# A Provably Secure Offline RFID Yoking-Proof Protocol with Anonymity

Daisuke Moriyama(✉)

National Institute of Information and Communications Technology, Tokyo, Japan
dmoriyam@nict.go.jp

**Abstract.** Yoking-proof in a Radio Frequency Identification (RFID) system provides the evidence that two RFID tags are simultaneously scanned by the RFID reader. Though there are numerous yoking-proof protocols, vulnerabilities related to security and privacy are found in many prior works. We introduce a new security definition that covers the man-in-the-middle (MIM) attack, and a privacy definition based on an indistinguishability framework. We also provide a simple construction of a provably secure offline yoking-proof protocol based on the pseudorandom function.

**Keywords:** RFID · Authentication · Yoking proof · Provable security

## 1 Introduction

Radio Frequency Identification (RFID) technology enables identification of objects with wireless communication. When a passive RFID tag is attached to the object and electricity is supplied from the RFID reader, a communication channel between the tag and reader is established and they can exchange messages. Since RFID tags are widely used in various commercial activities (e.g., logistics, transportation, product management), tracking of RFID tags by an authorized manager is a fundamental issue of concern. Notably, Juels introduced an RFID yoking-proof protocol in which two RFID tags are simultaneously scanned by the reader and coexistence of the tags is provided by the communication through the reader [13]. When the yoking-proof protocol is generalized so that a group of tags is communicated via the reader and generates a proof by which they all engage in the protocol, it is called a "grouping-proof protocol" [4, 22].

Afterwards, various yoking-proof and grouping-proof protocols have been proposed, but almost all of them were broken. Though Juels proposed two yoking-proof protocols, both are vulnerable to replay attacks in which an adversary combines flows from different sessions [4, 22]. Piramuthu showed that the grouping proof protocol proposed in [22] is still vulnerable to a replay attack [19]. Subsequently, Peris-Lopez et al. introduced a multi-proof session replay attack in [20] and provided an attack against the Piramuthu's protocol [19]. Burmester et al. proposed a provably secure yoking-proof protocol [3], but Peris et al. pointed out

T. Eisenbarth and E. Öztürk (Eds.): LightSec 2014, LNCS 8898, pp. 155–167, 2015.
DOI: 10.1007/978-3-319-16363-5_10

that their protocol is vulnerable to multiple impersonation attacks [21]. They also mentioned that the yoking-proof protocol proposed by Chien and Liu [6] has a privacy problem and the grouping proof protocol proposed by Huang and Ku [10] is not secure against impersonation attacks. Though Peris-Lopez et al. gave a guideline for constructing secure yoking-proof and grouping-proof protocols and proposed a new yoking-proof protocol, recently Bagheri and Safkhani showed that the tag's secret key can be calculated from the communication message in their protocol [5]. Batina et al. proposed a grouping-proof protocol based on public key cryptography [2], and Hermans and Peeters showed that an impersonation attack and man-in-the-middle (MIM) attack could be launched against it [11].

**Our Contributions.** In this paper we focus on a provably secure offline RFID yoking-proof protocol. Our contributions are twofold. First, we investigate a rigorous security model for yoking-proof protocols based on the security and privacy definition for canonical RFID authentication protocols [7,15,17]. Though [3] proposed a security model based on the universal composability (UC) framework, it is widely known that it is difficult to provide a security proof in the UC framework. Hermans and Peeters provided another security model in [11] that is based on the existing security model for the canonical RFID authentication protocols [12], but this definition only captures the impersonation attack on security and the adversary cannot learn whether the resulting yoking proof is valid. Different from the above prior security models, our security model covers general MIM attacks in the security definition so that a malicious adversary can interact with all RFID tags at any time and modify the communication. An indistinguishability-based privacy definition is also proposed to provide anonymity for the RFID tags.

Next, we show an example of a yoking-proof protocol that satisfies the proposed security model. Notably, our protocol is robust against an MIM attack that includes all attacks described in previous works [4,11,19,21,22], and satisfies the requirement of anonymity such that no information about the tag is leaked from the communication message. Compared to the existing offline yoking-proof protocol proposed by Hermans and Peeters [11], which has provable security, our protocol does not require public key cryptography and its main building block is a secure pseudorandom function. Thus, our protocol is quite efficient and easier to implement in low-cost RFID tags.

## 2 Security Model for RFID Yoking-Proof Protocols

Though there are many yoking-proof and grouping-proof protocols, there is no widely known security model and this area is still under development. In this paper, we formalize two basic requirements for yoking-proof protocols, correctness and security. In addition, we also consider the privacy issue as an additional property. Our security model for yoking-proof protocols is motivated by the security model for basic RFID authentication protocols [15]. However, we do not intentionally cover the tag corruption in the following definition because almost all previous works does not satisfy a classical man-in-the-middle attack nor privacy.

### 2.1 Execution Model

A yoking-proof protocol in an RFID system is executed among the verifier $\mathcal{V}$, RFID reader $\mathcal{R}$ and multiple RFID tags in $\mathcal{T} := \{t_0, t_1, \ldots, t_n\}$. The yoking-proof protocol consists of three phases: setup, generation and verification. In the setup phase, the verifier runs a setup algorithm with security parameter $k$ and obtains public parameter and secret keys. If the protocol is based on symmetric-key cryptography, the verifier shares the secret key with the RFID tags. In the generation phase, two RFID tags generate a yoking-proof via the reader. Then, two RFID tags in the same group execute an interactive protocol and finally the reader outputs a proof. The verifier checks the validity of the proof in the verification phase. If the verification is accepted, the verifier outputs 1 (acceptance); otherwise, it outputs 0 (rejection) as the verification result. In the following, we concentrate on the offline yoking-proof protocol wherein the verifier does not participate in the generation phase.

We consider that each session of the party (RFID reader and individual tags) is identified by a session identifier $\mathsf{sid}$, that contains the collected of the input/output messages of the party. A session is called finished when the party outputs the final message in the session. We say that a party $A$ has a matching session with the other party $B$ if all communication messages between the two are honestly transferred. The correctness of the yoking-proof protocol is that the verifier always accepts the yoking proof if it is generated by the session wherein two tags in the same group have a matching session with the peer.

We note that the role of the reader is different from the canonical RFID authentication protocol in which the RFID reader authenticates the RFID tag and outputs the authentication result. Therefore, the RFID reader actively participates in the protocol. On the other hand, the reader in the yoking-proof and grouping-proof protocols does not authenticate tags[1] and its task is to transfer the message among tags and obtain the yoking proof from the communication message. Instead, a verifier checks the validity of the yoking proof after a session is finished and the reader submits the yoking proof. Thus it is trivial that an adversary impersonates a reader and this is not a security issue in yoking-proof and grouping-proof protocols. Therefore, the protocol procedure is quite different and the existing security model for typical RFID authentication protocols is not directly applicable to the yoking-proof protocols.

### 2.2 Security

Intuitively, the security of the yoking-proof protocol requires that the verifier always rejects the yoking proof if it is not available in one honest protocol execution between the RFID tags. When there is an active adversary that can interfere, delay, interleave and modify the communication message, we can consider there are two security levels: resistance against the impersonation attack and

[1] Tag authentication by the reader can be one application, but it is not a necessary issue in yoking-proof protocols.

MIM attack. Hermans and Peeters [11] introduced a formal security model for the RFID yoking-proof protocol that captures the impersonation attack. In their security definition, the adversary can communicate with all tags in the learning phase. After that, the adversary cannot interact with one of the uncorrupted tags and the goal of the adversary is to impersonate the tag and output a valid yoking proof. Though the above impersonation resistance is also widely known as an active attack, it does not imply security against an MIM attack. Notably, several lightweight authentication protocols [1,8,14] are provably secure against an active adversary but vulnerable to a (general) MIM attack [8,9,18]. In this paper, we formalize security against a general MIM attack in RFID yoking-proof protocols.

When a general MIM attack is launched, the adversary can interact with all tags at any time. The adversary's goal is to output a valid yoking proof that is not generated in the sessions wherein the tag has a matching session to the other tag. We note that the above condition does not mean that the reader has no matching session to one of the RFID tags. Whereas the reader always transmits the communication messages between the RFID tags, the adversary can directly deliver the tag's output to the other tag. Even when the adversary sends an arbitrary random message to the reader in the session and the reader does not have a matching session, the adversary can obtain a valid yoking proof derived from the tags.

More formally, we provide the following general security experiment $\mathsf{Exp}^{\mathsf{Sec}}_{\Pi,\mathcal{A}}(k)$ between a challenger and adversary $\mathcal{A}$ against an RFID yoking-proof protocol $\Pi$.

**Setup.** The challenger runs the setup algorithm and provides a public parameter to the adversary.

**Learning.** Then $\mathcal{A}$ can then adaptively issue the following queries to interact with the reader, tags and verifier:

- $\mathsf{Launch}(1^k)$: Launch the reader to start a new session.
- $\mathsf{SendReader}(m)$: Send an arbitrary message $m$ to the reader.
- $\mathsf{SendTag}(t, m)$: Send an arbitrary message $m$ to the tag $t \in \mathcal{T}$.
- $\mathsf{Result}(\sigma)$: Output whether the verifier accepts the yoking-proof $\sigma$.

**Guess.** When the adversary finishes the interaction $\mathcal{A}$ outputs $(t_0^*, t_1^*, \sigma^*)$ where $(t_0^*, t_1^*) \subseteq \mathcal{T}$. The challenger outputs 1 if $\mathsf{Result}(\sigma^*) = 1$ and $\sigma^*$ is not derived from the session in which $t_0^*$ has the matching session to $t_1^*$. Otherwise, the challenger outputs 0.

The probability that the adversary wins the above security game is denoted by $\mathsf{Adv}^{\mathsf{Sec}}_{\Pi,\mathcal{A}}(k) := \Pr[\mathsf{Exp}^{\mathsf{Sec}}_{\Pi,\mathcal{A}}(k) \to 1]$.

**Definition 1.** *An RFID yoking-proof protocol $\Pi$ is secure against general MIM attacks if for any probabilistic polynomial time adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{Sec}}_{\Pi,\mathcal{A}}(k)$ is negligible in $k$.*

We note that a general MIM attack covers all previous attacks against previous works [4,11,19,21,22]. One typical approach of replay attacks is that a malicious adversary captures the past communication messages and reuses them in the target session. In the impersonation attack the adversary sends adversarial messages to the tag and obtains a meaningful information. Our security definition does not restrict any strategy from the view point of the adversary and only gives one exception wherein $\sigma^*$ honestly generated by two tags in one session cannot be submitted since it is trivial in terms of the correctness property.

Different from the existing security definition for RFID authentication protocols, the final goal of the adversary is not to submit an acceptable message to the reader but to output a forged yoking proof to the offline verifier. We therefore define the above experiment as the security definition against digital signature or message authentication code schemes.

### 2.3 Privacy

One application of the RFID yoking-proof protocol is to cover anonymity. Though the yoking-proof system is useful to prove when the tag interacts with the reader from the view point of the honest verifier, it is not desirable that a malicious adversary can trace the tag. Even if the adversary cannot verify the yoking proof two tags generate, several previous works specify that the tag explicitly outputs its identity to the reader, so the adversary can learn which tag tries to generate a yoking proof. Since the tag's anonymity is a critical issue in RFID authentication protocols, it is useful to consider privacy in the related topics.

In this paper we formalize the privacy definition for RFID yoking-proof protocols based on indistinguishability-based privacy for canonical RFID authentication protocols [7,15,17]. Consider the following privacy experiment against an RFID yoking-proof protocol $\Pi$ between a challenger and adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$:

**Setup.** The challenger runs the setup algorithm to initialize the verifier, reader and tag. The adversary obtains public parameter $pp$ and the identities of the reader and tag $(\mathcal{R}, \mathcal{T})$.

**Phase 1.** The adversary $\mathcal{A}_1$ can issue $\mathcal{O}{:=}\{\mathsf{Launch}, \mathsf{SendReader}, \mathsf{SendTag}, \mathsf{Result}\}$ to communicate with the reader and tag and check the validity of the yoking proof as a security game.

**Challenge.** $\mathcal{A}$ sends two sets of tags $\mathcal{T}_0^*$ and $\mathcal{T}_1^*$ $(\mathcal{T}_0^* \neq \mathcal{T}_1^* \ \wedge \ |\mathcal{T}_0^*| = |\mathcal{T}_1^*| = 2)$ to the challenger and outputs state information $st$. The challenger flips a coin $b \xleftarrow{\mathsf{U}} \{0,1\}$ and sets $\mathcal{T}'{:=}\mathcal{T} \setminus \{\mathcal{T}_0^*, \mathcal{T}_1^*\}$.

**Phase 2.** The adversary $\mathcal{A}_2$ receives $st$ and continues to interact with $\mathcal{R}$ and $\mathcal{T}'$ as Phase 1. If the adversary wants to send message $m$ to a tag in the group $\mathcal{T}_b^*$, he issues $\mathsf{SendTag}((\mathcal{I}, i), m)$ with an intermediate algorithm $\mathcal{I}$. $\mathcal{I}$ relays the communication between $\mathcal{A}_2$ and the $i$-th member of $\mathcal{T}_b^*$ to prevent the adversary from directly interacting with the target tag.

**Guess.** Finally, the adversary $\mathcal{A}_2$ outputs $b'$.

The adversary wins the above game if $b' = b$ holds and two tags in $\mathcal{T}_0^*/\mathcal{T}_1^*$ belong to the same group. Depending on the flipped coin $b$, we define the above experiment as $\mathsf{Exp}^{\mathsf{IND}}_{\Pi,\mathcal{A}}b(k)$. Then the advantage of the adversary is defined by

$$\mathsf{Adv}^{\mathsf{IND}}_{\Pi,\mathcal{A}}(k) = \left| \Pr[\mathsf{Exp}^{\mathsf{IND\text{-}0}}_{\Pi,\mathcal{A}}(k) \rightarrow 1] - \Pr[\mathsf{Exp}^{\mathsf{IND\text{-}1}}_{\Pi,\mathcal{A}}(k) \rightarrow 1] \right| .$$

**Definition 2.** *An RFID yoking-proof protocol $\Pi$ satisfies IND-privacy if for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{IND}}_{\Pi,\mathcal{A}}(k)$ is negligible in $k$.*

Recall that $\mathcal{T}$ contains identities of all RFID tags. Since there are many groups in $\mathcal{T}$, the adversary may want to distinguish between the group of RFID tags. Consider that there are two groups $A \subset \mathcal{T}$ and $B \subset \mathcal{T}$. If $\mathcal{T}_0^* \subseteq A$ and $\mathcal{T}_1^* \subseteq B$, the above definition claims that the adversary cannot distinguish between two groups even when it obtains the communication messages anonymously. On the other hand, if $\mathcal{T}_0^* \subset A$ and $\mathcal{T}_1^* \subset A$ for a group $A$, this means that the adversary tries to distinguish the tag frin among the members in $A$; thus, no information related to the group's or tag's identity should be leaked from the communication messages to satisfy the requirement of IND-privacy.

In this paper we do not formalize the forward secrecy so that a malicious adversary corrupts the RFID tag and obtains the internal secret. Since we focus on the offline yoking-proof protocol, it is not trivial for RFID tags to securely update the secret key with the offline verifier. The key updating mechanism is certainly an additional issue, and we leave this issue as an open problem. Since almost all the previous symmetric-key based protocols are broken and the security level of the existing provably secure yoking-proof protocol [11] is only an impersonation attack, the first task is to show a yoking-proof protocol provably secure against a general MIM attack.

## 3 Proposed Yoking-Proof Protocol

We present a new RFID yoking-proof protocol, provably secure against a general MIM attack and satisfies IND-privacy. It is based on the previous yoking-proof protocol proposed by Bermester et al. [3] and supports group authentication during yoking-proof generation. Assume that a secure pseudorandom function $\mathsf{PRF} : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^k$ is implemented in all RFID tags. The proposed protocol proceeds as follows:

**Setup Phase.** The verifier $\mathcal{V}$ randomly selects group secret key $k_X \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$ for each group and individual secret key $k_i \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$ for each tag. Tag $t_i$ receives $(k_X, k_i)$ from the verifier. Reader $\mathcal{R}$ selects a maximum delay time $\delta$ that denotes the upper bound of the execution for each session.

**Generation Phase.** For simplicity, we consider that tag $t_1$ and $t_2$ communicate with each other to generate the yoking proof.

1. Reader $\mathcal{R}$ sends time stamp $\mathsf{ts}$ and random nonce $r \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$ to tag $t_1$. $\mathcal{R}$ also starts an internal time clock.
2. Upon receiving $(\mathsf{ts}, r)$ from $\mathcal{R}$, $t_1$ randomly chooses $r_1 \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$ and computes $u_1 := \mathsf{PRF}(k_X, (\mathsf{ts}, r, r_1))$. $t_1$ sends $(r, r_1, u_1)$ to $\mathcal{R}$.
3. $\mathcal{R}$ sends $(\mathsf{ts}, r, r_1, u_1)$ to $t_2$.
4. Upon receiving $(\mathsf{ts}, r, r_1, u_1)$ from $\mathcal{R}$, $t_2$ verifies $u_1 = \mathsf{PRF}(k_X, (\mathsf{ts}, r, r_1))$. If this verification holds, $t_2$ randomly chooses $r_2 \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$ and computes

$u_2 := \mathsf{PRF}(k_X, (\mathsf{ts}, r, r_1, r_2))$ and $v_2 := \mathsf{PRF}(k_2, (\mathsf{ts}, r, r_1, r_2))$. Otherwise, $t_2$ randomly selects $(r_2, u_2, v_2) \overset{\mathsf{U}}{\leftarrow} \{0,1\}^{3k}$. $t_2$ sends $(r, r_2, u_2, v_2)$ to $\mathcal{R}$.

5. $\mathcal{R}$ sends $(r_1, r_2, u_2)$ to $t_1$.
6. Upon receiving $(r_1, r_2, u_2)$ from $\mathcal{R}$, $t_1$ checks that there is an unfinished session where $t_1$ outputs $r_1$ in the first round and verifies $u_2 := \mathsf{PRF}(k_X, (\mathsf{ts}, r, r_1, r_2))$. If this verification holds, $t_1$ computes $v_1 := \mathsf{PRF}(k_1, (\mathsf{ts}, r, r_1, r_2))$. Otherwise, $t_1$ selects $v_1 \overset{\mathsf{U}}{\leftarrow} \{0,1\}^{k}$. $t_1$ sends $(r, v_1)$ to $\mathcal{R}$.
7. $\mathcal{R}$ outputs $\sigma := (\mathsf{ts}, r, r_1, r_2, u_2, v_1, v_2)$ as a yoking proof if the above execution is finished within $\delta$. Otherwise, $\mathcal{R}$ aborts the session.

**Verification Phase.** The verifier checks $u_2 = \mathsf{PRF}(k_X, (\mathsf{ts}, r, r_1, r_2))$ for a group secret key $k_X$. If this verification holds, $\mathcal{V}$ checks $v_1 = \mathsf{PRF}(k_1, (\mathsf{ts}, r, r_1, r_2))$ and $v_2 = \mathsf{PRF}(k_2, (\mathsf{ts}, r, r_1, r_2))$ for some individual secret keys $k_1$ and $k_2$ sent to the group member. If these verifications hold, $\mathcal{V}$ accepts the yoking proof and outputs 1. If one of the three verifications fails, $\mathcal{V}$ outputs 0.

The main building block of our protocol is the pseudorandom function. Because it is natural for the existing RFID authentication protocols that the RFID tag can compute symmetric key cryptography, we do not restrict here that the computational resource of RFID tags must be EPC-compliant. In particular, it is quite hard to provide a security proof for EPC-compliant protocols because such tags can only execute basic operations such as AND, OR, XOR, or cyclic-shift. Though we extend a protocol in [3] that is vulnerable to the multiple impersonation attacks [21], our protocol specifies that $(u_2, v_1, v_2)$ is computed with fresh nonces selected by each tag and an impersonation attack always fails. One of the two nonces is always selected by itself and the other nonce restricts the peer of the session; so our protocol is secure against a general MIM attack (see more detailed discussion in the next section).

If the tag's identity is also input to $\mathsf{PRF}$, each tag can verify which member of the group executes the yoking-proof protocol. However, it is inefficient for the tag to run an exhaustive search to check the actual peer in the group when the group member is quite large (such as for the role of the reader in the canonical RFID authentication protocol). Even when the tag's identity is not included, each tag can check whether a valid tag in the same group tries to execute the yoking proof in our protocol.

In the above protocol, time stamp $\mathsf{ts}$ and nonce $r$ are always input to the pseudorandom function. The role of the time stamp is to specify when the reader invokes the session and the verifier learns it, while the time stamp is useless for the communicating RFID tags. On the other hand, $r$ is used to ensure the concurrent execution of the session. Especially, Lin et al. [16] considered the situation in which the tag communicates with multiple readers and executes many sessions concurrently. When some readers start different sessions at the same time, the time stamp may be recorded. However, the random nonce $r$ is also chosen at random and is unique for each reader (the collision probability is at most $2^{-k}$). Therefore, the communication message output from each tag contains $r$ to distinguish between multiple readers. Similarly, $t_1$ receives $r_1$ from $\mathcal{R}$ even after $r_1$ is sent to $\mathcal{R}$. Different from tag $t_2$, $t_1$ starts a new session whenever it receives $(\mathsf{ts}, r)$ and must keep sessions until the reader gives the response from $t_2$ to $t_1$. If we omit $r_1$ from $(r_1, r_2, u_2)$ transmitted from $\mathcal{R}$, $t_1$ cannot learn which session should be verified in the concurrent execution, so some extra information to distinguish multiple sessions is needed to transfer messages. Since $(r, r_1)$ is chosen randomly for each session and can be used as unique identifier, these nonces are contained in the communication message to support concurrent execution.

We note that the validity of $v_1$ and $v_2$ cannot be checked by the tags in our protocol and one may think that there is a chance for a malicious adversary to change these variables. Actually, the tags cannot detect the man-in-the-middle attack. But $v_1$ and $v_2$ are used to check by the (legitimate) verifier and these values are strictly determined by the time stamp and nonces related to the session. Since the goal of the adversary is to violate the security or privacy as explained in the previous section, no critical problem occurs in our protocol.

## 4 Security Proof

We prove that our protocol satisfies the security model described in Sect. 2. It is clear that the proposed protocol described in the previous section satisfies correctness.

**Theorem 1.** *Our yoking-proof protocol is secure against a general MIM attack if* $\mathsf{PRF}$ *is a secure pseudorandom function.*

*Proof.* We show that if an adversary $\mathcal{A}$ wins the security game described in Sect. 2, there is an algorithm $\mathcal{B}$ that breaks the security of pseudorandom function $\mathsf{PRF}$. $\mathcal{B}$ can issue oracle queries to the function that is actual pseudorandom function $\mathsf{PRF}(k_1, \cdot)$ or a truly random function, and the goal of $\mathcal{B}$ is to distinguish between them. Consider that the adversary outputs a set of the tag's identities $(t_1^*, t_2^*)$ and yoking proof $\sigma^* := (r^*, r_1^*, r_2^*, u_2^*, v_1^*, v_2^*)$. To satisfy $\mathsf{Result}(\sigma^*) = 1$, all verifications must be passed so that $u_2^* = \mathsf{PRF}(k_X, (r^*, r_1^*, r_2^*))$, $v_1^* = \mathsf{PRF}(k_1^*, (r^*, r_1^*, r_2^*))$ and $v_2^* = \mathsf{PRF}(k_2^*, (r^*, r_1^*, r_2^*))$ where $(k_1^*, k_2^*)$ is a secret key of $(t_1^*, t_2^*)$, respectively. On the other hand, $t_1^*$ does not have matching session to $t_2^*$ in the related session and the adversary cannot simply forward the communication message between these tags. The strategy of the adversary is divided into two cases:

Case 1: $\mathcal{A}$ outputs a valid $v_1^*$ while $v_1^*$ does not appear in the $t_1^*$'s output.
Case 2: $\mathcal{A}$ outputs $v_1^*$ derived from $t_1^*$.

When Case 1 occurs, we can construct an algorithm $\mathcal{B}$ that breaks the security of pseudorandom function $\mathsf{PRF}(k_1, \cdot)$. $\mathcal{B}$ internally runs $\mathcal{A}$ and simulates the above protocol. $\mathcal{B}$ selects all group secret keys and individual secret keys except for $t_1^*$'s secret key. $\mathcal{B}$ honestly simulates all communication messages except the case that $t_1^*$ is activated and it outputs a final message in the session. When $\mathcal{A}$ issues $\mathsf{SendTag}(t_1^*, (r_1, r_2, u_2))$, $\mathcal{B}$ checks $u_2 = \mathsf{PRF}(k_X, (r, r_1, r_2))$ and sends $(r, r_1, r_2)$

to the challenger of the pseudorandom function. When $\mathcal{B}$ receives the response $v_1$ from the challenger, $\mathcal{B}$ transfers $(r, v_1)$ to $\mathcal{A}$. When $\mathcal{A}$ outputs a forged yoking-proof $\sigma^*$, $\mathcal{B}$ issues $(r^*, r_1^*, r_2^*)$ to the challenger and obtains $v^*$. If $v^* = v_1^*$, $\mathcal{B}$ outputs 1 and halts the simulation. Otherwise, $\mathcal{B}$ outputs 0.

If $\mathcal{B}$ interacts with the actual pseudorandom function $\mathsf{PRF}(k_1^*, \cdot)$, the probability $v^* = v_1^*$ holds is non-negligible since we assume that $\mathcal{A}$ outputs a valid yoking proof. Thus $\mathcal{B}$ outputs 1 with non-negligible probability. On the other hand, if the challenger selects a truly random function, it is impossible to guess the valid output and $\mathcal{B}$ outputs 1 with negligible probability $1/2^{-k}$. Therefore $\mathcal{B}$ can break the security of the pseudorandom function.

We note that the same argument can be applied to $v_2^*$. If $v_2^*$ is not honestly generated by $t_2^*$, the adversary cannot output a valid yoking-proof based on the security of the pseudorandom function. On the other hand, $t_2^*$ outputs $v_2^* = \mathsf{PRF}(k_X, (\mathsf{ts}^*, r^*, r_1^*, r_2^*))$ if and only if $t_2^*$ receives $(\mathsf{ts}^*, r^*, r_1^*, u_1^*)$ and selects $r_2^*$. Since $(u_2^*, v_1^*, v_2^*)$ is checked with the same input $(\mathsf{ts}^*, r^*, r_1^*, r_2^*)$ and deterministically defined, the adversary cannot output forged yoking-proof that is not output by $(t_1^*, t_2^*)$. □

One may think that the verification check by $t_2$ in Fig. 1 can be passed even when the adversary issues another tag $t_3$ in the same group and transfers the message $(r, r_1, u_1)$. However, $(u_2, v_2)$ is computed with random nonces $r_1, r_2$ that specify that only the party that generates $r_1$ accepts the session with verification check of $u_2$. $t_1$ clearly rejects the session when the adversary sends $(r_1, r_2, u_2)$ to $t_1$, so $u_2$ binds the participants of the session. Similarly, while $t_1$ also accepts any messages generated by the same group, the output $v_1$ for each session rigorously

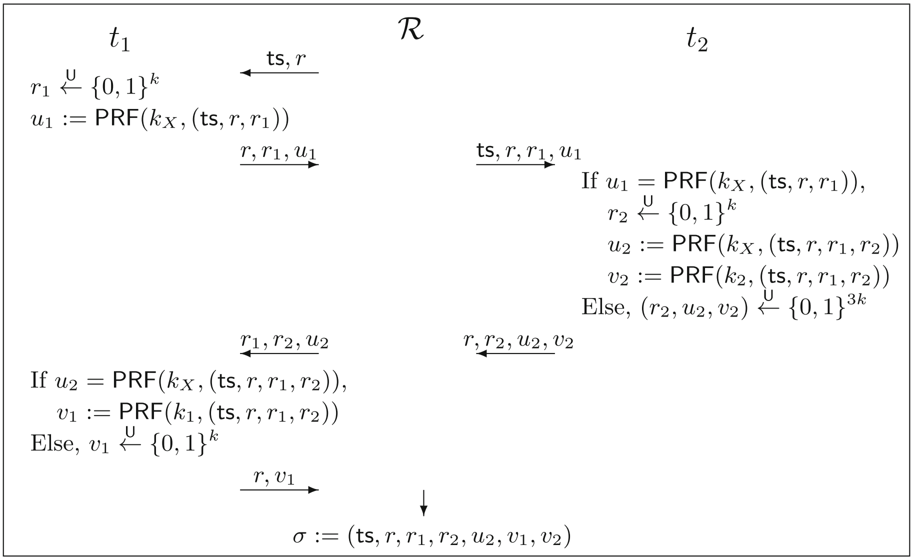
$t_1$
$\mathcal{R}$
$t_2$
$\mathsf{ts}, r$
$r_1 \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$
$u_1 := \mathsf{PRF}(k_X, (\mathsf{ts}, r, r_1))$
$r, r_1, u_1$
$\mathsf{ts}, r, r_1, u_1$
If $u_1 = \mathsf{PRF}(k_X, (\mathsf{ts}, r, r_1))$,
$r_2 \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$
$u_2 := \mathsf{PRF}(k_X, (\mathsf{ts}, r, r_1, r_2))$
$v_2 := \mathsf{PRF}(k_2, (\mathsf{ts}, r, r_1, r_2))$
Else, $(r_2, u_2, v_2) \overset{\mathsf{U}}{\leftarrow} \{0,1\}^{3k}$
$r_1, r_2, u_2$
$r, r_2, u_2, v_2$
If $u_2 = \mathsf{PRF}(k_X, (\mathsf{ts}, r, r_1, r_2))$,
$v_1 := \mathsf{PRF}(k_1, (\mathsf{ts}, r, r_1, r_2))$
Else, $v_1 \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$
$r, v_1$
$\sigma := (\mathsf{ts}, r, r_1, r_2, u_2, v_1, v_2)$

**Fig. 1.** The proposed yoking-proof protocol

restricts that the peer honestly receives $r_1$ (generated by $t_1$ itself) and validity check by the verifier is accepted only if $(u_2, v_2)$ and $v_1$ are computed by the same input. Therefore, even if the adversary transfers the communication message used in a different session, the verifier always rejects the yoking proof.

**Theorem 2.** *Our protocol satisfies IND-privacy if* $\mathsf{PRF}$ *is a secure pseudorandom function.*

*Proof.* Intuitively, communication messages derived from the tag in our protocol consist of only random nonces and outputs from the pseudorandom function. In addition, fresh nonces are selected for each individual session and the adversary cannot find any correlation between the sessions (e.g., whether the same tag executes the two sessions). We show the formal security proof of our protocol with the game transformation technique. Recall that $n$ denotes the number of RFID tags in the system. We consider that the number of group in the yoking-proof protocol is denoted as $n'$.

**Game 0.** This is the original privacy experiment between a challenger and adversary $\mathcal{A}$.

**Game 1-$j$.** For each $1 \leq i \leq j$, the challenger assigns random variables $(u_1, u_2) \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$ instead of $\mathsf{PRF}(k_X, \cdot)$ that $i$-th group member computes for any session.

**Game 2-$j$.** For each $1 \leq i \leq j$, the challenger assigns random variable $v_i \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$ instead of the $t_i$'s computation $\mathsf{PRF}(k_i, \cdot)$.

Let $S_i$ be a probability that the adversary wins the privacy experiment in Game $i$.

**Lemma 1.** *We have* $|S_{1\text{-}(j-1)} - S_{1\text{-}j}| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{PRF}}(k)$.

If the final output of the adversary $\mathcal{A}$ is different between Game 1-$(j-1)$ and Game 1-$j$, there is an algorithm $\mathcal{B}$ that breaks the security of the pseudorandom function. $\mathcal{B}$ generates all secret keys except $k_X$ and internally runs $\mathcal{A}$. If an activated tag belongs to the $i$-th group where $i < j$, $\mathcal{B}$ always selects random variables $(u_1, u_2) \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$ instead of computing pseudorandom function and proceeds with the simulation. If an activated tag belongs to the $i$-th group where $i > j$, $\mathcal{B}$ honestly computes $(u_1, u_2)$ with an actual pseudorandom function $\mathsf{PRF}(k_{X'}, \cdot)$ where $K_{X'}$ is a group secret key of the $i$-th group. When one of the members of $j$-th group is activated with input $r$, $\mathcal{B}$ proceeds as follows. $\mathcal{B}$ selects $r_1 \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$ and sends $(\mathsf{ts}, r, r_1)$ to the challenger. Upon receiving $u_1$ from the challenger, $\mathcal{B}$ sets $(r, r_1, u_1)$ as the tag's output. When $\mathcal{A}$ sends $(\mathsf{ts}, r, r_1, u)$ to a member of the $i$-th group, $\mathcal{B}$ issues $(\mathsf{ts}, r, r_1)$ to the challenger and compares its response with $u$. If it holds, $\mathcal{B}$ selects $r_2 \overset{\mathsf{U}}{\leftarrow} \{0,1\}^k$, issues $(\mathsf{ts}, r, r_1, r_2)$ to the challenger and obtains $u_2$. $v_2$ can be computed since $\mathcal{B}$ chooses an individual secret key by itself and $(r, r_2, u_2, v_2)$ is assigned as the output from the tag. When $\mathcal{A}$ sends $(r'_1, r'_2, u'_2)$ to a member of the $i$-th group, $\mathcal{B}$ checks whether the tag previously outputs $r'_1$. If so, $\mathcal{B}$ finds the corresponding nonce $r$ and issues $(r, r'_1, r'_2)$ to the oracle to compare the response with $u'_2$. $\mathcal{B}$ proceeds with the above simulation regardless of the choice of the two sets of tags $\mathcal{A}$ sends in the challenge phase. When $\mathcal{A}$ outputs a bit $b$, $\mathcal{B}$ stops the simulation and outputs the same bit.

If the challenger gives actual pseudorandom function $\mathsf{PRF}(k_X, \cdot)$ to $\mathcal{B}$, the above simulation is equivalent to Game 1-$(j-1)$. Otherwise, if $\mathcal{B}$ interacts with truly random function, the outputs of the $j$-th member are coming from the random function and it is equivalent to Game 1-$j$. Therefore, if $\mathcal{A}$ distinguishes the difference between Game 1-$(j-1)$ and Game 1-$j$, $\mathcal{B}$ can break the security of the pseudorandom function $\mathsf{PRF}$.

**Lemma 2.** *We have* $|S_{2\text{-}(j-1)} - S_{2\text{-}j}| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{PRF}}(k)$.

The proof strategy of Lemma 2 is analogous to the proof of Lemma 1 so we omit the details. When tag $t_j$ is activated we replace the communication message derived from the pseudorandom function to random strings. If an adversary finds a gap between these games, the security of the pseudorandom function can be broken.

Since Game 0 and Game 1-$n'$ can be considered as Game 1-0 and Game 2-0 respectively, we can transform Game 0 to Game 2-$n$ based on the security of the pseudorandom function. When we proceed with Game 2-$n$, there is no chance for the adversary to distinguish between the RFID tags. In this game, all communication messages generated by the tags in $\mathcal{T}$ consist of nonces freshly chosen per session and random strings derived from a truly random function, so no information about the tag's identity is observed in the communication message (including anonymous interaction). We can therefore say that $S_{2\text{-}n} = 0$.

Eventually, we have $\mathsf{Adv}_{\Pi,\mathcal{A}}^{\mathsf{IND}}(k) \leq (n' + n) \cdot \mathsf{Adv}_{\mathcal{B}}^{\mathsf{PRF}}(k)$. □

## 5 Conclusions and Future Work

In this paper, we introduced a new security model for an RFID yoking-proof protocol. Our model formalizes the security against a general MIM attack that covers all previous attacks for yoking-proof protocols. The indisinguishability-based privacy is also defined, which captures the RFID tags's anonymity. The example protocol given in this paper does not require public key cryptography and is simply described with a secure pseudorandom function.

Since we focus on the basic provably secure offline yoking-proof protocol based on the symmetric key primitive, the possibility of the key exposure problem of the RFID tag is ignored in this paper. Since the verifier is offline and cannot participate in the yoking-proof generation, updating the shared secret with the RFID tag to satisfy the requirement of forward secrecy is an open problem. Another problem tag corruption causes will be an impersonation attack by the corrupted tag in the same group.

## References

1. Bringer, J., Chabanne, H., Dottax, E.: HB++: a lightweight authentication protocol secure against some attacks. In: SecPerU 2006, pp. 28–33 (2006)
2. Batina, L., Lee, Y.K., Seys, S., Singelée, D., Verbauwhede, I.: Extending ECC-based RFID authentication protocols to privacy-preserving multi-party grouping proofs. J. Pers. Ubiquit. Comput. **16**(3), 323–335 (2012). Springer, Heidelberg
3. Burmester, M., de Medeiros, B., Motta, R.: Provably secure grouping-proofs for RFID tags. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 176–190. Springer, Heidelberg (2008)
4. Bolotnyy, L., Robins, G.: Generalized "yoking-proofs" for a groupe of RFID tags. In: Mobiquitous 2006, pp. 1–4. IEEE (2006)
5. Bagheri, N., Safkhani, M.: Securet disclosure attack on Kazahaya, a yoking-proof for low-cost RFID tags. Cryptology ePrint Archive, Report 2013/453
6. Chien, H.-Y., Liu, S.-B.: Tree-based RFID yoking proof. In: NSWCTC 2009, pp. 550–553. IEEE (2009)
7. Deng, R.H., Li, Y., Yung, M., Zhao, Y.: A new framework for RFID privacy. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 1–18. Springer, Heidelberg (2010)
8. Gilbert, H., Robshaw, M., Sibert, H.: An active attack against HB+ - a provably secure lightweight authentication protocol. IEEE Electron. Lett. **41**(21), 1169–1170 (2005). IEEE
9. Gilbert, H., Robshaw, M., Seurin, Y.: Good variants of $\mathrm{HB}^+$ are hard to find. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143, pp. 156–170. Springer, Heidelberg (2008)
10. Huang, H.-H., Ku, C.-Y.: A RFID grouping proof protocol for medication sefety of inpatient. J. Med. Syst. **33**(6), 467–474 (2009). Springer, Heidelberg
11. Hermans, J., Peeters, R.: Private yoking proofs: attacks, models and new provable constructions. In: Hoepman, J.-H., Verbauwhede, I. (eds.) RFIDSec 2012. LNCS, vol. 7739, pp. 96–108. Springer, Heidelberg (2013)
12. Hermans, J., Pashalidis, A., Vercauteren, F., Preneel, B.: A new RFID privacy model. In: Atluri, V., Diaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 568–587. Springer, Heidelberg (2011)
13. Juels, A.: "Yoking-proofs" for RFID tags. In: PerSec 2004, pp. 138–143. IEEE (2004)
14. Juels, A., Weis, S.A.: Authenticating pervasive devices with human protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005)
15. Juels, A., Weis, S.A.: Defining strong privacy for RFID. ACM TISSEC **13**(1), 7 (2009). ACM
16. Lin, C.-C., Lai, Y.-C., Tygar, J.D., Yang, C.-K., Chiang, C.-L.: Coexistence proof using chain of timestamps for multiple RFID tags. In: Chang, K.C.-C., Wang, W., Chen, L., Ellis, C.A., Hsu, C.-H., Tsoi, A.C., Wang, H. (eds.) APWeb/WAIM 2007 Ws. LNCS, vol. 4537, pp. 634–643. Springer, Heidelberg (2007)
17. Moriyama, D., Matsuo, S., Ohkubo, M.: Relations among notions of privacy for RFID authentication protocols. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012. LNCS, vol. 7459, pp. 661–678. Springer, Heidelberg (2012)
18. Ouafi, K., Overbeck, R., Vaudenay, S.: On the security of HB# against a man-in-the-middle attack. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 108–124. Springer, Heidelberg (2008)

19. Piramuthu, S.: On existence proofs for multiple RFID tags. In: PerSecU 2006, pp. 317–328. IEEE (2006)
20. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: Solving the simulutaneous scanning proglem anonymously: clumping proofs for RFID tags. In: SecPerU 2007, pp. 55–60. IEEE (2007)
21. Peris-Lopez, P., Orfila, A., Hernandez-Castro, J.C., Lubbe, J.C.A.: Flaws on RFID grouping-proofs. guidelines for future sound protocols. J. Netw. Comput. Appl. **34**(3), 833–845 (2011). Academic Press
22. Saito, J., Sakurai, K.: Grouping proof for RIFD tags. In: AINA 2005, vol. 2. pp. 621–624. IEEE (2005)