Comments on “Security Improvement of an RFID Security Protocol of ISO/IEC WD 29167-6”

Nasour Bagheri, Masoumeh Safkhani, Pedro Peris-Lopez, Juan E. Tapiador

***Abstract*—Song *et al.* recently scrutinized the security of a mutual authentication protocol published as a part of ISO/IEC WD 29167-6, denoted as Protocol 1, and showed its lack of protection against man-in-the-middle attacks [8]. In addition, they proposed an improved version, called Protocol 1+, and claimed resistance against this sort of attacks. Nevertheless, in this letter we show that the new protocol is as insecure as the original protocol against man-in-the-middle attacks.**

***Index Terms*—RFID, ISO/IEC WD 29167-6, Authentication, Attack.**

I. Introduction

Radio Frequency Identification (RFID) technology is a wireless identification method that uses radio frequency to send and receive data. Most of the RFID systems comprise of three components: tags, reader(s) and a back-end database. On the other hand, a passive tag is a highly constrained microchip with an antenna that stores an unique tag identifier and other linked information about the labelled item. To provide an adequate security level for such constrained devices, in the last years several ultralightweight RFID authentication protocols have been proposed, e.g., [3], [4], but all these schemes have flaws and vulnerabilities to a greater or lesser degree [2].

Regarding standardization, several standards have been adopted for diverse applications in different countries and unlikely there is no an universal standard. In UHF band, EPC Class-1 Generation-2 or equivalently ISO/IEC 18006 is one of the most widely standards [5]. In fact, ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) have contributed into the development of several international standards for internationally use-able frequencies for RFID systems. In this vain, ISO/IEC WD 29167-6 [7] has been recently proposed to strengthen the security of ISO/IEC 18000-6 standard. More precisely Part 6 (also known as 18000-6) defines parameters for air interface communications between 860 and 960 MHz to identify and manage items [6]. This standard describes three security protocols, denoted by **Protocol 1, 2** and **3** respectively. On the other hand, Song *et al.* recently analyzed the security of **Protocol 1** and showed that the proposed scheme is vulnerable to a man-in-the-middle (MITM) attack [8]. In addition, the authors proposed an improved version of protocol, named **Protocol 1+**, to overcome this security fault, while maintaining similar performance. Nevertheless, we show that they failed in their attempt and the improved protocol can also suffer a MITM attack. The success probability of the proposed attack is 1 while the attack complexity is just one execution of the protocol.

II. The Protocol Description

Step (0) to (10) of **Protocol 1+** [8] are exactly the same as step (0) to (10) of **Protocol 1** of ISO/IEC WD 29167-6 [7]. To facilitate reader understanding, all the steps of **Protocol 1+** are described below, where we denote a tag and a reader by T and R respectively. In addition, $E_k(m)$ symbolizes the encryption of a message m with the key k. The encryption process is an exclusive-OR (XOR) operation between a plaintext and a key-stream of the same bit-length.

0) R sends the $Select$ command to tags, to select a

Nasour Bagheri is with the Department of Electrical Engineering, Shahid Rajaee Teachers Training University, Tehran, Iran.

Masoumeh Safkhani is with the Department of Electrical Engineering, Iran University of Science and Technology (IUST), Tehran, Iran.

Pedro Peris-Lopez and Juan E. Tapiador are with COSEC Lab, Carlos III University of Madrid, Spain.

particular population of tags, based on the user-defined criteria.

1) R initiates an inventory round by sending a $Query$ command and decides which tags participate in the round among the tag population.
2) The selected tag T sends a 16-bit random number $RND16$ to R.
3) When R receives $RND16$, it replies an ACK command containing the same $RND16$.
4) Once T receives this message, it sends its protocol control (PC), extended protocol control (XPC) and void or random unique item identifier (UII), where PC contains physical-layer information in 16 bits, XPC indicates additional tag functions in 16 (or 32) bits, and UII is a code that identifies the tag holder.
5) R sends $Get\_Capabilities(RND16)$ to get T's capabilities.
6) In response to the command, T sends $Capabilities$, $CryptoFunc$ and $RND16$. $Capabilities$ indicates the support of crypto engine and file management services in 16 bits, and $CryptoFunc$ indicates in 16 bits the cryptographic suite and security functions that T supports.
7) R then replies $Sec\_Init(RnInt, RND16)$, where $RnInt$ is a 64-bit random number generated by R.
8) Upon its reception, T sends its security parameter ($SecParam$), key index (KI), $RnTag$ and $RND16$. $RnTag$ is a 64-bit random number generated by T. $SecParam$ indicates the support of the security mode and the master key of T and the word-length L of KI, in 16 bits. KI indicates in ($L \times 16$) bits the location of the master key in a database of the key pool that R stores. T then runs AES Engine, inputting ($RnInt, RnTag$) and keying the master key, and outputs a session key k of 128 bits.
9) When R receives T's response, it replies the second $ACK(RND16)$. R also runs AES engine, inputting ($RnInt, RnTag$) and keying the master key to generate a session key.
10) T encrypts a sequence of PC, XPC and UII with key k and sends it to R. The message is equal to $(PC \oplus k_1, XPC \oplus k_2, UII \oplus k_3)$, where k_i is the i-th block of key k and has the same bit-length as the message with which it is XORed.
11) R decrypts PC, XPC and UII from the received message. R then replies $Sec\_ReqRN(E_k(Len, ChInt << k_5, RND16 \oplus L_{16}(ChInt)))$, where Len is a 3-bit indicator of the word-length of $ChInt$ (one word is 16 bits), $ChInt$ is a random number generated by R, and $RND16$ is a 16-bit random number generated in step (2). The message is equal to $Sec\_ReqRN(Len \oplus k_4, (ChInt << k_5) \oplus k_6, (RND16 \oplus L_{16}(ChInt)) \oplus k_7))$, where k_i is the i-th block of the session key k and each of k_i has the same bit length as the message which it is XORed, excluding k_5 for which the bit-length is $log_2(Len \times 16)$. It should be noted that $x << y$ (resp. $x >> y$) means rotating all bits of x to the left (resp. right) by y bits.
12) T decrypts the message sent by R, and verifies whether the decrypted $RND16$ is correct. If so, T authenticates R and replies $E_k(ChTag << k_8, ChInt \oplus ChTag)$, where $ChTag$ is a random number generated by T in the same bit-length as $ChInt$, and the length of k_8 is $log_2(Len \times 16)$ bits. The message is equal to $((ChTag << k_8) \oplus k_9, (ChInt \oplus ChTag)) \oplus k_{10})$, where both k_9 and k_{10} have the same bit-length as $ChInt$.
13) R decrypts the message sent by T, and checks whether the decrypted $ChInt$ and $ChTag$ are correct. If they are correct, R authenticates T and also ensures that T has the same value of $Handle = L_8(ChInt) \| L_8(ChTag)$, which will be used as a secret parameter in the following access operation.

Step (11) to (13) of the original **Protocol 1** in ISO/IEC WD 29167-6 [7] are as follows:

11) R decrypts PC, XPC and UII from the message backscattered by T. R then sends $Sec\_ReqRN(E_k(Len, ChInt, RND16))$, where Len is a 3-bit indicator of the length of $ChInt$ in word size and $ChInt$ is a random number generated by R.
12) T decrypts the message sent by R, and checks whether the decrypted $RND16$ is equal to the initial $RND16$. If so, T authenticates R and replies $E_k(ChInt, Handle)$, where $Handle$ is a 16-bit random number generated by T used as a token

in the following access operation.

13) R decrypts the message sent by T, and checks whether the decrypted $ChInt$ is correct. If the check is valid, R authenticates T. Mutual authentication is completed and the channel is continuing. R will access T by sending access commands (e.g., $Read$, $Write$, $Kill$ and $Lock$) including $Handle$ as a parameter under encryption. If the decrypted $Handle$ is different from the original one, the channel is terminated.

It can be seen that **Protocol 1+** has three main differences in comparison to **Protocol 1**. The differences are outlined below:

1) The encrypted tokens in step (11) and (12) of **Protocol 1+** are a bit more complex than in **Protocol 1**. More precisely, **Protocol 1** only uses XOR operation (apart of the encryption engine) to generate the exchanged messages, while **Protocol 1+** employs XOR and rotation operations.

2) In **Protocol 1** the reader sends $Sec\_ReqRN(E_k(Len, ChInt, RND16)) = Sec\_ReqRN(Len \oplus k_4, ChInt \oplus k_5, RND16 \oplus k_6)$ to the tag and T only verifies the correctness of the recovered $RND16'$. If it passes the verification, then T accepts the recovered $ChInt'$ without any checking but this value could have been altered by the adversary during its transmission through the insecure radio channel. Similarly, when the tag sends $E_k(ChInt, Handle) = (ChInt \oplus k_7, Handle \oplus k_8)$ to the reader, R only verifies the correctness of the recovered $ChInt'$. If it passes the verification, then T accepts the recovered $Handle'$ without any checking again. Song *et al.* revised these messages in **Protocol 1+** such that the different message blocks sent by R (resp. by T) are correlated and also encrypted by k_i.

3) In **Protocol 1** the value of $Handel$ is only generated by the tag while in **Protocol 1+** this value is composed with random numbers $ChInt$ and $ChTag$ generated by R and T respectively.

In [8] Song *et al.* claim that these differences are enough to make the protocol secure against MITM attacks. Nevertheless in the next section we show how an attacker still can run a successful MITM attack against **Protocol 1+**.

III. MITM ATTACK ON **PROTOCOL 1+**

An adversary $\mathcal{A}$ is able to force R and T on sharing a different $Handle$ after a successful authentication. In particular, $\mathcal{A}$ does as follows:

0-10) When a session of **Protocol 1+** is executed, $\mathcal{A}$ eavesdrops the communications between R and T up to step (10).

11) When R sends $Sec\_ReqRN(Len \oplus k_4, (ChInt << k_5) \oplus k_6, (RND16 \oplus L_{16}(ChInt)) \oplus k_7)$ to the tag, $\mathcal{A}$ intercepts this message and replaces it by $Sec\_ReqRN(Len \oplus k_4, ((ChInt << k_5) \oplus k_6) \oplus \{1\}^{Len}, ((RND16 \oplus L_{16}(ChInt)) \oplus k_7) \oplus \{1\}^{16})$, where $\{1\}^{Len}$ is a Len-word of all-ones (one word is 16 bits).

12) T decrypts the message sent by R, and verifies whether the decrypted $RND16$ is correct. If the check is valid (which is the case here because $\mathcal{A}$ has not modified that fraction of the message originally sent by the reader and she preserves the correlation between the parts), T authenticates R and replies $((ChTag << k_8) \oplus k_9, ((ChInt \oplus \{1\}^{Len}) \oplus ChTag)) \oplus k_{10})$. $\mathcal{A}$ intercepts the message and replaces it by $((ChTag << k_8) \oplus k_9, (((ChInt \oplus \{1\}^{Len}) \oplus ChTag) \oplus k_{10}) \oplus \{1\}^{Len}) = ((ChTag << k_8) \oplus k_9, ((ChInt \oplus ChTag) \oplus k_{10}))$.

13) R decrypts the message sent by T, and checks whether the decrypted $ChInt$ and $ChTag$ are correct. If so, (which it is because the reader receives an unaltered message), R authenticates T and also ensures that T has the same value of $Handle = L_8(ChInt) \| L_8(ChTag)$, which will be used as a secret parameter in the following access operation.

Following the given attack, the reader generates $Handle = L_8(ChInt) \| L_8(ChTag)$ as a token for the following access operation, while the tag expects $Handle' = (L_8(ChInt) \oplus \{1\}^8) \| L_8(ChTag)$. As result, T will never accept the following access commands (e.g, $Read$ and $Lock$) sent by R for the entire duration of a tag access operation. Note that the success probability of the attack is 1 and only requires the execution of one protocol session.

The proposed attack is based on Theorem described below:

Theorem 1: Let A and B be L bit-strings. Then, $\forall\ n \in Z$+, $(A \oplus B)$ satisfies the following properties:

$$(A \oplus B) << n = (A << n) \oplus (B << n) \tag{1}$$

Proof: Let X_i the i-th bit of a L-bit string. Then $\forall j$ and being $C = (A \oplus B) << n$:

$$\begin{aligned} C_j &= A_{(j-n)mod\ L} \oplus B_{(j-n)mod\ L} \qquad (2) \\ &= (A << n)_j \oplus (B << n)_j \end{aligned}$$

■

Corollary 1: In particular, if in Eq. 1 we have $B = 0xFF...F$, then,

$$(A \oplus B) << n = (A << n) \oplus B \tag{3}$$

because $B << n = B$.

IV. Countermeasure

In the RFID context, there is a research area called ultra-lightweight cryptography. This sort of protocols are based on the use of simple and hardware efficient operations. In particular, bitwise operations (e.g., AND and XOR operations) and modular operations (e.g., additions) are commonly used. All these operations are Triangular functions (T-functions in short), which means that the i-th bit of output depends on $i = 0, 1, \cdots, i$ bits of input word and it does not depend on the more significant input bits. Exploiting this weak property, ultra-lightweight protocols are broken to a greater or lesser degree [2]. On the other hand, the use of non-triangular operations –mainly rotations– difficult the protocol analysis but it does not guarantee its security [1]. In fact, in our proposed attack an adversary exploits the non-resistance of XOR operation against active attackers and the weak property of rotation operation against ones-word ($X << N = X$, when $X = 0xFF..F$ and $\forall\ n \in Z$+).

Apart of this, the weak point in **Protocol 1+** (and **Protocol 1**) resides on the selected operation mode for the block cipher. The chosen mechanism is Output Feedback (OFB) mode. In this operation mode, the session key is computed taking the previous session key as input and the master key as a key. Besides an encrypted block is obtained by computing the XOR between the session key and the block message (i.e., $c_j = m_i \oplus k_i$). This encrypted message is sent over the insecure radio channel and an active attacker can alter it as shown in Section III.

As just mentioned the use of lightweight operators is not a strong mechanism to strengthen the security of a protocol. On the other hand, tags conforming ISO/IEC WD 29167-6 support on-board an AES cipher. Instead of using OFB mode, we recommend the use of Cipher-Block Chaining (CBC) mode with a secret IV –secret IV is a method for protecting the integrity of IV. Using this mode, the adversary does not have any control to manipulate the output of the cipher (i.e. $c_j = E_{key}(m_j \oplus c_{j-1})$).

V. Conclusion

ISO/IEC WD 29167-6 standard was proposed aiming to improve the security of ISO/IEC 18006. Unfortunately, **Protocol 1** suffers from MITM attacks. Song *et al.* proposed an improved schemed, called **Protocol 1+**. As shown in this letter, the new protocol is as insecure as its predecessors. An attacker can exploit the weak properties of T-functions and the non-resistance of the radio channel against active attack (i.e., bit manipulation). To overcome this weaknesses and taking advantage of having on-chip an AES engine, we propose to exchange OFB mode for CBC mode avoiding non-standard approaches.

References

[1] Z. Ahmadian, M. Salmasizadeh, and M. R. Aref. Recursive linear and differential cryptanalysis of ultralightweight authentication protocols. Cryptology ePrint Archive, Report 2012/489, 2012. http://eprint.iacr.org/.

[2] G. Avoine, X. Carpent, and B. Martin. Privacy-friendly Synchronized Ultralightweight Authentication Protocols in the Storm. *J. Network and Computer Applications*, 35(2):826–843, 2012.

[3] H.-Y. Chien. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity. *IEEE Trans. Dependable Sec. Comput.*, 4(4):337–340, 2007.

[4] M. David and N. Prasad. Providing strong security and high privacy in low-cost rfid networks. In *MobiSec*, volume 17 of *LNICST*, pages 172–179. Springer, 2009.

[5] K. Finkenzeller. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication. *Wiley & Sons*, 2010.

[6] Internation Organization for Standardization. ISO/IEC 18000-6, Information Technology Radio Frequency Identification for Item Management - Part 6: Parameters for Air Interface Communications at 860-960 MHz. Dec. 2010. http://www.iso.org.

[7] Internation Organization for Standardization. ISO/IEC WD 29167-6, Information technology - Automatic Identification and data Capture Techniques - Part 6: Air Interface for Security Services and File Management for RFID at 860-960 MHz. May 2010 - *currently withdrawn*. http://www.iso.org.

[8] B. Song, J. Y. Hwang, and K.-A. Shim. Security Improvement of an RFID Security Protocol of ISO/IEC WD 29167-6. *IEEE Communications Letters*, 15(12):1375–1377, 2012.