Secure Ownership Transfer in Multi-tag/Multi-owner Passive RFID Systems

Saravanan Sundaresan, Robin Doss *Member, IEEE,* Wanlei Zhou *Senior Member, IEEE*

***Abstract*—In this paper we propose a secure ownership transfer protocol for a multi-tag and multi-owner RFID environment. Most of the existing work in this area do not comply with the EPC Global Class-1 Gen-2 (C1G2) standard since they use expensive hash operations or sophisticated encryption schemes that cannot be implemented on low-cost passive tags that are highly resource constrained. Our work aims to fill this gap by proposing a protocol based on simple XOR and 128-bit Pseudo Random Number Generators (PRNG), operations that can be easily implemented on low-cost passive RFID tags. The protocol thus achieves EPC C1G2 compliance while meeting the security requirements. Also, our protocol provides additional protection using a blind-factor to prevent tracking attacks.**

Keywords: RFID, EPC C1G2, Passive Tags, Ownership Transfer, Multi-owner, Multi-tag.

I. Introduction

Radio Frequency Identification (RFID) enables the automatic identification of objects using radio waves without the need for a physical contact with the objects. One of the important features of an RFID system is *secure ownership transfer* of objects from one owner to another. For example, objects change hands frequently in different stages of a supply chain from manufacturing, to distribution, to warehousing, to retailing, to end-customers. It is imperative to make this transfer happen in a secure fashion and that the internal state of the RFID tag reflects these changes accurately. Ownership transfer requires that control (i.e., communication capabilities) of a tag is transferred from the current Owner/s to the new Owner/s. To elaborate, ownership transfer should ensure that only the new Owners are able to interrogate the tag and the previous Owners are prevented from communicating with the tag. However, in order to prevent against compromise of the ownership transfer process, security of the process needs to be guaranteed. Secure ownership transfer requires at a minimum the establishment of shared secrets between the tags and the new Owners. In order to achieve this, it is important that the establishment of new secrets is achieved in a secure fashion thus preventing the previous Owner from communicating with the tag after the ownership transfer. It is also important that the new Owner is not able to compromise previous communications of the tag. It is therefore imperative that any ownership transfer scheme incorporates security requirements and protects the privacy of both the new and old owners of the tag.

A. Motivation

Most of the approaches to ownership transfer in RFID systems do not comply with the EPC C1G2 standard for passive RFID tags because they use hash functions which require 8000 to 10000 gates [1] making them unsuitable for the low-cost passive tags due to the very limited computational resources. The EPC standard mandates security operations in these tags to Cyclic Redundancy Check (CRC) and 16-bit pseudo random number generators (PRNG). A 16-bit PRNG is not secure enough because it is vulnerable to brute-force attack and hence we recommend the use of 128-bit PRNG which are more secure and implementable in passive tags. Low cost passive tags can accommodate roughly 3K gates to implement security features [1], [2] which is insufficient for standard cryptographic techniques such as RSA [3]. Although cheaper cryptographic alternatives such as Elliptic Curve Cryptography (ECC) exist and schemes such as ERAP [4] based on ECC have been proposed for RFID systems, the practical implementation of ECC is still an open research problem. As demonstrated by Batina *et al.* [5], [6] implementation of ECC would require between $8.2k$ and $15k$ equivalent gates. Complex encryption schemes such as AES takes up to 3400 gates as seen in [7].

Our proposed protocol requires less than 2K gates which is a significant advantage considering the limitations of passive tags. Also, Burmester *et al.* [8] have formally shown using the *Universal Composability* (UC) framework [9] that 128-bit pseudo-random generators meet the security requirements of RFID. It is also noted that it is sufficient for the random numbers to be pseudo-random assuming that all entities of the RFID system have polynomially bounded resources. This assumption holds good for our proposed protocol also. Lee and Hong [10] have proposed an authentication protocol that achieves the required security using 128-bit PRNG with only 1435 gates (within 517 clock cycles and 64B memory). *Our main contributions* can be summarized as: A secure multi-owner, multi-tag ownership transfer protocol that is ultra lightweight in terms of the use of simple XOR and 128-bit PRNG operations that meets the necessary security requirements. Further, it does not use hash functions making it a viable option for large-scale implementations on low-cost passive tags.

The rest of the paper is organized as follows. Section II covers the related work in this area. The protocols are briefly described and vulnerabilities identified. Our proposed protocol is described in section III followed by the detailed security analysis in section IV. Section V concludes the paper.

S Sundaresan, R. Doss and W. Zhou are with the School of Information Technology, Deakin University, Melbourne, Australia. E-mail: ssundare@deakin.edu.au

II. Related Work

One of the earliest schemes proposed for ownership transfer was by Osaka *et al.* [11] based on hash and keyed encryption functions. However, Jappinen and Hamalainen [12] has shown that the scheme suffers from several security flaws such as desynchronization, replay attacks and compromise through noise injection. The improved version [12] has been shown to suffer from desynchronization problems [13] while the noise injection problem has been addressed by Chen *et al.* [14]. The authors propose the use of a hash function in order to protect the integrity of the key being transferred which is similar to the earlier work [12] and therefore suffers from desynchronization issues. Dimitriou [15] proposed "RFIDdot" an ownership transfer scheme based on random nonces and a keyed encryption function making the assumption that key updates are performed in a "private" environment. Such an assumption is questionable; further, the scheme suffers from desynchronization attacks due to selective blocking by an attacker leading to permanent DoS. It also cannot guarantee the privacy of the new owner [13]. Two lightweight ownership transfer protocols (one with a TTP and one without a TTP) are proposed by Kulseng *et al.* [16] based on Physically Unclonable Functions (PUF) and Linear Feedback Shift Registers (LFSR). However, on analysis both the protocols fail to provide the required security properties. As noted in [16] the protocol with TTP suffers from permanent desynchronization when an attacker selectively blocks messages; while the protocol without a TTP is designed based on the assumption that an attacker is not able to eavesdrop on the transmission over the wireless channel. This is not a valid assumption as noted by Kapoor *et al.* [13].

Fouldagar and Afifi [17] propose two privacy preserving schemes for ownership transfer based on hash functions and symmetric key cryptographic functions. In both the schemes the update of the secret keys K_U and K_P is not protected against desynchronization. Seo *et al.* [18] propose a scheme based on a Public Key Infrastructure (PKI) with the tag's computation moved to a "proxy" that manages each tag and is within the backward channel range of each tag. In our opinion, the infrastructure overhead of the scheme and the notion of a "proxy" makes the scheme impractical. More recently, Song and Mitchell [19] have proposed a tag ownership transfer method based on keyed hash functions and one-time tag identifiers using hash-chains. Kapoor and Piramuthu [20] propose ownership transfer schemes based on keyed hash and keyed encryption functions. The protocol with TTP suffers from desynchronisation as the tag updates its secret even before the new secret is given to the new owner by the TTP. The non-TTP version also suffers from vulnerabilities that can lead to forward secrecy compromise and tag cloning attacks. Recently Doss *et al.* [21] has proposed two ownership transfer schemes based on quadratic residues. The closed loop scheme addresses situations where tags are present in the range of both the old and new owners and in the open loop scheme, tags are only in the range of the new owner.

Table I
Symbol Notations

Notation	Description
TID, t_{id}	Unique Tag Identification Number and pre-computed value of $h(TID, t_s)$
OID, o_{id}	Owner ID and pre-computed value of $h(OID, o_s)$
t_s, o_s	Secret key for the tag; Secret key for owner known only to the TTP
$ot_s, ot_s^{'}$	Current and previous shared secret between Owners and Tag-Group
N_s	New Secret generated by TTP for the New Owners
$st_s, st_s^{'}$	Current and previous shared secret between the TTP and the Tag-Group.
$so_s, so_s^{'}$	Current and previous shared secret between the TTP and the New Owners.
$S1_r, S2_r$	Pseudo-Random numbers generated by the TTP
$O1_r, T1_r$	Pseudo-Random numbers generated by the New Owners & Tags in the Tag-Group respectively

III. The Proposed Protocol

Our proposed protocol is based on simple XOR and 128-bit PRNG operations and uses a blind-factor to hide the pseudo-random numbers. The protocol has two phases - the initialization phase and the ownership transfer phase. In the initialization phase, all the tags and owners are setup with their ids, shared/private secrets. This is assumed to occur in a secure environment. The TTP computes $t_{id} = h(TID, t_s)$ and $o_{id} = h(OID, o_s)$ using the secrets t_s, o_s respectively which are known only to the TTP. All the existing Owner ids in all the tags in the Tag-Group are retained as is. After the ownership transfer happens, the new owners use the same ids but have a new secret. The protocol can be easily extended to remove the existing Owner ids and add new ones by sending an additional encrypted message with o_{id} in Step 2A. The TTP shares a secret st_s with the group of tags which the current and new owners do not know. Similarly, the TTP shares a secret so_s with the new owners that are not known to the current owners. The tags store the tuple $\{t_{id}, st_s, st_s^{'},\ o_{id_1}, o_{id_2}, ...o_{id_n}, ot_s, ot_s^{'}\}$. The owners store the tuple $\{o_{id}, so_s, so_s^{'}, TID_1, TID_2, ..TID_j, ot_s\}$. Table I briefly describes the notations used in the proposed protocol.

A. Secure Ownership Transfer Scheme

Step 1: $TTP \rightarrow$ New Owners $\rightarrow TTP$
Step 1A: TTP performs the following:

- Generate pseudo-random number $S2_r$
- Generate a new secret N_s
- For each New Owner i:
  - Compute $X1_i = o_{id_i} \oplus PRNG(so_s \oplus S2_r)$
  - Compute $X2_i = S2_r \oplus PRNG(o_{id_i} \oplus so_s)$
  - Compute $X3_i = N_s \oplus PRNG(o_{id_i} \oplus so_s \oplus S2_r)$
  - For each Tag j in the Tag-Group, compute $X4_j = TID_j \oplus PRNG(N_s \oplus so_s \oplus S2_r)$
  - Compute $X_i^c = PRNG(X3_i \oplus so_s) \oplus PRNG(X4_1 \oplus X4_2 \oplus ... \oplus X4_j \oplus S2_r)$. For ease of representation, this computation is shown in the figure 1 as $X_i^c = PRNG(X3_i \oplus so_s) \oplus PRNG(X4_{(1..j)} \oplus S2_r)$.

  - TTP then sends $X1_i, X2_i, X3_i, X4_{(1..j)}$ and X_i^c to the i^{th} New Owner.

Step 1B: Each New Owner performs the following:

- Extract $S2_r$ using stored values as $X2_i \oplus PRNG(o_{id} \oplus so_s) \rightarrow S2_r$
- If $o_{id} = X1_i \oplus PRNG(so_s \oplus S2_r)$ (or use $so_s^{'}$) then the TTP is authenticated and the Owner knows that the message is for itself, in which case it performs the following. Otherwise, it aborts the protocol. For the reminder of the operations either so_s or $so_s^{'}$ will be used based on which one returned a successful match.
- Check if $PRNG(X3_i \oplus so_s) \oplus PRNG(X4_1 \oplus X4_2 \oplus ... \oplus X4_j \oplus S2_r) = X_i^c$. Otherwise, it aborts the protocol. This validation ensures that $X3_i$ and $X4_{(1..j)}$ are not tampered by an adversary during transmission. For ease of representation, this check is shown in the figure 1 as $PRNG(X3_i \oplus so_s) \oplus PRNG(X4_{(1..j)} \oplus S2_r) = X_i^c$.
- Extract N_s: $X3_i \oplus PRNG(o_{id} \oplus so_s \oplus S2_r) \rightarrow N_s$
- Extract TIDs: $X4_1 \oplus PRNG(N_s \oplus so_s \oplus S2_r) \rightarrow TID_{(1)}$. Repeat this step for $X4_2, X4_3 ... X4_j$ tags.
- Owner inserts the Tag ids $TID_1, TID_2 ... TID_j$ and the shared secret as $ot_s = N_s$
- Generate pseudo-random number $O1_r$
- Compute: $RND_o = O1_r \oplus o_{id} \oplus so_s$
- Compute: $ACK_o = o_{id} \oplus ot_s \oplus PRNG(so_s \oplus O1_r)$
- Owner sends RND_o, ACK_o to the TTP
- If the o_{id} was matched using so_s then the shared secret is updated as $so_s^{'} \leftarrow so_s$ and $so_s \leftarrow PRNG(so_s)$

Step 1C: For each New Owner's reply, TTP performs the following:

- Extract $O1_r$ using stored values as $RND_o \oplus o_{id_i} \oplus so_s \rightarrow O1_r$
- Check if $o_{id_i} \oplus N_s = ACK_o \oplus PRNG(so_s \oplus O1_r)$. If yes, it confirms the authenticity of the new Owner and that he has the new secret.
- If acknowledgements are not received from all the owners within a stipulated time, the process restarts from Step 1A. Otherwise the shared secret so_s is updated as $so_s \leftarrow PRNG(so_s)$ and goes to Step 2.

Step 2: $TTP \rightarrow$ Tags $\rightarrow TTP$

Step 2A: The TTP performs the following:

- Generate pseudo-random number $S1_r$
- Compute $M1 = N_s \oplus PRNG(st_s \oplus S1_r)$
- Compute $M^c = PRNG(M1 \oplus S1_r) \oplus st_s$
- Then for each Tag j in the Tag-Group:
  - Compute $M2_j = t_{id_j} \oplus PRNG(t_{id_j} \oplus st_s \oplus S1_r)$
  - Compute $M3_j = S1_r \oplus PRNG(t_{id_j} \oplus st_s)$
  - TTP then sends $M1, M2_j, M3_j$ and M^c to the j^{th} tag in the Tag-Group

Step 2B: Each tag in the Tag-Group performs the following:

- Extract $S1_r$ using stored values as $M3_j \oplus PRNG(t_{id} \oplus st_s) \rightarrow S1_r$
- If $t_{id} = M2_j \oplus PRNG(t_{id} \oplus st_s \oplus S1_r)$ (or use $st_s^{'}$) then the TTP is authenticated and the tag knows that the message is for itself, in which case it performs the following. Otherwise, it aborts the protocol. For the reminder of the operations either st_s or $st_s^{'}$ will be used based on which one returned a successful match.
- Check if $PRNG(M1 \oplus S1_r) = M^c \oplus st_s$. This validation ensures that $M1$ is not tampered by an adversary during transmission. Otherwise, the protocol aborts.
- The new secret N_s is extracted as $M1 \oplus PRNG(st_s \oplus S1_r) \rightarrow N_s$
- The current and previous secrets $ot_s, ot_s^{'}$ are updated as $ot_s \leftarrow N_s$; $ot_s^{'} \leftarrow N_s$
- Generate a pseudo-random number $T1_r$
- Compute $RND_t = T1_r \oplus t_{id} \oplus st_s$
- Compute $ACK_t = t_{id} \oplus ot_s \oplus PRNG(st_s \oplus T1_r)$
- Tag sends RND_t, ACK_t to the TTP
- If the t_{id} was matched using st_s then the shared secret is updated as $st_s^{'} \leftarrow st_s$ and $st_s \leftarrow PRNG(st_s)$

Step 2C: For each tag reply, the TTP performs the following:

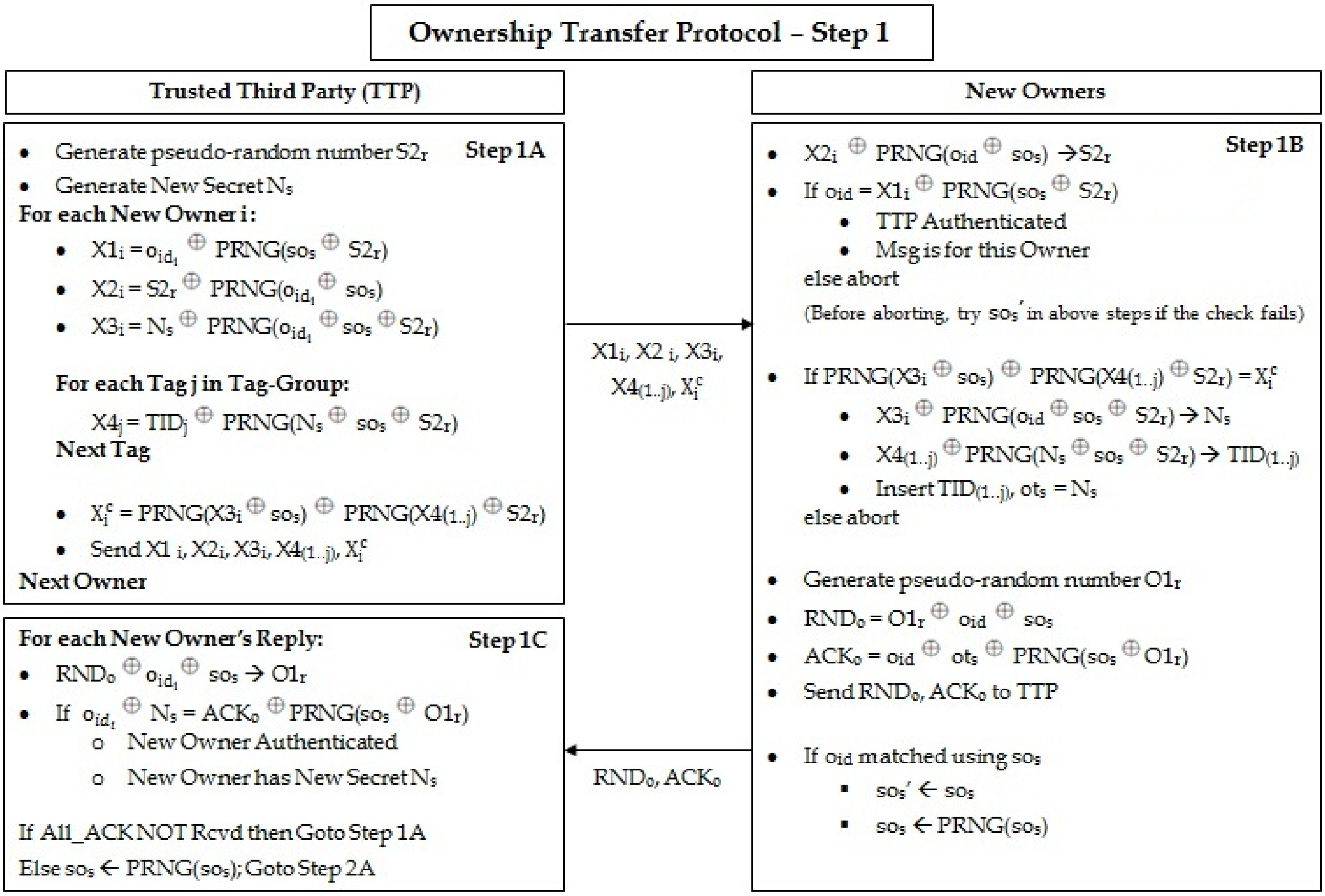
- Extract $T1_r$ using stored values as $RND_t \oplus t_{id_j} \oplus st_s \rightarrow T1_r$
- Check if $t_{id_j} \oplus N_s = ACK_t \oplus PRNG(st_s \oplus T1_r)$. If yes, it confirms the authenticity of the Tag and that its new owner secret ot_s has been successfully updated to N_s.
- If acknowledgements are not received from all the tags in the Tag-Group within a stipulated time, the process restarts from Step 2A. Otherwise the shared secret st_s is updated as $st_s \leftarrow PRNG(st_s)$. This completes the *ownership transfer process.*

B. Ownership Test Protocol

We define a ownership test protocol that is carried out in a virtual environment without any adversarial interference [22]. Because of this secure channel assumption, messages are not encrypted. For environments where this secure channel cannot be assumed, a mutual authentication protocol like [2] can be used to test ownership. For each New Owner i and for each Tag j in Tag-Group, the New Owner 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 $M_{tst} = o_{id}^i \oplus ot_s \oplus t_{id}$ and sends it back. For each Tag's Reply received, and for each Tag in the Tag-Group, each new Owner checks if $o_{id}^i \oplus ot_s = M_{tst} \oplus t_{id}^j$. If yes, the tag ownership is confirmed and the for-loop is exited at this point to reduce processing time. If all tags are not identified by all owners within a stipulated time, the ownership test protocol is restarted.

IV. Security Analysis

Basic Privacy/Eavesdropping: The protocol ensures basic communication privacy since one or more secret keys is used in combination with freshly generated pseudo-random numbers in each communication. As the messages are protected with the secret keys and also the pseudo-random number is hidden using a blind-factor, the attacker will not be able decipher anything from the messages even if they are captured using eavesdropping attack.



Ownership Transfer Protocol – Step 1

Trusted Third Party (TTP)

Step 1A

- Generate pseudo-random number $S2_r$
- Generate New Secret N_s

For each New Owner i:

- $X1_i = o_{id_i} \oplus PRNG(so_s \oplus S2_r)$
- $X2_i = S2_r \oplus PRNG(o_{id_i} \oplus so_s)$
- $X3_i = N_s \oplus PRNG(o_{id_i} \oplus so_s \oplus S2_r)$

For each Tag j in Tag-Group:

$X4_j = TID_j \oplus PRNG(N_s \oplus so_s \oplus S2_r)$

Next Tag

- $X_i^c = PRNG(X3_i \oplus so_s) \oplus PRNG(X4(1..j) \oplus S2_r)$
- Send $X1\ i, X2_i, X3_i, X4_{(1..j)}, X_i^c$

Next Owner

For each New Owner's Reply:

Step 1C

- $RND_o \oplus o_{id_i} \oplus so_s \rightarrow O1_r$
- If $o_{id_i} \oplus N_s = ACK_o \oplus PRNG(so_s \oplus O1_r)$
  - New Owner Authenticated
  - New Owner has New Secret N_s

If All_ACK NOT Rcvd then Goto Step 1A

Else $so_s \leftarrow PRNG(so_s)$; Goto Step 2A

$X1_i, X2\ i, X3_i, X4_{(1..j)}, X_i^c$

RND_o, ACK_o

New Owners

Step 1B

- $X2_i \oplus PRNG(o_{id} \oplus so_s) \rightarrow S2_r$
- If $o_{id} = X1_i \oplus PRNG(so_s \oplus S2_r)$
  - TTP Authenticated
  - Msg is for this Owner

  else abort

  (Before aborting, try so_s' in above steps if the check fails)
- If $PRNG(X3_i \oplus so_s) \oplus PRNG(X4(1..j) \oplus S2_r) = X_i^c$
  - $X3_i \oplus PRNG(o_{id} \oplus so_s \oplus S2_r) \rightarrow N_s$
  - $X4_{(1..j)} \oplus PRNG(N_s \oplus so_s \oplus S2_r) \rightarrow TID_{(1..j)}$
  - Insert $TID_{(1..j)}$, $ot_s = N_s$

  else abort
- Generate pseudo-random number $O1_r$
- $RND_o = O1_r \oplus o_{id} \oplus so_s$
- $ACK_o = o_{id} \oplus ot_s \oplus PRNG(so_s \oplus O1_r)$
- Send RND_o, ACK_o to TTP
- If o_{id} matched using so_s
  - $so_s' \leftarrow so_s$
  - $so_s \leftarrow PRNG(so_s)$

Figure 1. Proposed Ownership Transfer Protocol for Multi-Tag Multi-Owner Environment - Step 1

Mutual Authentication: When the tag performs the check If "$t_{id} = M2 \oplus PRNG(t_{id} \oplus st_s \oplus S1_r)$" and if they match, the tag can be sure that the message comes from the legitimate TTP because only it knows the shared secret st_s. Similarly the TTP verifies the response ACK_t received from the tag, thereby authenticating it. The same principle applies to the communication between the TTP and the New Owners.

Tag/Owner Anonymity: Messages $X1..X4, X^c, M1..M3$, M^c, RND_o, ACK_o, RND_t and ACK_t implicitly contain tag id t_{id} or owner id o_{id} but all are enciphered well and cannot be detected by the attacker. First, t_{id} contains only the pre-computed hash value of the actual TID. Due to the one-way property of the hash function, the attacker will not be able to find TID from t_{id}. Secondly, in order to obtain t_{id} from $M2$, the attacker has to know two different unknowns. Recall that $M2 = t_{id} \oplus PRNG(t_{id} \oplus st_s \oplus S1_r)$ in which st_s is the shared secret between the TTP and the Tag-Group and that changes after every successful run, $S1_r$ is a freshly generated by TTP for each run, which is not sent in the clear but is hidden using the blind-factor. Also, in $M2$, the XOR operation between t_{id}, st_s and $S1_r$ is further randomized using the PRNG operation. All of these measures prevent the real tag id from being revealed. Same level of protection is ensured for owner id OID as well.

Tag/Owner Location Privacy: The principles used above applies to providing Tag/Owner location privacy also. Even if the same tag-id/owner-id were to be sent repeatedly, due to the use of pseudo-random numbers (hidden during transmission), and also the further randomization of the XOR operation between the shared secret and the pseudo-random number using the PRNG operation, it ensures that the messages are different every time they are sent thus satisfying the security property.

Replay Attack (Owner/Tag/Server(TTP) Impersonation): Let us say an attacker captures n communications for later use. Let the messages in the n^{th} capture be denoted as $X1_i^n..X4_{1..j}^n$, $M1^n, M2_j, M3_j$, $RND_t^n, ACK_t^n, RND_o^n, ACK_o^n$. Now let us say the attacker replays the $(n-5)^{th}$ messages represented as $X1_i^{(n-5)}..X4_{1..j}^{(n-5)}$, $M1^{(n-5)}, M2_j^{(n-5)}, M3_j^{(n-5)}$, $RND_t^{(n-5)}, ACK_t^{(n-5)}, RND_o^{(n-5)}, ACK_o^{(n-5)}$. The tags/owners/TTP will not be able to extract the pseudo-random numbers properly because of the mismatch in the shared secrets st_s and so_s which are updated during each protocol run. Secondly the uniqueness of all the messages during each round

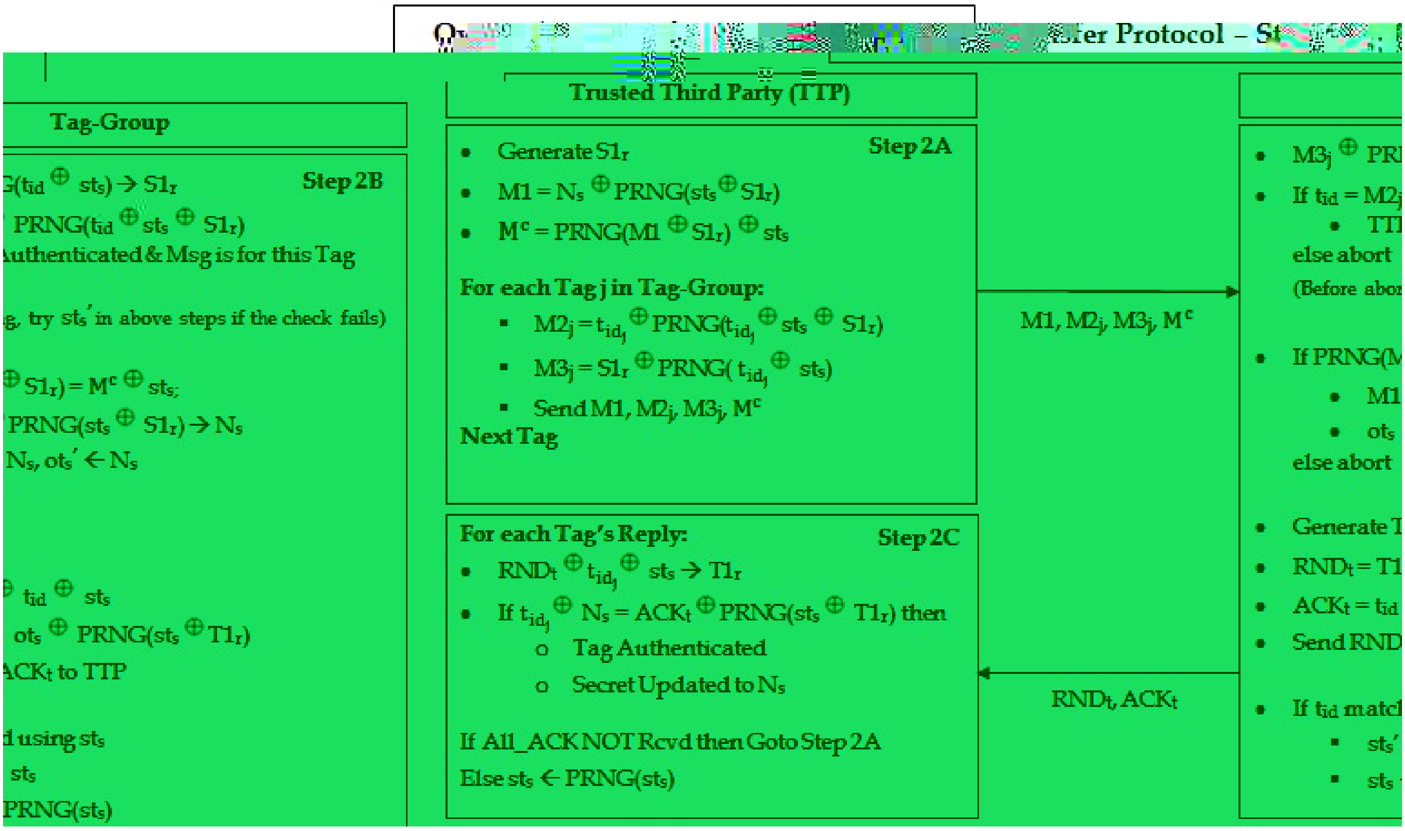


Figure 2. Proposed Ownership Transfer Protocol for Multi-Tag Multi-Owner Environment - Step 2

Table II
COMPARISON OF SECURITY AND PRIVACY PROPERTIES

Scheme	P1	P2	P3	P4	A1	A2	A3
Osaka *et al.* [11]	✓	No	No	✓	✓	No	✓
Fouldagar *et al.* [17]	✓	✓	✓	✓	✓	No	✓
Kulseng *et al.* [16]	✓	✓	✓	✓	✓	No	✓
Dimitriou [15]	✓	No	✓	No	✓	No	No
Song and Mitchell [19]	✓	✓	✓	No	✓	No	§
Kapoor *et al.* [13]	✓	§	§	No	✓	✓	✓
Our Scheme	✓	✓	✓	✓	✓	✓	✓

P1: Tag Anonymity	P2: Tag Location Privacy	P3: Forward Secrecy
P4: Forward Untraceability	A1: Replay Attack	A2: DoS/De-synchronization Attack
A3: Server Impersonation	✓ - Fully Satisfied	§: Partially satisfied under certain assumptions.

is ensured by the freshly generated pseudo-random numbers which are hidden during the transmission. This ensures that verifications such as "if $t_{id} = M2 \oplus PRNG(t_{id} \oplus st_s \oplus S1_r)$", "if $o_{id} = X1 \oplus PRNG(so_s \oplus S2_r)$" will also fail for the same reason. Hence the attacker will not be able to successfully impersonate a tag or a owner or a server by replaying messages from the previous communications.

Forward Secrecy: To show that the protocol achieves forward secrecy, we prove that even if a tag is compromised and its current resident data is obtained by an attacker, it does not enable tracing any of the previous communications. RND_t is computed using the freshly generated pseudo-random number $T1_r$ (hidden during transmission) along with t_{id} and st_s which changes after every protocol run). Similarly ACK_t is computed the same way and in addition, the XOR operation between st_s and $T1_r$ is further randomized using the 128-bit PRNG operation that ensures ACK_t is unique each time. So even if the attacker knows st_s and $st_s^{'}$ the previous messages of the tags cannot be deciphered thus ensuring forward secrecy. Same principle applies to RND_o, ACK_o and also the messages in the forward channel.

Forward Untraceability: The protocol ensures that the old owner is unable to trace or communicate with the tag post-ownership transfer by making sure that the old owner is not able to learn or compromise the new secret that has been established between the tag and the new owners. This is accomplished by enciphering the new secret N_s in $X3$ using the owner id o_{id}, the shared secret so_s which is unknown to the old owner and $S2_r$ which is a freshly generated pseudo-random number (hidden during transmission). Also, the XOR operation between o_{id}, so_s and $S2_r$ is further randomized using the PRNG operation that provides additional security. The same principle is applied when computing $M1$. Thus the protocol achieves forward untraceability.

DoS/De-synchronization Attack: An adversary can cause Denial of Service (DoS) by de-synchronizing the keys between the TTP and the Tag-Group (or) the keys between the TTP and the New Owners. In the protocol, the tag/new-owner update their secrets before sending the acknowledgement ACK_t. If ACK_t were to be blocked by the attacker or lost due to other communication issues it will prevent the TTP from updating its key. In order to overcome this, the tags and the new-owners store the shared-secret from the current and previous round (ex: st_s and $st_s^{'}$). As seen in the protocol, messages are verified using the previous secret if the current secret did not return a match. If a match occurs using either one of the values, the protocol run will complete successfully thus preventing DoS/De-synchronization attacks.

A. Comparison with Other Protocols

In Table II we compare the security properties of the various ownership transfer protocols that have been proposed. We observe that the schemes proposed by Osaka *et al.*, and Dimitriou fail to meet the security property of tag location privacy while Kapoor and Piramuthu's scheme only satisfies this property under the assumption that a third party cannot eavesdrop over the wireless channel. Forward secrecy is not satisfied by Osaka *et al.* and partially satisfied by Kapoor and Piramuthu's schemes. The property of forward untraceability is not satisfied by many of the current schemes. The schemes proposed by Dimitriou, Song and Mitchell and Kapoor and Pirmuthu all fail to guarantee this property. All schemes are protected against replay attacks. Protection against desynchronisation is only achieved by Kapoor and Piramuthu. Further, Dimitriou's scheme is vulnerable to server impersonation attacks while Song and Mitchell's scheme is only partially secure. As noted in the security analysis, our scheme satisfies all of the required security properties.

V. Conclusion

In this paper we have proposed a secure ownership transfer protocol for multi-tag multi-owner RFID systems. The protocol is ultra-lightweight as only simple XOR and 128-bit PRNG functions are employed. Importantly, the protocol does not use hash functions thereby meeting the EPC C1G2 standards. Hence the scheme is practical and can be implemented on passive tags. Security analysis of our proposed methods show that they achieve the required properties of tag anonymity, tag location privacy, forward secrecy, forward untraceability while being resistant to replay, desynchronisation and server impersonation attacks. In future work our aim is complete a test bed implementation of the proposed scheme.

References

[1] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan, "Lightweight Secure Search Protocols for Low-cost RFID Systems," *2009 29th IEEE International Conference on Distributed Computing Systems*, pp. 40–48, Jun 2009.

[2] R. Doss, S. Sundaresan, and W. Zhou, "A practical quadratic residues based scheme for authentication and privacy in mobile RFID systems," *Ad Hoc Networks*, July 2012. [Online]. Available: 10.1016/j.adhoc.2012.06.015

[3] A. Juels and S. Weiss, "Authenticating Pervasive Devices with Human Protocols," *Advances in Cryptology (CRYPTO 2005), Lecture Notes in Computer Science*, vol. 3621, pp. 293–308, 2005.

[4] S. Ahamed, F. Rahman, and M. Hoque, "ERAP: ECC based RFID Authentication Protocol," in *12th IEEE International Workshop on Future Trends of Distributed Computing Systems*, 2008, pp. 219–225.

[5] L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede, "An elliptic curve processor suitable for rfid-tags," *Cryptology ePrint Archive, Report 2006/227*, 2006.

[6] Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede, "Elliptic Curve Based Security Processor for RFID," *IEEE Transactions on Computers*, vol. 57, no. 11, pp. 1514–1527, 2008.

[7] M. Feldhofer and C. Rechberger, "A Case Against Currently Used Hash Functions in RFID Protocols," in *On the Move to Meaningful Internet Systems 2006 – OTM 2006*, ser. Lecture Notes in Computer Science, vol. 4277, Nov 2006, pp. 372–381.

[8] M. Burmester and B. D. Medeiros, "RFID Security : Attacks, Countermeasures and Challenges," 2007. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.87.941

[9] "Univerally Composable Security: A new paradigm for cryptographic protocols," *IEEE Symp. On Foundations of Computer Science (FOCS 2001)*, pp. 136–145, 2001.

[10] H. Lee and D. Hong, "The tag authentication scheme using self-shrinking generator on RFID system," *Transactions on Engineering, Computing and Technology*, vol. 18, pp. 52–57, 2006.

[11] K. Osaka, T. Takagi, K. Yamazaki, and O. Takahashi, "An Efficient and Secure RFID Security Method with Ownership Transfer," in *2006 International Conference on Computational Intelligence and Security*. IEEE, Nov 2006, pp. 1090–1095.

[12] P. Japinnen and H. Hamalainen, "Enhanced RFID security method with ownership transfer," in *Proc. of International Conference on Computational Intelligence and Security*, Dec 2008, pp. 382–385.

[13] G. Kapoor and S. Piramuthu, "Vulnerabilities in some recently proposed RFID ownership transfer protocols," *IEEE Communications Letters*, vol. 14, no. 3, pp. 260–262, Mar 2010.

[14] H. Chen, W. Lee, Y. Zhao, and Y. Chen, "Enhancement of the RFID Security Method with Ownership Transfer," in *3rd Int. Conf. Ubiquitous Information Management and Communication (ICUIMC 09)*, Jan 2009, pp. 251–254.

[15] T. Dimitriou, "RFIDDOT: RFID Delegation and Ownership Transfer made simple," in *4th Int. Conf. Security and Privacy in CommunicationNetworks (SecureCom 08)*, 2008.

[16] L. Kuseng, Z. Yu, Y. Wei, and Y. Guan, "Lighweight Mutual Authentication and Ownership Transfer for RFID Systems," in *IEEE INFOCOM 2010*, Mar 2010, pp. 1–5.

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

[18] Y. Seo, T. Asano, H. Lee, and K. Kim, "A lightweight protocol enabling ownership transfer and granular data access of RFID tags," in *2007 Symposium on Cryptography and Information Security*, 2007, pp. 23–26.

[19] B. Song and C. J. Mitchell, "Scalable RFID security protocols supporting tag ownership transfer," *Computer Communications*, vol. 34, no. 4, pp. 556–566, Apr 2011.

[20] G. Kapoor and S. Piramuthu, "Single RFID Tag Ownership Transfer Protocols," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 99, pp. 1–10, 2011.

[21] R. Doss, W. Zhou, and S. Yu, "Secure RFID Tag Ownership Transfer Based on Quadratic Residues," *IEEE Transactions On Information Forensics And Security*, vol. 8, no. 2, pp. 390–401, Feb 2013.

[22] T. van Deursen, S. Mauw, S. Radomirovic, and P. Vullers, "Secure Ownership and Ownership Transfer in RFID Systems," *European Symposium on Research in Computer Security (ECORICS) 2009 (LNCS 5789)*, pp. 637–654, 2009.