On the Security of Two Ownership Transfer Protocols and Their Improvements

Nasour Bagheri[1], Farhad Aghili[1], and Masoumeh Safkhani[2]
[1]Electrical Engineering Department, Shahid Rajaee Teacher Training University, Iran
[2]Computer Engineering Department, Shahid Rajaee Teacher Training University, Iran

Abstract: *In recent years, Radio Frequency Identification (RFID) systems are widely used in many applications. In some applications, the ownership of an RFID tag might change. To provide a solution, researchers have proposed several ownership transfer protocols based on encryption functions for RFID-tagged objects. In this paper, we consider the security of Kapoor and Piramuthu [3] ownership transfer protocol and Kapoor et al. [4] ownership transfer protocol. More precisely, we present de-synchronization attacks against these protocols. The success probability of all attacks is 1 while the complexity is only two runs of protocol. Finally, we present our suggestions to improve the security of these protocols.*

Keywords: *RFID, cryptanalysis, ownership transfer protocol, de-synchronization attack.*

Received February 4, 2014; accepted December 23, 2015

1. Introduction

Radio Frequency Identification (RFID) uses radio frequency signals to identify objects and humans. RFID has impacted many service industries in areas such as logistics, manufacturing and supply chain management [5, 11]. An RFID system commonly is composed of two main components: an RFID tag and an RFID reader. The tag is attached to a product and an RFID reader communicates with the tag to identify the product. At products flow through a supply chain [5], their ownership is transferred from an owner to another owner. This transfer of ownership extends to RFID tags attached to these products also. A desirable Ownership Transfer Protocol (OTP) should provide the security requirements for RFID tag ownership transfer [14]. Such a protocol should meet at least the following security and privacy conditions [12]:

1. When the ownership of a tag has been transferred to a new owner, only the new owner can identify the tag and has access to the information inside the tag and the old owner cannot identify and control the tag any more.
2. After the ownership of a tag has been transferred to a new owner, the new owner should not be able to trace back the past interactions between the tag and its previous owners.

1.1. Related Work

Several OTPs have been proposed in the literatures, e.g., [1, 2, 3, 4, 6, 7, 8, 15, 16, 17]. A majority of these protocols have been developed for the single-tag, single-owner e.g., [1, 2, 3, 6, 7, 8, 15] and some of these protocols have been proposed for multi-tag and multi-owner e.g., [4, 16,17] applications. Also, some of these protocols have used a Trusted Third Party (TTP) which acts as a secure channel to transfer some information between the entities e.g., [3, 4, 9, 13].

Munilla *et al*. [10] have presented an attack on Kapoor and Piramuthu [3] and Kapoor *et al*. [4] OTPs which puts a tag in a de-synchronized state. In their attack, they have assumed that upon receiving the message from TTP, the tag can update its secret key from s_1 (old owner R_1) to s_2 (new owner R_2) and loses s_1. With this assumption, they have shown that if an adversary intercepts the pair $(N_P, f'_{(N_P \oplus t_i \oplus s_1)}(s_2))$ which has been sent from TTP to the tag and replaces it by any desired pair, which is not equal to the pair $(N_P, f'_{(N_P \oplus t_i \oplus s_1)}(s_2))$, the tag is cheated to update its current secret to $s_2^{'}$ which does not match with any record of TTP. Without loss of generality, somebody may assume that the tag loses s_1 when it receives the message from the new owner (R_2). In this paper, based on this assumption; we describe that how an adversary can put a tag in a de-synchronized state.

1.2. Paper Organization

In section 2, we briefly describe Kapoor and Piramuthu [3] OTP; then we explain our de-synchronization attack against it and also our suggestions to improve its security. We also explain Kapoor *et al*. [4] protocol, our de-synchronization attack against it and our suggestions for its improvement in section 3. Finally, section 4 concludes the paper.

2. De-Synchronization Attack on Kapoor and Piramuthu̓ s Ownership Transfer Protocol

Recently Kapoor and Piramuthu [3] have proposed an ownership transfer protocol (KP protocol) with TTP which is shown in Figure 1. Throughout, the paper, we use the notations which are depicted in Table 1. The KP protocol is accomplished as below:

Table 1. Notations.

Notation	Description
TTP	Trusted third party server
Tag$_i$	RFID tag i
f$_k$	Keyed (with key k) encryption function
N$_J$,N$_{IJ}$	Random l-bit random numbers generated by entity J
R$_i$	Reader/Owner i
r$_i$	Shared secret between reader R$_i$ and TTP
s$_i$	Shared secret key between R$_i$, Tag$_i$ and TTP
t$_i$	Shared secret between Tag$_i$ and TTP

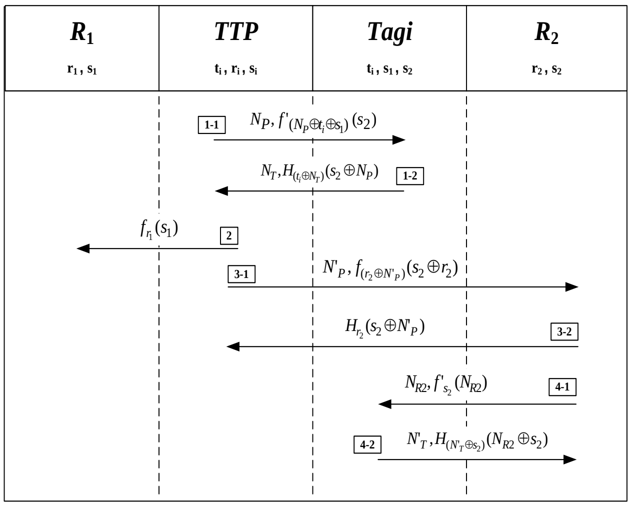

R1
r1, s1
TTP
ti, ri, si
Tagi
ti, s1, s2
R2
r2, s2
1-1
$N_P, f'_{(N_P \oplus t_i \oplus s_1)}(s_2)$
$N_T, H_{(t_i \oplus N_T)}(s_2 \oplus N_P)$
1-2
$f_{r_1}(s_1)$
2
3-1
$N'_P, f_{(r_2 \oplus N'_P)}(s_2 \oplus r_2)$
$H_{r_2}(s_2 \oplus N'_P)$
3-2
$N_{R2}, f'_{s_2}(N_{R2})$
4-1
4-2
$N'_T, H_{(N'_T \oplus s_2)}(N_{R2} \oplus s_2)$

Figure 1. The KP protocol [3].

- *Step 1-1.* (TTP→Tag): When TTP receives the ownership transfer request; it generates a random number N$_P$ and a secret key s$_2$. Then TTP computes $f'_{(N_P \oplus t_i \oplus s_1)}(s_2)$ and sends $(N_P, f'_{(N_P \oplus t_i \oplus s_1)}(s_2))$ to the tag, where s$_1$ is the tag's current key and t$_i$ is the secret shared between the tag and TTP. This message authenticates TTP to the tag and also updates s$_1$ to s$_2$.
- *Step 1-2.* (Tag→TTP): The tag responds by generating another random number N$_T$ and sending $(N_T, H_{(t_i \oplus N_T)}(s_2 \oplus N_P))$ to TTP.
- *Step 2.* (TTP→R$_1$): TTP sends a revoke message $(f_{(r_1)}(s_1))$ to R$_1$ and informs him that his privileges on Tag$_i$ are being revoked.
- *Step 3-1.* (TTP→R$_2$): TTP generates a random number N'$_P$ and sends a grant message $(N'_P, f'_{(r_2 \oplus N'_P)}(s_2 \oplus r_2))$ to the new owner (R$_2$) and grants him full permissions along with privileges for the tag.
- *Step 3-2.* (R$_2$→TTP): The new owner (R$_2$) uses a one-way hash function and replies to TTP by sending $H_{r_2}(s_2 \oplus N'_P)$, where s$_2$ is the new key value.
- *Step 4-1.* (*R*$_2$→Tag): The new owner (R$_2$) then establishes contact with the tag by generating a fresh random number N$_{R2}$, and sending $(N_{R2}, f'_{(s_2)}(N_{R2}))$ to the tag.
- *Step 4-2.* (Tag→*R*$_2$): The tag generates a fresh random number N'_T and acknowledges that the message is correct by sending $(N'_T, H_{(N'_T \oplus s_2)}(N_{R2} \oplus s_2))$ to the new owner (R$_2$).

2.1. De-Synchronization Attack

As described before, in the KP protocol, the tag can update its secret key from s$_1$ (old owner R$_1$) to s$_2$ (new owner R$_2$) and loses s$_1$ when it receives the message from TTP (after step 1-1) or when it receives the message from the new owner (R$_2$) (after step 4-1). If we assume that the tag updates its secret key and loses s$_1$ when it receives the message from TTP, the tag is de-synchronized forever and neither TTP nor the owners can access it anymore [10]. If we assume that the tag loses s$_1$ when it receives the message from the new owner (R$_2$), it would be vulnerable to the window attack in which, for a fraction of time, both current and new owners could access to the tag. On the other hand, an adversary can put the tag in a de-synchronized state. In this section, we describe a de-synchronization attack based on this assumption that the tag loses s$_1$ when it receives the message from the new owner (R$_2$). The attack's procedure is described as below:

- *Step 1.* The adversary eavesdrops the pair $(N_P, f'_{(N_P \oplus t_i \oplus s_1)}(s_2))$ sent by TTP and the pair $(N_{R2}, f'_{(s_2)}(N_{R2}))$ sent by R$_2$.
- *Step 2.* The adversary blocks the pair $(N'_T, H_{(N'_T \oplus s_2)}(N_{R2} \oplus s_2))$ sent from the tag to the new owner (R$_2$), now the tag has a new key s$_2$, but R$_2$ is waiting for an acknowledgement.
- *Step 3.* Now, R$_2$ does not accept the tag ownership transfer. Hereon, the protocol must be started again, as TTP sends $(N*_P, f'_{(N*_P \oplus t_i \oplus s_1)}(s_2))$ to the tag, now the tag acknowledges by generating a random nonce N$^*_T$ and sending $(N*_T, H_{(t_i \oplus N*_T)}(s'_2 \oplus N*_P))$. Then TTP again sends a revoke message $(f_{(r_1)}(s_1))$ to R$_1$ and informs R$_1$ that his privileges on Tag$_i$ are being revoked. Next, TTP sends the grant message to R$_2$, along with a freshly generated random number $N'*_P$ by sending $(N'*_P, f'_{(r_2 \oplus N'*_P)}(s'_2 \oplus r_2))$.
- *Step 4.* The adversary blocks the acknowledgement sent from R$_2$ to TTP and uses messages $(N_P, f'_{(N_P \oplus t_i \oplus s_1)}(s_2))$ and $(N_{R2}, f'_{(s_2)}(N_{R2}))$ that have eavesdropped before and sends them respectively to the tag. The tag is thus cheated to

update its current secret to (s_2). Thus, neither TTP records (new (s'_2) and old (s_1)) nor R_2's key (s'_2) are equal to the new tag's key (s_2). In fact, if TTP wants to take remedial action, it needs the tag's secret, which was just updated to s_2 due to the above attack, but TTP does not know s_2. Hence, the tag is de-synchronized forever and neither TTP nor the owners can access it any more. The success probability of this attack is 1 while the complexity is two runs of the protocol.

2.2. Improved KP Protocol

In the KP protocol [3], TTP sends the message $(N'_P, f'_{(r_2 \oplus N'_P)}(s_2 \oplus r_2))$ with a freshly generated random number N'_P and R_2 generates a fresh random number N_{R2} and establishes contact with the tag by sending $(N_{R2}, f'_{(s_2)}(N_{R2}))$. To improve the protocol's security, we replace N'_P by N_{PT} (where $N_{PT} = N'_P \oplus N_T$) and N_{R2} by N_{TR2} (where $N_{TR2} = N_T \oplus N_{R2}$). The improved protocol of KP is depicted in Figure 2. Further modifications to improve the KP protocol are described as below:

- *Step 1-1.* (TTP$\rightarrow$Tag): Upon receiving an ownership transfer request, TTP generates a random number N_P and a secret key s_2 and sends the tuple $(N_P, f_{(N_P \oplus t_i \oplus s_1)}(s_1 || s_2))$ to the tag. This message authenticates TTP to the tag and also updates s_1 to s_2.
- *Step 1-2.* (Tag$\rightarrow$TTP): The tag generates a random number N_T and responds by sending $(N_T, f_{(t_i \oplus N_T)}(s_2 \oplus N_P))$ to TTP.
- *Step 2.* (TTP$\rightarrow R_1$): TTP sends $(f_{(r_1)}(s_1))$ to R_1 and informs him that his privileges on Tag$_i$ are being revoked.
- *Step 3-1.* (TTP$\rightarrow R_2$): TTP computes ($N_{PT} = N'_P \oplus N_T$) and sends the grant message $(N_{PT}, f_{(r_2 \oplus N_{PT})}(s_2 || (r_2 \oplus N'_P)))$ to R_2 and grants the new owner (R_2) full permissions along with any delegation privileges for the tag.
- *Step 3-2.* ($R_2 \rightarrow$TTP): The new owner (R_2) sends $(f_{r_2}(s_2 \oplus N'_P))$ to TTP which is based on the new key value s_2.
- *Step 4-1.* ($R_2 \rightarrow$Tag): The new owner uses N_{PT} to decrypt the value sent by TTP in Step 3-1 $(N_{PT}, f_{(r_2 \oplus N_{PT})}(s_2 || (r_2 \oplus N'_P)))$ and determines N_T by XORing N_{PT} with N'_P. Then it generates a fresh random number N_{R2}, computes ($N_{TR2} = N_T \oplus N_{R2}$) and establishes contact with the tag by sending $(N_{TR2}, f_{(s_2)}(N_{R2}))$.
- *Step 4-2.* (Tag$\rightarrow R_2$): The tag acknowledges with a fresh random number N'_T along with N_{R2} and s_2 by sending $(N'_T, f_{(N'_T \oplus s_2)}(N_{R2} \oplus s_2))$. Also, the tag determines N_T by XORing N_{R2} with N_{TR2}.

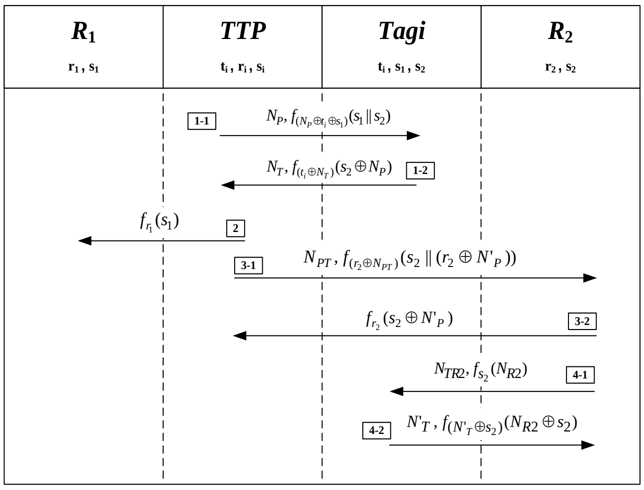


Figure 2. The improved KP protocol.

2.3. Security Analysis of the Improved KP Protocol

- *Secrecy/Data Integrity*: In the improved KP protocol, using cryptographic functions (f_k) guarantees the secrecy of the message. Without knowing keys, the adversary is not able to read the improved protocol's encrypted messages.
- *DoS/Synchronization Problem*: Proposed OTP sends the messages with a freshly generated random numbers (N_J, N_{IJ}). It means that blocking any message creates no breach in the system. In addition, we do not face the de-synchronization problem.
- *Forward Secrecy*: After the ownership has been transferred, the new owner (R_2) will not be able to decrypt the messages which were transmitted between the tag and its old owner (R_1) because the new owner (R_2) is never allowed to know the old keys shared between the tag and its old owner. Thus, we ensure the forward secrecy in the improved KP ownership transfer protocol.
- *Passive Replay*: In the improved KP protocol, each encrypted message contains a random value in each session. So an attacker cannot utilize eavesdropped messages. If an adversary eavesdrops messages which were exchanged between entities and then replays these messages to the tag, the tag will recognize these messages are invalid ones and will drop them. Because the random number of each message must be different in every use.
- *Windowing Problem*: In the improved KP protocol, TTP sends the message $f_{(N_{PT} \oplus t_i \oplus s_1)}(s_1 || s_2)$ to the tag and then the tag updates s_1 to s_2, so we cannot find a fraction of time, in which, both the new owner (R_2) and the old owner (R_1) can access the tag during ownership transaction. The serious

security features of KP protocol and improved KP protocol are compared in Table 2.

Table 2. Security features comparison between the KP protocol and the improved KP protocol.

Protocol / Feature	KP protocol	Improved KP Protocol
Secrecy	Yes [3]	Yes
Synchronization	No [3]	Yes
Forward Secrecy	Yes [3]	Yes
Passive Replay	Yes [3]	Yes
Resistance Against Windowing Problem	No [3]	Yes

2.4. Performance Analysis of the Improved KP Protocol

Performance comparison of the KP protocol and the improved KP protocol is depicted in Table 3. In this table "L", "H" and "f(.)" denote the bit length parameters, hash function and encryption function respectively. We evaluate the performance of the improved KP protocol theoretically. It should be noted that, due to the resources limitation of low-cost RFID tags, using hash function and encryption function together does not sound practically feasible.

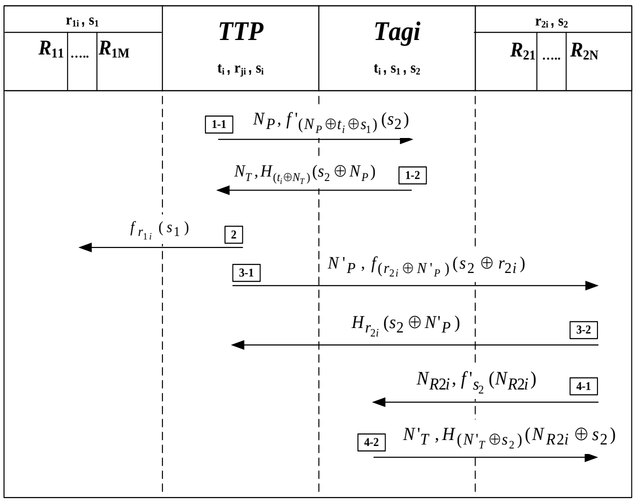
Table 3. Performance comparison between the KP protocol and the improved KP protocol.

#of Operations	KP Protocol [1]	Improved KP Protocol
#⊕	9	9
#H	3	0
#Transferred Bits	L	2L
#Random Numbers	5	5
#f(.)	4	7

3. De-synchronization Attack on Kapoor Ownership Transfer Protocol

Recently Kapoor *et al.* [4] have proposed a shared ownership transfer protocol with a TTP (KZP protocol), which is depicted in Figure 3. Similar to KP protocol, this protocol is accomplished as below:

- *Step 1-1.* (TTP→Tag): Upon receiving an ownership transfer request, TTP computes $f'_{(N_P \oplus t_i \oplus s_1)}(s_2)$ and sends the tuple $(N_P, f'_{(N_P \oplus t_i \oplus s_1)}(s_2))$ to the tag, where s$_1$ is the secret shared between the tag and the current owner R$_1$ (or the group R$_{11}$, R$_{12}$, …, R$_{1M}$) and t$_i$ is the secret shared between the tag and TTP. This message authenticates TTP to the tag and also updates s$_1$ to s$_2$.
- *Step 1-2.* (Tag→TTP): The tag generates a random number N$_T$ and acknowledges by sending $(N_T, H_{(t_i \oplus N_T)}(s_2 \oplus N_P))$.



r_{1i}, s_1
R_{11} ….. R_{1M}
TTP
t_i, r_{ji}, s_i
Tagi
t_i, s_1, s_2
r_{2i}, s_2
R_{21} ….. R_{2N}
1-1 $N_P, f'_{(N_P \oplus t_i \oplus s_1)}(s_2)$
1-2 $N_T, H_{(t_i \oplus N_T)}(s_2 \oplus N_P)$
2 $f_{r_{1i}}(s_1)$
3-1 $N'_P, f_{(r_{2i} \oplus N'_P)}(s_2 \oplus r_{2i})$
3-2 $H_{r_{2i}}(s_2 \oplus N'_P)$
4-1 $N_{R2i}, f'_{s_2}(N_{R2i})$
4-2 $N'_T, H_{(N'_T \oplus s_2)}(N_{R2i} \oplus s_2)$

Figure 3. The KZP protocol [4].

- *Step 2.* (TTP→(R$_{11}$, R$_{12}$, …., R$_{1M}$)): TTP informs the current owners (R$_{11}$, R$_{12}$, …., R$_{1M}$) that their privileges on Tag$_i$ are being revoked. It sends a revoke message and a keyed encryption function $(f_{(r_{1i})}(s_1))$.
- *Step 3-1.* (TTP→(R$_{21}$, R$_{22}$, …., R$_{2N}$)): TTP generates a random number N'_P and grants new owners (R$_{21}$, R$_{22}$, …., R$_{2N}$) full permissions along with privileges for the tag, by sending $(N'_P, f'_{(r_{2i} \oplus N'_P)}(s_2 \oplus r_{2i}))$.
- *Step 3-2.* ((R$_{21}$, R$_{22}$, …, R$_{2N}$)→TTP): The new owners (R$_{21}$, R$_{22}$, …., R$_{2N}$) send $H_{r_{2i}}(s_2 \oplus N'_P)$ to TTP which is based on the new key value s$_2$.
- *Step 4-1.* ((R$_{21}$, R$_{22}$, …, R$_{2N}$)→Tag): The new owners (R$_{21}$, R$_{22}$, …., R$_{2N}$) then generate a fresh random number N$_{R2i}$, and establish contact with the tag by sending $(N_{R2i}, f'_{(s_2)}(N_{R2i}))$.
- *Step 4-2.* (Tag→(R$_{21}$, R$_{22}$, …., R$_{2N}$)): The tag generates a fresh random number N'_T and replies with $(N'_T, H_{(N'_T \oplus s_2)}(N_{R2i} \oplus s_2))$.

3.1. De-Synchronization Attack

In the KZP protocol, the tag can update its secret key from s$_1$ (old owners) to s$_2$ (new owners) and loses s$_1$ after two different steps of the protocol.

1. After step 1-1, when the tag receives the message from TTP; based on this assumption, the tag is de-synchronized forever and neither TTP nor the owners can access it anymore [10].
2. After step 4-1, when it receives the message from the new owners; based on this assumption, it would be vulnerable to the window attack in which, for a fraction of time, both the current and new owners could access to the tag. On the other hand, an adversary can put a tag and consequently the ownership transfer system in a de-synchronized state. In this section, we describe a de-synchronization attack based on this assumption that

the tag loses s_1 when it receives the message from the new owners (R_{21}, R_{22}, …., R_{2N}). The attack's procedure is described as below:

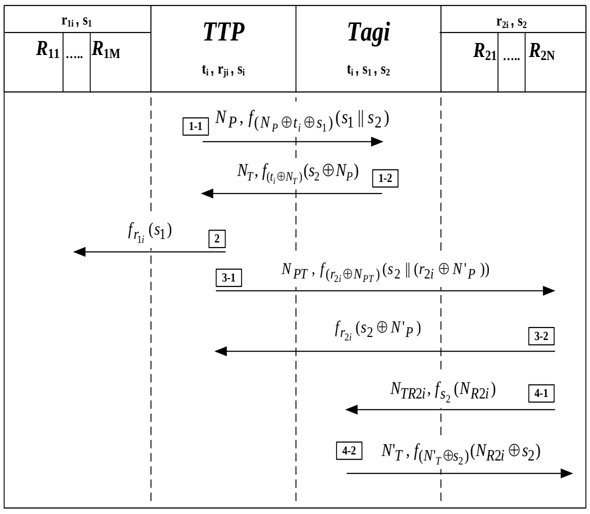
- *Step 1*. The adversary eavesdrops the pair $(N_P, f'_{(N_P \oplus t_i \oplus s_1)}(s_2))$ sent from TTP to the tag and the pair $(N_{R2i}, f'_{(s_2)}(N_{R2i}))$ sent from the new owners to the tag.
- *Step 2*. The adversary blocks the pair $(N'_T, H_{(N'_T \oplus s_2)}(N_{R2i} \oplus s_2))$ sent by the tag, now the tag has a new key s_2, but the new owners are waiting for an acknowledgement.
- *Step 3*. Now, new owners do not accept the tag ownership transfer. Hereon the protocol must be started again, as TTP sends $(N^*_P, f'_{(N^*_P \oplus t_i \oplus s_1)}(s'_2))$ to the tag. Now the tag generates a random number N^*_T and acknowledge by sending $(N^*_T, H_{(t_i \oplus N^*_T)}(s_2' \oplus N^*_P))$ to TTP. Then TTP sends a revoke message $(f_{(r_{1i})}(s_1))$ to the old owners and informs them that their privileges on Tag_i are being revoked. Next, TTP sends $(N'^*_P, f'_{(r_{2i} \oplus N'^*_P)}(s'_2 \oplus r_{2i}))$ as a grant message to the new owners, along with a freshly generated random number N'^*_P.
- *Step 4*. The adversary blocks the acknowledgement sent from the new owners to TTP and uses messages $(N_P, f'_{(N_P \oplus t_i \oplus s_1)}(s_2))$ and $(N_{R2i}, f'_{(s_2)}(N_{R2i}))$ that have eavesdropped before and sends them respectively to the tag. The tag is thus deceived to update its current secret to s_2. Thus, neither TTP records (new (s'_2) and old (s_1)) nor new owner's key (s'_2) are equal to the new tag's key (s_2).

In fact, if TTP wants to take remedial action, it needs the tag's secret (s_2) and TTP does not know it. Hence, the tag is de-synchronized forever and neither TTP nor the owners can access it any more. The success probability of this attack is 1 while the complexity is two runs of protocol.

3.2. Improved KZP Protocol

Similar to the KP protocol, in the KZP protocol, TTP sends the message $(N'_P, f'_{(r_{2i} \oplus N'_P)}(s_2 \oplus r_{2i}))$ with a freshly generated random number N'_P and R_2 (or a group of owners (R_{21}, R_{22}, …, R_{2N}) generates a fresh random number N_{R2i} and establishes contact with the tag by sending $(N_{R2i}, f'_{(s_2)}(N_{R2i}))$. To improve the protocols security, we replace N'_P by N_{PT} (where $N_{PT} = N'_P \oplus N_T$) and N_{R2i} by N_{TR2i} (where $N_{TR2i} = N_T \oplus N_{R2i}$). Figure 4 shows the improved KZP protocol. Further modifications to improve the KZP protocol is described as below:

- *Step 1-1*. (TTP→Tag): When TTP receives the ownership transfer request, it generates a random number N_P and a secret key s_2 and computes $(N_P, f_{(N_P \oplus t_i \oplus s_1)}(s_1 || s_2))$, it then sends $(N_P, f_{(N_P \oplus t_i \oplus s_1)}(s_1 || s_2))$ to the tag, where s_1 is the secret shared between the tag and the current owners (R_{11}, R_{12}, …., R_{1M}) and t_i is the secret shared between the tag and TTP. This message authenticates TTP to the tag and also updates s_1 to s_2.
- *Step 1-2*. (Tag→TTP): The tag generates a random number N_T and sends $(N_T, f_{(t_i \oplus N_T)}(s_2 \oplus N_P))$ to TTP as an acknowledgement.
- *Step 2*. (TTP→(R_{11}, R_{12}, …., R_{1M})): TTP sends $(f_{(r_{1i})}(s_1))$ to the old owners as a revoke message.
- *Step 3-1*. (TTP → (R_{21}, R_{22}, …., R_{2N})): TTP computes ($N_{PT} = N'_P \oplus N_T$) and sends the tuple $(N_{PT}, f_{(r_{2i} \oplus N_{PT})}(s_2 || (r_{2i} \oplus N'_P)))$ to the new owners as a grant message.
- *Step 3-2*. ((R_{21}, R_{22}, …, R_{2N})→TTP): The new owners (R_{21}, R_{22}, …., R_{2N}) send $(f_{r_{2i}}(s_2 \oplus N'_P))$ to TTP using a new key value s_2.
- *Step 4-1*. ((R_{21}, R_{22}, …, R_{2N})→Tag): The new owners use N_{PT} to decrypt the value sent by TTP in Step 3-1 $(f_{(r_{2i} \oplus N_{PT})}(s_2 || (r_{2i} \oplus N'_P)))$ and determine N_T by XORing N_{PT} with N'_P. Then they compute ($N_{TR2i} = N_T \oplus N_{R2i}$) and establish contact with the tag by sending $(N_{TR2i}, f_{(s_2)}(N_{R2i}))$.
- *Step 4-2*. (Tag→(R_{21}, R_{22}, …., R_{2N})): The tag determines N_T by XORing N_{R2i} with N_{TR2i}, then it generates a random number N'_T and sends $(N'_T, H_{(N'_T \oplus s_2)}(N_{R2i} \oplus s_2))$ to the new owners as an acknowledgement.



r_{1i}, s_1
R_{11} ….. R_{1M}
TTP
t_i, r_{ji}, s_i
Tagi
t_i, s_1, s_2
r_{2i}, s_2
R_{21} ….. R_{2N}
1-1 $N_P, f_{(N_P \oplus t_i \oplus s_1)}(s_1 \| s_2)$
$N_T, f_{(t_i \oplus N_T)}(s_2 \oplus N_P)$ 1-2
$f_{r_{1i}}(s_1)$ 2
3-1 $N_{PT}, f_{(r_{2i} \oplus N_{PT})}(s_2 \| (r_{2i} \oplus N'_P))$
$f_{r_{2i}}(s_2 \oplus N'_P)$ 3-2
$N_{TR2i}, f_{s_2}(N_{R2i})$ 4-1
4-2 $N'_T, f_{(N'_T \oplus s_2)}(N_{R2i} \oplus s_2)$

Figure 4. The improved KZP protocol.

Similar to the improved KP protocol, in this subsection we analyze the security of the improved KZP protocol.

- *Secrecy/Data Integrity*: In the improved KZP protocol, using the keyed encryption function (f_k) secures all the communications.
- *DoS/Synchronization Problem*: We use the messages that contain freshly generated random numbers (N_J, N_{IJ}) which means that adversaries are unable to cause de-synchronization problem by blocking any messages.
- *Forward Secrecy*: After the ownership has been transferred, the new owners (R_{21}, R_{22}, …, R_{2N}) are never allowed to know the old keys shared between the tag and its old owners and are unable to decrypt the messages between the tag and its old owners (R_{11}, R_{12}, …., R_{1M}).
- *Passive Replay*: In the improved KZP protocol, each encrypted message contains a random value in each session, so the adversary is unable to pass the protocol by replaying eavesdropped messages.
- *Windowing Problem*: In the improved KZP protocol, upon receiving the message from TTP the tag updates s_1 to s_2, so we cannot find a fraction of time, in which, both the new owners and the old owners can access the tag.

Table 4 compares the serious security features of KZP protocol and improved KZP protocol.

Table 4. Security features comparison between the KZP protocol and the improved KZP protocol.

Protocol Feature	KZP Protocol	Improved KZP Protocol
Secrecy	Yes [4]	Yes
Synchronization	No [4]	Yes
Forward Secrecy	Yes [4]	Yes
Passive Replay	Yes [4]	Yes
Resistance Against Windowing Problem	No [4]	Yes

3.3. Performance Analysis of the Improved KZP Protocol

In this paper, we evaluate the performance of the improved KZP protocol theoretically. Table 5 depicts performance comparison of the KZP protocol and the improved KZP protocol.

Table 5. Performance comparison between the KZP protocol and the improved KZP protocol.

# of Operations	KZP Protocol [2]	Improved KZP Protocol
#⊕	9	9
#H	3	0
# Transferred Bits	L	2L
# Random Number	5	5
#f(.)	4	7

In Table 5, "L", "H" and "*f*(.)" denote the bit length parameters, hash function and encryption function respectively.

4. Conclusions

In this paper, we analyzed the security of two Ownership Transfer (OT) protocols which have been proposed by Kapoor and Piramuthu [3] and Kapoor *et al*. [4]. We proved that an attacker can put a tag in a de-synchronized state. The success probability of the given attacks is 1 while the complexity of all is only two runs of the protocols. Finally, we proposed two OTPs which are more secure than Kapoor and Piramuthu [3] and Kapoor *et al*. [4] OTPs and we proved that the proposed protocols are immune against de-synchronization attacks.

Acknowledgment

This work was supported by Shahid Rajaee Teacher Training University under contract number 21030.

References

[1] Chen C., Lai Y., Chen C., Deng Y., and Hwang Y., "RFID Ownership Transfer Authorization Systems Conforming Epcglobal Class-1 Generation-2 Standards," *International Journal of Network Security*, vol. 13, no. 1, pp. 41-48, 2011.

[2] Fouladgar S. and Afifi H., "A Simple Privacy Protecting Scheme Snabling Delegation and Ownership Transfer for RFID Tags," *Journal of Communications*, vol. 2, no. 6, pp. 6-13, 2007.

[3] Kapoor G. and Piramuthu S., "Single RFID Tag Ownership Transfer Protocols," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, no. 2, pp. 164-173, 2012.

[4] Kapoor G., Zhou W., and Piramuthu S., "Multi-Tag and Multi-Owner RFID Ownership Transfer in Supply Chains," *Decision Support Systems*, vol. 52, no. 1, pp. 258-270, 2011.

[5] Kochar B. and Chhillar R., "An Effective Data Warehousing System for RFID Using Novel Data Cleaning, Data Transformation and Loading Techniques," *The International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 208-216, 2012.

[6] Kulseng L., Yu Z., Wei Y., and Guan Y., "Lightweight Mutual Authentication and Ownership Transfer for RFID Systems," *in Proceedings of IEEE INFOCOM*, San Diego, pp. 251-255, 2010.

[7] Lo N., Ruan S., and Wu T., "Ownership Transfer Protocol for RFID Objects Using lightweight Computing Operators," *in Proceedings of International Conference for Internet Technology and Secured Transactions*, Abu Dhabi, pp. 484-489, 2011.

[8] Lopez P., Hernandez-Castro J., Tapiador J., Li T., and Li Y., "Vulnerability Analysis of RFID Protocols for Tag Ownership Transfer,"

Computer Networks, vol. 54, no. 9, pp. 1502-1508, 2010.

[9] Molnar D., Soppera A., and Wagner D., “A scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags, ” *in Proceedings of the 12th International Conference on Selected Areas in Cryptography*, Kingston, pp. 276-290, 2005.

[10] Munilla J., Peinado A., Yang G., and Susilo W., Enhanced Ownership Transfer Protocol for RFID in an Extended Communication Model, http://eprint.iacr.org/, Last Visited 2014.

[11] Nasir M., Norman A., Fauzi S., and Azmi M., “An RFID-based Validation System for Halal Food,” *The International Arab Journal of Information Technology*, vol. 8, no. 2, pp. 204-211, 2011.

[12] Safkhani M., Bagheri N., Naderi M., and Mahani A., On the Security of Lo *et al*.’s ownership transfer protocol, http://eprint.iacr.org/, Last Visited, 2014.

[13] Saito J., Imamoto K., and Sakurai K., “Reassignment Scheme of an RFID Tag's Key for Owner Transfer,” *in Proceedings of International Conference on Embedded and Ubiquitous Computing*, Taiwan, pp. 1303-1312, 2005.

[14] Song B. and Mitchell C., “Scalable RFID Security Protocols Supporting Tag Ownership Transfer,” *Computer Communications*, vol. 34, no. 4, pp. 556-566, 2011.

[15] Yang M. and Hu H., “Protocol for Ownership Transfer Across Authorities: With the Ability to Assign Transfer Target,” *Security and Communication Networks*, vol. 5, no. 2, pp. 164-177, 2012.

[16] Yang M., “Secure Multiple Group Ownership Transfer Protocol for Mobile RFID,” *Electronic Commerce Research and Applications*, vol. 11, no. 4, pp. 361-373, 2012.

[17] Zuo Y., “Changing Hands Together: a Secure Group Ownership Transfer Protocol for RFID Tags,” *in Proceedings of 43rd Hawaii International Conference on System Sciences*, Honolulu, pp. 1-10, 2010.

Nasour Bagheri is an assistant professor at Electrical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. He is the author of over 70 articles in information security and cryptology.

Farhad Aghili received his MSc of Electrical Engineering from SRTTU, 2013. His research interest includes RFID security.

Masoumeh Safkhani is an assistant professor at Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. She is the author of 30 articles in cryptology. Her current research interest includes RFID security.