Inclusion/Exclusion Protocol for RFID Tags

Selwyn Piramuthu

RFID European Lab, Paris, France &
Information Systems and Operations Management, University of Florida
Gainesville, Florida 32611-7169, USA
selwyn@ufl.edu

Abstract. It is not uncommon to encounter objects with several RFID tags. However, the tags on these objects are generally mobile and move from or to (or, both) the object. No existing RFID authentication protocol considers this scenario. Moreover, an authentication protocol in such a scenario has the potential to be vulnerable to relay attacks where a tag that is not present on the object may pretend to be present. We present an authentication protocol that facilitates inclusion as well as exclusion of RFID tags on an object while simultaneously providing immunity to relay attacks.

Keywords: RFID, tag inclusion/exclusion, authentication protocol.

1 Introduction

Objects with multiple RFID tags are not uncommon. An 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. To our knowledge, no existing RFID authentication protocol addresses this scenario, and there is a clear need for such protocols.

RFID tags broadcast information about tagged object to any reader with appropriate credentials without physical verification of the reader. Given security and privacy concerns, lightweight cryptographic protocols have been proposed that restrict when, how, and what communication occurs between RFID tags and reader. Although these protocols prevent most problems that are associated with secure and privacy concerns associated with communication between RFID tag and reader, relay attacks pose a dire threat.

Relay attacks occur when an adversary simply relays signals between honest reader and tag without modifying it in any way. Since the signal content is not modified by the adversary, almost all of the extant RFID cryptographic protocols are immune to such attacks. There have been several proposed protocols that purport to alleviate this problem for a single tag. We propose an authentication

N. Meghanathan et al. (Eds.): CCSIT 2011, Part III, CCIS 133, pp. 431–437, 2011.

protocol for inclusion/exclusion of RFID tags that also resists relay attacks. This protocol is an extension of those presented in Kapoor and Piramuthu (2008) and Piramuthu (2010) addressing some identified vulnerabilities.

This paper is organized as follows: The next section discusses relay attacks and their variants known as mafia attack and terrorist attack. Section 3 provides a sketch of the proposed protocol for multiple tags on an object. Section 4 provides a brief security analysis of the proposed protocol. Section 5 concludes the paper with a brief discussion.

2 Relay Attacks

The ISO air-interface protocol (e.g., ISO 14443) requires the tags to be within about 4 inches from the reader. This, in principle, would deter adversaries operating in-between a tag and a reader. However, exploiting or circumventing a weakness in authentication protocols is not the only means to compromise an RFID tag enabled system. Relay attacks are one such attack that does not require physical proximity of a valid tag and reader. An adversary places two devices - a ghost (or proxy) and a leech (or mole) - between a tag and a reader. The ghost relays the reader's signal to the leech, which is in physical proximity to the tag. To the tag, the leech is a valid reader. The adversary then relays messages between the tag and reader without necessarily exploiting any weakness in the authentication protocol. Examples of scenarios that could fall prey to this type of vulnerability include RFID-enabled credit card, building access card, passport, etc.

Pervasive computing has motivated interest in systems that can precisely determine the location of a mobile device. The integrity and privacy of a location-proving system are important to prevent dishonest provers from falsifying location as well as to prevent adversaries from learning or mimicking privileged location information. Although one could verify location through use of GPS coordinates (e.g., Denning and MacDoran, 1996), RFID tags do not lend themselves to such applications. Distance bounding protocols to prevent such distance fraud attacks can be broadly classified as two types, one based on measuring the signal strength and the other based on measuring the round-trip time between prover and verifier.

The proof based on measuring signal strength is not secure. An adversary can easily amplify signal strength as desired or use stronger signals to read from afar. For example, the maximum range of a Bluetooth device is about 10 meters which can be increased to about 100 meters (328 feet) by increasing the power. John Hering (2004) and his colleagues at Flexilis created the BlueSniper 'rifle' and used it to grab the phone book and text messages from a Nokia 6310i phone that was 1.1 miles away. This example illustrates that measuring signal strength does not prevent distance-based attacks. It has been shown that using only electronics hobbyist supplies and tools, a cheap (for about $100) skimmer can be built that can read RFID tags from a distance longer than their typical range (e.g., Kirschenbaum and Wool, 2006; Kfir and Wool, 2005).

The proof based on measuring the round-trip time relies on the fact that no information can propagate faster through space-time than light (Hancke and Kuhn, 2005). The adversary under such a scenario can claim only to be farther away from its current location by delaying the response. Since we are dealing with very small numbers, the verifier must be capable of precisely measuring the round-trip time. For most practical purposes, this also implies that processing delay at the prover's end must be negligible compared to propagation delay between prover and verifier. In addition to simple distance fraud attacks, two other types of attacks have been identified under such scenarios: mafia (man-in-the-middle) fraud, and terrorist fraud attacks (Desmedt, 1988).

The mafia fraud attack is where the adversary consists of a cooperating rogue prover ($\bar{T}$) and rogue verifier ($\bar{R}$) where ($\bar{T}$) interacts with the honest verifier (R) and ($\bar{R}$) interacts with the honest prover (T) as follows: $R - \bar{T} - \bar{R} - T$. I.e., the adversary relays signals between the verifier and prover as if they were in close proximity to each other. Since the adversary does not modify any of the signal it receives, no amount of secure encryption could prevent these types of attacks. Brands and Chaum (1994) presented a distance bounding protocol based on a series of rapid bit challenge-response iterations to determine the distance between the prover and the verifier based on round-trip times. The authenticity of the prover and verifier still needs to be done. Clearly, the prover needs additional hardware (e.g., gates, etc.) dedicated to this protocol.

The terrorist fraud attack is where a dishonest prover collaborates with the adversary to convince the honest verifier of its proximity. Here, although the prover and adversary cooperate, the adversary does not know the secret key of the prover. Clearly, if the adversary knows the secret key, it would be hard to distinguish it from the prover.

3 Protocol for Multi-tagged Object

The following notations are used throughout the paper:

- $N_T, N_R, N_R^{'}, N_P, N_T, r_A, r_B$: random l-bit nonce
- s_c, s_{c+1}: Tag's current and subsequent keys
- f_k', f_k: keyed (with key k) encryption function
- H_k: keyed (with key k) one-way hash function
- t_j: shared secret between tag$_i$ and TTP, Reader
- r_i: shared secret between Reader R_i and TTP

3.1 Inclusion/Exclusion of Tag(s)

This protocol (Figure 1) can be used for inclusion and exclusion of tags in a multiple-tagged object. We assume that a TTP mediates between the reader and tags in accomplishing this change in shared secret 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) were 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 is a single round protocol that has three main "loops." 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.

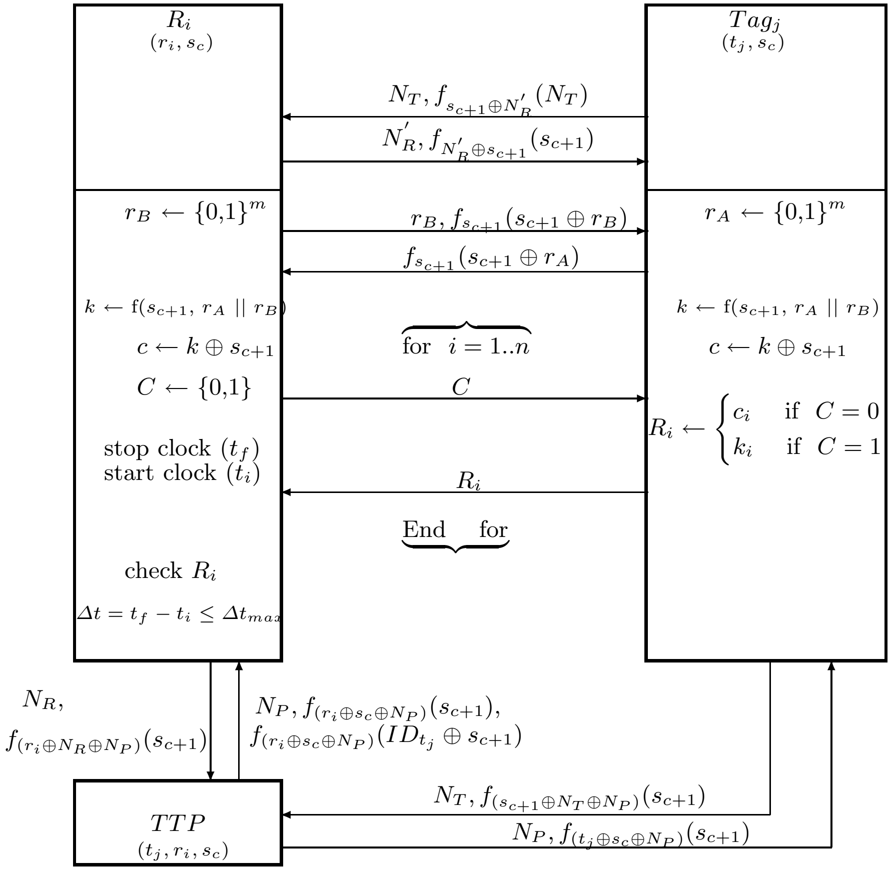
R_i
(r_i, s_c)
Tag_j
(t_j, s_c)
$N_T, f_{s_{c+1} \oplus N'_R}(N_T)$
$N'_R, f_{N'_R \oplus s_{c+1}}(s_{c+1})$
$r_B \leftarrow \{0,1\}^m$
$r_B, f_{s_{c+1}}(s_{c+1} \oplus r_B)$
$r_A \leftarrow \{0,1\}^m$
$f_{s_{c+1}}(s_{c+1} \oplus r_A)$
$k \leftarrow$ f$(s_{c+1}, r_A \mid\mid r_B)$
$\overbrace{\text{for} \quad i = 1..n}$
$k \leftarrow$ f$(s_{c+1}, r_A \mid\mid r_B)$
$c \leftarrow k \oplus s_{c+1}$
$c \leftarrow k \oplus s_{c+1}$
$C \leftarrow \{0,1\}$
C
$R_i \leftarrow \begin{cases} c_i & \text{if } C = 0 \\ k_i & \text{if } C = 1 \end{cases}$
stop clock (t_f)
start clock (t_i)
R_i
$\underbrace{\text{End} \quad \text{for}}$
check R_i
$\Delta t = t_f - t_i \leq \Delta t_{max}$
$N_R,$
$f_{(r_i \oplus N_R \oplus N_P)}(s_{c+1})$
$N_P, f_{(r_i \oplus s_c \oplus N_P)}(s_{c+1}),$
$f_{(r_i \oplus s_c \oplus N_P)}(ID_{t_j} \oplus s_{c+1})$
TTP
(t_j, r_i, s_c)
$N_T, f_{(s_{c+1} \oplus N_T \oplus N_P)}(s_{c+1})$
$N_P, f_{(t_j \oplus s_c \oplus N_P)}(s_{c+1})$

Fig. 1. Tag Inclusion/Exclusion Protocol

The first loop between the TTP and tag begins the shared key change process when the TTP generates a new shared secret for the tags and distributes this secret to each tag currently attached to the object. The TTP waits for response from the tag for a predetermined amount of time. If the TTP does not hear back from the tag during this time, it generates a fresh random nonce (N_P) and the forward part of the loop is repeated. This process continues until this loop is completed. After completion of the first loop, the tag attached to the object knows the new shared secret (s_{c+1}). For those tags that are no longer a part of the object of interest, the same process is followed but with s_{c+1} set to some predetermined string (e.g., null bits). The second loop is between the TTP and the reader. Here, the TTP shares the new shared secret key with the reader. This process is repeated for all relevant readers. The first term ($N_P, f_{(r_i \oplus s_c \oplus N_P)}(s_{c+1})$) is used to transfer the new shared key. The second term ($f_{(r_i \oplus s_c \oplus N_P)}(ID_{t_j} \oplus s_{c+1})$) conveys the unique ID value of the tag to the reader so that the reader can associate the ID value with its corresponding shared secret key value. This process is repeated until the TTP hears back from the reader. Once ID and s_{c+1} values are retrieved, the reader verifies the new common shared secret directly with the tag in the final loop. The third loop is between the reader and the tag. The third "loop" accomplishes two goals: mutually authenticating the tag using the new key (i.e., s_{c+1}) and then ensuring that this tag does indeed exist in the field of the reader. The former is initiated by the reader. The latter (distance bounding part) addresses issues related to relay attacks, and is adapted from Reid, et al. (2006).

The distance bounding part of the protocol operates through measuring the round-trip taken by the signal between reader and tag. In the proposed protocol, there is a need for the distance bounding part only between the reader and tags and not between the other two pairs of entities. The reader and TTP are generally assumed to be linked through a secure connection and, moreover, the distance between these are not of concern in most systems. Communication between the TTP and tags primarily involves generating and transferring the updated shared key from TTP to tags and the physical distance between TTP and tags is really not of concern for security/privacy purposes. However, the physical proximity of reader and tags is of concern since an adversary can initiate a relay attack to modify the physical distance between reader and tag(s) while still ensuring that the reader gets what it needs from the tag(s) for authentication purposes.

4 Security Analysis

1. Secrecy/Data Integrity and Authenticity:
   The cryptographic methods used (e.g., the function ensemble f_{s_i}) reasonably guarantees the secrecy of the message.
2. DoS/Synchronization problem:
   The DoS problem is addressed in the following way: Consider a situation where an adversary blocks a message. Since acknowledgements are expected for the key change and first post-key change communique between two entities, blocking any message creates no breach in the system.

3. Prevention of Mafia attack:
   Mafia attacks are prevented by using both timed and untimed phases, where the timed phase is used to verify distance between the tag and the reader and the un-timed phase is used to authenticate tag and reader. Thus, an adversary cannot respond to the reader in time, if the tag is farther away from the reader. The round trip times (Δt) are used to verify the distance between tag and reader.
4. Prevention of Replay attack:
   Replay attacks are prevented by generating fresh random nonce for every round of the protocol. Using freshly generated random nonce on both ends makes it hard to impersonate either the tag or the reader. In addition, the nonce is XORed with the secret keys to avoid revealing them to outside entities. Using a random nonce with every message in the protocol renders it difficult for an adversary to track the tag. Moreover, during the timed phase, the fast bit exchanges between the reader and tags are dependent on one another and therefore cannot be successfully recorded and replayed.
5. Prevention of Terrorist attack:
   Adapted from Reid et al. (2006), generating c (similarly, c') from both x and k prevents terrorist attacks by ensuring that the colluding tag does not share its secrets with an adversary. The secret (x) can be retrieved from simultaneous knowledge of both c and k.
6. Forward Security:
   This signifies that when the current key of a tag is known, it can be used to extract previous messages (assuming that *all* its past conversations are recorded). Most messages between any two entities are hashed or encrypted with freshly generated nonce in the ensemble. In the distance bounding part, to prevent a malicious reader from obtaining the c (or, c'_i) or k (or, k'_i) values by repeatedly sending the same R_i (or, R'_i) to a tag, the key update uses the complement of R_i (or, R'_i) that was transmitted to the reader or malicious adversary as the case may be.

5 Discussion

Inclusion/exclusion of RFID tags on any given (composite) object is not uncommon and there is an urgent need for authentication protocols that consider this scenario. The protocol presented in this paper purports to fill this gap, and is only a first attempt.

Relay attacks are difficult to prevent since these attacks do not depend on cryptography. Moreover, these attacks are passive, and occur without the knowledge of the tag as well as the reader involved. Of the means that have been proposed in the literature thus far, the ones that seem promising are based on measuring the round-trip distance traveled by signals between tag and reader. We use one such method and seamlessly incorporate the same in developing the proposed inclusion/exclusion protocol.

References

1. Brands, S., Chaum, D.: Distance-Bounding Protocols. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 344–359. Springer, Heidelberg (1994)
2. Denning, D.E., MacDoran, P.F.: Location-Based Authentication: Grounding Cyberspace for Better Security. In: Computer Fraud & Security, pp. 12-16 (February 1996)
3. Desmedt, Y.: Major Security Problems with the 'Unforgeable' (Feige)-Fiat-Shamir Proofs of Identity and How to Overcome Them. In: Proceedings of the Securicom 88, 6th Worldwide Congress on Computer and Communications Security and Protection, pp. 147–159 (1988)
4. Hancke, G.P., Kuhn, M.G.: An RFID Distance Bounding Protocol. In: Proceedings of the IEEE/Create-Net SecureComm, pp. 67–73 (2005)
5. Hering, J.: The BlueSniper 'rifle.' presented at 12th DEFCON. Las Vegas (2004)
6. Kapoor, G., Piramuthu, S.: Protocols for Objects with Multiple RFID Tags. In: Proceedings of the Sixteenth International Conference on Advanced Computing and Communications (ADCOM), pp. 208–213 (2008)
7. Kfir, Z., Wool, A.: Picking Virtual Pockets using Relay Attacks on Contactless Smartcard Systems. In: Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm), pp. 47–58 (2005)
8. Kirschenbaum, I., Wool, A.: How to Build a Low-Cost, Extended-Range RFID Skimmer. Cryptology ePrint Archive: Report 2006/054 (2006)
9. Piramuthu, S.: Relay Attack-Resisting Inclusion/Exclusion Protocol for RFID. In: 2nd International Workshop on DYnamic Networks: Algorithms and Security (DYNAS), Bordeaux (2010)
10. Reid, J., Gonzalez Nieto, J.M., Tang, T., Senadji, B.: Detecting Relay Attacks with Timing-Based Protocols. Queensland University of Technology ePrint (2006), `http://eprints.qut.edu.au/view/year/2006.html`