Authentication Protocols for an Object with Dynamic RFID Tags

Selwyn Piramuthu(✉)

Information Systems and Operations Management,
University of Florida, Gainesville, FL 32611-7169, USA
selwyn@ufl.edu

Abstract. A majority of existing RFID authentication protocols consider tagged items that are independent of other tagged items. However, as RFID tags permeate to item-level granularity where several items comprise an object of interest there is a need to develop protocols that seamlessly accommodate inclusion and exclusion of tags on such an object. We propose protocols for this scenario.

1 Introduction

As RFID tags become ubiquitous, there is a need to develop authentication protocols that ensure secure communication between the tagged item and the reader. This is a challenging task given the over-the-air communications medium and the RFID tag resource constraints including its processing capacity, memory, and power source. Since the early 2000s, there has been an explosion of interest in this area both among researchers and practitioners. While there is a vast amount of literature on RFID authentication protocols (e.g., http://www.avoine.net/rfid/index.html), several of the proposed protocols have been plagued by (1) vulnerabilities to attack by a resourceful adversary, and/or (2) the use of primitives that are not lightweight and therefore cannot be implemented in commonly used tags.

Given the diversity of idiosyncrasies streams of research have developed over the years (e.g., [4]). Among the various streams, the ones that deal with the simultaneous authentication of multiple tags are those that evolved from the Yoking Proof introduced by Juels [2]. These protocols authenticate the simultaneous presence of multiple tags in the field of the reader. While this works well for authenticating independent items, there is a need for protocols that consider objects with multiple components with their individual RFID tags. This is predicated on recent trends where, for example, item-level or component-level tagging is in place and these items or components are highly likely to be added or removed from the primary object over time. Objects with multiple RFID-tagged components do not, generally speaking, have the need for authentication of tags as in yoking proof and its variants since these components are attached

R. Doss et al. (Eds.): FNSS 2018, CCIS 878, pp. 93–101, 2018.
https://doi.org/10.1007/978-3-319-94421-0_7

or bundled together to (form) the object. However, these situations dictate a need for continual communication between the object and its component parts as a group.

Consider a supply chain where individual items are RFID-tagged. For example, consider a stack of item-level RFID tagged Wrangler jeans of a certain size (say, 30 × 30) on an RFID-tagged pallet that leave the manufacturing facility to a Walmart warehouse. When several such pallets reach the warehouse, their contents are redistributed and then sent over to individual stores. For example, 20 jeans of size 30 × 30, 15 jeans of size 36 × 36, and 25 jeans of size 42 × 34 maybe included in a pallet that is shipped to a Walmart store in Gainesville, Florida. From the perspective of a pallet, various different items (different quantities of different sizes of Wrangler jeans in this example) are associated with it across different points in time. It should be noted that once its contents are assembled together, the pallet is tracked and traced as a whole and its contents are generally not scanned until it is 'disassembled' and its contents change. Considered at a higher level of granularity, a delivery truck (with RFID reader) can continually communicate with its pallets to determine their destination, which can be modified *en route* when necessary and appropriate. Incidentally, Wrangler jeans' sold in Walmart stores in the U.S. are item-level RFID-tagged beginning August 2010 [1].

Objects with multiple RFID tags are not uncommon. Another example scenario that illustrates this include a primary object (e.g., car chasis) with several attached parts (e.g., car door, wheels) each with its own RFID tag. In such scenarios, both the number of tags as well as the individual tags themselves may vary over time. I.e., when a tire is replaced, the new tire may come with its own embedded RFID tag; when the owner decides to add a GPS system, it may come with its own RFID tag; when the spare tire is removed from the car, there would be one less RFID tag on the car. As seen from these example scenarios, the set of component RFID-tagged items that belong to the main object (here, delivery truck and car respectively) varies over the lifetime of the object (i.e., delivery truck, car). Clearly, there is a need to manage the 'content' of such an object over time from an authentication perspective. Generally speaking, delivery truck X is not interested nor required to know details of the content of delivery truck Y (where X $\neq$ Y) in a similar vein as car A is not interested in information about car B's (A $\neq$ B) speaker system.

We propose authentication protocols that address inclusion and exclusion of several components over time. These protocols avoid some of the identified vulnerabilities of the protocol presented in [5] while being relatively lightweight.

This paper is organized as follows: The next section provides a sketch of the proposed protocol for multiple tags on an object. Section 3 provides an alternative approach to the same scenario. Section 4 provides a brief security analysis of the proposed protocol. Section 5 concludes the paper with a brief discussion.

2 Protocols for Multi-tagged Object

The following notations are used throughout the paper:

- N_t, N_p, N_r, N_u: random n-bit nonce
- s_c, s_{c+1}: group of tags' current and subsequent keys
- $\{\}_k$: keyed (with key k) encryption function
- t_j: shared secret between tag$_j$ and TTP
- r_i: shared secret between Reader R_i and TTP
- id_{t_j}: tag t_j identifier.

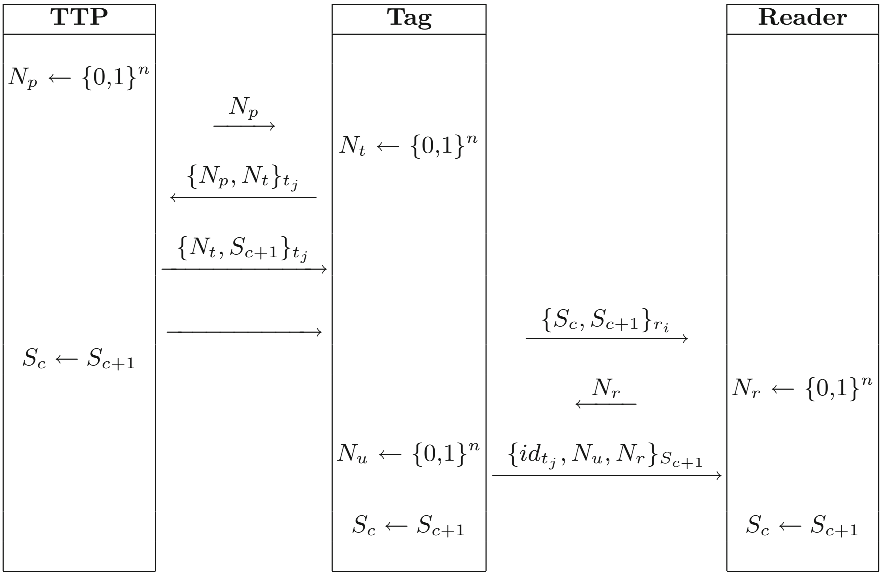

Fig. 1. The proposed protocol

2.1 The Proposed Protocol

There are several entities in this context - a primary object (e.g., car) and a set of component items (e.g., tire, door) that belong to the primary object and the RFID tags on the component items are associated with (the RFID tag on) only one primary object at any given point in time. We do not consider the possibility where a component item could simultaneously belong to several primary objects. The process of inclusion and exclusion of component tags is accomplished in the proposed protocol through a common shared secret key among all the included tags. We assume that a TTP mediates between the reader and tags in accomplishing this change in shared key. The actors involved in this protocol include the reader, the TTP, and every tag that is a part of the object of interest either before or after components (tags) are added or removed.

We assume that every component (tag) that is a part of the object of interest share a common secret key (s_c). This key is updated every time the object of interest experiences addition or removal of a component or group of components. The primary purpose here is to ensure that the updated key is known only to the reader, the TTP, and the tags that are currently attached to the object. The components (tags) that were dropped from this object should not have knowledge of this new shared key. This protocol is repeated for each tag that is associated with the object including those that are present on the object and those that were just removed from the object.

The reasoning for adopting a single common key are (1) ease of key maintenance and (2) fewer messages from reader to tags in the long run since all tags understand any given message that is encrypted with the common key. Drawbacks of this setup include the potential for compromising the entire system when a key is compromised and the initial setup cost of changing every tag's key when a tag enters or leaves the 'system.'

The proposed protocol follows three stages: TTP updates key and communicates this to the component tags, the reader is updated on the new component key, and the reader authenticates the tags.

The TTP initiates the process when a component is either added or removed from the object by generating and sending a nonce (N_p) to all currently existing component tags on the object. These tags then respond by generating a nonce and encrypting both nonce using their shared secret with the TTP (i.e., t_j).

The TTP then sends the updated (group-)key to the component tags encrypted with their shared keys. The component tags update their keys and acknowledge receipt of the same to the TTP. Now, the reader is informed of the new component key through messages that are encrypted using the shared key between reader and TTP.

Finally, the reader authenticates the tags by sending them a nonce and the tags respond by encrypting with the new key a message including their ID, a new nonce and the reader's nonce. This completes the process of updating the common key among the tags.

The new common component key is not known to the (component) tags that were just excluded from the object since they do not receive this new key from the TTP. If and when an excluded tag gets assigned to another object (e.g., an used tire from car A is put on car B after appropriate retreading), the previous reader (here, car A) will not have access to it since the previous reader cannot decrypt communication between TTP and new reader (here, car B).

3 Alternative Approach

The following (set of) protocols may be considered as a solution to the same problem, but the TTP does not have to be invoked for every update of the group key. If we do it this way, then the adversary could record messages and then later crack open a tag to obtain S_c. By repeatedly applying the hash he could end up at the S_c that was used for this encryption. A possible way to address this is to not use S_c for encryption, but a "salted" version of it (e.g. $h(\text{salt}_i, S_c)$. An attacker then additionally needs to have the "salt" which he can only have if he eavesdropped on *all* group key updates.

We propose a set of three protocols to perform different tasks. An *initialization* protocol run between a reader R, a tag T, and a trusted third party TTP. The initialization protocol writes the group key to the tag in a secure and private manner. The *group authentication* protocol authenticates tags to a reader based on the group key. The *group update protocol* updates the group key of a tag to its next value. The next value is the previous value in the hash chain and thus the validity of the new key can be verified by the tag.

We assume that each tag is equipped with an identity id_{t_j} and a key k_j. Readers are equipped with a key r_i. The keys r_i and k_j are shared with the trusted third party. The idea behind our protocols is that tags that belong to the same group share a group key S_c. If a tag has to be included or excluded

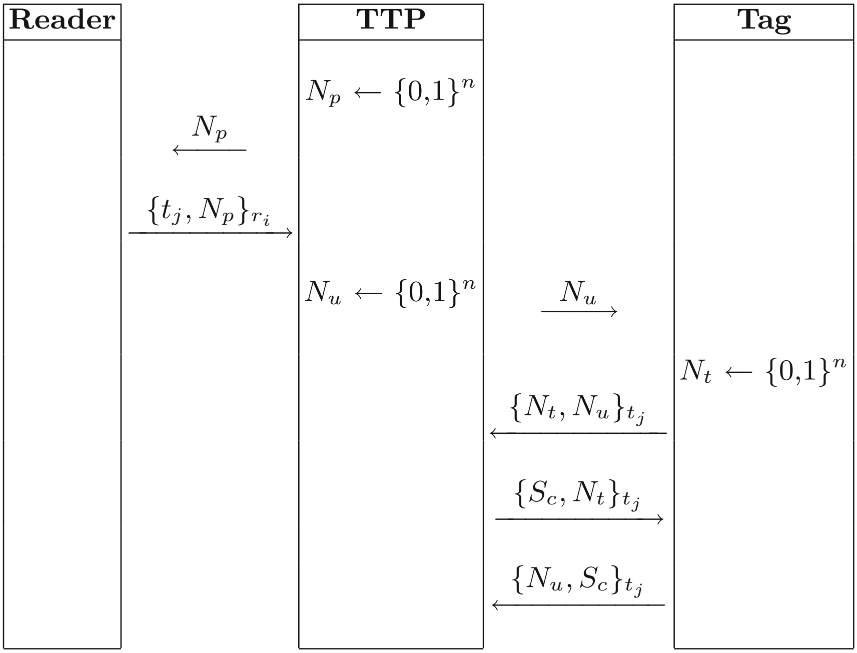

Fig. 2. Initialization protocol

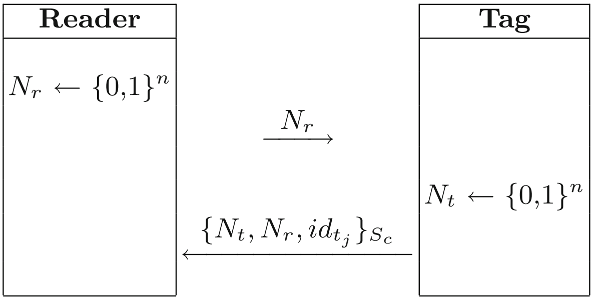
Reader
Tag
$N_r \leftarrow \{0,1\}^n$
N_r
$N_t \leftarrow \{0,1\}^n$
$\{N_t, N_r, id_{t_j}\}_{S_c}$

Fig. 3. Group authentication protocol

all keys get updated by a protocol involving tag and reader. The new group key S_{c+1} is chosen such that $S_{c+1} \leftarrow hash(S_c)$.

The *initialization* protocol is executed between a reader R, a tag T and a trusted third party TTP. It allows the reader to update the group key on a tag without knowing the tag-specific secret k_j. To update the secret, the TTP challenges the reader with a nonce N_p. The reader replies with the group key S_c and N_p encrypted under the secret r_i. The TTP updates the group key on the tag as follows. He challenges the tag with a nonce N_u. The tag generates a nonce N_t and encrypts both these under the key t_j. The TTP now sends the group key S_c and the tag nonce after which the tag updates the group key. Finally, the tag acknowledges the receipt of the message by encrypting the nonce N_u and the group key S_c for the TTP. The protocol is depicted in Fig. 2.

The *group authentication* protocol authenticates a tag T to a reader R based on the group key S_c. The protocol, depicted in Fig. 3, follows a challenge-response

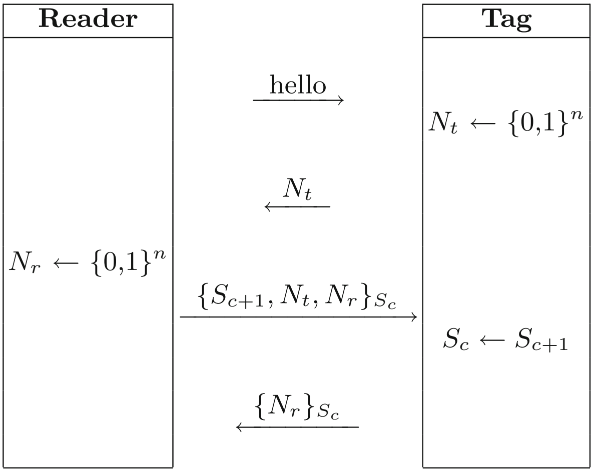
Reader
Tag
hello
$N_t \leftarrow \{0,1\}^n$
N_t
$N_r \leftarrow \{0,1\}^n$
$\{S_{c+1}, N_t, N_r\}_{S_c}$
$S_c \leftarrow S_{c+1}$
$\{N_r\}_{S_c}$

Fig. 4. Group update protocol

structure. The reader R initiates the protocol by sending a nonce N_r to the tag. The tag generates a nonce N_t and replies with the encryption of N_r, N_t, and id_{t_j} under the group key S_c.

The *group update* protocol (see Fig. 4) updates the group key S_c on a tag to the new value S_{c+1}. The reader initiates the protocol by sending a hello message. The tag responds by generating and sending a nonce N_t. The reader generates a nonce N_r and replies with both nonce and the new key S_{c+1} encrypted with the previous key S_c. The tag verifies that S_c is the preimage of S_{c+1} and updates its key. It then responds with the encryption of N_r under the new key.

4 Security Analysis

We do not assume the presence of secure channel between any pairs of entities. We assume the existence of an online TTP. The protocols proposed are to be executed during the physical transfer of a tagged item either into a group or away from a group of items.

The proposed protocols have several characteristics that ensure their security. Freshly-generated nonce (N_p, N_r, N_u, N_t) are used during every run of the protocol. Knowledge of any one of the shared secrets (t_j, r_i) does not lead to any advantage to the adversary since the authentication protocol cannot be successfully completed without knowledge of both the shared secrets (Figs. 1 and 2). However, it is difficult to retrieve any of the shared secrets from passively observing the messages passed among tag, TTP, and reader or even through active capture and modification of messages.

We now consider a few specific attacks on such authentication protocols.

Tag/Reader Anonymity: The tag and reader identification information (e.g., secret keys) are protected from the possibility of information leakage since this information can be used to track and/or trace the tag or (mobile) reader. This is significant since knowledge of such information can allow for the possibility of cloning the tag or reader. We include the possibility of the reader being mobile, as is the case in some RFID applications.

Forward Security: If all shared secrets are somehow known to an adversary, these secrets can be used to decrypt all earlier messages that also include the group key.

Tag/Reader Location Privacy: Since the messages are seemingly random between any two authentication rounds, it is difficult for an adversary to use any of the messages to track the tag and/or the (mobile) reader.

Secrecy/Data Integrity and Authenticity: The integrity of the messages passed between tag and reader is ensured by not sending anything that could compromise the security of the protocol in cleartext. The protocols are designed to be secure and to maintain the secrets regardless of active or passive attacks from adversaries.

DoS/Desynchronization: Since the shared secret keys are not updated after every authentication round, desynchronization is not an issue. The possibility for Denial of Service (DoS) attacks in the proposed protocol is only through blocking and/or modification of message(s). Blocking messages will not grant an adversary any advantage: the reader and TTP wait for acknowledgement message from the recipient of their message within a pre-determined amount of time, and aborts if this does not happen. Modification of any of the messages by an adversary similarly will not allow for protocol compromise. The group key is updated at the very end (Fig. 1), after which the TTP and reader store both the current and previous group keys just in case of DoS attack. These attacks therefore will not succeed.

Passive Replay: Passive replay of any of the three messages that are passed between tag and reader from a previous authentication round will not result in successful authentication due to the existence of N_p, N_t, N_r, N_u that introduce sufficient randomness in the passed messages during each authentication round.

Reader/Tag Impersonation Attack: For an adversary to impersonate a reader, tag, or TTP to one another, it should have the ability to generate messages that seem appropriate and valid to the recipient. An adversary cannot successfully impersonate any entity to any other entity (here, TTP, tag, reader) due to the built-in dependencies among the messages in the authentication protocols.

5 Discussion

Ownership transfer protocols (e.g., [3,6]) are essential for seamless integration of RFID-tagged items in environments such as supply chains. Although not too common at this point in time, it won't be too long before components with RFID tags are put together in a higher-level object with its own RFID tag and possibly a reader. As components enter and leave the domain of the object of interest over time, there is a need to capture this dynamic and be able to deal with the related constraints including those associated with privacy and security issues. The protocol presented in this paper is an attempt at addressing ownership transfer issues from the perspective of component tags and the changing set of ownership from the perspective of the primary object.

It is likely that whenever a group of RFID-tagged items are present, it might be necessary to verify that all these tags are indeed simultaneously present together. We did not consider this scenario since there exist protocols (e.g., Yoking Proof and its variants) that are exclusively designed to accomplish this purpose. Such a protocol can easily be appended to the protocols presented in this paper to form a complete suite of multi-tag authentication and verification protocols.

Acknowledgement. I thank Ton van Deursen for discussions/feedback during the writing of this paper. All error(s) remain my own.

References

1. Bustillo, M.: Wal-Mart radio tags to track clothing. Wall Street J. Bus. Technol. Sect. (2010)
2. Juels, A.: Yoking-Proofs for RFID tags. In: Proceedings of the First International Workshop on Pervasive Computing and Communication Security, pp. 138–143. IEEE Press (2004)
3. Kapoor, G., Piramuthu, S.: Single RFID tag ownership transfer protocols. IEEE Trans. Syst. Man. Cybern. Part C **42**(2), 164–173 (2012)
4. Piramuthu, S.: Lightweight cryptographic authentication in passive RFID-tagged systems. IEEE Trans. Syst. Man Cybern. Part C **38**(3), 360–376 (2008)
5. Piramuthu, S.: Inclusion/exclusion protocol for RFID tags. In: Meghanathan, N., Kaushik, B.K., Nagamalai, D. (eds.) CCSIT 2011. CCIS, vol. 133, pp. 431–437. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17881-8_41
6. Sundaresan, S., Doss, R., Zhou, W.L., Piramuthu, S.: Secure ownership transfer for multi-tag multi-owner passive RFID environment with individual-owner-privacy. Comput. Commun. **55**, 112–124 (2015)