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.

Keywords: Authentication Protocol, Radio Frequency Identification (RFID), Random Permutation, RFID Authentication, XOR Protocol

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 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

DOI: 10.4018/jal.2011010105
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 $\rightarrow$ Tag: $a^{(i)} = x^{(0)\oplus k^{(i)}}$

// Reader picks a random number $x^{(i)}$, calculates $k^{(i)}$, then sends a challenge $a^{(i)} = x^{(0)\oplus k^{(i)}}$ to Tag.

Tag $\rightarrow$ Reader: $b^{(i)} = x^{(0)\oplus k^{(0)}}$

// Tag calculates $k^{(i)}$, extracts the challenge $x^{(i)}$ by $k^{(i)} \oplus a^{(i)}$, then sends a response $b^{(i)} = x^{(0)\oplus k^{(0)}}$ 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 $x^{(0)}$ is random, so are $a^{(i)} = x^{(0)\oplus k^{(0)}}$ and $b^{(i)} = x^{(0)\oplus k^{(0)}}$. If the $x^{(0)}$ 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 $k^{(i-1)}_{0,R}$, $k^{(i-1)}_{L,R}$, ..., $k^{(i-1)}_{15,R}$ to form $k^{(i-1)}_L$, concatenate all right nibbles $k^{(i-1)}_{0,R}$, $k^{(i-1)}_{L,R}$, ..., $k^{(i-1)}_{15,R}$ to form $k^{(i-1)}_R$.

Step-2: At the run (i)th iteration, the right half key $k^{(i)}_R$ is a permutation of $k^{(i-1)}_R$ controlled by $k^{(i-1)}_L$: i.e., swapping the 0th and the $k^{(i-1)}_{0,L}$-th, the 1-st and the $k^{(i-1)}_{1,L}$-th, ..., the 15-th and the $k^{(i-1)}_{15,L}$-th nibbles of $k^{(i-1)}_R$.

Step-3: The left half key $k^{(i)}_L$ for run (i) is a permutation of $k^{(i-1)}_L$ controlled by $k^{(i-1)}_R$ in the similar nibble swaps.

Step-4: Finally the next run session key $k^{(i)}$ is obtained from rearranging (interleaving) the half bytes of $k^{(i)}_L$ and $k^{(i)}_R$, i.e.,
Related Content

Ordering Policy Using Multi-Level Association Rule Mining
www.irma-international.org/article/ordering-policy-using-multi-level-association-rule-mining/211220/

A Stochastic Perturbation Algorithm for Inventory Optimization in Supply Chains
www.irma-international.org/article/stochastic-perturbation-algorithm-inventory-optimization/4003/

Esin Okay (2017). *Ethics and Sustainability in Global Supply Chain Management* (pp. 178-196).
www.irma-international.org/chapter/a-new-barrier-for-the-future-of-energy-market-in-turkey/173946/

Internal Supply Chain Integration: Effective Integration Strategies in the Global Context
www.irma-international.org/chapter/internal-supply-chain-integration/48469/

Supply Chain Management in Retailing Business
www.irma-international.org/chapter/supply-chain-management-in-retailing-business/173947/