Cryptanalysis and Enhancement of a Secure Group Ownership Transfer Protocol for RFID Tags

Hoda Jannati and Abolfazl Falahati

Department of Electrical Engineering (DCCS Lab)
Iran University of Science and Technology, Tehran
{hodajannati,afalahati}@iust.ac.ir

Abstract. Ownership transfer and grouping proof protocols are the two most important requirements for RFID tag in various applications such as pharmaceutical distribution and manufacturing. In 2010, Zuo integrated these two requirements and introduced a protocol for RFID tag group ownership transfer (GOT), i.e., transferring the ownership of a group of tags in one session. However, this paper shows that Zuo's protocol is vulnerable to de-synchronization attack and tag impersonating in the presence of cheating old owner. This paper also proposes solutions to fix the security flaws of Zuo's GOT protocol.

Keywords: de-synchronization attack, grouping proof, ownership transfer, RFID tags.

1 Introduction

Too much attention has recently been given to RFID systems because of the ease of its deployment over a wide range of applications. In fact, RFID systems have become very popular and concrete tools in various applications such as identifying, target tracking, sense ambient conditions of tagged objects, guarding patient safety and etc.; indeed there is an enormous growing for such system implementations [1]. Due to these so many advantages, a large number of research scientists have begun to improve RFID systems recently [2–4].

With the rapid development of RFID tags, different kinds of security requirements have been revealed within RFID communication network. In many applications, tag ownership transfer and grouping proofs with tag privacy, mutual authentication as well as data confidentiality are considered as the most critical requirements [5].

Furthermore, in many applications, an RFID tag may change its owner a number of times during its life cycle. Thus all information associated with the tag must be passed from the old owner to the new owner. Hence, in the secure tag ownership transfer protocol, the new owner privacy, the old owner privacy and the authorization recovery must be well satisfied [6–9].

Moreover, in 2004, Juels proposed a different concept which was called yoking proof or grouping proof [10]. According to his concept, a pair or group of RFID

H. Jahankhani et al. (Eds.): ICGS3/e-Democracy 2011, LNICST 99, pp. 186–193, 2012.

tags can generate a proof which certifies the same reading device to scan the tags simultaneously [11, 12]. Recently, the grouping proof protocol has been adopted to improve inpatient safety and can indeed avoid death due to medication related errors [13].

However, it is possible to transfer the ownership of a group of RFID tags one by one but it is inefficient and time consuming, and cannot ensure the simultaneous presence of multiple tags. In order to solve this problem, Zuo integrated ownership transfer and grouping proof protocols and introduced RFID tag group ownership transfer (GOT) protocol, i.e., transferring the ownership of a group of tags in one session [14].

However, this paper shows the Zuo's protocol has some security weaknesses in the presence of cheating old reader. Zuo's protocol is vulnerable to desynchronization attack. Under such attacks, a valid tag is identified as an illegal tag. Also, under certain circumstances, an attacker can obtain the secret key of the tags and impersonate them. These weaknesses are of importance here for further improvements.

Organization. The remainder of this paper is organized as follows: in Section 2, Zuo's group ownership transfer protocol is reviewed. Weaknesses of Zuo's protocol are discussed in Section 3. In Section 4, our improved protocol is described. Finally, we summarize our research in Section 5.

2 A Review of Zuo's Group Ownership Transfer (GOT) Protocol

Zuo proposed the Group Ownership Transfer (GOT) protocol in [14]. He assumed that there are n tags in the group whose ownership is to be transferred from the current owner to a new owner. For simplicity he illustrated his protocol with two tags, but the protocol can be extended to any number of tags.

There are three phases in Zuo's protocol: RFID tags identification phase, group ownership transfer phase and verification phase. In this section, we describe Zuo's GOT protocol. In order to describe Zuo's protocol, we will use the following notations:

- $S_{current}$: the server of the current owner,
- $R_{current}$: the reader of $S_{current}$,
- S_{new} : the server of the new owner,
- TS : the trusted server in the system,
- T_i : i^{th} tag,
- ID_i : the identification of T_i,
- $f(k, m)$: pseudorandom function taking seed k and message m,
- $E_k(m)$: message m encrypted with key k using standard cryptographic function, i.e., AES,
- k_{s1} : l-bits secret shared between $S_{current}$ and TS,
- k_{s2} : l-bits secret shared between S_{new} and TS,

- k_i : l-bits secret shared between T_i and owner,
- k_{group} : l-bits secret shared among the members of a group,
- s_i : l-bits secret shared between T_i and TS,
- k_{ss} : l-bits secret shared between $S_{current}$ and S_{new},
- N_R : a random nonce generated by $S_{current}$,
- N_T : a random nonce generated by TS,
- $H(\cdot)$: a secure one-way hash function,
- $\|$: the operation of concatenation,
- $\oplus$: the operation of Exclusive-OR (XOR),
- $\{,\}$: a set of elements.

- **RFID Tags Identification Phase:**

1. S_{new} submits a ownership transfer request and its credentials with an identification G_{id} to $S_{current}$ for ownership transfer over a group of tags.

2. $S_{current}$ evaluates the ownership transfer request, S_{new}'s credentials and condition of ownership. If the based business transaction is authorized, the ownership transfer request will be honored. Then, $R_{current}$ scans the tags in its field and collects their IDs. Next, $S_{current}$ confirms that all the tags in the group are present and sends an acknowledgement message ACK to S_{new} which includes IDs of the tags in the group.

- **Group Ownership Transfer Phase:**

1. For $1 \leq i \leq 2$: S_{new} chooses a new secret key k_{i-new} to be shared with tag T_i and a new group key $k_{group-new}$ to be shared among the members of the group. Then, S_{new} randomly chooses k_{i-mask} and $k_{group-mask}$ and compute $M_{1,i}$ according to (1).

$$M_{1,i} = \{ID_i \| (k_{i-new} \oplus k_{i-mask}) \| (k_{group-new} \oplus k_{group-mask}) \\ \| E_{k_{s2}}(k_{i-mask}) \| E_{k_{s2}}(k_{group-mask})\} \tag{1}$$

After that, S_{new} sends $E_{k_{ss}}(M_{1,1})$, $E_{k_{ss}}(M_{1,2})$ and G_{id} to $S_{current}$.

2. For $1 \leq i \leq 2$: $S_{current}$ checks ID_i in message $M_{1,i}$. If so, $S_{current}$ constructs $M_{2,i}$ according to (2).

$$M_{2,i} = \{ID_{snew} \| ID_{scurrent} \| ID_i \| E_{k_{s2}}(k_{i-mask}) \| E_{k_{s2}}(k_{group-mask})\} \tag{2}$$

Then, $S_{current}$ sends $E_{k_{s1}}(M_{2,1})$ and $E_{k_{s1}}(M_{2,2})$ to TS.

3. TS randomly chooses N_T. Then, For $1 \leq i \leq 2$: TS checks ID_{snew}, $ID_{scurrent}$ and ID_i in message $M_{2,i}$. If so, TS applies k_{s2} to retrieve k_{i-mask} and $k_{\ group-mask}$ by performing the decryption function on

$E_{k_{s2}}$ (k_{i-mask}) and $E_{k_{s2}}(k_{group-mask})$ respectively. Then, TS constructs $M_{3,i}$ according to (3).

$$M_{3,i} = \{ID_i \| (f(s_i, N_T) \oplus k_{i-mask}) \| (f(s_i, N_T) \oplus k_{group-mask})\} \quad (3)$$

Finally, TS Sends $E_{k_{s1}}(M_{3,1})$, $E_{k_{s1}}(M_{3,2})$ and N_T to $S_{current}$.

4. $S_{current}$ randomly chooses N_R and transfers all necessary information to $R_{current}$ in a secure way, so that $R_{current}$ can interact with each tag in the group.

5. For $1 \leq i \leq 2$: $R_{current}$ constructs $M_{4,i}$ according to (4),

$$M_{4,i} = \{(f(s_i, N_T) \oplus k_{i-mask}),\ C_{1,i}\ , (f(s_i, N_T) \oplus k_{group-mask}),\ C_{2,i},\ (k_{i-new} \oplus k_{i-mask}),\ C_{3,i}\ , (k_{group-new} \oplus k_{group-mask}),\ C_{4,i}\} \quad (4)$$

where $C_{j,i}$ is a credential on j -th message, i.e., the credential on $f(s_i, N_T)$ $\oplus\, k_{i-mask}$ is computed as $C_{1,i} : \{N_{1,i} = f(k_i, N_R) \oplus f(s_i, N_T) \oplus k_{i-mask}$ and $N_{2,i} = H((f(s_i, N_T) \oplus k_{i-mask} \oplus N_R) \| k_i)\}$.

Then, $R_{current}$ sends $M_{4,1}$ and N_R to T_1 and also, $M_{4,2}$ and N_R to T_2.

6. T_1 verifies the credentials in $M_{4,1}$. If so, T_1 constructs $M_6 = f(k_{group}, N_R \| c)$ and $M_7 = f(k_1, N_R \oplus c)$, where c represents a counter set by T_1. Then, T_1 sends M_6, M_7 and c to $R_{current}$.

7. $R_{current}$ sends M_6 and c to T_2.

8. T_2 verifies M_6. If so, T_2 knows that it is interacting with a tag in the same group. Then, it performs the following operations to update its new keys:

   - Apply s_2 to retrieve k_{2-mask} and $k_{group-mask}$ by performing XOR operations on $f(s_2, N_T)$ and the received messages $f(s_2, N_T) \oplus k_{2-mask}$ and $f(s_2, N_T) \oplus k_{group-mask}$ in message $M_{4,2}$ respectively.

   - Apply k_{2-mask} and $k_{group-mask}$ to retrieve the new secret key k_{2-new} and the new group key $k_{group-new}$ by performing XOR operations on k_{2-mask} and $k_{group-mask}$ and the received messages $(k_{2-new} \oplus k_{2-mask})$ and $(k_{group-new} \oplus k_{group-mask})$ in message $M_{4,2}$ respectively.

   Then, T_2 computes $M_8 = \{f(k_{2-new}, N_R \| c) || f(k_2, N_R \oplus c)\}$ and $M_9 = f(k_{group-new}, N_R \oplus c)$ and sends M_8 and M_9 to $R_{current}$.

9. $R_{current}$ sends M_9 to T_1.

10. T_1 performs the following operations to update its new keys in a similar way with T_2. Then, T_1 verifies M_9. If so, T_1 knows that it is interacting with a tag in the same group and computes M_{10} according to (5) and sends it to $R_{current}$. Finally, T_1 updates $c = c + 1$.

$$M_{10} = \{f(k_{1-new}, N_R \| c) \| f(k_{group-new}, N_R \oplus c)\} \tag{5}$$

11. $R_{current}$ constructs a group ownership transfer proof message $M_{11} = \{f(k_{1-new}, N_R \| c) \| f(k_{group-new}, N_R \oplus c) \| f(k_{2-new}, N_R \| c)\}$. Then it is forwarded to S_{new} for verification.

– **Verification Phase:**

At this stage, S_{new} verifies M_{11}. It is supposed that all the tags in the group have already updated their secret keys as set by the new owner. Then, as the final step of a complete group ownership transfer process, S_{new} conducts a challenge response process using a grouping-proof protocol or using a tag-reader authentication protocol.

3 Weaknesses of Zuo's GOT Protocol

Unfortunately Zuo's GOT Protocol described above is completely insecure in the presence of cheating old owner. In this section, we propose several attacks to Zuo's protocol.

– *De-synchronization attack:*
In this attack we assume that the protocol has been performed till step 4 in group ownership transfer phase. So, $S_{current}$ knows messages $M_{1,1}$ and $M_{1,2}$, therefore it knows:

$$k_{1-new} \oplus k_{1-mask} \tag{6}$$

$$k_{2-new} \oplus k_{2-mask} \tag{7}$$

$$k_{group-new} \oplus k_{group-mask} \tag{8}$$

Also, $S_{current}$ knows messages $M_{3,1}$ and $M_{3,2}$, therefore it knows:

$$f(s_1, N_T) \oplus k_{1-mask} \tag{9}$$

$$f(s_2, N_T) \oplus k_{2-mask} \tag{10}$$

$$f(s_1, N_T) \oplus k_{group-mask} \tag{11}$$

$$f(s_2, N_T) \oplus k_{group-mask} \tag{12}$$

By performing XOR operations on (6) and (9), $S_{current}$ obtains:

$$f(s_1, N_T) \oplus k_{1-new} \tag{13}$$

By performing XOR operations on (7) and (10), $S_{current}$ obtains:

$$f(s_2, N_T) \oplus k_{2-new} \tag{14}$$

By performing XOR operations on (11) and (13), $S_{current}$ obtains:

$$k_{group-mask} \oplus k_{1-new} \tag{15}$$

By performing XOR operations on (12) and (14), $S_{current}$ obtains:

$$k_{group-mask} \oplus k_{2-new} \tag{16}$$

And also, by performing XOR operations on (15) and (16), $S_{current}$ obtains:

$$k_{1-new} \oplus k_{2-new} \tag{17}$$

Now in step 5 in group ownership transfer phase, $R_{current}$ sends $f(s_1, N_T) \oplus k_{1-new}$ and $(k_{1-new} \oplus k_{2-new})$ instead of $f(s_1, N_T) \oplus k_{1-mask}$ and $(k_{1-new} \oplus k_{1-mask})$ in message $M_{4,1}$ to T_1. Also, $R_{current}$ sends $f(s_2, N_T) \oplus k_{2-new}$ and $(k_{1-new} \oplus k_{2-new})$ instead of $f(s_2, N_T) \oplus k_{2-mask}$ and $(k_{2-new} \oplus k_{2-mask})$ in message $M_{4,2}$ to T_2.

Therefore, in step 8 in group ownership transfer phase, T_2 retrieves k_{1-new} and in step 10 in group ownership transfer phase, T_1 retrieves k_{2-new} instead of their new secret keys. But, the new owner stores k_{1-new} for the new secret key of T_1 and k_{2-new} for the new secret key of T_2 in its data base. Such an attack on a tag causes loss of synchronization between the tag and the new owner. Later, when tags want to use their keys, the reader identifies tags as illegal tags. Note that, in verification phase, $R_{current}$ must change messages sent on behalf of the new owner to tags.

– *Obtain the secret key of the tag:*
When there is a group of tags, the members of the group have a group key which is common among the members of the group but they do not access the secret key of each other. From (17), it is known that new secret keys of the tags relate to each other. It is a serious problem. Because, $S_{current}$ can obtain $k_{1-new} \oplus k_{2-new}$ and T_1 knows k_{1-new}. So, if $S_{current}$ and T_1 conspire, they can obtain the secret key of T_2, i.e., k_{2-new}. Therefore, $S_{current}$ and T_1 can impersonate T_2.

Also, this attack can be performed on the T_1 too similar to T_2. If $S_{current}$ and T_2 conspire, they can obtain the secret key of T_1, i.e., k_{1-new}. Therefore, $S_{current}$ and T_2 can impersonate T_1.

4 The Improved Zuo's GOT Protocol

Some vulnerabilities of GOT protocol that have been employed through the above attacks are as the random number N_T used in $f(s_i, N_T) \oplus k_{group-mask}$

is the same as that used in $f(s_i, N_T) \oplus k_{i-mask}$. Also, the ID of tags has no impact on the computation of M_8, M_9 and M_{10}.

In this section, we improve GOT protocol to overcome against described attacks. In fact, the only changes of the GOT protocol are described in this section are summarized as follows:

- In step 3 in group ownership transfer phase, TS must randomly choose two numbers N_{T1} and N_{T2} and compute $M_{3,i} = \{ID_i \parallel (\ f(s_i, N_{T1}) \oplus k_{i-mask}\ ) \parallel (f(s_i, N_{T2}) \oplus k_{group-mask})\}$ and send N_{T1} and N_{T2} to $S_{current}$ along with $E_{k_{s1}}(M_{3,1})$ and $E_{k_{s1}}(M_{3,2})$.

- In step 5 in group ownership transfer phase, $R_{current}$ must construct $M_{4,i}$ in form of $M_{4,i} = \{f(s_i, N_{T1}) \oplus k_{i-mask},\ C_{1,i},\ f(s_i, N_{T2}) \oplus k_{group-mask},\ C_{2,i},\ (k_{i-new} \ \oplus \ k_{i-mask}),\ C_{3,i}, (k_{group-new} \oplus k_{group-mask}), C_{4,i}\}$.

- In step 8 in group ownership transfer phase, T_2 must compute M_8 and M_9 in the form of $M_8 = \{f(k_{2-new}, N_R \| c \| ID_2) \| f(k_2, N_R \oplus c)\}$ and $M_9 = f(k_{group-new}, N_R \oplus c \oplus ID_2)$ respectively.

- In step 10 in group ownership transfer phase, T_1 must compute M_{10} in form of $M_{10} = \{f(k_{1-new}, N_R \| c \| ID_1) \| f(k_{group-new}, N_R \oplus c \oplus ID_1)\}$.

These modifications strengthen the security of GOT protocol against the mentioned weaknesses.

5 Conclusion

In 2010, Zuo integrated two important requirements for RFID tags (tag ownership transfer and grouping proof protocols) and introduced a protocol for RFID tag group ownership transfer (GOT). In this paper, it is shown that Zuo's protocol has some security weaknesses in the presence of cheating old owner. Zuo's protocol suffers from de-synchronization attack and tag impersonating. Under these kinds of attacks, a valid tag is identified as an illegal tag. Also, under certain circumstances, an attacker can obtain the secret key of the tags. Here, we improved Zuo's GOT protocol to overcome such weaknesses.

References

1. Finkelzeller, K.: The RFID Handbook, 2nd edn. John Wiley-Sons (2003)
2. Han, D., Kwon, D.: Vulnerability of an RFID Authentication Protocol Conforming to EPC Class 1 Generation 2 Standards. Computer Standards and Interfaces 31(4) (2009)
3. Rizomiliotis, P., Rekleitis, E., Gritzalis, S.: Security Analysis of the Song-Mitchell Authentication Protocol for Low-cost RFID Tags. IEEE Communications Letters 13(4) (2009)

4. Jannati, H., Falahati, A.: Cryptanalysis and Enhancement of two Low Cost RFID Authentication Protocols. International Journal of UbiComp (IJU) 3(1), 1–9 (2012)
5. Fouladgar, S., Afifi, H.: A Simple Privacy Protecting Scheme Enabling Delegation and Ownership Transfer for RFID Tags. Journal of Communications 2(6), 6–13 (2007)
6. Chen, C.-L., Chen, Y.-Y., Huang, Y.-C., Liu, C.-S., Lin, C.-I., Shih, T.-F.: Anti-Counterfeit Ownership Transfer Protocol for Low Cost RFID System. WSEAS Transactions on Computers 7(8), 1149–1158 (2008)
7. Kapoor, G., Piramuthu, S.: Vulnerabilities in Some Recently Proposed RFID Ownership Transfer Protocols. IEEE Communications Letters 14(3), 260–262 (2010)
8. Alaraj, A.-M.: Ownership Transfer Protocol. In: IEEE International Conference for Internet Technology and Secured Transactions (ICITST 2010), pp. 1–6 (2010)
9. Li, T., Jin, Z., Pang, C.: Secured Ownership Transfer Scheme for Low-Cost RFID Tags. In: IEEE International Conference on Intelligent Networks and Intelligent Systems (ICINIS 2010), pp. 584–587 (2010)
10. Juels, A.: Yoking proofs for RFID Tags. In: Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, Washington, DC, USA, pp. 138–143 (2004)
11. Chien, H.-Y., Liu, S.-B.: Tree-based RFID yoking proof. In: IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC 2009), pp. 550–553 (2009)
12. Lopez, P.-P., Orfila, A., Castro, J.-C.-H., van der Lubbe, J.-C.-A.: Flaws on RFID Grouping-Proofs, Guidelines for Future Sound Protocols. Journal of Network and Computer Applications 34(3) (2011)
13. Yu, Y.-C., Hou, T.-W., Chiang, T.-C.: Low Cost RFID Real Lightweight Binding Proof Protocol for Medication Errors and Patient Safety. Journal of Medical Systems (2010), doi:10.1007/s10916-010-9546-4
14. Zuo, Y.: Changing Hands Together: A Secure Group Ownership Transfer Protocol for RFID Tags. In: The 43rd IEEE Hawaii International Conference on System Sciences (HICSS 2010), Honolulu, HI, pp. 1–10 (2010)