

ON THE PRIVACY OF TWO TAG OWNERSHIP TRANSFER PROTOCOLS FOR RFIDS

Mohammad Reza Sohizadeh Abyaneh*

Abstract. In this paper, the privacy of two recent RFID tag ownership transfer protocols are investigated against the tag owners as adversaries.

The first protocol called ROTIV is a scheme which provides a privacy-preserving ownership transfer by using an HMAC-based authentication with public key encryption. However, our passive attack on this protocol shows that any legitimate owner which has been the owner of a specific tag is able to trace it either in the past or in the future. Tracing the tag is also possible via an active attack for any adversary who is able to tamper the tag and extract its information.

The second protocol called, *Chen et al.'s* protocol, is an ownership transfer protocol for passive RFID tags which conforms EPC Class1 Generation2 standard. Our attack on this protocol shows that the previous owners of a particular tag are able to trace it in future. Furthermore, they are able even to obtain the tag's secret information at any time in the future which makes them capable of impersonating the tag.

Key Words: RFID, Ownership Transfer, Privacy.

1 INTRODUCTION

Radio frequency identification(RFID) is currently considered as the next generation technology that mainly used to identify massive objects

*Department of Informatics, University of Bergen

in an automated way and will substitute traditional optical barcode system in near future. The RFID advantages such as reducing supply chain inefficiencies and improving inventory flow leaves no doubt that the dominant deployment of barcodes nowadays in supply chain will be promptly taken over by RFID tags. But it has its own drawbacks too.

As products flow through a supply chain, their ownership is transferred from one partner to the next. This transfer of ownership extends to the RFID tags attached to these products. Thus all information associated with the tag will need to be passed from the current to the new owner. However, at the moment of tag ownership transfer, both the current and new owners have the information necessary to authenticate a tag, and this fact may cause an infringement of tag *owner privacy* [5].

To handle this problem, tag ownership transfer protocols are proposed to transfer the ownership of a tag from one owner to another securely. The proposed schemes for ownership transfer protocols are divided into two groups. Some schemes exploit a trusted third party(TTP) which acts as a secure channel to transfer some information between the entities. One of the first solution of this kind was proposed by Saito *et al.*[6]. However, the security of their scheme is only based on the short read range of the backward channel (tag to reader communication) by assuming that it is hard for adversaries to eavesdrop on this channel. Another scheme with TTP is proposed by Molnar *et al.* [7]. They exploit the TTP to manage tag keys by a tree structure. But in this protocol one key is shared by several tags which makes this protocol vulnerable. The privacy of the whole system decreases quickly when more tags are compromised [8].

There also exist some *decentralized* proposals without a using TTP. Most of these schemes have two following assumptions: there is a secure channel between the current and new owner to pass the tag's information securely. They also assume that the new owner and the tag will be able to execute an authentication session in an isolated environment without presence of the current owner after the ownership transfer is completed in order to update some secret parameters.

For instance, Soppera and Burbridge [9] adopt the scheme of Molnar *et al.* by replacing the TTP with some distributed local devices called RFID *acceptor* tag. In [13], the authors have also proposed a decentralized protocol relying on the assumption that owners are able to change the tag key in an isolated environment. However, this protocol has security vulnerabilities well described in [14]. Song *et al.* [11] proposed a scheme with introduction of a new property called *authorization recovery*

which facilitates the ownership transfer of a tag to its previous owner. But Pedro *et al.* [12] showed that their schemes has some vulnerabilities as well.

Recently, two other tag ownership transfer protocols have been proposed. The first scheme is called an RFID ownership transfer with issuer verification (ROTIV) [16] which provides a constant-time, privacy-preserving tag ownership transfer. The ROTIV's main idea is to combine an HMAC-based authentication with public key encryption. The second scheme which is proposed by Chen *et al.* [17], proposes an RFID ownership transfer systems which conforms the requirements of EPCglobal Class-1 Generation-2 Standard.

Our Contribution. In this paper, we investigate the privacy of two aforementioned ownership transfer protocols. The investigation includes some attacks to violate the forward and backward privacy as well as previous and new owner privacy properties of the schemes.

Outline. The remainder of this paper is organized as follows. Section 2 describes the privacy issues and properties required for tag ownership transfer protocols as well as system and adversary modelings. In Sections 3 and 4 the description of the the ROTIV and Chen *et al.* protocols and our attack on them are presented respectively, and finally, Section 5 concludes the paper.

2 PRELIMINARIES

To lend clarity to our discussions in the subsequent sections, in this section, we outline the models and properties used in ownership transfer protocol.

2.1 SYSTEM MODEL

In ownership transfer protocols, there are mainly three active entities involved: *current owner*, *tag* and *new owner*. The owners in an ownership transfer protocols are some readers in practice which take the role of ownership in these kinds of protocols. The ownership transfer protocols typically provide a solution to transfer the tag's information from the current owner to the new owner.

Most of the ownership transfer protocols consist of two phases, an *authentication phase* and a *ownership transfer phase*. By the former phase, the tag and two owners are mutually authenticated and the latter phase

assures all three entities that the ownership of the tag is transferred in a proper and privacy-preserving way.

2.2 PRIVACY PROPERTIES

Generic privacy properties and how to formalize them for RFID systems have been extensively explored in the literature [1–4]. The two generic privacy property we address in this paper are:

- *Backward Privacy*: an adversary should not be able to trace past transactions between an owner and a tag, even if it compromises/tamper the tag.
- *Forward Privacy*: an adversary should not be able to trace future transactions between an owner and a tag, even if it compromises/tamper the tag.

On the other hand, in tag ownership transfer protocols changes of tag owner could occur frequently and at the moment of tag ownership transfer, both the current and new owners have the information necessary to authenticate a tag, and this fact may cause an infringement of tag *owner privacy*. Therefore, there are two extra privacy issues dedicated for ownership transfer protocols in the literature [10, 15]:

- *New owner privacy*: Once ownership of a tag has been transferred to a new owner, only the new owner should be able to identify and control the tag. The previous owner of the tag should no longer be able to identify or *trace* the tag.
- *Current/previous owner privacy*: When ownership of a tag has been transferred to a new owner, the new owner of a tag should not be able to *trace* past interactions between the tag and its previous owner.

2.3 ADVERSARY MODEL

In [3], Juels and Weis give a formal model of the privacy in RFID systems. In this model, tags (\mathcal{T}) and readers/owners (\mathcal{R}) interact in protocol sessions. During this interaction there is also an adversary entity \mathcal{A} which passively or actively interacts with them. The adversary may have access to an oracle which can be queried by the following queries:

- $\text{Execute}(\mathcal{T}, \mathcal{R}, i)$: This query is responded by the information of \mathcal{T} and \mathcal{R} interactions in an honest protocol session at time instance i .
- $\text{Send}(\mathcal{P}_1, \mathcal{P}_2, i, m)$: This query models active attacks by allowing the adversary \mathcal{A} to impersonate some entity, a tag or a reader, \mathcal{P}_1 in some protocol session i and send a message m of its choice to an instance of some other entity \mathcal{P}_2 .
- $\text{Corrupt}(\mathcal{T})$: This query allows the adversary \mathcal{A} to tamper the tag to learn the stored secret information of the tag \mathcal{T}
- $\text{Test}(i, \mathcal{T}_0, \mathcal{T}_1)$: This query is responded by a random bit $b \in \{0, 1\}$ and the interaction information of the tag \mathcal{T}_0 and \mathcal{T}_1 with the reader/owner at i^{th} time instance.

2.4 ATTACK SCENARIO

In [3], the adversary \mathcal{A} aims at tracing a specific target tag T . To do so, she,

- absorbs the information she requires about the target tag T by the means of queries previously described.
- choose two test tags T_0 and T_1 where one of them is T , and asks the oracle for the challenge by Test query. The response will be the interactions between the T_0 and T_1 tags with the reader R at a specific time instance.

The adversary succeeds to violate the privacy of the tag by tracing it, if she is able to distinguish the tag T between the two tested tags by outputting 0 or 1.

2.5 NOTATIONS

Here, we explain the notations used hereafter.

- $E_k(\cdot)$: Symmetric/asymmetric encryption function operation with the key k .
- pk_X, sk_X : Public and private key of entity X respectively.
- $h_k(\cdot)$: Keyed hash function with key k .
- $h(\cdot)$: Hash functions.

- $PRNG(.)$: Pseudo random number generator.
- T, O_n, O_{n+1} : Tag, current owner and new owner.
- ID_X : The identification (ID) of entity X .
- N_X : Random numbers generated by entity X .
- m_i : dynamic value m at time instance i .

3 ROTIV PROTOCOL

ROTIV is a decentralized scheme which does not require a trusted third party to perform tag ownership transfer. This protocol provides issuer verification that allows prospective owners to check the identity of the entity which has issued the tag. The authors have claimed that their scheme ensures both forward and backward privacy and it also preserves current and new owner privacy.

There are four entities involved in the protocol, a tag T , current owner O_n , new owner O_{n+1} and issuer I which initializes the tag and owners.

In ROTIV, the T stores a symmetric key k , a *state parameter* s , where k is a key shared between the tag and its owner and s is an Elgamal encryption of T 's identification information.

3.1 PRELIMINARIES

Bilinear pairing

Let G_1, G_2 and G_T be groups, such that G_1 and G_2 have the same prime order q . Pairing $e : G_1 \times G_2 \rightarrow G_T$ is a bilinear pairing if has the following properties:

1. *bilinear*: $\forall a, b \in \mathbb{Z}_q, g_1 \in G_1$ and $g_2 \in G_2, e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
2. *computable*: there is an efficient algorithm to compute $e(g_1, g_2)$ for any $(g_1, g_2) \in G_1 \times G_2$;
3. *non-degenerate*: if g_1 is a generator of G_1 and g_2 is a generator of G_2 , then $e(g_1, g_2)$ is a generator G_T .

3.2 DESCRIPTION

Setup: The issuer I outputs $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$, where $\mathbb{G}_1, \mathbb{G}_T$ are subgroups of prime order q , g_1 and g_2 are random generators of \mathbb{G}_1 and \mathbb{G}_2 respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a bilinear pairing.

The issuer chooses $x \in \mathbb{Z}_q^*$ and computes the pair (g_1^x, g_2^x) . The I 's public and secret keys are:

$$sk_I = (x, g_1^x), pk_I = g_2^x \quad (1)$$

I randomly selects $\alpha_n \in \mathbb{Z}_q^*$ and provides each owner O_n with a secret key $sk_{O_n} = \alpha_n$ and a public key $pk_{O_n} = (g_1^{\alpha_n}, g_2^{\alpha_n})$. All owners know each other's public keys.

Tag Initialization: The issuer I picks a random number $t \in \mathbb{F}_q$, where \mathbb{F}_q is the finite field with q elements. Using a cryptographic hash function $h : \mathbb{F}_q \rightarrow \mathbb{G}_1$, I computes $u_0 = 1$ and $v_0 = h^x(t)$. Finally, I chooses randomly a key $k_0 \in \mathbb{F}_q$ and stores: (k_0, s_0) , where $s_0 = (u_0, v_0)$ into the tag. I also provides O_n with T 's information ref^{O_n} . This information includes two *dynamic* values k_{old}, k_{new} which are updated after each successful transaction and two *static* values $\delta = t, \psi = h^x(t)$ which represent the identification of the issuer of the tag.

$$ref^{O_n} = (k_{old}, k_{new}, \delta, \psi) = (k_0, k_0, t, h^x(t)) \quad (2)$$

Before accepting the tag, the owner can read the tag and checks the authenticity of the static values of the tag:

$$e(h(\delta), pk_I) = e(\psi, g_2) \quad (3)$$

Ownership Transfer: The ROTIV ownership transfer protocol (Fig.1) is a combination of two mutual authentication sessions between the tag and current and new owners with the ownership transfer protocol between the current owner O_n and the new owner O_{n+1} .

In i^{th} time instance of the ROTIV protocol:

1. New owner O_{n+1} generates a random nonce $N_{O_{n+1}}$ and sends it to the tag and the current owner simultaneously.
2. The tag T also generates a random number N_T and send it with its status parameter $s_i = (u_i, v_i)$ and a hash $m_i = h_{k_i}(N_{O_{n+1}}, N_T, s_i)$ to the new owner.
3. O_{n+1} selects a random number r_v and computes $A_v = u_i^{r_v}$. Then, it

sends $N_{O_{n+1}}, N_T, s_i, m_i$ and A_v to the current owner O_n . In this way, O_n is able to authenticate the tag by computing,

$$\psi = \frac{v_i}{(u_i)^{\alpha_n^2}} \quad (4)$$

Then, it searches in the database to see if ψ is in the database or not. If not, it aborts authentication. Otherwise, it looks up T 's ownership references ref^{O_n} in the database to checks if $m_i = h_{k_i^{new}}(N_{O_{n+1}}, N_T, s_i)$ or $m_i = h_{k_i^{old}}(N_{O_{n+1}}, N_T, s_i)$. For the former case $k_i = k_i^{new}$ and for the latter case $k_i = k_i^{old}$.

4. If the authentication process succeeds O_n gives O_{n+1} the following information via a secure channel:

$$ref^V = (A, B, C) = (t, h^x(t), A_v^{\alpha_n}) \quad (5)$$

$$ref^{O_n} = (k_{old}, k_{new}, \delta, \psi) = (k_i, k_{i+1}, t, h^x(t)) \quad (6)$$

The new owner O_{n+1} check the validity of the provided information by (3).

Now, the new owner can verify whether the issuer of the tag T is I by checking whether the following equations hold:

$$e(h(A), pk_I) = e(B, g_2) \quad (7)$$

$$e(C, g_2) = e(A_v, g_2^{\alpha_n}) \quad (8)$$

$$e(v_i, g_2)^{r_v} = e(B, g_2)^{r_v} e(C, g_2^{\alpha_n}) \quad (9)$$

5. If the verification succeeds, O_{n+1} chooses a new random number r_{i+1} and computes:

$$s_{i+1} = (u_i, v_i) = (g_1^{r_{i+1}}, h^x(t) \cdot g_1^{\alpha_n^2 r_{i+1}}) \quad (10)$$

$$m_{i+1} = h_{k_i}(N_T, s_{i+1}) \quad (11)$$

and sends s_{i+1}, m_{i+1} to the tag and updates its database. Now, T authenticates O_{n+1} by checking the content of m_{i+1} . If the authentication succeeds T updates its state parameter to s_{i+1} and its symmetric key to the new key k_{i+1} where,

$$k_{i+1} = PRNG(k_i, N_{O_{n+1}}) \quad (12)$$

In order to prevent the current owner from tracing the tag later in the future, the new owner has to run a mutual authentication with the tag outside the range of the current owner after the ownership transfer is complete.

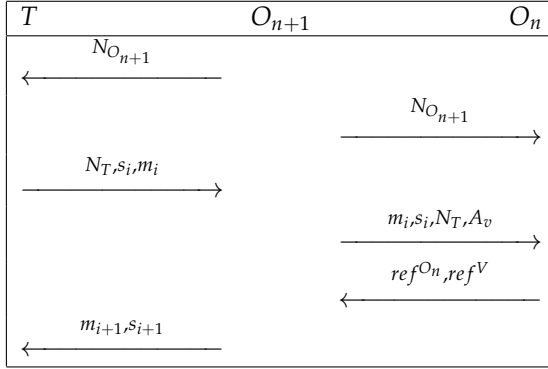


Fig. 1: Ownership transfer in ROTIV

3.3 OUR ATTACKS

In this attack, we target mainly the *ownership privacy* including current and new owner privacy of the ROTIV protocol. Correspondingly, the adversary \mathcal{A} has been one of the owners of the tag T at least once. For example, without loss of generality, we can assume that $\mathcal{A} = O_n$. Therefore, at a time instance e.g. i , she has had access to the tags's information ref^{O_n} . We also assume that the adversary is passive and thus has access only to Execute and Test queries.

According to the attacking scenario described in Section 2.4, the adversary follows the procedure below to trace the tag T via distinguishing that which of the two test tags, T_0 and T_1 , are T .

1. \mathcal{A} retrieves the static information of the tag T , $\delta = t, \psi = h^x(t)$, from the information she has been give at time i , ref^{O_n} .
2. \mathcal{A} queries $\text{Test}(j, \mathcal{T}_0, \mathcal{T}_1)$ and obtains (13) and (14).

$$\{N_{O_i}, N_{T_0}, m_j, m_{j+1}, s_j, s_{j+1}\} \quad (13)$$

$$\{N_{O'_i}, N_{T_1}, m'_j, m'_{j+1}, s'_j, s'_{j+1}\} \quad (14)$$

which are the messages exchanged between the owner O_i and tags T_0 and T_1 respectively.

3. \mathcal{A} saves $s_j = (u_j, v_j)$ and $s'_i = (u'_j, v'_j)$.

4. \mathcal{A} checks whether (15) or (16) holds,

$$e(v_j, g_2) = e(h(\delta), pk_I) e\left(\left(\frac{v_j}{\psi}\right), g_2\right) \quad (15)$$

$$e(v'_j, g_2) = e(h(\delta), pk_I) e\left(\left(\frac{v'_j}{\psi}\right), g_2\right) \quad (16)$$

5. If (15) is correct then \mathcal{A} outputs 0 i.e. $T = T_0$, otherwise she outputs 1 i.e. $T = T_1$.

Note that we can write (15) because according to bilinear pairing properties of e , we have:

$$\begin{aligned} e(v_i, g_2) &= e(h^x(t), g_1^{\alpha_1^2 r_i}, g_2) \\ &= e(\psi \cdot g_1^{\alpha_1^2 r_i}, g_2) \\ &= e(\psi, g_2) e(g_1^{\alpha_1^2 r_i}, g_2) \\ &= e(h(\delta), g_2^x) e(g_1^{\alpha_1^2 r_i}, g_2) \\ &= e(h(\delta), pk_I) e(g_1^{\alpha_1^2 r_i}, g_2) \\ &= e(h(\delta), pk_I) e\left(\left(\frac{v_i}{\psi}\right), g_2\right) \end{aligned}$$

Using the scenario above, any owner in the protocol which has had the ownership of the tag T is able to trace it. It is worth mentioning that since the update procedure of state values s are performed independent of their previous values (step 5 of ownership transfer), the aforementioned tracing scenario can be applied both on the state values of the past and the future. Hence any owner who has accessed to the static values of a tag is able to trace it at any time in the past or future by only eavesdropping state parameter of the tag s . It implies that the ROTIV protocol lacks both previous owner and new owner privacy properties.

Remark 1. It should be noted that if an adversary \mathcal{A}' has access to Corrupt query which gives her this privilege to tamper the tag and access to the tag's static information $t, h^x(t)$, her state of knowledge about the tag is exactly the same as that the adversary \mathcal{A} in the stated attack. Hence, she will also be able to exploit (15) to trace T in any time in the past and future. This implies that the ROTIV protocol lacks forward and backward privacy as well.

4 CHEN *et al.*'S PROTOCOL

Chen *et al.*'s protocol is designed to meet the requirements of EPC Class1 Generation2 standard (ISO18000-6C) for passive RFID tags. According to this standard, RFID tags's computation capabilities is restricted to only performing a 16-bit Cyclic Redundancy Code (CRC) and 16-bit Pseudo-Random Number Generator (PRNG). The authors have claimed that their scheme ensures both forward and backward privacy and it also preserves current and new owner privacy.

There are four entities involved in the protocol, a tag T , current owner O_n , new owner O_{n+1} and issuer I which issues a new issuer identification to be stored into the tags after each ownership transfer phase.

4.1 DESCRIPTION

Chen *et al.*'s ownership transfer protocol consist of three phases: *requiring phase*, *authentication phase* and *ownership transfer phase*. In Chen *et al.*'s protocol, the T stores two dynamic symmetric keys k_i, k_i^* and the $h(t_i)$ which is the hash of the issuer identification. In addition to the tag's information the owner has the issuer identification t_i .

In the i^{th} time instance of requiring phase (Fig.2), the current owner first signs the tag's certificate t_i and the identification of the new owner:

$$SG_{O_n} = Sign_{sk_{O_k}}(t_i, ID_{O_{n+1}}) \tag{17}$$

After that, it encrypts this message with the next owner's public key to get C_i :

$$C_i = E_{pk_{O_{n+1}}}(t_i, SG_{O_n}) \tag{18}$$

and transfers the message (ID_{O_k}, C_i) to the new owner O_{n+1} .

In authentication phase (Fig.3), the current owner first generates a random number N_{O_n} and then computes A_i :

$$A_i = CRC(k_i \oplus N_{O_n}) \tag{19}$$

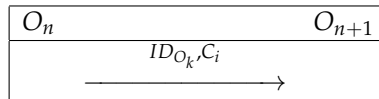


Fig. 2: Requiring phase

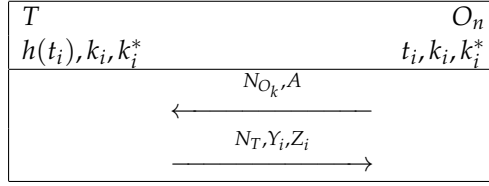


Fig. 3: Authentication phase

and sends it with N_{O_n} to the tag. Upon receiving these messages, the tag verifies the content of the message A_i . If the verification succeeds, the tag generates a new random value N_T , and computes the X_i, Y_i and Z_i as following.

$$X_i = \text{CRC}(N_T \oplus k_i^*) \quad (20)$$

$$Y_i = k_i^* \oplus ID_T \oplus X_i \oplus k_{i+1} \quad (21)$$

$$Z_i = \text{CRC}(X_i \oplus k_i \oplus Y_i) \quad (22)$$

Moreover, the tag updates its keys as:

$$k_{i+1} = (k_i^* \oplus ID_T \oplus N_T \oplus Y_i) \quad (23)$$

$$k_{i+1}^* = \text{PRNG}(k_i^*) \quad (24)$$

and transfers (N_T, Y_i, Z_i) to the current owner. Upon receiving the message, O_n checks the content of X_i and Z_i . If this verification succeeds, it obtains k_{i+1} and updates its values accordingly.

In the ownership transfer phase (Fig.4), the new owner O_{n+1} uses its own private key to decrypt C_i received in the requiring phase and obtains SG_{O_k} and t_i . Then, it uses the O_n 's public key pk_{O_n} to verify the correction of SG_{O_k} . If the signature is verified successfully, the new owner signs the ID of its own as well as the current owner's:

$$SG_{O_{n+1}} = \text{Sign}_{sk_{O_{k+1}}}(ID_{O_k}, ID_{O_{n+1}}) \quad (25)$$

And sends the tuple $\{ID_{O_i}, ID_{O_{i+1}}, SG_{O_i}, SG_{O_{i+1}}, t_i\}$ to the issuer I to issue a new issuer identification for the tag.

The issuer checks the content of this message and if it is correct, it issues the t_{i+1} and computes $t_{i+1} \oplus k_{i+1}$ and $h(t_{i+1})$ and transmits them to O_n . Upon receiving this message, O_n sends the former message to

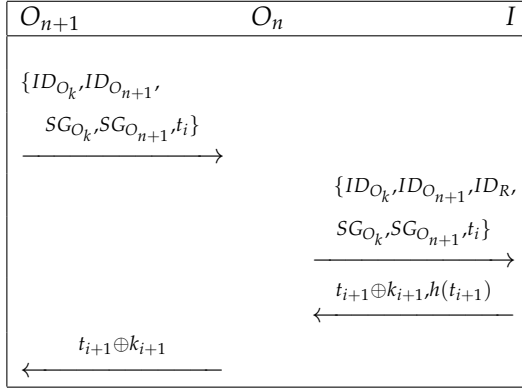


Fig. 4: Ownership transfer phase

the new owner and writes the latter one into the tag's memory. The new owner can also obtain the t_{i+1} by XORing the message received from the current owner and the new key stored in the memory.

$$t_{i+1} = (t_{i+1} \oplus k_{i+1}) \oplus k_{i+1} \quad (26)$$

4.2 OUR ATTACK

The adversary \mathcal{A} in our attack is one of the previous owners of the tag T . Therefore, she has had access to ID_T, k_i and k_i^* , where the ID_T is the static ID of the tag T or the tag's electronic product code(EPC) and k_i and k_i^* are the dynamic keys of the tag at time instance i when the tag has been in the possession of \mathcal{A} as the owner.

Being given the messages exchanged between two tags T_0, T_1 , which one of them is the tag T , and another owner O_l at two consecutive time instance j and $j + 1$, the adversary follows the procedure below to distinguish which of the test tags is the tag T .

1. \mathcal{A} retrieves the static identity of the tag T , ID_T .
2. \mathcal{A} queries $\text{Test}(j, \mathcal{T}_0, \mathcal{T}_1), \text{Test}(j + 1, \mathcal{T}_0, \mathcal{T}_1)$ and obtain

$$\{A_j, N_{T_0}, N_{O_i}, Y_j, Z_j\}, \{A_{j+1}, N'_{T_0}, N'_{O_i}, Y_{j+1}, Z_{j+1}\}$$

$$Y_j = k_j^* \oplus ID_{T_0} \oplus X_j \oplus k_j \quad (27)$$

$$Y_{j+1} = k_{j+1}^* \oplus ID_{T_0} \oplus X_{j+1} \oplus k_{j+1} \quad (28)$$

$$Z_j = CRC(X_j \oplus k_{j+1} \oplus Y_j) \quad (29)$$

$$Z_{j+1} = CRC(X_{j+1} \oplus k_j \oplus Y_{j+1}) \quad (30)$$

From (27), we have:

$$k_j = k_j^* \oplus Y_j \oplus ID_{T_0} \oplus X_j \quad (31)$$

By substituting k_j from (31) in (30), we can write:

$$Z_{j+1} = CRC(k_j^* \oplus ID_{T_0} \oplus X_j \oplus X_{j+1} \oplus Y_j \oplus Y_{j+1}) \quad (32)$$

3. Now the adversary \mathcal{A} defines the maximum number of iterations as τ and follows the following steps to determine whether T_0 is the tag T . It should be noted that the same process can be used to determine whether T_1 is the tag T .

a) $c = 1$

b) computes:

$$k^* = PRNG^c(k_i^*) = \underbrace{PRNG(PRNG(\dots(k_i^*)\dots))}_{c \text{ times}}$$

c) computes

$$X_j = CRC(k^* \oplus N_{T_0}),$$

$$X_{j+1} = CRC(PRNG(k^*) \oplus N'_{T_0}).$$

d) computes $\Delta_X = X_j \oplus X_{j+1}, \Delta_Y = Y_j \oplus Y_{j+1}$.

e) If $Z_{j+1} \neq CRC(k^* \oplus ID_T \oplus \Delta_X \oplus \Delta_Y)$ and $c < \tau$ then $c = c + 1$ and go to b

f) Else \mathcal{A} outputs 0 i.e. $T_0 = T$ and $k_j^* = k^*$.

This attack shows that the current owner of tag T will be able to trace it at any time in future. Therefore, we can conclude that Chen *et al.*'s protocol lacks new owner privacy.

Remark 2. It should be noted that the procedure above will work when the number of iterations τ is less than the all possible values for the key k_j^* . This implies that if the length of key k_j^* is n , $\tau \ll 2^n$. So,

the tracing process will work efficiently unless the number of passed sessions are comparable to 2^n .

Remark 3. Any adversary of this kind who has already obtained k_j^* from the above procedure is also able to calculate k_{j+1} by (23). Then she will be able to extract t_{i+1} from the last message of the tag ownership transfer protocol by using (26). This results in a more dangerous attack in which the current owner is able to even *impersonate* the tag for future interrogations.

5 CONCLUSION

In this paper, we investigated the privacy of two ownership transfer protocols. The investigation included the attacks to target the forward and backward privacy as well as previous and new owner privacy properties. Our results showed both protocols are vulnerable to the attacks where the adversary is one of the owners in the system.

Any owner in the system as well as any adversary with the capability of tampering the tag are able to trace the tag in the previous and future interrogations in the ROTIV protocol. Therefore, this protocol lacks four stated privacy properties, forward privacy, backward privacy, previous owner privacy and new owner privacy.

Chen *et al.*'s protocol was also shown to be susceptible to the attacks in which the adversary is one of the previous owners of the tag and thus not to fulfil the forward privacy and new owner privacy. This protocol also revealed the whole tag's information to any previous owner and makes the adversary capable of impersonating the tag in further interrogations.

REFERENCES

- [1] G. Avoine, *Adversarial model for radio frequency identification*, Cryptology ePrint Archive, Report 2005/049, 2005.
- [2] S. Vaudenay, *On privacy models for RFID*. In Proceedings of the Advances in Cryptology 13th international conference on Theory and application of cryptology and information security, ASIACRYPT'07, Berlin, Heidelberg, 2007.
- [3] A. Juels and S.A. Weis. *Defining Strong Privacy for RFID*. In PerCom Workshops, White Plains, USA, 2007.

- [4] J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel, *A New RFID Privacy Model*, In European Symposium on Research in Computer Security (ESORICS 2011), Lecture Notes
- [5] A. Fernández-Mir, R. Trujillo-Rasua, J. Castellà-Roca, and J. Domingo-Ferrer, *A Scalable RFID Authentication Protocol Supporting Ownership Transfer and Controlled Delegation*, In the proceeding of RFIDSec2011, Amherst, USA, June 2011.
- [6] J. Saito, K. Imamoto, and K. Sakurai. *Reassignment scheme of an RFID tags key for owner transfer*. Embedded and Ubiquitous Computing, 2005.
- [7] D. Molnar, A. Soppera, and D. Wagner. *A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags*. In Selected Areas in Cryptography. Springer, 2006.
- [8] G. Avoine, E. Dysli, and P. Oechslin. *Reducing Time Complexity in RFID Systems*. In Bart Preneel and Stafford Tavares, editors, Selected Areas in Cryptography (SAC) 2005, volume 3897 of Lecture Notes in Computer Science, Kingston, Canada, August 2005.
- [9] A. Soppera and T. Burbridge. *Secure by default: The RFID acceptor tag (RAT)*, In: 2nd Workshop on RFID Security - RFIDSec, Graz, Austria, July 2006, IAIK TU Graz.
- [10] K. Osaka, T. Takagi, K. Yamazaki, and O. Takahashi. *An Efficient and Secure RFID Security Method with Ownership Transfer*. In Computational Intelligence and Security, 2006.
- [11] B. Song. *RFID Tag Ownership Transfer*. In Workshop on RFID Security - RFIDSec'08, Budapest, Hungary, July 2008.
- [12] P. Peris-Lopez, J. C. Hernandez-Castro, J. M.E. Tapiador, T. Li, and Y. Li. *Vulnerability analysis of rfid protocols for tag ownership transfer*. Computer Networks, 2010.
- [13] E.J. Yoon and K.Y. Yoo. *Two security problems of RFID security method with ownership transfer*. In Network and Parallel Computing, 2008.
- [14] G. Kapoor and S. Piramuthu. *Vulnerabilities in some recently proposed RFID ownership transfer protocols*. In 2009 First International Conference on Networks and Communications. IEEE, 2009.

- [15] B. Song and C.J. Mitchell. *Scalable RFID security protocols supporting tag ownership transfer*. Computer Communications, 2010.
- [16] K. Elkhyaoui, E. Blass, R. Molva. *ROTIV: RFID Ownership Transfer with Issuer Verification*, In the proceeding of RFIDSec2011, Amherst, USA, June 2011.
- [17] C. Chen, Y. Lai, C.Cheng Chen, Y. Deng, Y. Hwang, *RFID Ownership Transfer Authorization Systems Conforming EPCglobal Class-1 Generation-2 Standards*, International Journal of Network Security, Vol.13, No.1, July 2011.