

Article

A Secure Partial RFID Ownership Transfer Protocol with Multi-Owners

Jia-Ning Luo ^{1,*}  and Ming-Hour Yang ^{2,*}¹ Department of Information and Telecommunications Engineering, Ming-Chuan University, Taoyuan 33350, Taiwan² Department of Information and Computer Engineering, Chung Yuan Christian University, Taoyuan 32023, Taiwan

* Correspondence: deer@mail.mcu.edu.tw (J.-N.L.); mhyang@cycu.edu.tw (M.-H.Y.)

Received: 1 November 2019; Accepted: 16 December 2019; Published: 19 December 2019



Abstract: Mobile radio frequency identification (RFID) has been extensively applied in a wide range of fields. In supply chain management, RFID is used to more efficiently manage the ownership transfer of cargo. The transfer of a group of tags belonging to multiple owners is often required at the front end of a supply chain. This study, therefore, proposes a secure, high-performance threshold multi-owner partial tag ownership transfer protocol that supports a mobile RFID environment and features the capabilities and security required for supporting existing ownership transfer environments (e.g., application for different authorities, designation of the transfer target, and ownership transfer of a group of tags). Moreover, the proposed protocol can resist against most of the known attacks on RFID.

Keywords: mobile radio frequency identification (RFID); ownership transfer

1. Introduction

Mobile radio frequency identification (RFID) is a combination of a wireless network, mobile telecommunication technology, and RFID systems [1,2]. Mobile RFID is characterized by simple computing power, storage capacity, and the simultaneous reading and writing of multi-tag information. These features facilitate product identification and follow-up management. Mobile RFID is widely applied in supply chain management, access control, bill payment, smart home development, military supply control, and health care medication administration. Because RFID can effectively manage the flow and processing of goods, large-scale retail chain stores such as Walmart have saved roughly \$1.4 billion USD in cost by using RFID technologies [1].

RFID has become an integral part of supply chain management in recent years and has been continually advancing and becoming more affordable. Therefore, various components of a supply chain, including raw material supplier, product manufacturer, wholesaler, retailer, and end consumer, can employ RFID for follow-up management. Manufacturers use RFID tags to identify goods information and conduct inventory. Retailers use RFID tags to keep track of and manage product information and provide consumers with a convenient shopping platform and various services. Consumers use RFID tags to obtain product and post-sale information. To facilitate the management of supply chain automation and effectively engage in product ownership transfer [3–9], products labeled with RFID tags undergo multiple ownership transfers throughout their life cycle from their introduction to the decline stage.

For secure transfer of product ownership, designation must be ensured while also avoiding the windowing problem (i.e., new and old owners simultaneously owning a tag) and providing forward and backward secrecy. Forward secrecy means that the new owner cannot identify and decrypt

messages that were transmitted between the tag and its old owner. Backward secrecy means that the old owner cannot receive and decrypt the messages transmitted between the tag and its new owner. In the process of tag ownership transfer, since a tag has limited computing power and an RFID system employs wireless transmission, hackers can access messages sent by the tag or reader. The RFID system may also suffer from security threats, such as message modification, replay attack, man-in-the-middle attack, tracking attack, denial-of-service (DoS) attack, and a counterfeit tag reader attack.

When products are at the end of a supply chain, each product owner usually owns only a small number of products. Therefore, owners perform ownership transfer only once for a small number of products. Osaka et al. [6] proposed an ownership transfer method for a single tag in such an application environment in which few tags are being transferred. However, their method is associated with security flaws. For example, update-key messages are vulnerable to modification attacks, which causes an asynchronous service, compromised forward secrecy, and the windowing problem during the transfer period [9,10]. Jäppinen et al. [11] verified the integrity of update-key messages to reduce the likelihood of asynchronous communication between tags and end servers, but the methods for doing so still engender vulnerability to attacks, which results in persistent asynchronous problems [12]. Hence, Yoon et al. [13], Chen et al. [14], Yang et al. [15], and Dimitriou et al. [16] proposed new ownership transfer methods to address the asynchronous service problem and ensure the forward secrecy of ownership transfer. However, the protocol presented by Dimitriou et al. [16] is vulnerable to counterfeiting and replay attacks [17]. The methods developed by Yoon et al. [13] and Chen et al. [14] are also associated with security concerns such as lack of support for backward secrecy, the inability to ensure location privacy, and windowing problems [9,18]. Yang et al. [19] proposed layered object transport protocol (LOTP), which is applicable for environments employing mobile RFID. LOTP [19] involves the transfer of tag ownership through a trusted third party (TTP) to overcome attacks that occur during ownership transfer through mobile RFID. However, LOTP can only transfer one tagged object at a time and cannot efficiently transfer a large number of tagged objects.

Therefore, when products are at the front end of a supply chain, manufacturers or wholesalers generally perform a single transfer of ownership for an extremely large number of products. However, a protocol causes problems due to inefficiency if it can only transfer the ownership of a single tag. Zuo [12] and He et al. [20] used group keys to simultaneously authenticate and transfer the ownership of all tags in a group. However, Zuo's protocol resulted in denial of service (DoS) attack when the updated key was subject to a desynchronization attack. Subsequently, Jannati et al. [21] proposed a solution to the DoS problem caused by a modification to update-key messages. However, all of these group transfer protocols have a limitation of only being able to perform a single transfer for all tags of an owner and not being able to perform a partial transfer of only some tags in a group. In other words, the flexibility in object ownership transfer is limited, which renders relevant methods impractical. Therefore, Molnar et al. [22] proposed using a split back-end server and a tree of secrets shared between a large number of tags to achieve partial ownership transfer. The number of nodes in the tree represents the number of times that a reader is authorized to read a tag after a binary tree has been transferred from the reader to the back-end server. The new owner can achieve partial ownership transfer by obtaining the tree of secrets of a tag through the back-end server. Tsai et al. [23] proposed an ownership transfer method with grouping the proof protocol that allows for grouping proof and partial ownership transfer of tag groups while ensuring the integrity of the tagged cargo. Yang et al. proposed a tag group ownership transfer protocol with a trust third party (TTP) [24] and without a TTP [25]. This protocol generates a key for a tree of partial group communication by employing the group communication key shared between the tags and server to achieve partial ownership transfer of tag groups. In addition, this protocol can resist most known attacks and protect and secure the privacy of owners.

Ownership transfer is frequently required for a large number of products, particularly when the raw materials or goods of upstream industries are distributed along the supply chain. These raw materials and goods generally belong to different owners and are simultaneously loaded onto

the same cargo ship or cargo truck. However, existing methods for secure tag group ownership transfer are limited to only the transfer of objects of a single owner and are not applicable to ownership transfer for multi-object owners. Hence, Kapoor et al. [26] proposed a multi-owner ownership transfer method. However, their method is vulnerable to DoS attacks because it places the tag and server key in a desynchronized state when the key is updated. Moreover, their protocol can only transfer the ownership of a single tag of multiple owners. The transfer efficiency is reduced when transferring the ownership of multiple tags because each tag must independently perform all steps to authenticate the updated key, which increases the information and calculation load. To address this problem, Sundaresan et al. [27] proposed a multi-owner/multi-tag ownership transfer method that uses a group secret value shared between the owners and a group of tags to generate acknowledgments for every tag that must be transferred and send the acknowledgments to all tags in the group. Because each tag group that is designated for transfer generates a message based on its tag identification number (ID), each tag must examine every message received for a tag ID to acknowledge that its tag ID is contained in the transfer of this tag group to simultaneously partially transfer the ownership of tag groups of the owners. However, because the owners in a group use a shared secret to protect the tag message, the method cannot protect the data privacy of the owners in that group. Subsequently, Sundaresan et al. proposed another approach for protecting group communication privacy by applying different group secret values to each owner and tag group [28]. However, in both methods, the process of ownership transfer requires each tag in a group to compute the message for each tag that needs to be transferred in order to acknowledge that its tag needs to be transferred. For example, if a group containing 2000 tags needs to transfer 1000 tags to a new owner, then these 2000 tags must acknowledge that the 1000 messages contain its tag ID. Therefore, each tag requires a large amount of information, a high calculation load, and long transfer time. In addition, the two previously mentioned methods of ownership transfer for multi-owner and partial tag group environments are vulnerable to attacks (e.g., replay, tracking, or DoS attacks) and do not achieve forward secrecy [29]. Table 1 summarized the categories of RFID ownership transfer protocols.

Table 1. Categories of RFID ownership transfer protocols.

Category	Protocols
Single owner/single tag	Osaka et al. [6], Fouladgar et al. [8], Taqieddin et al. [9], Kapoor et al. [10], Jäppinen et al. [11], Zuo [12], Yoon et al. [13], Chen et al. [14], Yang et al. [15], Dimitriou et al. [16], Kapoor et al. [18], and Yang [19].
Single owner/multiple tags	He et al. [20], Jannati et al. [21], Molnar et al. [22], and Tsai et al. [23]
Single owner/partial tags	Yang et al. [24,25].
Multiple owners/multiple tags	Kapoor et al. [26], Sundaresan et al. [27,28], and Munilla et al. [29].

This study proposes a secure, high-performance multi-owner partial ownership transfer protocol to overcome the problems concerning the performance of existing multi-owner tag ownership transfer methods and address the security threats and privacy concerns that may arise in the process of ownership transfer. In this proposed multi-owner ownership transfer protocol, the old owners and new owners of a tag group may differ regarding their jurisdiction. In our protocol, the permissions of several owners are required to transfer a tag or multiple tags from a group of owners to the others. A single user cannot transfer his/her ownership of the tags to others. Our protocol is useful in the supply chain management. The factory assigns ownership of tags to a group of employees. However, it is not necessary to obtain the consent of each employee when transferring ownership. When one of the old owners initiates ownership transfer, a threshold scheme is used to ensure that (1) the consent of a certain number of old owners is obtained before the owner can partially transfer the ownership of a tag group to new owners, and (2) the proposed method can resist most of the known attacks and offer most of the security and privacy protection properties for ownership transfer. This study makes the following contributions to the literature. The multi-owner multi-tag partial ownership transfer protocol (1) is applicable in a mobile RFID environment, (2) can transfer the ownership of one,

some, or all tags, (3) provides two-way authentication between a tag, reader, and a back-end server, (4) ensures that tag ownership is only transferred to the designated owners, (5) is secure and immune to replay attacks, eavesdropping, message modification attacks, and tracking attacks (i.e., protects owner privacy) and provides forward and backward secrecy. Lastly, it (6) features high performance that is not related to the reader participating in the transfer or the number of tags and does not increase information and calculation load considerably when the number of owners and tags increases.

This paper is organized as follows. Section 2 introduces the environmental assumptions of the proposed ownership transfer protocol and the relationships among the tag, reader, and the back-end server. Section 3 provides a detailed description of the proposed protocol. Section 4 compares and analyzes the security of the proposed method and other RFID ownership transfer methods. Section 5 presents the calculation performance of the proposed protocol for analysis and a comparison with those introduced in relevant studies, and, lastly, Section 6 concludes this study.

2. Multi-Owner Multi-Tag Ownership Transfer Method

This study proposes a secure multi-owner multi-tag ownership transfer method for a mobile RFID network environment. Figure 1 illustrates the architecture of the proposed system. The mobile device of a member of the old owner group sends a signal for ownership transfer, reads the partial tags in the group to be transferred, and sends the transfer message collectively generated by these tags to a back-end server. The server then notifies the group of old owners. If the partial signature of n old owners who consent to partial tag ownership transfer exceeds the threshold value, then the group of old owners and the group of new owners jointly conduct ownership transfer.

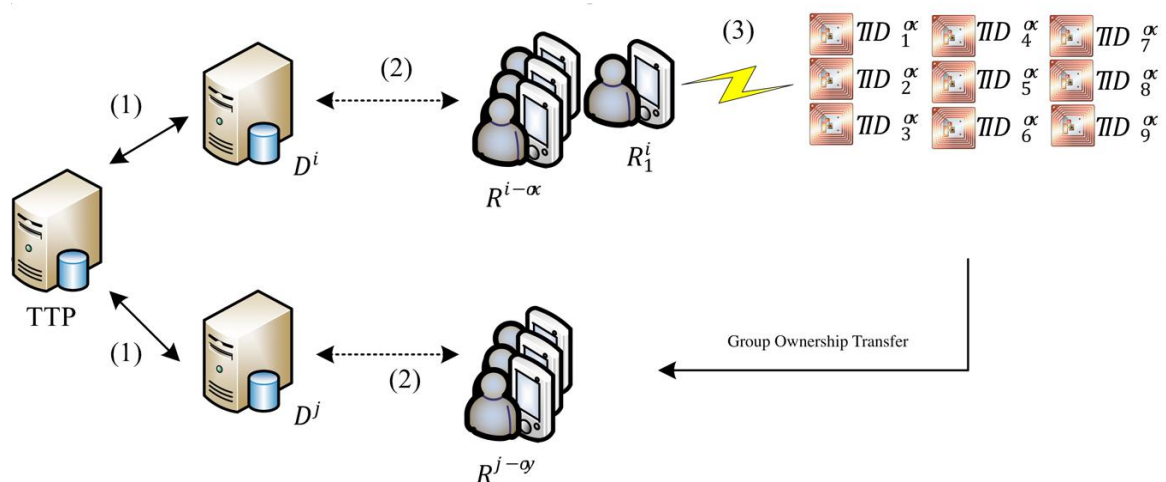


Figure 1. Multi-owner ownership transfer architecture.

2.1. System Architecture

The ownership network architecture in Figure 1 reveals that the servers, mobile readers, and tags have different computing capabilities. Connection security is discussed in three parts marked by (1), (2), and (3) in Figure 1. Connection (1) in Figure 1 shows that the server exhibits a computing capability sufficiently powerful to support existing encryption algorithms such as the Advanced Encryption Standard (AES) and Rivest–Shamir–Adleman (RSA). Therefore, the proposed protocol employs existing encryption methods to ensure secure communication between the servers. Connection (2) in Figure 1 indicates that the mobile readers engage in two-way communication with other mobile readers by using existing mobile communication technologies or wireless network technologies and back-end servers. Extant security communication technologies such as the X.509 security architecture of telecommunication networks or IEEE802.11i are employed to protect the transfer security of intermediate messages. Connection (3) in Figure 1 reveals that, because tags have limited computing capability, lightweight cryptography methods such as the data encryption standard lightweight [30]

or Grain [31] should be used when communicating with mobile readers to secure the messages communicated between the tag and mobile reader. Because the environment involved in part (3) is the riskiest, the present study assumes that the network environment in connection (3) is an insecure communication environment that is vulnerable to attacks. To provide a detailed description of the proposed ownership transfer method, the ownership transfer environment of the tag management service is assumed to be characterized by four properties.

The first property is that a mobile reader connects only to a single back-end server during communication to manage the tags and that the back-end server under the authority of the mobile reader to which ownership is transferred in and out is likely the same back-end server or a back-end server under different authority. In the set R^i of a readers under the authority of back-end server D^i with ID DID^i , any one of the readers R_a^i with independent ID RID_a^i must satisfy the relation in Equation (1).

$R^i = \{R_1^i, R_2^i, \dots, R_a^i\}$ where R^i is under the authority of D^i and R^j is under the authority of D^j , which satisfies the equation below.

$$\forall i, j \ R^i \cap R^j = \emptyset \text{ and iff } i \neq j; \text{ otherwise, } R^i \cap R^j = R^i \quad (1)$$

The second property is that the set R^i of readers under the back-end server D^i consists of a set R^{i-ox} of readers owned by p owners, and, in this multi-owner set R^{i-ox} , all of the owners have collective ownership to the tags in group G_0^{ox} . Without loss of generality, an ownership transfer protocol implemented through reader R_1^i must satisfy the relation in Equation (2).

$$R^{i-ox} = \{R_1^i, R_2^i, \dots, R_p^i\}, R^{i-ox} \subseteq R^i, \text{ satisfying } \forall ox, oy \ G_0^{ox} \cap G_0^{oy} = \emptyset \quad (2)$$

$$\text{iff } ox \neq oy, \text{ otherwise } G_0^{ox} \cap G_0^{oy} = G_0^{ox}$$

The third property is that v tags in multi-owner set R^{i-ox} , which is managed by the back-end server D^i , belong to group G_0^{ox} , with the group key defined as GK_0^{ox} . The server splits the tag groups into a k -ary tree of group keys, which generates a key tree with the height $h_{max} = \lceil \log_k(\frac{v}{k}) \rceil$. Next, the sequence of the k -ary group ID moves from top to bottom and from left to right. The parent node $G_{\lfloor \frac{s-1}{k} \rfloor}^{ox}$, the children nodes G_{s*k+1}^{ox} through to G_{s*k+k}^{ox} , and their relationship with the rules are shown in Figure 2a. Group key GK_1^{ox} can encrypt the group message to TID_{13}^{ox} through to TID_{39}^{ox} , and tag IDs TID_1^{ox} , TID_2^{ox} , and TID_3^{ox} can use the keys Kt_1^{ox} , Kt_2^{ox} , and Kt_3^{ox} shared with the server to decrypt the group message that is encrypted by group key GK_9^{ox} . Therefore, the node with the group ID G_s^{ox} is composed of one to k subtrees, and the group key containing any node G_s^{ox} is expressed in Equation (3). The group key GK_s^{ox} consists of a parent group G_{spar}^{ox} containing G_s^{ox} , the intersection of G_s^{ox} with G_{spar}^{ox} equals G_s^{ox} , and the intersection of the difference between G_0^{ox} and G_{spar}^{ox} with G_s^{ox} is the empty set, as shown in Equation (4).

$$G_s^{ox} = \left\{ GK_l^{ox} \mid \forall l \ GK_l^{ox} \in G_s^{ox}, sk^h + \frac{k^h - 1}{k - 1} \leq l \leq sk^h + \frac{k(k^h - 1)}{k - 1}, h \in \mathbb{Z}_0^+, s \in \mathbb{Z}_0^+ \right\} \quad (3)$$

$$G_{spar}^{ox} = \left\{ GK_s^{ox} \in G_s^{ox}, h \in \mathbb{Z}_0^+, s \in \mathbb{Z}_0^+ \right\}, \quad (4)$$

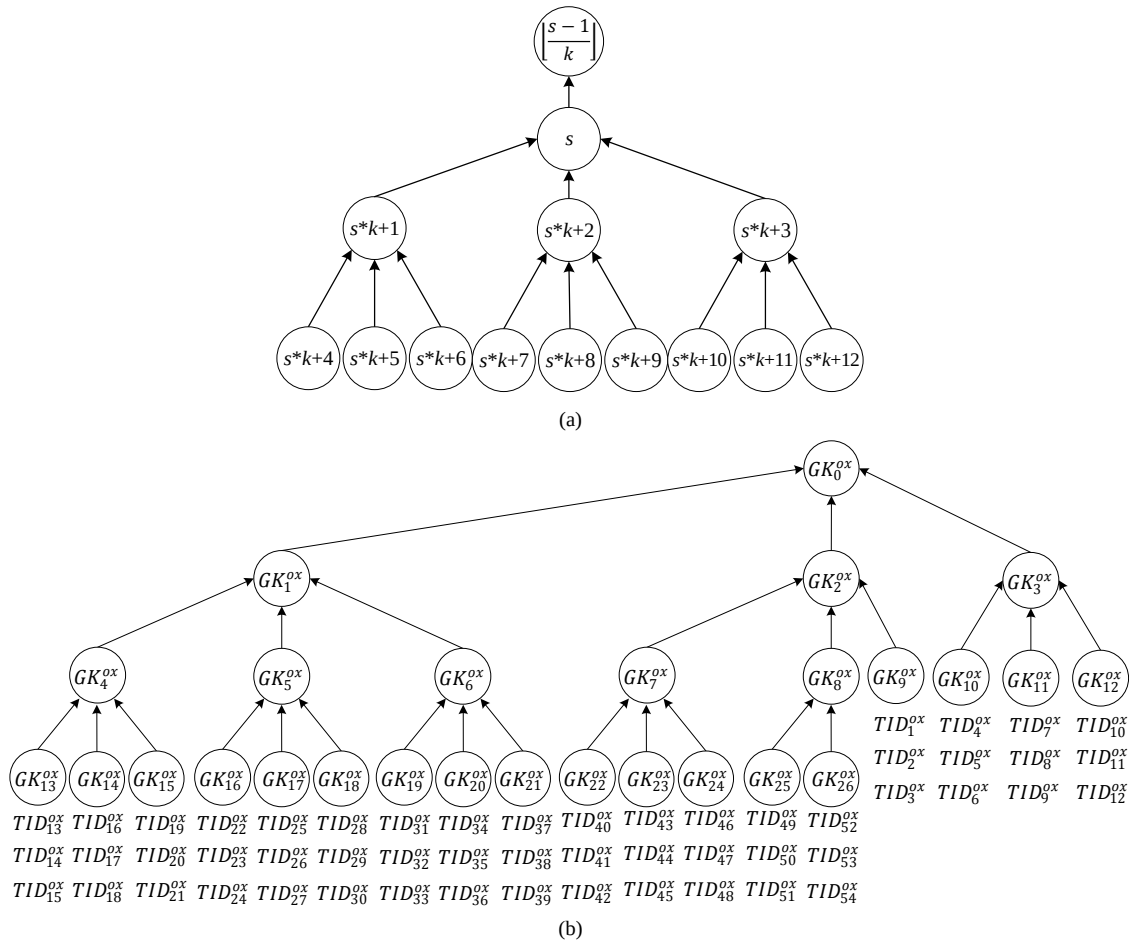


Figure 2. (a) Tags naming rule (b) Tree of tag group key.

$\forall s$ GK_s^{ox} is the authority of R^{i-ox} , satisfying $G_s^{ox} \cap G_{spar}^{ox} = G_s^{ox}$ and $(G_0^{ox} - G_{spar}^{ox}) \cap G_s^{ox} = \emptyset$

The fourth property is the leaf group that is defined as the leaf node with IDs $\left\lceil \frac{(\frac{v}{k})-1}{k-1} \right\rceil$ through $\left\lceil \frac{(\frac{v}{k})-1}{k-1} \right\rceil + \left\lceil \frac{v}{k} \right\rceil - 1$ on $\left\lceil \frac{v}{k} \right\rceil$ key trees connected to the tags, as shown in Equation (5). For example, tag IDs TID_1^{ox} , TID_2^{ox} , and TID_3^{ox} are connected to the leaf node $G_{leaf,1}^{ox}$ with group ID G_9^{ox} .

$$G_{leaf,m}^{ox} = \left\{ TID_l^{ox} \mid \forall TID_l^{ox} \in G_{leaf,m}^{ox}, (m-1)k+1 \leq l \leq mk, 1 \leq m \leq \left\lceil \frac{v}{k} \right\rceil \right\} \quad (5)$$

Figure 3 presents an example in which an owner's reader R_1^i intends to transfer the ownership of tag IDs TID_4^{ox} , TID_5^{ox} , and TID_6^{ox} in tag group G_5^{ox} (indicated by the dashed line in the middle of the key tree) from the owner group R^{i-ox} under the authority of D^i to the owner group R^{j-oy} under the authority of D^j . If the owner's reader R_1^i obtains consent from the group of old owners through the server D^i of the old owner group to partially transfer tags, then server D^i uses the communication key shared between D^i and the tags on the topmost level in Figure 3 to generate and send the ownership transfer message to a trust third party (TTP). Subsequently, the TTP sends an update-key message to multi-owner sets R^{i-ox} and R^{j-oy} and tag group G_0^{ox} to simultaneously update the key and avoid the windowing problem. As indicated on the right of Figure 3, because the back-end server D^j has authority over its own key tree, when tag IDs TID_4^{ox} , TID_5^{ox} , and TID_6^{ox} are transferred to multi-owner sets R^{j-oy} , the owner inserts tag IDs TID_7^{oy} , TID_8^{oy} , and TID_9^{oy} in tag group G_3^{oy} on the far right, and these IDs are not necessarily the same as the output tag IDs. Lastly, the TTP changes the shared keys of tags TID_4^{ox} , TID_5^{ox} , and TID_6^{ox} under D^i , as shown at the bottom of Figure 3, to a shared management

key of tags TID_7^{oy} , TID_8^{oy} , and TID_9^{oy} under D^j . Details of the ownership transfer protocol are provided in Section 3.

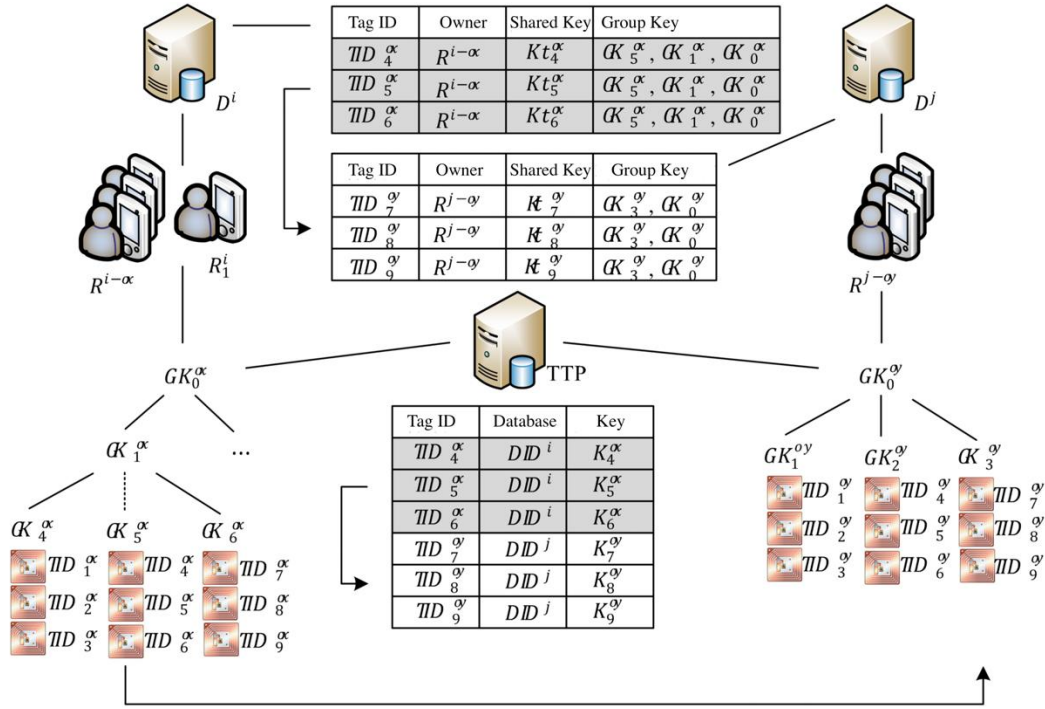


Figure 3. Example of tag group transfer IDs.

2.2. Transfer of Multiple Tag Groups

Tags to be transferred may belong to different groups. In other words, the transfer of ownership of all the tags to be transferred cannot be achieved by transferring the tags of a single group. In this case, the proposed multi-owner multi-tag ownership transfer method is implemented multiple times to solve this problem. As indicated in Figure 2b, because the tags $TID_{40}^{ox}-TID_{42}^{ox}$, $TID_{46}^{ox}-TID_{48}^{ox}$, and $TID_{52}^{ox}-TID_{54}^{ox}$ belong to different groups and the transfer of ownership of all of the tags cannot be achieved by transferring the tags of a single group. Transferring these tags requires the ownership of tag groups G_{22}^{ox} , G_{24}^{ox} , and G_{26}^{ox} to be transferred three times. The owners' readers are activated to send the ownership transfer request (OT) of groups IDs G_{22}^{ox} , G_{24}^{ox} , and G_{26}^{ox} to the tags. The ownership transfer of group IDs G_{22}^{ox} , G_{24}^{ox} , and G_{26}^{ox} can be simultaneously conducted by implementing the proposed protocol multiple times.

3. Multi-Owner Multi-Tag Ownership Transfer Protocol

3.1. Initialization

Table 2 lists the notations in this paper. In our protocol, consent must be obtained from the majority of owners of the reader set R^{i-ox} before ownership transfer. Therefore, the threshold signature scheme presented by Harn [32] is used to confirm majority consent before proceeding to transfer ownership. When n owners receive a transfer request, the partial signatures of only t consenting owners are required and are sent to the server for grouping and verification. If the partial signatures match the ownership transfer message, then most owners consented to tag ownership transfer. The next three steps are described as follows.

Table 2. Notations.

Symbol	Description
DID^i	ID of owner server D^i .
R^{i-ox}, R^{j-oy}	Multi-owner set belong to server D^i and D^j
R_1^i	One of the owners belonging to R^{i-ox} .
TID_v^{ox}	v th tag ID owned by R^{i-ox} .
G_s^{ox}	sth group ID owned by R^{i-ox} .
GK_s^{ox}	Group key of group ID G_s^{ox} and server D^i .
$K_{TID_v}^{ox}$	Key shared between tag ID TID_v^{ox} and server D^i .
K_v^{ox}	Management key shared between tag ID TID_v^{ox} and the TTP.
SK_p^i	Key shared between the two entities i and p .
K_{TTP}^{ox}	Secret value shared between the tag group and the TTP.
N_r	Random number generated by the readers of multiple old owners.
OT	Ownership transfer request.
$\parallel$	Signal connection notation.
$E(\text{key}, \text{msg})$	Symmetric keyed encryption/decryption function, which uses a key to encrypt and decrypt a message.
$LE(\text{key}, \text{msg})$	Lightweight symmetric keyed encryption/decryption function, which uses a key to encrypt and decrypt a message.

The first step is the group key and the secret key generation stage. This step must be completed before ownership transfer. Each owner is allocated a secret key for partial signature generation, a verification key for partial signature generation, and a verification key for group signature generation.

The second step is the partial signature generation stage. When one of the owners of the reader set R^{i-ox} sends an ownership transfer request message (OT), the server asks other owners whether they consent to ownership transfer. Consenting owners send the partial signature generated by the request message to the server for verification.

The third step is the group signature verification stage. The server conducts verification and grouping after receiving the partial signature and then verifies whether the group signature matches the OT message. If the message matches, then most owners agreed to proceed with the ownership transfer.

3.2. Ownership Transfer Request

Figure 4 shows the selected reader wants to get the ownership transfer permission from the original owners. In Step 1, the reader R_1^i belonging to one of the owners in the multi-owner set R^{i-ox} of tag group G_s^{ox} expresses intent to transfer the ownership of the object represented by each tag in the tag group G_s^{ox} to the owners in the multi-owner set R^{j-oy} . First, owner R_1^i generates a random number N_r and uses the key SK_1^i to encrypt an OT, tag list (TL) of all the tags to be transferred, pseudo-random number N_r , server ID DID^j of the transfer target, and multi-owner set R^{j-oy} of the new owner. After encryption, message M_1 is generated and sent to the management server D^i of R_1^i requesting D^i to ask other owners whether they consent to ownership transfer.

In Step 2, after server D^i receives M_1 , it decrypts the message by using the key SK_1^i shared with owner R_1^i , and confirms that owner R_1^i wants to transfer the tags in the TL. Server D^i encrypts the OT, server ID DID^j of the transfer target, multi-owner set R^{j-oy} of the new owner, TL of all the tags to be transferred, and pseudo-random number N_r by using the key SK_p^i , which is shared between server D^i and each owner in the multi-owner set R^{i-ox} . Lastly, the server sends message M_p to each owner in the multi-owner set R^{i-ox} asking whether they consent to the ownership transfer.

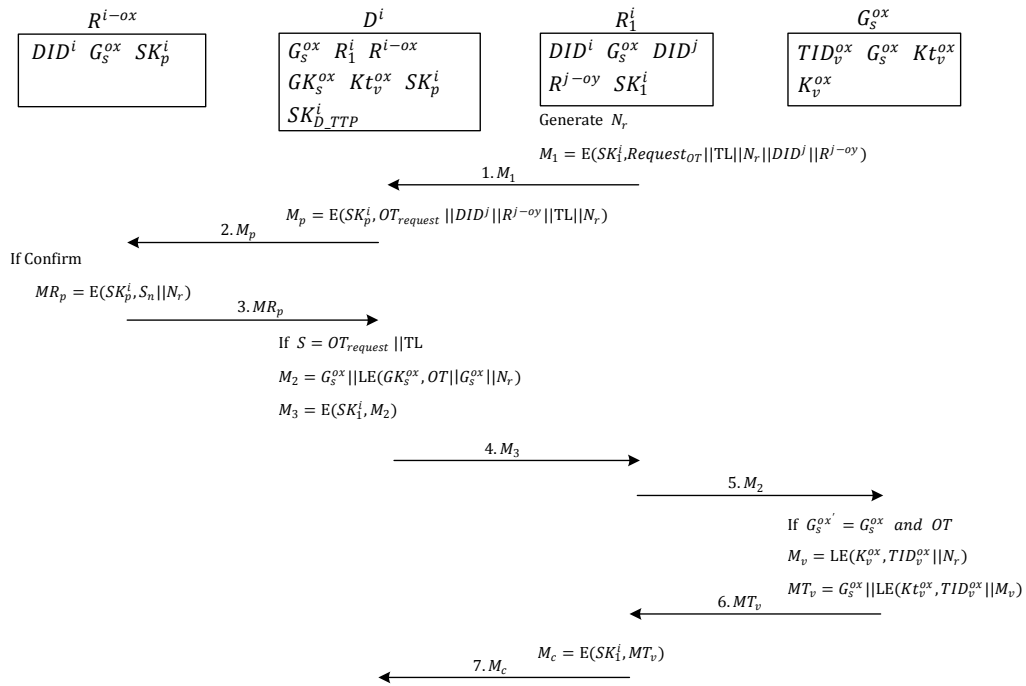


Figure 4. Ownership transfer request.

In Step 3, when each owner in the multiowner set R^{i-ox} with the ownership of tag group G_s^{ox} receives M_p , the message M_p is decrypted using the key SK_p^i , which is shared between server D^i and each owner. The consent of the owner to the ownership transfer is verified. If consent is provided, then the key SK_p^i shared between server D^i and each owner is used to encrypt the partial signature S_n and pseudo-random number N_r , which generates the message MR_p . This is then sent to server D^i .

In Step 4, server D^i collects the message MR_p from owners in the multi-owner set R^{i-ox} and uses the key SK_p^i , which is shared between server D^i and owners, to decrypt message MR_p and perform the verification of partial signature S_n to check whether it matches the signature message $OT_{request} || TL$. If the signature matches, then most owners consented to ownership transfer and the protocol proceeds with the ownership transfer. If the collected partial signatures are not enough, or the signature does not match, then owners did not consent to ownership transfer and the protocol terminates the ownership transfer.

When most owners provide consent to proceed with ownership transfer, server D^i first identifies the communication key GK_s^{ox} of the tag group in the TL. Next, it uses the group communication key GK_s^{ox} to encrypt the OT (approved by the owners in the multi-owner set R^{i-ox}), group G_s^{ox} of tags to be transferred, the pseudo-random number N_r , and group ID G_s^{ox} , subsequently generating message M_2 . Message M_2 is then encrypted using the communication key SK_1^i , which is shared with the owner, to produce message M_3 , which is transmitted to owner R_1^i .

In Step 5, after owner R_1^i receives message M_3 , it decrypts the message by using the communication key SK_1^i shared with server D^i . Then the extracted message M_2 is sent to tag group G_s^{ox} as a broadcast message.

In Step 6, when any tag TID_v^{ox} in tag group G_s^{ox} receives message M_2 , the tag confirms whether the group tag ID in the message matches the ID G_s^{ox} of the tag group to which it belongs. After confirming that it belongs to the tag group G_s^{ox} , each tag uses the communication key Kt_v^{ox} shared with server D^i to decrypt the OT (approved by owners in the multi-owner set R^{i-ox}) and then reconfirms that the tag group G_s^{ox} to be transferred matches the ID of the tag group G_s^{ox} to which it belongs. If the IDs match, then the tag uses the management key K_v^{ox} shared with the TTP to encrypt its tag ID TID_v^{ox} and the pseudo-random number N_r from owner R_1^i , which produced the authentication message M_v . Next,

the tag uses the communication key Kt_v^{ox} shared with server D^i to encrypt tag ID TID_v^{ox} , pseudo-random number N_r , and its tag group ID G_s^{ox} , which generates message MT_v and is sent to owner R_1^i .

In Step 7, after owner R_1^i receives message MT_v , it decrypts the message by using the communication key SK_1^i shared with server D^i , subsequently producing message M_c , which is then sent to the management server D^i of R_1^i .

3.3. Authentication of Tags and Transfer of Ownership

Figure 5 shows the authentication of tags and the ownership transfer process. In Step 8, because tag group G_s^{ox} may contain more than one tag, server D^i must collect message M_c sent by all of the tags in tag group G_s^{ox} . Subsequently, because tag group G_{sleaf}^{ox} is a set of all leaf nodes of the group ID subtree of group G_s^{ox} , G_{sleaf}^{ox} comprises all of the tags in tag group G_s^{ox} . When server D^i receives message M_c , the server decrypts each message M_c by using the key SK_1^i shared with owner R_1^i to extract MT_v and then decrypts each message MT_v by using the secret value Kt_v^{ox} shared between server D^i and each tag to compare the tag ID with the authentication tag. Next, the server performs comparisons to determine whether all of the tag IDs in tag group G_{sleaf}^{ox} are consistent with the tag IDs received, checks whether all tags in tag group G_{sleaf}^{ox} are available, and checks whether R_1^i has ownership of tag group G_s^{ox} .

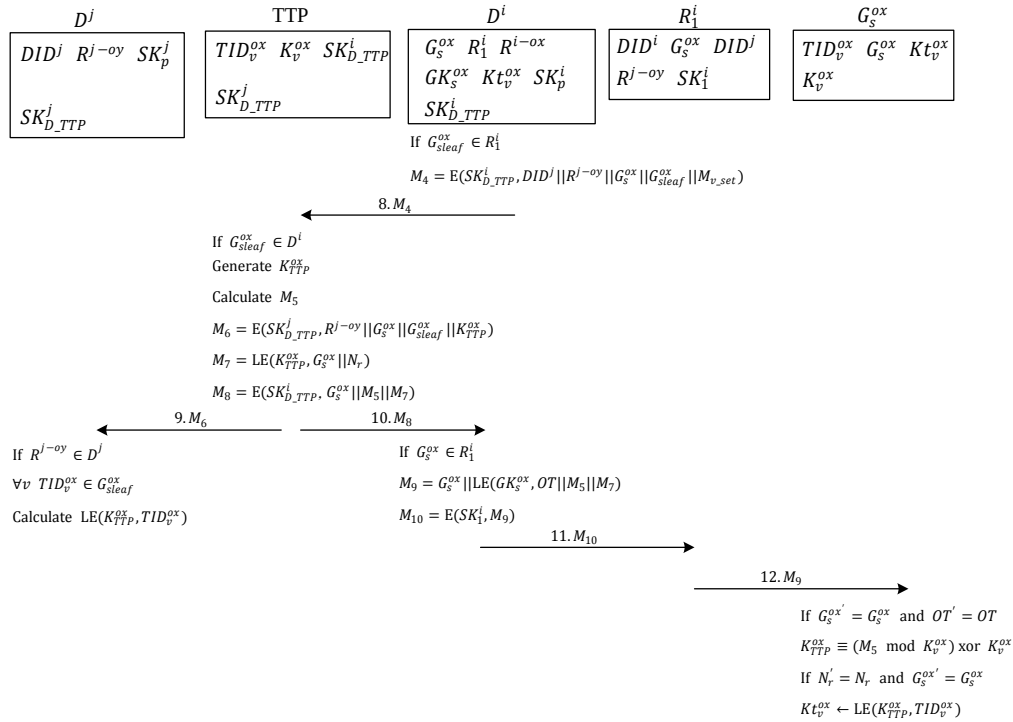


Figure 5. Authentication of tags and transfer of ownership.

Server D^i uses the communication key $SK_{D_TTP}^i$ shared with a TTP to encrypt the server's identity DID^j of the transfer target, multi-owner set R^{j-oy} of the new owner, tag group G_s^{ox} , and tag group G_{sleaf}^{ox} , which consists of the IDs of all the tags to be transferred. M_{v_set} of message M_v is transmitted by all of the tags in tag group G_{sleaf}^{ox} . M_{v_set} is defined in Equation (6). Subsequently, message M_4 is generated and transmitted to the TTP, which requests that the TTP use the management key shared between the TTP and the tags to update the communication key on the tag for completing the ownership transfer.

$$M_{v_set} = \left\{ LE(K_v^{ox}, TID_v^{ox} || N_r) \mid \forall v TID_v^{ox} \in G_{sleaf}^{ox} \right\} \quad (6)$$

In Steps 9 and 10, after the TTP receives message M_4 , the communication key $SK_{D_TTP}^i$ shared with server D^i is used to decrypt message M_4 . After confirming that the tag group G_{sleaf}^{ox} belongs to server D^i , the TTP uses the management key K_v^{ox} shared between each tag and the TTP to decrypt each tag message M_v in the M_{v_set} . The TTP then extracts tag ID TID_v^{ox} to authenticate the tag and determine whether each tag ID in tag group G_{sleaf}^{ox} within message M_4 matches the tag IDs in M_{v_set} and checks whether the pseudo-random number is the same for all tag messages M_v . If all of them match, then the TTP randomly generates a new secret value K_{TTP}^{ox} and uses the Chinese remainder theorem [33] in Equation (7) to calculate message M_5 .

$$M_5 \equiv \sum_{s=1}^{k^{h_{max}-\log_k^r}} \left((K_{TTP}^{ox} \text{ xor } K_s^{ox}) * m_s^{ox} * m'_s^{ox} \right) \pmod{M}, \text{ where } M = \prod_{s=1}^{k^{h_{max}-\log_k^r}} K_s^{ox}, \quad (7)$$

$$m_s^{ox} = \frac{M}{K_s^{ox}}, m'_s^{ox} * m_s^{ox} \equiv 1 \pmod{K_s^{ox}}$$

The TTP encrypts the multi-owner set R^{j-oy} of the new owner, tag group G_s^{ox} , and tag group G_{sleaf}^{ox} , which consists of the IDs of all of the tags to be transferred, and the secret value K_{TTP}^{ox} into message M_6 by using the communication key $SK_{D_TTP}^j$ shared between the TTP and server D^i . Next, the random number N_r and G_s^{ox} are encrypted into message M_7 by using the secret value K_{TTP}^{ox} . M_5 , M_7 , and G_s^{ox} are encrypted into message M_8 by using the communication key $SK_{D_TTP}^i$ shared between the TTP and server D^i . Messages M_6 and M_8 are then sent to the new owners and the old owners for ownership transfer.

After server D^i of the new owners receives messages M_6 , it decrypts the message by using the communication key $SK_{D_TTP}^j$ shared with the TTP. Next, server D^i determines that the tag group G_s^{ox} is to be transferred to the multi-owner set R^{j-oy} . It first verifies that R^{j-oy} is under the authority of D^i , obtains the secret value K_{TTP}^{ox} that is used to generate the communication key of all individual tags G_{sleaf}^{ox} in tag group G_s^{ox} , and uses this secret value to encrypt each tag ID TID_v^{ox} to update the communication key of the new owner's back-end server.

In Step 11, after server D^i of the old owners receives messages M_8 , it decrypts the message by using the communication key $SK_{D_TTP}^i$ shared with the TTP and determines that G_s^{ox} is part of the transfer message of the requesting owner R_1^i in R^{i-ox} . The server then uses group communication key GK_s^{ox} to encrypt the OT, M_5 , M_7 , and group ID G_s^{ox} into message M_9 . Subsequently, it uses the communication key SK_1^i shared with owner R_1^i to encrypt message M_9 into message M_{10} , which is then sent to owner R_1^i .

In Step 12, after the owner R_1^i receives messages M_{10} , it decrypts the message by using the communication key SK_1^i shared with server D^i to extract message M_9 , which is sent to tag group G_s^{ox} as a broadcast message. When any tag receives message M_9 , it confirms whether the group tag ID in the message matches the ID G_s^{ox} of the tag group to which it belongs. After confirming that it belongs to the tag group G_s^{ox} , each tag uses the communication key K_v^{ox} shared with server D^i to decrypt and compare the OT. Tag TID_v^{ox} uses message M_5 to calculate the secret value K_{TTP}^{ox} that is used to generate the communication key of all individual tags in tag group G_s^{ox} . Next, the tag uses the calculated secret value K_{TTP}^{ox} to decrypt message M_7 , checks whether tag group G_s^{ox} matches the claimed tag group ID, and then checks whether the random numbers N_r are identical (if they are identical, then they represent the same communication). After authentication, the secret value K_{TTP}^{ox} is used to encrypt tag ID TID_v^{ox} to replace tag TID_v^{ox} and the communication key of the back-end server. When all tags in tag group G_s^{ox} are completely updated, the ownership of this group has been transferred from the multi-owner set R^{i-ox} to the set R^{j-oy} .

3.4. Group Update and Balancing of the Key Tree

When implementation of the ownership transfer protocol is complete, the shared key on each transferred tag has been updated to the key shared with the back-end server of the new multi-owner set. Therefore, the server of the old owners can no longer be updated. However, a group communication key has yet to be established. When a member joins or leaves a group, the group communication key must be updated, and the balance state of the tree architecture must be checked. After the tag group architecture has been reconstructed in the server, existing methods for updating the group communication key, such as the approach proposed by Xu et al. [34], are used to update the communication key shared between groups.

4. Security Analysis

The method proposed in this study assumes that, after the mobile reader and back-end server authenticate each other by using an existing network security architecture, a shared communication key can be used in the subsequent protocol to identify the message deliverer and that, during communication, a shared communication key can be used for encryption to ensure secure communication. Hence, the following sections provide a security analysis of confidentiality between the tag and mobile reader, anti-replay attack, anti-man-in-the-middle attack, forward and backward secrecy, the windowing problem, location privacy protection, and an anti-DoS attack.

A. Confidentiality

Communication between a mobile reader and a back-end server is encrypted using a shared communication key to ensure secure communication. Communication between a tag and mobile reader is encrypted using a communication key shared by the tag and the back-end server and a management key shared by the tag and the TTP. Attackers cannot decrypt the encrypted messages and, thus, cannot access the communicated information.

B. Anti-Replay Attack

A random number and the communicated message are collectively encrypted so that, during the process of communication at all stages of the protocol, the sent message read by each tag changes, according to the random number. In step 1, if the attacker resends M_1 , the server will easily find it after decryption. This prevents attackers from completing authentication by replaying the previously acquired message.

C. Anti-Man-in-the-Middle Attack

Communication among a mobile reader, tags, and the back-end server is encrypted, and a communication key shared among these three entities is used to confirm identity in order to proceed with ownership transfer. Additionally, because attackers do not have a shared key and cannot complete authentication through replay attacks, attackers cannot counterfeit the reader or tag to implement a man-in-the-middle attack.

D. Forward Secrecy

In the protocol, the authentication and communication encryption key currently used by a tag are not given to the new owner. Instead, the new owner receives a new communication key, which is derived from a tag ID encrypted with a secret value that is randomly generated by the TTP. Therefore, the new owner cannot obtain the tag's original communication key to decrypt any messages that were encrypted previously using the tag's original key. Thus, forward secrecy is achieved.

E. Backward Secrecy

After the new owner acquires the new communication key of a tag, the TTP updates the key, which is shared between the old owner and the tag, through the old owner. Because the old owner's server and mobile reader have no access to the management key shared by the tags and the TTP, the old owner cannot access the secret value, which is randomly generated by the TTP and used to generate a new communication key. Therefore, the old owner cannot continue to track the tag's subsequent information. Thus, backward secrecy is achieved.

F. Windowing Problem

Because the old owner's server and mobile reader have no access to the management key shared by the tags and the TTP, the old owner cannot access the secret value, which is randomly generated by the TTP and used to generate a new communication key. Moreover, neither the old owner nor attackers can successfully update the tag's key by replaying the update-key message in the previous stage. This approach, thus, prevents the windowing problem in which both old and new owners hold tag ownership.

G. Location Privacy Protection

In the protocol, the mobile reader is only responsible for sending out a message because a mobile reader in a mobile RFID environment may be a malicious attacker. Therefore, the message sent between a tag and the back-end server is encrypted along with a random number. Thus, the attacker cannot track the tag by analyzing the content of the message sent between the mobile reader and tag or by analyzing the messages sent at different stages between the mobile reader and tag. In other words, the method proposed in the present study can protect the tag's location privacy.

H. Asynchronous Denial-of-Service Attack

Asynchronous DoS attack on the back-end server and tags may occur when a message is lost or when attackers maliciously block a message. In the proposed protocol, the back-end server retains the tags' keys before and after they are updated. Therefore, if the update-key message is lost or maliciously blocked, then the tag's pre-updated key can be used to decrypt the message sent by a tag, which enables the owner to read the message. This is otherwise prevented when the server and tag become asynchronized.

Table 3 presents a comparison of the proposed protocol and other ownership transfer methods in terms of the following security concerns: forward secrecy (FS), backward secrecy (BS), replay attack (RA), DoS attack, the windowing problem (WP), and group ownership transfer (GO). In Table 3, "V" represents a secure protocol and "X" denotes an insecure protocol.

Table 3. Comparison of security.

	Kapoor et al. [26]	Sundaresan et al. [27,28]	Our Protocol
Forward secrecy (FS)	V	X	V
Backward secrecy (BS)	V	X	V
Replay attack (RA)	X	X	V
Denial of service attack (DoS)	X	X	V
Windowing problem (WP)	V	V	V
Group ownership transfer (GO)	X	V	V

5. Performance Analysis

This section details the analysis conducted in the present study on the calculation and information load of the proposed ownership transfer method. In this study, the proposed method is compared with

other methods by analyzing and comparing the calculation load required to transfer the ownership of n tags belonging to old owners os to new owners ns . T_E represents the amount of time required to calculate encryption and decryption once. T_{LE} represents the amount of time required to calculate lightweight encryption and decryption once. T_{RNG} represents the amount of time required to generate a random number, and T_H represents the amount of time required to compute a hash function once.

The following aspects were compared between the method proposed in the present study and those proposed by Kapoor et al. [26] and Sundaresan et al. [27,28]: the calculation load of tags, readers, servers, and the information load of the whole protocol. In the multi-owner single-tag ownership transfer method developed by Kapoor et al., encryption/decryption, hash functions, exclusive or (XOR), and random numbers are used. In their protocol, tags are transferred individually. To transfer a large number of tags, the protocol must be implemented multiple times, which results in decreased efficiency. Similarly, the multi-owner multi-tag ownership transfer method presented by Sundaresan et al. uses only pseudorandom number generator (PRNG), XOR, and random numbers. However, in their protocol, the group key shared among the server, the tag group, and the owners is used to individually generate transfer request messages for the transfer of each tag. Consequently, transfer efficiency is affected by the number of owners and tags. The method proposed in the present study uses k -ary to generate a tag group key tree architecture, which leads to different situations when transferring ownership.

Situation (1): the whole group of tags is transferred, as illustrated in Figure 6. In this situation, the server only requires one set of group keys GK_0^{ox} to notify all tags of the transfer request. Regarding the calculation, each tag only requires encryption/decryption to be performed six times. The reader, in order to send the message to all tags and read the message returned by each tag to the server, only needs to perform multi-owner confirmation once and message broadcasting once. The server is only required to perform the calculation once.

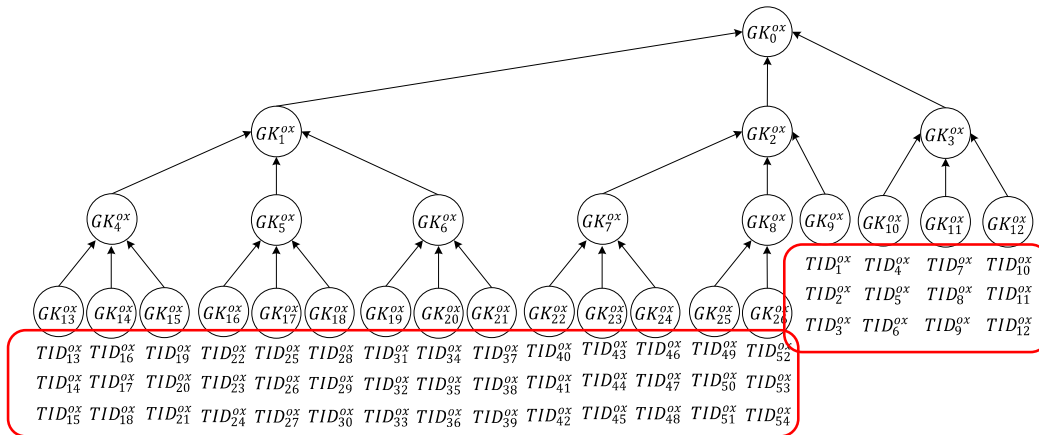


Figure 6. Tag transfer in Situation (1).

Situation (2): Only two tag groups are transferred, as shown in Figure 7. In this situation, the server requires two sets of group keys GK_{24}^{ox} and GK_{26}^{ox} to notify all tags of the transfer request. Regarding calculation, each tag is only required to conduct encryption/decryption computation six times. The reader, in order to send a message to all tags to be transferred and read the message that is returned by each of these tags to the server, is only required to perform multi-owner confirmation once and message broadcasting twice. The server is only required to perform calculation twice.

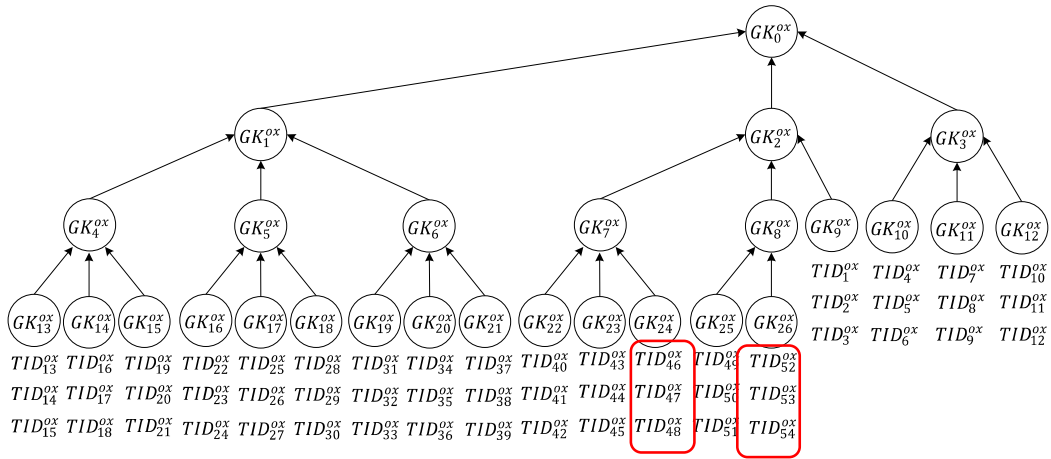


Figure 7. Tag transfer in Situation (2).

Situation (3): Only a single tag is not transferred, as depicted in Figure 8. In this situation, the server requires four sets of group keys and two sets of tag keys GK_1^{ox} , GK_2^{ox} , GK_{10}^{ox} , GK_{11}^{ox} , Kt_{10}^{ox} , and Kt_{11}^{ox} to notify the tags that are to be transferred of the transfer request. Regarding calculation, each tag is only required to conduct encryption/decryption computation six times, and the reader, in order to send a message to all tags to be transferred and read the message that is returned by each of these tags to the server, is only required to perform multi-owner confirmation once and message broadcasting six times. The server only needs to perform the calculation six times.

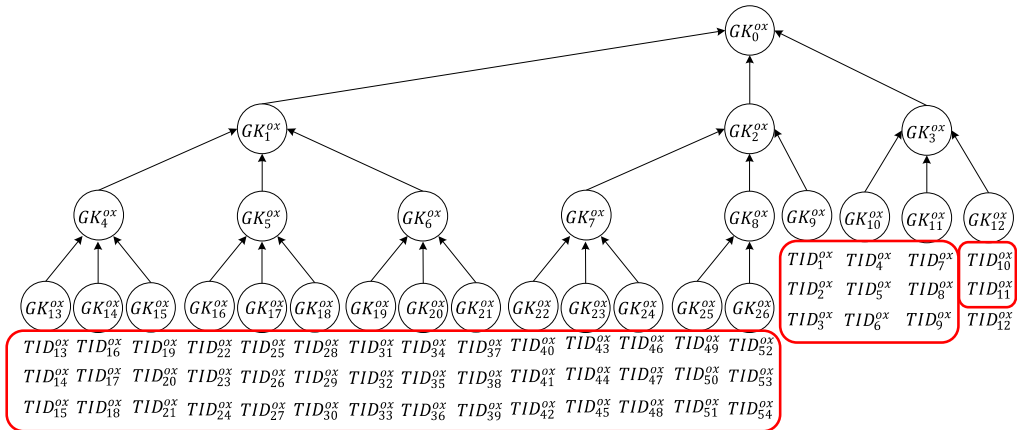


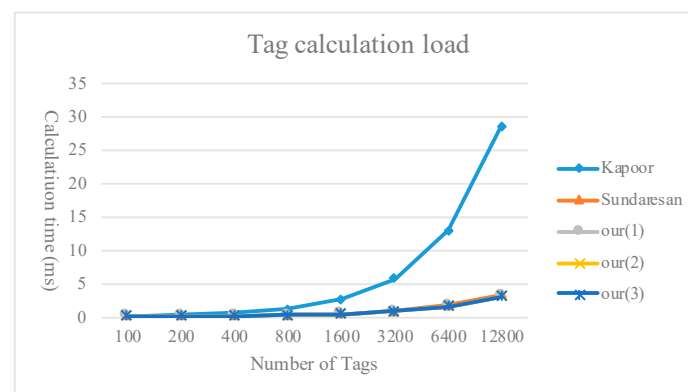
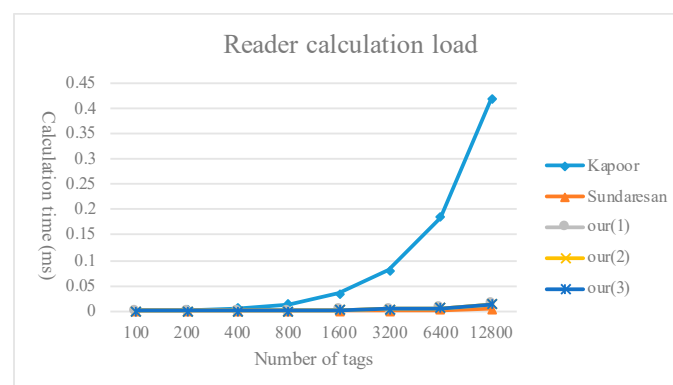
Figure 8. Tag transfer in Situation (3).

The calculation load of the proposed ownership transfer method is indicated in Table 4, which provides an analysis of all components participating in the ownership transfer process. Compared with the encryption method, XOR operation and comparison operation exhibit a lighter calculation load and, thus, are negligible. In Table 4, message encryption and decryption are assumed to be performed through one calculation. The server calculates T_{RNG} once for each key produced.

Table 4. Performance comparison.

	Device	Calculation Load	Information Load
Kapoor et al. [26]	Tag	$(ns + 1)T_{LE} + (ns + 1)T_H + 2T_{RNG}$	$os + 4ns + 2$
	Reader	$(os + ns)T_E + nsT_{LE} + 2nsT_H + nsT_{RNG}$	
	Server	$(os + ns)T_E + T_{LE} + (ns + 1)T_H + 3T_{RNG}$	
Sundaresan et al. [27,28]	Tag	$(5n + 3ns)T_{RNG}$	$9ns + 5n + (2n)ns$
	Reader	$(6ns + 2n)T_{RNG}$	
	Server	$(9ns + 4n + (2n)ns + 4)T_{RNG}$	
Our protocol (1)	Tag	$6nT_{LE}$	$2os + 2n + 8$
	Reader	$(2os + n + 3)T_E + T_{RNG}$	
	Server	$(2os + n + 9)T_E + (3n + 3)T_{LE} + T_{RNG}$	
Our protocol (2)	Tag	$6nT_{LE}$	$2os + 2n + 15$
	Reader	$(2os + n + 5)T_E + T_{RNG}$	
	Server	$(2os + n + 17)T_E + (3n + 6)T_{LE} + 2T_{RNG}$	
Our protocol (3)	Tag	$6nT_{LE}$	$2os + 2n + 43$
	Reader	$(2os + n + 13)T_E + T_{RNG}$	
	Server	$(2os + n + 49)T_E + (3n + 18)T_{LE} + 6T_{RNG}$	

Figures 9–11 compare the calculation load of the tags, readers, and servers between the method proposed in the present study and existing multi-owner ownership transfer methods.

**Figure 9.** Comparison of the tag calculation load.**Figure 10.** Comparison of the reader calculation load.

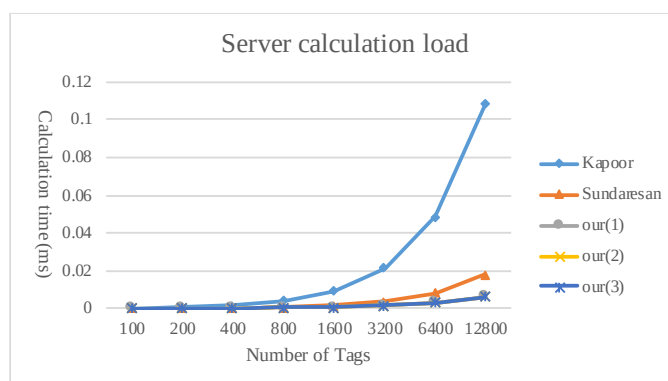


Figure 11. Comparison of the server calculation load.

6. Conclusions

The present study proposes a secure, high-performance multi-owner partial tag ownership transfer protocol. A threshold scheme is adopted to ensure that the consent of a specific number of old owners is obtained before the ownership of a tag group can be partially transferred to new owners. This study introduces a method that can transfer the ownership of one, some, or all tags. The calculation load of the tags, readers, and corresponding servers was analyzed and compared with those of other multi-owner ownership transfer protocols. The results verified the high performance of the proposed protocol. Additionally, compared with other multi-owner ownership transfer methods, the proposed protocol is effectively more immune to RFID-defined attacks, such as eavesdropping, replay, man-in-the-middle, and DoS attacks. With the proposed protocol, the owners at the front end of a supply chain can transfer their ownership securely and efficiently.

Author Contributions: J.-N.L. and M.-H.Y.: conceived and designed the protocols. J.-N.L.: security and performance analysis, wrote the paper. J.-N.L.: review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The Taiwan Information Security Center (TWISC) and the Ministry of Science and Technology (MOST) under Grant Nos. MOST 108-2218-E-011-021 and MOST 108-2221-E-033-016 funded this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Juels, A. RFID security and privacy: A research survey. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 381–394. [CrossRef]
2. Seidler, C. RFID Opportunities for Mobile Telecommunication Services, ITU-T Lighthouse Technical Paper. 2005. Available online: <http://www.itu.int/ITU-T/techwatch/rfid.pdf> (accessed on 17 December 2019).
3. Lee, H.; Kim, J. Privacy threats and issues in mobile RFID. In Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06), Vienna, Austria, 20–22 April 2006.
4. Engberg, S.J.; Harning, M.B.; Jensen, C.D. Zero-knowledge Device Authentication: Privacy & Security Enhanced RFID preserving Business Value and Consumer Convenience. In Proceedings of the Second Annual Conference on Privacy, Security and Trust, Fredericton, NB, Canada, 13–15 October 2004; pp. 89–101.
5. Van Deursen, T.; Mauw, S.; Radomirović, S.; Vullers, P. Secure ownership and ownership transfer in RFID systems. In Proceedings of the European Symposium on Research in Computer Security, Saint-Malo, France, 21–23 September 2009; pp. 637–654.
6. Osaka, K.; Takagi, T.; Yamazaki, K.; Takahashi, O. An efficient and secure RFID security method with ownership transfer. In *RFID Security*; Springer: Boston, MA, USA, 2008; pp. 147–176.
7. Juels, A.; Pappu, R. RFID privacy: An overview of problems and proposed solutions. *IEEE Secur. Priv.* **2005**, *3*, 34–43.

8. Fouladgar, S.; Afifi, H. An efficient delegation and transfer of ownership protocol for RFID tags. In Proceedings of the First International EURASIP Workshop on RFID Technology, Vienna, Austria, 24 September 2007; pp. 68–93.
9. Taqieddin, E.; Al-Dahoud, H.; Niu, H.; Sarangapani, J. Tag Ownership Transfer in Radio Frequency Identification Systems: A Survey of Existing Protocols and Open Challenges. *IEEE Access* **2018**, *6*, 32117–32155. [\[CrossRef\]](#)
10. Kapoor, G.; Piramuthu, S. Vulnerabilities in Some Recently Proposed RFID Ownership Transfer Protocols. *IEEE Commun. Lett.* **2010**, *14*, 260–262. [\[CrossRef\]](#)
11. Jäppinen, P.; Hämäläinen, H. Enhanced RFID security method with ownership transfer. In Proceedings of the 2008 International Conference on Computational Intelligence and Security, Suzhou, China, 13–17 December 2008; pp. 382–385.
12. Zuo, Y.J. Changing Hands Together: A Secure Group Ownership Transfer Protocol for RFID Tags. In Proceedings of the 2010 43rd Hawaii International Conference on System Sciences, Honolulu, HI, USA, 5–8 January 2010; pp. 1–10.
13. Yoon, E.J.; Yoo, K.Y. Two Security Problems of RFID Security Method with Ownership Transfer. In Proceedings of the 2008 IFIP International Conference on Network and Parallel Computing, Shanghai, China, 18–21 October 2008; pp. 68–73.
14. Chen, H.B.; Lee, W.B.; Zhao, Y.H.; Chen, Y.L. Enhancement of the RFID Security Method with Ownership Transfer. In Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, SKKU, Suwon, Korea, 15–16 January 2009; pp. 251–254.
15. Yang, X.; Xu, C.; Li, C. A privacy model for RFID tag ownership transfer. *Secur. Commun. Netw.* **2017**, *2017*, 5084636. [\[CrossRef\]](#)
16. Dimitriou, T. Key Evolving RFID Systems: Forward/Backward Privacy and Ownership Transfer of RFID Tags. *Ad Hoc Netw.* **2016**, *37*, 195–208. [\[CrossRef\]](#)
17. Lee, C.C.; Cheng, C.L.; Lai, Y.M.; Li, C.T. Cryptanalysis of Dimitriou’s key evolving RFID systems. In Proceedings of the Fifth International Conference on Network, Communication and Computing, Kyoto, Japan, 17–21 December 2016; pp. 229–233.
18. Kapoor, G.; Piramuthu, S. Single RFID tag ownership transfer protocols. *IEEE Trans. Syst. Man and Cybern. Part C (Appl. Rev.)* **2010**, *42*, 164–173. [\[CrossRef\]](#)
19. Yang, M.H. Across-authority Lightweight Ownership Transfer Protocol. *Electron. Commer. Res. Appl.* **2011**, *10*, 375–383. [\[CrossRef\]](#)
20. He, L.Y.; Gan, Y.; Yin, Y. Secure Group Ownership Transfer Protocol with Independence of Old Owner for RFID Tags. *Comput. Model. New Technol.* **2014**, *18*, 209–214.
21. Jannati, H.; Falahati, A. Cryptanalysis and enhancement of a secure group ownership transfer protocol for RFID tags. In Proceedings of the Global Security, Safety and Sustainability & e-Democracy, Thessaloniki, Greece, 24–26 August 2011; pp. 186–193.
22. Molnar, D.; Soppera, A.; Wagner, D. A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags. In Proceedings of the Selected Areas in Cryptography, Kingston, ON, Canada, 15–17 August 2005; Volume 3897, pp. 276–290.
23. Tsai, K.Y.; Luo, J.N.; Yang, M.H.; Liew, W.T. Novel designated ownership transfer with grouping proof. *Appl. Sci.* **2019**, *9*, 724. [\[CrossRef\]](#)
24. Yang, M.H.; Xie, K.P. TTP-Based Group Ownership Transfer in A Mobile RFID Environment. *Int. J. Digit. Content Technol. Its Appl.* **2013**, *7*, 51.
25. Yang, M.H. Secure Multiple Group Ownership Transfer Protocol for Mobile RFID. *Electron. Commer. Res. Appl.* **2012**, *11*, 361–373. [\[CrossRef\]](#)
26. Kapoor, G.; Zhou, W.; Piramuthu, S. Multi-tag and Multi-owner RFID Ownership Transfer in Supply Chains. *Decis. Support Syst.* **2011**, *52*, 258–270. [\[CrossRef\]](#)
27. Sundaresan, S.; Doss, R.; Zhou, W. Secure ownership transfer in multi-tag/multi-owner passive RFID systems. In Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, USA, 9–13 December 2013; pp. 2891–2896.
28. Sundaresan, S.; Doss, R.; Zhou, W.; Piramuthu, S. Secure Ownership Transfer for Multi-tag Multi-owner Passive RFID Environment with Individual-owner-privacy. *Comput. Commun.* **2015**, *55*, 112–124. [\[CrossRef\]](#)

29. Munilla, J.; Burmester, M.; Peinado, A. Attacks on ownership transfer scheme for multi-tag multi-owner passive RFID environments. *Comput. Commun.* **2016**, *88*, 84–88. [\[CrossRef\]](#)
30. Poschmann, A.; Leander, G.; Schramm, K.; Paar, C. New light-weight crypto algorithms for RFID. In Proceedings of the 2007 IEEE International Symposium on Circuits and Systems, New Orleans, LA, USA, 27–30 May 2007; pp. 1843–1846.
31. Ågren, M.; Hell, M.; Johansson, T.; Meier, W. Grain-128 a: A new version of Grain-128 with optional authentication. *Int. J. Wirel. Mob. Comput.* **2011**, *5*, 48–59. [\[CrossRef\]](#)
32. Harn, L. Group-oriented (t, n) threshold digital signature scheme and digital multisignature. *IEEE Proc. Comput. Digit. Tech.* **1994**, *141*, 307–313. [\[CrossRef\]](#)
33. Zhou, J.; Ou, Y.H. Key tree and Chinese remainder theorem based group key distribution scheme. In Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing, Taipei, Taiwan, 8–11 June 2009; pp. 254–265.
34. Xu, L.; Huang, C. Computation-efficient multicast key distribution. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *19*, 577–587.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).