Chapter 1
An Improved Lightweight RFID Authentication Protocol

Xiaowen Zhang
College of Staten Island – CUNY, USA

Zhanyang Zhang
College of Staten Island – CUNY, USA

Xinzhou Wei
New York City College of Technology – CUNY, USA

ABSTRACT

This study extends the vulnerability analysis of a RFID authentication protocol and offers solutions to security weaknesses through enhanced measures. Vajda and Buttyan (VB) proposed a lightweight RFID authentication protocol, called XOR. Defend, Fu, and Juels (DFJ) analyzed it and proposed repeated keys and nibble attacks to the protocol. In this paper, we identify the source of vulnerability within VB's original successive session key permutation algorithm. We propose three improvements, namely removing bad shuffles, hopping the runs, and authenticating mutually, to prevent DFJ's attacks, thereby significantly strengthening the security of the protocol without introducing extra resource cost.

1. INTRODUCTION

As a consequence of the massive deployment of Radio Frequency Identification (RFID) systems in a variety of applications, security and privacy issues are still paramount concerns. Some consumer rights protection organizations, like CASPIAN (Consumers Against Supermarket Privacy Invasion and Numbering), are against the use of RFID (CASPIAN, 1999).

In general, a RFID system consists of three kinds of components: RFID tags (or transponders), RFID readers (or interrogators), and backend computer servers. An RFID tag is a tiny microchip embedded with a radio frequency antenna. It is capable of emitting the identification and other information for the tagged item. A reader is another electronic device located between tags and backend server. A reader receives information from or sends information to a tag, which in turn

DOI: 10.4018/978-1-4666-2773-4.ch001

communicates with (updates) the backend server. A backend server runs application software, hosts databases, and processes tag information received from a reader. It communicates with readers through wireless or wired connection on one end and with the enterprise network infrastructure on the other end. The wireless communication links between tags and readers are considered the most vulnerable part in terms of security and privacy threats. As documented in the literatures (Avoine, 2005; Juels, Pappu & Parno, 2008; Sarma, Weis & Engels, 2002), RFID security experts have devoted much effort to address these threats. Among them, new RFID authentication protocols and analysis are most active areas of research (Chatmon, Le & Burmester, 2006; Gilbert, Robshaw & Sibert, 2005; Juels, 2004; Juels & Weis, 2005; Le, Burmester & Medeiros, 2007; Li & Deng, 2007; Peris-Lopez et al., 2006).

Adding security features to low-cost RFID tags is a daunting and challenging task because these tags are extremely resource limited and cannot afford strong cryptographic algorithms. Practical RFID authentication protocols should have the following characteristics: lightweight, anonymity (un-traceability), and mutual authentication.

Vajda and Buttyan (Vajda & Buttyan, 2003) proposed a set of five lightweight authentication protocols and also gave a brief analysis. Each protocol is extremely lightweight in terms of resources required, and is considered suitable for resource limited devices, like RFID tags.

Defend, Fu, and Juels (Defend, Fu & Juels, 2007) performed cryptanalysis on two of them – XOR and SUBSET. DFJ proposed repeated keys and nibbles attacks in an attempt to compromise the XOR protocol. In this paper, we identify the source of vulnerability that existed in VB's original successive session key permutation algorithm. We propose three improvements, i.e. removing bad shuffles, hopping the runs, and authenticating mutually, to prevent DFJ's attacks, thereby significantly improve the security strength of the protocol without introducing extra resource cost.

2. ORIGINAL XOR PROTOCOL AND REPEATED KEYS ATTACK

The original XOR protocol by VB (Vajda & Buttyan, 2003) is a challenge-response protocol as shown in Figure 1. Under the following assumptions: (1) the readers and tags initially share a piece of secret key $k^{(0)}$, (2) both reader and tag are capable of calculating a permutation $\prod$ (given soon), and (3) reader and tag maintain a synchronized counter *i* to indicate the current run of authentication, the challenge-response process at the *i*th run can be described as follows:

Reader –> Tag: $a^{(i)} = x^{(i)} \oplus k^{(i)}$
// Reader picks a random number x(i), calculates $k^{(i)}$, then sends a challenge $a^{(i)} = x^{(i)} \oplus k^{(i)}$ to Tag.
Tag –> Reader: $b^{(i)} = x^{(i)} \oplus k^{(0)}$
// Tag calculates $k^{(i)}$, extracts the challenge $x^{(i)}$ by $k^{(i)} \oplus a^{(i)}$, then sends a response $b^{(i)} = x^{(i)} \oplus k^{(0)}$ to Reader. Following that, the Reader verifies the Tag, because only the Tag knows $k^{(0)}$.

Here $k^{(i)} = \prod(k^{(i-1)})$, and $\prod: \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a permutation starting from the initial secret key $k^{(0)}$. That is, $k^{(1)} = \prod(k^{(0)})$, $k^{(2)} = \prod(k^{(1)})$, …, $k^{(i-1)} = \prod(k^{(i-2)})$, $k^{(i)} = \prod(k^{(i-1)})$, …. Because $x^{(i)}$ is random, so are $a^{(i)} = x^{(i)} \oplus k^{(i)}$ and $b^{(i)} = x^{(i)} \oplus k^{(0)}$. If the $x^{(i)}$ is truly random, no information about the secret $k^{(0)}$ is revealed from the communication.

Suppose n = 128 bits as key length, the algorithm for the permutation $\prod$ is as follows:

Step 1: At the run (i-1)th iteration, the session key $k^{(i-1)}$ is split into 16 bytes, then we cut each byte into two nibbles of 4-bit each. Following that, we concatenate all left nibbles $k_{0,L}^{(i-1)}$, $k_{1,L}^{(i-1)}$, …, $k_{15,L}^{(i-1)}$ to form $k_L^{(i-1)}$, concatenate all right nibbles $k_{0,R}^{(i-1)}$, $k_{1,R}^{(i-1)}$, …, $k_{15,R}^{(i-1)}$ to form $k_R^{(i-1)}$.

Step 2: At the run (i)th iteration, the right half key $k_R^{(i)}$ is a permutation of $k_R^{(i-1)}$ controlled