Computer Communications 000 (2016) 1–5

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Attacks on ownership transfer scheme for multi-tag multi-owner passive RFID environments

J. Munilla[a,*], M. Burmester[b], A. Peinado[a]

[a] *E.T.S.I. Telecomunicación, Universidad de Málaga, Málaga, Spain*
[b] *Department of Computer Science, Florida State University, Tallahassee, Florida, USA*

ARTICLE INFO

Article history:
Received 20 November 2014
Revised 24 June 2015
Accepted 14 May 2016
Available online xxx

Keywords:
RFID
EPCC1G2
Ownership transfer
Cryptanalysis

ABSTRACT

Sundaresan et al. proposed recently a novel ownership transfer protocol for multi-tag multi-owner RFID environments that complies with the EPC Class1 Generation2 standard. The authors claim that this provides individual-owner privacy and prevents tracking attacks. We show that this protocol falls short of its security objectives, and describe attacks that allow: (*a*) an eavesdropper to trace a tag, (*b*) the previous owner to obtain the private information that the tag shares with the new owner, and (*c*) an adversary that has access to the data stored on a tag to link this tag to previous interrogations (violating forward-secrecy). We analyze the security proof and show that while the first two cases can be addressed with a more careful design, strong privacy remains an open problem for lightweight RFID applications.

1. Introduction

The term “Internet of Things” (IoT) was coined in 1999 by Kevin Ashton, a cofounder of the Auto-ID [1] center that promoted the development of tracking products for supply-chain management by using low-cost RFID tags. RFID tags and sensors enable computers to observe, identify and understand situational awareness without requiring human intervention. Initial designs focused on performance with less attention paid to resilience and security. However this technology is currently used in many applications that need to be protected. Protection must take into account the special features of RFID, such as the vulnerabilities of the radio channel, power-constraints, low-cost, limited functionality, reply upon request, as well as resistance to the risks of RFID, such lack of privacy, malicious traceability and data corruption. The increasing concern with security is evidenced by the inclusion of some optional cryptographic features in the recently ratified second version of the EPCglobal Gen2 specifications [2].

Ownership Transfer Protocols (OTPs) allow the secure transfer of tag ownership from a current owner to a new owner. They support distributed RFID applications and are a basic component of the IoT. Three entities are present in an OTP: the tag $\mathcal{T}$, whose rights are being transferred, the current owner, who has the initial control of $\mathcal{T}$, and the new owner, who will get control of $\mathcal{T}$ when the protocol is completed. OTPs must incorporate security requirements that protect the privacy of both the new and the previous owner of the tag. To prevent previous owners from accessing a tag once ownership has been transferred either a Trusted Third Party (TTP) is employed or an Isolated Environment (IsE). The first provides security for stronger adversarial scenarios while the second is more appropriate when tags belong to independent authorities.

For RFID applications privacy addresses *anonymity* that protects the identity of tags, and *untraceability* that protects past interrogations (partial or completed) of a tag being linked. Formal definitions for secure ownership and ownership transfer are provided by van Deursen et al. [3] while several theoretical models have been proposed in the literature to address the privacy of RFID systems [4–7]. The theoretical framework of Vaudenay [7] distinguishes between strong and weak attackers. Privacy preserving protocols against strong adversaries support *forward secrecy* [8].

Molnar et al. [9] and Saito et al. [10] presented the first OTP for RFID applications in 2005. This was followed by several OTPs that address practical scenarios. Recently Sundaresan et al. [11] proposed an OTP for multi-tag multi-owner RFID environment that provides individual-owner-privacy. The protocol uses a TTP for secure management and an IsE for verifying ownership transfer. This complies with the EPCglobal Gen2 specifications, with protection afforded by simple XOR and 128-bit PRNGs. The protocol is claimed to provide tag anonymity, tag location privacy, forward secrecy, and forward untraceability; while being resistant to replay, desynchronization, server impersonation and active attacks. We shall show in this paper that this protocol falls short of these claims. In particular that it is subject to:

* Corresponding author.
E-mail address: munilla@ic.uma.es (J. Munilla).

http://dx.doi.org/10.1016/j.comcom.2016.05.007
0140-3664/© 2016 Elsevier B.V. All rights reserved.

Trusted Third Party $(ST_s, T_{id_j}, O_{id_i}, N_{s_i})$	**Each Tag j in Tag-Group** $(ST_s, ST'_s, T_{id_j}, O_{id_i}, OT_{s_i}, OT'_{s_i})$
Step 2A Generate $S1_r$ Generate a new Tag-Group secret ST_{sn} For each New Owner i $M5_i = N_{s_i} \oplus PRNG(ST_s \oplus S1_r)$ $M6_i = O_{id_i} \oplus PRNG(N_{s_i} \oplus ST_s \oplus S1_r)$ $M_i^c = PRNG(M5_i \oplus S1_r \oplus ST_{sn}) \oplus PRNG(M6_i \oplus ST_s)$ Next Owner For each Tag j in Tag-Group $M7_j = T_{id_j} \oplus PRNG(T_{id_j} \oplus ST_s \oplus S1_r)$ $M8_j = S1_r \oplus PRNG(T_{id_j} \oplus ST_s)$ $M9_j = ST_{sn} \oplus PRNG(M7_j \oplus T_{id_j} \oplus ST_{sn})$ $\xrightarrow{M5_{(1..i)}, M6_{(1..i)}, M7_j, M8_j, M9_j, M^c_{(1..i)}}$ Next Tag **Step 2C** For each Tag's reply $T1_r \leftarrow RND_t \oplus T_{id_j} \oplus ST_s$ If $T_{id_j} \oplus (O_{id} \Vert N_i)_{(1..i)} = ACK_t \oplus PRNG(ST_s \oplus T1_r)$ Tag Authenticated New Owners & Secrets Successfully Inserted If ALL_ACK NOT Received then Goto Step 2A else $ST_s \leftarrow ST_{sn}$	**Step 2B** $S1_r \leftarrow M8_j \oplus PRNG(T_{id_j} \oplus ST_s)$ if $T_{id_j} = M7_j \oplus PRNG(T_{id_j} \oplus ST_s \oplus S1_r)$ TTP Authenticated & Message is for this tag else Use ST'_s in the above steps and try again; if unsuccessful, abort. $ST_{sn} \leftarrow M9_j \oplus PRNG(M7_j \oplus T_{id_j} \oplus ST_s)$ For all i if $PRNG(M5_i \oplus S1_r \oplus ST_{sn}) \oplus PRNG(M6_i \oplus ST_s) = M_i^c$ $N_{s_i} \leftarrow M5_i \oplus PRNG(ST_s \oplus S1_r)$ $O_{id_i} \leftarrow M6_i \oplus PRNG(N_{s_i} \oplus ST_s \oplus S1_r)$ else abort Remove Previous Owner's IDs & Secure Secrets from Tag Insert $O_{id_{(1..i)}}, OT_s \leftarrow N_{s_{(1..i)}}, OT'_s \leftarrow N_{s_{(1..i)}}$ Generate $T1_r$ $RND_t = T1_r \oplus T_{id_j} \oplus ST_s$ $ACK_t = T_{id_j} \oplus (O_{id} \Vert OT_s)_{(1..i)} \oplus PRNG(ST_s \oplus T1_r)$ $\xleftarrow{RND_t,\ ACK_t}$ If T_{id_j} matches using ST_s $ST'_s \leftarrow ST_s$ $ST_s \leftarrow ST_{sn}$

Fig. 1. Step 2 of the Sundaresan *et al.* protocol.

(a) De-synchronization and/or replay attacks (Theorem 1);
(b) Traceability (tag location privacy) attacks: an eavesdropper can trace a tag (Theorems 2, 3);
(c) Impersonation attacks: a previous owner can compute the secret data that tags share with the new owner (Theorems 4, 5);
(d) Forward secrecy attacks: compromised tags can be linked to earlier interrogations (Theorem 6);
(e) De-synchronization attacks: if the shared secrets are generated using a random non-deterministic process (Theorem 7).

The rest of the paper is organized as follows. Section 2 discusses the Sundaresan et al. protocol and describes the phase that is cryptanalyzed. Section 3 describes the security flaws listed above. Section 4 analyzes the cryptographic causes of these weaknesses and Section 5 concludes the paper.

2. The Sundaresan et al. ownership transfer protocol

This is a TTP-based scheme developed for multi-tag multi-owner RFID environments [11]. Two kinds of associations are considered: tags with multiple owners and owners with multiple tags. Every tag and owner is setup with identifiers and several shared and private secrets in the initialization phase. The protocol begins when a group of owners sends an ownership transfer (OT) request to the TTP. The protocol has two steps: Step 1 involves the TTP and new owners while Step 2 involves the TTP and the tags in Tag-Group, and is intended to transfer the identifiers of the new owners and the secret keys to the tags. In this paper we are only concerned with Step 2, since our analysis will focus on its weaknesses. This step is shown in Fig. 1, and is described below. For convenience we use the abbreviations $(O_{id}||OT_s)_{(1..i)}$ for $(O_{id_1}||OT_{s_1}) \oplus (O_{id_2}||OT_{s_2}) \oplus \cdots \oplus (O_{id_i}||OT_{s_i})$.

2.1. Step 2 of the Sundaresan et al. OTP: TTP → Tag-Group → TTP

TTP uses the values: $\{ST_s, T_{id_j}, (O_{id}, N_s)_{(1..i)}\}$, with ST_s a secret shared with Tag-Group, T_{id_j} an identifier for tag j in Tag-Group, O_{id_i} an identifier for each new owner i of tag j, and N_{s_i} a new secret for this owner.

Each tag j in Tag-Group uses the values: $\{ST_s, ST'_s, T_{id_j}, (O_{id}, OT_s, OT'_s)_{(1..i)}\}$, with ST'_s the value of ST_s used in the previous interaction (initially $ST_s = ST'_s$), and O_{id_i}, OT_{s_i} and OT'_{s_i} the identifier of its (previous) owner i, and the current and the previous secret shared with this owner. The execution of the protocol will result in the updating of these later values.

Step 2A TTP generates a pseudorandom number $S1_r$ and a new secret ST_{sn} to be shared with the tags in Tag-Group. Then TTP computes:
for each new owner i,
$M5_i = N_{s_i} \oplus PRNG(ST_s \oplus S1_r)$, $M6_i = O_{id_i} \oplus PRNG(N_{s_i} \oplus ST_s \oplus S1_r)$ and $M_i^c = PRNG(M5_i \oplus S1_r \oplus ST_{sn}) \oplus PRNG(M6_i \oplus ST_s)$,
and for each tag j,
$M7_j = T_{id_j} \oplus PRNG(T_{id_j} \oplus ST_s \oplus S1_r)$, $M8_j = S1_r \oplus PRNG(T_{id_j} \oplus ST_s)$, and $M9_j = ST_{sn} \oplus PRNG(M7_j \oplus T_{id_j} \oplus ST_s)$.
Then, TTP sends $M_{TG} = (M5_{(1..i)}, M6_{(1..i)}, M7_j, M8_j, M9_j, M^c_{(1..i)})$ to each tag j in Tag-Group:
TTP → Tag-Group: M_{TG}.

Step 2B Each tag j in Tag-Group checks if for its T_{id_j}: $T_{id_j} \stackrel{?}{=} M7_j \oplus PRNG(T_{id_j} \oplus ST_s \oplus S1_r)$, where $S1_r = M8_j \oplus PRNG(T_{id_j} \oplus ST_s)$. If this fails it uses ST'_s instead of ST_s. If both fail, it aborts. Otherwise *TTP is authenticated* and the tag knows that the message is for itself. For the remainder of the protocol, either ST_s or ST'_s is used, depending on which one returned a match.
Tag j checks if for all i: $M_i^c \stackrel{?}{=} PRNG(M5_i \oplus S1_r \oplus ST_{sn}) \oplus PRNG(M6_i \oplus ST_s)$, where $ST_{sn} = M9_j \oplus PRNG(M7_j \oplus T_{id_j} \oplus ST_s)$. If this fails for some i, it aborts. Otherwise it computes $O_{id_i} = M6_i \oplus PRNG(N_{s_i} \oplus ST_s \oplus S1_r)$ and $N_{s_i} = M5_i \oplus PRNG(ST_s \oplus S1_r)$, and replaces the previ-

Please cite this article as: J. Munilla et al., Attacks on ownership transfer scheme for multi-tag multi-owner passive RFID environments, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.05.007

ous owner identifiers with O_{id_i}, and the secrets with $OT_s = OT'_s = N_{s_i}$, for every owner. Then, it replies:
Each tag in Tag-Group → TTP : (RND_t, AKC_t)
where $RND_t = T1_r \oplus T_{id_j} \oplus ST_s$, $ACK_t = T_{id_j} \oplus (O_{id}||N_s)_{(1..i)} \oplus PRNG(ST_s \oplus T1_r)$, with $T1_r$ a fresh pseudorandom number. If there was a match with ST_s the tag updates the shared secrets: $ST'_s \leftarrow ST_s$ and $ST_s \leftarrow ST_{sn}$.

Step 2C For each tag reply, TTP checks if $ACK_t \oplus PRNG(ST_s \oplus T1_r) \stackrel{?}{=} T_{id_j} \oplus (O_{id}||N_s)_{(1..i)}$, where, where $T1_r = RND_t \oplus T_{id_j} \oplus ST_s$. If there is a match, the tag is authenticated and the new owners and their secrets have been successfully inserted in that tag. If TTP does not receive acknowledgements from all tags in Tag-Group, the protocol is restarted from the beginning of Step 2. Otherwise TTP updates $ST_s \leftarrow ST_{sn}$ and sends a message to confirm the transfer to the previous owner. The new owner can verify that the transfer has been successful with the Ownership Test Protocol (described in Appendix A).

2.2. Security claims of the Sundaresan et al. protocol

It is claimed in [11] that this protocol provides:

– Tag/Reader Anonymity: it protects against information leakage that can lead to disclosure of a tag's/reader's real identifier.
– Tag/Reader Location Privacy: messages cannot be used to track the tag's/reader's location(s).
– Forward Secrecy: the protocol ensures that on compromise of the internal secrets of the tag, its previous communications cannot be traced by the attacker.
– Forward Untraceability: the protocol ensures that the previous owner is unable to trace or communicate with the tag post-ownership transfer.
– Replay Attacks: the protocol resists compromise by an attacker through the replay of messages that have been collected by an attacker during previous protocol sequences.
– Denial of Service: an attacker cannot lead to desynchronization between the parties.
– Tag/Reader/Impersonation: the protocol ensures that legitimate parties cannot be impersonated by an attacker to another legitimate party.
– Active Attacks: the protocol is resistant to attacks where an adversary has the ability to modify messages during the communication.

3. Weaknesses of the Sundaresan et al. protocol

To simplify our notation below, and when there is no ambiguity, we replace $T_{id_j}, M7_j, M8_j, M9_j$ by T_{id}, $M7$, $M8$, $M9$ respectively.

3.1. Replay attacks (or desynchronization)

Theorem 1. *The Sundaresan et al. protocol is subject to desynchronization or replay attacks.*

Proof. The proof is by contradiction. Suppose the protocol resists desynchronization attacks. Then: □

Lemma 1. *Only an authorized TTP can cause a tag $\mathcal{T}$ to update its current and previous secrets ST_s, ST'_s (shared by TTP and Tag-Group).*

Proof. Otherwise $\mathcal{T}$ and the TTP can get desynchronized. □

Lemma 2. *If $M7$, $M8$ (Section 2.1) are accepted by tag $\mathcal{T}$ as valid at time t_a, then the same messages will be accepted by $\mathcal{T}$ at time t_b, provided there is no interaction between $\mathcal{T}$ and TTP during the interval $[t_a, t_b]$.*

Proof. Let $ST^1 = ST_s$, $ST^0 = ST'_s$ be the current and previous secrets shared by TTP and the tags in $\mathcal{T}$'s Tag-Group. If $M7$, $M8$ are accepted at time t_a by $\mathcal{T}$, with identifer T_{id}, then either:

$$T_{id} = M7 \oplus PRNG(T_{id} \oplus ST^0 \oplus M8 \oplus PRNG(T_{id} \oplus ST^0), \text{ or}$$
$$T_{id} = M7 \oplus PRNG(T_{id} \oplus ST^1 \oplus M8 \oplus PRNG(T_{id} \oplus ST^1).$$

In the first case (when ST^0 is used), $\mathcal{T}$ will not update its secrets and by Lemma 1 the same values ST^1, ST^0 will be stored on $\mathcal{T}$ until time t_b. Then $M7$, $M8$ will be accepted by $\mathcal{T}$ at time t_b. In the second case (when ST^1 is used), two situations are possible. If $\mathcal{T}$ aborts (other messages are not accepted), then it will not update its secrets, and as in the first case, $M7$, $M8$ will be accepted by $\mathcal{T}$ at time t_b. Otherwise, if $\mathcal{T}$ does not abort, it will update its secrets: $ST_s \leftarrow ST^2$, $ST'_s \leftarrow ST^1$, where ST^2 is the next value of the secret (i.e. ST_{sn}). As there is no interaction with TTP, these values will not be updated (Lemma 1) during $[t_a, t_b]$. Then $M7$, $M8$ will be accepted at time t_b if either:

$$T_{id} = M7 \oplus PRNG(T_{id} \oplus ST^1 \oplus M8 \oplus PRNG(T_{id} \oplus ST^1)), \text{ or}$$
$$T_{id} = M7 \oplus PRNG(T_{id} \oplus ST^2 \oplus M8 \oplus PRNG(T_{id} \oplus ST^2)).$$

Since we are assuming that ST^1 is used, the first equation holds. Therefore $M7$, $M8$ are accepted. □

Lemma 3. *If tag $\mathcal{T}$ accepts the messages M_{TG} (Section 2.1) as valid at time t_a, then $\mathcal{T}$ will accept a replay of M_{TG} at time t_b, provided there is no interaction between $\mathcal{T}$ and TTP during $[t_a, t_b]$.*

Proof. By Lemma 2, $M7$, $M8$ will be accepted at time t_b. According to the protocol, the other messages will be accepted if $M^c_{(1..i)}$ is accepted. We shall show that $M^c_{(1..i)}$ is accepted at time t_b and that consequently all the other messages of $M_{TG} = (M5_{(1..i)}, M6_{(1..i)}, M7, M8, M9, M^c_{(1..i)})$ will be accepted. The proof is similar to Lemma 2. Since $M^c_{(1..i)}$ is accepted at time t_a, either:

$$M^c_{(1..i)} = PRNG(M5_{(1..i)} \oplus S1_r \oplus ST_{sn}) \oplus PRNG(M6_{(1..i)} \oplus ST^0), \text{ with } S1_r = M8 \oplus PRNG(T_{id} \oplus ST^0)), ST_{sn} = M9 \oplus PRNG(M7 \oplus T_{id} \oplus ST^0), \text{ or}$$
$$M^c_{(1..i)} = PRNG(M5_{(1..i)} \oplus S1_r \oplus ST_{sn}) \oplus PRNG(M6_{(1..i)} \oplus ST^1), \text{ with } S1_r = M8 \oplus PRNG(T_{id} \oplus ST^1)), ST_{sn} = M9 \oplus PRNG(M7 \oplus T_{id} \oplus ST^1).$$

In the first case $\mathcal{T}$ will not update the secrets and by Lemma 1 these values will remain stored on $\mathcal{T}$. Then $M^c_{(1..i)}$ will be accepted at time t_b. In the second case (ST^1 is used) $\mathcal{T}$ updates its secrets: $ST_s \leftarrow ST^2$, $ST_s{}' \leftarrow ST^1$, where ST^2 is the next value of the secret. Since there is no interaction with TTP, these values will not be updated during $[t_a, t_b]$, so at time t_b, $M^c_{(1..i)}$ will be accepted if either:

$$M^c_{(1..i)} = PRNG(M5_{(1..i)} \oplus S1_r \oplus ST_{sn}) \oplus PRNG(M6_{(1..i)} \oplus ST^1), \text{ with } S1_r = M8 \oplus PRNG(T_{id} \oplus ST^1)), ST_{sn} = M9 \oplus PRNG(M7 \oplus T_{id} \oplus ST^1), \text{ or}$$
$$M^c_{(1..i)} = PRNG(M5_{(1..i)} \oplus S1_r \oplus ST_{sn}) \oplus PRNG(M6_{(1..i)} \oplus ST^2), \text{ with } S1_r = M8 \oplus PRNG(T_{id} \oplus ST^2)), ST_{sn} = M9 \oplus PRNG(M7 \oplus T_{id} \oplus ST^2).$$

The first case holds since we are assuming that ST^1 is used. Therefore $M^c_{(1..i)}$ and the other messages are accepted during $[t_a, t_b]$.

We conclude the proof of Theorem 1 by observing that if an adversary $\mathcal{A}$ eavesdrops on a protocol execution between $\mathcal{T}$ and TTP with messages M_{TG} and then later replays M_{TG} to $\mathcal{T}$, $\mathcal{T}$ will accept these as valid by Lemma 3. □

Please cite this article as: J. Munilla et al., Attacks on ownership transfer scheme for multi-tag multi-owner passive RFID environments, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.05.007

3.2. Traceability

Theorem 2. *An adversary $\mathcal{A}$ that eavesdrops on the last successful execution of the Sundaresan et al. protocol between TTP and a group of tags G can later determine if a tag $\mathcal{T}$ belongs to G.*

Proof. Let $\mathcal{A}$ be an eavesdropping adversary. Then: □

Lemma 4. *$\mathcal{A}$ can determine if the messages sent to $\mathcal{T}$ are accepted by $\mathcal{T}$.*

Proof. Tags only respond with RND_t, ACK_t after checking that the received messages M_{TG} are correct; otherwise they abort. $\mathcal{A}$ only needs to check that there is a response to determine if messages are accepted. □

Lemma 5. *$\mathcal{A}$ can determine if an execution of the Sundaresan et al. protocol is successful.*

Proof. By Lemma 4, $\mathcal{A}$ knows that the messages M_{TG} are accepted by a tag $\mathcal{T}$ when $\mathcal{T}$ replies. Consequently $\mathcal{A}$ can determine when the protocol is successfully executed by checking that (in response to the queries sent by TTP) each tag in Tag-Group replies. □

To conclude the proof suppose that $\mathcal{A}$ eavesdrops on a successful execution (Lemmas 5) of the protocol between TTP and G to get the messages M_{TG} for each tag j in G. Later, $\mathcal{A}$ replays these to $\mathcal{T}$ for each j. $\mathcal{T}$ belongs to G if any of these is accepted (Lemma 4).

Theorem 3. *An adversary $\mathcal{A}$ that eavesdrops on a successful execution of the Sundaresan et al. protocol between TTP and a group of tags G can trace any tag $\mathcal{T}$ that belongs to G. Traceability extends until $\mathcal{T}$ is transferred to a new owner.*

Proof. Let $\mathcal{A}$ be an eavesdropping adversary. Then: □

Lemma 6. *If $\mathcal{A}$ knows the last set of messages M_{TG} that $\mathcal{T}$ accepted then $\mathcal{A}$ can trace $\mathcal{T}$, even if the protocol was not successful* (e.g. *if this or other responses were not received by TTP).*

Proof. If $\mathcal{T}$ accepted M_{TG} in the last interaction then by Lemma 3 it must again accept M_{TG}. So $\mathcal{A}$ only needs to replay M_{TG} to a tag and check if it is accepted to determine if it is $\mathcal{T}$ (Lemma 4). □

To trace $\mathcal{T}$, $\mathcal{A}$ first determines if it belongs to group G (Theorem 2). If so, $\mathcal{A}$ stores the specific messages M_{TG} that cause $\mathcal{T}$ to reply. Later $\mathcal{A}$ replays M_{TG} to determine if the tag is $\mathcal{T}$ (Lemma 6). Traceability is possible while the values ST_s and ST'_s are not updated (until a new successful OTP is executed).

3.3. Previous owner attack

Theorem 4. *The previous owner of tag $\mathcal{T}$ that eavesdrops on the ownership transfer of $\mathcal{T}$ to a single owner in the Sundaresan et al. protocol can compute the identity of this owner and the secret that it shares with $\mathcal{T}$.*

Proof. We first show that: □

Lemma 7. *$(O_{id}||OT_s)_{(1..i)}$ can be computed from the identification number T_{id} of $\mathcal{T}$ used to generate the pair of known messages RND_t, ACK_t.*

Proof. $(O_{id}||OT_s)_{(1..i)} = ACK_t \oplus T_{id} \oplus PRNG(RND_t \oplus T_{id})$.

The previous owner can apply Lemmma 7 to compute $(O_{id}||OT_s)$ by eavesdropping on the replies RND_t, ACK_t of $\mathcal{T}$ with identifier T_{id}. □

Theorem 5. *The Sundaresan et al. protocol does not guarantee privacy for a new owner: the previous owner can still have access to transferred tags.*

Proof. The previous owner by Theorem 4 can compute $(O_{id}||OT_s)$. Since this is all the secret information a tag shares with its owner, the previous owner will be able to do whatever the new owner can do, including the test protocol (described in Appendix A). □

3.4. Forward secrecy

Theorem 6. *The Sundaresan et al. protocol does not guarantee forward secrecy.*

Proof. Let $\mathcal{A}$ be an eavesdropping adversary. Then: □

Lemma 8. *Given the identifier T_{id} of a tag $\mathcal{T}$ and the value $(O_{id}||OT_s)_{(1..i)}$, it is possible to determine with overwhelming probability if the messages RND_t, ACK_t were computed by $\mathcal{T}$ using $(O_{id}||OT_s)_{(1..i)}$.*

Proof. By Lemma 7 the value $(O_{id}||OT_s)^c_{(1..i)}$ corresponding to T_{id}, RND_t, ACK_t can be computed. If this matches $(O_{id}||OT_s)_{(1..i)}$ then the probability that it was generated by $\mathcal{T}$ (with identifier T_{id}) using $(O_{id}||OT_s)_{(1..i)}$ is overwhelming $(1-\varepsilon)$. Indeed, the probability that RND_t, ACK_t is generated by another tag with identifier T'_{id} using $(O_{id}||OT_s)'_{(1..i)}$ is negligible, since $PRNG(T'_{id} \oplus RND_t) = ACK_t \oplus T'_{id} \oplus (O_{id}||OT_s)'_{(1..i)}$ happens with negligible probability (ε). □

We complete the proof. At time t_a, $\mathcal{A}$ eavesdrops on a successful execution of the protocol between TTP and a tag $\mathcal{T}$ that responds with RND_t, ACK_t. Suppose at time $t_b > t_a$, $\mathcal{A}$ is given access to the secret information stored on $\mathcal{T}$: T_{id}, $(O_{id}||OT_s)_{(1..i)}$, ST_s and ST'_s. Then $\mathcal{A}$ uses Lemma 8 to determine whether the earlier response RND_t, ACK_t was computed by $\mathcal{T}$.

3.5. De-synchronization attack in case of non-deterministic secrets

The description of the protocol just says that new secret values SO_{sn}, N_{s_i} (in Step 1A) and ST_{sn} (in Step 2A) are generated, but it does not provide any detail about this generation process. Thus, these values could be deterministic or non-deterministic. If they are deterministic, there exists only one possible value ST_{sn} that follows a specific ST_s. By contrast, if they are non-deterministic, different values for ST_{sn} are possible; this happens, for example, if they are computed using randomness of that specific session.

Although it cannot be considered a weakness, as we do not know if it is the case, we prove next that if these values were non-deterministic (i.e. they depend on the specific session), then the protocol would be subject to a desynchronization attack.

Theorem 7. *The Sundaresan et al. protocol can be desynchronised by an adversary $\mathcal{A}$ if the values of new secrets are non-deterministic.*

Proof. This combines a man-in-the-middle with an impersonation/replay attack. First $\mathcal{A}$ impersonates a tag $\mathcal{T}$ to get $M^1_{TG} = (M5_{(1..i)}, M6_{(1..i)}, M7_j, M8_j, M9_j, M^c_{(1..i)})$ from TTP, computed using $ST^1_{sn}, ST_s, T_{id}, (O_{id}, N_s)_{(1..i)}$ and $S1_r$ (Section 2.1). Then $\mathcal{A}$ replays M^1_{TG} to $\mathcal{T}$ to get $R^1 = (RND_t, ACK_t)$, computed using ST_s, T_{id}, $(O_{id}||OT_s)_{(1..i)}$ and $T1_r$. $\mathcal{T}$ updates: $ST'_s \leftarrow ST_s$, $ST_s \leftarrow ST^1_{sn}$.

$\mathcal{A}$ impersonates $\mathcal{T}$ again to get M^2_{TG} from TTP, computed using ST^2_{sn}, ST_s, T_{id}, $(O_{id}, N_s)_{(1..i)}$ and $S1^2_r$, with value $ST^2_{sn} \neq ST^1_{sn}$ (as we are assuming they are non-deterministic). $\mathcal{A}$ responds with R^1, which will be accepted by TTP as the computation of RND_t and ACK_t does not depend on the fresh session values $\{S1^2_r, ST^2_{sn}\}$. TTP then updates $ST_s \leftarrow ST^2_{sn}$. Now $\mathcal{T}$ (that stores $ST^1_{sn} \neq ST^2_{sn}$) and TTP are desynchronized. □

4. Cryptanalysis

In this section we analyze the causes for the weaknesses of the Sundaresan et al. protocol discussed in the previous section. Observe that the replay and traceability attacks described in Sections 3.1, 3.2 exploit the fact that the messages $M7_j$, $M8_j$ do not authenticate TTP (in contrast to the authors's claims). To explain this, we revisit the verification proof presented in [11]. This involves the messages $M5_i, M6_i, M_i^c$ for owner i and $M7_j$, $M8_j$, $M9_j$ for tag $\mathcal{T}$ (Section 2.1). The proof uses GNY logic formalization [12]:

V9 Apply the *being-told rule* T1: $\mathcal{T} \triangleleft M5_i, M6_i, M_i^c, M7_j, M8_j, M9_j$.
V10 Apply the *possession rule* P1 to V9: $\mathcal{T} \ni M5_i, M6_i, M_i^c, M7_j, M8_j, M9_j$.
V11 Apply the *freshness rule* F1 to V9: $\mathcal{T} \mid\equiv \sharp M5_i, \sharp M6_i, \sharp M_i^c, \sharp M7_j, \sharp M8_j, \sharp M9_j$.
V12 Use V11, the initial assumptions TTP $\ni S1_r$ (TTP possesses $S1_r$) and TPP $\mid\equiv$ TTP $\leftrightarrow ST_s \mathcal{T}$ (TTP believes that ST_s is a suitable secret to be shared with $\mathcal{T}$), and the postulates I1, J1 and P2 to derive: $\mathcal{T} \mid\equiv$ TTP $\mid\sim M5_i, M6_i, M_i^c, M7_j, M8_j, M9_j$ ($\mathcal{T}$ believes that the messages were actually sent by TTP).

The verification Step V11, and consequently Step V12 (derived from it), is incorrect because the *freshness rule*,

$$\text{F1}: \quad \frac{\mathcal{T} \mid\equiv \sharp X}{\mathcal{T} \mid\equiv \sharp(X, Y),\ \mathcal{T} \mid\equiv \sharp f(X)}$$

requires that $\mathcal{T}$ not only possesses the messages but also that the messages have been computed using a value that $\mathcal{T}$ believes to be fresh. This is not the case in the Sundaresan et al. protocol, and therefore the security proof is flawed. From this analysis we see that for TTP to be authenticated, the messages used to authenticate TTP *must include a value that* $\mathcal{T}$ *believes to be fresh*. This is commonly implemented by including a random number generated by $\mathcal{T}$ for each session. although the tags in the Sundaresan *et al.* protocol generate a fresh number $T1_r$, this is *not* included in the messages $M7$, $M8$ that are used to authenticate TTP, and therefore we get replay and traceability attacks.

The previous owner attack described in Section 3.3 is due to a flawed implementation of the Blum-Micali encryption scheme [13], that simulates a one-time-pad to obfuscate data (with simple XOR). This implementation should be replaced by a more careful design where the input data of the PRNG cannot be recovered by the previous owner using known data (such as the tag identifier T_{id} and exchanged messages) and XOR. This is commonly implemented with one-way (*hash*) functions.

The last attack in Section 3.4 is much harder to address. While the previous attacks can be prevented with more careful designs, a solution for forward privacy is particularly challenging. Indeed achieving forward privacy using only symmetric cryptography is still an open problem. Recently it has been shown that hash-based systems cannot achieve forward privacy in the Byzantine threat model [14], and that there is a trade-off between privacy and availability. Some authors claim that one must use public-key cryptography for forward privacy [7].

5. Conclusions

The Sundaresan et al. ownership transfer protocol falls short of its security goals despite the fact it uses a Trusted Third Party to control/manage private information/keys. This protocol is subject to desynchronization and/or replay atatcks and impersonation, traceability and forward secrecy attacks.

We analysed these weaknesses and discussed possible fixes. We noted that forward privacy may not be achievable using only symmetric cryptography.

Acknowledgments

This work has been supported by the MICINN under project “TUERI: Technologies for secure and efficient wireless networks within the Internet of Things with applications to transport and logistics”, TIN2011-25452.

Appendix A. Ownership test protocol

It must carried out in a virtual environment without any adversarial interference, and therefore messages do not need to be encrypted. For each new owner i and for each Tag j in Tag-Group, the protocol sends O_{id_i}, T_{id_j} to the Tag-Group. Each tag in the Tag-Group checks if $T_{id} = T_{id_j}$ and if so, computes and sends back $M_{tst} = O_{id_i} \oplus OT_s \oplus T_{id}$ using OT_s for that O_{id_i}. For each Tag_Reply received, and for each tag in the Tag-Group, each new owner checks if $(O_{id_i} \oplus OT_{s_i}) = M_{tst} \oplus T_{id_j}$. If all tags are not identified by all owners within a stipulated time, the ownership test protocol is restarted.

References

[1] K. Ashton, That ’internet of things’thing, RFID J. 22 (2009) 97–114. http://www.rfidjournal.com/articles/view?4986.
[2] E. Global, UHF air interface protocol standard generation2/version2, http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2.
[3] T. van Deursen, S. Mauw, S. Radomirovic, P. Vullers, Secure Ownership and Ownership Transfer in RFID systems, in: Proceedings of the Computer Security – ESORICS 2009: 14th European Symposium on Research in Computer Security, Saint-Malo, France, September 21-23, 2009, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 637–654.
[4] G. Avoine, Adversarial model for radio frequency identification, IACR Cryptology ePrint Arch. 2005 (2005) 49.
[5] A. Juels, S.A. Weis, Defining strong privacy for RFID, IACR Cryptology ePrint Arch. 2006 (2006) 137.
[6] C.Y. Ng, W. Susilo, Y. Mu, R. Safavi-Naini, S. Jajodia, J. López, Rfid privacy models revisited., in: Proceedings of the Computer Security - ESORICS 2008: 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 251–266.
[7] S. Vaudenay, K. Kurosawa, On Privacy Models for RFID, in: Proceedings of the Advances in Cryptology – ASIACRYPT 2007: 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 68–87.
[8] M. Burmester, J. Munilla, Lightweight RFID authentication with forward and backward security, ACM Trans. Inf. Syst. Secur. 14 (1) (2011).
[9] D. Molnar, A. Soppera, D. Wagner, A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags, in: Workshop on RFID Security and Light-Weight Crypto, Graz, Austria, 2005, pp. 276–290.
[10] J. Saito, K. Imamoto, K. Sakurai, T. Enokido, L. Yan, B. Xiao, D. Kim, Y.-S. Dai, L.T. Yang, Reassignment scheme of an RFID tag’s key for owner transfer, in: Proceedings of the Embedded and Ubiquitous Computing – EUC 2005 Workshops: EUC 2005 Workshops: UISW, NCUS, SecUbiq, USN, and TAUES, Nagasaki, Japan, December 6-9, 2005, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 1303–1312.
[11] S. Sundaresan, R. Doss, W. Zhou, S. Piramuthu, Secure ownership transfer for multi-tag multi-owner passive RFID environment with individual-owner privacy, Comput. Commun. In press 55 (2015) 112–124.
[12] L. Gong, R. Needham, R. Yahalom, Reasoning about belief in cryptographic protocols, in: IEEE Computer Society Symposium on Research in Security and Privacy, 1990, pp. 234–248.
[13] M. Blum, S. Micali, How to generate cryptographically strong sequences of pseudo-random bits, SIAM J. Comput. 13 (4) (1984) 850–864.
[14] M. Burmester, J. Munilla, Pre vs post state update: trading privacy for availability in RFID, IEEE Wireless Commun. Lett. 3 (4) (2014) 317–320.