs can be prevented.

C. Discussion on "yoking" and "grouping" proofs

In addition to the 'replay attack,' mentioned in Saito and Sakurai (2005), that can be played out on "yoking proof," there is yet another (complementary) scenario that falls prey to a similar 'replay attack' but on the other tag. For example, since r_A is not used by tag T_A after it is transmitted to the reader, any random number (r) can be used from this time on and the end result would not be different. I.e., we can capture the transmissions between the reader and tag T_B and replay it in its absence to the other tag (T_A) as shown in Figure 4.

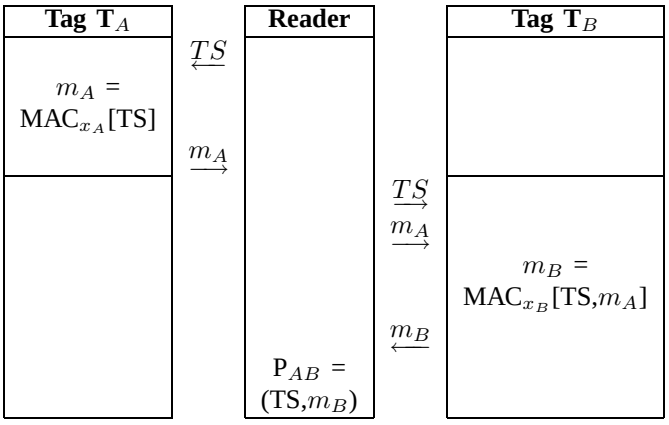

Fig. 5. Replay attack against "Yoking Proof" using time stamp

Similar to "yoking proof," the "grouping proof" too is vulnerable to 'replay attacks' as given in Figure 5. Here, an adversary begins by repeatedly transmitting messages to the 'left' tag (T_A) using several different time stamps from some later points in time. Various disparate combinations of (TS, m_A) can be gathered in this manner. Then, at some later point in time when TS becomes true, the 'replay attacks' can be instantiated without the presence of T_A. It is worth noting that, unlike "yoking proof," the complementary scenario is not vulnerable to 'replay attack.' This is because m_B is dependent on m_A and therefore cannot be generated before generation of m_A by the 'left' tag (T_A). We use this principle in the modified proof presented in the next section. The fact that P_{AB} uses only m_B (and not m_A) could also lead to other vulnerabilities.

III. Modified Proof

The proposed modified proof given in Figure 6 is a variation of "yoking proof." It uses a principle partially used in the "grouping proof," as mentioned at the end of the previous section. The idea is to ensure that the inputs to a tag are based on parameters that are necessary for the other tag, and to create dependence of the tags on each other so that they cannot be processed separately in the proof without the presence of the other tag.

We assume that the reader authenticates itself with the back-end verifier before beginning the process of obtaining r from V as well as when returning P_{AB} at the end of the process. Although this assumption by itself should provide a reasonable amount of support against attacks on the system, we disregard any such influence. While generating a proof, when a transmission of interest fails to reach its intended receiver,

as evidenced by a lack of response within a pre-specified time limit, the transaction is cancelled and started all over again with a fresh r from V. Beyond this, the main differences of the proposed proof (vs. "yoking proof") are as follows:

- The addition of a random variable (r) sent to both the tags from the verifier through the reader. This helps us keep track of the time duration between the initial transmission from the reader to the 'left' tag and final submission of P_{AB} for verification by the verifier. The random variable r is also used as seed for generating r_A and r_B by the tags.
- The MAC generated by T_B depends on both r and r_A. The use of r_A in generating m_B is crucial. Since r_A is generated and used internally in T_A for generating m_A as well, an adversary cannot run 'replay attack' on either of the tags. Because r is generated by the verifier, the dependence on r for generating m_B adds yet another layer of protection against attacks.
- The fifth transmission in the proof is m_B instead of r_B (as in "yoking proof"). This helps in the generation of m_A.
- The use of m_B in generating m_A is crucial since T_A has to wait for T_B to generate m_B. Therefore, T_A's part of the proof cannot occur before T_B's part and T_B's part cannot happen independently since it too is dependent on input from T_A (r_A). T_A also generates r_A, which is kept internal (i.e., it is not received as input from an outside entity). Hence it cannot be corrupted by an outside entity.

IV. Security Analysis

Following Dimitriou (2005), we present a brief security analysis on the proposed proof for simultaneous scan of two RFID tags.

Attack on a tag. This type of attack refers to the scenario where an adversary pretends to be the reader. Since the proof is based on avoiding exactly this type of attack, the adversary will not be able to succeed in completing the proof even if parts of the steps are violated.

Attack on the reader. Here, the adversary pretends to be a valid tag. This type of attack will not succeed because of the shared secret keys (x_A and x_B). The adversary will not succeed with 'replay attack' either because each of the tags need fresh input from the other which changes every time the proof is run.

Attack on the communication between tag and reader. An adversary can block messages between the reader and tag(s). When this happens, the proof is broken and it doesn't succeed. The entire transaction is re-started with a fresh r from V. If the adversary continues to block messages between reader and tag(s), the proof will not succeed even though both the tags are simultaneously present in the field of the reader.

Attack on user privacy. Since no 'private' information is transmitted during the proof, this is of no concern here.

Attack on location privacy. Since data used in transmissions (r, a, r_A, r_B, B, m_A, and m_B) are refreshed every time the proof is run and none of these are stored for future runs of the proof, location privacy is guaranteed.

Attack against the key. This happens when an attacker listens in on the transaction and tries to identify the key values. Again, if the keys are selected appropriately (e.g., Lenstra and Verheul, 2001), this is not of concern.

Attack against implementation. Provided the keys and the random numbers are genereted with caution, this is not of concern.

Disassembling the tags. These tags are clearly not tamper-resistant, and can be disassembled to retrieve MAC as well as xs. Even if this happens, due to the forward privacy of the proof, past transactions are secure.

V. Conclusion

We evaluated the two proofs that have been proposed thus far in the literature for ascertaining the simultaneous presence of two RFID tags in the field of the reader. We showed that both these proofs have minor areas of concern, and proposed a means to address these concerns using minimal processing for tags that cannot execute standard cryptographic primitives. We also provided brief security analysis of the proposed proof.

Although we discussed only the case where we are interested in proving the simultaneous presence of two tags in the field of the reader, extending this to a case where several tags are simultaneously present is relatively straight-forward. One of the ways of extending this would be to collapse the messages sent to tag T_B into the reader and let the reader generate m_{B_i} (i = 1..n, where n is the number of tags of interest) values for each of the tags. In the end, P_{AB} can be evaluated based on $r_{A_1}, r_{A_2}, ...r_{A_n}, r, m_{A_1}, m_{A_2}, ...m_{A_n}$.

References

[1] G. Avoine and P. Oechslin. "RFID Traceability: A Multilayer Problem," *Financial Cryptography - FC'05*, LNCS, Springer, 2005.

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

[3] K. Finkenzeller. *RFID Handbook*, second edition, Wiley & Sons, 2002.

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

[5] A. Lenstra and E. Verheul. "Selecting Cryptographic Key Sizes," *Journal of Cryptography*, volume 14, number 4, pp. 255-293, 2001.

[6] 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.

[7] S. Weis, S. Sarma, R. Rivest, and D. Engels. "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," *Proceedings of the 1st Security in Pervasive Computing*, LNCS, volume 2802, pp. 201-212, 2004.

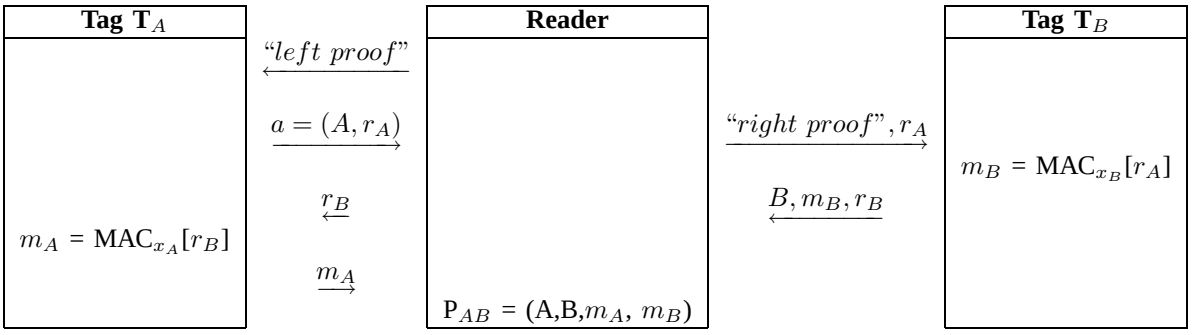

Fig. 1. "Yoking Proof" for RFID Tags (Juels, 2004)

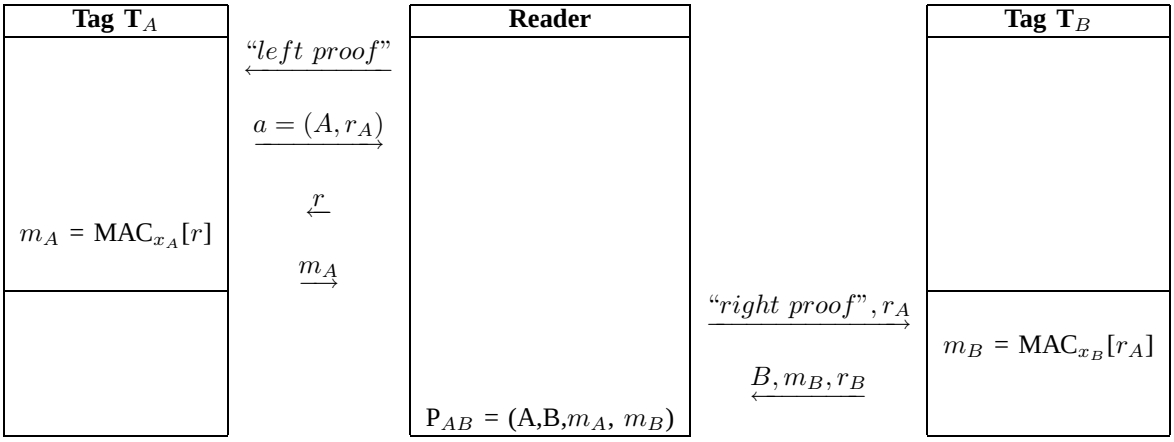

Fig. 2. Replay attack against "Yoking Proof" (Saito and Sakurai, 2005)

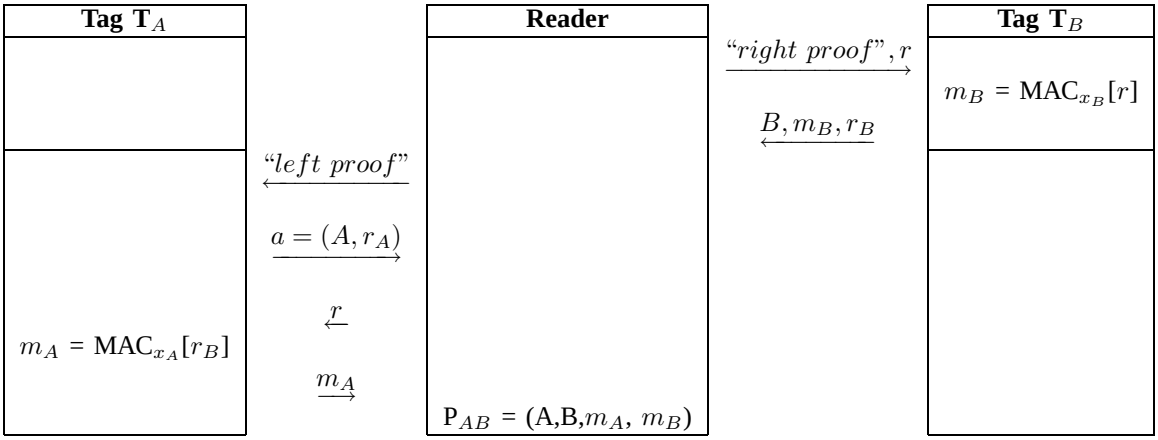

Fig. 4. Replay attack against "Yoking Proof"

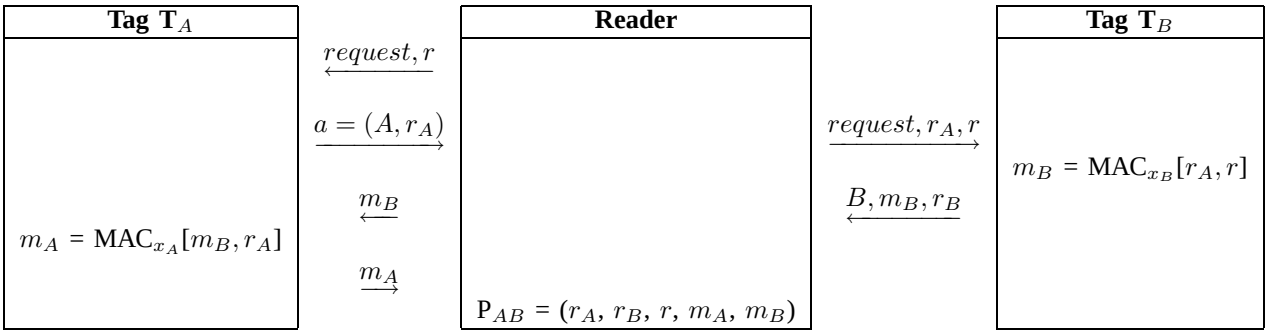

Fig. 6. Modified proof for RFID tags