Computer Standards & Interfaces 31 (2009) 648–652

Contents lists available at ScienceDirect

Computer Standards & Interfaces

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

Vulnerability of an RFID authentication protocol conforming to EPC Class 1 Generation 2 Standards

Daewan Han *, Daesung Kwon

National Security Research Institute, 161 Gajeong-dong, Yuseong-gu, Daejeon 305-350, Republic of Korea

ARTICLE INFO

Article history:
Received 9 July 2007
Received in revised form 20 May 2008
Accepted 28 June 2008
Available online 9 July 2008

Keywords:
RFID
EPCglobal
Authentication
Impersonation
Forgery
DoS attack

ABSTRACT

Recently, Chien et al. proposed an RFID authentication protocol, which consists of only the cyclic redundancy code (**CRC**) and the pseudo-random number generator (**PRNG**) [H. Chien, C. Chen, Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 Standards, Computer Standards & Interfaces, vol. 29, Elsevier, 2007, pp. 254–259]. They claimed that the protocol conforms to current EPC tags, and would be secure against all attacks on RFID systems. However, in this paper, we show that the protocol is not secure; firstly an attacker can impersonate a valid tag temporarily by a single eavesdropping. Secondly the attacker can forge a tag permanently by eavesdropping two consecutive sessions. Finally he can make a valid tag useless (DoS attack) by modifying the second attack slightly. The computational complexities of the attacks are so practicable that Chien et al.'s protocol cannot enhance the RFID security any more than the original EPC standard.

1. Introduction

1.1. RFID systems and standards

A Radio-Frequency IDentification (RFID) system is an automated identification technology in which a small transponder (tag), attached to a real world object, receives and responds to radio-frequency queries from a transceiver (reader). This technology has been already widely used in daily life for access control, various payment systems, electronic identification cards, and so on [7]. However, the most notable and revolutionary application of RFID system in near future will be the replacement of a current barcode system for supply chain management, inventory control, and anti-counterfeiting. The attractiveness of the RFID over the barcode is twofold. Firstly, unlike a barcode scanner, an RFID reader does not require line-of-sight or physical contact to scan an RFID tag. This feature reduces the cumbersome need for manual intervention in the scanning process. Secondly, an RFID tag assigns a unique serial number to an individual item, while a barcode typically specifies the type of product it is printed on. The unique identifier associated with an object can serve as a pointer to a database entry containing detailed history of the object. Thanks to these features; automated scanning and unique identification, the RFID promises fine-grained tracking of inventory on an unprecedented scale.

If RFID will be used globally for the above usage, worldwide standards are needed, and they are currently being developed by an organization called EPCglobal [3]. The most important evolving standard of this organization is the EPCglobal Class-1 Generation-2 UHF tag standard [4]. We call RFID tags compliant with this standard EPC tags. Since EPCglobal unifies the two biggest organizations, UCC and EAN, which are responsible for barcode technology in the U.S. and Europe, EPC tags seem certain to become *de facto* standard for low-cost RFID tags.

1.2. Security of RFID systems and related works

Although RFID systems have many benefits as described above, they have some problems on security and privacy, which prohibit more rapid and widespread deployments of them. We refer to [13] for more details on these issues.

To address these problems, numerous physical protections and logical protocols have been suggested [1,8]. Though physical protections such as the blocker tag [9] might be efficient solutions for some applications, they have limits for general and broad usage, hence did not attract much attention. On the contrary, logical protocols for RFID security have been intensively studied in various directions.

In the early days several protocols using cryptographic hash functions and block ciphers were suggested. The representatives of this group appeared in [5,13], and see also [1,8] for surveys on numerous protocols designed in this fashion. However, this approach was proven to be inadequate for RFID security, since currently used hash functions and block ciphers are too expensive to be operated in EPC tags [6]. Another approach was to design protocols by using very light-weight logics such as bitwise Boolean operations or binary matrix operations [10–12]. Unfortunately, almost all these protocols

* Corresponding author.
E-mail addresses: dwh@ensec.re.kr (D. Han), ds_kwon@ensec.re.kr (D. Kwon).

0920-5489/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2008.06.006

were proved insecure. Final and recent approach is to design more EPC-friendly protocols, that is, to design protocols without varying existing standard specifications so much. One of them pursing this approach is Chien and Chen's protocol which is our interest in this paper [2]. We will call this protocol CC-RAP (Chien and Chen's RFID Authentication Protocol) hereafter.

1.3. Our contribution

CC-RAP is based on the cyclic redundancy code (**CRC**) and the pseudo-random number generator (**PRNG**) which are supported on current EPC tags. Thus, if CC-RAP should give a reasonable security to EPC tags, it will be the breakthrough in the area of RFID security; it means that we can enhance the security of EPC systems practically without additional expenses and cumbersome efforts for standardization.

Naturally, Chien et al. claimed that their protocol would be secure against all possible attacks against RFID systems. However, the main primitive of CC-RAP is a **CRC** function, which is well known to be linear in itself. As we will see in the next section, the security of CC-RAP heavily depends on the expectation that **CRC** will be a one-way function in partially varied inputs, which is wrong for the linearity of the function. Thus, we had convinced that CC-RAP is insecure, which was the motivation of this paper, and the anticipation turned out true from more detailed analysis on the protocol.

In this paper, we show that CC-RAP is not any more secure than current EPC standard; firstly an attacker can impersonate a valid tag temporarily by a single eavesdropping (or, an active query to a tag) with a few off-line **CRC** calculations. Secondly the attacker can clone a valid tag by eavesdropping two consecutive sessions with also practical amount of calculations. Finally he can make a valid tag useless (DoS attack) by modifying the second attack slightly. The most remarkable point of our attacks is that they can be accomplished passively and very easily, while several other attacks against RFID authentication protocols need active queries to tags or special devices which can operate between a tag and a reader during their communications.

The rest of this paper is organized as follows. We introduce CC-RAP briefly in Section 2. We present our attacks and analyze their practicability in Section 3, and conclude this paper in Section 4.

2. Chien et al.'s RFID authentication protocol

We briefly introduce CC-RAP in this section. The assumptions on the structure of RFID systems and the security model are referred to original paper [2]. The protocol consists of two phases; the initialization phase and the authentication phase.

For each tag T_x, the server randomly selects an initial authentication key K_{x_0} and initial access key P_{x_0}. The server initially stores three values (EPC$_x$, K_{x_0}, P_{x_0}) in the tag, where EPC$_x$ is the EPC code of the tag. The authentication key and the access key will be updated after each successful authentication, and those after the i-th successful session are denoted by K_{x_i} and P_{x_i}, respectively. For each tag, the server also maintains in its database a record of six values:

1) EPC$_x$
2) K_{old}: the old authentication key for this tag, initially set to K_{x_0}.
3) P_{old}: the old access key for this tag, initially set to P_{x_0}.
4) K_{new}: the new authentication key, initially set to K_{x_0}, too.
5) P_{new}: the new access key, initially set to P_{x_0}, too.
6) DATA: all the other information about the tagged object

After initialization, the reader and the tag can perform authentications, and the $(i+1)$-th authentication between the tag (T_x) and the server (S) via the reader (R) is described as follows.

(1) $R \rightarrow T_x : N_1$

The reader sends a random nonce N_1 as a challenge to the tag.

(2) $T_x \rightarrow R \rightarrow S : M_1, N_1, N_2$

The tag generates a random number N_2, computes

$$M_1 = \mathbf{CRC}(\text{EPC}_x |\ N_1 |\ N_2) \oplus K_{x\_i},$$

and sends the values (M_1, N_1, N_2) to the reader, which forwards these values to the server.

When the server receives the authentication request from the reader, it iteratively picks up an entry (EPC$_x$, K_{old}, K_{new}, P_{old}, P_{new}, DATA) from its database, computes the values

$$I_{\text{old}} = M_1 \oplus K_{\text{old}} \text{ and } I_{new} = M_1 \oplus K_{\text{new}},$$

and checks whether any two equations

$$I_{\text{old}} = \mathbf{CRC}(\text{EPC}_x |\ N_1 |\ N_2) \text{ and } I_{\text{new}} = \mathbf{CRC}(\text{EPC}_x |\ N_1 |\ N_2)$$

hold. The process is iteratively repeated for each entry until it finds a match. If it can find a match, then the authentication of the tag succeeds, and the server performs the next step; otherwise, it sends a “failure” message to the reader to stop the process.

(3) $S \rightarrow R : M_2, \text{DATA}$

If the server successfully authenticates the tag in the previous step, it computes

$$M_2 = \mathbf{CRC}(\text{EPC}_x |\ N_2) \oplus P_{\text{old}} \text{ or } M_2 = \mathbf{CRC}(\text{EPC}_x |\ N_2) \oplus P_{\text{new}},$$

depending on which value K_{old} or K_{new} satisfies the verification equation in the previous step. It also updates K_{old}, P_{old}, K_{new} and P_{new} as

$$K_{\text{old}} = K_{\text{new}}, P_{\text{old}} = P_{\text{new}}, K_{\text{new}} = \mathbf{PRNG}(K_{\text{new}}), P_{\text{new}} = \mathbf{PRNG}(P_{\text{new}}).$$

The server then sends (M_2, DATA) to the reader.

(4) $R \rightarrow T_x : M_2$

The reader retrieves the product information DATA and forwards M_2 to the tag. Upon receiving M_2, the tag verifies whether the equation

$$M_2 \oplus P_{x\_i} = \mathbf{CRC}(\text{EPC}_x |\ N_2)$$

holds. If so, it updates its keys as

$$K_{x\_i+1} = \mathbf{PRNG}(K_{x\_i}) \text{ and } P_{x\_i+1} = \mathbf{PRNG}(P_{x\_i}).$$

This authentication procedure is depicted in Fig. 1.

3. Vulnerability of Chien et al.'s protocol

3.1. Definitions, notations, and assumptions

Let A be the m-bit string in $\{0,1\}^m$. We denote the i-th less significant bit of A by A_{i-1}. That is, $A = A_{m-1} || A_{m-2} ||^{\cdots}|| A_0$. We define $A_{\ll n}$ as the $m+n$-bit string A' which is left-shift of A by n-bit, that is, A'_i is defined by

$$A'_i = \begin{cases} 0 & \text{for } 0 \le i \le n-1, \\ A_{i-n} & \text{for } n \le i \le n+m-1. \end{cases}$$

Let B be another n-bit string, where n is smaller than or equal to m. Then, we define m-bit string $A \oplus B = B \oplus A$ as follow:

$$(A \oplus B)_i = \begin{cases} A_i \oplus B_i & \text{for } 0 \le i \le n-1, \\ A_i & \text{for } n \le i \le m-1. \end{cases}$$

Then, the set of bit strings $\{0,1\}^\infty$ forms a group under the operation ⊕. Let F_2 be the binary field and $F_2[x]$ be the ring of