Hindawi
Security and Communication Networks
Volume 2017, Article ID 5084636, 10 pages
https://doi.org/10.1155/2017/5084636

Research Article

A Privacy Model for RFID Tag Ownership Transfer

Xingchun Yang,[1,2] Chunxiang Xu,[1] and Chaorong Li[3]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2]*Department of Computer Science and Technology, Sichuan Police College, Luzhou 646000, China*
[3]*Department of Computer Science and Information Engineering, Yibin University, Yibin 644000, China*

Correspondence should be addressed to Xingchun Yang; yangxc2004@163.com

Received 7 December 2016; Revised 13 February 2017; Accepted 13 February 2017; Published 5 March 2017

Academic Editor: Muhammad Khurram Khan

The ownership of RFID tag is often transferred from one owner to another in its life cycle. To address the privacy problem caused by tag ownership transfer, we propose a tag privacy model which captures the adversary's abilities to get secret information inside readers, to corrupt tags, to authenticate tags, and to observe tag ownership transfer processes. This model gives formal definitions for tag forward privacy and backward privacy and can be used to measure the privacy property of tag ownership transfer scheme. We also present a tag ownership transfer scheme, which is privacy-preserving under the proposed model and satisfies the other common security requirements, in addition to achieving better performance.

1. Introduction

RFID (Radio-Frequency Identification) technology is widespread in commercial industry such as supply chain management, inventory management, and access control for people and vehicles. A RFID application system mainly consists of tags, readers/interrogators, and back-end server. A passive tag is basically a device embedded with a small chip and a coiled antenna, in which the chip has limited computation capability and small memory to store its secret key and identifier, and the antenna communicates with its reader via radio-frequency signal. A reader is used to interrogate tags and send the data received from tags to a back-end server for product identification or inventory tracking. The back-end server stores tag's secret keys, identifiers, and the information of the items labeled by tags and executes product identification or inventory tracking.

However, the privacy issue (e.g., information leakage, location tracking, and profiling individuals) caused by RFID technology has raised grave concerns among the public. A recommendation on this issue was published by the Commission of the European Communities [1], which gave particular attention to the individual tracking and the access to personal data. Take the medicine supply chain for example, those tags attached to medicines are often transferred from one owner to another, however, the previous owner of a tag may infer the tag's track from its future interactions with a new owner, and as a result, the new owner's privacy may be infringed. Another serious scenario is that terrorists may exploit this technology to track their target who holds the RFID tags. Once those tags are distinguished at specific point (e.g., the checkpoint or toll station) by the terrorist's surreptitious devices, particular devices such as a bomb may be triggered.

The privacy of RFID tag means anonymity and untraceability [2], namely, an adversary cannot distinguish or track a tag from other tags at the protocol level. It is observed that many studies [2–16] on tag privacy have focused on the privacy problems caused by tag authentication or tag identification, but little attention was paid to the tag ownership transfer which may leak out tag's privacy.

Meanwhile, there are some other works [17–31] focusing on design and analysis for concrete tag ownership transfer schemes or protocols, but few of them use formal methods to analyze the privacy problem resulting from tag ownership transfer. As Munilla et al. [32] pointed out, strong privacy remains an open problem for lightweight RFID applications.

Actually, a malicious owner has access to the back-end server which stores all the information of readers and tags, and he has advantage to distinguish a tag after the tag ownership transfer or inferring the past activities of a tag when getting the tag's ownership. This attack belongs to insider attack, which is serious in practice, but IND-CCA2 encryption can be employed to prevent such kind of attack [33].

Concentrating on tag ownership transfer, we propose a privacy model which introduces strong adversaries, who have abilities to obtain the full information of readers, to authenticate tags, to observe the whole transfer process, and to corrupt tags. With this model, we briefly analyze the scheme [17] which is based on public key encryption on tags. We also present a tag ownership transfer scheme, which is forward and backward privacy-preserving under our model.

The rest of this paper is organized as follows. In Section 2, we review the relevant work on tag ownership transfer, and then we describe the proposed model in Section 3. In Section 4, a recent tag ownership transfer scheme is briefly analyzed, and the proposed ownership transfer scheme is described and analyzed in Section 5. Section 6 concludes the paper.

2. Related Work

Soundness, correctness, and privacy are the required properties for RFID system. Briefly, soundness which is also called security in [2, 9] means that a fake tag cannot be accepted by the system except with negligible probability; correctness means a legitimate tag is always accepted by the system with an overwhelming probability. Canard et al. [6] gave the definitions of soundness and correctness.

In terms of tag ownership transfer scenario, tag forward privacy means an owner of a tag T_i cannot distinguish T_i from others if T_i's ownership was transferred to another owner, and tag backward privacy means the current owner of T_i cannot link T_i to its previous interactions (e.g., the transcripts of authentication and ownership transfer process with its previous owner).

In Section 3, we will give the proposed privacy model, which defines tag forward privacy and backward privacy. Since our model concentrates on the privacy problem caused by tag ownership transfer, the properties of soundness and correctness will not be discussed further. However, their definitions can be combined compatibly with our model.

2.1. Tag Ownership. A tag is often attached to an item and authenticated to its back-end server, and it would be transferred from one owner to another in its lifetime. As for a tag ownership transfer, the current owner may launch authentication procedure to authenticate or identify the tag and then transfer the tag's secret key or identifier to a new owner's server. Upon getting these secret data, the new owner has ability to authenticate or identify the tag. In this sense, these secret data used for tag authentication/identification are called ownership. In order to prevent the previous owner from authenticating or identifying the tag, the new owner will launch update procedure to make the tag and the new server refresh these shared secret data.

2.2. Tag Ownership Transfer. Tag ownership transfer is more complicated than tag authentication to reader or their mutual authentication, because the current owner and the new owner are involved in the ownership transfer process. Moreover, tag ownership transfer is closely related to tag authentication to its reader.

Several tag ownership transfer schemes are derived from tag authentication protocols. Molnar et al. [22] proposed a scalable and delegatable pseudonym authentication protocol which enables tag ownership transfer; however, a trusted center is required. Lim and Kwon [21] proposed a robust authentication protocol enabling ownership transfer, whereas their scheme cannot achieve tag untraceability under the model in [34]. Song [25] suggested a tag ownership transfer scheme, which enables authorization recovery and protects the privacy of both current and previous owners, but this scheme is vulnerable to tag location tracking, tag forward traceability, and desynchronization attack [23]. Later, Song and Mitchell [26] proposed another RFID protocol and claimed it supports tag ownership transfer, tag delegation, and authorization recovery. Recently, Kardaş et al. [20] introduced an authentication protocol enabling tag ownership transfer with hash and XOR operation and claimed the protocol achieves tag untraceability against strong adversaries.

There are some ownership transfer schemes for the tags supporting public key encryption. Fu and Guo [27] designed a mutual authentication protocol supporting tag ownership transfer based on the SQUASH scheme [35], but the authors did not exhibit the tag ownership transfer process. Cheng et al. [17] presented an ownership transfer scheme, which employs elliptic curve cryptography (ECC) and will be briefly analyzed in Section 4.

There are some other ownership transfer schemes based on the public key encryption on readers. Elkhiyaoui et al. [18] designed a transfer scheme consisting of three subprotocols, but the scheme is vulnerable to the privacy track initiated by the tag's previous owner [36], because their definitions neglected the ability of the previous owner who ever controls the secret key of the target tag, and some revisions in [29] were presented to correct this flaw. Xin et al. [28] proposed a privacy-preserving ownership transfer scheme and claimed it guarantees privacy and other security properties; however, this scheme employed a powerful Trusted Third Party.

Some researches on tag ownership transfer focus on mobile RFID environment, and the readers interested in this may refer to [30, 37, 38] for further investigation.

3. The Proposed Privacy Model

Most schemes introduced in Section 2.2 overlooked the fact that the owner may be malicious, and it lacks formal definitions to analyze the privacy property of tag ownership transfer scheme.

van Deursen and Radomirović [33] introduced the insider attack which is serious to the RFID system, because adversaries have the knowledge of readers and tags. In this

section, we propose a privacy model for RFID tag ownership transfer. This model provides adversaries with abilities to get the reader's secret information, to constantly eavesdrop on the communications between reader and its tags, to corrupt tags, and to transfer tag's ownership to another owner. In the model, the goal of the adversary is to distinguish or infer the target tag from others.

3.1. Entities in the Proposed Model. For simplicity, we suppose the RFID system in the model consists of two owners denoted by A with a reader R_A and by B with a reader R_B. Because the reader and its back-end server are powerful devices and communicate via secure channel, we also suppose the back-end server is integrated with the reader. Moreover, we suppose the temporary information (e.g., the nonce generated by reader and tags) will be automatically erased when the authentication or ownership transfer process is completed. The notations used in the following sections are listed in Notations for readability.

3.2. Oracles Provided for Adversaries. In a realistic scenario, adversaries can exploit the following information to attack tag's privacy: (1) the secret information inside reader and tags, (2) the authentication information between reader and tags, (3) the results that adversaries launch authentication procedures with tags, and (4) the ownership transfer information between reader and tags.

We give an adversary $\mathscr{A}$ the following oracles to simulate his abilities to attack the privacy of a tag T_{id}.

(1) Authenticate$(R, T_{\text{id}}) \rightarrow (\tau, k_{\text{DB}}, t_{\text{DB}}, k_{\text{id}}, t_{\text{id}}, \text{result}, \ldots)$: This oracle is provided for $\mathscr{A}$ to make a reader R launch authentication session with T_{id}. If the adversary controls the secret information of R and T_{id}, it returns τ, the secret information such as key and identifier pairs $(k_{\text{DB}}, t_{\text{DB}})$, $(k_{\text{id}}, t_{\text{id}})$ as well as the authentication result. Otherwise, it only returns the authentication result and the process transcripts; namely, it returns $(\tau, -, -, -, -, \text{result}, \ldots)$. This oracle simulates the adversary's abilities to launch active attack and to get side channel information (e.g., the result whether or not a tag is accepted by its reader).

(2) Observe$(R, T_{\text{id}}) \rightarrow (\tau, \text{result})$: This oracle makes R launch authentication session with the tag T_{id} and returns the execution transcripts as well as the authentication result. The adversary can query this oracle to eavesdrop on the communication between the target tag and its reader.

(3) Corrupt$(T_{\text{id}}) \rightarrow (k_{\text{id}}, t_{\text{id}}, \ldots)$: This oracle is provided for the adversary to corrupt tags. It returns the secret key, the identifier, and the other information inside T_{id}.

(4) Transfer$(R_A, T_{\text{id}}, R_B) \rightarrow (t_{\text{vid}}, \tau)$: The adversary can query this oracle to get the information of the ownership transfer process. It transfers T_{id}'s ownership from a current owner who controls R_A to a new owner who controls R_B and returns a virtual identifier t_{vid} for the tag as well as the transcripts of the transfer process.

(5) Test$(T^j_{\text{id}_0}, T^j_{\text{id}_1}) \rightarrow (t^{j+1}_{\text{id}}, \tau)$: This oracle is provided for the adversary only once at any time. It accepts two tags and then selects a bit $b \in_R (0, 1)$ to transfer the ownership of $T^j_{\text{id}_b}$ to a new owner. This oracle returns the identifier of $T^{j+1}_{\text{id}_b}$ and the transcripts of the transfer process.

We denote the first four oracles by o_1, o_2, o_3, o_4 and the number of times that the adversary queries them by n^{o_1}, n^{o_2}, n^{o_3}, n^{o_4}, respectively, and denote the set (o_1, o_2, o_3, o_4) by O. We say that an adversary $\mathscr{A}$ is a (t, O)-adversary, if the number of times that $\mathscr{A}$ queries the above oracles is at most $\sum_{i=1}^{4} n^{o_i} \leq t$, where t is polynomial in ℓ.

As for a tag ownership transfer process, a new owner will get the tag's secret information (e.g., the key and identifier) from the current owner and then update some information inside the tag to prevent the current owner from successfully identifying or tracking the tag.

3.3. Definition of Forward Privacy. We denote a tag T_{id} at time point just before its ownership is transferred by T^j_{id} and denote it by T^{j+k}_{id} $(k = 1, 2, \ldots)$ after the transfer process is finished. The transfer scheme should guarantee the previous owner cannot infer the identity of T^{j+k}_{id} $(k = 1, 2, \ldots)$ from the identity of T^j_{id}. Without loss of generality, we suppose the current reader of the tag is R_A and the new reader is R_B.

Definition 1. Provided with the information of R_A any (t, O)-adversary $\mathscr{A}$ can query oracles in O at most t times, we denote the probability that $\mathscr{A}$ selects two corrupted tags $(T^j_{\text{id}_0}, T^j_{\text{id}_1}) \in \text{ID}_A$ for querying Test$(T^j_{\text{id}_0}, T^j_{\text{id}_1})$ and correctly guesses b by $\text{succ}^{\text{forward}}_A(\ell)$, with permission to query Corrupt$(T^{j+k}_{\text{id}_b})$ $(k = 1, 2, \ldots)$. The tag is forward privacy if $\text{adv}^{\text{forward}}_A(\ell) = |\text{succ}^{\text{forward}}_A(\ell) - 1/2|$ is negligible.

3.4. Definition of Backward Privacy. After the ownership of the tag T^j_{id} has been successfully transferred to a new owner, the transfer scheme should guarantee the new owner cannot link the information of T^{j+k}_{id} $(k = 1, 2, \ldots)$ to the previous activities of the tag. With the same way to define tag forward privacy, we define tag backward privacy as follows.

Definition 2. Provided with all the information of R_B any (t, O)-adversary $\mathscr{A}$ queries oracles in O at most t times, we denote the probability that $\mathscr{A}$ chooses uncorrupted tags $(T^j_{\text{id}_0}, T^j_{\text{id}_1}) \in \text{ID}_A$ for querying Test$(T^j_{\text{id}_0}, T^j_{\text{id}_1})$ and then correctly guesses b by $\text{succ}^{\text{backward}}_A(\ell)$ with being given the identifier of $T^j_{\text{id}_b}$. We also denote the advantage that $\mathscr{A}$ infers which

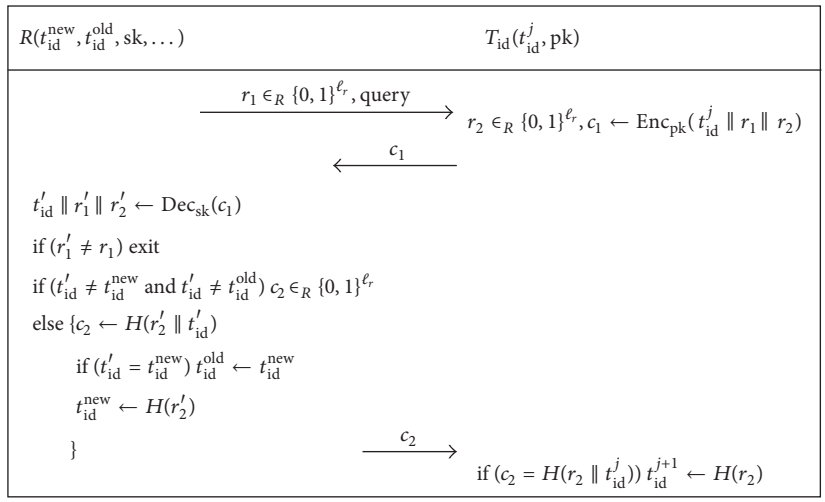

Figure 1: The proposed mutual authentication protocol.

of $T_{id_0}^{j}$ and $T_{id_1}^{j}$ is chosen by $adv_A^{backward}(\ell) = |succ_A^{backward}(\ell) - 1/2|$. The tag is backward privacy if $adv_A^{backward}(\ell)$ is negligible.

4. Brief Analysis for an ECC-Based Tag Ownership Transfer Scheme

Some tag ownership transfer schemes such as [18, 20, 26] are not based on public key encryption on tags. To achieve forward privacy or backward privacy for tags, tag owner is required to run extra tag authentication sessions in an environment where adversaries cannot eavesdrop on the authentication sessions. Such requirements do not satisfy our model in which adversaries can always eavesdrop on the interactions between the tag and its reader, in addition to corrupting the tag after the ownership transfer.

To guarantee security and privacy, a few tag authentication protocols [27, 39–41] are based on tags that support public key encryption, and to the best of our knowledge, the authors of [17] presented a complete ownership transfer scheme based on tags supporting ECC. We briefly analyze this scheme as follows and demonstrate that it is not forward privacy under our definitions.

This ownership transfer scheme consists of four subprotocols: tag key change protocol (P1), tag key update protocol (P2), ownership transfer protocol (P3), and controlled delegation protocol (P4); however, the authentication protocol between reader and tags is not given. Before the current owner (e.g., R_A) transfers the tag T_{id} to the new owner (e.g., R_B), R_A first launches P1 to refresh the information inside T_{id} and then runs P3 to transfer the ownership of T_{id} to R_B. Finally, R_B launches P2 to update the information inside T_{id}.

Under Definition 1, after querying the oracles in O at most t times, a (t, O)-adversary $\mathscr{A}$ chooses two corrupted tags $(T_{id_0}^{j}, T_{id_1}^{j}) \in ID_A$ for querying $Test(T_{id_0}^{j}, T_{id_1}^{j})$ and then guesses a bit $b \in (0, 1)$. However, since $\mathscr{A}$ can corrupt $T_{id_b}^{j+1}$ to get its secret key k_b, and k_b is not updated throughout the whole ownership transfer process, $\mathscr{A}$ can always correctly guess the value of b.

Under Definition 2, after querying $Test(T_{id_0}^{j}, T_{id_1}^{j})$, the adversary $\mathscr{A}$ will guess a bit b, provided that the secret key k_b is given. Yet it is unclear whether or not $\mathscr{A}$ can link k_b to the previous transcripts that $T_{id_b}^{j-1}$ authenticates to its reader, because the authentication protocol is not given in [17]. In other words, it is not ensured whether or not $\mathscr{A}$ can infer k_b from the previous authentication information.

5. The Proposed Tag Ownership Transfer Scheme

Motivated by the slightly higher performance tags like [42, 43] that support public key encryption, we propose a tag ownership transfer scheme in this section. This scheme consists of a mutual authentication protocol (AP) and an ownership transfer protocol (TP). We demonstrate it is both forward privacy and backward privacy under our model and give the security analysis in Appendix.

We suppose the database DB of the back-end server stores $(t_{id}^{new}, t_{id}^{old}, sk, \dots)$ for each tag T_{id} as well as the information of the products that are labeled by tags, and DB is integrated into the reader R. T_{id} stores its identifier and its reader's public key (t_{id}^{j}, pk). In the following sections, we first describe the mutual authentication protocol and then the tag ownership transfer protocol.

5.1. The Mutual Authentication Protocol. This protocol provides mutual authentication for the reader and tags; we give the details in Figure 1 and the interpretation as follows.

(1) R first sends a nonce r_1 and an access command query to T_{id}.

(2) Upon receiving r_1 and query, T_{id} selects another nonce r_2 and responds with $c_1 = Enc_{pk}(t_{id}^{j} \| r_1 \| r_2)$.

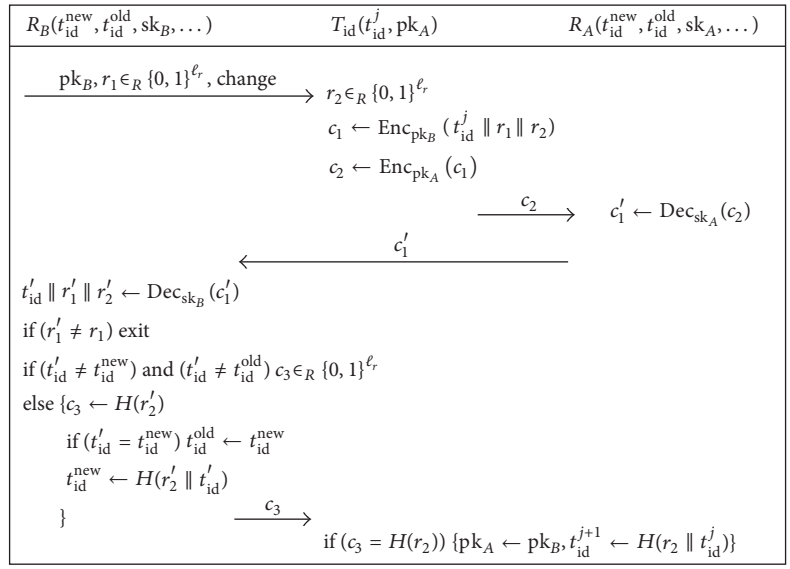

Figure 2: The proposed tag ownership transfer protocol.

(3) Upon receiving c_1, R decrypts it to get t_{id}', r_1', r_2'. If $(r_1' = r_1)$ does not hold, R interrupts this process.

If $(r_1' = r_1)$ holds, R then retrieves t_{id}' in its DB. If both $(t_{id}' \neq t_{id}^{new})$ and $(t_{id}' \neq t_{id}^{old})$ hold, R assigns another nonce to the variable c_2. Otherwise, R assigns the hash value $H(r_2' \| t_{id}')$ to c_2 and updates t_{id}^{old} with t_{id}^{new} if t_{id}' equals t_{id}^{new}. R also updates t_{id}^{new} with the value $H(r_2')$. Finally, R sends back c_2 to T_{id}.

(4) Upon receiving c_2, if $(c_2 = H(r_2 \| t_{id}^{j}))$ holds, T_{id} updates t_{id}^{j} with the value $H(r_2)$; namely, $t_{id}^{j+1} \leftarrow H(r_2)$.

5.2. The Tag Ownership Transfer Protocol. This protocol transfers T_{id}'s ownership from the current owner of R_A to the new owner of R_B. Before the transfer process, R_A and R_B should authenticate to each other and setup a secure channel, and then R_A sends t_{id}^{new}, t_{id}^{old} and the other information of T_{id} to R_B.

After finishing the ownership transfer process, T_{id}'s identifier is updated with a new value which is shared with R_B, and the public key stored in T_{id} is replaced with R_B's public key; Figure 2 shows the details.

(1) R_B first sends its public key pk_B, a nonce r_1, and the access command change to T_{id}.

(2) Once receiving pk_B, r_1 and change, T_{id} selects another nonce r_2 and assigns the value $Enc_{pk_B}(t_{id}^{j} \| r_1 \| r_2)$ to c_1 and then sends $c_2 = Enc_{pk_A}(c_1)$ to R_A.

(3) Upon receiving c_2, R_A decrypts it to get c_1' and then forwards c_1' to R_B.

(4) Upon receiving c_1', R_B decrypts it to get t_{id}', r_1', and r_2'. If $(r_1' \neq r_1)$ holds, R_B interrupts this process. If both $(t_{id}' \neq t_{id}^{new})$ and $(t_{id}' \neq t_{id}^{old})$ hold, R_B assigns a nonce to a variable c_3; otherwise, R_B assigns the value of $H(r_2')$ to c_3 and updates t_{id}^{old} with t_{id}^{new} if t_{id}' equals t_{id}^{new}. R_B also updates t_{id}^{new} with $H(r_2' \| t_{id}')$. Finally, R_B sends back c_3 to T_{id}

(5) If c_3 is equal to $H(r_2)$, T_{id} replaces pk_A with pk_B and updates t_{id}^{j} with $H(r_2 \| t_{id}^{j})$ to successfully finish the transfer process.

5.3. Forward Privacy of the Proposed Scheme. According to Definition 1 of the proposed model, an (t, O)-adversary $\mathcal{A}$ obtains all the information of R_A and queries oracles in O in the first stage. $\mathcal{A}$ selects two corrupted tags $(T_{id_0}^{j}, T_{id_1}^{j}) \in ID_A$ in the second stage to query $\text{Test}(T_{id_0}^{j}, T_{id_1}^{j})$ and guess a bit $b \in (0, 1)$ with permission to corrupt $T_{id_b}^{j+k}$ $(k = 1, 2, \ldots)$.

First, there is no link between the identifier $t_{id_b}^{j+1}$ and the knowledge that $\mathcal{A}$ obtains in the first stage, because the scheme employs the hash value of a nonce to update the tag's identifier in authentication and ownership transfer process. Hence, $\mathcal{A}$ cannot benefit from this stage to enhance his advantage to infer $T_{id_b}^{j+k}$ $(k = 1, 2, \ldots)$ from $T_{id_0}^{j}$ and $T_{id_1}^{j}$, except to guess b with probability of 1/2.

Second, $\text{Test}(T_{id_0}^{j}, T_{id_1}^{j})$ will return the transcripts $\tau(pk_B, r_1, \text{change}, c_2, c_1' = Enc_{pk_B}(t_{id_b}^{j} \| r_1 \| r_2), c_3 = H(r_2'))$ in the second stage; therefore $\mathcal{A}$ can take advantage of these transcripts to guess b in the following ways.

(1) Decrypting c_1' to get the nonce r_2 and compare the hash value $H(r_2 \| t_{id_0}^{j}), H(r_2 \| t_{id_1}^{j})$ with $t_{id_b}^{j+1} = H(r_2 \| t_{id_b}^{j})$, respectively, in order to determine the value of b. However, since $\mathcal{A}$ does not hold the secret key sk_B, the probability that he decrypts c_1' with a random

secret key is $1/\ell$. In other words, the advantage that $\mathscr{A}$ correctly guesses b is $1/\ell$ in this way.

(2) Inverting the one-way hash function $H(\cdot)$ to get the nonce r_2' from $c_3 = H(r_2')$ and then calculating $H(r_2 \parallel t^j_{\mathrm{id}_0})$ and $H(r_2 \parallel t^j_{\mathrm{id}_1})$ to compare the results with $t^{j+1}_{\mathrm{id}_b}$. However, as we all know, it is hard to invert one-way hash function so far, and the adversary can only guess the value of r_2'. Hence, the probability $\mathscr{A}$ correctly guesses b is $1/\ell_r$ in this way.

Third, after querying Test$(T^j_{\mathrm{id}_0}, T^j_{\mathrm{id}_1})$, $\mathscr{A}$ can corrupt $T^{j+1}_{\mathrm{id}_b}$ to obtain its identifier $t^{j+1}_{\mathrm{id}_b}$ which is the hash value of $H(r_2 \parallel t^j_{\mathrm{id}_b})$ and then invert this value (namely, $t^{j+1}_{\mathrm{id}_b}$) to get $t^j_{\mathrm{id}_b}$. However, it is difficult to invert the one-way hash function, and $\mathscr{A}$ can only guess a value as the input of the hash function. As a result, the probability that $\mathscr{A}$ correctly guesses b is $1/\ell_r$.

Finally, the adversary could keep on corrupting $T^{j+k}_{\mathrm{id}_b}$ ($k = 2, 3, \ldots$) in the future interactions between the tag and its reader. However, this does not help $\mathscr{A}$ link $T^j_{\mathrm{id}_b}$ to $T^{j+k}_{\mathrm{id}_b}$ ($k = 2, \ldots$), because the hash value of secret nonce is employed to update the tag's identifier in our scheme.

To sum it up, according to the proposed model the advantage that a (t, O)-adversary $\mathscr{A}$ attacks the forward privacy of the scheme is negligible.

5.4. Backward Privacy of the Proposed Scheme. Under Definition 2, a (t, O)-adversary $\mathscr{A}$ receives all the information of R_B and queries oracles in O in the first stage. $\mathscr{A}$ selects two uncorrupted tags $(T^j_{\mathrm{id}_0}, T^j_{\mathrm{id}_1}) \in \mathrm{ID}_A$ in the second stage for querying Test$(T^j_{\mathrm{id}_0}, T^j_{\mathrm{id}_1})$ and then guesses a bit $b \in (0, 1)$, provided the identifier $t^j_{\mathrm{id}_b}$ is given.

Except for guessing b with probability 1/2, $\mathscr{A}$ can guess b by the following ways.

(1) Inferring b from the relation between the interaction transcripts that R_A authenticates $T^{j-1}_{\mathrm{id}_0}/T^{j-1}_{\mathrm{id}_1}$ and the identifier $t^j_{\mathrm{id}_b} = H(r_2) = H(r_2')$:

$\mathscr{A}$ queries Authenticate$(R_A, T^{j-1}_{\mathrm{id}_0}/T^{j-1}_{\mathrm{id}_1})$ or Observe$(R_A, T^{j-1}_{\mathrm{id}_0}/T^{j-1}_{\mathrm{id}_1})$ to get the authentication transcripts $c_1 = \mathrm{Enc}_{\mathrm{pk}_A}(t^{j-1}_{\mathrm{id}_0} \parallel r_1 \parallel r_2)$, $c_2 = H(r_2' \parallel t'_{\mathrm{id}_0})$ (or $c_1 = \mathrm{Enc}_{\mathrm{pk}_A}(t^{j-1}_{\mathrm{id}_1} \parallel r_1 \parallel r_2)$, $c_2 = H(r_2' \parallel t'_{\mathrm{id}_1})$). However, $\mathscr{A}$ cannot gain r_2 from c_1 because c_1 is a ciphertext by the public key pk_A; he can only guess the secret key sk_A with probability $1/\ell$ to decrypt c_1. $\mathscr{A}$ also cannot gain r_2' from c_2 by inverting the one-way hash function $H(\cdot)$, except to guess r_2' with probability $1/\ell_r$. Hence, in this way the probability that $\mathscr{A}$ correctly guesses b is negligible.

(2) Inferring b from the test transcripts $(\mathrm{pk}_B, r_1, \mathrm{change}, c_2, c_1' = \mathrm{Enc}_{\mathrm{pk}_B}(t^j_{\mathrm{id}_b} \parallel r_1 \parallel r_2), c_3 = H(r_2'))$ as well as the identifier $t^{j+1}_{\mathrm{id}_b} = H(r_2 \parallel t^j_{\mathrm{id}_b})$: However, $t^j_{\mathrm{id}_b}$ is not related to $(\mathrm{pk}_B, r_1, \mathrm{change})$ and c_3, which is a hash value of a nonce. Moreover, c_2 is the result of the encryption for c_1', and c_1' is the ciphertext of r_1, r_2, and $t^j_{\mathrm{id}_b}$. Hence, this information cannot contribute to the adversary to infer which of $T^j_{\mathrm{id}_0}$ and $T^j_{\mathrm{id}_1}$ is selected. In other words, $\mathscr{A}$ can only infer $t^j_{\mathrm{id}_b}$ from $T^j_{\mathrm{id}_0}/T^j_{\mathrm{id}_1}$ with negligible probability in this way.

Furthermore, since $t^{j+1}_{\mathrm{id}_b}$ is the hash value of $t^j_{\mathrm{id}_b}$ and a nonce, the probability that $\mathscr{A}$ directly links $t^{j+1}_{\mathrm{id}_b}$ to $T^j_{\mathrm{id}_0}/T^j_{\mathrm{id}_1}$ is negligible.

In summary, the advantage of the adversary attacking the backward privacy of the scheme is negligible according to the proposed model.

Table 1: Comparisons of privacy property.

Schemes	Forward privacy	Backward privacy	Extra process
[18]	No	No	Yes
[20]	No	No	Yes
[26]	No	No	Yes
[17]	No	—	—
Ours	Yes	Yes	No

5.5. Privacy Comparison with Some Related Work. Recently, some tag ownership transfer schemes have been proposed; we compare the privacy property between those schemes and ours in Table 1 under the proposed model.

From the table, it can be seen that our scheme enjoys privacy property. Although the schemes proposed in [17, 18, 20, 26] are forward privacy and backward privacy under their model or their specific processes, which need extra steps to protect tag's privacy, those schemes cannot achieve forward privacy and backward privacy under our model, because in our model adversaries are permitted to corrupt tags after tag ownership transfer. Moreover, our model does not need extra processes to protect tag's privacy, and our scheme uses evolving hash value of secret nonce to update tag's identifier after each authentication or ownership transfer.

5.6. Performance Comparisons. Since RFID reader and server support enough complex cryptographic primitives, we only analyze the computation cost on tag side and the communication cost for messages that tag sends and directly receives from reader side, and we suppose our scheme is based on ECC.

For the sake of fair comparison, we suppose an elliptic curve is defined on finite field $F(2^{160})$, which needs 40 bytes and 20 bytes to store an elliptic curve point and an element in the field, respectively. We employ the hash scheme H-PRESENT-128 [44] with 128 bits' (16 bytes) output. We also suppose the bit length of a random number is 4 bytes and suppose the length of a tag's identifier is 12 bytes in compliance with the EPC (Electronic Product Code) Class-1 Generation-2 standard. The length of an access command sent by reader is negligible.

Comparisons of Communication Cost. Table 2 shows the comparison results of communication cost for some recently proposed tag ownership transfer schemes.

Table 2: Comparisons of communication cost.

Schemes	Reader side	Tag side	Total cost (bytes)
[18]	60	60	120
[20]	96	144	240
[26]	80	32	112
[17]	320	160	480
TP	60	60	120

Table 3: Comparisons of computation cost.

[18]	[20]	[26]	[17]	TP
3Ha	12Ha	6Ha	14Ecm	≈ 4Ecm + 2Ha

The results listed in the table show our scheme achieves better performance on communication cost, and we explain the results as follows. The tag ownership transfer process in [18] uses ElGamal encryption scheme to encrypt tag's identification information, so we suppose the length of a prime number in the scheme is 20 bytes. The scheme in [20] runs the authentication protocol three times to finish a tag's ownership transfer, and the protocol (denoted by P2) proposed in [26] will be executed twice to complete the ownership transfer for a tag, while our scheme just runs the ownership transfer protocol only once. To finish a tag's ownership transfer, a tag in the scheme [17] receives 8 elliptic curve points and sends 4 elliptic curve points; hence it sustains the heaviest communication cost.

Comparisons of Computation Cost. We denote the running time of a scalar multiplication operation over an elliptic curve by Ecm and a hash function operation by Ha; Table 3 shows the computation cost on tag side for some recent ownership transfer schemes and ours.

Because our ownership transfer protocol TP aims at those tags supporting ECC, thus the computation cost on tag side is higher than those tags not supporting ECC [18, 20, 26]; however, the computation cost of the TP is superior to the schemes in [17], which is also based on ECC on tags.

6. Conclusions

The privacy leakage caused by RFID tags is an important issue and has drawn wide attention. Some studies focused on the privacy problem caused by authentications between reader and tags, and a few researches paid attention to the privacy problem caused by tag ownership transfer. Yet few of them take the malicious owner into account or use formal methods to analyze the privacy leakage caused by tag ownership transfer.

In this paper, we propose a privacy model, which concentrates on the privacy problem caused by RFID tag ownership transfer. This model can be used to measure the privacy property of tag ownership transfer scheme, yet it cannot be directly applied to the authentications between reader and tags.

We also designed a tag ownership transfer scheme for the tags supporting public key encryption. According to the proposed model, we demonstrate our scheme enjoys both forward privacy and backward privacy. We also give the security analysis in Appendix. Upon comprehensive consideration on privacy protection, communication, and computation cost, our scheme is superior to those compared ones, and the implementation of this scheme would be our next work.

Appendix

Security of the Proposed Tag Ownership Transfer Scheme

We briefly analyze the security properties of the proposed scheme as follows.

Tag Impersonation Resistance. Note that almost all the RFID authentication protocols (including our scheme) keep tag's identifiers as secrets in order to prevent malicious parties from tracking tags.

For the AP of our scheme, upon intercepting the first-round message of a reader querying a tag T_{id}, an adversary $\mathscr{A}$ should respond to the reader with the second-round message. However, $\mathscr{A}$ cannot correctly compute the second-round message $c_1 = \mathrm{Enc}_{\mathrm{pk}}(t_{\mathrm{id}}^j \parallel r_1 \parallel r_2)$ without knowing the identifier t_{id}^j, unless he guesses an identifier, while the probability that $\mathscr{A}$ correctly guess the value of t_{id}^j is $1/\ell_t$, which is negligible. Moreover, after each successful authentication process, a tag's identifier will be updated with a hash value of a nonce concatenating the tag's previous identifier.

We can use the same way to analyze tag impersonation resistance of the TP. Without knowing tag's identity, the probability that adversaries correctly respond with the second-round message $c_2 = \mathrm{Enc}_{\mathrm{pk}_A}(\mathrm{Enc}_{\mathrm{pk}_B}(t_{\mathrm{id}}^j \parallel r_1 \parallel r_2))$ is $1/\ell_t$, which is negligible.

Reader Impersonation Resistance. According to the specification of AP, a reader R will first send a nonce r_1 along with a command to a tag T_{id} to launch a new session. Once receiving the first-round message, the tag generates another secret nonce r_2 and responds to R with ciphertext $\mathrm{Enc}_{\mathrm{pk}}(t_{\mathrm{id}} \parallel r_1 \parallel r_2)$. It is obvious that an adversary cannot correctly decrypt this ciphertext without knowing the reader's secret key sk in order to get r_2. Hence, he cannot correctly respond to the tag with the last round message, which is a hash value directly related to r_2. In other words, if an adversary tries to impersonate a reader, the verification for the last round message by tag will be failed.

The similar analysis can be applied to the reader impersonation resistance of our TP. To sum up, without knowing reader's secret key, adversaries cannot correctly compute the last round message to pass the verification of the tag, except with negligible probability.

Replay Attack Resistance. The authentication sessions of the AP in our scheme are initiated by RFID reader, and our AP employs a secret nonce r_2 and an evolved identifier t_{id} to

compute the second-round message $c_1 = \mathrm{Enc}_{\mathrm{pk}}(t_{\mathrm{id}} \parallel r_1 \parallel r_2)$ and the third-round message $c_2 = H(r_2 \parallel t_{\mathrm{id}})$. For readability of analysis, we denote the nonce and tag's identifier in the j^{th} session and $(j+1)^{\mathrm{th}}$ session by r_2^j, t_{id}^j and r_2^{j+1}, t_{id}^{j+1}, respectively, and denote the exchanged message in the j^{th} session and $(j+1)^{\mathrm{th}}$ session by c_1^j, c_2^j and c_1^{j+1}, c_2^{j+1}, respectively. We suppose the current session is $(j+1)^{\mathrm{th}}$, and an adversary $\mathscr{A}$ has obtained the old message c_1^j and c_2^j.

Firstly, $\mathscr{A}$ can replay c_1^j to a reader R. Upon receiving c_1^j, R decrypts it to get the nonce r_1^j; however, the verification for the value of r_1^j by R will be failed, because R initiates the current session with r_1^{j+1} other than r_1^j.

Secondly, $\mathscr{A}$ can replay c_2^j to the tag T. Once receiving c_2^j, T verifies whether it is equal to $H(r_2^{j+1} \parallel t_{\mathrm{id}}^{j+1})$; it is obvious that the verification will be failed because c_2^j is the hash value of $r_2^j \parallel t_{\mathrm{id}}^j$ other than the hash value of $r_2^{j+1} \parallel t_{\mathrm{id}}^{j+1}$.

For the TP of our scheme, adversaries also cannot replay old messages in order to pass the verification of the reader/tag. On one hand, if he replays an old message $c_2 = \mathrm{Enc}_{\mathrm{pk}_A}(\mathrm{Enc}_{\mathrm{pk}_B}(t_{\mathrm{id}}^j \parallel r_1 \parallel r_2))$ to a reader R_A, after decrypting c_2, R_A gets $c_1 = \mathrm{Enc}_{\mathrm{pk}_B}(t_{\mathrm{id}}^j \parallel r_1 \parallel r_2)$ and forward c_1 to a reader R_B. However, once R_B decrypts c_1 to get the value r_1 (which is not equal to the value that R_B has sent to the tag in the current session), the verification for r_1 by R_B will be failed. On the other hand, if an adversary replays an old message $c_3 = H(r_2)$ to a tag, the verification for c_3 will be failed because the current nonce that the tag generates is not r_2.

Desynchronization Attack Resistance. In our scheme, the back-end database stores two identifiers for each tag. One is $t_{\mathrm{id}}^{\mathrm{old}}$, which is the latest synchronization identifier between the reader and the tag. The other is $t_{\mathrm{id}}^{\mathrm{new}}$, which is computed by the reader in the latest authentication process.

In a new session, if an adversary blocks the third-round message c_2 of the AP (or the fourth-round message c_3 of the TP) to desynchronize a reader with its tag, the reader can always recover synchronization with its tag in the next session using the old identifier $t_{\mathrm{id}}^{\mathrm{old}}$.

Man-in-the-Middle Attack Resistance. For the AP, the second-round message c_1 sent by a tag T_{id} is encrypted with its identifier, the nonces r_1 and r_2. If an adversary intercepts this message and replaces it with another one to respond to a reader, the process that the reader identifies the tag will be failed with overwhelming probability. In other words, without knowing the secret identifier of a tag, an adversary cannot successfully launch the Man-in-the-Middle attack. Moreover, the third-round message c_2 that a reader sends to a tag is a ciphertext of a nonce and the tag's identifier; without knowing the tag's identifier, the adversary cannot generate a valid message to pass the verification of the tag.

For the TP of our scheme, the second-round message c_2 sent by a tag is also a ciphertext related to the secret identifier of the tag. Without knowing the tag's identifier, adversaries cannot compute a valid message to pass the authentication of the reader.

In summary, both AP and TP resist to the Man-in-the-Middle attack because secret identifier is employed to generate the exchanged messages.

Notations

R_A, R_B, R:	The reader controlled by the owner A, by owner B and in a general sense, respectively.
$T_{\mathrm{id}}, T_{\mathrm{id}}^j$:	A tag with identifier id and at the time point j.
$\mathrm{sk}_A, \mathrm{pk}_A$:	The secret key and public key of R_A.
$\mathrm{sk}_B, \mathrm{pk}_B$:	The secret key and public key of R_B.
sk, pk:	The secret key and public key of R.
$k_{\mathrm{id}}, t_{\mathrm{id}}, t_{\mathrm{id}}^j$:	T_{id}'s secret key, identifier, and the identifier at time j respectively, which are stored in T_{id}.
$\mathrm{ID}_A, \mathrm{ID}_B$:	The set that consists of tags authenticated by R_A and R_B, respectively.
$\mathrm{DB}_A, \mathrm{DB}_B, \mathrm{DB}$:	The database integrated in R_A, R_B and R respectively.
$k_{\mathrm{DB}}, t_{\mathrm{DB}}$:	Secret key and identifier of the tag T_{id}, which are stored in the database DB.
$t_{\mathrm{id}}^{\mathrm{new}}, t_{\mathrm{id}}^{\mathrm{old}}$:	The current identifier and the previous identifier of T_{id} respectively, which are stored in DB.
τ:	Interactive information like the transcripts of authentication process or ownership transfer process between reader and tags.
$-$:	The unknown information.
result:	The result that a reader authenticates a tag, and 1 indicates the tag is accepted by the reader, or otherwise 0.
$\parallel$:	String concatenation.
$\leftarrow, \rightarrow$:	Assigning the right value to the left variable and returning the left value, respectively.
$=, \in_R$:	The equal relationship and the operation that randomly selects an element from a finite set.
ℓ:	The security parameter which is the length of a secret key.
ℓ_t, ℓ_r:	The length of a tag's identifier and the length of a nonce, respectively.
$H(m)$:	One-way hash function with input message m.
$\mathrm{Enc}_{\mathrm{sk}}(m)$:	An encryption function with input message m and secret key sk.
$\mathrm{Dec}_{\mathrm{pk}}(c)$:	A decryption function for a ciphertext c with public key pk.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This paper is partially supported by the China Postdoctoral Science Foundation (no. 2016M602675) and by the Nature Science Foundation of Sichuan Province Education Department (no. 13ZB0127).

References

[1] COMMUNITIES TCOTE, "Commission recommendation of 12 may 2009 on the implementation of privacy and data protection principles in applications supported by radio-frequency identification," May 2009, https://ec.europa.eu/digital-single-market/en/news/commission-recommendation-12-may-2009-implementation-privacy-and-data-protection-principles/.

[2] S. Vaudenay, "On privacy models for RFID," in *Advances in cryptology—ASIACRYPT 2007*, vol. 4833 of *Lecture Notes in Computer Science*, pp. 68–87, Springer, Berlin, Germany, 2007.

[3] F. Armknecht, A. R. Sadeghi, A. Scafuro, I. Visconti, and C. Wachsmann, "Impossibility results for RFID privacy notions," in *Transactions on Computational Science XI*, pp. 39–63, Springer, 2010.

[4] G. Avoine, "Adversarial model for radio frequency identification," *IACR Cryptology ePrint Archive*, vol. 2005, article 49, 2005.

[5] G. Avoine, "Radio frequency identification: adversary model and attacks on existing protocols," Tech. Rep., 2005.

[6] S. Canard, I. Coisel, J. Etrog, and M. Girault, "Privacypreserving rfid systems: model and constructions," *IACR Cryptology ePrint Archive*, vol. 2010, article 405, 2010.

[7] R. H. Deng, Y. Li, M. Yung, and Y. Zhao, "A new framework for RFID privacy," in *Computer Security—ESORICS 2010*, pp. 1–18, Springer, 2010.

[8] J. Ha, S. Moon, J. Zhou, and J. Ha, "A new formal proof model for RFID location privacy," in *Proceedings of the 13th European Symposium on Research in Computer Security (ESORICS '08)*, pp. 267–281, Springer, Torremolinos, Spain, 2008.

[9] J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel, "A new RFID privacy model," in *Proceedings of the 16th European Symposium on Research in Computer Security (ESORICS '11)*, pp. 568–587, Springer, Leuven, Belgium, 2011.

[10] A. Juels and S. A. Weis, "Defining strong privacy for RFID," *ACM Transactions on Information and System Security*, vol. 13, no. 1, article 7, 2009.

[11] M. Langheinrich, "A survey of RFID privacy approaches," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 413–421, 2009.

[12] C. Y. Ng, W. Susilo, Y. Mu, and R. Safavi-Naini, "New privacy results on synchronized RFID authentication protocols against tag tracing," in *Computer Security—ESORICS 2009*, pp. 321–336, Springer, 2009.

[13] T. van Deursen, S. Mauw, and S. Radomirović, "Untraceability of RFID protocols," in *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks*, vol. 5019 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, Berlin, Germany, 2008.

[14] T. Van Deursen and S. Radomirović, "On a new formal proof model for RFID location privacy," *Information Processing Letters*, vol. 110, no. 2, pp. 57–61, 2009.

[15] K. Ouafi and S. Vaudenay, "Strong privacy for RFID systems from plaintext-aware encryption," in *Proceedings of the 11th International Conference on Cryptology and Network Security (CANS '12)*, pp. 247–262, Darmstadt, Germany, December 2012.

[16] M. Alizadeh, M. Zamani, A. R. Shahemabadi, J. Shayan, and A. Azarnik, "A survey on attacks in RFID networks," *Open International Journal of Informatics (OIJI)*, vol. 1, no. 1, pp. 15–24, 2012.

[17] S. Cheng, V. Varadharajan, Y. Mu, and W. Susilo, "A secure elliptic curve based RFID ownership transfer scheme with controlled delegation," in *RFIDSec Asia*, vol. 11 of *Cryptology and Information Security Series*, pp. 31–43, IOS Press, 2013.

[18] K. Elkhiyaoui, E. O. Blass, and R. Molva, "Rotiv: Rfid ownership transfer with issuer verification," in *RFID. Security and Privacy*, pp. 163–182, Springer, Berlin, Germany, 2012.

[19] S. Fouladgar and H. Afifi, "An efficient delegation and transfer of ownership protocol for RFID tags," in *Proceedings of the 1st International EURASIP Workshop on RFID Technology*, vol. 66, pp. 68–93, Vienna, Austria, 2007.

[20] S. Kardaş, S. Çelik, A. Arslan, and A. Levi, "An efficient and private RFID authentication protocol supporting ownership transfer," in *Lightweight Cryptography for Security and Privacy*, vol. 8162 of *Lecture Notes in Computer Science*, pp. 130–141, Springer, Berlin, Germany, 2013.

[21] C. H. Lim and T. Kwon, "Strong and robust RFID authentication enabling perfect ownership transfer," in *Information and Communications Security*, pp. 1–20, Springer, 2006.

[22] D. Molnar, A. Soppera, and D. Wagner, "A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags," in *Selected Areas in Cryptography*, vol. 3897 of *Lecture Notes in Comput. Sci.*, pp. 276–290, Springer, Berlin, Germany, 2006.

[23] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. E. Tapiador, T. Li, and Y. Li, "Vulnerability analysis of RFID protocols for tag ownership transfer," *Computer Networks*, vol. 54, no. 9, pp. 1502–1508, 2010.

[24] Y. Seo, T. Asano, H. Lee, and K. Kim, "A lightweight protocol enabling ownership transfer and granular data access of RFID tags," in *Proceedings of the Symposium on Cryptography and Information Security*, Sasebo, Japan, January 2007.

[25] B. Song, "RFID tag ownership transfer," in *Proceedings of the Workshop on RFID Security*, Budapest, Hungary, July 2008.

[26] B. Song and C. J. Mitchell, "Scalable RFID security protocols supporting tag ownership transfer," *Computer Communications*, vol. 34, no. 4, pp. 556–566, 2011.

[27] X. Fu and Y. Guo, "A lightweight RFID mutual authentication protocol with ownership transfer," in *Advances in Wireless Sensor Networks*, pp. 68–74, Springer, 2013.

[28] W. Xin, Z. Guan, T. Yang, H. Sun, and Z. Chen, "An efficient privacy-preserving RFID ownership transfer protocol," in *Web Technologies and Applications*, pp. 538–549, Springer, 2013.

[29] Y. Xing-Chun, X. Chun-Xiang, M. Jian-Ping, and L. Jian-Ping, "An improved RFID tag ownership transfer scheme," in *Proceedings of the IEEE 10th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP '13)*, pp. 356–361, IEEE, Chengdu, China, December 2013.

[30] M. H. Yang, "Across-authority lightweight ownership transfer protocol," *Electronic Commerce Research and Applications*, vol. 10, no. 4, pp. 375–383, 2011.

[31] M. Alizadeh, W. H. Hassan, M. Zamani, S. Karamizadeh, and E. Ghazizadeh, "Implementation and evaluation of lightweight

encryption algorithms suitable for RFID," *Journal of Next Generation Information Technology*, vol. 4, no. 1, pp. 65–77, 2013.

[32] J. Munilla, M. Burmester, and A. Peinado, "Attacks on ownership transfer scheme for multi-tag multi-owner passive RFID environments," *Computer Communications*, vol. 88, pp. 84–88, 2016.

[33] T. van Deursen and S. Radomirović, "Insider attacks and privacy of RFID protocols," in *Public Key Infrastructures, Services and Applications*, vol. 7163 of *Lecture Notes in Computer Science*, pp. 91–105, Springer, Berlin, Germany, 2012.

[34] K. Ouafi and R. C. W. Phan, "Traceable privacy of recent provably-secure rfid protocols," in *Applied Cryptography and Network Security*, pp. 479–489, Springer, Berlin, Germany, 2008.

[35] A. Shamir, "SQUASH—a new MAC with provable security properties for highly constrained devices such as RFID tags," in *Fast Software Encryption*, pp. 144–157, Springer, 2008.

[36] M. R. S. Abyaneh, "On the privacy of two tag ownership transfer protocols for RFIDs," in *Proceedings of the International Conference for Internet Technology and Secured Transactions (ICITST '11)*, pp. 683–688, IEEE, Abu Dhabi, United Arab Emirates, December 2011.

[37] J. N. Luo and M. H. Yang, "Mobile RFID mutual authentication and ownership transfer," in *Proceedings of the 6th International Conference on Systems (ICONS '11)*, pp. 88–94, January 2011.

[38] M. H. Yang, "Controlled delegation protocol in mobile RFID networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, Article ID 170150, 2010.

[39] Z. Zhang and Q. Qi, "An efficient RFID authentication protocol to enhance patient medication safety using elliptic curve cryptography," *Journal of Medical Systems*, vol. 38, article 47, 2014.

[40] Y. Jin, H. Sun, W. Xin, S. Luo, and Z. Chen, "Lightweight RFID mutual authentication protocol against feasible problems," in *Information and Communications Security*, pp. 69–77, Springer, 2011.

[41] M. Burmester, B. De Medeiros, and R. Motta, "Anonymous RFID authentication supporting constant-cost key-lookup against active adversaries," *International Journal of Applied Cryptography*, vol. 1, no. 2, pp. 79–90, 2008.

[42] Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede, "Elliptic-curve-based security processor for RFID," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 57, no. 11, pp. 1514–1527, 2008.

[43] NXP Semiconductors, Mifare smartcard ic's, https://www.mifare.net/en/products/chip-card-ics/mifare-classic/.

[44] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, and Y. Seurin, "Hash functions and RFID tags: mind the gap," in *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 283–299, Springer, Washington, DC, USA, August 2008.