International Journal of Advances in Computer Networks and Its Security– IJCNS
Volume 4: Issue 2 [ISSN: 2250-3757]
Publication Date : 25 June 2014

# A RFID yoking proof protocol to preserve an offline verification using the commitment disclosure

[ Hyoungmin Ham, Jooseok Song ]

*Abstract*— **A RFID (Radio Frequency Identification) yoking-proof provides a verification manner that a pair of RFID tags is scanned simultaneously by one reader device to guarantee the physical proximity of multiple objects. However, the previous studies provide the verification that requires the online verifier. The connectivity between a reader and a verifier limits the practicality of a yoking proof. In this paper, we propose an offline yoking proof protocol to preserve the offline verification that does not require the online verifier. In addition, the protocol for a pair of tags is easily extended to the one for multiple tags without additional expensive devices. Our analysis shows that the proposed protocol provides offline verification securely and effectively.**

*Keywords*— **Anonymity, Privacy, RFID, Verification, Yoking proof**

## I. Introduction

A RFID (Radio Frequency Identification) tag is a small microchip that has a unique identifier. Each tag delivers its identifier to RFID readers through a wireless channel. RFID system reduces costs in logistics management thus a number of researches and commercial products focus on it. Among the studies on RFID, in 2004, A. Juels proposed a novel idea called yoking proof to prove the co-existence of a pair of RFID tags [1]. In the yoking proof, a single RFID reader scans a pair of tags simultaneously and generates a proof to verify two tags read by the same reader at once. The yoking proof has a number of promising applications. For example, medicine factories need to check whether a product is packaged with a safety cap.

However, all the previous yoking proof schemes require the trustable verifier that is connected with a reader device. Furthermore, the previous schemes in [1]–[4] do not provide appropriate countermeasures against the attacks that obstruct the process for generating correct proofs although the purpose of them is to be resilient against the attacks.

In this paper, we propose a new yoking proof protocol to preserve offline verification. Our contributions can be summarized as follows:

- We propose a noble approach of a yoking proof. The offline verifiable yoking proof protocol (OV-yoking proof) provides offline verification via the spatial data of tags and the commitment disclosure. The verification of our protocol requires a reader, but a trustworthy online verifier.
- Security and privacy preserving protocol: Our protocol is secure against attacks to forge a yoking proof (Replay attack, Reassembled proof), and satisfies requirements to preserve the privacy protection (Confidentiality and Unlinkability, as defined in Section 4).
- Lightweight protocol: Our offline verifiable yoking proof protocol requires tags to have only a cryptographic hash function. Neither encryption function, used in [2], [7], nor a clock, used in [1], [2], [4], [8], is needed.

## II. Background

In this section, we introduce the basic yoking proof protocol of A. Juels, and its variants.

### A. *Yoking proof protocols preserving the online verification*

A. Juels proposed the basic yoking proof protocols [1]. The yoking proof protocols involve two responses from a pair of tags in generating the proof to guarantee the multiple scans in a single session. Hence, the RFID reader uses the response from one tag as a challenge to the other tag. However, if a time interval between the responses is too long, we cannot assure whether the responses are simultaneously generated or not. Thus, every session should be finished within an appropriate time period to generate the trustable proofs.

The protocols in [1] use timeout to limit response time for a session. That is, tags terminate a session when a predefined time period expired. The yoking proof protocols assume that each tag is initialized with a unique secret key, and a trusted verifier knows their secret keys that are stored in its database.

Table I explains the notations used in the procedure for this protocol and other yoking proof protocols. To simplify, we suppose a procedure to generate a yoking proof for two tags: *TagA* and *TagB*. Fig. 1 depicts the procedure, and the detailed procedure is as follows:

Hyoungmin Ham, Jooseok Song

Dept. of computer science, Yonsei Univ.
Korea

TABLE I. COMPUTATION COSTS OF TAGS

| Symbol | Description |
|---|---|
| *V* | Verifier |
| *R* | RFID reader |
| $Tag_A$ | RFID tag which is an initiator tag in its subgroup. |
| $Tag_B$ | RFID tag which is an another participant in its subgroup |
| *IDA or A* | ID of *TagA* |
| $ID_R$ | ID of a RFID reader |
| $r_A$ | Random number generated for *TagA* |
| *r* | Random number generated by a verifier |
| $C_A$ | Counter of *TagA* |
| $X_A$ | Symmetric secret key of *TagA* |
| $f_x[m]$ | One-way hash function using key *X* |
| $\text{MAC}_x[\text{m}]$ | MAC (Message Authentication Code) of *m* using key *X* |
| *cmt*, *cmtd* | Commitment of a yoking proof and its disclosure |
| $P_{AB}$ | Proof of *TagA* and *TagB* |
| Δ | Pre-defined time window |

- 1. *R* sends a left proof query to $Tag_A$.
- 2. $Tag_A$ generates a random number $r_A$ and sends it back to *R* with its ID, *A*.
- 2. *R* sends a right proof query to $Tag_B$ with $r_A$.
- 3. $Tag_B$ calculates $m_B$ = $\text{MAC}_{X_B}[r_A]$ using its secret key $X_B$ and $r_A$. $Tag_B$ then generates a random number $r_B$ and sends it back to *R* along with $m_B$ and its ID, *B*.
- 4. *R* sends $r_B$ to $Tag_A$.
- 5. $Tag_A$ calculates $m_A$ = $\text{MAC}_{X_A}[r_B]$, using its secret key $X_A$ and $r_B$, and sends it back to *R*.
- 6. *R* then sends a proof, $P_{AB} = (A, B, m_A, m_B)$ to *V*.
- 7. Since *V* has its database that has the secret parameters of the tags, it generates its own proof $P_{AB}'$ with received $m_A$ and $m_B$ and compares it with the received $P_{AB}$. If these two values are identical, *V* verifies that *TagA* and *TagB* exist simultaneously. On timeouts or incorrect inputs, any entity can terminate its participation in the protocol.

Schemes of [1–4] cannot satisfy the fundamental requirements to generate the valid proof. To overcome this limitation, schemes in [5]–[8] preserve the secure yoking proof generation, but they require additional modules to tags, such as cryptographic encryption [2], [7], PRNG [1], [3]–[8], MAC [1]–[7], timestamp [2] and timer [1], [2], [4], [8]. Furthermore, all of them [1]–[8] require multiple modules of a tag.

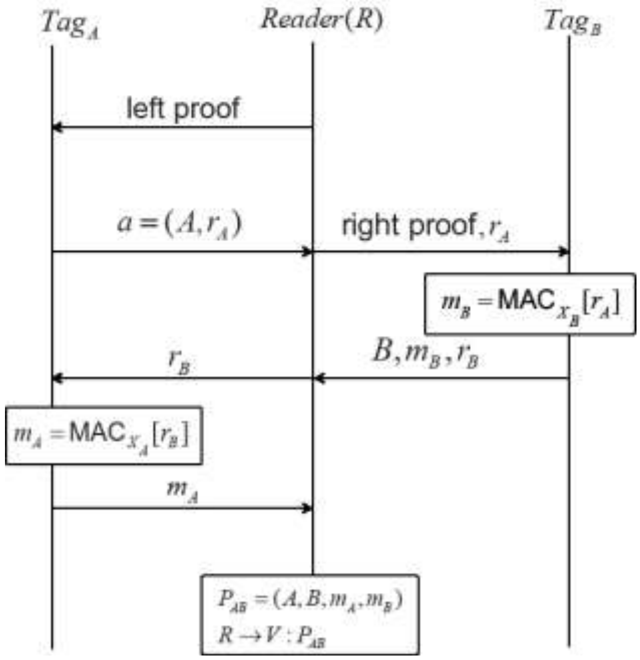
$Tag_A$
$Reader(R)$
$Tag_B$
left proof
$a = (A, r_A)$
right proof, $r_A$
$m_B = \text{MAC}_{X_B}[r_A]$
$r_B$
$B, m_B, r_B$
$m_A = \text{MAC}_{X_A}[r_B]$
$m_A$
$P_{AB} = (A, B, m_A, m_B)$
$R \rightarrow V : P_{AB}$

Figure 1. Yoking proof protocol

### B. *Security treats of a yoking proof*

The goals of an adversary are twofold: 1) to forge a valid proof, and 2) to trace a tag. In order to forge a valid proof, the adversary employs the following attacks:

- Replay attacks [2], [3]: The adversary impersonates a valid tag by replaying the messages sniffed previously.
- Reassembled proof: The adversary combines several incomplete proofs to generate a forged proof that seems like a valid proof, such as a Multi-proof (N) session attack [5].

## III. Proposed yoking proof scheme to preserve the offline verification

In this section, we propose an offline yoking proof protocol to preserve the novel way to verify a yoking proof without the online verifier. While the previous schemes [1]–[8] require multiple modules mentioned in Section 2 in a tag, the proposed scheme requires only one. We also extend the proposed protocol to group yoking proof protocol that generates a yoking proof for multiple tags efficiently.

### A. *Requirements for the secure yoking proof protocol*

The yoking proof for their RFIDs can certify the co-existence of the two objects. The requirements of yoking proof protocol are as follows:

- *R1*: Multiple tags which are scanned by a single reader in the same session should be able to generate a proof.
- *R2*: The proof should be verifiable by a trusted entity.

We define other requirements to provide security for the yoking proof protocol and anonymity of tags for the privacy. The requirements of secure yoking proof protocol are as follows:

- *SR1* (*Confidentiality*): No parameters that may identify a tag should be exposed in replies.
- *SR2* (*Unlinkability*): Responses from a tag should be dissociated from another.

## B. *Assumptions*

The followings are assumptions which are widely applied to yoking proof protocols.

- Tags have a one-way hash function.
- Tags have an access-control method like schemes in [9], [10], to prevent unauthorized querying during idle time.
- Tags have no timer.
- The verifier and the reader are trustable

## C. *Initial setup*

Initial setup phase of the protocol is as follows:

- RFID system T = {$T_1$, $T_2$, ... , $T_n$} consists of *n* tags to be divided into pre-defined *m* subgroups T*sub* = {$Tsub_1$, $Tsub_2$, ...,$Tsub_m$} which have *gt* tags for each of them, where *n*, *m* and *gt* are the number of tags in T, the number of subgroups in *Tsub* and the number of tags for each subgroup, respectively.
- Each tag $T_i$ ($i^{th}$, $0 < i \leq n$) is initialized with *e*-bit $ID_i$, *d*-bit secret key $X_i$ and *c*-bit counter value $C_i$ which is initialized as 0.
- A verifier stores the initial parameters of tags to perform verification (*ID*, *X*, *C*, their subgroup and role of each tag in their subgroup) in its database VDB.
- A verifier has the initial parameters of a reader ($ID_r$, and $X_r$) to perform the mutual authentication between them. They can authenticate to each other.

The classification of the subgroups can be achieved via traditional generic query based tag scanning. When multiple tags reply to a query, it means that they are located in a near area to each other. The spatial data are employed to divide *n* tags into *m* subgroups, and stored in VDB as subgroup location information (SGLI).

## D. *Proposed offline yoking proof protocol*

Our offline verifiable yoking proof (OV-yoking proof) protocol consists of three logical phases as follows: (1) Tag-specific commitment value generation (2) Yoking proof generation offline, and (3) Offline verification. Fig. 2 depicts the protocol, and the followings are to explain steps of the proposed protocol to generate the yoking proof for a subgroup composed of a pair of tags, *TagA* and *TagB*.

### 1) Tag-specific commitment generation phase

- 1. After a successful mutual authentication phase between a reader *R* and a verifier *V*, *V* computes *k*-bit random value *r* , pre-computed proof $PP_{AB}$ = ($m_A$, $m_B$, *r*), and commitment *cmt* = $f_{X_R}$[$PP_{AB}$, *r*], and then provides the values to *R* with SGLI and access control authority of the tags.

### 2) Offline proof generation phase

- 2. *R* divides *r* into two parts ($r_{left}$ *and* $r_{right}$), where $r_{left}$ (or $r_{right}$) is the *k/*2- bit part of the most significant bits (or the least significant bits), and sends the left proof request to the initiate tag $Tag_A$ with $r_{left}$.
- 3. $Tag_A$ recognizes its role by seeing left proof query and computes *ra* and *ma* to $f_{X_A}$[$r_{left}$, $C_A$] and $f_{XA}$[*ra*, $ID_A$], respectively. $Tag_A$ then sends back the message $m_A$ containing both *ra* and *ma* to *R* and increase its counter $C_A$ by one.
- 4. *R* sends right proof request with $t_{right}$ and *ma* to the

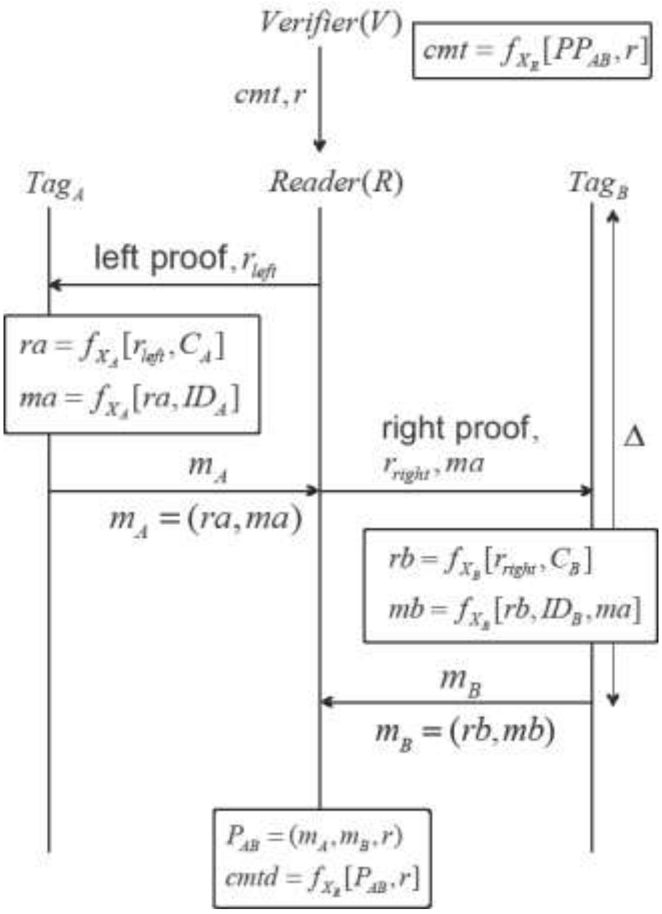

Figure 2. Offline verifiable yoking proof protocol (OV-yoking proof)

TABLE II. COMPUTATION COSTS OF TAGS

| Costs of tags | Yoking proof schemes for tag-privacy | | |
|---|---|---|---|
| | *Anonymous yoking* [4] | *Clumping proof* [5] | *OV-yoking proof* |
| *TagA* | *1CTIMER + 2Cf + 1CMAC + 1Cshift* | *1CTIMER +1Cf + 2CMAC + 1CXOR + 1Cshift* | *2CMAC + 1Cshift* |
| *TagB* | *1CTIMER + 1Cf + 1CMAC + 1Cshift* | *1CTIMER +1Cf + 1CMAC + 1CXOR + 1Cshift* | *2CMAC + 1Cshift* |
| Total | *2CTIMER + 1Cf + 1CMAC + 1Cshift* | *2CTIMER +2Cf + 3CMAC + 2CXOR + 2Cshift* | *4CMAC + 2Cshift* |

other tag $Tag_B$. $Tag_B$ calculates *rb* and *mb* as $f_{X_B}[r_{right}, C_B]$ and $f_{X_B}[rb, ID_B, ma]$, respectively, and then replies to the reader with the message $m_B$ that consists of *rb* and *mb*. It also increases its counter $C_B$ by one.

### 3) Offline verification phase

- 5. The reader verifies proof $P_{AB} = (m_A, m_B, r)$ by computing the commitment disclosure value $cmtd = f_{X_R}[P_{AB}, r]$. If *cmt* is equal to *cmtd*, the verification completes.

## E. *Extended Group yoking proof*

Our OV-yoking proof protocol can be easily extended to offline group yoking proof (OV-grouping proof) protocol without additional devices.

To extend our protocols for multiple tags, both left proof and right proof are replaced with sequential number to indicate a pre-defined sequential order of tags in their subgroup. When a reader sends queries with the sequential number, the tags respond accordingly, and group yoking proof is generated in the sequential order without additional devices, such as pallet tags in [2], [7].

# IV. Analysis

## A. *Security and privacy*

In this section, we show that our scheme prevents the attacks described in Section 3 while satisfying *R1*, *R2*, *SR1* and *SR2* (Section 1, 4). We assume an adversary *Adv* that tries to forge a proof through the attacks in Section 3 and to trace a tag. We model the adversary with the following lemma:

**Lemma:** *Adv can have hashes, such as* $ra_1$, $ma_2$, $rb_1$, *and* $mb_2$, *but cannot obtain their pre-images,* $X_i$, $ID_i$, *and* $C_i$ *from them.*
**Proof:** Given the random-oracle assumption, if *Adv* tries to guess *l*-bit hash (*ra*, *ma*, *rb*, *mb*) generated with secret parameters ($X_i$, $ID_i$, $C_i$), the probability of success is bounded by $2^l$. Therefore, the lemma holds.
**Claim 1**: Our protocol satisfies *R1* and *R2* against attacks to forge a yoking proof.
**Proof** (Case 1 - Replay attacks): $P_{AB}$ and *cmt* is generated with tag's reply messages ($m_A$ and $m_B$). The hashed messages are changed because of $C_A$ (or $C_B$) (Section 4, Step 3 and 4). Hence, *Adv* cannot reuse captured messages generated with the tag's reply messages.
**Proof** (Case 2 – Reassembled Proof): A reassembled proof cannot be verifiable, since all subgroups are already defined in *Tsub* and the verifier computes $PP_{AB}$ via the pre-knowledge of *Tsub* that is stored in VDB (Step 6). Therefore, *Adv* cannot use the reassembled proof.
**Claim 2**: Our protocol satisfies *SR1* against threats to reveal tag's secret parameters.
**Proof** (*Confidentiality*): Our scheme satisfies *SR1* via hashed reply messages (**Lemma**).
**Claim 3**: Our protocol satisfies *SR2* against threats to trace a specific tag.
**Proof** (*Unlinkability*): When two replies of tags are presented to *Adv*, she cannot know whether the replies are from the same tag or not because of the hashed reply message computed with increased counters ($C_A$ and $C_B$) (Section 4, Step 3 and 4). Therefore, our scheme satisfies *SR2*.

## B. *Efficiency*

(Computation cost of a tag): We compare the computational overhead of a tag to previous yoking proof protocols to deal with privacy issues [4], [5]. Table II gives the computation costs of a tag, where *CTIMER*, *Cf*, *CMAC*, *CXOR*, and *Cshift* are the cost for timer, hash (or PRNG), MAC, XOR, and shift operations, respectively. The result shows the proposed protocol outperforms the other schemes.

(Searching cost): Due to the randomization of a tag reply, the searching cost of the schemes in [5], [6], [8] is O(n) (in case of [2], the cost is O($n^2$)). In our OV-yoking proof protocol, the verification process requires no searching cost to identify the tags of the proof. Since the tag-specific commitment is pre-computed before the offline verification phase (Section 4, Step 5), the verification cost in our scheme is O(1).

# V. Conclusion

We propose a new method to verify a yoking proof without the online verifier. Our OV-yoking proof protocol provides the offline verification via spatial data of tags and commitment disclosure. The analysis shows that our offline yoking proof protocol is more secure against attacks on the tag side and more efficient than the previous online yoking proof protocols.

## *Acknowledgment*

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012R1A1B3004161)

## *References*

[1] A. Juels, "Yoking Proofs for RFID tags," First International Workshop on pervasive Computing and Communication Security, pp.621-624, 2004.

[2] J. Saito and K. Sakurai, "Grouping proof for RFID tags," In 19th International Conference on Advanced Information Networking and Applications 2005, volume 2, pp.621-624, 2005.

[3] S. Piramuthu, "On Existence Proofs for Multiple RFID Tags," IEEE International Conference on Pervasive Services, pp.317-320, 2006.

[4] L. Bolotnyy and G. Robins, "Generalized Yoking Proofs for a Group of RFID Tags," International Conference on Mobile and Ubiquitous Systems, pp.1-4, 2006.

[5] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, and A. Ribagorda, "Solving the simultaneous scanning problem anonymously: clumping proofs for RFID tags," In Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, SecPerl07, pp.55-60, 2007.

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

[7] Yuanhung Lien, Xuefei Leng, Mayes, K. Jung-Hui Chiu, "Reading order independent grouping proof for RFID tags," Intelligence and Security Informatics, pp.128-136, pp.17-20, 2008.

[8] Mike Burmester, Breno Medeiros, Rossana Motta, "Provably Secure Grouping-Proofs for RFID Tags," In Proceedings of the 8th IFIP international conference on Smart Card Research and Advanced Applications, pp.176-190, 2008.

[9] "EPCglobal class1 gen2 RFID specifications," Whitepaper, 2008.

[10] K. Finkenzeller, "RFID Handbook, Fundamentals and Applications in Contactless Smart Cards and Identification," John Wiley and Sons Ltd, pp.226-232, 2003.