1-1-2013

Secure RFID ownership transfer protocols

Nan Li
University of Wollongong, nl864@uowmail.edu.au

Yi Mu
University of Wollongong, ymu@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Vijay Varadharajan
Macquarie University

Secure RFID ownership transfer protocols

Abstract

An RFID tag could change hands many times during its lifetime. In a retail chain, the ownership of the tag is instituted by the supplier who initially owns the tag. In the view of a buyer, the validity of the current tag ownership and the originality of supplier are most important. In typical RFID ownership transfer protocols, the knowledge of the tag's authentication key proves the ownership. However, it is insufficient against an active attacker, since tags are usually lack of tamper-proof protections. Ownership transfer relies on a successful verification of tag's supplier and current ownership. In this paper, we formally define the security model of ownership transfer protocols and propose a secure ownership transfer protocol. In our scheme, current owner provides a new owner with the evidence of transfer and a proof of tag origin. Key management becomes easy in our system, since the one asymmetric verification key of the owner can be used to verify multiple tags that belong to the owner.

Keywords

rfid, protocols, ownership, secure, transfer

Disciplines

Engineering | Science and Technology Studies

Publication Details

Li, N., Mu, Y., Susilo, W. & Varadharajan, V. (2013). Secure RFID ownership transfer protocols. Lecture Notes in Computer Science, 7863 189-203.

Secure RFID Ownership Transfer Protocols*

Nan Li[1], Yi Mu[1], Willy Susilo[1]**, and Vijay Varadharajan[2]

[1] Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Wollongong, Australia
{nl864,ymu,wsusilo}@uow.edu.au
[2] Information and Networked Systems Security Research
Department of Computing, Faculty of Science
Macquarie University, Sydney,Australia
vijay.varadharajan@mq.edu.au

Abstract. An RFID tag could change hands many times during its lifetime. In a retail chain, the ownership of the tag is instituted by the supplier who initially owns the tag. In the view of a buyer, the validity of the current tag ownership and the originality of supplier are most important. In typical RFID ownership transfer protocols, the knowledge of the tag's authentication key proves the ownership. However, it is insufficient against an active attacker, since tags are usually lack of tamper-proof protections. Ownership transfer relies on a successful verification of tag's supplier and current ownership. In this paper, we formally define the security model of ownership transfer protocols and propose a secure ownership transfer protocol. In our scheme, current owner provides a new owner with the evidence of transfer and a proof of tag origin. Key management becomes easy in our system, since the one asymmetric verification key of the owner can be used to verify multiple tags that belong to the owner.

1 Introduction

A basic RFID system comprises three components: RFID reader, RFID tag, and backend database. RFID has exhibited many practical applications such as serving as identity of an object in supply chains, supermarkets and hospitals. A tag attached to a product has a unique identifier stored in its backend database. In practice, a product (with a tag) is owned by a user. Often, the product needs to change hands due to selling or buying. This process is referred to as ownership transfer.

In the lifetime of a tag, its ownership is likely to be transferred from one owner to another. An ownership transfer protocol runs between the current owner and the new owner. Generally speaking, the protocol is considered in two phases,

* This work is supported by the Australian Research Council Discovery Project DP110101951.
** This work is supported by ARC Future Fellowship FT0991397.

namely ownership verification and ownership transfer. A new owner firstly verifies the current ownership of the tag. If the current ownership is confirmed, he can request the ownership transfer. After a successful ownership transfer, the current owner who becomes the previous owner of the tag can no longer access the tag and the new owner who becomes the current owner of the tag can prove the ownership of the tag. According to the current ownership of the tag, a user can be a previous owner, current owner or new owner of the tag.

The security of RFID ownership transfer protocols is considered in threefold: the secure ownership, exclusive ownership and secure ownership transfer [3]. The first two are related to the phase of the ownership verification. Informally, they guarantee that the actual owner always has the ownership of a tag and no others can simultaneously obtain the ownership. The criteria of secure ownership transfer evaluates the phase of the tag ownership transfer. A new user who is unauthorized by the current owner cannot gain the ownership of the tag. A secure ownership transfer protocol should satisfy all these requirements.

The traditional RFID ownership transfer protocols are based on the lightweight symmetric key authentication schemes. The backend server and a tag share a predefined symmetric key and the tag's identity. The tag's ownership is checked by implementing the authentication protocol. However, most (passive) tags are not tamper-resistant, so that adversaries can launch active attacks. It is possible to physically corrupt or clone a tag and obtain the internal state. Once the internal state is leaked, the adversary can control the tag as the real owner. Therefore, it can prove the ownership and even transfer the tag to others.

1.1 Motivation

The aim of this paper is to propose a secure RFID ownership transfer protocol. In most previous RFID ownership transfer protocols, the proof of ownership relies on the knowledge of the tag's authentication key. If the user can provide a valid secret key, the verifier accepts its ownership of the tag. While it is insufficient against the attacker who compromises the tag. In practice, we call the party who currently owns the tag as a seller and the party who receives the ownership as a buyer. The symmetric authentication key shared between the seller and the tag provides no identity information of the seller. Anyone who has the key is able to prove the ownership and transfer it to other parties. It may injure the rights of seller and buyer. As a buyer, it usually concerns the origin of the product and the validity of the seller. He expects to check them during a purchase. The key management in large RFID system is also an issue. A tag normally requires a unique key for proving the ownership. The buyer has to obtain a large number of keys to check ownerships of tags. It not only requires a secure channel in communication, but also hard to maintain the records of these transactions. It would be desirable that one verification key can do the job. With this key, anyone can verify the ownership of tags that belong to the owner.

We look into an RFID system, where a supplier obtains products from a manufacturer. The supplier authorizes the manufacturer, via a warrant, to make specific products. After the products are ready for the supplier, the manufacturer

setups RFID tags and attaches them to products, respectively. When a buyer purchases the product from the supplier, the ownership transfer is required. The buyer checks the information of supplier and the product prior to making a payment. Once the deal is complete, the buyer owns the tag and supplier can no longer claim the ownership. Meanwhile, the seller provides the undeniable transfer proof which includes the information of seller, buyer and tag. The buyer can also resell the product in the future. One aim in this paper is to construct an ownership transfer scheme in this scenario.

Symmetric key based protocols are insufficient to reach a strong security level for ownership transfer protocols. It is a challenge to resist an active attack. We assume that the tag authentication can be done by using a traditional RFID authentication protocol, while we only focus on the ownership transfer protocol. An owner is usually a powerful entity which can perform public key cryptographic algorithms for ownership transfer, which does not rely on the computation power of tag.

1.2 Our Contributions

In this paper, we enhance the security of ownership transfer protocols by considering some strong attacks, such as the replacement attacks. In our model, the ability of an adversary is assumed by allowing more oracle queries. A formal definition of security model is given in this paper. An RFID ownership transfer protocol which is secure against the presented model is proposed. We consider a chain of the ownership transfers. It guarantees the actual ownership even if the internal state of the tag is disclosed. The protocol prevents an unauthorized owner from transferring the ownership to another. In other words, the validity of the current owner is verified during the ownership transfer. As a feature of our protocol, instead of using different authentication key to check the ownership of each tag, a buyer can use the seller's public key to check the all the tags. We analyse the security of proposed ownership transfer protocol and provide a formal security proof.

2 Related Work

Saito, Imamoto and Sakurai [13] introduced an ownership transfer protocol using two approaches. Both provide the privacy and security protection of the current owner and the new owner. One is based on the three-party model and the other is on two-party model. Since the schemes are based on symmetric key cryptographic algorithms, the secret key of tag is pre-shared with the owner. In the three-party model, the second key is shared between the trusted third party (TTP) and the tag. In ownership transfer, the TTP helps the new owner to update the tag's new secret. While the online TTP is required during the ownership transfer. Once the tag is compromised, the shared secret key between the tag and the TTP is also disclosed.

Independently, Molnar, Soppera and Wagner [9] proposed an ownership transfer protocol of RFID tags. The protocol addresses the privacy problems of ownership transfer through the *pseudonym*. The proposed scheme employs a tree based key structure to enable the *time-limited delegation* for temporarily ownership transfer. It is that the current owner can temporarily delegate the ownership of the tag to another party. After a period of time, the ownership is returned to the original owner without the agreement of the delegatee. However, the scheme needs a counter which is in the non-volatile memory to count the number of authentications. A *Trusted Center* (TC) who controls all the secret of tags assists the readers to authenticate the tag. Unfortunately, most trusted third party based ownership transfer protocols [6, 8, 11, 9] suffer from the similar issues as in [13].

Several security properties of ownership transfer protocols were introduced by Ng, Susilo, Mu and Safavi-Naini [10], where they introduced four new properties: tag assurance, current ownership proof, undeniable ownership transfer and owner initiation. The proposed scheme satisfies most security properties of ownership transfer while only some hash calculations are required on the tag. Elkhiyaoui, Blass and Molva [4] presented the problem of issuer verification during the ownership transfer. In this paper, the privacy and security of ownership transfer protocols are formally defined and the proposed scheme achieves the constant time authentication. The scheme prevents the attacker from injecting fake tags in the supply chains. The origin of the tag is verified prior to the transfer. Abyaneh [2] shows that the forward and backward privacy are broken if the attacker was an owner of the tag. Additionally, the definition of the security model does not allow the adversary to rewrite the tag's content. It may be vulnerable against some active attacks.

A scalable authentication protocol which supports the ownership transfer was proposed in [5]. The protocol provides the controlled delegation without using the non-volatile memory to store a counter. The feature of desynchronization engages the protocol runs without the TTP. It employs a table which consists of two hash chains to identify a tag. While, the cost of storage on the server is questionable when the maximum size of the hash chains increased. Meanwhile, it also suffers form the denial-of-service attack.

Deursen, Mauw, Radomirović and Vullers [3] introduced a formal definition of secure ownership transfer in RFID systems. They described two roles: the *tag owner* and the *tag holder*. Basically, both of them can pass the ownership test but only the owner is engaged to transfer the ownership. It was claimed that the tag owner and holder are coincide in the notion of secure ownership. However, the holder of the tag may not be the owner in decentralized systems. Since the security of ownership is based on the authentication of the tag, most symmetric-key ownership transfer protocols [14, 13, 9, 12] assume that the tag is incorruptible. In [10] and [4], a tag is allowed to be compromised. Nevertheless, the content of the tag cannot be rewrite after the adversary disclosed the key.

3 System Model

In this section, we formally define the ownership transfer protocols using the retail chain as an instance.

3.1 Entities

- **Tag** T_i: An object is attached by one tag T. The tag has a small memory which stores the current state s_i of the tag. T_i is a low-cost device which can at most calculate the hash function F.
- **Manufacturer** M_i: The manufacturer is the one who makes the products for suppliers. One manufacturer can cooperate with many different suppliers while the product must be authorized by the specific supplier.
- **Supplier** S_i: The supplier is the one who sales the products to customers. It handles the first ownership transfer of the tag. The supplier authorizes the manufacturer to produce expected number of products meanwhile S provides a unique warrant for each product.
- **Previous Onwer** $O_{(t_i,k-1)}$: The previous owner $O_{(t_i,k-1)}$ is the one who previously owns the tag T_i at the time $k-1$. It provides the proof of transfer $\Sigma_{(t,k-1,k)}$ to the current owner.
- **Current Owner** $O_{(t_i,k)}$: The current owner $O_{(t_i,k)}$ is the one who currently owns the tag T_i at the time k. It maintains a database which stores the states of tags and authenticates tags though a reader R_k. The current owner can prove the current ownership $\sigma_{(t_i,k)}$ of the tag and show the valid transfer obtained from the previous owner. $O_{(t_i,k)}$ is allowed to transfer the current ownership of T_i to the new owner.
- **New Owner** $O_{(t_i,k+1)}$: The new owner $O_{(t_i,k+1)}$ is the one who is a potential owner of the tag T_i. Prior to accepting the ownership of tag T_i, the new owner verifies the tag's supplier S, the previous transfer proof $\Sigma_{(t,k-1,k)}$ and the current ownership $\sigma_{(t_i,k)}$. It provides an evidence of the acceptance once the transfer is completed.

Remark 1. The supplier can be considered as a special owner of tag and the manufacturer is an agent of particular supplier. The previous owner, current owner and new owner are roles which are changeable in different periods of the tag ownership. That means the new owner becomes a current owner or previous owner once he receives or transfers the tag ownership, respectively.

3.2 RFID Ownership Transfer Systems

In our system model, we do not employ the centralized server which is normally a trusted third party. Instead, we adopt the two-party mode that each party maintains an isolated database and readers. A party who engages in the ownership transfer is an owner of a tag. From now on, we refer to an owner as an entity which is supported by RFID readers and a backend database. While one owner has a public/private key pair where the public key is known to anyone. In the

model, we only need the secure communication channel during the authentication key exchange. Since the proposed scheme applies symmetric-key based tag authentication, it is impossible to securely update the key with shared secret [7]. The key update of the protocol should be performed outside the control range of the previous owner.

The ownership transfer system is described in Fig. 1. Different from the previous models, we consider the ownership of the tag as a chain. To handle an ownership transfer, the information of tag's supplier, previous owner, current owner and new owner are all required. Nevertheless, only the current owner needs to provide its secret.

In the model, the ownership transfer stems from the supplier. Let one owner be a level. Level 0 is the supplier of the tag. The manufacturer generates the proof of ownership under the supplier's warrant and stores it on the tag. Anyone who has the supplier's public key can verify the ownership of the product. In this level, the supplier simultaneously plays the role of the previous owner since the product is brand new. Then, it transfers the ownership to a new owner who is in level 1. Owner 1 accepts the ownership from the supplier and takes the role of the current owner. At this time, the supplier transferred the current ownership but remains the role of supplier and previous owner of the tag. Following the process, the ownership of the tag is generally in the k-th level.

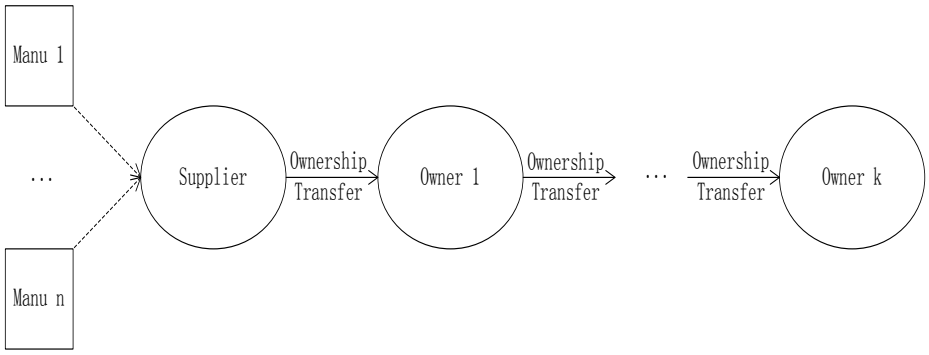

Fig. 1: Ownership transfer systems

A complete ownership transfer process has two phases: ownership verification and ownership transfer. In the ownership verification phase, the buyer checks the supplier of the tag, previous authenticated transfer proof and the validity of current ownership. Only if all the verifications are successful, two owners play the game of the ownership transfer. In the completion of an ownership transfer, the seller outputs a new authenticated transfer proof and the buyer outputs a proof of new ownership.

3.3 Ownership Transfer Protocols

An RFID ownership transfer protocol consists of seven algorithms: system setup (Setup), key generation (KeyGen), tag initiation (TagInit), authentication (Auth), ownership transfer (Transfer), ownership prove (OwnerProve) and ownership verification (OwnerVerify). The seven algorithms in RFID ownership transfer protocols are defined as follows.

- $params \leftarrow \mathsf{Setup}(\lambda)$: Taking as input a security parameter λ, outputs a set of public parameters $params$.
- $(pk, sk) \leftarrow \mathsf{KeyGen}(params)$: Taking as input the system parameters $params$, outputs a pair of public and private keys (pk, sk).
- $(c, \sigma_{(t,0)}) \leftarrow \mathsf{TagInit}(T, pk_s, sk_s, pk_m, sk_m)$: Taking as input a tag T, a pair (pk_s, sk_s) of supplier's public/private keys and a pair (pk_m, sk_m) of manufacturer's public/private keys, outputs the tag's initial state c and ownership proof $\sigma_{(t,0)}$. It runs between a manufacturer and a supplier.
- $Info \leftarrow \mathsf{Auth}(T, O_{(t,k)})$: Taking as input a tag T and the current owner $O_{(t,k)}$, outputs a set of information $Info$ of tag. It runs between the current owner and the tag.
- $\Sigma_{(t,k,k+1)} \leftarrow \mathsf{Transfer}(ID_t, pk_s, pk_{k-1}, pk_k, sk_{k+1}, \Sigma_{(t,k-1,k)})$: Taking as input a tag's identity ID_t, the public key pk_s of supplier, a pair of public/private key (pk_k, sk_k) of current owner and a new owner's public key pk_{k+1}, outputs an authenticated transfer proof $\Sigma_{(t,k,k+1)}$. It is run by the current owner.
- $\sigma_{(t,k)} \leftarrow \mathsf{OwnerProve}(ID_t, sk_k, \Sigma_{(t,k-1,k)}, \sigma_{(t,k-1)})$: Taking as input a tag's identity ID_t, a private key sk_k of current owner and an authenticated transfer proof $\Sigma_{(t,k-1,k)}$, outputs a proof $\sigma_{(t,k)}$ of ownership. It is run by the current owner.
- $\{true, false\} \leftarrow \mathsf{OwnerVerify}(ID_t, pk_s, pk_{k-1}, pk_k, \sigma_{(t,k)})$: Taking as input a tag's identity ID_t, the supplier's verification key pk_s, the previous owner's verification key pk_{k-1} and the current owner's verification key pk_k and a proof $\sigma_{(t,k)}$ of ownership, outputs $true$ if the proof is valid, outputs $false$ otherwise.

Without loss of generality, we describe the Auth algorithm in the protocol. While, it is unnecessary to the security of ownership transfer protocols. In the paper, Auth is assumed to be a privacy-preserving authentication protocol. The interaction of one ownership transfer is depicted as in Fig.2.

4 Proposed Protocol

The mathematical preliminaries and concrete construction of the proposed scheme are presented in the section.

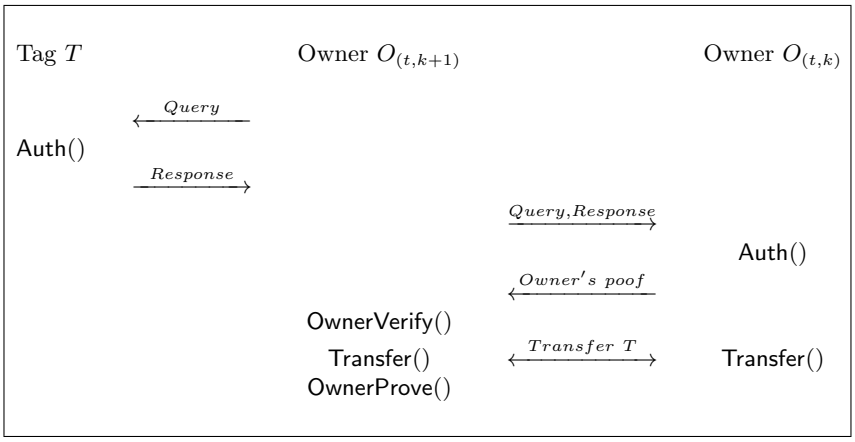

Fig. 2: Ownership transfer protocol.

4.1 Preliminaries

Bilinear Maps Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be three multiplicative cyclic groups of same prime order p. g and h are generators of group $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. The map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a bilinear mapping (pairing) and $(g, h, p, e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ is a bilinear group. Let ψ be a computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ that $\psi(h) = g$. We say it is a symmetric bilinear group if $\mathbb{G} = \mathbb{G}_1 = \mathbb{G}_2$. A bilinear pairing satisfies the properties as follows:

- **Bilinearity**: for all $g \in \mathbb{G}_1$, $h \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p^*$, we have the equation $e(g^a, h^b) = e(u, v)^{ab}$.
- **Non-Degeneracy**: for all $g \in \mathbb{G}_1$, $h \in \mathbb{G}_2$, if g, h are generators respectively, we have $e(g, h) \neq 1$ is a generator of $\mathbb{G}_T$.
- **Efficiency**: There is an efficient algorithm to calculate $e(g, h)$ for all $g \in \mathbb{G}_1$, $h \in \mathbb{G}_2$.

Definition 1 (Computational Diffie-Hellman (CDH) assumption). *Let g be a generator of group $\mathbb{G}_1$. Given a tuple $< g, g^a, g^b >$, where $a, b \in_R \mathbb{Z}_p^*$, the CHD problem is to output $g^{ab} \in \mathbb{G}_1$. We say that the (ϵ, t)-CDH assumption is hold in $\mathbb{G}_1$, if no t-time algorithm $\mathcal{A}$ can solve the CDH problem in $\mathbb{G}_1$ with advantage at least ϵ.*

4.2 Construction

- Setup: Select a symmetric bilinear paring $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, where the order of group $\mathbb{G}$ and $\mathbb{G}_T$ are the same p. Let $g, h \in \mathbb{G}$ be two generators. $H_1 : \mathbb{G} \times \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_2 : \mathbb{G} \times \mathbb{G} \times \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ and $F : \{0,1\}^* \rightarrow \{0,1\}^l$, where l is a security parameter, are collision-resistant cryptographic hash functions. Sets the public parameters $params = (\mathbb{G}, \mathbb{G}_T, g, h, p, e, H)$.

– KeyGen: Randomly chooses $x \in \mathbb{Z}_p^*$ and sets the public/private key pair as $(pk, sk) = (x, g^x)$.
– TagInit: Let the public/private key pairs of a manufacturer M and a supplier respectively be $(pk_m, sk_m) = (x, g^x)$ and $(pk_s, sk_s) = (\alpha, g^\alpha)$. Firstly, the manufacturer and the supplier interacts as in Fig.3. The manufacturer generates an ownership proof σ_0 for the supplier. It randomly chooses an authentication key y from the key space $\mathcal{S}$ and sets the tag state $c = (y, F(\sigma_{(t,0)})$. The supplier is the owner $O_{(t,0)}$.

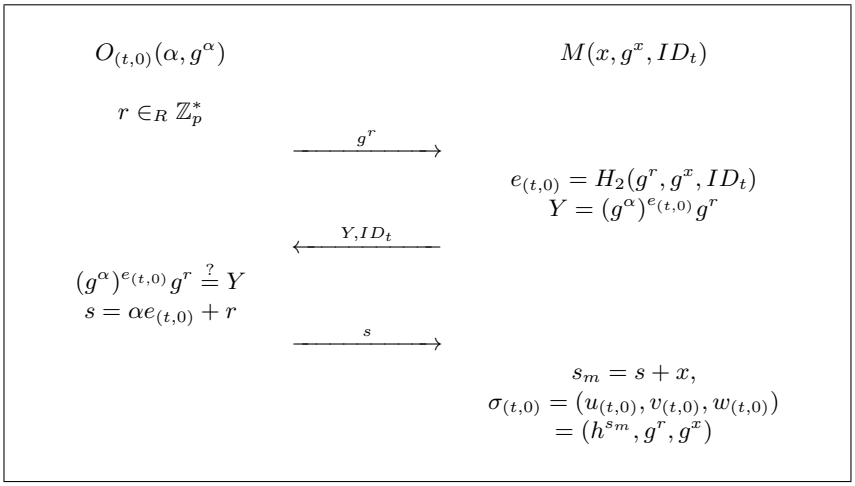

Fig. 3: Ownership initiation

– Auth: It is a general symmetric-key based authentication protocol. The current owner $O_{(t,k)}$ interacts the tag T using a pre-shared symmetric authentication key y. Once the authentication protocol outputs 1, the owner collects the tag's information $info$ which includes the tag's identity ID_t, ownership proof $\sigma_{(t,k)}$, etc.
– Transfer: To transfer the ownership, the current owner $O_{(t,k)}$ interacts with the new owner $O_{(t,k+1)}$. If the current owner is a supplier, it follows the description as in Fig.4. Otherwise, it follows the description as in Fig.5. Assume that the identity of tags and public information of two owners are mutually known.
– OwnerProve: To generate a proof of ownership, the current owner $O_{(t,k)}$ retrieves the proof $\Sigma_{(t,k-1,k)} = (s_{(t,k-1,k)}, R_{(t,k-1,k)})$ of authenticated transfer and the ownership poof $\sigma_{(t,k-1)}$ of owner $O_{(t,k-1)}$. Computes $s_{(t,k)} = s_{(t,k-1,k)} + sk_k$, where sk_k is the private key of $O_{(t,k)}$, and sets the proof $\sigma_{(t,k)} = (u_{(t,k)}, v_{(t,k)}, w_{(t,k)}) = (h^{s_{(t,k)}}, v_{(t,k-1)}, R_{(t,k-1,k)})$. In the case $k = 1$, set $v_{(t,1)} = g^{r_t}$, where g^{r_t} is from Fig.4.

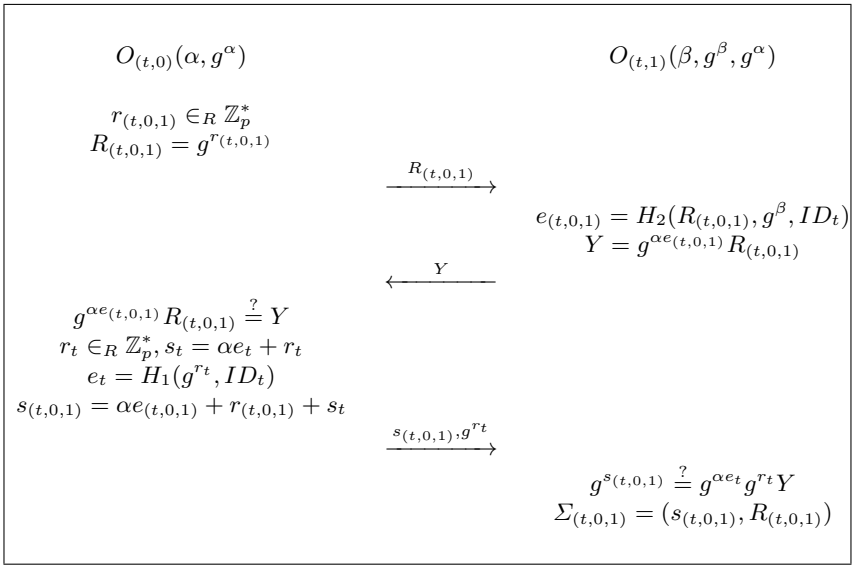

Fig. 4: Transfer from supplier to new owner

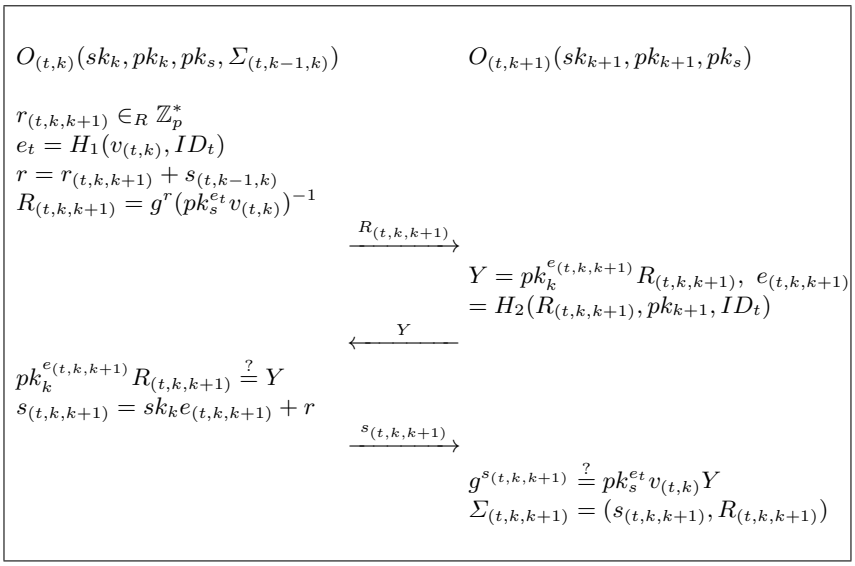

Fig. 5: General transfer from current owner to new owner on level k

– OwnerVerify: On input a proof $\sigma_{(t,k)} = (u_{(t,k)}, v_{(t,k)}, w_{(t,k)})$ of tag T, there are three cases. The verifier checks as follows
  - Case 1 ($k = 0$):

$$e_{(t,0)} = H_2(v_{(t,0)}, w_{(t,0)}, ID_t), \quad e(g, u_{(t,0)}) \stackrel{?}{=} e(pk_s^{e_{(t,0)}} v_{(t,0)} w_{(t,0)}, h).$$

  - Case 2 ($k = 1$):

$$e_t = H_1(v_{(t,1)}, ID_t), \quad e_{(t,0,1)} = H_2(w_{(t,1)}, pk_1, ID_t),$$

$$e(g, u_1) \stackrel{?}{=} e(pk_s^{e_t + e_{(t,0,1)}} pk_1 v_{(t,1)} w_{(t,1)}, h).$$

  - Case 3 ($k > 1$):

$$e_t = H_1(v_{(t,k)}, ID_t), \quad e_{(t,k-1,k)} = H_2(w_{(t,k)}, pk_k, ID_t),$$

$$e(g, u_{(t,k)}) \stackrel{?}{=} e(pk_s^{e_t} pk_k pk_{k-1}^{e_{(t,k-1,k)}} v_{(t,k)} w_{(t,k)}, h).$$

  Outputs *true* if any equation holds, otherwise outputs $false$.

Correctness Without loss of generality, we show the correctness of our RFID ownership transfer protocol in Case 3 as follows:

$$\begin{aligned}
&e(g, u_{(t,k)}) = e(g, h^{s_{(t,k-1,k)} + sk_k}) \\
&= e(g, h^{sk_{k-1} e_{(t,k-1,k)} + r + sk_k}) \\
&= e(pk_{k-1}^{e_{(t,k-1,k)}} g^r pk_k, h) \\
&= e(pk_s^{e_t} g^{r_t} pk_{k-1}^{e_{(t,k-1,k)}} g^r pk_k (pk_s^{e_t} g^{r_t})^{-1}, h) \\
&= e(pk_s^{e_t} pk_k pk_{k-1}^{e_{(t,k-1,k)}} v_{(t,k)} w_{(t,k)}, h).
\end{aligned}$$

5 Security Models of Ownership Transfer Protocols

The security of a RFID ownership transfer protocol usually relies on the underlining authentication protocols. It is extremely hard to provide the strong security if a symmetric-key authentication protocol is employed. Typically, the security model of symmetric-key based ownership transfer protocols does not provide corruption oracle which outputs the state of a tag. Once the key is exposed, the security of tag is completely compromised. Elkhiyaoui, Blass and Molva [1] recently presented a ROTIV protocol secure against the key corruption. It applies the public key cryptography in the authentication while the tag is only required to compute a hash function. However, the proposed security model cannot capture the adversary who can rewrite the content of a tag. It is possible when an adversary gains the key of tag. In this section, we enhance the security models of ownership transfer protocols. A general assumption is that owners are not able to launch collusion attacks in an ownership transfer [10].

5.1 Adversaries and Oracles

The ability of the adversary is essentially restricted by the actions that he is allowed to carry out. In security models, we specify the actions of adversary via the oracle queries. We now define the oracles which are used in the security models of ownership transfer protocols in this paper.

Definition 2 (Oracles). *The adversary plays with a challenger by given public information of the system and the following oracle calls.*

- $(O, pk) \leftarrow \mathsf{SetupOwner}(ID)$*: Taking as input an identity* ID*, it creates an owner* O *and runs the algorithm* KeyGen *to output a public key* pk*.*
- $T \leftarrow \mathsf{TagInit}(ID_t)$*: Creates a tag* T *with the identity* ID_t *and sets the authentication key* y*. It runs the algorithm* TagInit *and outputs the tag* T*.*
- $(ID_t, \sigma_{(t,k)}) \leftarrow \mathsf{Auth}(T, O_k)$*: Taking as input a current owner* O_k *and a tag* T*, it outputs the identity* ID_t *of tag and its ownership proof* $\sigma_{(t,k)}$ *if* T *is valid, outputs* $\perp$ *otherwise.*
- $c \leftarrow \mathsf{CorruptTag}(T)$*: Taking as input a tag* T*, and outputs the complete internal state* c *of* T*. Note that the oracle does not destroy the tag* T *and the tag is available in the future oracle calls.*
- $sk \leftarrow \mathsf{CorruptOwner}(ID)$*: Taking as input an owner's identity* ID*, and outputs the private key* sk *of the owner.*
- $\{0, 1\} \leftarrow \mathsf{Rewrite}(T, c', y)$*: Taking as input a tag* T*, a new state* c' *and an authentication key* y*, it rewrites the state by* c' *and outputs* 1 *if the key is valid,* 0 *otherwise.*
- $\sigma_{(t,k)} \leftarrow \mathsf{OwnerProve}(T, ID_s, ID_{k-1}, ID_k)$*: Taking as input a tag* T*, an identity* ID_s *of supplier, an identity* ID_{k-1} *of previous owner and an identity* ID_k *of current owner, it outputs an ownership proof* $\sigma_{(t,k)}$ *of the tag.*
- $\Sigma_{(t,k,k+1)} \leftarrow \mathsf{Transfer}(T, ID_k, ID_{k+1})$*: Taking as input a tag* T*, an identity* ID_k *of current owner and an identity* ID_{k+1} *of new owner, it outputs an authenticated ownership transfer proof* $\Sigma_{(t,k,k+1)}$ *of the tag.*

Definition 3 (Type I and Type II adversary). *The adversary is defined by the oracle calls and the goal of the experiment.*

- Type I Adversary*(*$\mathcal{A}_I$*): is also allowed to query all above oracles except the* CorruptOwner*. It aims to output a valid proof of authenticated transfer which cannot be detected during the transfer.*
- Type II Adversary*(*$\mathcal{A}_{II}$*): is allowed to query all above oracles. It aims to output a valid proof of ownership of the target tag which cannot be detected in the ownership verification.*

5.2 Security Models

We define the security models of ownership transfer protocols in this section. Each model captures the capability of different adversaries. A security model is defined as an experiment which plays between the adversary and the challenger.

We denote that the security parameters as r, s and n, which are respectively the number of owner initiations, the number of oracle calls and the number of tag initiations. There are two experiments defined in our security model. An RFID ownership transfer protocol is secure iff it is secure in both experiments. The security models defined in this section are suitable to ownership transfer protocols in the two-party model.

Security Against Type I Attack Type I adversary is a person who attempts to forge a valid proof of authenticated transfer. $\mathcal{A}_I$ interacts with the challenger via oracle calls and outputs a proof of transfer. It is described as in experiment $\mathbf{Exp}^{secure}_{\mathcal{A}_I,\mathcal{S}}[r,s,n]$ in Fig.6.

Experiment $\mathbf{Exp}^{secure}_{\mathcal{A}_I,\mathcal{S}}[r,s,n]$:

- **Setup:** The challenger runs the algorithm Setup to generate public parameters $params$ and returns to $\mathcal{A}_I$. It initiates a supplier S^*.
- **Phase 1(Learning):**
  - $\mathcal{A}_I$ can query all above oracles except CorruptOwner to C.
  - Outputs two sets $\mathcal{T} = \{T_1, \ldots, T_n\}$ and $\mathcal{O} = \{O_1, \ldots, O_r\}$, which are created tags and owners.
- **Phase 2(Forge):**
  - $\mathcal{A}_I$ submits a target tag $\mathsf{T}^* \in \mathcal{T}$, current owner O_k^* and new owner O_{k+1}^* to $\mathcal{C}$, such that $(O_k^*, O_{k+1}^*) \in \mathcal{O} \cup \{S^*\}$.
  - $\mathcal{A}_I$ queries oracles Auth, CorruptTag, Rewrite, Transfer and OwnerProof to $\mathcal{C}$.
  - $\mathcal{A}_I$ outputs a proof Σ^* of authenticated ownership transfer.

Exp outputs success if $true \leftarrow \mathsf{OwnershipVerify}(ID_t^*, pk_s^*, pk_k^*, pk_{k+1}^*, \mathsf{OwnerProve}(ID_t^*, sk_{k+1}^*, \Sigma^*, \sigma^*))$, such that $\Sigma^* \nleftarrow Transfer(ID_t^*, ID_k^*, ID_{k+1}^*)$.

Fig. 6: Type I security experiment of the ownership transfer protocols

Definition 4. *An ownership transfer protocol is (r,s,n,ϵ)-secure against the Type I attack, if any $\mathcal{A}_I$ who succeeds in $\mathbf{Exp}^{secure}_{\mathcal{A}_I,\mathcal{S}}[r,s,n]$ has advantage*

$$\Pr[success \leftarrow \mathbf{Exp}^{secure}_{\mathcal{A}_I,\mathcal{S}}[r,s,n]] \leq \epsilon,$$

where ϵ is negligible.

Security Against Type II Attack The Type II adversary acts as a person who attempts to forge a valid proof of ownership. $\mathcal{A}_{II}$ interacts with the challenger $\mathcal{C}$ via oracle calls and outputs a proof of ownership at the end of the experiment. The experiment $\mathbf{Exp}^{secure}_{\mathcal{A}_{II},\mathcal{S}}[r,s,n]$ is defined as in Fig. 7.

Experiment $\mathbf{Exp}^{secure}_{\mathcal{A}_{II},\mathcal{S}}[r,s,n]$:

- **Setup:**The challenger runs the algorithm Setup to generate public parameters $params$ and returns to $\mathcal{A}_{II}$. It initiates a supplier S^*.
- **Phase 1(Learning):**
  - $\mathcal{A}_{II}$ can query all above oracles to $\mathcal{C}$.
  - Outputs two sets $\mathcal{T} = \{T_1, \ldots, T_n\}$ and $\mathcal{O} = \{O_1, \ldots, O_r\}$, which are created tags and owners.
- **Phase 2(Forge):**
  - $\mathcal{A}_{II}$ submits a target tag $\mathsf{T}^* \in \mathcal{T}$, previous owner O^*_{k-1} and current owner O^*_k to $\mathcal{C}$, such that $(\mathsf{O}^*_{k-1}, \mathsf{O}^*_k) \in \mathcal{O} \cup \{S^*\}$.
  - $\mathcal{A}_{II}$ queries oracles Auth, $\mathsf{CorruptTag}$, $\mathsf{Rewrite}$, $\mathsf{Transfer}$ and $\mathsf{OwnerProve}$ to $\mathcal{C}$.
  - $\mathcal{A}_{II}$ outputs a proof σ^* of ownership.

Exp outputs *success* if it satisfies the restrictions as follows,

1. $true \leftarrow \mathsf{OwnershipVerify}(ID^*_t, pk^*_s, pk^*_{k-1}, pk^*_k, \sigma^*)$,
2. $\sigma^* \nleftarrow \mathsf{OwnerProve}(\mathsf{ID}^*_\mathsf{t}, \mathsf{ID}^*_\mathsf{s}, \mathsf{ID}^*_{\mathsf{k}-1}, \mathsf{ID}^*_\mathsf{k})$,
3. $sk^*_k \nleftarrow \mathsf{CorruptOwner}(ID^*_k) \ \vee \ (\Sigma^* \nleftarrow \mathsf{Transfer}(ID^*_t, ID^*_{k-1}, ID^*_k) \wedge sk^*_{k-1} \nleftarrow \mathsf{CorruptOwner}(ID^*_{k-1}))$.

Fig. 7: Type II security experiment of the ownership transfer protocols.

Definition 5. *An ownership transfer protocol is (r,s,n,ϵ)-secure against the Type II attack, if any $\mathcal{A}_{II}$ who succeeds in $\mathbf{Exp}^{secure}_{\mathcal{A}_{II},\mathcal{S}}[r,s,n]$ has advantage*

$$\Pr[success \leftarrow \mathbf{Exp}^{secure}_{\mathcal{A}_{II},\mathcal{S}}[r,s,n]] \leq \epsilon,$$

where ϵ is negligible.

Lemma 1. *If an ownership transfer protocol is secure against the Type II attack, it is secure against the Type I attack.*

Due to the page limitation, the proof of Lemma 1 is referred to the full version.

6 Security Analysis

An ownership transfer protocol is secure if it is against two types of attacks defined in Section 5.2. Without loss of generality, we analyse the security of proposed protocol on the k-th level. According to Lemma 1, we only show the security proof of the proposed protocol in $\mathsf{Type\ II}$ experiment.

Theorem 1. *The proposed ownership transfer protocol is (r,s,n,ϵ)-secure against the Type II attack if the CDH assumption is held.*

Due to the page limitation, the proof of Lemma 1 is referred to the full version.

7 Conclusion

In this paper, we defined a new secure model of ownership transfer protocols. It enhances the existing security models. We provided a definition of RFID ownership transfer and proposed a secure ownership transfer protocol. It achieves a single verification key to all the tags from an owner. The protocol satisfies all the security requirements. A formal proof of our proposed protocol was given.

References

1. Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n signatures from a variety of keys. In: Zheng, Y. (ed.) Advances in Cryptology - Asiacrypt '02. LNCS, vol. 2501, pp. 415–432. Springer (2002)
2. Abyaneh, M.R.S.: On the privacy of two tag ownership transfer protocols for RFIDs. CoRR abs/1202.4663 (2012)
3. van Deursen, T., Mauw, S., Radomirovic, S., Vullers, P.: Secure ownership and ownership transfer in RFID systems. In: Backes, M., Ning, P. (eds.) ESORICS. LNCS, vol. 5789, pp. 637–654. Springer (2009)
4. Elkhiyaoui, K., Blass, E.O., Molva, R.: Rotiv: RFID ownership transfer with issuer verification. In: RFIDSec. Lecture Notes in Computer Science, vol. 7055, pp. 163–182. Springer (2011)
5. Fernàndez-Mir, A., Trujillo-Rasua, R., Castellà-Roca, J., Domingo-Ferrer, J.: A scalable RFID authentication protocol supporting ownership transfer and controlled delegation. In: Juels, A., Paar, C. (eds.) RFIDSec. LNCS, vol. 7055, pp. 147–162. Springer (2011)
6. Fouladgar, S., A, H.: An efcient delegation and transfer of ownership protocol for RFID tags. In: First International EURASIP Workshop on RFID Technology. ACM (2007)
7. Kapoor, G., Zhou, W., Piramuthu, S.: Multi-tag and multi-owner RFID ownership transfer in supply chains. Decision Support Systems 52(1), 258–270 (2011)
8. Kulseng, L., Yu, Z., Wei, Y., Guan, Y.: Lightweight mutual authentication and ownership transfer for RFID systems. In: INFOCOM. pp. 251–255. IEEE (2010)
9. Molnar, D., Soppera, A., Wagner, D.: A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags. In: Preneel, B., Tavares, S.E. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 3897, pp. 276–290. Springer (2006)
10. Ng, C.Y., Susilo, W., Mu, Y., Safavi-Naini, R.: Practical RFID ownership transfer scheme. Journal of Computer Security 19(2), 319–341 (2011)
11. Osaka, K., Takagi, T., Yamazaki, K., Takahashi, O.: An efficient and secure RFID security method with ownership transfer. In: Wang, Y., ming Cheung, Y., Liu, H. (eds.) CIS. LNCS, vol. 4456, pp. 778–787. Springer (2006)
12. Rizomiliotis, P., Rekleitis, E., Gritzalis, S.: Security analysis of the song-mitchell authentication protocol for low-cost RFID tags. Comm. Letters. 13(4), 274–276 (2009)
13. Saito, J., Imamoto, K., Sakurai, K.: Reassignment scheme of an RFID tag's key for owner transfer. In: Enokido, T., Yan, L., Xiao, B., Kim, D., Dai, Y.S., Yang, L.T. (eds.) EUC Workshops. LNCS, vol. 3823, pp. 1303–1312. Springer (2005)
14. Song, B.: RFID tag ownership transfer. In: 4th Workshop on RFID Security - RFIDSec (2008)