On the security of Lo *et al.*'s ownership transfer protocol

Masoumeh Safkhani[1], Nasour Bagheri[2], Majid Naderi[1] and Ali Mahani[3]

[1] Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran. {M_Safkhani,M_Naderi}@iust.ac.ir

[2] Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. Nbagheri@srttu.ac.ir

[3] Department of Electrical and Electronic Engineering, Shahid Bahonar University of Kerman, Iran. AMahani@uk.ac.ir

Abstract. Recently Lo *et al.* have proposed an ownership transfer protocol for RFID objects using lightweight computing operators and claim their protocol provides stronger security robustness and higher performance efficiency in comparison with existing solutions. However, in this paper we show that their claim unfortunately does not hold. More precisely, we present tag's secret disclosure attack, new owner's secret disclosure and fraud attack against the Lo *et al.*'s ownership transfer protocol. The success probability of all attacks is "1" while the complexity is only one run of protocol. Our observation shows that this protocol compromise the privacy of the tag and the new owner.

Keywords: RFID, Ownership Transfer Protocol, Disclosure Attack, Fraud Attack.

1 Introduction

Radio Frequency Identification (or in short term RFID) has many applications for both business and private individuals. Several of these applications will include items that change owners at least once in their lifetime. The swapping and resale of items is a practice that is likely to be popular in the future, and so any item that depends on RFID for function or convenience should be equipped to deal with change of ownership.

In the case of ownership transfer, we want to prevent the old owner of a tag from accessing that tag or the services that the tag provides to the current owner. For instance, an old owner should not be able to use any keys or identifying values to impersonate the tag to make a purchase, open a door, or even change the ownership back to him. In summary, a secure ownership transfer protocol must satisfy the following properties:

- The old owner should not be able to access the tag after the ownership transfer has taken place.
- The new owner should be able to perform mutual authentication with the tag after the ownership transfer has taken place.

Ownership transfer presents its own set of threats and therefore demands the attention of security researchers. Observing that tags may become highly ubiquitous in the future, with tagged object changing hands often, secure owner transfer would be essential to the RFID systems. Hence, several ownership transfer protocols have been proposed in the literatures [1–3,5–7,9,10,13]. However, the later analysis [4,11,12] demonstrated their security flaws that shows that it is not an easy task to design a secure ownership transfer protocol.

1.1 Overview of the Current Work

Recently Lo *et al.* have proposed an ownership transfer protocol using lightweight computing operators [8]. They have claimed their protocol provides suitable security and performance in comparison to other schemes in this field. However, in this paper we provide a detailed security analysis of the protocol which leads to several efficient attacks against the protocol, i.e. tag's/new owner's secret disclosure attack and fraud attack.

Paper Organization: Notations used in the paper are presented in Section 2. Lo *et al.*'s ownership transfer protocol is described in Section 3.Section 4 and Section 5 describe our tag's secret disclosure attack and fraud attack respectively. Finally, we conclude the paper in Section 6.

2 Preliminaries

Throughout the paper, we use the following notations:
- TTP: The trusted third party server.
- O_C: The current owner of the tagged object.
- O_N: The new owner of the tagged object after ownership transfer.
- $\mathcal{A}$: Adversary.
- Tag_i : An RFID tag i.
- TID_i : Unique identifier of tag i.

- OID_j : The identifier of owner j.
- $TTPID$: The identifier of TTP.
- C_j : The credential of the owner j.
- KT_i : The shared secret key between Tag_i and TTP.
- KO_i : The shared secret key between Tag_i and O_C.
- $h()$: One way hash function.
- r_x: The x-th random number where $x \in Z^+$.
- $\oplus$: The exclusive-or operation.
- $Nun(m, n)$: A lightweight function which is built with bit-shift and add operations.

$Nun(m, n)$ function has two input values, m and n, where m is the value to be verified and n is a random number to reshuffle the original input value m. The pseudo-code of Nun is shown as below:

```
string Nun(bit string m, bit string n){
string X=m
int L=bit length of X value
for(int i=0;i<L;i++)
{
X=(X>>1)+(X<<1)+n
}
return X
}
```

3 Lo *et al.*'s Ownership Transfer Protocol Description

Lo *et al.*'s [8] protocol, has three phases including registration phase, current ownership suspension phase and new ownership establishment phase. The details of Lo *et al.* protocol is described as below:

- **Registration Phase** All tags will be initialized before using and all object owners have to register their identities at TTP. TTP shares a secret key KT_i with each tag i, where KT_i is preloaded into the memory of tag i and neither current owner nor new owner can access (or know) the key. Each owner j has a credential key $C_i = h(TTPID \| OIDj)$ such that TTP can verify the identity of requesting owner when receiving an ownership transfer request from an object owner. In addition, the current owner and the Tag_i are set to share the same secret key KO_i.
- **Current Ownership Suspension Phase** The details of current ownership suspension phase which is depicted in Fig. 1 are as follows:

1. O_C generates a random number r_1 and sends $(OID_C, OID_N, TID_i, r_1, C_C)$ to TTP.
2. Once receipt the message, TTP calculates $C'_C = h(TTPID||OID_C)$. If $C'_C \neq C_C$, TTP will terminate this session. Otherwise, TTP:
   - Finds corresponding KT_i for Tag_i,
   - Generates a temporary secret key K_{temp},
   - Computes $M_1 = KT_i \oplus K_{temp} \oplus r_1$ and a verification message $V_1 = Nun(M_1, r_1)$,
   - Sends M_1 and V_1 to O_C.
3. Upon receipt the message, O_C computes $M_2 = M_1 \oplus KO_i$ and forwards (V_1, M_2, r_1) to Tag_i through its reader and starts a timer to control the tag's responding time.
4. When Tag_i receives the message, extracts $M'_1 = M_2 \oplus KO_i$ and computes $V'_1 = Nun(M'_1, r_1)$. If $V'_1 = V_1$, Tag_i extracts K_{temp} as $K_{temp} = M_1 \oplus KT_i \oplus r_1$ and assignees it to its secret KO_i and sends $ACK_1 = Nun(KT_i, r_1) \oplus K_{temp}$ to O_C, as its response. However, if $V'_1 \neq V_1$, tag sends a random number r_2 as ACK_1.
5. If O_C receives the tag response in the appropriate time(before timeout occurs) it forwards ACK_1 to TTP. Otherwise, O_C tries to use the old key KO_i to access Tag_i. If O_C cannot access the tag, it sends $ACK_1 = OID_C$ to TTP. Otherwise, it restarts Step 4.
6. Once TTP receives the message, it verifies whether $ACK_1 = Nun(KTi, r1) \oplus K_{temp}$ or $ACK_1 = OID_C$. In the case of equality, TTP passes $ACK_2 =$ 'success' to O_C. In addition, TTP transfers K_{temp} and TID_i to O_N. Otherwise, TTP passes a failure message, $ACK_2 =$'go to step 4', to O_C to restart the process from Step 4 .

– **New Ownership Establishment Phase** The details of this phase which is depicted in Fig. 2 are as follows:

1. Once O_N receives K_{temp}, it does as follows:
   - Generates the new secret key $KO_{i_{new}}$ and random numbers r_3 and r_4,
   - Computes $M_3 = KO_{i_{new}} \oplus r3 \oplus KO_i$ and $V_2 = Nun(KO_{i_{new}}, r_4) \oplus KO_i$,
   - Sends (M_3, V_2, r_3, r_4) to Tag_i and starts a timer.
2. On receiving the message, Tag_i computes $KO'_{i_{new}} = M_3 \oplus r_3 \oplus KO_i$ and $V'_2 = Nun(KO'_{i_{new}}, r_4) \oplus KO_i$. If $V'_2 = V_2$, it assigns the extracted $KO'_{i_{new}}$ to KO_i and sends $ACK_3 = Nun(KT_i, r_4) \oplus KT_i$ to O_N. Otherwise, it generates a random number and assigns it to ACK_3 and sends it to O_N.

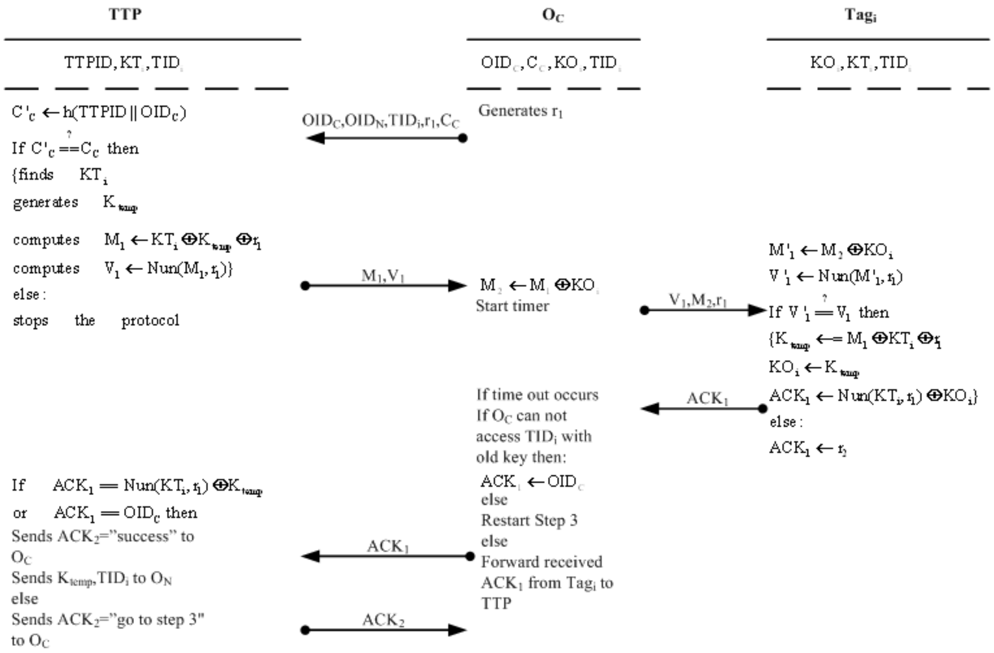

Fig. 1. The Current Ownership Suspension Phase in Lo *et al.*'s Ownership Transfer Protocol.

3. If O_N receives the tag's response in the appropriate time (before timeout occurs) it forwards ACK_3 to TTP. Otherwise, O_N tries to use the old key KO_i to access Tag_i. If O_N cannot access the tag, it sends $ACK_3 = OID_N$ to TTP. Otherwise, it restarts Step 1.
4. Once receipt the message, TTP verifies whether $ACK_3 = Nun(KT_i, r_4) \oplus KT_i$ or $ACK_3 = OID_N$. In the case of equality, TTP sends $ACK_4 =$'success' to O_N. Otherwise, TTP sends a restart message, $ACK_4 =$'restart', to O_N.
5. If O_N receives the success message, it replaces KO_i with $KO_{i_{new}}$. Otherwise, O_N restarts the procedure from Step 1.

Lo *et al.* have stated that the channel between the tag and the owner is insecure. However, they have not clarified whether the channel between owners and TTP is secure. If the channel between owners and TTP is insecure, it would be trivial to disclose the tag's secret key KO_i by computing $M_1 \oplus M_2$. Hence, in this paper we consider the channel between owners and TTP secure and we assume that the attacker cannot eavesdrop $(OID_C, OID_N, TID_i, C_C, M_1)$ and ACK_2.

4 Tag's and New Owner's Secrets Disclosure Attack

In Lo *et al.* ownership transfer protocol, Tag_i and TTP share a secret key KT_i which TTP employs it to transfer the tags ownership from O_C to O_N. KT_i is a secret parameter and neither O_C nor O_N are expected to be able to retrieve it. However, in this section we show that how a curious owner can disclose the secret of the tag, KT_i. In this attack O_C transfers the ownership of Tag_i to itself once and based on the transferred messages, it can retrieves KT_i. To extract KT_i, curious owner O_C can do as follow:

1. O_C starts a new ownership transfer protocol and sends $(OID_C, OID_C, TID_i, r_1, C_C)$ to TTP.
2. On receiving the message, TTP authenticates O_C and does as follows:
   - Finds corresponding KT_i for Tag_i,
   - Generates a temporary secret key K_{temp},
   - Computes $M_1 = KT_i \oplus K_{temp} \oplus r_1$ and a verification message $V_1 = Nun(M_1, r_1)$,
   - Sends $M_1 = KT_i \oplus K_{temp} \oplus r_1$ and V_1 to O_C.
3. Once O_C receives (M_1, V_1), it sends OID_C as ACK_1 to the TTP.
4. On receiving the message, since ACK_1 equals OID_C, TTP passes $ACK_2 =$ 'success' to O_C. In addition, TTP transfers K_{temp} and TID_i to O_N which is also O_C.

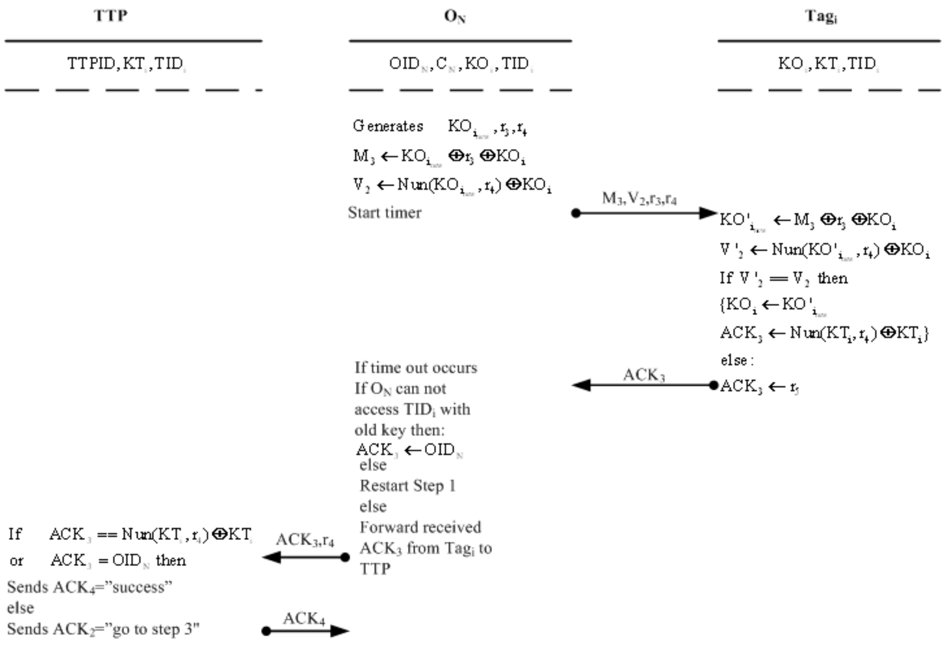

Fig. 2. The New Ownership Establishment Phase in Lo *et al.*'s Ownership Transfer Protocol.

5. Given M_1, r_1 and K_{tempt}, O_C extracts KT_i as $KT_i = M_1 \oplus K_{tempt} \oplus r_1$.

Hence, the curious owner O_C can disclose secret of the Tag_i, i.e. KT_i. The success probability of the given attack is "1" and the attack complexity is only one run of protocol. If KT_i is known by O_C, then whenever O_C transferred the ownership of Tag_i to a new owner O_C, O_C would be able to trace Tag_i which compromise the privacy of the new owner O_N. Even worse, the adversary can extract the secret key of the new owner $KO_{i_{new}}$ as follows:

1. O_C generates a random number r_1 and sends $(OID_C, OID_N, TID_i, r_1, C_C)$ to TTP.
2. Upon receipt the message, TTP authenticates O_C and does as follows:
   - Finds corresponding KT_i for Tag_i,
   - Generates a temporary secret key K_{temp},
   - Computes $M_1 = KT_i \oplus K_{temp} \oplus r_1$ and a verification message $V_1 = Nun(M_1, r_1)$,
   - Sends M_1 and V_1 to O_C.
3. Upon receipt the message, O_C computes $K_{temp} = M_1 \oplus KT_i \oplus r_1$.
4. The rest of the second phase of protocol continue as it has described in section 3.
5. Once O_N receives K_{temp}, it does as follows:
   - Generates the new secret key $KO_{i_{new}}$ and random numbers r_3 and r_4,
   - Computes $M_3 = KO_{i_{new}} \oplus r_3 \oplus KO_i$ and $V_2 = Nun(KO_{i_{new}}, r_4) \oplus KO_i$ and sends (M_3, V_2, r_3, r_4) to Tag_i.
6. O_C eavesdrops the transferred message (M_3, V_2, r_3, r_4) and does as follows :
   - Given K_{temp} from step 5 and the eavesdropped M_3 and r_3, computes $KO_{i_{new}}$ as $KO_{i_{new}} = M_3 \oplus r_3 \oplus K_{temp}$.
7. O_C does not disrupt the rest of the protocol and Tag_i and O_N finish the protocol successfully.

Hence, following the given attack, the old owner O_C can disclose the secret key of the new owner O_N. Given the owner secret key it would be easy to do what you want, trace the tag, impersonate the owner, etc. It must be noted that the success probability of the given attack is "1" and the attack complexity is only one run of protocol.

Remark 1. One may argue that to prevent the curious O_C to retrieve the tag's secret key and therefore the new owner's secret key, we can force TTP to not allow the owner to transfer the ownership to itself. However, then two person, e.g. O_C and O'_N, can collude to retrieve the tag secret and then transfer its ownership to the victim owner O_N.

5 Fraud Attack

In this section we present another attack against Lo *et al.*'s protocol in which the owner O_C can satisfy the new owner O_N, confirmed by TTP certification, while after ownership transformation O_N cannot access Tag_i but O_C can do. Hence, O_C can sell the same tag to several person. Whenever insincere O_C decides to sell the ownership of Tag_i to a victim owner O_N, it does as below:

1. O_C generates a random number r_1 and sends $(OID_C, OID_N, TID_i, r_1, C_C)$ to TTP.
2. On receiving the message, TTP authenticates O_C and does as follows:
   - Finds corresponding KT_i for Tag_i,
   - Generates a temporary secret key K_{temp},
   - Computes $M_1 = KT_i \oplus K_{temp} \oplus r_1$ and a verification message $V_1 = Nun(M_1, r_1)$,
   - Sends M_1 and V_1 to O_C.
3. Upon receipt the message, O_C does not send anything to Tag_i and just sends its OID_C as ACK_1 to the TTP.
4. When TTP receives the message, since ACK_1 equals OID_C, passes $ACK_2 =$ 'success' to O_C. In addition, TTP transfers K_{temp} and TID_i to O_N.
5. Once O_N receives K_{temp} and TID_i replaces KO_i by the new generated secret key $KO_{i_{new}}$. Then, it generates the tuple (M_3, V_2, r_3, r_4) following the given procedure in section 3 and sends it to Tag_i.
6. O_C blocks (M_3, V_2, r_3, r_4).
7. Since O_N does not receive the tag response in the appropriate time(before timeout occurs), O_N tries to use the old key KO_i to access Tag_i. However, O_N cannot access the tag, so it sends $ACK_3 = OID_N$ and r_4 to TTP,
8. Once receipt the message, TTP verifies $ACK_3 = OID_N$ and sends $ACK4 =$'success' to O_N.
9. When O_N receives a success message, it replaces KO_i with $KO_{i_{new}}$ and ensures that it has received the Tag_i ownership successfully.

Hence, following the given attack, the new owner beliefs that the ownership transfer has been done successfully and he is only one which can identify Tag_i and access the information inside the tag and the old owner cannot identify and control the tag any more. However, the given attack shows that a dishonest owner can deceive new owners, sell his tagged objects while he still has possession of them, because the tag has not updated its owner key yet. The success probability of our fraud attack is "1" while the complexity is only one run of protocol for each tag sell.

Remark 2. The given attack works properly if any owner which has a record in TTP decides to deceive a new owner O_N that it can transfer the ownership of Tag_i to O_N. It comes from this fact that in Lo *et al.*'s protocol, TTP does not check whether O_C has the permission to transfer the ownership of Tag_i and it just verifies whether O_C has a record in the TTP database, while that record can be not necessarily to access Tag_i.

6 Conclusion

In this paper we considered Lo *et al.*'s RFID ownership transfer protocol and demonstrated that, in contrary to its designers' claims, this protocol does not provide resistance tag's and new owner's secrets disclosure and fraud attacks. The success probability of all attacks are "1" while the cost of all is only one run of protocol. In fact, this paper shows how an adversary can use Lo *et al.*'s ownership transfer protocol's in his business to get rich, where it can sell his tagged objects to new owners while he still has possession of them and it can sell them to someone else.

References

1. C. Chen, Y. Lai, C. Chen, Y. Y. Deng, , and Y. C. Hwang. Rfid ownership transfer authorization systems conforming epcglobal class-1 generation-2 standards. In *International Journal of Network Security,*, volume 13, pages 41–48, 2011.
2. S. Fouladgar, , and H. Afifi. An efficient delegation and transfer of ownership protocol for rfid tags. In *The First International EURASIP Workshop on RFID Technology, Vienna, Austria*, 2007.
3. S. Fouladgar and H. Afifi. A Simple Privacy Protecting Scheme Enabling Delegation and Ownership Transfer for RFID Tags. In *JOURNAL OF COMMUNICATIONS*, volume 2, pages 1502–1508, 2007.
4. Gaurav Kapoor and Selwyn Piramuthu. Vulnerabilities in Some Recently Proposed RFID Ownership Transfer Protocols . In *Networks and Communications, 2009. NETCOM '09. First International Conference on*, pages 354–357, 2009.
5. G. Kapoor and S. Piramuthu. Single rfid tag ownership transfer protocols. In *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2010*, pages 1–10, 2010.
6. L. Kulseng, Z. Yu, Y. Wei, and Y. Guan. Lightweight Mutual Authentication and Ownership Transfer for RFID systems. In *The proceedings of IEEE INFOCOM 2010*, pages 1–5, March 2010.
7. H. Lei and T. Cao. Rfid protocol enabling ownership transfer to protect against traceability and dos attacks. In *The First International Symposium on Data, Privacy, and E-Commerce, Chengdu, China, 2007*, pages 508–510, 2007.

8. N.-W. Lo, S.-H. Ruan, and T.-C. Wu. Ownership transfer protocol for rfid objects using lightweight computing operators. In *6 th International Conference on Interney Technology and Secured Transactions,11–14 December 2011, Abu Dhabi, United Arab Emirates*, pages 484–489, 2011.
9. K. Osaka, T. Takagi, K. Yamazaki, , and O. Takahashi. An efficient and secure rfid security method with ownership transfer. In *International Conference on Computational Intelligence and Security, Guangzhou, China*, pages 1090–1095, 2006.
10. P. Peris-Lopez, J. Hernandez-Castro, J. Estevez-Tapiador, T. Li, and Y. Li. vulnerability analysis of rfid protocols for tag ownership transfer. In *Computer Networks*, volume 54, pages 1502–1508, 2010.
11. M. Safkhani, M. Naderi, N. Bagheri, and S. K. Sanadhya. Cryptanalysis of Some Protocols for RFID Systems. Cryptology ePrint Archive, Report 2011/061, 2011. `http://eprint.iacr.org/`.
12. Suleyman Kardas and Mete Akgun and Mehmet Sabr Kiraz and Huseyin Demirci. Cryptanalysis of Lightweight Mutual Authentication and Ownership Transfer for RFID Systems. In *2011 Workshop on Lightweight Security& Privacy: Devices, Protocols, and Applications*, pages 20–25, 2011.
13. Y. Zuo. Changing hands together: a secure group ownership transfer protocol for rfid tags. In *Hawaii International Conference on System Sciences, Koloa, Kauai, HI, 2010,*, pages 1–10, 2010.