An Efficient and Secure Method for Simultaneous Ownership Transfer of Multiple Mobile Readers

Ming-Hour Yang, *Member, IEEE*, Chien-Hung Chen, Jia-Ning Luo, M. Vijayalakshmi, and S. Mercy Shalinie

***Abstract*—In recent years, radio frequency identification technology has developed rapidly and has been applied in supply chains. Products in supply chains are varied and may belong to several different owners. Ownership transfer requires the consent of a majority of owners. This article proposed a method suitable for simultaneously transferring the ownership of a large number of goods. The method does not require a trusted third party and can securely and efficiently transfer group ownership for multiple owners with multiple tags. The old and new owner groups can be classified as different authorities. This method can be used in mobile readers to transfer one, some, or all tags in a group. It includes mutual authentication between tags, readers, and backend servers, and can ensure that only assigned owners can obtain the tag ownership. The method proved can resist most known attacks, such as secret disclosure attacks and replay attacks. It can also prevent attacks from the dishonest original owners. Finally, through experiments, we compared the proposed method with other many-to-many ownership transfer methods and demonstrated that the proposed method has better security and fewer transmitted messages than other methods.**

***Index Terms*—Designated ownership transfer, multiowner, ownership transfer, radio frequency identification (RFID), tag groups, threshold.**

I. Introduction

IN RECENT years, radio frequency identification (RFID) technology has developed rapidly and has been integrated in various everyday applications, such as rapid payment system, Internet of Things applications, continuous monitoring of physiological signals, medical-oriented services, and logistics systems [1], [2]. The EPCglobal Inc., approved the Class 1 Gen 2 air protocol in 2004, which leverages on the UHF specifications of RFID tag communications [3]. The Class 1 Gen 2 has faster and more flexible read and write speed, higher reliability, and robust performance. However, EPC Gen2 might yield to security and privacy violations if not handled properly. The RFID tags can be accessed by any nearby readers, which means that anyone could use the RFID tags to track items or identify the people associated with them through time and space [4].

Ownership transfer is a secure process of transferring ownership from old owners to new owners. In a supply chain management system, the manufacturers manufacture a product with RFID tag and sell it to a reseller and then to the consumer, the ownership of the tag is transferred to the last owner (the consumer), and the tag information is also transferred in the end of the process [5].

Ownership transfer schemes are categorized by the number of readers and tags involved in the transfer, such as one reader on one tag [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], one reader on multiple tags [16], [17], [18], [19], [20], and multiple readers on multiple tags [21], [22], [23]. Osaka et al. [7] proposed the one-to-one ownership transfer method. However, in this method, the message for updating the key may be tampered with by attackers, resulting in a desynchronization attack (DA). Forward security (FS) cannot be ensured, and the windowing problem (WP) cannot be avoided [24], [25]. Although Jappinen and Hämäläinen [26] inspected the integrity of the updated key's message to reduce the possibility of asynchrony between the tag and the backend server, they were unable to eliminate the problem because the method of inspecting the updated key's message could still be tampered with by an attacker [25]. Thus, Chen et al. [8] proposed a new protocol for avoiding FS and DAs. However, this method still had problems in terms of backward security (BS), inability of ensuring the privacy of the location, and the WP [9], [27]. Shen et al. [11] proposed using Chebyshev polynomials to ensure secure transmission during ownership transfer. By reducing the amount of message necessary for transfer protocols, efficiency was increased. However, if the attacker initiated an attack within the time threshold value, replay attacks (RAs) could still occur. Yang and Hu [28] proposed the self-organized time-division multiple access protocol that can support mobile readers without requiring trusted third party (TTP) for the transfer of tag ownership, resolving all of the aforementioned attacks. Aghili and Mala [29] proposed an ownership transfer protocol that could handle a dishonest original owner. The protocol prevents the original owner from attempting to regain ownership immediately after transferring ownership. Although the protocol could prevent attacks by the original owner, it was only effective for transfers of one tag at a time. Ownership could not be effectively transferred for a large number of tags.

In addition, in recent years, some studies have used blockchain technology to perform ownership transfer [12], [13], [14]. In

Manuscript received 4 May 2022; revised 4 November 2022; accepted 20 February 2023. Date of publication 13 March 2023; date of current version 30 August 2023. This work was supported in part by the National Science and Technology Council of Taiwan under Grant MOST 111-2221-E-033-047, Grant MOST 111-2218-E-A49-013-MBK, and Grant NSTC 111-2218-E-002-038 and in part by the National Defense Science and Technology Academic Collaborative Research Project. *(Corresponding author: Ming-Hour Yang.)*

Ming-Hour Yang and Chien-Hung Chen are with the Department of Information and Computer Engineering, Chung Yuan Christian University, Taoyuan City 320314, Taiwan (e-mail: mhyang@cycu.edu.tw; asd14526@cycu.org.tw).

Jia-Ning Luo is with the Department of Computer Science and Information Engineering, National Defense University, Taoyuan City 33448, Taiwan (e-mail: deer@ccit.ndu.edu.tw).

M. Vijayalakshmi and S. Mercy Shalinie are with the Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai 625015, India (e-mail: mviji@tce.edu; shalinie@tce.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSYST.2023.3251970, provided by the authors.

Digital Object Identifier 10.1109/JSYST.2023.3251970

[14], the transferred information is recorded on the blockchain using smart contracts of the Ethereum system.

To overcome this previous owner attack (OA), conducting attacks to secret disclosure after ownership transfer, and to protect owner privacy [29], this study proposes a multiowner multitag group ownership transfer (GOT) protocol that does not require a TTP, is secure, and is efficient. In the proposed method, the old and new owner groups are divided under different authorities. When an old owner asks for ownership transfer, we examine whether the number of agreeing owners exceeds a threshold to determine whether to transfer ownership. The ownership of the tag can then be safely transferred to the new ownership group. The proposed method has the following advantages.

1) This method can be used with mobile readers.
2) It can transfer tags belonging to two different authorities.
3) This method can transfer ownership for one tag, some tags, or all tags.
4) Mutual authentication between the tags, the readers, and the backend server occurs.
5) Only the designated multiowner can obtain ownership of the tag.
6) Ownership transfer can be conducted when most owners agree.
7) It prevents attacks by the original owner.
8) The proposed secure ownership transfer protocol can prevent secret disclosure attacks (SDAs), tag/reader impersonation, RAs, and message modification. It also provides FS and BS and prevents the WP.
9) The ownership transfer protocol is efficient. The method is effective for any number of readers and tags. An increase in owners or tags does not substantially increase the number of messages or the computational requirements.

The rest of this article is structured as follows. Section II describes the related work of ownership transfer protocols. Section III introduces the settings for an ownership transfer protocol and the relationship between tags, readers, and backend servers. Section IV details the proposed protocol. Section V analyzes the security of this ownership transfer method and compares it to relevant studies. Section VI analyzes the performance of this method and compares it to relevant studies. Finally, Section VII concludes the article.

II. Related Works

A factory at the beginning of a supply chain typically conducts ownership transfer for a large number of products simultaneously. Zuo [16] proposed a method for this situation. It involved using a group key to simultaneously verify and transfer ownership of multiple tags. However, a DA was possible when the key is updated at the end of the process [11]. Jannati and Falahati [17] proposed a method that increased the difficulty of an attack after updating a key. However, this method could only transfer all of the tags at once; it could not transfer only some tags. Lee et al. [19] integrated quadratic residue theory and homomorphic encryption to strengthen security. They used a cloud server to increase the efficiency of the method. However, their proposal still had problems, including SDAs, reader impersonation attacks (IAs), tag tracking, and FS [30]. Moazami and Safkhani [30] solved these problems and solved the DA in [31]. Yang [18] proposed a protocol for the ownership transfer of a group of tags. The method used the group communication key shared by the backend servers and the tags to generate a partial group communication key to transfer the ownership of all tags in a group simultaneously. In addition to supporting mobile readers, the method could fend off most known attacks. However, the method still has the WP in which the previous and new owners simultaneously have ownership before a TTP updates the key for the tags [27]. Tsai et al. [20] and Yang et al. [33] proposed a method of ownership transfer and grouping proof. In the protocol, the ownership transfer of a subset of tags can be conducted, and the method could ensure that the tags within a group were simultaneously and comprehensively transferred. Lee et al. [34] proposed a time bound group ownership delegation protocol based on homomorphic encryption and quadratic residue that is similar to their GOT protocol [35]. Kumar et al. [36] proposed time bound group ownership delegation protocols, which will revoke the ownership after a certain period of time, and over time, the ownership is revoked. Moazami and Safkhani resolved the security issues in [35].

If a product simultaneously has multiple owners, such as a product purchased by a joint venture, ownership cannot be transferred with the agreement of just one owner. The agreement of the majority of the owners must be obtained to conduct ownership transfer. Kapoor et al. [21] proposed a multiowner group tag ownership transfer protocol. However, the TTP's protocol is threatened by RAs and DAs 0. Moreover, the method could only transfer ownership for one tag. To transfer the ownership of numerous tags, the protocol must be run once for each tag, resulting in poor efficiency. Sundaresan et al. [22] proposed a TTP-based lightweight multiowner multitag ownership transfer method. However, for each tag and owner, substantial messages, calculations, and transmission time were required. Moreover, this method had difficulties with tag tracking and suffered from the FS problem [35]. Luo and Yang [23] proposed a group tag ownership transfer protocol via TTP. The protocol supports multiple transfers, and only assigned owners could participate in the ownership transfer. Most attacks could be prevented. However, the WP of dishonest owners remained. Previous owners could take back ownership before the tag key is updated [29].

III. Ownership Transfer Method Involving Multiowners Transferring Multitags

In this study, we proposed a novel RFID ownership transfer mechanism without using a TTP. In our system, there will be multiple owners who jointly own a group of RFID tags. To initiate the ownership transfer process of the RFID tags, a majority of owners must first agree to the transfer. After confirmation of the agreement, a delegated mobile reader will transfer the ownership of a subset of the tags to another group of owners. The contents of the selected RFID tags will also be transferred from the original owners' server to the new owners' server. We assume that the tag has limited computing power. In order to avoid tags from becoming a bottleneck, tags will only use lightweight ciphers. The detailed system structure is described in the following sections.

The proposed framework is presented in Fig. 1. In the original ownership group, one member's mobile reader R_1^i initiates the ownership transfer. Most readers in R^{i-o} respond and agree to the transfer of ownership. Then, a message is sent to the new owners' reader R^{j-n} to transfer ownership. The details of the process are described in Section IV.

Fig. 1 indicates that mutual connections are required between server and server, server and mobile reader, and mobile reader

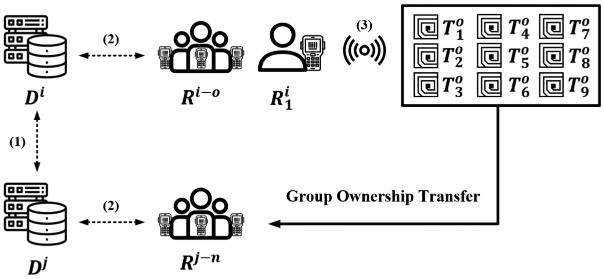


Fig. 1. Framework for a GOT with multiple owners.

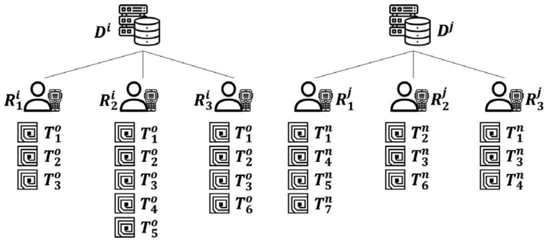

D^i
D^j
R_1^i R_2^i R_3^i R_1^j R_2^j R_3^j
T_1^o T_2^o T_3^o
T_1^o T_2^o T_3^o T_4^o T_5^o
T_1^o T_2^o T_3^o T_6^o
T_1^n T_4^n T_5^n T_7^n
T_2^n T_3^n T_6^n
T_1^n T_3^n T_4^n

Fig. 2. Relationship between server, readers, and tags.

and tag. On the basis of the computing capacity of these devices, we divided these connections into two parts for the discussion. The dashed lines (1) and (2) in Fig. 1 indicate secure communications channels between servers and the mobile readers. The radio symbol (3) in Fig. 1 indicates the wireless secure communication channels between the mobile readers and the tags, but the encryption methods that establish secure communications between tags and mobile readers are relatively weak [44] due to the limitation of RFID tag's hardware. The wireless channels are under threat of attacks, such as eavesdropping attacks, RAs, and man-in-the-middle attacks (MitMs). The symbols and their definitions are presented in Table I.

We assumed the environment for the ownership transfer of tag management services had four characteristics.

First, the backend server manages and stores the ownership relationship between all mobile readers and tags, as presented in Fig. 2. For example, suppose D^i and D^j each manage three mobile readers $\{R_1^i, R_2^i, R_3^i\}$ and $\{R_1^j, R_2^j, R_3^j\}$, respectively. Tags $\{T_1^o, T_2^o, T_3^o\}$ simultaneously belong to three mobile readers $\{R_1^i, R_2^i, R_3^i\}$. Mobile reader R_2^i has other tags $\{T_4^o, T_5^o\}$, and mobile reader R_3^i has another tag T_6^o. Mobile readers, such as R_1^i, do not store ownership data; they are only responsible for transferring messages. Mobile readers are each controlled by only one server. The backend server managing ownership transfer with mobile readers may be controlled by the same or different authorities. We assumed that a mobile reader has one owner to simplify the introduction of the ownership transfer process. Among the backend servers D^i, the server numbered DID^i has authority over a total of m readers. Any one reader R_m^i has an independent number RID_m^i. For a reader R_x^i controlled by D^i that intends to transfer the ownership to reader R_y^i under D^j's authority, the relationship in (1) must be satisfied

$$\{R_1^i, R_2^i, \ldots, R_m^i\} \in D^i,\ \left\{R_1^j, R_2^j, \ldots, R_n^j\right\} \in D^j,$$
$$\text{where } \forall i, j\ D^i \cap D^j = \emptyset. \qquad (1)$$

TABLE I
NOTATION

D^i	Backend server of the old owner
D^j	Backend server of the new owner
R^{i-o}	Reader set under
R^{j-n}	Reader set under D^j's authority
R_x^i	Reader in R^{i-o} with index number x
G_s^o	Tag set under R^{i-o}'s authority with index number s
T_v^o	Tag under D^i's authority v
DID^i	Identifier of the old server D^i
DID^j	Identifier of the new server D^j
TID_v^o	Identifier with index number v, under R^{i-o}'s authority
GID_s^o	Identifier of Tag group with index number s, under R^{i-o}'s authority
K^{ij}	Key shared with server D^i and server D^j
K^{i-o}	Key shared with server D^i and R^{i-o}
K_x^i	Key shared with server D^i and reader R_x^i
TK^i	Key generated by server D^i
TK_v^o	Key shared with TID_v^o and D^i
GK_s^o	Group key shared with tag group G_s^o and D^i
UGK^i	Threshold public group key
RSK_x^i	Private signing key with index number x, under R^{i-o}'s authority
S^{i-o}	Secret value shared with tag and server D^i
S^{j-n}	Secret value shared with tag and server D^j
$E(,)$	Symmetric key cryptography
$LE(,)$	Lightweight symmetric key cryptography
$H( )$	Hash function
N_r	Random number generated by Reader set R^{i-o}
OT	Ownership transfer request; includes $OT_{confirm}$ and OT_{Fail}
$\parallel$	Connection of messages
$\oplus$	XOR of messages

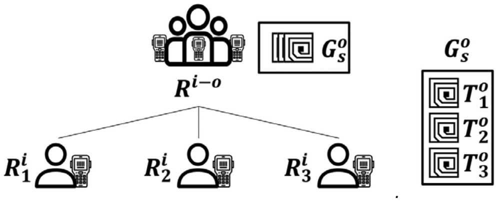


Fig. 3. Relationship graph of reader group R^{i-o} containing group tag G_s^o.

Second, each mobile reader has ownership of one or more tags. Each tag belongs to one or more owners. As presented in Fig. 2, the mobile readers $\{R_1^i, R_2^i, R_3^i\}$ co-own three tags $\{T_1^o, T_2^o, T_3^o\}$.

To facilitate the description of GOT, we use G_s^o (see Fig. 3) to indicate the three tag groups that are to be transferred out, where $G_s^o = \{T_1^o, T_2^o, T_3^o\}$. The set of readers that co-own the tag group G_s^o is R^{i-o}, where $R^{i-o} = \{R_1^i, R_2^i, R_3^i\}$. If the mth owner of

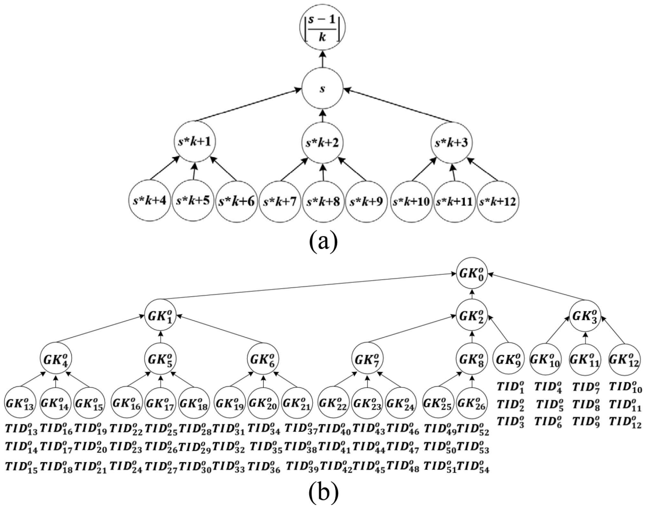

$\lfloor\frac{s-1}{k}\rfloor$
s
s*k+1
s*k+2
s*k+3
s*k+4
s*k+5
s*k+6
s*k+7
s*k+8
s*k+9
s*k+10
s*k+11
s*k+12
(a)
GK_0^o
GK_1^o
GK_2^o
GK_3^o
GK_4^o
GK_5^o
GK_6^o
GK_7^o
GK_8^o
GK_9^o
GK_{10}^o
GK_{11}^o
GK_{12}^o
GK_{13}^o GK_{14}^o GK_{15}^o GK_{16}^o GK_{17}^o GK_{18}^o GK_{19}^o GK_{20}^o GK_{21}^o GK_{22}^o GK_{23}^o GK_{24}^o GK_{25}^o GK_{26}^o
TID_1^o TID_4^o TID_7^o TID_{10}^o
TID_2^o TID_5^o TID_8^o TID_{11}^o
TID_3^o TID_6^o TID_9^o TID_{12}^o
TID_{13}^o TID_{16}^o TID_{19}^o TID_{22}^o TID_{25}^o TID_{28}^o TID_{31}^o TID_{34}^o TID_{37}^o TID_{40}^o TID_{43}^o TID_{46}^o TID_{49}^o TID_{52}^o
TID_{14}^o TID_{17}^o TID_{20}^o TID_{23}^o TID_{26}^o TID_{29}^o TID_{32}^o TID_{35}^o TID_{38}^o TID_{41}^o TID_{44}^o TID_{47}^o TID_{50}^o TID_{53}^o
TID_{15}^o TID_{18}^o TID_{21}^o TID_{24}^o TID_{27}^o TID_{30}^o TID_{33}^o TID_{36}^o TID_{39}^o TID_{42}^o TID_{45}^o TID_{48}^o TID_{51}^o TID_{54}^o
(b)

Fig. 4. Tag group key tree. (a) Numbering method. (b) Example of a 3-ary tree.

the reader set R^{i-o} that co-owns the tag group G_s^o comprising of p tags would like to transfer the ownership of G_s^o's ownership to the nth owner of the reader set R^{j-n} that co-owns the tag group G_s^n comprising q tags, these reader sets and tag sets must satisfy the relationship in (2). A reader of an old owner does not transfer any tags to any reader of an original owner, and no tags are shared by either transferring party

$$\begin{aligned} R^{i-o} &= \left\{R_1^i, R_2^i, \ldots, R_m^i\right\} ,\\ G_s^o &= \left\{T_1^o, T_2^o, \ldots, T_p^o\right\} ,\ G_s^o \in R^{i-o}\\ R^{j-n} &= \left\{R_1^j, R_2^j, \ldots, R_n^j\right\} ,\\ G_s^n &= \left\{T_1^n, T_2^n, \ldots, T_q^n\right\} ,\ G_s^n \in R^{j-n}\\ &\quad \text{where } \forall o, n\ G_s^o \cap\ G_s^n\\ &= \emptyset,\ R^{i-o} \cap\ R^{j-n} = \emptyset,\ \text{iff } o \neq n\,. \end{aligned} \tag{2}$$

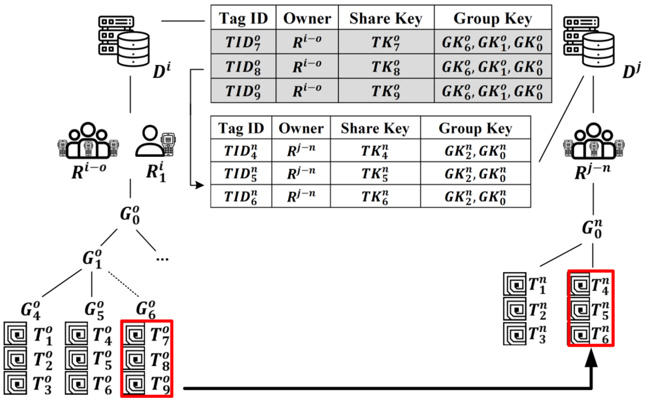
Third, to use a broadcast message to simultaneously transfer part of the group of tags in the reader set R^{i-o} containing the tag group G_s^o with p tags, we generate a k-ary group key tree. The height of the tree is $h_{\max} = \lceil \log_k(\frac{p}{k}) \rceil + 1$. The numbering sequence of the k-ary group is from top to bottom and left to right. The parent node number is $G^o_{\lfloor \frac{s-1}{k} \rfloor}$, and the child node numbers are from G^o_{s*k+1} to G^o_{s*k+k}. The numbering rules are presented in Fig. 4(a). We divided the group into a 3-ary group key tree ($k = 3$), and we defined the key at the top layer that owns all tag groups as GK_0^o. The group relationships are presented in Fig. 4(b). A total of 54 tags are included, namely TID_1^o–TID_{54}^o. Three tags comprise a tag group. These three tags share a group key. Groups at higher levels have more tags. The group key GK_1^o can encrypt broadcasted messages to TID_{13}^o–TID_{39}^o, and tags $TID_1^o,\ TID_2^o,\ and\ TID_3^o$ can use keys shared with the server, $TK_1^o,\ TK_2^o,\ and\ TK_3^o$, to decrypt the group message encrypted using the group key GK_9^o. Therefore, the node under group G_s^o has 1 to k child trees, and the group keys in any node G_s^o are defined in (3). The group key GK_s^o is stored in the parent group G_{spar}^o that includes G_s^o and satisfies the conditions that the intersection set of G_s^o and G_{spar}^o equals G_s^o and that the intersection set of the difference set of G_0^o and G_{spar}^o and of G_s^o is the empty set (4)

$$\begin{aligned} G_s^o = \Bigg\{ GK_l^o \Bigg| \forall l\ GK_l^o \in G_s^o, sk^h + \frac{k^h-1}{k-1} \leq l \\ \leq sk^h + \frac{k\left(k^h-1\right)}{k-1}, h \in \mathbb{Z}_0^+, s \in \mathbb{Z}_0^+ \Bigg\} \end{aligned} \tag{3}$$

$$\begin{aligned} G_{spar}^o &= \left\{ GK_s^o \middle| \forall s\ GK_s^o \in G^o_{\frac{s-\frac{k^h-1}{k-1}}{k^h}},\ h \in \mathbb{Z}_0^+,\ s \in \mathbb{Z}_0^+ \right\}\\ &\quad \forall s\ GK_s^o \text{ is under } R^{i-o} \text{ sauthority,}\\ &\quad \text{where } G_s^o \cap\ G_{spar}^o\\ &= G_s^o \text{ and } \left(G_0^o - G_{spar}^o\right) \cap\ G_s^o =\ \emptyset\,. \end{aligned} \tag{4}$$

Fourth, the definition of a leaf group is a group connecting $\lceil \frac{p}{k} \rceil$ tags from $\left\lceil \frac{(^p/_k)-1}{k-1} \right\rceil$ to $\left\lceil \frac{(^p/_k)-1}{k-1} \right\rceil + \lceil \frac{p}{k} \rceil - 1$ to the same leaf node (5). For example, tags numbered $TID_1^o, TID_2^o\,, and\, TID_3^o$ are connected to the leaf node $G_{lf,1}^o$ of group G_9^o

$$\begin{aligned} G_{lf,m}^o &= \{TID_l^o | \forall l TID_l^o \in G_{lf,m}^o (m-1)k\\ &\quad + 1 \leq l \leq mk, 1 \leq m \leq \left\lceil \frac{p}{k} \right\rceil \}\,. \end{aligned} \tag{5}$$



Tag ID	Owner	Share Key	Group Key
TID_7^o	R^{i-o}	TK_7^o	GK_6^o, GK_1^o, GK_0^o
TID_8^o	R^{i-o}	TK_8^o	GK_6^o, GK_1^o, GK_0^o
TID_9^o	R^{i-o}	TK_9^o	GK_6^o, GK_1^o, GK_0^o

Tag ID	Owner	Share Key	Group Key
TID_4^n	R^{j-n}	TK_4^n	GK_2^n, GK_0^n
TID_5^n	R^{j-n}	TK_5^n	GK_2^n, GK_0^n
TID_6^n	R^{j-n}	TK_6^n	GK_2^n, GK_0^n

Fig. 5. Example of tag group numbering for transferred tags.

Fig. 5 presents an example of transferring tags T_7^o, T_8^o, and T_9^o owned by reader R_1^i in the ownership group R^{i-o} under D^i's authority to the ownership group R^{j-n} under D^j's authority. First, we assume that when transferring ownership, all owner readers and receiver readers can communicate with each other. When the original owner, reader R_1^i, initiates ownership transfer, D^i performs a database lookup and notifies all the owner readers in the ownership group R^{i-o} to which the tags belong. If most owners agree to transfer ownership, server D^i uses the key shared by D^i and the tag to generate an ownership transfer message to D^j. After D^j confirms, D^i will transfer a message to update the key in the multiowner reader set R^{j-n} and the tag group G_0^o to simultaneously update keys and avoid the WP. The authorizing

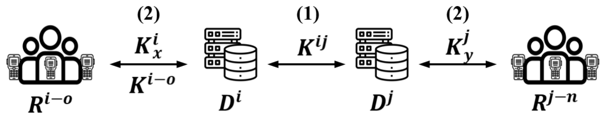


Fig. 6. Keys shared between participants.

backend server D^j has a key tree; thus, the right side of Fig. 5 indicates that on the transfer of a key tree tag to the multiowner reader group R^{j-n}, the tag is inserted on the far right of the tag group G_2^n. The tag names are $T_4{}^n$, $T_5{}^n$, and $T_6{}^n$. The tag number in R^{j-n} may not be the same as that in R^{i-o}. This example reveals how each server can provide its controlled tags with unique numbers; thus, number overlaps between servers will not occur. The process is described in detail in Section IV.

IV. Multiowner Multitag Ownership Transfer Protocol

In this section, we propose a new RFID protocol for transferring the ownership of some or all of a group of tags with multiple owners. The protocol contains three stages: initial, obtaining group tag transfer licensing, and transferring group tag ownership. During the initial stage, each participant must securely obtain the shared key. Next, group tag transfer licensing is obtained. The old owner first confirms that most owners agree to transfer ownership and collect the tag information. Next, group tag ownership is transferred. The tag information is verified, and the key and the secret value for the server assuming ownership (the receiving server) and for the tag are updated.

A. *Initial Stage*

Before implementing the protocol, each participant must securely obtain a shared key (see Fig. 6).

1) Servers D^i and D^j share the key K^{ij}.
2) The server that initially owns the tags (the sending server), D^i, and its reader set, $R^{i-o} = \{R_1^i, R_2^i, \ldots, R_m^i\}$, share a key K^{i-o}. D^i and each reader in the set share keys K_1^i, K_2^i, …, K_m^i. The receiving server D^j and each reader in the reader set $R^{j-n} = \{R_1^j, R_2^j, \ldots, R_n^j\}$ share keys K_1^j, K_2^j, …, K_n^j.

B. *Obtaining Group Tag Transfer Licensing*

Without loss of generality, Fig. 7 shows that any reader R_1^i of the ownership group R^{i-o} may initiate ownership transfer, transferring part of the tag groups G_s^o to the reader group R^{j-n}. Reader R_1^i uses the key K_1^i it shares with the server D^i to encrypt the ownership transfer request $OT_{request}$, the server identification code of the receiving server DID^j, the multiowner reader set of the object transferred into R^{j-n}, the tag group to be transferred G_s^o, and the random number N_r used to generate message M_1 sent to server D^i to bind the objects to be transferred.

When D^i receives the message M_1 and uses the key K_1^i shared with the reader R_1^i to decrypt M_1, it verifies that the message is from the reader R_1^i, and $OT_{request}$ in the message indicates that reader R_1^i is about to initiate ownership transfer. Because the reader R_1^i belongs to the owner reader group R^{i-o}, it uses message M_1 and the secret hash value $H(S^{i-o})$ shared by the server and the tag to use the group key K^{i-o} shared by server D^i and the reader set R^{i-o} to encrypt and to generate message M_2. M_2 is broadcast to readers in the reader set R^{i-o} to ask each owner whether they agree to transfer the ownership of the tag group G_s^o.

When R_x^i, a reader in R^{i-o}, uses the group key K^{i-o} to decrypt M_2 and agrees to transfer its ownership, it uses its private signing key RSK_x^i to sign the hash value $H(S^{i-o})$ of M_2. The signed MPS_x and the random number N_r in M_2 are be encrypted using the key K_x^i shared by both parties to form message M_3, which is sent to server D^i.

When D^i receives M_3 from any reader R_x^i in the reader set R^{i-o}, it uses the key it shares with that reader K_x^i to decrypt the message. If the message includes the random number N_r sent to the reader, as the signature example in (3, *m*) in Fig. 8, readers R_1^i, R_2^i, and R_3^i return part of the signatures MPS_1, MPS_2, and MPS_3, respectively. Then, a threshold signature is generated [43]. D^i uses the public group key UGK^i of the reader set R^{i-o} to verify whether the number of owners agreeing to transfer ownership surpasses the threshold value. D^i uses the group key GK_s^o to encrypt the confirm message $OT_{Confirm}$, the group tag number that is to be transferred, GID_s^o, and a random number N_r. If the threshold is not met, $OT_{Confirm}$ is replaced with OT_{Fail}, and together with GID_s^o and N_r, message M_5 is generated. This process inhibits guessing attacks (GAs) by attackers pretending to be readers. Finally, M_5 is encrypted using the key K_1^i shared by the owner reader R_1^i to produce message M_4, which is sent to owner reader R_1^i.

After the owner reader R_1^i receives M_4, it uses the shared key K_1^i to obtain message M_5, and M_5 is broadcast to the group tag G_s^o. After G_s^o receives M_5, each tag TID_v^o uses the group key GK_s^o for decryption and uses GID_s^o to confirm that they have correctly received message and checked whether it includes $OT_{Confirm}$. If it does, then key TK_v^o shared with server D^i is used to encrypt tag number TID_v^o and random number N_r. After adding the group number GID_s^o, message MT_v is generated and sent to reader R_1^i. MT_v is encrypted using the key K_1^i shared with D^i to generate message M_7 to send to the managing server D^i of R_1^i.

C. *Transfer the Ownership of a Tag Group*

Fig. 9 shows when D^i receives M_7, it uses the key K_1^i shared with the owner reader R_1^i to decrypt M_7 and obtain MT_v. D^i uses GID_s^o to confirm the receiving group and to confirm that R^{i-o} has ownership of the tag group G_s^o and to confirm that it has collected all MT_v returned by each tag in the group. Then, the secret value TK_v^o shared by D^i and each tag is used to decrypt all MT_v message to compare and verify each tag identification code TID_v^o and random number N_r. If so, server D^i then uses key K^{ij} shared with the receiving server D^j to encrypt the ownership transfer request $OT_{request}$, the server identification code of the transfer object DID^j, the multiowner reader set of the transfer object R^{j-n}, group tag G_s^o, and random number N_r to create message M_8 to send to D^j. D^j prepares to update the shared key on the tags.

When D^j receives message M_8, it uses the shared key K^{ij} to decrypt the message. It first verifies whether DID^j is the same as that received and verifies whether R^{j-n} is under DID^j's authority. If the verification is successful, D^j uses key K^{ij}

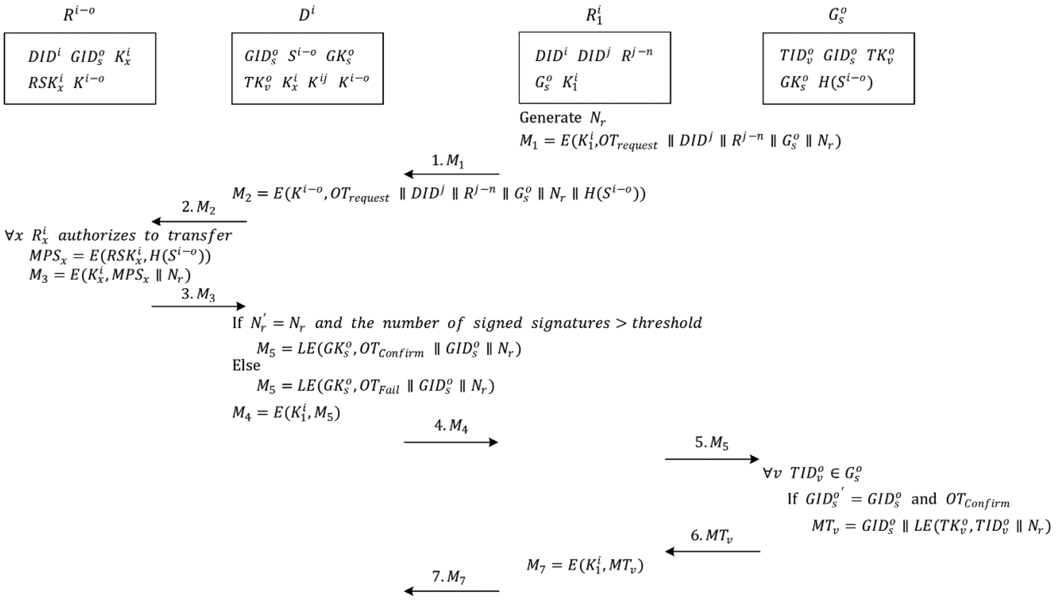

R^{i-o}
DID^i GID_s^o K_x^i
RSK_x^i K^{i-o}
D^i
GID_s^o S^{i-o} GK_s^o
TK_v^o K_x^i K^{ij} K^{i-o}
R_1^i
DID^i DID^j R^{j-n}
G_s^o K_1^i
G_s^o
TID_v^o GID_s^o TK_v^o
GK_s^o $H(S^{i-o})$
Generate N_r
$M_1 = E(K_1^i, OT_{request} \parallel DID^j \parallel R^{j-n} \parallel G_s^o \parallel N_r)$
1. M_1
$M_2 = E(K^{i-o}, OT_{request} \parallel DID^j \parallel R^{j-n} \parallel G_s^o \parallel N_r \parallel H(S^{i-o}))$
2. M_2
$\forall x$ R_x^i authorizes to transfer
$MPS_x = E(RSK_x^i, H(S^{i-o}))$
$M_3 = E(K_x^i, MPS_x \parallel N_r)$
3. M_3
If $N_r' = N_r$ and the number of signed signatures > threshold
$M_5 = LE(GK_s^o, OT_{Confirm} \parallel GID_s^o \parallel N_r)$
Else
$M_5 = LE(GK_s^o, OT_{Fail} \parallel GID_s^o \parallel N_r)$
$M_4 = E(K_1^i, M_5)$
4. M_4
5. M_5
$\forall v$ $TID_v^o \in G_s^o$
If $GID_s^{o'} = GID_s^o$ and $OT_{Confirm}$
$MT_v = GID_s^o \parallel LE(TK_v^o, TID_v^o \parallel N_r)$
6. MT_v
$M_7 = E(K_1^i, MT_v)$
7. M_7

Fig. 7. Obtaining owner consensus to initiate tag ownership transfer.

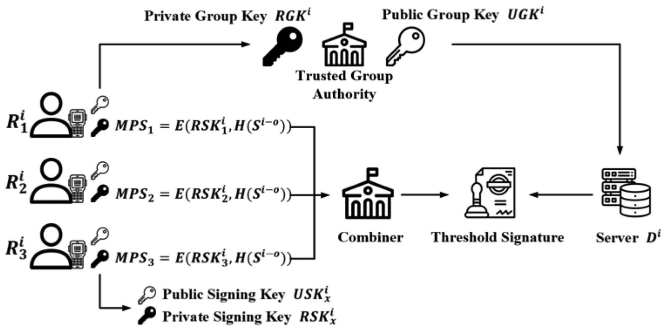

Private Group Key RGK^i
Public Group Key UGK^i
Trusted Group Authority
R_1^i
$MPS_1 = E(RSK_1^i, H(S^{i-o}))$
R_2^i
$MPS_2 = E(RSK_2^i, H(S^{i-o}))$
R_3^i
$MPS_3 = E(RSK_3^i, H(S^{i-o}))$
Combiner
Threshold Signature
Server D^i
Public Signing Key USK_x^i
Private Signing Key RSK_x^i

Fig. 8. Example of R^{i-o} partial signature licensing [42].

shared by both servers to encrypt D^j's confirmation message $OT_{Confirm}$ and the hashed secret value $H(S^{j-n})$ to create message M_9 to return to D^i.

When D^i uses the shared key K^{ij} to decrypt M_9, it verifies the random number N_r to avoid RAs. It also checks for $OT_{Confirm}$. If so, a new group key GK^n is generated and the Chinese remainder theorem (6) is used to calculate message M_{10} to send the new group key GK^n to each tag

$$M_{10} \equiv \sum_{s=1}^{p} ((GK^n \oplus TK_s^o) * m_s^o * m'^o_s) \pmod{M},$$

where $M = \prod_{s=1}^{p} TK_s^0$

$$m_s^o = \frac{M}{TK_s^o}, m_s^o * m'^o_s \equiv 1 \pmod{TK_s^o}. \qquad (6)$$

Then, the key K^{ij} shared by both servers is used to encrypt the reader set R^{j-n}, the group number GID_s^o, and the new group key GK^i into message M_{11} to send to the receiving server D^j. D^i uses GK_s^o to encrypt M_{10}, the hashed secret value of the original owner $H(S^{i-o})$, the hashed secret value of the new owner $H(S^{j-n})$, and a random variable N_r with GID_s^o to produce message M_{13}. To avoid the WP, D^i simultaneously sends message M_{11} to the receiving server D^j and M_{12} to the owner reader R_1^i. If the random number is incorrect or if it receives a failure message OT_{Fail}, another random number N_r'' is generated to replace the random number in M_{13}. This procedure prevents attackers from conducting GAs.

When D^j receives the ownership transfer message M_{11} from D^i, it uses the shared key K^{ij} to decrypt M_{11}. Server D^j first verifies whether R^{j-n} is under the authority of D^j. Then, after D^j obtains the group tag G_s^o, it transfers the tag to reader R^{j-n} and obtains group key GK^n. It also uses the group key encryption tag identification TID_v^n to add each tag TID_v^n, the new shared key of the backend server TK_v^n, and the group key GK_s^n into the corresponding cells in the database TID_v^n of D^j, as shown in the top of Fig. 5.

After the owner reader R_1^i receives message M_{12}, the key K_1^i shared with D^i is used to decrypt, and then M_{13} is directly broadcast to the tags. After the tags receive M_{13}, they first use GID_s^o to confirm that the message includes the group tag G_s^o, and then they use the group key GK_s^o to decrypt the message and verify the random number N_r and the hashed secret value $H(S^{i-o})$. After verification, the tags conduct a modulus operation on M_{10} and their own keys TK_v^o. Next, they conduct exclusive or computing (XOR) with TK_v^o to obtain the new group key GK^n. Subsequently, the hashed secret value of the original owner, $H(S^{i-o})$, is updated to the hashed secret value of the new owner, $H(S^{j-n})$. The new group key GK^n is used to encrypt tag identification code TID_v^n to replace the key TK_v^n shared by the tag and the receiving server D^j. Finally, the group key is updated in GK_s^n.

D. Transferring Multiple Group Tags Simultaneously

If the transferred tags belong to the same group, transfer is only conducted once. However, if the number of tags to be

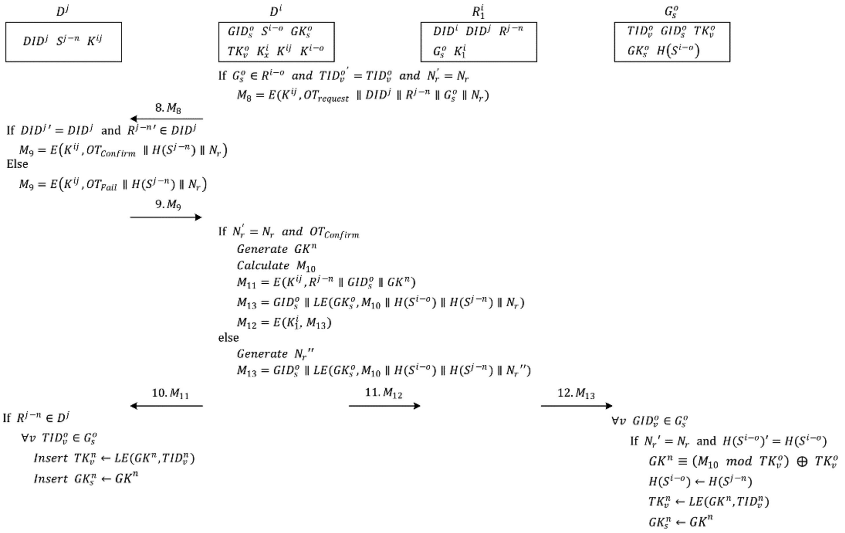

D^j
DID^j S^{j-n} K^{ij}
D^i
GID_s^o S^{i-o} GK_s^o
TK_v^o K_x^i K^{ij} K^{i-o}
R_1^i
DID^i DID^j R^{j-n}
G_s^o K_1^i
G_s^o
TID_v^o GID_s^o TK_v^o
GK_s^o $H(S^{i-o})$
If $G_s^o \in R^{i-o}$ and $TID_v^{o'} = TID_v^o$ and $N_r' = N_r$
$M_8 = E(K^{ij}, OT_{request} \parallel DID^j \parallel R^{j-n} \parallel G_s^o \parallel N_r)$
8. M_8
If $DID^{j\prime} = DID^j$ and $R^{j-n\prime} \in DID^j$
$M_9 = E(K^{ij}, OT_{Confirm} \parallel H(S^{j-n}) \parallel N_r)$
Else
$M_9 = E(K^{ij}, OT_{Fail} \parallel H(S^{j-n}) \parallel N_r)$
9. M_9
If $N_r' = N_r$ and $OT_{Confirm}$
Generate GK^n
Calculate M_{10}
$M_{11} = E(K^{ij}, R^{j-n} \parallel GID_s^o \parallel GK^n)$
$M_{13} = GID_s^o \parallel LE(GK_s^o, M_{10} \parallel H(S^{i-o}) \parallel H(S^{j-n}) \parallel N_r)$
$M_{12} = E(K_1^i, M_{13})$
else
Generate N_r''
$M_{13} = GID_s^o \parallel LE(GK_s^o, M_{10} \parallel H(S^{i-o}) \parallel H(S^{j-n}) \parallel N_r'')$
10. M_{11}
11. M_{12}
12. M_{13}
If $R^{j-n} \in D^j$
$\forall v$ $TID_v^o \in G_s^o$
Insert $TK_v^n \leftarrow LE(GK^n, TID_v^n)$
Insert $GK_s^n \leftarrow GK^n$
$\forall v$ $GID_v^o \in G_s^o$
If $N_r' = N_r$ and $H(S^{i-o})' = H(S^{i-o})$
$GK^n \equiv (M_{10} \bmod TK_v^o) \oplus TK_v^o$
$H(S^{i-o}) \leftarrow H(S^{j-n})$
$TK_v^n \leftarrow LE(GK^n, TID_v^n)$
$GK_s^n \leftarrow GK^n$

Fig. 9. Verifying tags and transferring ownership.

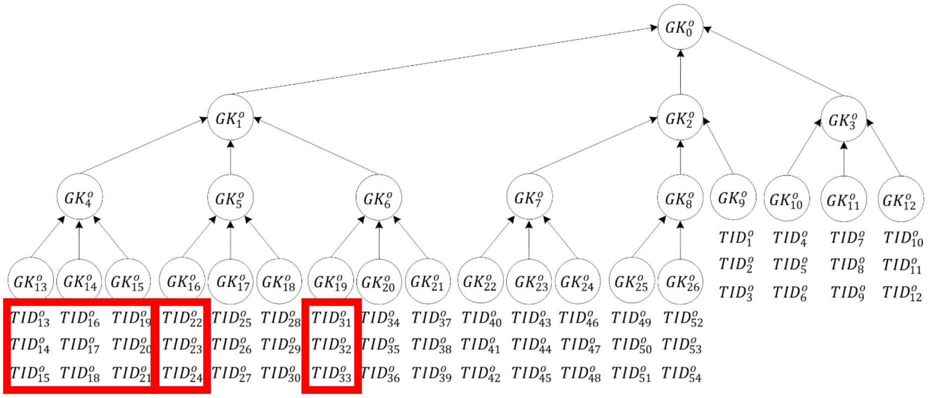

GK_0^o
GK_1^o GK_2^o GK_3^o
GK_4^o GK_5^o GK_6^o GK_7^o GK_8^o GK_9^o GK_{10}^o GK_{11}^o GK_{12}^o
TID_1^o TID_2^o TID_3^o
TID_4^o TID_5^o TID_6^o
TID_7^o TID_8^o TID_9^o
TID_{10}^o TID_{11}^o TID_{12}^o
GK_{13}^o GK_{14}^o GK_{15}^o GK_{16}^o GK_{17}^o GK_{18}^o GK_{19}^o GK_{20}^o GK_{21}^o GK_{22}^o GK_{23}^o GK_{24}^o GK_{25}^o GK_{26}^o
TID_{13}^o TID_{16}^o TID_{19}^o TID_{22}^o TID_{25}^o TID_{28}^o TID_{31}^o TID_{34}^o TID_{37}^o TID_{40}^o TID_{43}^o TID_{46}^o TID_{49}^o TID_{52}^o
TID_{14}^o TID_{17}^o TID_{20}^o TID_{23}^o TID_{26}^o TID_{29}^o TID_{32}^o TID_{35}^o TID_{38}^o TID_{41}^o TID_{44}^o TID_{47}^o TID_{50}^o TID_{53}^o
TID_{15}^o TID_{18}^o TID_{21}^o TID_{24}^o TID_{27}^o TID_{30}^o TID_{33}^o TID_{36}^o TID_{39}^o TID_{42}^o TID_{45}^o TID_{48}^o TID_{51}^o TID_{54}^o

Fig. 10. Example of transferring tags in multiple groups.

transferred does not equal to k^n, then multiple transfers are required. Thus, to transfer v tags, the ownership transfer protocol must be executed $\sum_{i=0}^{m} a_i$ times to complete the ownership transfer of all tags, where *m* is the number of *k*-ary key tree, and a_i is the total elements of the *i*th tree, where $a_i = \frac{v - \sum_{j=i+1}^{m} a_j k^j}{k^i}$. The total number of tags $v = \sum_{i=0}^{m} a_i$.

As presented in Fig. 10, the protocol must be executed multiple times to transfer $TID_{13}^o - TID_{21}^o, TID_{22}^o - TID_{24}^o$, and $TID_{31}^o - TID_{33}^o$, because they belong to different groups; the ownership transfer of group tags G_4^o, G_{16}^o, and G_{19}^o must be performed three times. The owner reader initiating the ownership transfer must broadcast the ownership transfer request of group numbers G_4^o, G_{16}^o, and G_{19}^o to the tags. Thus, implementing the proposed protocol three times can simultaneously conduct group tag ownership transfer.

E. Update Groups and Balance Key Tree

After the proposed ownership transfer protocol is used to update the shared key of each tag and the receiving server and the tag group key, the group tags join the tag group under the authority of the receiving group. When tags join or leave a group, we must add or delete the group key because if the key tree is imbalanced, transfer efficiency is reduced. In the worst case, a tag must store p group keys. Thus, we can use the balanced tree management protocol proposed by Ng et al. [44] to solve the key tree imbalance problem following tags joining or leaving a tree. Moreover, we can use the method proposed by Xu and Huang [45] using maximum distance separable codes, to update group keys. The key of a child group can use the maximum distance separable matrix to calculate an update message and broadcast it to the tag groups. Tags owning a child group key can use the received update message to calculate the new parent group key. Moreover, they proved that using the key tree to build 3-ary trees results in the fewest calculations and optimal efficiency. We can use these methods to update the group communication key and to balance the key tree.

V. Security Analysis

In the proposed method, we have three different communications channels: 1) between backend servers, 2) between backend servers and mobile readers, and 3) between mobile readers and tags. It is easy to establish secure communication in the first two channels by using modern cryptography tools. Due to the limitation of RFID hardware, many cryptographic models of security fail to express important features of RFID systems [32], therefore the third channel is not secure; thus, this section discusses common threats during ownership transfer, such as secret disclosure, replay, man-in-the-middle, tracking, desynchronization, tag/reader impersonation, windowing, dishonest original owners, and FS and BS.

A. Prevent SDA

The protocol must prevent attackers from obtaining sensitive information from messages exchanged by participants. In our protocol, preshared symmetric key encryption was used. The attacker cannot obtain the preshared key, nor can they read the encrypted messages.

B. Prevent RAs

Attackers may eavesdrop on messages and store them. Therefore, protocols must prevent attackers from replacing messages with previously stored messages. In our protocol, a random number N_r is generated and encrypted with the message during each stage of the communication process. Thus, messages in each ownership transfer are unique and cannot be replayed.

C. Prevent Tag Tracking Attacks (TAs)

A protocol must prevent attackers from tracking the locations of tags. Even for the same tag, the messages differ due to the random number N_r and changed tag keys. Thus, attackers cannot analyze the relationship between messages by obtaining several messages, nor can they decrypt the message content to obtain the tag identification code TID_v^o. Thus, they cannot track tag locations.

D. Prevent DAs

A protocol must prevent attackers from denying key updating or causing message loss resulting in asynchronous keys and unreadable tags. Because messages are encrypted using a key and N_r, we only consider the situations in which messages are blocked by an attacker or are lost during transmissions. Until the key is updated, tags are still owned by the sending server. During protocol execution, if any message is lost or blocked, the process can be restarted. The backend server stores the initial key before the final tag key update; thus, if a tag does not update its key, the server can still use the previous key to communicate with the server.

E. Prevent Tag/Reader IAs

The protocol can prevent attackers from impersonating a tag or reader to gain ownership. To impersonate a tag, attackers must obtain $TID_v^o,\ GID_s^o,\ TK_v^o,\ GK_s^o,\ and\ H(S^{i-o})$. However, attackers can only gain GID_s^o from messages. Keys $TK_v^o\ and\ GK_s^o$ are transmitted to the tag securely during the registration stage. Because messages transmitted during this process are encrypted by the key, the attacker cannot decrypt the message and thus cannot impersonate the tag. To impersonate a reader, attackers must obtain $DID^i,\ DID^j,\ R^{j-n},\ G_s^o,\ and\ K_1^i$. However, these were also all securely transmitted to the reader during the registration stage; thus, attackers cannot impersonate readers.

F. Prevent MitMs

The protocol must prevent attackers from intercepting messages, editing their content, and resending them. Because all messages are encrypted with the random number N_r, which is made at the beginning during communication, attackers cannot impersonate tags or readers to modify the message. Attackers also cannot use RAs. Thus, they cannot impersonate tags or readers to conduct MitMs.

G. Prevent WP

A protocol must avoid situations in which both the old and new owners simultaneously have ownership of a tag. During the final key update, the sending server simultaneously transmits the key to the receiving server and the tag, and updates the secret value, the tag, the key shared by the servers, and the group key of the tag. After this update, the old owner no longer has ownership. The old owner or attackers also cannot replay the key update message from the previous stage to update the key on the tag. Thus, we can avoid the WP.

H. Reduce GAs

Although messages are encrypted, communications between readers and tags use wireless communication; thus, attackers can infer that the transfer has failed if no message is transmitted. Otherwise, the transfer was successful. An attacker can use this information to conduct a GA. Message M_{13} is an example of how the protocol avoids this attack. If the verification fails, only the random number is changed, and a message of the same length is transmitted. Thus, attackers are prevented from guessing the transmitted content. However, an attacker can attack successfully in two situations.

First, the attacker can use brute force to guess the final updated key message M_{13} to cause the key of the tag and the server to be asynchronous. If the message after encryption is d bits, the probability of guessing the correct message is $\frac{1}{2^{2d}}$.

Second, an attacker may have eavesdropped on all steps, collected all messages transmitted between tags and readers, and intercepted the last updated key message M_{13}. Next time the reader generates the same random number N_r, an attacker can replay an intercepted message to cause the key of the tag and the server unsynchronized. The probability of this occurring is analyzed using the birthday problem. The probability of success is approximately $1 - e^{-\frac{u^2}{2^{d+1}}}$, where u is the number of attacker attempts. If the random number is 32 b, the attacker must intercept 9.3×10^3 messages to achieve a success rate of 1%.

I. Backward Security

The key shared by the tag and the server is not directly sent to the next owner. Instead, a randomly generated group key and tag identification code are encrypted to generate a new shared key. Thus, the new owner cannot use the new shared key to decrypt the content in the tag about previous transactions.

TABLE II
COMPARISON OF THE SECURITY OF OWNERSHIP TRANSFER

	Kapoor et al.[21]	Sundaresan et al[22]	Luo & Yang[23]	TBGODP+[37]	Our protocol
SDA	O	O	O	O	O
RA	X	X	O	O	O
MitM	X	O	O	O	O
TA	X	X	O	O	O
DA	X	X	O	O	O
IA	O	O	O	O	O
WP	X	O	O	O	O
GA	X	X	O	O	O
BS	O	X	O	O	O
FS	O	X	O	O	O
OA	X	X	X	O	O
GOT	X	△	O	O	O
OPT	X	X	O	X	O
ATT	X	X	O	O	O
POA	X	X	△	X	O

TABLE III
CALCULATION TIME REQUIRED FOR m OLD OWNERS TO TRANSFER p TAGS TO n NEW OWNERS

Protocol	Device	Calculation amount
Kapoor et al.[21]	Tag	$(pn+p)T_{LE}+(pn+p)T_H+2pT_{PRNG}$
	Reader	$(pm+pn)T_E+pnT_{LE}+2pnT_H+pnT_{PRNG}$
	Server	$(pm+pn)T_E+pT_{LE}+(pn+p)T_H+3pT_{PRN}$
Sundaresan et al.[22]	Tag	$(5p+3n)T_{PRNG}$
	Reader	$(6n+2p)T_{PRNG}$
	Server	$(9n+4p+2pn+4)T_{PRNG}$
Luo and Yang[23]	Tag	$6pT_{LE}$
	Reader	$(2m+p+3)T_E+T_{PRNG}+mT_{SIG}$
	Server	$(2m+p+9)T_E+(3p+3)T_{LE}+T_{PRNG}+T_{VE}$
Our protocol	Tag	$4pT_{LE}$
	Reader	$(2m+3)T_E+T_{PRNG}+mT_{SIG}$
	Server	$(m+9)T_E+(2p+2)T_{LE}+2T_H+T_{PRNG}+T_{VE}$

J. Forward Security

In the final key update, the sending server simultaneously transmits the new key to the receiving server and the tag. Readers of the sending server are only responsible for transmitting the message. Even if a sending server's reader obtains this message, it cannot decrypt the content because it lacks the new shared key.

K. Dishonest Original Owner

A previous owner may be an attacker; after ownership is transferred, they immediately carry out an attack to regain ownership. In our protocol, we update the secret value of the tag, the shared key of the servers, and the group key after ownership transfer. Moreover, we ensured BS and avoided the WP; thus, after the new owner updates the key and calculates a new secret value and key, the old owner cannot regain ownership.

We compared our protocol and other ownership transfer methods. We compared SDA, RA, MitM, TA, DA, IA, the WP, GA, FS, BS, dishonest original OA, GOT, transferring partial tags (OPT), assigning transfer target (ATT), and partial owner agreement (POA). The symbol O indicates that a protocol is secure for the attack, X indicates that the protocol is vulnerable to the attack, and △ means that the protocol is not completely secure.

Table II reveals that other protocols are vulnerable to some attacks. For example, the protocol of Kapoor et al. is vulnerable to WP and DA because the attacker can intercept the key update message. It is also vulnerable to RAs and TAs [38]. In the protocol of Sundaresan et al., an attacker can decrypt messages to obtain secret values and can replay messages; thus, the method is vulnerable to RA and TA and lacks FS [35]. Moreover, although that protocol could simultaneously transfer multiple tags, it allows ownership transfer without the consent of a majority of owners. Thus, this protocol is incomplete in terms of group ownership. Although the protocol of Luo and Yang can prevent most attacks, it is vulnerable to a dishonest original owner [29]. TBGODP$^+$ proposed by Moazami and Safkhani [37] cannot provide partial ownership transfer. Although they claim that they use threshold signature to obtain the agreement of most owners, their protocol does not include a comprehensive partial signature and inspection process. Thus, only our proposed protocol is secure from all existing attacks. Specifically, our protocol is superior to others in that only our protocol is secure for a dishonest old owner. Moreover, we use Proverif [48], a cryptographic protocol verifier in the formal model to prove the correctness of our protocol. The result shows that our protocol is secure.

VI. PERFORMANCE ANALYSIS

In this section, we analyze the calculations and the number of messages required for the proposed ownership transfer protocol. To fairly compare our method with other methods, we assumed that there are m old owners and n new owners, and p tags are successfully transferred. T_E, T_{LE}, T_{PRNG}, and T_H indicate the time required for conducting one encrypting or decrypting calculation, for one lightweight encrypting or decrypting calculation, for generating a random number, and for conducting a hash function calculation, respectively. Each time the server generates a key, we conduct one T_{PRNG}. The time required for a reader to generate a partial signature is T_{SIG}, and the time required for the server to generate and verify a signature is T_{VE}. Because the key required for each reader to generate a partial signature is generated before the protocol, the generation time is not included in the calculation time. Compared with the time required for cryptographic calculations, the time required for logical calculations is negligible. Thus, logical calculations were not included in the analysis.

Table III reveals that if the number of tags is large, the required calculation time of Kapoor et al. is excessive. The protocol proposed by Sundaresan et al. only involves lightweight computational pseudo-random number generator (PRNG); thus, its reader and server require few calculations. However, the protocol has several security problems. The protocol also does not require majority owner agreement before conducting the transfer. Moreover, this protocol can only transfer all tags from the owners all at once; it cannot transfer a subset of the tags. These limitations enable the protocol to have favorable performance for the reader and the server. Still, that protocol owners cannot use a single

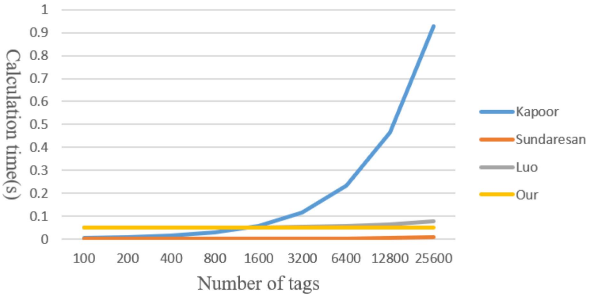


Fig. 11. Calculation amount conducted by readers.

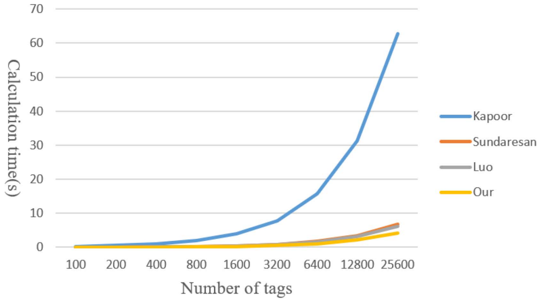


Fig. 12. Calculation amount conducted by servers.

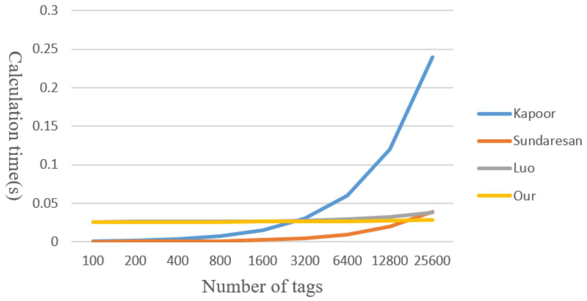


Fig. 13. Calculation amount conducted by tags.

broadcast message to transfer all tags but must use the group key for each tag to generate transfer messages. Thus, the calculation time is determined by the number of owners and tags, affecting the performance. Compared to the method of Luo and Yang, the proposed protocol does not require additional processing or calculations for communicating with a TTP. Moreover, if the server asks owners whether they agree to the ownership transfer, the protocol of Luo and Yang involves using the key shared by the server and each owner to generate individual messages and transmit them to each owner. Our protocol only generates one message and broadcasts it to all owners to reduce the required calculations. Thus, although the proposed method requires partial signing and verification so to obtain majority owner approval, our computational efficiency is still superior to that of Luo and Yang.

As an example, the number of the original owner and the new owners both as 10, the lightweight symmetrical key encryption method for the RFID tags as DES lightweight extension (DESL) with each encryption and decryption requires 144 cycles and the general symmetrical key encryption method as AES-128 with each encryption and decryption requires 1032 cycles [40]. For an RFID tag with a computational speed of 3.55 MHz clock cycle in a second [46], we calculated the clock cycles and obtained the time required for ownership transfer. We used the method of RSA+digital signature algorithm (DSA) to calculate the time required for the threshold signature verification. The time required to generate a partial signature is 5 ms, and that for synthesizing and verifying a signature is 26 ms [47]. Logistics applications typically require numerous objects to be transferred; thus, to analyze these multitag protocols' performance in logistics applications, we increased the number of tags from 100 to 25 600 to observe changes in the calculation times of these protocols.

Fig. 11 presents the computations conducted by readers. The protocol proposed by Kapoor et al. can only transfer one tag at a time; thus, its calculation times increase rapidly as the number of tag increases. The protocol of Luo and Yang cannot use broadcast messages to communicate between servers and readers to obtain partial signatures for the readers. The protocol of Sundaresan et al. has the shortest reader calculation time primarily because that method only used lightweight computational element PRNG, does not conduct extra calculations to gain majority owner approval, and because it does not calculate the partial group key required for transferring a subset of tags. These limitations result in their protocol having several security problems.

Fig. 12 reveals the required computations for servers. The protocol proposed by Kapoor et al. can only transfer one tag at a time; thus, its calculation times increase rapidly as the number of tag increases. The protocol of Sundaresan does not conduct extra calculations to gain majority owner approval, so its calculation time is substantially reduced. However, because the owner cannot use a single broadcast message to transfer all tags but instead must use the group key to generate a transfer message for each tag, if the number of tags exceeds approximately 20 000 the required calculation time exceeds that of our method.

Fig. 13 presents the calculations required by the tags. Because the method of Kapoor et al. requires executing the entire protocol for each transferred tag, the calculation time rapidly increases as the number of tags increases. Because the tag calculations in our protocol are only four lightweight encryption and decryption calculations for each tag (see Table III), which is far lower than other protocols, our protocol requires the least calculation time.

Compared with readers and servers, tags have limited calculation ability; thus, the tag calculation amount is typically a performance bottleneck and should be reduced. Our protocol substantially reduces the calculation burden for tags. Fig. 13 indicates that for any number of tags, the calculation time required by our method is less than that of other methods. For readers, because both our protocol and the protocol of Luo and Yang require owner consent and because the time required for signature and verification is 5000 times greater than for regular encryption [40], [47], both methods require more time than Sundaresan et al. did. However, as presented in Table II, this step is required for security.

In addition to the calculation time, the number of messages required to complete a protocol is a major factor affecting a protocol's performance. Because the network delay time is typically far greater than the calculation time, sending fewer messages substantially affects the performance of the algorithm. Table IV compares the number of messages required by our protocol and by other protocols.

TABLE IV
NUMBER OF MESSAGES REQUIRED FOR m OLD OWNERS TO TRANSFER p TAGS TO n NEW OWNERS

Protocol	Number of messages
Kapoor et al.[21]	$pm + 4pn + 2p$
Sundaresan et al[22]	$9n + 5p + 2pn$
Luo and Yang[23]	$2m + 2p + 8$
Our protocol	$m + p + 10$

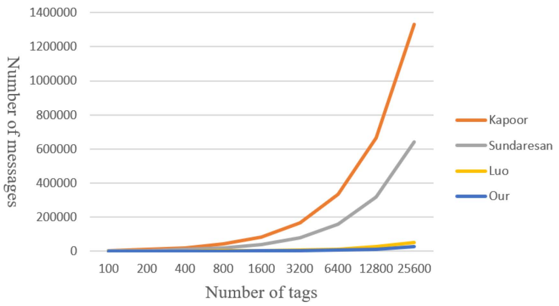


Fig. 14. Number of messages sent.

Table IV reveals that the required number of messages increases rapidly as the number of tags increases in Kapoor et al.'s protocol. For the protocol of Sundaresan et al., the number of messages also increases rapidly as the number of tags and readers increases. In the protocol of Luo and Yang, the server cannot use a single message to ask whether each reader agrees to transfer ownership. Moreover, when collecting tag information, each tag must send a message to the server via the reader; thus, the protocol generates a higher number of messages than our proposed protocol.

Fig. 14 presents the correlation between tag number and the number of messages. If the number of tags is small, the numbers of messages sent in the protocol do not vary substantially. If the number of tags exceeds 400, a clear difference can be observed between protocols. Both the protocols of Kapoor et al. and Sundaresan et al. require quickly increasing numbers of messages as the number of tags increases. Also, the number of messages required by our method is approximately half of that of Luo and Yang's protocol.

VII. CONCLUSION

In this article, we proposed a secure RFID ownership transfer protocol with multiple owners and multiple tags. Our proposed protocol is the only protocol that can obtain agreement from a majority of owners before transferring the ownership of a subset of a tag group. Compared with other multiowner multitag ownership transfer methods, our method is the most secure and requires the fewest messages transmitted; thus, it has the highest computational efficiency. The protocol uses partial signature to confirm whether the number of agreeing owners exceeds a threshold value. We proved our protocol can resist the most common attacks during RFID ownership transfer, such as secret disclosure, replay, man-in-the-middle, tracking, desynchronization, tag or reader impersonation, the WP, and GA. We ensured FS and BS and could assign the transfer subject. The method was also not vulnerable to attacks by the previous owner.

To verify the performance of our method, we used experimental analysis to compare our protocol and other protocols in terms of the computational efficiency and the number of messages required for the tags, readers, and servers participating in the protocol for a set number of owners. The experimental results showed that compared to other multiowner multitag transfer methods, the proposed protocol required fewer messages and computations, and thus could have practical applications in the logistics.

REFERENCES

[1] L. Cui, Z. Zhang, N. Gao, Z. Meng, and Z. Li, "Radio frequency identification and sensing techniques and their applications—A review of the state-of-the-art," *Sensors*, vol. 19, no. 18, 2019, Art. no. 4012.

[2] B. Chander and K. Gopalakrishnan, "A secured and lightweight RFID-tag based authentication protocol with privacy-preserving in Telecare medicine information system," *Comput. Commun.*, vol. 191, pp. 425–437, 2022.

[3] EPCglobal, "EPCTM radio-frequency identity protocols," Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz –960 MHz Version 1.1.0, 2006.

[4] D. Molnar, A. Soppera, and D. Wagner, "A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags," in *Selected Areas in Cryptography* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2006, pp. 276–290.

[5] M. Shariq, K. Singh, C. Lal, M. Conti, and T. Khan, "ESRAS: An efficient and secure ultra-lightweight RFID authentication scheme for low-cost tags," *Comput. Netw.*, vol. 217, 2022, Art. no. 109360.

[6] J. Saito, K. Imamoto, and K. Sakurai, "Reassignment scheme of an RFID tag's key for owner transfer," in *Proc. Int. Conf. Embedded Ubiquitous Comput.*, 2005, pp. 1303–1312.

[7] K. Osaka, T. Takagi, K. Yamazaki, and O. Takahashi, "An efficient and secure RFID security method with ownership transfer," in *RFID Security*. Boston, MA, USA: Springer, 2008, pp. 147–176.

[8] H.-B. Chen, W.-B. Lee, Y.-H. Zhao, and Y.-L. Chen, "Enhancement of the RFID security method with ownership transfer," in *Proc. 3rd Int. Conf. Ubiquitous Inf. Manage. Commun.*, 2009, pp. 251–254.

[9] G. Kapoor and S. Piramuthu, "Single RFID tag ownership transfer protocols," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 2, pp. 164–173, Mar. 2012.

[10] M. H. Yang, "Across-authority lightweight ownership transfer protocol," *Electron. Commerce Res. Appl.*, vol. 10, no. 4, pp. 375–383, 2011.

[11] Z. Shen, P. Zeng, Y. Qian, and K.-K. R. Choo, "A secure and practical RFID ownership transfer protocol based on Chebyshev polynomials," *IEEE Access*, vol. 6, pp. 14560–14566, 2018.

[12] G. Yong, Z. Yuan, and H. Lei, "RFID tag ownership transfer protocol using blockchain," *Int. J. Performability Eng.*, vol. 15, no. 9, pp. 2544–2552, 2019.

[13] D. Qingkaun, G. Wenxin, L. Li, R. Xiaolong, and Z. Xiaoqian, "Lightweight RFID ownership transfer protocol based on blockchain," in *Proc. IEEE Globecom Workshops*, 2021, pp. 1–7.

[14] M. Vijayalakshmi, S. M. Shalinie, M. H. Yang, S.-C. Lai, and J.-N. Luo, "A blockchain-based secure radio frequency identification ownership transfer protocol," *Secur. Commun. Netw.*, vol. 2022, 2022, Art. no. 9377818.

[15] V. Cherneva and J. L. Trahan, "TP-OTP: Two-party, ownership transfer protocol for RFID tags based on quadratic residues," in *Proc. IEEE Green Energy Smart Syst. Conf.*, 2021, pp. 1–6.

[16] Y. Zuo, "Changing hands together: A secure group ownership transfer protocol for RFID tags," in *Proc. 43rd Hawaii Int. Conf. Syst. Sci.*, 2010, pp. 1–10.

[17] H. Jannati and A. Falahati, "Cryptanalysis and enhancement of a secure group ownership transfer protocol for RFID tags," in *Global Security, Safety and Sustainability & e-Democracy*. Berlin, Germany: Springer, 2011, pp. 186–193.

[18] M. H. Yang, "Secure multiple group ownership transfer protocol for mobile RFID," *Electron. Commerce Res. Appl.*, vol. 11, no. 4, pp. 361–373, 2012.

[19] C.-C. Lee, C.-T. Li, C.-L. Cheng, and Y.-M. Lai, "A novel group ownership transfer protocol for RFID systems," *Ad Hoc Netw.*, vol. 91, 2019, Art. no. 101873.

[20] K.-Y. Tsai, M. H. Yang, J. N. Luo, and W.-T. Liew, "Novel designated ownership transfer with grouping proof," *Appl. Sci.*, vol. 9, no. 4, 2019, Art. no. 724.

[21] G. Kapoor, W. Zhou, and S. Piramuthu, "Multi-tag and multi-owner RFID ownership transfer in supply chains," *Decis. Support Syst.*, vol. 52, no. 1, pp. 258–270, 2011.

[22] S. Sundaresan, R. Doss, W. Zhou, and S. Piramuthu, "Secure ownership transfer for multi-tag multi-owner passive RFID environment with individual-owner-privacy," *Comput. Commun.*, vol. 55, pp. 112–124, 2015.

[23] J.-N. Luo and M.-H. Yang, "A secure partial RFID ownership transfer protocol with multi-owners," *Sensors*, vol. 20, no. 1, 2020, Art. no. 22.

[24] M. H. Yang and K. P. Xie, "TTP-based group ownership transfer in a mobile RFID environment," *Int. J. Digit. Content Technol. Appl.*, vol. 7, no. 2, pp. 51–69, 2013.

[25] G. Kapoor and S. Piramuthu, "Vulnerabilities in some recently proposed RFID ownership transfer protocols," *IEEE Commun. Lett.*, vol. 14, no. 3, pp. 260–262, Mar. 2010.

[26] P. Jäppinen and H. Hämäläinen, "Enhanced RFID security method with ownership transfer," in *Proc. Int. Conf. Comput. Intell. Secur.*, 2008, vol. 2, pp. 382–385.

[27] E. Taqieddin, H. Al-Dahoud, H. Niu, and J. Sarangapani, "Tag ownership transfer in radio frequency identification systems: A survey of existing protocols and open challenges," *IEEE Access*, vol. 6, pp. 32117–32155, 2018.

[28] M. H. Yang and H. Y. Hu, "Protocol for ownership transfer across authorities: With the ability to assign transfer target," *Secur. Commun. Netw.*, vol. 5, no. 2, pp. 164–177, 2012.

[29] S. F. Aghili and H. Mala, "New authentication/ownership transfer protocol for RFID objects," *J. Inf. Secur. Appl.*, vol. 49, 2019, Art. no. 102401.

[30] F. Moazami and M. Safkhani, "SEOTP: A new secure and efficient ownership transfer protocol based on quadric residue and homomorphic encryption," *Wireless Netw.*, vol. 26, no. 7, pp. 5285–5306, 2020.

[31] D. Zhu, W. Rong, D. Wu, and N. Pang, "Lightweight anonymous RFID group ownership transfer protocol in multi-owner environment," in *Proc. IEEE 19th Int. Conf. High Perform. Comput. Commun/IEEE 15th Int. Conf. Smart City/IEEE 3rd Int. Conf. Data Sci. Syst.*, 2017, pp. 404–411.

[32] A. Juels, "RFID security and privacy: A research survey," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 381–394, Feb. 2006,.

[33] M. H. Yang, J. N. Luo, and S. Y. Lu, "A novel multilayered RFID tagged cargo integrity assurance scheme," *Sensors*, vol. 15, no. 10, pp. 27087–27115, 2015.

[34] C.-C. Lee, S.-D. Chen, C.-T. Li, C.-L. Cheng, and Y.-M. Lai, "Security enhancement on an RFID ownership transfer protocol based on cloud," *Future Gener. Comput. Syst.*, vol. 93, pp. 266–277, Apr. 2019.

[35] C.-C. Lee, C.-T. Li, C.-L. Cheng, Y.-M. Lai, and A. V. Vasilakos, "A novel group ownership delegate protocol for RFID systems," *Inf. Syst. Front.*, vol. 21, no. 5, pp. 1153–1166, 2019.

[36] V. Kumar, M. Ahmad, D. Mishra, S. Kumari, and M. K. Khan, "RSEAP: RFID based secure and efficient authentication protocol for vehicular cloud computing," *Veh. Commun.*, vol. 22, 2020, Art. no. 100213.

[37] F. Moazami and M. Safkhani, "TBGODP+: Improvement of TBGODP, a time bound group ownership delegation protocol," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 6, pp. 3283–3302, 2022.

[38] N. Bagheri, S. F. Aghili, and M. Safkhani, "On the security of two ownership transfer protocols and their improvements," *Int. Arab J. Inf. Technol.*, vol. 15, no. 1, pp. 87–93, 2018.

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

[40] A. Poschmann, G. Leander, K. Schramm, and C. Paar, "New light-weight crypto algorithms for RFID," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2007, pp. 1843–1846.

[41] S. Sallam and B. D. Beheshti, "A survey on lightweight cryptographic algorithms," in *Proc. IEEE Region 10 Conf.*, 2018, pp. 1784–1789.

[42] G. Bleumer, "Threshold signature," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg, Ed. Boston, MA, USA: Springer, 2005, pp. 611–614.

[43] L. Harn, "Group-oriented (t, n) threshold digital signature scheme and digital multisignature," *IEE Proc.—Comput. Digit. Techn.*, vol. 141, no. 5, pp. 307–313, 1994.

[44] W. H. D. Ng, M. Howarth, Z. Sun, and H. Cruickshank, "Dynamic balanced key tree management for secure multicast communications," *IEEE Trans. Comput.*, vol. 56, no. 5, pp. 590–605, May 2007.

[45] L. Xu and C. Huang, "Computation-efficient multicast key distribution," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 5, pp. 577–587, May 2008.

[46] A. S. Man, E. S. Zhang, V. K. N. Lau, C. Y. Tsui, and H. C. Luong, "Low power VLSI design for a RFID passive tag baseband system enhanced with an AES cryptography engine," in *Proc. 1st Annu. RFID Eurasia*, 2007, pp. 1–6.

[47] F. J. Aufa and A. Affandi, "Security system analysis in combination method: RSA encryption and digital signature algorithm," in *Proc. 4th Int. Conf. Sci. Technol.*, 2018, pp. 1–5.

[48] B. Blanchet, "Automatic verification of security protocols in the symbolic model: The verifier proverif," in *Foundations of Security Analysis and Design VII*. Cham, Switzerland: Springer, 2014, pp. 54–87, doi: 10.1007/978-3-319-10082-1_3.

Ming-Hour Yang (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Taoyuan City, Taiwan, in 2001.

His research interests include network security and system security with particular interests on security issues in RFID and NFC security.

Chien-Hung Chen received the master's degree in information and computer engineering from Chung Yuan Christian University, Taoyuan City, Taiwan, in 2021.

Jia-Ning Luo received the Ph.D. degree in computer science of National Chiao Tung University, Taiwan, in 2006.

He is currently an Associate Professor with the Department of Computer Science and Information Engineering, National Defense University, Taoyuan City, Taiwan.

M. Vijayalakshmi received the Ph.D. degree in information and communication engineering from Anna University, India, in 2001.

She is currently a Professor with the Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai, India. Her research interests include network security, digital forensics, and the Internet of Things.

S. Mercy Shalinie received the Ph.D. degree in computer science and engineering from Madurai Kamaraj University, India, in 2001.

She is currently a Professor and Dean of the Thiagarajar College of Engineering, Madurai, India. Her research interests include AI, machine learning, and information security.