Information Processing Letters 116 (2016) 618–622

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Security analysis of an RFID tag search protocol

Hoda Jannati [a,*], Behnam Bahrak [b]

[a] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
[b] Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

ARTICLE INFO

Article history:
Received 26 May 2015
Received in revised form 24 February 2016
Accepted 3 May 2016
Available online 9 May 2016
Communicated by L. Viganò

Keywords:
Tag search protocol
RFID security
Impersonation attack
De-synchronization attack
Location privacy
Analysis of algorithms

ABSTRACT

Over the past decade, tag search protocols have been suggested to efficiently acquire a specific RFID tag among a large group of tags by an RFID reader. For instance, in a warehouse, where there are thousands of packages each having an RFID tag attached, staffs may find specific packages using a reader that employs a tag search protocol. Although tag search protocols promise convenience, most of them can threaten the privacy of RFID tags in different ways. For instance, an attacker can impersonate a tag to replace it with another tag or can find the identity of a tag to track it. Recently, Sundaresan et al. have proposed an RFID tag search protocol based on 128-bit pseudo random number generators and exclusive-or operations which both can be easily implemented on low-cost RFID passive tags in EPC global Class-1 Gen-2 standard even for large-scale implementations. They claim that their protocol not only offers anonymity, location privacy and forward secrecy for the reader and the tag, but also resists against de-synchronization, replay and impersonation attacks. In this paper, we analyze the security of their proposed tag search protocol and show that the protocol is vulnerable to de-synchronization and impersonation attacks and also cannot provide location privacy for the tag.

1. Introduction

Radio Frequency Identification (RFID) is a wireless technology for the purposes of automatic identification of electronic tags physically attached to objects using an RFID reader [1]. Recently, RFID systems are widely employed in supply chain management, pharmacy management, library collection management, electronic payment systems, automatic toll collection, proximity cards, hospital patient care, container search within seaports and many more applications [2]. In all such applications, process for the authentication of RFID tags by an RFID reader is necessary to ensure the validity of the RFID tags when they appear in the vicinity of the reader [3,4].

In addition to the authentication process, an RFID reader must also be able to efficiently find out a specific tag among a large group of tags. However, in RFID authentication protocols, the reader is allowed to query only one tag at each session. Hence, the authentication process cannot support such a target efficiently since the reader has to check each item separately. Thus, utilizing the authentication process to find out a tag among a group of tags can be slow or impractical as the number of tags increases [5]. Many tag search protocols have been proposed to achieve an efficient solution for this problem [5–10]. In these tag search protocols, the reader broadcasts a query for a specific tag with a known identity in its field of operation. If the tag is in the vicinity of the reader, it will reply back.

The existing tag search protocols have been investigated from various viewpoints such as strength against impersonation attack, de-synchronization attack, and reply attack, anonymity, location privacy, and computational

[*] Corresponding author.
E-mail addresses: hodajannati@ipm.ir (H. Jannati), bahrak@ece.ut.ac.ir (B. Bahrak).

http://dx.doi.org/10.1016/j.ipl.2016.05.001
0020-0190/© 2016 Elsevier B.V. All rights reserved.

Table 1
Notations utilized to formulate the Sundaresan et al. tag search protocol.

S, R, T_j	Server, Reader, jth tag
TID_j	The unique identity for T_j
$(t_s)_j$	Secret key for T_j
rts_j	Shared secret key between T_j and R
rts_j^{-1}	The previous value of rts_j
id_j	Stores the pre-computed hashed value of TID_j as $id_j = H(TID_j \| (t_s)_j)$
$ctrmax_j$	The number of allowed searches for T_j
ctr_j	The current counter value for T_j
t_r	The pseudo-random number generated by the tag in the current session
r_r	The pseudo-random number generated by the reader in the current session
$(r_r^{-1})_j$	The pseudo-random number generated by R in the last successful session of searching for T_j
$\oplus$	The bitwise exclusive-or operation
$H(.)$	A one-way hash function
$PRNG(w)$	A pseudo-random number generator with seed w
$PRNG^m(.)$	Composing the function $PRNG(.)$ with itself for m times

costs. However, not all tag search schemes can achieve these security and privacy requirements [11–13]. For instance, Piramuthu [12] showed that the Zou's search protocol [6] is vulnerable to de-synchronization attack. Moreover, Safkhani et al. [13] showed that the Tan et al.'s search protocol [5] is vulnerable to id disclosure and traceability attacks. Furthermore, implementation of secure tag search protocols is costly in terms of resources and consumption power. Such protocols utilize hash functions which require 8000 to 10000 two-input NAND gate equivalents (GEs) for implementation. Hence, they are not applicable on the low-cost devices which have at most 2000 GEs available for security properties.

To this end, Sundaresan et al. proposed an efficient RFID tag search protocol which is highly constrained in computational resources, and is claimed to preserve the security requirements for the tag and the reader [14]. Their protocol relies only on 128-bit pseudo random number generators and exclusive-or operations for execution. Both operations are easily implemented on low-cost RFID passive tags that comply with the Electronic Product Code Class 1 Generation 2 (EPC-C1G2) standard [15] even for large-scale implementations [16]. Sundaresan et al. claim that their protocol is resistant against de-synchronization, replay and impersonation attacks and preserves anonymity, location privacy and forward secrecy for the reader and the tag.

In this letter, we analyze the security of the tag search protocol proposed by Sundaresan et al. and show that it has pernicious security vulnerabilities in hostile environments. In particular, an adversary is able to perform de-synchronization attack and impersonate the tag and the reader with a high probability of success. Moreover, we show that the protocol cannot provide location privacy for the RFID tags.

The rest of this paper is organized as follows: In Section 2, we briefly review the tag search protocol proposed by Sundaresan et al. Section 3 discusses the vulnerabilities of this protocol. And finally Section 4 concludes the paper.

2. Review of the Sundaresan et al. tag search protocol

There are three types of players in the protocol proposed by Sundaresan et al. [14]:

1. A server S;
2. A set of readers;
3. A set of tags.

In this protocol, each tag T_j has a unique identity TID_j, two secret keys $(t_s)_j$ and rts_j, a required number of allowed searches $ctrmax_j$ and a current counter value ctr_j which all these parameters are shared with the server S. After authenticating the reader R, the server S feeds R via a secure channel with information of X tags that it has permission to search. Finally, R has access to $id_j = H(TID_j \| (t_s)_j)$ (where $H(.)$ is a one-way hash function), rts_j, $ctrmax_j$ and ctr_j for each tag T_j of X tags.

At the end of each successful session performed between the reader R and the tag T_j two parameters rts_j and ctr_j are updated by R and the tag T_j. In order to prevent de-synchronization attack, the tag keeps the backup of its previous state rts_j as rts_j^{-1} too. Moreover, the tag T_j stores $(r_r^{-1})_j$ which is the pseudo-random number sent by the reader R in the last successful session to prevent replay attack. Table 1 lists the notations deployed for this protocol. Fig. 1 also shows the details of the interaction between the reader R and the tag T_k in the Sundaresan et al. protocol.

1. The reader R first checks the correctness of $ctr_j < ctrmax_j$. If it is not, the protocol aborts and the reader goes back to the server to renew the search access permission. Otherwise, R generates a pseudo-random number r_r, and computes $M_1 = id_j \oplus PRNG(rts_j \oplus r_r)$ and $M_2 = r_r \oplus rts_j \oplus id_j$. Then, it broadcasts M_1 and M_2 as the query for searching the tag T_j among all the tags in its field of operation.
2. After receiving M_1 and M_2 from the reader, each tag T_k which its current counter value (ctr_k) is smaller than its maximum counter value ($ctrmax_k$), computes $\beta = rts_k \oplus id_k$, checks $id_k = M_1 \oplus PRNG(rts_k \oplus M_2 \oplus \beta)$ and $(r_r^{-1})_k \neq M_2 \oplus \beta$. If both are valid, the tag T_k knows that the query is for itself, i.e., T_k is the tag T_j. Hence, in this case, the tag T_k generates a pseudo-random number t_r, computes $M_3 = rts_k \oplus PRNG(id_k \oplus t_r)$ and $M_4 = t_r \oplus rts_k \oplus id_k$ and sends the messages M_3 and M_4 as its reply to the reader R. Then, the tag T_k updates rts_k^{-1} to rts_k and rts_k to $PRNG(rts_k)$ as well as ctr_k is incremented by 1. The tag T_k also updates $(r_r^{-1})_k$ to $M_2 \oplus rts_k \oplus id_k$. But, if